

ESCUELA POLITÉCNICA DEL EJÉRCITO

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**ESTUDIO DE LA METODOLOGÍA MIDAS Y DESARROLLO
DEL SISTEMA DE ADMINISTRACIÓN DE COMPETENCIAS
DE BÁSQUETBOL (BASYS) PARA LA FACULTAD DE
EDUCACIÓN FÍSICA, DEPORTES Y RECREACIÓN DE LA
ESPE**

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

**PUERTAS VÉJAR ANDREA ELIZABETH
ROBAYO TIPÁN FRANCISCO JAVIER**

SANGOLQUÍ, 30 de Agosto de 2006

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por los señores PUERTAS VÉJAR ANDREA ELIZABETH Y ROBAYO TIPÁN FRANCISCO JAVIER como requerimiento parcial a la obtención del título de INGENIEROS EN SISTEMAS E INFORMÁTICA

7 de Agosto de 2006

ING. TATIANA GUALOTUÑA

DEDICATORIA

“Todo lo puedo en Cristo que me fortalece”

(Filipenses 4:13)

Todo el esfuerzo, dedicación y empeño que he puesto en este trabajo, le doy gracias a Dios, quien siempre ha estado a mi lado y me ha dado sabiduría, conocimiento, las fuerzas y el apoyo necesario para terminar exitosamente este trabajo.

A mis queridos padres, por toda la entrega personal que siempre me han dado, porque han sido el ejemplo de mi vida y su amor me ha llenado. Y ahora que estoy culminando una meta más, se la entrego a ellos como un regalo de gratitud.

La escalera de la vida, no tiene fin, pero lo importante es subir peldaño a peldaño, con seguridad, no detenerse, vencer todos los obstáculos y confiar totalmente en Dios, quien es nuestra fortaleza.

*“Nadie que enciende una luz la cubre
con una vasija, ni la pone debajo de la cama,
sino que la pone en un candelero
para los que entran vean la luz.”*

(Lucas 8:16)

Andrea E. Puertas V.

DEDICATORIA

A mi Señor y mi Dios, Creador de Cielos y Tierra, por la vida, por el aire,
por el agua, por el fuego... por todo lo que me ha dado.

A los mejores padres que un hijo puede tener, Gerardo y Carmita,
son mis héroes, por su amor, cariño, entrega, ejemplo,
porque me dan fuerzas para continuar mi camino,
porque han hecho de mi una persona de bien...

Les dedico este trabajo y mi vida entera.

A los mejores hermanos y hermanas de todo el mundo,
por su apoyo incondicional, su cariño, su amor;
Danilo, Santiago, Rosita, Natalia y Carolina...

No lo habría logrado sin ustedes.

A Yuliana, Alexander y Domenique,
porque llenan mi corazón con solo su mirada,
porque son la alegría de mi vida,
porque por ustedes creo en el futuro.

Francisco J. Robayo T.

AGRADECIMIENTOS

Agradezco a Dios, porque me ha dado la oportunidad de vivir y ser feliz, porque ha estado conmigo en la realización de este trabajo, ha sido mi sustento, mi apoyo y mi fortaleza.

A mis padres, porque gracias a su esfuerzo, me han dado la oportunidad de estudiar, de prepararme para el presente y el futuro de mi vida; y me han enseñado valiosos valores de fe, amor e integridad que los he puesto en práctica y que estarán siempre presentes en el camino de mi vida.

A mi abuelito, porque es una persona muy especial, por sus sabios consejos y su aliento de cada día.

A mis hermanos Edú y Pao, por su amor constante, confianza plena y cooperación desinteresada.

A Francisco mi gran amigo y compañero de tesis, porque juntos con responsabilidad y dedicación pudimos realizar un maravilloso trabajo que representa la expresión misma de nuestros conocimientos y de nuestra formación profesional, que se manifiestan a través de nuestra Tesis de Grado.

A nuestra directora Ing. Tatiana Gualotuña y codirectora Ing. Jenny Ruiz, por su valioso tiempo, sus consejos profesionales y su gran aporte, que nos permitieron realizar de la mejor manera el presente proyecto.

Finalmente un agradecimiento especial a todos quienes conforman la Escuela Superior Politécnica del Ejército, mis profesores que han sido los gestores del conocimiento profesional, a mis compañeros por brindarme su amistad y su apoyo durante mi permanencia en la ESPE y a mis amigos por su entrega desinteresada y sincero aprecio.

A todos ellos, Muchas Gracias.

Andrea E. Puertas V.

AGRADECIMIENTOS

Doy gracias a Dios, porque ha sido mi luz en los momentos oscuros, mi guía en el camino angosto, mi fuerza en los momentos difíciles. Porque cuando veo solo un par de huellas en la arena, es Él quien me lleva en sus brazos... A Él debo todo lo que soy y todo lo que tengo.

A mi papito y mi mamita, por las noches en vela que han estado a mi lado, por el sacrificio que han hecho para sacarme adelante, por darme todo el amor del mundo, por darme la oportunidad de ser profesional y creer en mí. Solo puedo decirles, Dios les pague.

A mis hermanos, hermanas, sobrino y sobrinas, porque siempre han estado conmigo, en las buenas y en las malas, me dan aliento, alegría y fortaleza.

A Andreíta, mi amiga y compañera, porque solo juntando nuestros esfuerzos y conocimientos pudimos concluir esta etapa de nuestras vidas.

A mis amigos, porque los cinco somos una mano fuerte que abre todas las puertas, una mente clara que busca el mejor camino, porque son mis amigos.

A la Ing. Tatiana Gualotuña y la Ing. Jenny Ruiz por sus conocimientos, su tiempo y apoyo; a mis compañeros y profesores de la Escuela Politécnica del Ejército, y a todos quienes la conforman, porque cada día han aportado a mi crecimiento personal y profesional.

De todo corazón, muchas gracias.

Francisco J. Robayo T.

Índice de Contenidos

CAPITULO I	2
INTRODUCCIÓN	2
1.1- <i>Introducción</i>	2
1.2- <i>Justificación</i>	3
1.3- <i>Objetivos</i>	4
1.3.1- <i>Objetivo general</i>	4
1.3.2- <i>Objetivos específicos</i>	5
1.4- <i>Alcance</i>	6
CAPITULO II	7
MARCO TEÓRICO: CONTEXTO	7
2.1- <i>El Baloncesto</i>	7
2.1.1- <i>Definición</i>	7
2.1.2- <i>Dimensiones y Líneas</i>	7
2.1.3- <i>Desarrollo del Encuentro</i>	13
2.1.4- <i>Reglamentación</i>	15
2.1.5- <i>Violaciones a la reglamentación</i>	18
2.1.6- <i>Control del juego</i>	20
2.1.7- <i>Disposiciones generales</i>	22
2.2- <i>Evaluación del Jugador en los partidos de baloncesto</i>	23
2.2.1- <i>Efectividad del Jugador</i>	24
2.2.2- <i>Tiros de Campo</i>	25
2.2.3- <i>Rebotes</i>	26
2.2.4- <i>Tiros Libres</i>	26
2.2.5- <i>Faltas</i>	27
2.2.6- <i>Control del balón</i>	27
2.2.7- <i>Posesiones de balón</i>	28
2.2.8- <i>Normalización de datos estadísticos</i>	29
2.3- <i>La Facultad de Educación Física, Deportes y Recreación de la ESPE</i>	30
2.3.1- <i>Introducción</i>	30
2.3.2- <i>El básquetbol en la ESPE.-</i>	31

CAPITULO III	32
MARCO TEÓRICO: METODOLOGÍA	32
3.1- Arquitectura dirigida por Modelos MDA	32
3.1.1- Introducción	32
3.1.2- Modelo Independiente de Computación (CIM).....	35
3.1.3- Modelo Independiente de la Plataforma (PIM)	35
3.1.4- Modelo Específico de la Plataforma (PSM)	36
3.1.5- Otros conceptos en MDA.....	37
Nuevo proceso de desarrollo de sistemas.-	37
Plataforma.-	38
Metamodelado.-.....	39
MOF o Meta Object Facility.-	40
Perfiles UML.-	41
Object Constraint Lenguaje (OCL) .-	42
3.1.6- Transformaciones de modelos.....	42
Definiciones de Transformación.-	43
Tipos de definiciones de transformación.-.....	44
Características de las transformaciones.-	44
3.2- Metodología para el desarrollo de Sistemas de Información Web: MIDAS..	46
3.2.1- Introducción	46
3.2.2- Desarrollo ágil de Sistemas de Información Web	48
Las metodologías tradicionales vs. Las metodologías ágiles.-.....	48
Desarrollo ágil en la Web.-	50
3.2.3- El proceso metodológico de MIDAS.....	51
Modelos Independientes de Computación (CIM).-	58
Modelo de dominio.-	58
Modelo de negocio.-.....	59
Modelos Independientes de la Plataforma (PIM).-	60
Modelo de Servicios de Usuario.-.....	63
Modelo de Casos de Uso Extendido.-	63
Modelo de Composición de Servicios.-	65
Modelo de Fragmentos.-	66
Modelo de Fragmentos Extendido.-	66
Modelo de Navegación Extendido.-	67
Modelo de Servicios.-	67
Modelos Específicos de la Plataforma (PSM).-	68

Modelo Objeto Relacional.-	69
Modelo XML Schema.-	72
Modelo X-Link.-	75
Modelo WSDL.-	79
Modelo BPEL4WS.-	82
3.3- La tecnología .NET	84
3.3.1- Introducción	84
3.3.2- Web Services XML	85
3.3.3- El .NET Framework	86
Características de Common Language Runtime.....	89
Biblioteca de clases de .NET Framework.....	90
Código administrado en el servidor	91
3.3.4- Visual Basic.NET	92
CAPITULO IV	96
DESARROLLO DEL SISTEMA BASYS	96
4.1- Metodología MIDAS	96
4.1.1- Análisis del sistema	96
Especificación de Requerimientos de Software según IEEE 830	96
Introducción	96
<i>Propósito</i>	96
<i>Alcance</i>	96
<i>Visión General</i>	99
Descripción General	99
<i>Perspectiva del Producto</i>	99
<i>Funciones del Producto</i>	102
Requisitos Específicos	107
<i>Interfaces Externas</i>	107
<i>Requisitos funcionales</i>	108
<i>Requisitos del banco de datos lógicos</i>	140
<i>Requisitos de Rendimiento</i>	142
<i>Restricciones de Diseño</i>	142
<i>Atributos del Sistema de Software</i>	143
Determinación de costos del sistema	144
Costos de personal adicional	145
Costos generales	145

Costo total del proyecto	145
4.1.2- Diseño del Sistema	147
Modelos Independientes de Computación – CIM.....	147
Modelos Independientes de la Plataforma – PIM.....	149
Modelos Especificos de la Plataforma – PSM.....	254
4.1.3- Construcción del sistema	266
Generación de la Base de Datos.....	266
Diseño de la Interfaz de Usuario	267
Estructuración del Sistema.....	268
4.1.4- Pruebas del sistema	274
Desarrollo del protocolo de pruebas.....	274
Planificación de Prueba de Unidad	274
Planificación de Prueba de Integración	275
Planificación de Prueba de Aceptación	276
Planificación de Pruebas Generales del sistema	276
Ejecución del protocolo de pruebas	277
Pruebas de Unidad	277
Pruebas de Integración	281
Pruebas de Aceptación	282
Pruebas Generales del Sistema	283
 CAPITULO V	 286
CONCLUSIONES Y RECOMENDACIONES	286
5.1- Conclusiones	286
5.2- Recomendaciones	288
BIBLIOGRAFÍA	290
Referencias Bibliográficas	290
Referencias Electrónicas	291
GLOSARIO DE TÉRMINOS	294

Listado de Tablas

TABLA 3.1: COMPARACIÓN ENTRE PROCESOS ÁGILES Y DESARROLLOS WEB	51
TABLA 4.1: TABLA DE DEFINICIONES DE ACRÓNIMOS Y ABREVIATURAS	98
TABLA 4.2: COSTOS DE DESARROLLO DEL SIW. RECURSOS HUMANOS	144
TABLA 4.3: COSTOS DE HARDWARE Y SOFTWARE DE DESARROLLO.....	144
TABLA 4.4: COSTOS DE PERSONAL ADICIONAL	145
TABLA 4.5: COSTOS GENERALES.....	145
TABLA 4.6: RESUMEN DE COSTOS DEL PROYECTO	145
TABLA 4.7: PLAN DE PRUEBAS DE UNIDAD EN INICIO DE SESIÓN.....	274
TABLA 4.8: PLAN DE PRUEBAS DE UNIDAD EN ACTUALIZACIÓN DE PUNTOS ANOTADOS EN JUGADOR Y EQUIPO.....	275
TABLA 4.9: PLAN DE PRUEBAS DE INTEGRACIÓN	275
TABLA 4.10: PLAN DE PRUEBAS DE ACEPTACIÓN	276
TABLA 4.11: PLAN DE PRUEBAS GENERALES.....	276
TABLA 4.12: PRUEBA DE UNIDAD EN INICIO DE SESIÓN (CAJA BLANCA)	277
TABLA 4.13: PRUEBA DE UNIDAD, VALIDACIÓN DE DATOS (CAJA BLANCA)	277
TABLA 4.14: PRUEBA DE UNIDAD EN INICIO DE SESIÓN (CAJA NEGRA).....	278
TABLA 4.15: PRUEBA DE UNIDAD, VALIDACIÓN DE DATOS (CAJA NEGRA)	278
TABLA 4.16: PRUEBA DE UNIDAD EN ACTUALIZACIÓN DE PUNTOS ANOTADOS EN JUGADOR Y EQUIPO (CAJA BLANCA)	279
TABLA 4.17: PRUEBA DE UNIDAD, DETALLE DEL PROCESO (CAJA BLANCA).....	280
TABLA 4.18: ACTUALIZACIÓN DE PUNTOS ANOTADOS EN JUGADOR Y EQUIPO (CAJA NEGRA).....	280
TABLA 4.19: PRUEBA DE UNIDAD, DETALLE DEL PROCESO (CAJA NEGRA).....	281
TABLA 4.20: FORMULARIO DE REVISIÓN DE FUNCIONALIDAD.....	282
TABLA 4.21: PRUEBA DE RECUPERACIÓN	284
TABLA 4.22: PRUEBA DE SEGURIDAD.....	284
TABLA 4.23: PRUEBA DE STRESS	285

Listado de Gráficos

FIGURA 2.1: DIMENSIONES DEL TERRENO DE JUEGO	7
FIGURA 2.2: DISTANCIA MÍNIMA A LOS OBSTÁCULOS.....	8
FIGURA 2.3: PASILLOS DE TIROS LIBRES	9
FIGURA 2.4: DIMENSIONES DE LA ZONA DE BÁSQUET DE DOS PUNTOS	10
FIGURA 2.5: ZONAS DE BÁSQUET DE DOS Y TRES PUNTOS.....	10
FIGURA 2.6: POSICIONES DE LA MESA DE ANOTADORES	11
FIGURA 2.7: DIMENSIONES Y MARCAS REGLAMENTARIAS DE LOS TABLEROS	11
FIGURA 2.8: ARO REGLAMENTARIO	12
FIGURA 2.9: BALÓN DE BÁSQUETBOL	12
FIGURA 2.10: DISPOSITIVO DE 24 SEGUNDOS	15
FIGURA 2.11: MARCADOR.....	20
FIGURA 2.12: LENGUAJE DEL ÁRBITRO	22
FIGURA 3.1: PROCESO DE DESARROLLO CON MDA	33
FIGURA 3.2: LOGOTIPO DE MDA	33
FIGURA 3.3: NUEVO PROCESO DE DESARROLLO DE SISTEMAS CON MDA.....	37
FIGURA 3.4: CAPAS DEL METAMODELADO	40
FIGURA 3.5: DIAGRAMA RESUMEN SOBRE TRANSFORMACIONES	46
FIGURA 3.6: ORIGEN DE MIDAS	48
FIGURA 3.7: RELACIÓN EXISTENTE ENTRE EL ESPACIO DEL PROBLEMA Y EL ESPACIO DE LA SOLUCIÓN.....	52
FIGURA 3.8: CICLO DE VIDA DEL DESARROLLO DE SIW CON MIDAS	53
FIGURA 3.9: DIMENSIONES DEL PROCESO METODOLÓGICO DE MIDAS.	55
FIGURA 3.10: PROCESO DE DESARROLLO DE MIDAS.....	57
FIGURA 3.11: CLASE DE CLIENTE	59
FIGURA 3.12: DIAGRAMA DE CASOS DE USO SENCILLO	60
FIGURA 3.13: PROCESO DEL MODELADO DE HIPERTEXTO Y FUNCIONALIDAD.....	61
FIGURA 3.14: TAXONOMÍA DE CASOS DE USO	62
FIGURA 3.15: PROCESO DE DESARROLLO DE MIDAS. MODELOS ESPECÍFICOS DE LA PLATAFORMA.....	69
FIGURA 3.16: MODELO RELACIONAL Y MODELO OBJETO-RELACIONAL.....	72
FIGURA 3.17: EJEMPLO DE UN MODELO XML SCHEMAS	75
FIGURA 3.18: RELACIÓN ENTRE LOS ELEMENTOS DE WSDL.....	80
FIGURA 3.19: JERARQUÍA DE SCOPES	84
FIGURA 3.20: ARQUITECTURA DE .NET	87

FIGURA 3.21: .NET FRAMEWORK EN CONTEXTO.....	89
FIGURA 4.1: BANCO LÓGICO DE DATOS.....	140
FIGURA 4.2: MODELO DE DOMINIO	147
FIGURA 4.3: MODELO DE NEGOCIO	148
FIGURA 4.4: MODELO CONCEPTUAL DE DATOS.....	149
FIGURA 4.5: MODELO DE CASOS DE USO EXTENDIDO – USUARIO.....	156
FIGURA 4.6: MODELO DE CASOS DE USO EXTENDIDO – JUGADOR.....	156
FIGURA 4.7: MODELO DE CASOS DE USO EXTENDIDO – EQUIPO.....	157
FIGURA 4.8: MODELO DE CASOS DE USO EXTENDIDO – CAMPEONATO	157
FIGURA 4.9: MODELO DE CASOS DE USO EXTENDIDO – PARTIDO	158
FIGURA 4.10: MODELO DE CASOS DE USO EXTENDIDO – INFORMACIÓN DEL PARTIDO.....	158
FIGURA 4.11: MODELO DE CASOS DE USO EXTENDIDO – ESTADÍSTICAS	159
FIGURA 4.12: MODELO DE FRAGMENTOS	251
FIGURA 4.13: MODELO DE FRAGMENTOS EXTENDIDO	252
FIGURA 4.14: MODELO DE NAVEGACIÓN EXTENDIDO	253
FIGURA 4.15: MODELO OBJETO RELACIONAL	254
FIGURA 4.16: MODELO OBJETO-RELACIONAL EN SQL SERVER.....	267
FIGURA 4.17: PANTALLA DE USUARIOS EN VISTA DISEÑO	268
FIGURA 4.18: PROYECTOS DE LA SOLUCIÓN BASYS	269
FIGURA 4.19: PROYECTO FUNCIONALIDAD	269
FIGURA 4.20: PROYECTO HIPERTEXTO.....	271
FIGURA 4.21: REFERENCIAS WEB	272

Listado de Anexos

ANEXO A	298
MANUAL DE INSTALACIÓN	298
INTRODUCCIÓN	300
REQUERIMIENTOS MÍNIMOS DE INSTALACIÓN	300
<i>Requerimiento de Plataforma</i>	300
<i>Requerimientos de Bases de datos</i>	301
<i>Requerimientos de Infraestructura</i>	301
INSTALADOR DE BASYS V1.0	302
<i>Prerrequisitos para la instalación</i>	302
Levantamiento del IIS	302
.Net Framework 3.0	306
Windows Installer 3.1	308
<i>Instalación de MS SQL Server 2005</i>	310
<i>Configuración de la Base de Datos Basys</i>	321
Subir el backup de la base de datos	321
<i>Reportes de Basys</i>	326
<i>Basys Funcionalidad</i>	328
Personalización del archivo de configuración.....	331
<i>Basys Hipertexto</i>	332
ANEXO B	336
MANUAL DE USUARIO	336
INTRODUCCIÓN	338
FUNCIONAMIENTO	339
<i>Características comunes en Basys</i>	339
Campos Obligatorios	339
Campos NO Obligatorios	339
Restricciones en la eliminación de datos.....	340
Desplazamiento de la información.....	340
AMBIENTES DE BASYS	341
<i>Salir de la aplicación</i>	344
<i>Cambio de contraseña</i>	344

NAVEGANDO POR BASYS	345
<i>Opción Usuario</i>	345
Ingresar un nuevo Usuario.....	346
Seleccionar un Usuario.....	347
Modificar un Usuario	348
Eliminar un Usuario.....	348
Asignar el Perfil a un Usuario	348
<i>Opción Jugador</i>.....	350
Ingresar un nuevo Jugador	351
Buscador de Jugadores	352
Seleccionar un Jugador	353
Modificar un Jugador	353
Eliminar un Jugador	353
<i>Opción Equipo</i>.....	354
Ingresar un nuevo Equipo.....	355
Buscador de Equipos.....	356
Seleccionar un Equipo.....	356
Modificar un Equipo	356
Eliminar un Equipo.....	356
<i>Opción Campeonato</i>.....	357
Ingresar un nuevo Campeonato	358
Buscador de Campeonatos	359
Seleccionar un Campeonato.....	359
Modificar un Campeonato.....	359
Eliminar un Campeonato	359
<i>Opción Partido</i>	360
Ingresar un nuevo Partido.....	361
Buscador de Partidos.....	363
Seleccionar un Partido.....	363
Modificar un Partido	364
<i>Opción Información del Partido</i>.....	364
Ingreso de Jugadores e Inicio del Partido.....	366
Registro de los datos del partido	369
Cambio de un jugador.....	370
Estados del partido	370
Calcular las estadísticas	371

Opción Reportes	373
Partido.....	374
Estadísticas del Jugador por Partido	376
Estadísticas del Jugador por Campeonato	377
Estadísticas del Equipo por Campeonato.....	378
Jugador más valioso	379
Equipo más valioso.....	380
ANEXO C	381
CARTA DE ACEPTACIÓN DE LA FEFDER	381

Resumen

El Internet es hoy en día parte indisoluble de la vida de los seres humanos, ya que ha permitido romper las barreras que impone la geografía mundial, haciendo que mucha gente sea partícipe de sus beneficios. Por este motivo se ha popularizado en gran medida el desarrollo y consecuente uso de Sistemas de Información Web. En este ámbito se enmarca el presente documento, que profundiza el estudio de la Metodología ágil para el Desarrollo de Sistemas de Información Web, MIDAS.

Esta nueva metodología propone cambios que benefician el desarrollo de Sistemas Web, aprovechando las ventajas que provee la Arquitectura Dirigida por Modelos, los Servicios Web XML y la tecnología .Net.

Estos beneficios se aprovechan al desarrollar el Sistema para la Administración de competencias de Básquetbol (BASYS) para la Facultad de Educación Física, Deportes y Recreación (FEFDER) de la Escuela Politécnica del Ejército, utilizando MIDAS.

Siendo el baloncesto el deporte en el que más éxitos ha obtenido la ESPE, tanto a nivel provincial, como nacional e internacional, BASYS aporta a la FEFDER herramientas que ayudan en el proceso de control de jugadores, equipos, campeonatos y generación de estadísticas.

El análisis de MIDAS y su aplicación práctica, ha conducido a la obtención de conocimientos profundos en el desarrollo de Sistemas de Información Web.

CAPITULO I

INTRODUCCIÓN

1.1- Introducción

En la actualidad el desarrollo de Sistemas de Información Web (SIW), por el apogeo de Internet, se encuentra en auge. Existen muchas tecnologías y plataformas que permiten el desarrollo de dichos sistemas y aún mayor es el número de investigadores, docentes, empresas, organizaciones y personas vinculadas con el desarrollo que se han dedicado a implementar Sistemas de Información para la Web.

Se puede encontrar una cantidad inmensa de herramientas que permiten la implementación de SIW, pero muy poca información de procesos metodológicos que se orienten de forma estricta al desarrollo de SIW, y que permitan tener procesos de desarrollo integral de estos sistemas.

En este contexto se enmarca el presente proyecto, que se orienta al análisis de la Metodología para el Desarrollo de Sistemas de Información Web MIDAS, la misma que está basada en el marco de trabajo de la Arquitectura Dirigida por Modelos (MDA) que fue propuesta por la OMG (Object Management Group).

Además se plantea realizar el análisis, diseño, implementación, pruebas, instalación y puesta en marcha del Sistema para la Administración de competencias de Básquetbol (BASYS) que será desarrollado para la Facultad de

Educación Física, Deportes y Recreación (FEFDER) de la Escuela Politécnica del Ejército; aplicando para ello la metodología MIDAS.

El estudio de la metodología planteada, así como su aplicación práctica busca la obtención de conocimientos integrales en el desarrollo de SIW.

1.2- Justificación

La Facultad de Educación Física, Deportes y Recreación de la Escuela Politécnica del Ejército entre sus múltiples actividades, tiene a su cargo el control de diversos campeonatos de básquetbol, tanto dentro de la Escuela, como en los que participan las diferentes selecciones de la ESPE, y dicho control se ha venido realizando de forma manual, debido a que no ha existido el interés necesario de parte de autoridades y alumnos; por lo tanto el control del proceso se torna lento, tedioso e incluso ambiguo, por lo que no se puede llevar un control real de los datos, lo que provoca que no se entreguen resultados estadísticos precisos y los reportes necesarios con el tiempo y la exactitud requeridos.

Es importante señalar que el deporte en el cual más éxito ha tenido la Escuela Politécnica del Ejército a nivel Interinstitucional es precisamente el Básquetbol, en el cual se ha consagrado como penta campeón nacional y referente universitario del Ecuador, y por lo tanto es imperante administrar los datos que son generados por este deporte de la manera más adecuada posible.

Automatizar los procesos de control de los campeonatos de básquetbol permitirá llevar un historial de los campeonatos, control adecuado de equipos,

jugadores, y estadísticas de todos y cada uno de los partidos jugados. Se podrán obtener reportes ágiles y precisos cuando estos sean solicitados.

La metodología que se va a utilizar para el desarrollo del presente proyecto es una metodología ágil de reciente aparición, orientada al desarrollo de Sistemas de Información Web (SIW) que se denomina MIDAS, basada en la Arquitectura Dirigida por Modelos MDA (Model Driven Architecture) que fue propuesta por la OMG (Object Management Group),.

El estudio de dicha metodología y el desarrollo del sistema permitirán a los autores del presente proyecto la obtención de conocimientos actualizados en desarrollo de SIW, y a la Facultad de Educación Física, Deportes y Recreación de la ESPE mejorar importantes procesos de su sistema administrativo.

1.3- Objetivos

1.3.1- Objetivo general

Estudiar la metodología MIDAS de desarrollo de Sistemas de Información Web para la implementación de BASYS para la Facultad de Educación Física, Deportes y Recreación de la ESPE, con el fin de proveer de un sistema informático Web que apoye a la gestión de las competencias de básquetbol que controla la FEFDER.

1.3.2- Objetivos específicos

- Analizar la metodología ágil denominada MIDAS para desarrollo orientado a la Web, que permita desarrollar SIW de acuerdo a los requerimientos de desarrollo actuales.
- Utilizar la metodología MIDAS en el proceso de desarrollo del sistema de administración de competencias de básquetbol BASYS, para que el mismo cumpla con los requerimientos de gestión que necesita la Facultad de Educación Física, Deportes y Recreación de la ESPE, mediante la implementación de los módulos de Campeonatos, Equipos, Jugadores, y Estadísticas.
- Documentar todo el proceso de ingeniería de software para el desarrollo del sistema, de tal manera que se brinde el marco conceptual necesario para un manejo óptimo de la metodología y herramientas que se van a utilizar en la implementación e implantación del software.
- Realizar las pruebas que requiera el sistema, con el fin de garantizar la calidad del software.
- Capacitar en el correcto uso de BASYS a los usuarios finales del sistema, con la finalidad de conseguir los resultados que espera la FEFDER ESPE.

1.4- Alcance

El proyecto será desarrollado en su totalidad, siguiendo la metodología de desarrollo de Sistemas de Información Web MIDAS, usando una arquitectura n capas, de acuerdo a los módulos que se identifican y llegará hasta la instalación y puesta en marcha del sistema.

Los módulos a implementar son:

Módulo de Seguridad de acceso al sistema: Mediante el uso de usuarios y contraseña se controlará el ingreso al sistema en base a perfiles.

Módulo de Campeonatos: Se podrán registrar todas las competencias en la disciplina de baloncesto que la FEFDER quiera administrar, sean estas internas o en las cuales participen las diferentes selecciones de la ESPE.

Módulo de Equipos: Se mantendrá el control de todos los equipos que quieran registrarse en los campeonatos que organice la FEFDER o de las selecciones de basketball de la ESPE que participen en competencias organizadas por otras instituciones.

Módulo de Jugadores: Se llevará información de todos y cada uno de los jugadores registrados para competencias internas o externas.

Módulo de Estadísticas: Se recogerán estadísticas por jugador en todos y cada uno de los partidos jugados. Además se entregará los reportes necesarios por jugador, equipo y campeonato.

CAPITULO II

MARCO TEÓRICO: CONTEXTO

2.1- El Baloncesto

2.1.1- Definición

Es un juego en el que participan dos equipos, conformados por cinco jugadores cada uno, cuyo objetivo es introducir el balón dentro del cesto del contrincante, además de evitar que el mismo obtenga el balón y lo enceste. El equipo debe defender su canasta y atacar la canasta del contrincante.

Quien obtenga el mayor número de puntos al final del partido, será el ganador.

El balón puede realizar varios movimientos como: ser pasado, lanzado, palmeado, rodado o botado en varias direcciones dentro de las dimensiones del terreno de juego, que se van a detallar a continuación.

2.1.2- Dimensiones y Líneas

El terreno de juego debe tener una superficie rectangular, plana, dura y sin obstáculos. Sus dimensiones oficiales son de 28 metros de largo por 15 metros de ancho, estas medidas van desde el borde interior de las líneas que delimitan el terreno de juego, como se muestra en la *Figura 2.1*.

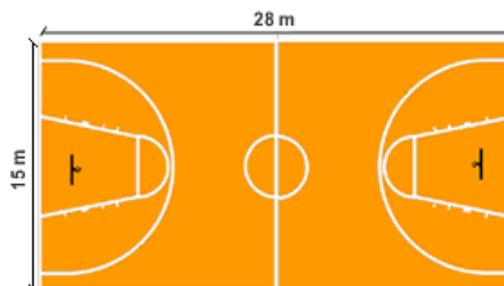


Figura 2.1: Dimensiones del terreno de juego

La altura del obstáculo más bajo o del techo, debe ser mínimo de 7 metros. Su iluminación debe ser uniforme de tal manera que no dificulte la visión de los jugadores y árbitros.

Las líneas del terreno de juego deben ser de un solo color (de preferencia blanco), ser completamente visibles y con un ancho de 5 cm. Estas líneas se dividen en dos: *líneas laterales (largo)* y *líneas de fondo (ancho)*. A estas líneas no se las considera dentro del terreno de juego. La mesa de anotadores, los banquillos de los equipos, carteles publicitarios y demás son considerados como obstáculos y deben estar a una distancia mínima de 2 metros de la cancha, como se muestra en la *Figura 2.2*.

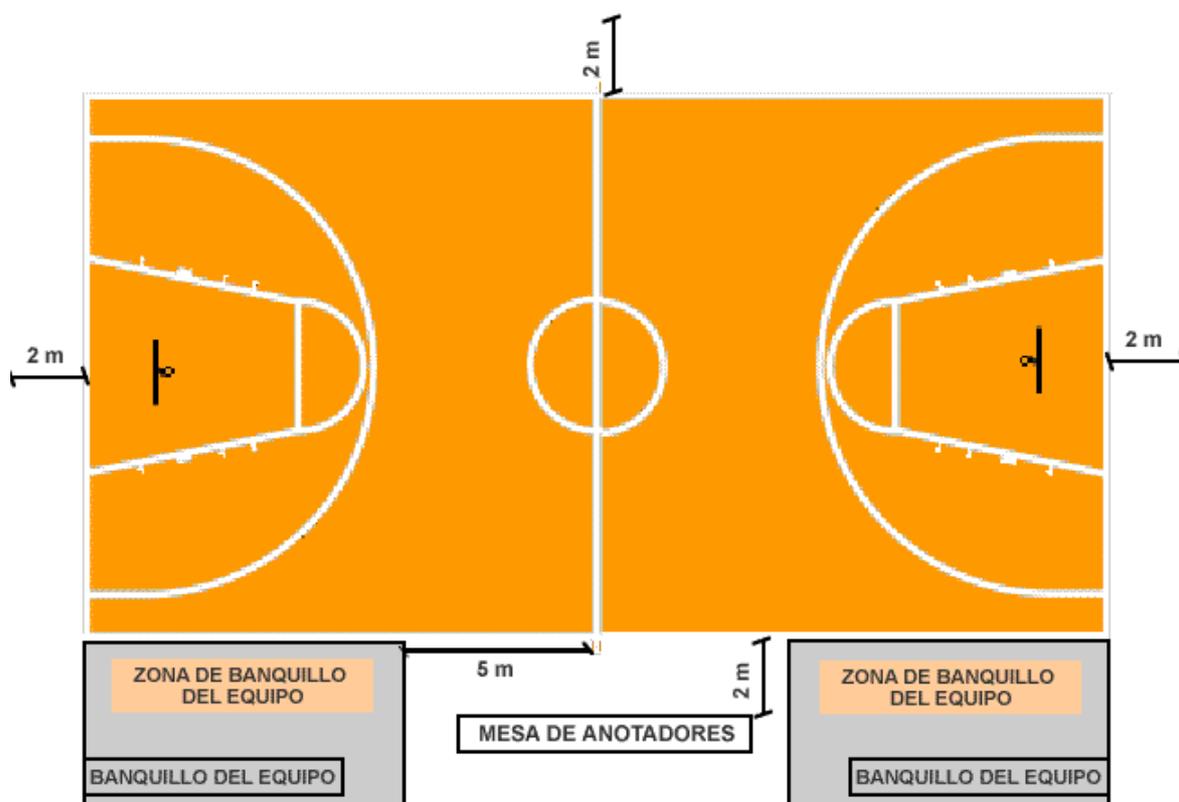


Figura 2.2: Distancia mínima a los obstáculos

La línea central es trazada paralelamente a las líneas de fondo, y sobresale de la parte exterior de la línea lateral 15 cm.

La línea de tiros libres debe estar a 5,80 metros de la línea de fondo y ser paralela a la misma, como se muestra en la *Figura 2.3*.

“Las áreas restringidas son los espacios marcados en el terreno de juego limitados por las líneas de fondo, las líneas de tiros libres y las líneas que parten de las líneas de fondo, tienen sus bordes exteriores a 3 m de los centros de las mismas y terminan en el borde exterior de las líneas de tiros libres.”¹

Los pasillos de tiros libres son áreas ampliadas por semicírculos con un radio de 1,8 metros y se trazarán también semicírculos iguales, pero con una línea discontinua en el interior de las áreas restringidas.

Los jugadores durante los lanzamientos de tiros libres pueden situarse a lo largo de los pasillos de tiros libres, y se marcan como se muestra en la *Figura 2.3*.

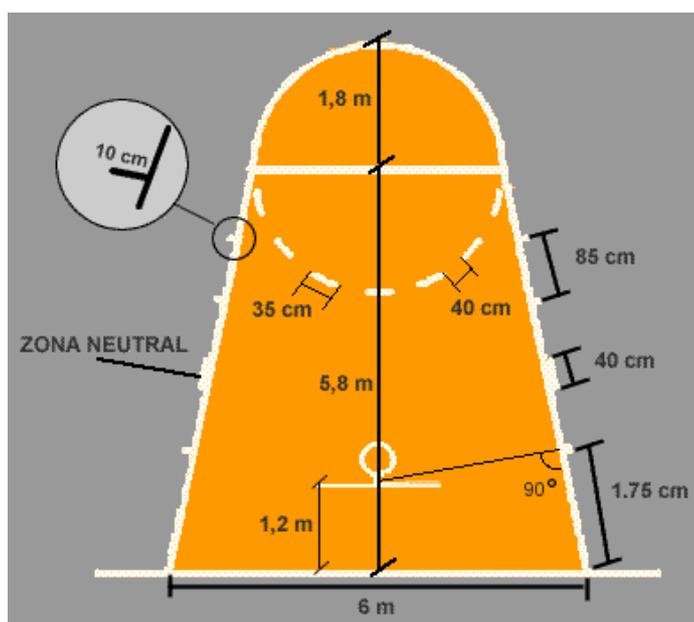


Figura 2.3: Pasillos de tiros libres

¹ Tomado de: <http://usuarios.lycos.es/basketball/balconcesto1.htm>

El círculo central debe ubicarse en el centro del terreno de juego y tener un diámetro de 3,60 metros. Si el interior del círculo es pintado, las áreas restringidas deberán tener el mismo color.

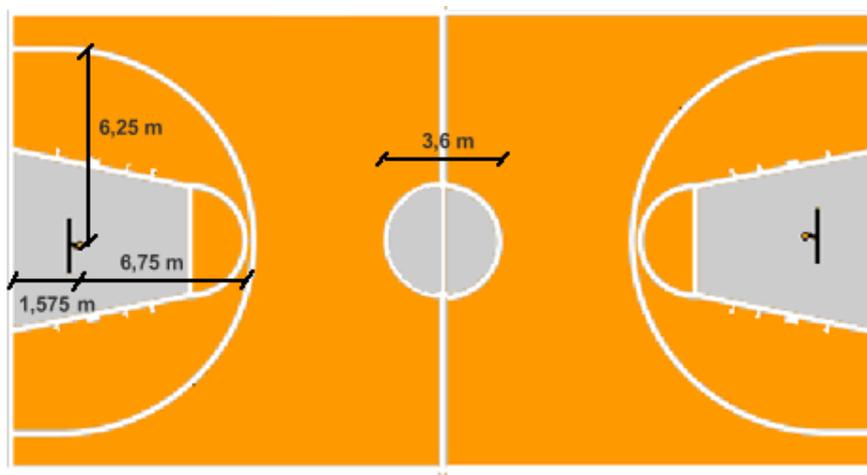


Figura 2.4: Dimensiones de la zona de básquet de dos puntos

La zona de básquet de tres puntos de un equipo es toda el área del terreno de juego, exceptuando el área limitada por un semicírculo próxima a la canasta de sus contrincantes, es decir la zona de básquet dos puntos, cuyas dimensiones se detallan en la *Figura 2.4* y las zonas de básquet de dos y tres puntos se muestran en la *Figura 2.5*.



Figura 2.5: Zonas de básquet de dos y tres puntos

Las posiciones de la mesa de anotadores y las sillas de los sustitutos son obligatorias para las principales competencias de la Federación Internacional de Baloncesto y son recomendadas para todas las competencias. Estas posiciones se detallan a continuación en la *Figura 2.6*.

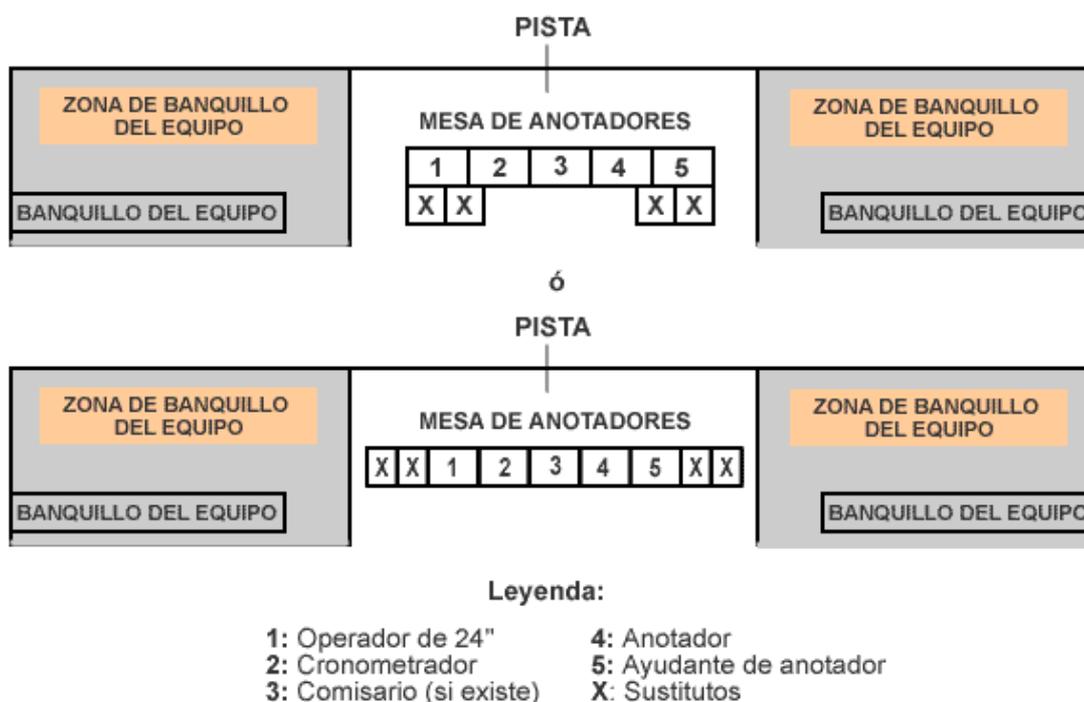


Figura 2.6: Posiciones de la mesa de anotadores

Los tableros deben ser de un material transparente adecuado, de una sola pieza y con un espesor de 3 cm. Si no fuera transparente, deberá ser pintado de color blanco. Las dimensiones del tablero se detallan en la *Figura 2.7*.

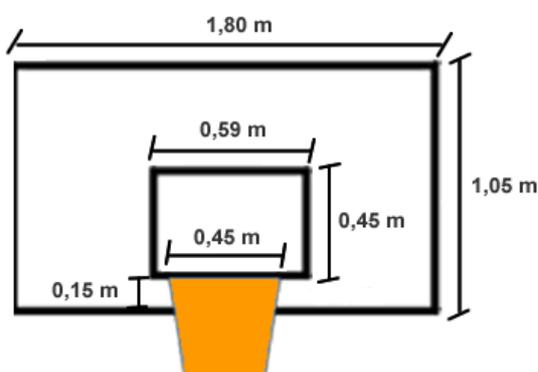


Figura 2.7: Dimensiones y marcas reglamentarias de los tableros

Las canastas se componen de un aro y una red. El aro debe tener un diámetro de 45 cm. y ser pintado de color naranja. La red debe ser una cuerda blanca y estar colgada en el aro, debe medir entre 40 cm. y 45 cm. de longitud. Cada red debe tener 12 bucles para sujetarse en el aro. Estas consideraciones se las puede observar en la *Figura 2.8*.

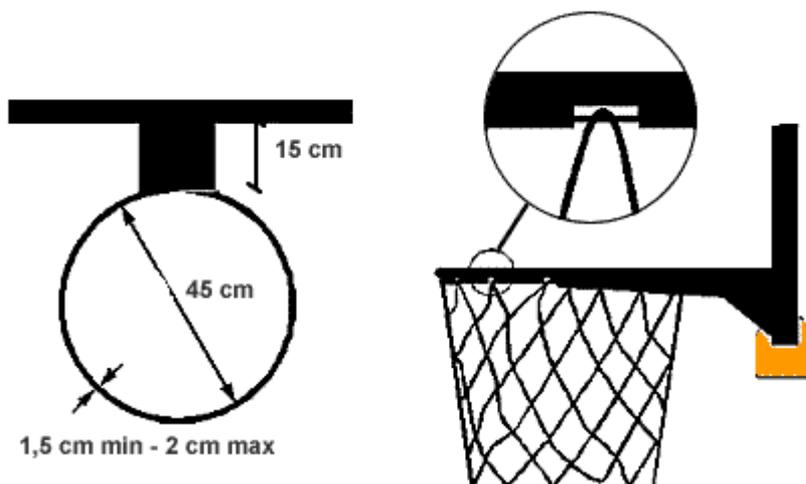


Figura 2.8: Aro reglamentario

El balón debe ser esférico, cuya circunferencia sea de 75 cm. y su peso debe fluctuar entre los 600 y 620 gramos. Su color debe ser naranja con ocho sectores de forma tradicional y juntas negras. Al dejarlo caer desde una altura de 1,80 metros, debe rebotar entre 1,20 a 1,40 metros sobre una superficie de madera sólida.



Figura 2.9: Balón de básquetbol

2.1.3- Desarrollo del Encuentro

El partido inicia en el momento en el que el árbitro lanza el balón al aire en el círculo central, donde los pivots saltan para intentar pasar el balón a uno de sus compañeros. El equipo que obtuvo el balón, trata de avanzar al área de su contrincante para tratar de encestar y conseguir dos o tres puntos dependiendo de la zona en la que se encuentre.

El pase y el regate.-

Un jugador en posición de ataque debe separar sus pies a la altura de sus hombros, con un pie ligeramente adelantado. Luego debe decidir si tiene posibilidad de encestar, y si los defensas están obstaculizando el tiro, si es así deberá pasar el balón a un compañero en mejor posición o en su defecto tratar de hacer un regate para tener una mejor posición.

El momento de realizar un pase, se tienen dos opciones:

- El *pase de pecho*, en el que la pelota se lanza a la altura del pecho, dando un paso para adelante.
- El *pase a dos manos*, que se lo puede lanzar igualmente a la altura del pecho o sobre la cabeza del contrincante.

El jugador que recibe el balón, no debe dar más de dos pasos antes de rebotar o regatear el balón. Cuando realiza una parada en salto, es decir cuando cae con los pies al mismo tiempo, puede girar en cualquier dirección sobre cualquiera de los dos pies.

El regate le permite a un jugador librarse del acoso de su contrincante. Existen dos tipos de regatees:

- *Regate de frente*, el jugador con las piernas semiflexionadas se inclina hacia un lado, pasa el balón a la mano contraria, cambia de dirección de forma no prevista y consigue escaparse de su contrincante, por otro lado.
- *Regate de espaldas*, el jugador consigue despistar a su contrincante con un giro, en el que al dar la vuelta, cambia el balón a la mano contraria y con su cuerpo protege el balón.

El ataque.-

Existen tres tipos de jugadores, los cuales son bases, aleros y pívots, que tienen diferentes papeles durante el partido y son:

- Los *bases*, son los más livianos y más pequeños cuya función es de organizar el ataque ya que tienen un gran dominio de las técnicas del pase y rebote.
- Los *aleros*, son los jugadores que avanzan en el ataque, por lo que se desplazan velozmente. Tienen gran dominio para rebotear y tienen gran habilidad para encestar desde cualquier ángulo.
- Los *pívots*, son los jugadores más altos y tienen gran habilidad en el tiro, y para moverse en pequeños espacios. Su función es desmarcar a los atacantes y tener una mejor condición para encestar.

La defensa.-

Existe un principio básico en la defensa y es que, se puede tocar a la pelota pero no al contrincante. Lo que trata la defensa es provocar que el contrincante falle. La mejor posición del defensa es mantener los pies separados, las piernas y caderas flexionadas y el cuerpo en tensión. Se ubicará de espaldas al cesto y

evitará que el atacante avance, si el mismo realiza un tiro, entonces el defensa levantará la mano para evitar que el balón pase.

2.1.4- Reglamentación

Del tiempo.-

Se puede tomar el tiempo de dos formas:

- Dividiendo al partido en dos períodos de 20 minutos
- Dividiendo al partido en cuatro períodos de 12 minutos, en donde existe un receso de 2 minutos entre el primer y segundo período y el tercer y cuarto período.

Para el receso intermedio se puede dar un tiempo de 10 ó 15 minutos.

El reloj del partido nos permite cronometrar los periodos del partido y sus intervalos, y debe estar situado en un lugar en el que sea visible para todas las personas interesadas en el partido.

La regla de los 24 segundos se la mide mediante un dispositivo que se muestra en la *Figura 2.10*. Esta regla consiste en que cuando un jugador consigue el control del balón en el terreno de juego, él y su equipo deben realizar un lanzamiento a la canasta antes de cumplirse 24 segundos.



Figura 2.10: Dispositivo de 24 segundos

Pueden existir varios tiempos muertos a lo largo del partido, que cada equipo puede hacer uso para diferentes fines. La disposición del número de tiempos muertos, depende del número de períodos del partido, para lo cual:

- Si el partido es de dos períodos, entonces cada equipo tiene derecho a dos en el primer período, tres en el segundo y uno en cada período extra.
- Si el partido es de cuatro períodos, cada equipo tiene derecho a tres tiempos muertos en cada mitad del partido y un tiempo para cada período extra.

Si un equipo no utilizó todos sus tiempos muertos, los puede acumular para la siguiente mitad o para los períodos extras.

Si el partido finalizó y los equipos se encuentran empatados entonces se tendrá que realizar periodos extras de 5 minutos cada uno, hasta que el marcador desempate. Se dispondrá de 2 minutos de receso entre cada periodo extra. La dirección del juego de cada equipo se mantendrá como la del último período. Cada período extra deberá comenzar con un salto entre los pivots de cada equipo, en el centro del terreno de juego.

Del juego.-

Para comenzar un partido los dos equipos deben estar presentes en el terreno de juego, con cinco jugadores cada uno. Se realizará el cambio de la canasta en la segunda mitad del partido.

Cuando existe un *salto entre dos jugadores*, los mismos se ubicarán en el semicírculo más cercano a su canasta y con un pie cerca del centro de la línea.

Luego el árbitro lanza el balón a una altura en la que los dos jugadores no alcancen saltando para permitir que el balón caiga entre los dos. Ninguno de los jugadores puede abandonar su posición hasta que uno de los dos palmeé legalmente el balón. El jugador que palmeó el balón no podrá cogerlo hasta que toque a cualquiera de los ocho jugadores restantes. Si el balón no fue palmeado antes de caer al piso, entonces se tendrá que repetir el salto.

El baloncesto es un juego que se juega únicamente con las manos. Si un jugador corre con el balón, lo golpea con alguna parte inferior a la rodilla o con el pie o el puño de forma intencional, se lo considera como una violación.

Un equipo encesta, cuando el balón ingresa por la canasta y permanece en ella o la atraviesa. Existen diferentes puntuaciones a los tiros realizados y son:

- *Un punto*, cuando se ejecuta un tiro libre
- *Dos puntos*, cuando encesta desde el terreno de juego
- *Tres puntos*, cuando encesta desde la zona de básquet de tres puntos.

Si un equipo encesta sin intención en su canasta, los puntos obtenidos se los acreditará al capital del equipo contrario. Si lo hace con intención, se lo considera como una violación y no es válido el cesto. Si un jugador encesta por debajo del canasto el balón sin intención, este no se lo considera válido y se realizará un salto entre dos jugadores.

Un jugador realiza un saque después de que ha enceestado su equipo, un tiro libre convertido, un tiempo muerto o cualquier interrupción luego de un cesto. Uno de los árbitros debe poner a disposición del jugador el balón e indicarle en que

punto debe realizar el saque. Para lo cual un jugador se ubica en el punto de la línea de fondo indicado anteriormente o detrás de ella y realiza el paso a un compañero que se encuentre en el terreno de juego. El jugador que va a realizar el saque tiene 5 segundos para efectuarlo, desde que obtiene el balón y sale del terreno de juego.

Cuando se ha realizado un tiro libre debido a una falta técnica o descalificante señalada a un entrenador, o a una falta antideportiva, técnica antideportiva o descalificante cometida por un jugador, no deberá realizar el saque en la línea de fondo, sino que lo realizará en el centro del terreno de juego, enfrente de la mesa de anotadores, haya o no encestado el tiro libre.

2.1.5- Violaciones a la reglamentación

Existen diferentes tipos de infracciones a las reglas. Cuando un equipo comete una violación pierde el derecho al balón, concediéndole al equipo contrario para que realice un saque desde el punto en el que lo señale uno de los árbitros.

Regla del avance.-

Cuando un jugador está con el balón en juego y con los dos pies en el suelo, puede realizar un *pivote*, que consiste en mantener un pie en el suelo en el mismo lugar (*pie de pivote*), y con el otro pie moverse en cualquier dirección dando uno o más pasos.

Un jugador no puede permanecer más de tres segundos en el área restringida o dentro de ella si su equipo está controlando el balón, para lo cual el jugador deberá salir completamente del área.

Si un jugador está tremendamente marcado y está controlando el balón, entonces deberá lanzarlo, pasarlo, hacerlo rodar o botarlo antes de que se cumplan cinco segundos.

El jugador que está controlando el balón en su pista trasera tendrá diez segundos para que con la ayuda de su equipo pase el balón a su pista delantera.

Un jugador que se encuentre con el balón en su pista delantera no podrá devolverlo a su pista trasera.

Faltas personales.-

Una falta que involucra el contacto personal con un contrincante o un jugador con una conducta antideportiva es anotada y penalizada posteriormente de acuerdo al reglamento.

Una falta personal se puede dar cuando el balón esté en juego o no, e implica que un jugador ha realizado un contacto ilegal contra su contrincante, ya sea al agarrarlo, bloquearlo, empujarlo, impedirle avanzar utilizando su mano, brazo, codo, hombro, cadera, pierna, rodilla o el muslo, ó cuando realiza un juego brusco o violento.

Siempre que exista una falta esta se la anotará al infractor y se considera diferentes penalizaciones en el momento en que se ejecuta la falta, que son:

- Si el jugador cometió la falta contra un jugador que no está en posición de tiro entonces se reanuda inmediatamente el juego mediante un saque del equipo contrincante desde el punto exterior más próximo a donde se cometió la falta.
- Si el jugador contrincante está en posición de tiro entonces se toman en cuenta tres consideraciones:

- Si el lanzamiento se ejecuta es válido y se le concede un tiro libre.
- Si el lanzamiento se lo realiza desde la zona de básquet de dos puntos no se convierte, entonces se le concede dos tiros libres
- Y si esta en la zona de tres puntos, se le concede tres tiros libres.
- Si el equipo del jugador que comete la falta tiene el control del balón, entonces el equipo contrario realizará un saque desde el punto exterior más próximo a donde se cometió la falta.

Marcador.-

En la *Figura 2.11*, se muestra el tablero que permite registrar el número de faltas que ha cometido un jugador y un equipo, los tiempos muertos de cada equipo, el período en el que se encuentra el partido, el tanteo y el reloj del mismo.

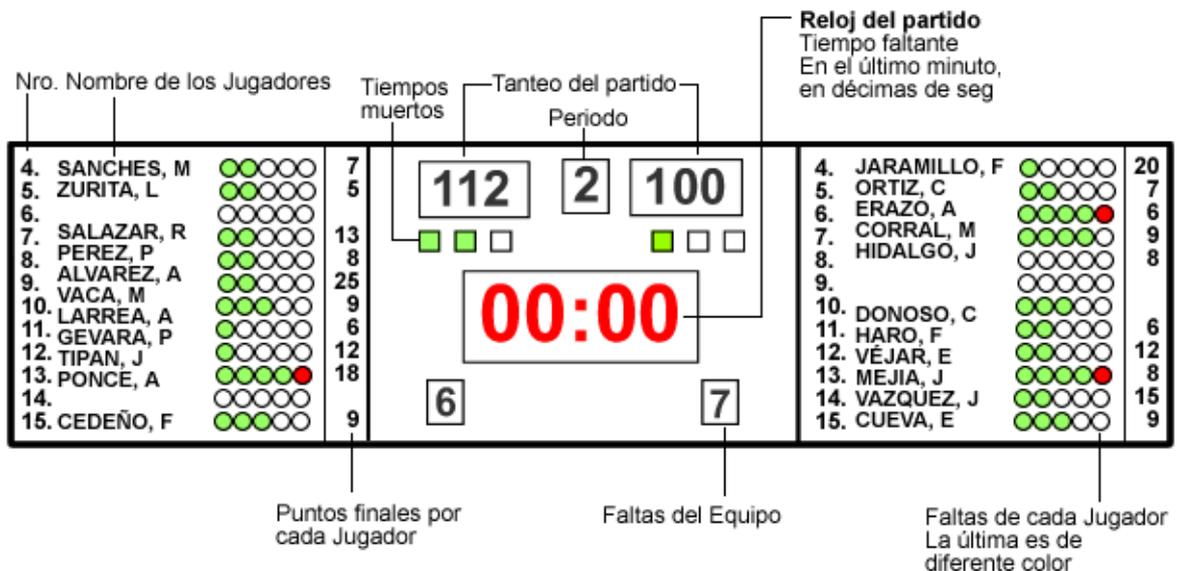


Figura 2.11: Marcador

2.1.6- Control del juego

Para controlar el juego, existen dos árbitros, uno principal y un auxiliar, y en la mesa de anotadores tres jueces con las funciones a continuación:

- Un juez denominado *juez cronómetro*, que se encarga de registrar el tiempo de juego y el tiempo interrumpido.
- Un *juez anotador*, quien registra en el acta del juego los nombres y números de todos los jugadores que participan, así como los puntos obtenidos y las faltas cometidas por equipo y jugador; y también registra los tiempos muertos por cada equipo. Adicionalmente le informa al árbitro el momento que un jugador ha acumulado cinco faltas.
- Un *juez operador*, que se encarga del dispositivo de 24 segundos para controlar el tiempo de posesión del balón.

El árbitro tiene varias funciones como:

- Analizar y aprobar la instalación.
- Iniciar el partido mediante el lance de la pelota en el centro de la cancha.
- Dictaminar la validez de un encesto.
- Eliminar a un equipo en el caso de que fuera necesario.
- Decidir la acción a tomar en algún aspecto que se produzca y que no este reglamentado.
- Revisar el marcador y aprobarlo al finalizar cada período.

Su lenguaje lo manifiesta a través de sus manos. En la actualidad existen 25 signos que permiten manifestar órdenes o intenciones. En la *Figura 2.12* se pueden observar a algunos de ellos:

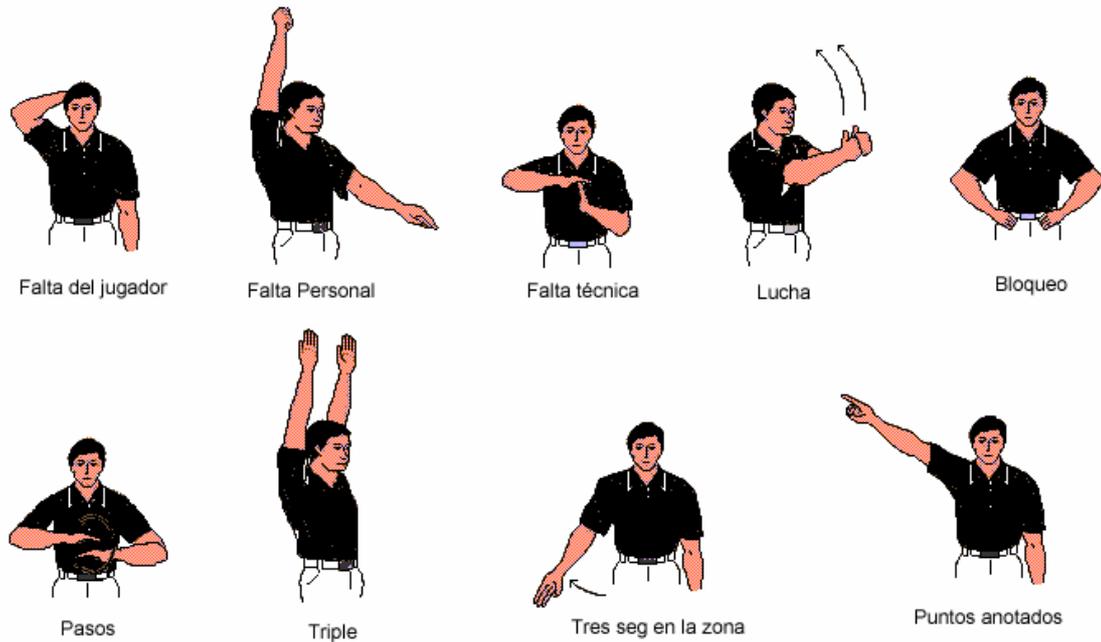


Figura 2.12: Lenguaje del árbitro

2.1.7- Disposiciones generales

Principios básicos.-

Cada árbitro, de forma independiente, tiene la potestad de marcar infracciones en cualquier momento del juego. Se puede marcar el número necesario de infracciones a uno u otro equipo. Sea cual fuere la falta, se registrará ésta en el acta, al trasgresor por cada infracción.

Faltas por jugador.-

Cuando un jugador ha acumulado cinco faltas ya sean técnicas o personales en un partido de 2 períodos o seis faltas en un partido de cuatro períodos, el juez anotador informará al árbitro, para que el jugador abandone inmediatamente el terreno de juego y sea sustituido por otro jugador.

Faltas por equipo.-

Se anota una falta de equipo cuando se ha acumulado siete faltas ya sean personales o técnicas de sus jugadores en un período de 20 minutos o cuatro

faltas en un período de 12 minutos; y todas las faltas personales de jugador que se realicen posteriormente serán penalizadas con dos tiros libres ejecutados por el jugador contra el que se haya cometido la falta.

Las faltas cometidas en los períodos extras se los considera como faltas del último período.

2.2- Evaluación del Jugador en los partidos de baloncesto

La evaluación del jugador en los partidos de baloncesto es un proceso que despierta mucha atención de los entrenadores e investigadores, ya que permite recoger datos que, por una parte, ayudaran a mejorar aspectos deficientes del jugador y reforzar sus fortalezas en el juego, y por otra ofrecer incentivos a los jugadores que sean más efectivos y tengan mejor rendimiento en su posición, como los llamados premios al jugador más valioso.

Algunos importantes estudiosos del baloncesto², consideran que el proceso de evaluación es fundamental para ayudar a los entrenadores a tomar decisiones relativas a la gestión del partido, la competencia y los entrenamientos.

De manera general, los resultados de las evaluaciones convergen para el análisis de las estadísticas representativas del rendimiento global, defensivo u ofensivo de los jugadores en los partidos.

El gran interés en la obtención de estas estadísticas, tanto a nivel deportivo como comercial, ha ocasionado un ambiente adecuado a su desarrollo y, de la misma forma, ha suscitado su divulgación en grandes cantidades. [Brown, 1991], describe que la aparición de nuevas estadísticas y formas de cálculo, además de expresar un progreso natural en la comprensión de este proceso, está también

² Bradshaw (1984), Swalgin (1994).

relacionado a la necesidad de que las estadísticas se ajusten a las diferentes filosofías de juego de los entrenadores.

Es de esta forma que a continuación se van a mostrar las propuestas más conocidas y usadas en el mundo para el cálculo de las estadísticas para la evaluación de los jugadores en los partidos de baloncesto y que van a ser empleadas en el presente trabajo.

La evaluación del jugador en los partidos va a estar centralizada al cálculo de la efectividad, tiros de campo y tiros libres, rebotes, faltas y control del balón, normalizadas a 100 posesiones de balón³, para de esta forma neutralizar la “contaminación” impuesta a los datos estadísticos por el ritmo del equipo en el partido.

2.2.1- Efectividad del Jugador

Uno de los coeficientes de evaluación del rendimiento (efectividad) de los jugadores más usado por los entrenadores es el *Tendex*⁴. Este coeficiente se constituye la base metodológica de la publicación anual *Basketball Abstract* en la cual se revisa y evalúa el rendimiento de todos los jugadores de la NBA⁵.

El Tendex se calcula por medio de la siguiente ecuación:

$$Tendex = PA + TR + As - TCF - TLF - PB$$

En donde:

- PA: Total de puntos anotados
- TR: Total de rebotes (defensivos + ofensivos)
- As: Total de Asistencias

³ La posesión del balón permite calcular el ritmo (lento, medio o rápido) del equipo en un partido.

⁴ Heeren (1998).

⁵ National Basketball Association: Liga Profesional de Baloncesto de Norte América

- TCF: Tiros de campo fallados (2pts. + 3pts.)
- TLF: Tiros libres fallados
- PB: Pérdidas de balón

2.2.2- Tiros de Campo

Este coeficiente de evaluación debe expresar la eficacia de un jugador en los tiros de campo (de corta y larga distancia) y también tener en cuenta el número de veces que consigue convertir. En primer lugar se calculan los puntos por tiro:

$$PT = \frac{PA - TLA}{TCI}$$

En donde:

- PT: Puntos por tiro
- PA: Puntos anotados
- TLA: Tiros libres anotados
- TCI: Tiros de campo intentados (convertidos y fallados)

Luego se dividen los tiros de campo convertidos por los puntos por tiro, así:

$$TC = \frac{TCA}{PT}$$

En donde:

- TC: Tiros de campo
- TCA: Tiros de campo anotados
- PT: Puntos por tiro

2.2.3- Rebotes

La evaluación de los rebotes defensivos y ofensivos se hace de igual forma partiendo de una idea de eficacia. En este caso se calcula el porcentaje de rebotes defensivos (y ofensivos) del jugador, con respecto al equipo en el cual participa.

$$RD = \left(\frac{RdJ}{RdE} \right) * 100$$

En donde:

- RD: Rebotes defensivos
- RdJ: Rebotes defensivos del jugador
- RdE: Rebotes defensivos del equipo

De igual manera con los rebotes ofensivos:

$$RO = \left(\frac{RoJ}{RoE} \right) * 100$$

En donde:

- RO: Rebotes ofensivos
- RoJ: Rebotes ofensivos del jugador
- RoE: Rebotes ofensivos del equipo

2.2.4- Tiros Libres

El coeficiente de evaluación de los tiros libres consiste en una estadística de eficacia, que está basada en el porcentaje de tiros libres y el número de tiros libres convertidos. Primero se calcula el porcentaje de tiros libres:

$$\%TL = \frac{TLA}{TTL}$$

En donde:

- %TL: Porcentaje de tiros libres del jugador
- TLA: Tiros libres anotados
- TTL: Total de tiros libres del jugador

Luego se calcula el coeficiente de eficacia de tiros libres:

$$ETL = \%TL * TLA$$

En donde:

- ETL: Eficacia de tiros libres
- %TL: Porcentaje de tiros libres
- TLA: Tiros libres anotados

2.2.5- Faltas

Este dato simplemente se obtiene calculando la relación entre las faltas recibidas y las faltas cometidas:

$$F = \frac{FR}{FC}$$

En donde:

- F: Faltas
- FR: Faltas recibidas
- FC: Faltas cometidas

2.2.6- Control del balón

El control de balón es la relación entre las asistencias y las pérdidas de balón.

$$CB = \frac{As}{PB}$$

En donde:

- CB: Control del balón
- As: Asistencias
- PB: Pérdidas de balón

Con el cálculo del control del balón se termina la primera parte del tratamiento de los datos. Seguidamente se necesita calcular el ritmo del partido (posesiones de balón). Luego, la interpretación de los datos obtenidos deberá ser normalizada en función del ritmo del equipo en el partido, ya que existen marcadas diferencias entre los datos estadísticos de un jugador cuando el juego es más rápido (por ejemplo 85 posesiones de balón) o más lento (por ejemplo 55 posesiones de balón). Adicionalmente este proceso es fundamental para que se pueda comparar y medir el rendimiento de los jugadores en partidos distintos.

2.2.7- Posesiones de balón

Este coeficiente que mide el ritmo de un equipo en el partido, es de amplio uso y fue presentado por Turcoliver (1990), que realizó análisis de ecuaciones anteriormente usadas, ampliando su campo de aplicación.

$$PoB = TCI - RO + PB + 0.4 * TLI$$

En donde:

- PoB: Posesiones de balón del equipo
- TCI: Tiros de campo intentados por el equipo.
- RO: Rebotes ofensivos del equipo
- PB: Pérdidas de balón del equipo
- TLI: Tiros libres intentados por el equipo

2.2.8- Normalización de datos estadísticos

Una vez que se ha calculado las posesiones de balón, se realiza el último paso de la evaluación, que consiste en la normalización de los datos estadísticos antes obtenidos a 100 posesiones de balón. Por ejemplo los Tiros de campo obtenidos para un jugador se los multiplica por 100 y se divide para las posesiones de balón del equipo.

$$TCN = \frac{(TC * 100)}{PoB}$$

En donde:

- TCN: Tiros de campo del jugador normalizados
- TC: Dato estadístico Tiros de Campo del jugador
- PoB: Posesiones de balón del equipo

De esta forma se elimina la “contaminación” que el ritmo del partido impone a las estadísticas y hace más factible comparar el rendimiento de los jugadores a lo largo de un campeonato con mayor validez.

Este procedimiento se repite de igual forma con todas las estadísticas (tiros libres, faltas y control del balón), con excepción de rebotes defensivos y ofensivos que son porcentajes.

El básquetbol es un juego de 5x5 jugadores, no son 5 partidos de 1x1. Los equipos están formados por jugadores con funciones diferentes, pero que no se pueden omitir o disociar. Es decir que se debe entender el rendimiento de un jugador como una contribución para todo el equipo y así este rendimiento debe ser necesariamente asociado al equipo.

Actualmente la toma de decisiones por parte de cuerpo técnico y entrenadores es cada vez más difícil de realizar. La presión de directivos, jugadores y público es grande y puede afectar el juicio de los entrenadores al tomar decisiones que en la mayor parte de los casos suelen ser subjetivas.

El uso sistemático, regular y consciente de un sistema de evaluación de jugadores y equipos como el que se ha analizado, no necesariamente va a ser una fórmula preestablecida para resolver los problemas de los equipos, pero es un paso decisivo para aportar mayor objetividad a las decisiones que se toman dentro y fuera del campo de juego.

2.3- La Facultad de Educación Física, Deportes y Recreación de la ESPE

2.3.1- Introducción

La Facultad de Educación Física, Deportes y Recreación (FEFDER) inicia sus actividades académicas como Instituto de Educación Física de la ESPE, el 5 de noviembre de 1989, cumpliendo de esta manera la demanda de formación de profesionales de esta rama en el país.

Teniendo ya más de tres lustros de existencia, la FEFDER ha dado un giro académico y administrativo, que le ha permitido mejorar su infraestructura y su oferta educativa.

La formación académica que brinda la FEFDER, está fundamentada en principios científicos y técnicos que propenden el desarrollo físico de las personas, mediante la práctica deportiva y el ejercicio; y adicionalmente con la

utilización de programas de recreación, promueve el desarrollo psicológico y anímico de los estudiantes.

Además la FEFDER coordina las selecciones de los diferentes deportes en los que la ESPE participa a nivel interinstitucional, provincial, nacional e internacional, entre los que se destaca el básquetbol, por los múltiples triunfos que esta disciplina ha dado a la Escuela.

El actual Coordinador de la Carrera de Educación Física, Deportes y Recreación de la ESPE es el Capitán Carlos Espinosa.

2.3.2- El básquetbol en la ESPE.-

El básquetbol es el deporte que mayor eco ha tenido en la Escuela Politécnica del Ejército. Tanto las selecciones masculinas como femeninas han alcanzado lugares de prestigio en el ámbito deportivo no solo de la provincia, sino del país e incluso en el plano internacional.

El básquetbol masculino de la ESPE ostenta en la actualidad el campeonato provincial y nacional inter-clubes, es penta-campeón del torneo preparación, tetra-campeón del torneo oficial y tetra-campeón nacional. El básquetbol femenino de la Escuela es el actual vice-campeón nacional inter-clubes y el vice-campeón nacional universitario.

CAPITULO III

MARCO TEÓRICO: METODOLOGÍA

3.1- Arquitectura dirigida por Modelos MDA

3.1.1- Introducción

MDA son las siglas de *Model Driven Architecture*, cuyo significado en español es *Arquitectura dirigida por modelos*. Fue creada en el año 2001 por OMG (*Object Management Group*)⁶, el cual es un consorcio internacional de empresas informáticas sin ánimo de lucro, creado en el año 1990, cuyo objetivo es potenciar el desarrollo de aplicaciones orientadas a objetos distribuidas. Ha definido numerosas especificaciones y estándares, entre los que se encuentran: CORBA, UML, XML, MOF Y CWM.

MDA es un marco de trabajo cuyo enfoque de desarrollo de aplicaciones está centrado en la arquitectura. Nos permite modelar software distribuido mediante el uso de diferentes modelos que sirven de guía en todo el proceso de desarrollo, cuyo nuevo paradigma se lo conoce como *Ingeniería de modelos* o *Desarrollo basado en modelos*, de los cuales se obtienen beneficios importantes como *productividad, portabilidad, interoperabilidad* y el *mantenimiento*.

MDA acomete la separación del sistema en tres modelos diferentes pero que se encuentran relacionados entre sí. Su proceso de desarrollo comienza con los requisitos del negocio que generan *Modelos Independientes de Computación* o *CIM*, de los cuales se obtienen *Modelos Independientes de la Plataforma* o *PIM* y

⁶ Cuyo sitio Web es: <http://www.omg.org/>

éstos a su vez se transforman en varios *Modelos Específicos de la Plataforma* o *PSM* donde finalmente se obtendrá el respectivo código, como se puede observar en la *Figura 3.1*:



Figura 3.1: Proceso de desarrollo con MDA

Como pudimos observar, MDA separa la especificación de las operaciones y datos de un sistema y los detalles de la plataforma en la que el sistema será construido, proporcionando una flexibilidad duradera tanto en implementación, integración, mantenimiento y pruebas, como se muestra en la *Figura 3.2*:

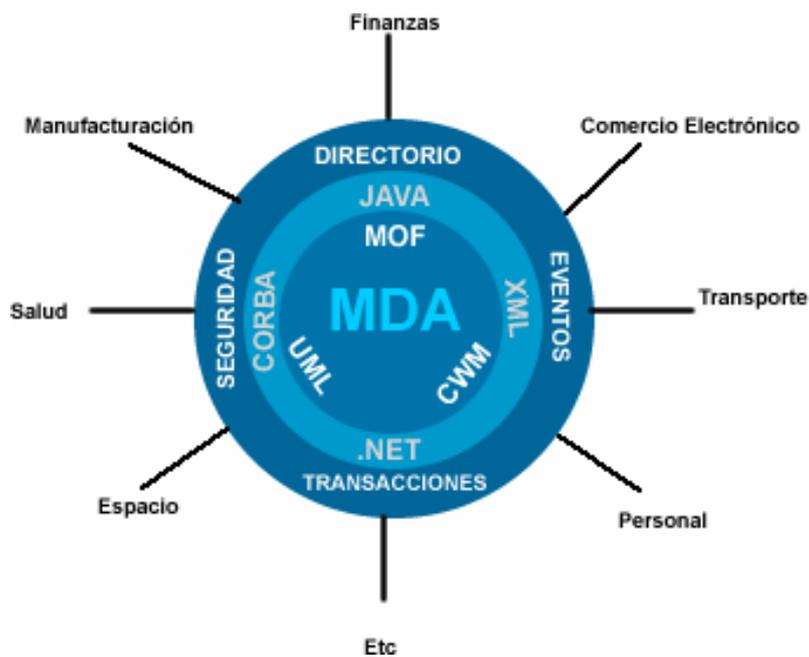


Figura 3.2: Logotipo de MDA

Según Rodríguez Jesús, 2004: “*La ingeniería de modelos es considerada como un nuevo paso en el camino hacia lenguajes de programación más expresivos y hacia una mayor automatización.*”⁷

⁷ Facultad de Informática de la Universidad de Murcia, Rodríguez Jesús (2004). Ingeniería de Modelos con MDA

Las ventajas de MDA son:

- EL sistema tiene un menor impacto si se presentan cambios en la plataforma.
- Existe trazabilidad⁸ de los requisitos del sistema
- Su arquitectura esta preparada para trabajar con diferentes tecnologías, sean nuevas o antiguas.
- No mezcla los detalles de implementación con las funciones de negocio.

A continuación se va a desarrollar una breve descripción sobre cada uno de los modelos, pero para ello es necesario conocer el concepto de modelo.

Un modelo (en general) nos permite producir algo que existe en la realidad. Desde el punto de vista informático, un modelo es una descripción de una parte o de todo un sistema, escrito en un lenguaje *bien definido*, es decir, en el cual se tienen asociados una sintaxis y una semántica bien definidas.

En MDA a diferencia de los métodos de desarrollo de software tradicionales, los modelos no son utilizados únicamente para la documentación e implementación, sino que constituyen el armazón principal en el proceso de desarrollo de MDA, ya que los modelos a través de diferentes transformaciones llegan a convertirse en el producto final.

El lenguaje UML (Unified Modeling Language) es un lenguaje de modelado bien definido que se ha convertido en el lenguaje de modelado *oficial* de MDA.

⁸ *Trazabilidad*: significa que a partir del elemento del modelo destino del cual se ha generado, se puede conocer el elemento origen.

Cabe recalcar que en MDA se puede utilizar cualquier lenguaje bien definido formalmente y que sea independiente de la plataforma. El presente proyecto construirá sus modelos utilizando UML.

UML define una serie de *submodelos* que permiten modelar negocios, aplicaciones de software y arquitecturas de sistemas, es decir permiten expresar diferentes características de un proyecto en cada una de sus fases de desarrollo. Es un lenguaje gráfico flexible y customizable. Permite crear diferentes tipos de modelos, en los que se encuentran procesos de negocios, secuencias de consultas, aplicaciones, bases de datos, arquitecturas, Web, entre otros.

3.1.2- Modelo Independiente de Computación (CIM)

Es una vista de un sistema de cómputo desde un punto de vista independiente, es decir el modelo describe los procesos de negocio que el sistema debe resolver sin tener en cuenta que será ejecutado por un computador. Un CIM no muestra detalles de la estructura de sistemas.

3.1.3- Modelo Independiente de la Plataforma (PIM)

Es un modelo del sistema con un nivel de abstracción alto, en el que se materializa el sistema ya que permite representar la *estructura, funcionalidad y restricciones* del mismo sin tomar en cuenta una plataforma establecida, es decir es la descripción de la lógica del negocio. Está basado en UML y es la base en el proceso de desarrollo de MDA.

Existen dos aspectos importantes que se deben tomar en cuenta:

- Es fácilmente comprensible por los usuarios del sistema, lo que permite que la validación del sistema sea sencilla.
- Permite diferentes implementaciones del sistema en diversas plataformas, sin afectar su estructura y funcionalidad básicas.

3.1.4- Modelo Específico de la Plataforma (PSM)

Es un modelo del sistema con un nivel de abstracción más bajo, mucho más cercano a la vista de código que el PIM. Un PSM es un PIM que puede incluir más o menos detalles específicos para ser implementado en una plataforma, dependiendo de su propósito. El PSM describe la arquitectura final del sistema.

De un PIM, se pueden obtener varios PSMs, donde cada PSM representa una parte del sistema, como por ejemplo: un esquema de base de datos, un componente, un servicio WEB, etc.

Para poder construir los PSMs es necesario conocer el concepto de los *Perfiles UML*, que son extensiones de UML que consisten en agregar información semántica, es decir, agregar detalles específicos de la plataforma a los modelos, como por ejemplo: estereotipos, métodos, etc., sobre los perfiles se hablará mas adelante en el apartado *Otros conceptos en MDA*.

Un PSM puede ser depurado sistemáticamente a niveles más bajos, hasta que pueda ser transformado a código directamente.

3.1.5- Otros conceptos en MDA

Nuevo proceso de desarrollo de sistemas.-

A diferencia del proceso tradicional, el nuevo proceso cambia las fases de *análisis, diseño y codificación*, manteniendo las fases de *requisitos, pruebas e implementación*.

- En la fase de análisis, se desarrolla el CIM y el PIM tomando en cuenta las necesidades y funcionalidad del negocio.
- En la fase de diseño, se transforma el PIM a uno o más PSMs, en esta fase, los desarrolladores deben tener conocimiento sobre la plataforma y la arquitectura que requiere el sistema.
- En la fase de Codificación, al tener como base el PSM, se genera el código del sistema ya sea manualmente o utilizando una herramienta de transformación.

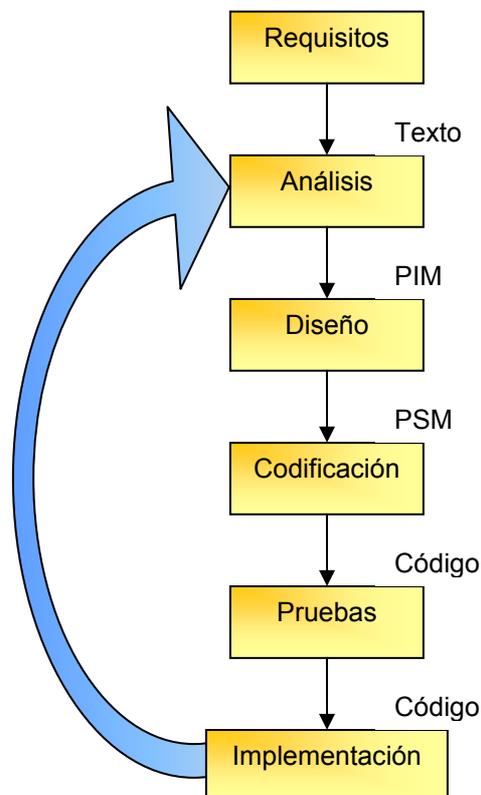


Figura 3.3: Nuevo proceso de desarrollo de sistemas con MDA

Plataforma.-

MDA propone el desarrollo de un Modelo Independiente de la Plataforma y Modelos Específicos de la Plataforma, por lo tanto es necesario conceptualizar una definición clara sobre el concepto de *plataforma*.

Según la Guía de MDA⁹, una plataforma en general es una agrupación de subsistemas y tecnologías que proveen un conjunto coherente de funcionalidad, a través de interfaces y patrones de uso específicos, en donde cualquier subsistema que dependa de la plataforma puede usarse sin preocuparse por los detalles de la funcionalidad provista por la plataforma que se va a implementar.

Existen tres interpretaciones posibles acerca de plataforma, por lo cual el desarrollador deberá especificar claramente cual es su interpretación, ya que eso determinará que con un PSM se pueda generar automáticamente el sistema o no. Estas interpretaciones son:

- *Plataforma Genérica.-* en la que se aprecian tres conceptos:
 - *Objeto:* es una plataforma que soporta el estilo familiar de la arquitectura de objetos con interfaces, respuestas de servicios, mensajes, etc.
 - *Lote:* es una plataforma que soporta una serie de programas independientes donde cada uno se ejecuta completamente antes de que el siguiente programa comience.
 - *Dataflow:* es una plataforma que soporta un flujo de datos continuo entre partes de software.

⁹ Object Management Group. MDA Guide Version 1.0.1

- *Plataforma con Tecnología Abstracta.*- en la que considera la programación orientada a objetos en términos abstractos como por ejemplo la especificación del estándar CORBA. Es decir, se realizan descripciones abstractas, que necesitan detalles específicos de una plataforma determinada para que se pueda efectuar la transformación y obtener el software final.
- *Plataforma con Tecnología Concreta.*- esta interpretación relaciona la plataforma con tecnologías concretas como Enterprise JavaBeans, Microsoft .NET, J2EE, etc.

Finalmente se señala que el concepto de plataforma se relaciona con la tecnología de implementación.

Metamodelado.-

Es un mecanismo para definir formalmente lenguajes de modelado, donde un metamodelo de un lenguaje provee una definición precisa de sus elementos. Este concepto es importante en MDA ya que actúa como un mecanismo para *especificar lenguajes de modelado* que permita evitar la ambigüedad y permitir que una herramienta pueda leer, escribir y comprender modelos.

Cabe recalcar que cuando se ve la clase como un objeto, la clase es una instancia de otra clase.

Debido a la idea de ver las clases como objetos, se puede crear una jerarquía de instanciación de clases y objetos, por lo que OMG ha definido cuatro capas o niveles de modelado como se puede observar en la *Figura 3.4*, los cuales se describen brevemente a continuación:

- *Capa M0.-* en esta capa se especifican todas las *Instancias Reales* del sistema, es decir las entidades físicas existentes en el sistema, como por ejemplo el *nombre, teléfono y dirección de los clientes*, cada una de estas entidades son instancias que pertenecen a la capa M0.
- *Capa M1.-* esta capa representa el modelo de un sistema de software, cada elemento de M0 es una instancia de cada elemento de M1, como por ejemplo una *Clase Cliente* con los atributos nombre, teléfono y dirección.
- *Capa M2.-* se representan entidades de un lenguaje modelado, donde los elementos de M1 son instancias de M2, como por ejemplo una entidad *UML Class* que pertenece al metamodelo de UML.
- *Capa M3.-* se representan entidades para definir lenguajes de modelado, donde de forma similar que las capas anteriores, los elementos de M2 son instancias de M3, como por ejemplo *MOF Class* de MOF.

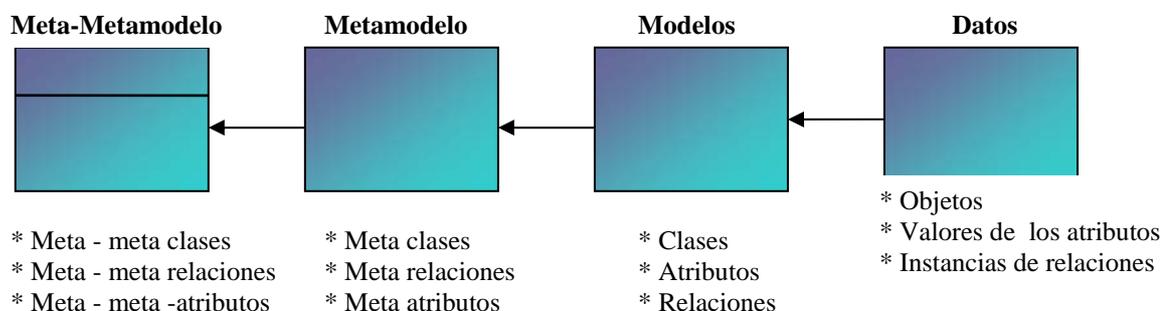


Figura 3.4: Capas del metamodelado

MOF o Meta Object Facility.-

Es un estándar propuesto por OMG que define un lenguaje *común y abstracto* para lenguajes de modelado para poder leer e intercambiar otros modelos que se encuentren expresados en este lenguaje. Este lenguaje también facilita la construcción de herramientas para definir lenguajes de modelado. Por lo tanto, su

rol fundamental dentro de MDA es que se pueda definir transformaciones sobre los metamodelos de forma estándar.

Este lenguaje define cinco conceptos básicos que se detallan a continuación:

- *Clases.*- permiten definir tipos de elementos en un lenguaje modelado.
- *Generalización.*- también conocido como la herencia entre clases.
- *Atributos.*- definen propiedades como el tipo y la multiplicidad de los elementos del modelo.
- *Asociaciones.*- conocidas también como las relaciones entre las clases que constan de un nombre, navegabilidad y multiplicidad.
- *Operaciones.*- son las diferentes operaciones que se va a realizar dentro de la clase, las cuales deben definir sus respectivos parámetros.

Como no existen una capa superior a M3, MOF se define usándose a sí mismo.

Perfiles UML.-

Son un conjunto de estereotipos, restricciones y valores etiquetados que permiten obtener una variante de UML para un propósito específico, lo que permite que un modelo sea más completo ya que contiene detalles específicos de una tecnología o plataforma determinada.

- *Estereotipos.*- permiten crear nuevos tipos de elementos de modelado tomando en cuenta elementos existentes en el metamodelo de UML.
 - a. *Formato:* <<nombre _ estereotipo>>
 - b. *Ejemplo:* << EJBEntity>>

- *Restricciones.*- determinan las condiciones que deben seguir los elementos del modelo tomando en cuenta un dominio específico de la aplicación. Generalmente para definir una restricción se utiliza una expresión en OCL.
 - a. *Formato:* {restricción} conectado por una relación de dependencia al elemento.
 - b. *Ejemplo:* {ordenado}
- *Valor Etiquetado.*- permite agregar información en la especificación de un elemento, es decir es una extensión de las propiedades de un elemento UML.
 - a. *Formato:* {nombre_etiqueta = valor_etiqueta}
 - b. *Ejemplo:* {Periodo de rotación : días = 0}

Object Constraint Lenguaje (OCL) .-

Es un lenguaje que permite definir modelos más completos y exactos ya que se pueden escribir expresiones para consultas, condiciones, etc. Este lenguaje permite lo siguiente:

- Inicializar variables.
- Definir consultas.
- Establecer restricciones a las clases o atributos.
- Definir reglas para la transformación de atributos o asociaciones.

3.1.6- Transformaciones de modelos.

Según el diccionario, transformación es: hacer cambiar de forma a una persona o cosa, es decir cambiar algo de un estado en particular, a otro diferente. En MDA, una transformación es la concepción de un modelo objetivo desde un

modelo fuente, de acuerdo a ciertos parámetros que se llaman *definiciones de transformación*.

Definiciones de Transformación.-

Una definición de transformación (mapping MDA) describe como un modelo debe ser transformado, es decir, suministra la especificación de la transformación de un PIM a un PSM. Es una agrupación de reglas que detallan cómo un modelo fuente puede ser transformado a un modelo objetivo.

- *Regla de Transformación.-* Es una descripción de cómo una o varias partes de un modelo fuente (en lenguaje fuente), pueden ser transformadas en una o varias partes del modelo objetivo (en lenguaje final). Una regla de transformación debe definir, sin ambigüedades y de forma clara, la relación expresa que existen entre el modelo fuente y el objetivo, permitiendo ir de una parte a otra de forma indistinta.

Se puede transformar de PIM a PSM de varias formas, como se muestra a continuación:

- *Manualmente*, tomando como base el PIM
- *Semiautomática*, al generar un PSM prototipo que se lo completa manualmente.
- *Automáticamente*, al generar un PSM completo mediante la utilización de herramientas de transformación que toman como base el PIM y diferentes definiciones de transformaciones. Estas herramientas usan algoritmos de transformación que permiten pasar de un modelo a otro, verifican modelos e incluso generan código.

Tipos de definiciones de transformación.-

- *De tipos.-* permite transformar cada tipo de elemento del modelo PIM usando reglas preestablecidas en uno o más elementos PSM. Las reglas se las puede asociar a tipos de metamodelado como clase, atributo, relación o mediante la utilización de estereotipos.
- *De instancias.-* esta transformación toma en cuenta elementos específicos de una plataforma en particular, es decir que ciertos elementos del PIM sean transformados al PSM pero tomando en cuenta una plataforma específica. Por lo tanto es necesario el uso de *marcas*, que tienen un significado en el PSM, no son parte del PIM; por lo tanto el desarrollador deberá indicar cómo debe ser la transformación.

Es necesario que en una transformación de un PIM a un PSM se utilice una combinación de estos dos tipos de transformaciones para lograr que el PSM sea completo.

Características de las transformaciones.-

Existen varias características que deberían tener las transformaciones, las cuales se detallan a continuación:

- *Ajustar las transformaciones:* es importante que el desarrollador tenga cierto control en el proceso de las transformaciones, ya sea de forma manual donde el desarrollador define qué elementos van a ser transformados por ciertas reglas de transformación, o mediante la utilización de condiciones en las que el desarrollador aplica una condición previa a cada regla de transformación, o también el desarrollador puede definir parámetros en las transformaciones.

- *Trazabilidad:* es una característica muy útil en MDA, que permite conocer el elemento origen a partir del elemento destino generado. Esto permite que cuando se realice un cambio en el PSM, éste se refleje directamente en el PIM.
- *Consistencia incremental:* al momento de generar el modelo destino, siempre se requiere realizar cambios al mismo, como por ejemplo mejorar la interfaz de usuario; pero cuando se necesita realizar un cambio al modelo origen, es deseable que el trabajo adicional que se desarrolló se mantenga. Por lo tanto esta característica se refiere a que el proceso de transformación sepa qué elementos del modelo origen fueron cambiados, para reemplazarlos en el modelo destino manteniendo la información extra.
- *Bidireccionalidad:* esta característica se refiere a que las transformaciones se las pueda realizar en ambas direcciones. Para lo cual se lo puede lograr ya sea utilizando una misma definición de transformación o aplicando definiciones de transformación diferentes en cada caso. Esta propiedad es un poco difícil de alcanzar ya que una misma definición de transformación no es muchas veces necesariamente la inversa de otra. Y a veces es hasta imposible de alcanzar puesto que para generar el modelo destino se suele ignorar información que no es relevante para el modelo destino y por tal motivo no se podrá volver a generar el modelo origen.

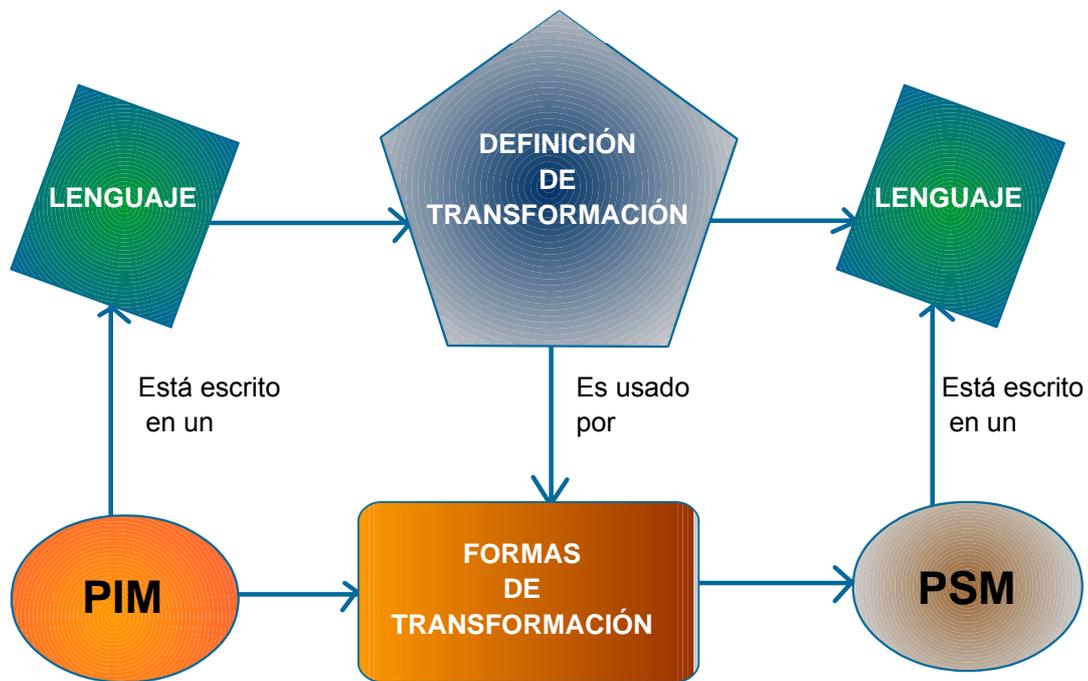


Figura 3.5: Diagrama resumen sobre transformaciones

3.2- Metodología para el desarrollo de Sistemas de Información Web: MIDAS

3.2.1- Introducción

En la actualidad la masificación del uso de Internet ha popularizado el desarrollo y utilización de Sistemas de Información Web (SIW) que hacen posible que se puedan realizar por la red infinidad de operaciones que van desde comprar un libro hasta adquirir una casa o realizar complejas transacciones bancarias.

Las herramientas para desarrollar este tipo de sistemas han ido evolucionando casi a la par de los SIW, pero no ha sucedido de igual manera con el desarrollo y fundamentalmente la aplicación de metodologías adecuadas para SIW. Lo que se ha venido haciendo es adaptar las metodologías tradicionales, al desarrollo orientado a la Web; pero sin tomar en cuenta que este tipo de sistemas tienen características particulares, por lo que los sistemas desarrollados de esta manera producen confusión al usuario final, que se termina perdiendo mientras navega en

el sistema, porque desconoce con certeza los pasos que debe seguir para realizar una determinada operación.

Entre las características que se notan como especiales de las aplicaciones Web, es decir que son específicas y no se pueden generalizar con otros tipos de sistemas, se encuentran:

- La arquitectura de hardware y software.
- El diseño de la interfaz de usuario.
- El diseño de los hipertextos.
- La existencia de datos heterogéneos en diferentes plataformas.
- Aspectos de seguridad, entre otras.

Estas cuestiones determinan la necesidad de Metodologías específicas para el desarrollo de SIW.

Partiendo del hecho que los SIW actualmente tienen como base tres aspectos: el contenido (datos), la dimensión estructural o navegacional (hipertexto) y la dimensión de comportamiento; se plantea en el presente trabajo el estudio y aplicación de una metodología que toma en cuenta estos factores claves, y adicionalmente plantea el modelado del sistema desde una perspectiva orientada a los servicios que requiere el usuario: la **Metodología Midas**.

Midas es una metodología ágil para el desarrollo de SIW, presentada por la Dra. Paloma Cáceres y la Dra. Esperanza Marcos del grupo de investigación Kybele de la Universidad Rey Juan Carlos de España. Esta metodología junta las bondades de la Arquitectura Dirigida por Modelos (MDA) propuesta por *Object Management Group* (OMG), y las mejores prácticas de las metodologías ágiles.

Midas se presenta como un marco de trabajo completo que tiene como base los parámetros que dicta la Ingeniería Web, integra importantes recomendaciones

de las metodologías ágiles (eXtreme Programming XP), los métodos de modelado de hipertexto y la estructura y características de MDA, dando lugar a una metodología que permite realizar un óptimo análisis y diseño de SIW, de acuerdo a los parámetros que requieren en la actualidad este tipo de aplicaciones.

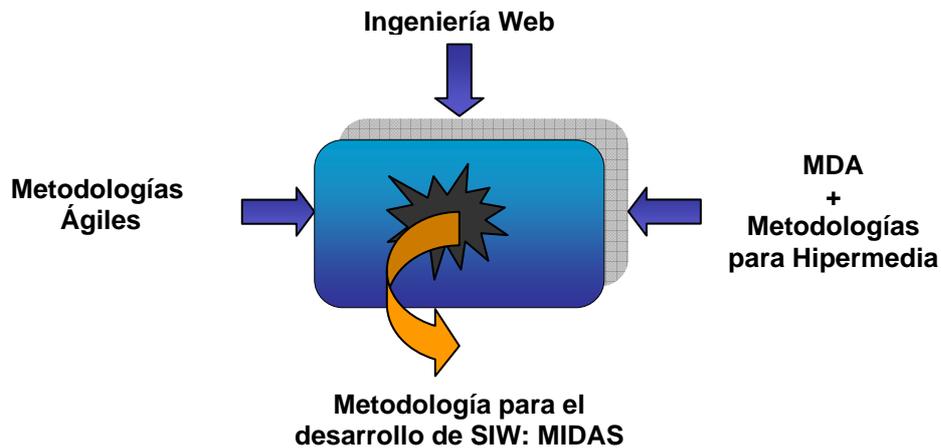


Figura 3.6: Origen de Midas

3.2.2- Desarrollo ágil de Sistemas de Información Web

Las metodologías tradicionales vs. Las metodologías ágiles.-

Las ya conocidas y ampliamente utilizadas metodologías para el desarrollo de sistemas “tradicionales”, juntan un sin número de parámetros que son fundamentales a la hora del desarrollo de software, ya que permiten tener organizado el desarrollo por el seguimiento al proceso que se da, lo que se traduce en el incremento de calidad del producto final. Pero ciertas prácticas de este tipo de metodologías pueden ser tomadas como desventajas frente al desarrollo de sistemas:

- Requieren de excesiva documentación.
- Son poco adaptables.
- No son flexibles al cambio en los requisitos del usuario.
- Están orientadas al proceso de desarrollo de software.

Debido a esto, hace algún tiempo atrás se plantearon las denominadas *Metodologías Ágiles* que tratan de poner solución a dichos factores, tratando de forma exitosa, de hacer menos engorroso el desarrollo de sistemas. A diferencia de las anteriores, este tipo de métodos tienen características como las siguientes:

- El código es la documentación.
- Son adaptativas.
- Tienen alta flexibilidad para el cambio en los requisitos del usuario.
- Están orientadas al personal (usuarios, arquitectos de software, ingenieros de software y desarrolladores en general).

Una de ellas, y muy utilizada es *eXtreme Programming (XP)*, que marca un nuevo paradigma en el desarrollo de sistemas, ya que prioriza la participación del usuario final durante todo el proceso de desarrollo de software, es decir combina las prioridades del cliente o negocio en sí con las condiciones técnicas (desarrollo).

Esta metodología sugiere que el cliente debe incorporarse al equipo de desarrollo, para que en el caso que surja cualquier duda el proyecto no se retrase.

XP recomienda pasar a producción de forma rápida pequeñas versiones del producto, esto implica ciclos de vida cortos.

Trata de evitar la complejidad, forzando la realización de diseños lo más sencillos posibles.

La Generación de pruebas continuas por parte de los desarrolladores (pruebas Alpha) es sumamente importante en XP para continuar el desarrollo.

Recomienda la programación del sistema en pares de personas, además de integrar y completar el sistema continuamente, varias veces al día, cada vez que se termina una tarea.

XP como parte fundamental de su metodología exige que todos los programadores generen una codificación estándar (términos normalizados), para que al integrarse el código, exista armonía en su contenido.

Estas prácticas han sido adaptadas e integradas al marco de trabajo de Midas, lo que le permite aprovechar las mejores prácticas de las metodologías ágiles en el desarrollo de SIW.

Desarrollo ágil en la Web.-

Es muy importante aclarar que existen ciertos parámetros para determinar cuando se debe utilizar procesos ágiles de desarrollo de software, y principalmente estos son:

- Los requisitos del usuario son desconocidos o variables.
- Se requiere una rápida generación de prototipos.
- Es necesaria la entrega de versiones previa a la entrega final.
- Existe una alta probabilidad de realizar cambios.
- Los ciclos de desarrollo no son demasiado largos.

Estos aspectos determinan la utilización o no de una metodología ágil en el desarrollo del software, es decir si la mayoría de estos parámetros se cumplen es recomendable aplicar procesos ágiles.

Ahora bien, en el caso específico de estudio, también se debe tomar en cuenta los factores que determinan el desarrollo de un SIW o un sistema de otro tipo,

para lo cual se listan a continuación las principales características de los desarrollos orientados a la Web:

- Se necesita una disponibilidad rápida del software en la red.
- Los ciclos de desarrollo de los sistemas son generalmente más cortos.
- En la mayoría de los casos los requisitos de usuario son insuficientes o desconocidos.
- Los sistemas y su desarrollo revisten una aparente complejidad media.

Si se realiza un análisis comparativo entre las características de los procesos ágiles con los desarrollos Web, como se muestra en la *Tabla 3.1*, se encontrarán coincidencias que justifican de forma sobrada la utilización de metodologías ágiles en el desarrollo de Sistemas de Información Web, y es justamente en este contexto en el cual se enmarca Midas.

Tabla 3.1: Comparación entre procesos ágiles y desarrollos Web

Procesos Ágiles	Desarrollos Web
Requisitos desconocidos	Requisitos desconocidos
Rápida generación de prototipos	Rápida disponibilidad del software en la red
Entrega de versiones previa a la final	
Alta probabilidad de cambios	Desarrollos complejidad media, aparente
Ciclos de desarrollo no muy largos	Ciclos cortos de desarrollo

3.2.3- El proceso metodológico de MIDAS

El proceso metodológico, surge de integrar al proceso ágil de desarrollo de SIW, con las ventajas de MDA (metodología dirigida por modelos). Es importante considerar que los procesos ágiles y los procesos dirigidos por modelos no contemplan de igual manera el *espacio del problema* (lo que se tiene que resolver) y el *espacio de la solución* (como resolverlo); como se puede observar

en la *Figura 3.7*, la parte sombreada corresponde al área en que se acercan ambas propuestas, y se toman en cuenta cinco aspectos fundamentales.

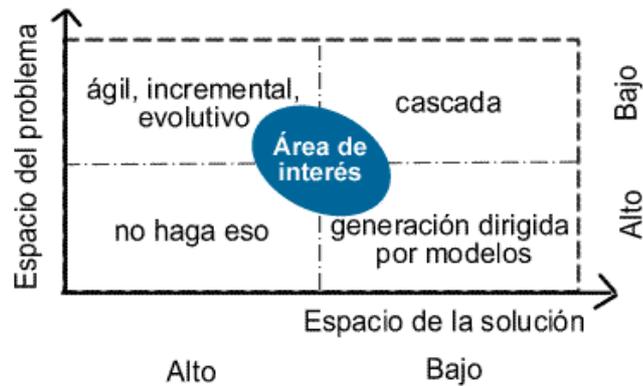


Figura 3.7: Relación existente entre el espacio del problema y el espacio de la solución

Los cinco aspectos que se toman en cuenta en el marco de trabajo de MIDAS son:

- **Personas.**- MIDAS le da una alta prioridad a la relación con el cliente, ya que esa relación es fundamental desde el principio: para establecer cuál es el espacio del problema, definirlo y utilizarlo posteriormente como base entre los desarrolladores (analista, arquitecto, diseñador), etc. y el cliente. El cliente es parte primordial en todo el proceso, ya que es la forma de avalar un producto de calidad y que cumpla sus necesidades.
- **Proceso.**- es relevante dentro de MIDAS y toma en cuenta a los procesos ágiles ya que es un modelo *iterativo, incremental, adaptativo y prototipado* aportando grandes ventajas al obtener versiones del producto software antes de la entrega final
- **Tecnología.**- MIDAS se basa en tecnología XML y objeto-relacional (OR).
- **Modelos.**- estos al igual que las personas tienen una alta prioridad en MIDAS puesto que son los artefactos que se generan. Los modelos junto

con el código son la única documentación que se realiza, pero esta se debe dejar constancia por escrito utilizando una notación concisa a través de modelos en UML o UML Extendido, ya que el conocimiento del sistema y la lógica del negocio, no pueden quedarse únicamente en la mente del cliente y los desarrolladores; esto satisface tanto al cliente cuanto a la buena realización del proyecto.

- **Software.-** cada iteración propuesta en MIDAS tiene como objetivo la obtención de un prototipo o versión del producto software, utilizando ciclos cortos de desarrollo (proceso incremental), así se podrán realizar entregas rápidas al cliente antes del producto terminado con la finalidad de garantizar el progreso del software.

Dado que el proceso de MIDAS es iterativo, incremental, adaptativo y prototipado; es necesario considerar la *Figura 3.8* que muestra las diferentes fases, características y actividades del desarrollo de SIW utilizando MIDAS.

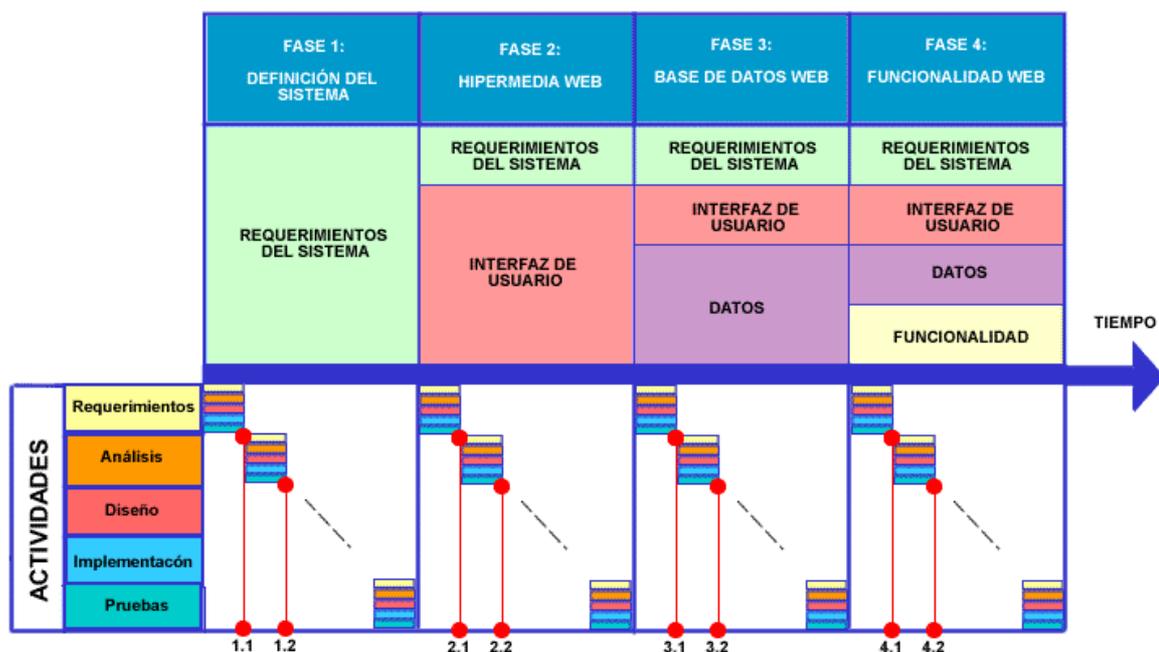


Figura 3.8: Ciclo de vida del desarrollo de SIW con MIDAS

Como se puede observar en el gráfico, se encuentran cuatro fases que permiten obtener un SIW con una funcionalidad compleja, esto quiere decir que contempla tanto criterios internos (estructura de la información y lógica de la aplicación) como externos (presentación y navegación).

Cada una de las fases va incorporando una nueva característica de manera incremental, y en cada característica se debe considerar las cinco actividades que darán como resultado una versión inicial que será presentada al cliente para su aprobación o modificación, en cuyo caso se volverá a realizar nuevamente todas la actividades, donde se obtendrá una nueva versión. Este proceso continuará hasta que el cliente quede satisfecho.

En este proceso se pueden destacar ciertas consideraciones que fueron analizadas y tomadas en base a las recomendaciones que propone eXtreme Programing, las cuales son:

- *Priorización de requisitos:* darle prioridad a las necesidades del cliente para determinar la fase de inicio del desarrollo.
- *Pequeñas versiones:* ya que es un proceso iterativo, cada iteración da como resultado en producto software que puede ser un documento, prototipo o programa.
- *Diseño sencillo:* utilizar las técnicas y actividades que sean necesarias en la realización de los diseños, sin mucha complejidad.
- *Pruebas:* como se pudo observar en el ciclo de desarrollo, se debe realizar pruebas en cada iteración para garantizar el desarrollo del sistema.

- *Producción por pares:* se recomienda que el desarrollo se lo realice en grupos de dos personas.
- *Cliente en el equipo:* el cliente debe prácticamente pertenecer al equipo de desarrollo, ya que la participación de éste garantiza calidad en el proceso de desarrollo y en el producto final.
- *Codificación estándar:* mantener un estándar en la codificación del sistema entre todos los desarrolladores.

MIDAS consta de una arquitectura de capas, donde cada una representa una *vista del sistema*, obteniendo una representación bidimensional del proceso metodológico, como se muestra en la *Figura 3.9*.

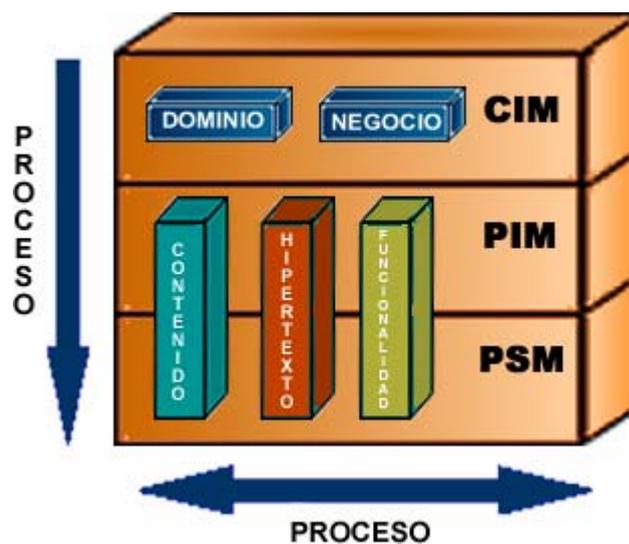


Figura 3.9: Dimensiones del proceso metodológico de MIDAS.

Como se puede observar en el gráfico, el eje X corresponde a los aspectos del SIW (contenido, hipertexto y funcionalidad); el eje Y corresponde al CIM, PIM y PSM.

La unión de ambas dimensiones da lugar a un conjunto de modelos que van desde el modelado del dominio y del negocio en el CIM, el modelado del

hipertexto, del contenido y de la funcionalidad del sistema (etapa de análisis y diseño) en el PIM y el modelado del hipertexto, del contenido y de la funcionalidad del sistema (etapa de diseño lógico hasta la implementación) en los PSM.

En resumen MIDAS propone especificar todo el sistema mediante los diferentes modelos mencionados anteriormente (CIM, PIM, PSMs) y especificar las reglas de transformación entre los modelos (PIM a PIM, PIM a PSM, PSM a PSM).

Debido a que MIDAS utiliza un enfoque orientado a servicios se toma como referencia las arquitecturas middleware (plataformas de desarrollo de servicios Web) como .NET, y se propone el desarrollo tomando en cuenta una *arquitectura de n-capas*. En este tipo de arquitectura se identifican comúnmente tres capas que son:

- Interfaz gráfica de usuario que corresponde con la vista de hipertexto
- Persistencia que corresponde con la vista de contenido
- Lógica del negocio que corresponde con la vista de comportamiento o funcionalidad

Dentro de este enfoque se encuentran los diferentes modelos que deben cumplir el proceso de desarrollo de MIDAS, mostrado en la *Figura 3.10* a continuación:

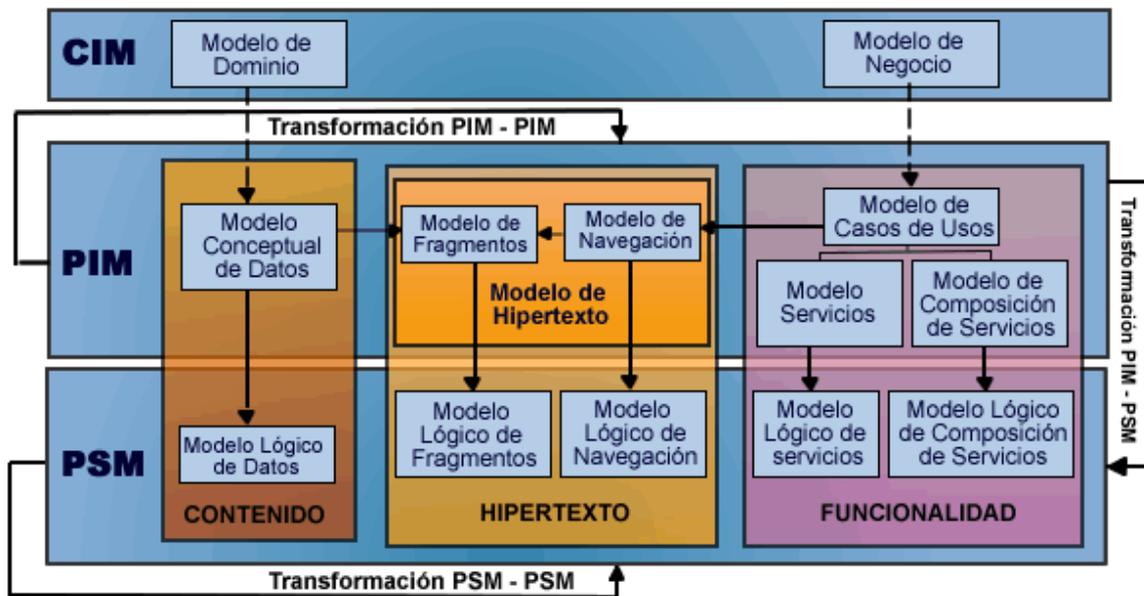


Figura 3.10: Proceso de desarrollo de MIDAS

Como se puede observar en el gráfico, MIDAS define diferentes modelos a nivel de CIMs, PIMs y PSMs para los aspectos de Contenido, Hipertexto y Funcionalidad, y también las reglas de transformación entre ellos.

El proceso comienza con la definición de los CIM, seguidamente los PIM y los PSM. Desde el punto de vista del eje X, al finalizar los CIM el proceso puede comenzar en diferentes puntos a nivel de PIM, ya sea en la vista de contenido o de funcionalidad y utilizando las *guías de transformación entre modelos*, se podría definir la vista de hipertexto, o sino seguir trabajando en los mismos aspectos para obtener los modelos a nivel de PSM; esto dependerá de las necesidades del cliente o de las especificaciones de requisitos realizada anteriormente.

El primer paso es entonces generar los modelos a nivel de CIM, para lo cual se procede a explicar en que consisten estos modelos.

Modelos Independientes de Computación (CIM).-

Existen dos modelos que pertenecen a los CIM, los modelos de dominio y de negocio del sistema. Los modelos de dominio se los representa a través de un diagrama de clases y los modelos de negocio a través del diagrama de casos de uso. Estos modelos son la base en el proceso de desarrollo de MIDAS y permiten modelar el contexto del sistema.

Modelo de dominio.-

Esta representado por el diagrama de clases, que es el más común usado en el modelado con UML, este diagrama representa las cosas estáticas que existen en el sistema, su estructura y sus relaciones. Son típicamente usados para realizar un diseño lógico y físico del sistema.

Una clase esta representada como un rectángulo que contiene dos líneas horizontales. En la parte superior de la primera línea se encuentra el *nombre* de la clase, que debe ser único, el cual es la descripción de la clase y es típicamente un sustantivo. Arriba de la segunda línea se localizan los *atributos* que describen las diferentes propiedades de la clase que se encuentran en todas las instancias de la clase. Estos se pueden representar de tres formas: solo su nombre, su nombre y su tipo, e incluso su valor por defecto. Finalmente se encuentran los *métodos u operaciones* ubicados en la parte posterior de la segunda línea, son las implementaciones de los servicios de la clase, que permiten mostrar un comportamiento común a todos los objetos. Las operaciones definen la lógica que será ejecutada por el sistema. En la *Figura 3.11* se encuentra un gráfico que permite mostrar un ejemplo de una clase.

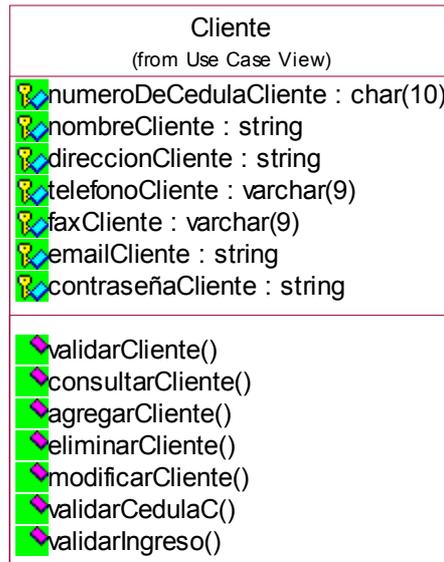


Figura 3.11: Clase de cliente

Modelo de negocio.-

Se lo representa a través del diagrama de casos de uso el cual muestra la funcionalidad completa que ofrece un sistema, una parte de él o una sola clase que un sistema implementará, mostrando la relación entre los actores (agentes externos) y los casos de uso (acciones) dentro del sistema. Definen conjuntos de comportamientos afines que el sistema debe desempeñar, con el fin de satisfacer los requerimientos. Es decir, los diagramas de casos de uso definen cómo trabajará un sistema y con quien interactuará.

Un caso de uso, es una unidad coherente de funcionalidad que describe una secuencia de acciones que se llevan a cabo entre un actor y el sistema, para realizar una tarea determinada. Se lo representa utilizando una elipse, en cuyo interior se encuentra un nombre, el cual debe reflejar una tarea específica (funcionalidad). En la *Figura 3.12* se encuentra un ejemplo sencillo de un diagrama de uso.

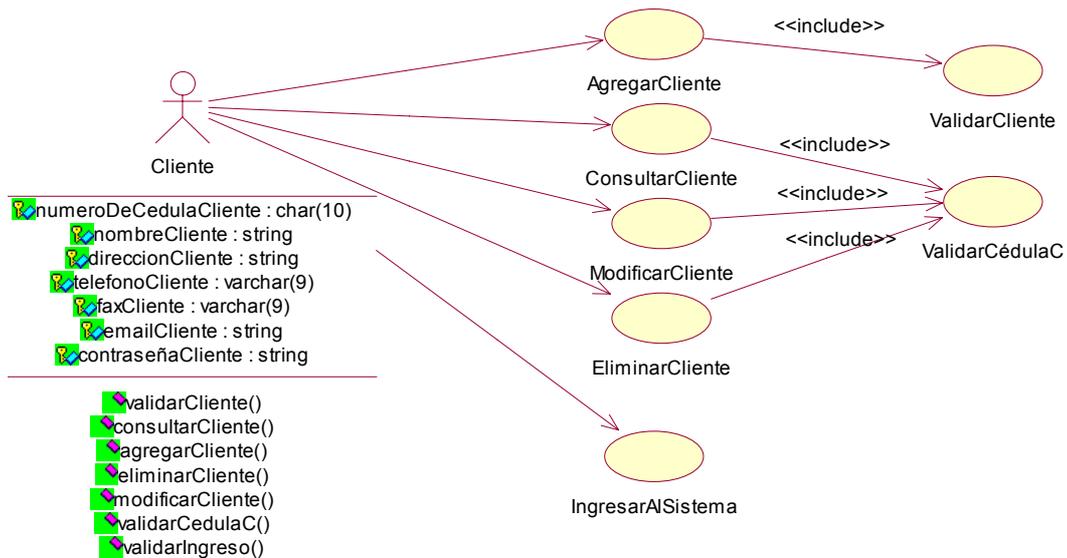


Figura 3.12: Diagrama de casos de uso sencillo

Modelos Independientes de la Plataforma (PIM).-

Un PIM define la especificación formal del sistema independientemente de la tecnología y aspectos técnicos específicos para la implementación.

Como se ha mencionado anteriormente, existen tres aspectos en los cuales se debe modelar el sistema, los cuales son:

Contenido: en este aspecto se plantea el modelo conceptual de datos, el cual se genera utilizando únicamente los atributos del modelo de dominio del CIM y se lo representa de igual manera con un diagrama de clases.

Hipertexto y Funcionalidad: para modelar estos aspectos se requiere de un proceso en el cual, los servicios de usuario conceptuales, serán la guía para obtener el *modelo navegacional* de un Sistema de Información Web y su enfoque debe estar orientado a los servicios que el usuario va a demandar del sistema. Este proceso se lo puede visualizar en la siguiente *Figura 3.13:*

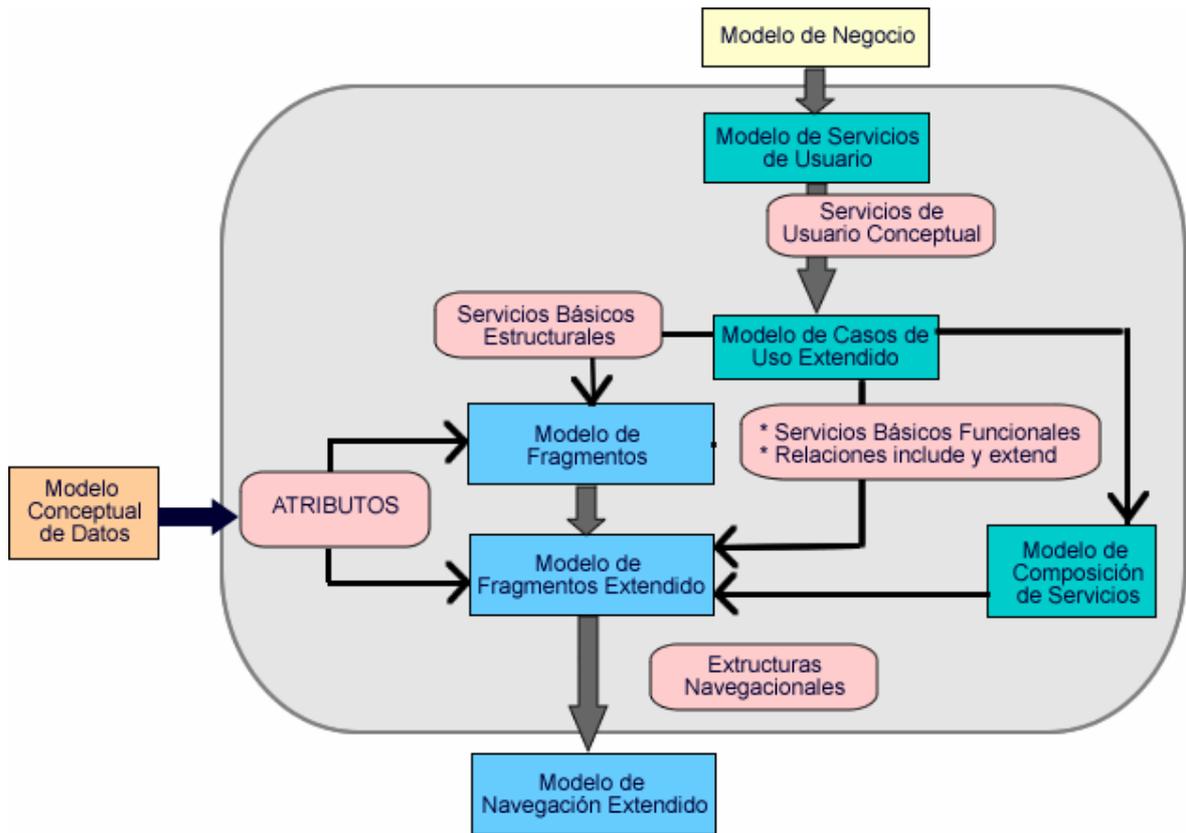


Figura 3.13: Proceso del Modelado de Hipertexto y Funcionalidad

Como se muestra en el gráfico, El proceso toma como entradas el Modelo de Negocio y el Modelo Conceptual de Datos y tiene como salida el Modelo de Navegación Extendido. Dentro del proceso se generan un conjunto de modelos representados mediante extensiones de UML y también se definen las reglas de transformación que existen entre los modelos. Estos modelos son el Modelo de servicios de usuario, el Modelo de Casos de Uso Extendido, el Modelo de Composición de servicios, el Modelo de Fragmentos, el Modelo de Fragmentos Extendido y finalmente el Modelo de Navegación Extendido.

Antes de empezar a describir el proceso, es necesario aclarar algunos conceptos que forman parte de una taxonomía de casos de uso. Los siguientes conceptos se los puede visualizar en la *Figura 3.14*.

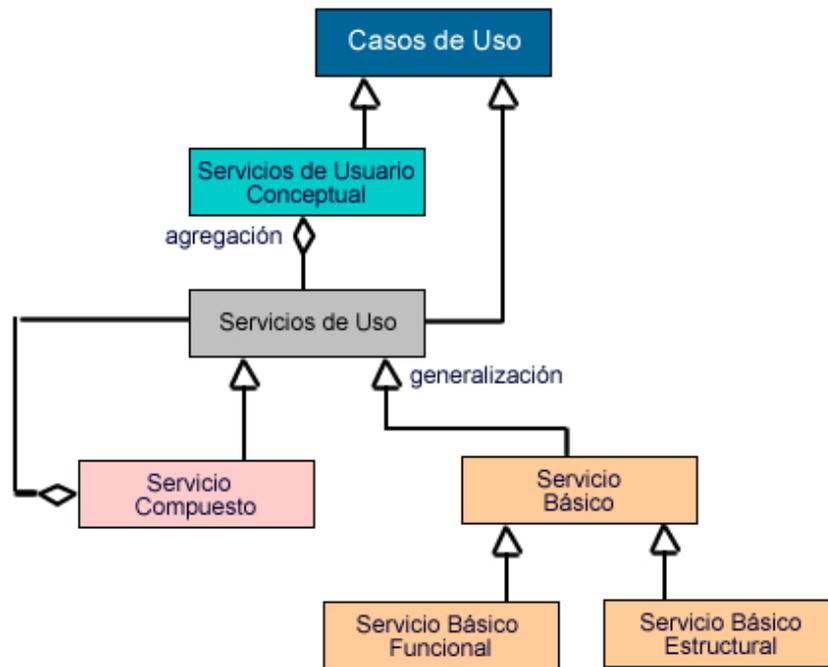


Figura 3.14: Taxonomía de casos de uso

- *Servicios de usuario conceptuales*: son un tipo especial de casos de uso que definen una funcionalidad específica que un usuario requiere del sistema. Esta definición conceptual solo existirá a nivel de PIM y puede ser implementada o no, mediante la utilización de servicios Web. Estos servicios se representan en el *modelo de servicios de usuario* y constituyen una agregación de servicios de uso.
- Los servicios de usuario conceptuales se descomponen en *Servicios de uso*, los cuales, son servicios más sencillos que constituyen funcionalidades requeridas por el SIW necesarios para completar el servicio de usuario conceptual. Estos servicios son los casos de uso que se encuentran en el *modelo de casos de uso extendido*. Estos servicios pueden ser:
 - *Servicio de usuario compuesto*.- es una agregación de servicios de uso básicos o compuestos. Estos servicios implementan varias

tareas para su ejecución, como Ingresar los datos de un cliente, lo que implica que se validen los datos de entrada, la cedula, etc.

- *Servicio de usuario básico.*- es una funcionalidad atómica específica del SIW desde el punto de vista del usuario. Estos servicios son básicos e incluyen una sola función, como Consultar los datos de un cliente. Este a su vez, desde el punto de vista de presentación, puede ser:

- *Estructural.*- cuando provee una vista de datos. Por ejemplo: consultar la información de un cliente.
- *Funcional.*- cuando existe interacción con el usuario, Por ejemplo en el caso que se requiera que el mismo ingrese parámetros ya sea para consultas o ingreso de datos, etc.

Modelo de Servicios de Usuario.-

Representa los servicios de usuario conceptuales que necesita del sistema. Es necesario identificar cuándo un servicio es *requerido* o *necesario* para realizar alguna actividad. Por lo tanto la navegación a través del sistema, debe ser capaz de guiarle al usuario de tal manera que se ejecuten los servicios que son necesarios realizarlos antes de un servicio de usuario conceptual.

La generación de este modelo empieza con identificar los servicios de usuario conceptuales, que son casos de uso estereotipados con <<CUS>>.

Modelo de Casos de Uso Extendido.-

Representa las diferentes funcionalidades y tareas en que se descomponen los servicios de usuario conceptuales. La generación de este modelo empieza con

la descomposición de los CUS en las diferentes tareas que se necesitan para efectuar dicho servicio. Luego de forma iterativa se deben clasificar los servicios descompuestos ya sea como básicos o compuestos y los servicios de uso básicos, como estructurales o funcionales; este proceso se lo debe realizar mientras haya servicios de uso compuestos. Finalmente se deben añadir los actores y las relaciones *include* y *extend* entre los servicios de uso.

- Un *actor*, personifica a un usuario del sistema que requiere o usa los casos de uso. Se lo representa mediante un muñeco y en la parte posterior se le asigna un nombre significativo.
- Una relación *include*, especifica la localización en un flujo de eventos, en el que existe una relación entre dos casos de uso; en el que un caso de uso base incluye el comportamiento de otro caso de uso. Su estereotipo es: <<include>>.
- Una relación *extend*, es también una relación entre dos casos de uso en el que se especifica que el comportamiento de un caso de uso base puede extender el comportamiento de otro caso de uso, o especializarse de otro. Su estereotipo es: <<extend>>.

En resumen un modelo de casos de uso contiene: actores, servicios de usuario básicos (estructurales o funcionales) y las relaciones *include* y *extend*.

Estos servicios de uso se estereotipan de la siguiente manera: los servicios de uso compuestos con <<CS>>, los servicios de uso básico funcional con <<FBS>> y los servicios de uso básico estructural con <<SBS>>.

Modelo de Composición de Servicios.-

Permite modelar la secuencia ordenada de servicios de uso que se identificaron en el modelo de casos de uso extendido, es decir cómo se realizará la composición entre cada uno de los servicios ya sean básicos funcionales o estructurales y además establecer la ruta que servirá de guía al usuario mientras navega en el SIW. Para lo cual se propone la utilización de Diagramas de Actividad y Estados propuestos por UML.

Un *diagrama de actividad*, es un caso especial de un diagrama de estados, en el cual casi todos los estados identifican que acción se ejecuta al estar en él y casi todas las transiciones son enviadas al terminar dicha acción. Generalmente modelan los pasos que se deben realizar en una actividad dada. Un diagrama de actividad está compuesto de un círculo sólido el cual indica el comienzo de la actividad, luego vienen las actividades que se encuentran dentro de un ovalo y finaliza con un círculo sólido rodeado de otro círculo.

Un *diagrama de estados*, se lo realiza cuando un objeto tiene un comportamiento significativo y muestra la secuencia de los estados de un objeto durante su ciclo de vida en respuesta a un estímulo recibido. La transición entre estados es instantánea y se debe a la ocurrencia de un evento. Los estados de los objetos están dados por el valor de sus atributos (estados) lo cual cambia sus comportamientos (métodos). En este diagrama se indican los eventos que hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera. Se representa mediante grafo dirigido, cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos. La

representación para el comienzo y fin del diagrama se lo realiza de la misma manera que un diagrama de actividad.

Modelo de Fragmentos.-

Este modelo está definido como la descomposición del sistema en unidades significativas de información, conocidas como *fragmentos (estructurales)* y los hipervínculos entre dichos fragmentos. Dado que en el modelo de casos de uso extendido se representan dos tipos de servicios de usuario básicos, en este modelo también se incorporan dos tipos de fragmentos:

- *Fragmentos Estructurales.-* es una pieza significativa de información que se lo muestra al usuario mediante una página Web.
- *Fragmentos Funcionales.-* representa una página Web interactiva, es decir permite representar la interacción del usuario con el SIW.

El proceso inicia al tomar los servicios básicos estructurales y sus relaciones para dar origen a los fragmentos estructurales con sus respectivos hipervínculos. Sus atributos se los obtendrá del modelo conceptual de datos. Estos fragmentos se esteriotipan con <<SS>>.

Modelo de Fragmentos Extendido.-

Se lo especifica como un modelo conceptual de fragmentos al que se le agregan los fragmentos funcionales.

El proceso comienza con la identificación de los fragmentos, donde cada servicio básico funcional del modelo de casos de uso extendido se transforma automáticamente en fragmentos funcionales que se esteriotipan con <<FS>>.

Luego se identifican los atributos que son utilizados en cada fragmento, es decir los atributos que son exclusivamente utilizados en esa página Web; los cuales provienen del modelo conceptual de datos.

Modelo de Navegación Extendido.-

Se lo define como el modelo de fragmentos extendido al que se le agregan estructuras navegacionales como menús, índices, vistas, etc., es decir incorpora los fragmentos extendidos y funcionales, además de las rutas específicas que se encuentran en el diagrama de actividad.

El proceso inicia al identificar las estructuras de navegación antes mencionadas. Luego se propone especificar un menú principal, el cual incorpore una entrada para cada servicio de usuario, siendo este el comienzo de cada una de las rutas que se identificaron en el modelo de composición de servicios.

Modelo de Servicios.-

Finalmente dentro del aspecto de funcionalidad a nivel de PIM, se encuentra este modelo, el cual permite modelar cada uno de los servicios conceptuales que ofrece el sistema de información Web. Para este modelo se propone la utilización de diagramas de colaboración de UML.

Un diagrama de colaboración es un tipo de diagrama de interacción que muestra una interacción organizada basándose en los objetos que toman parte en la interacción y sus relaciones, agregando los mensajes que pueden ser enviados entre ellos. Este diagrama muestra las relaciones que existen entre los roles de los objetos, es decir se centran en las responsabilidades de cada objeto, en lugar del tiempo en que los mensajes son enviados.

Su representación gráfica consiste en una serie de objetos con los enlaces entre los mismos y los mensajes que se intercambian, los cuales se representan mediante flechas que van junto al enlace por el que circulan, donde se indica su nombre y en caso de tener parámetros se los especifica dentro de paréntesis. Cada uno de estos mensajes tiene un número de secuencia que expresa cuál es el mensaje que le precede. El primer nivel de la secuencia es 1, y los mensajes que son enviados durante la misma llamada a un método se numeran 1.1, 1.2 y así sucesivamente según los niveles que sean necesarios, en cuyo caso el nivel 1.2 significa que el mensaje con número de secuencia 1 no acaba de ejecutarse hasta que no se han ejecutado todos los mensajes en ese nivel.

Modelos Específicos de la Plataforma (PSM).-

Un PSM proporciona un modelo en términos de constructores de implementación que estén disponibles en una plataforma o tecnología específica. Los modelos PSM están basados en una tecnología objeto-relacional y XML concernientes con la modelación lógica y física, y las fases de implementación. La tecnología objeto-relacional es usada para representar el aspecto de contenido mientras que la tecnología XML representa los aspectos de hipertexto y presentación.

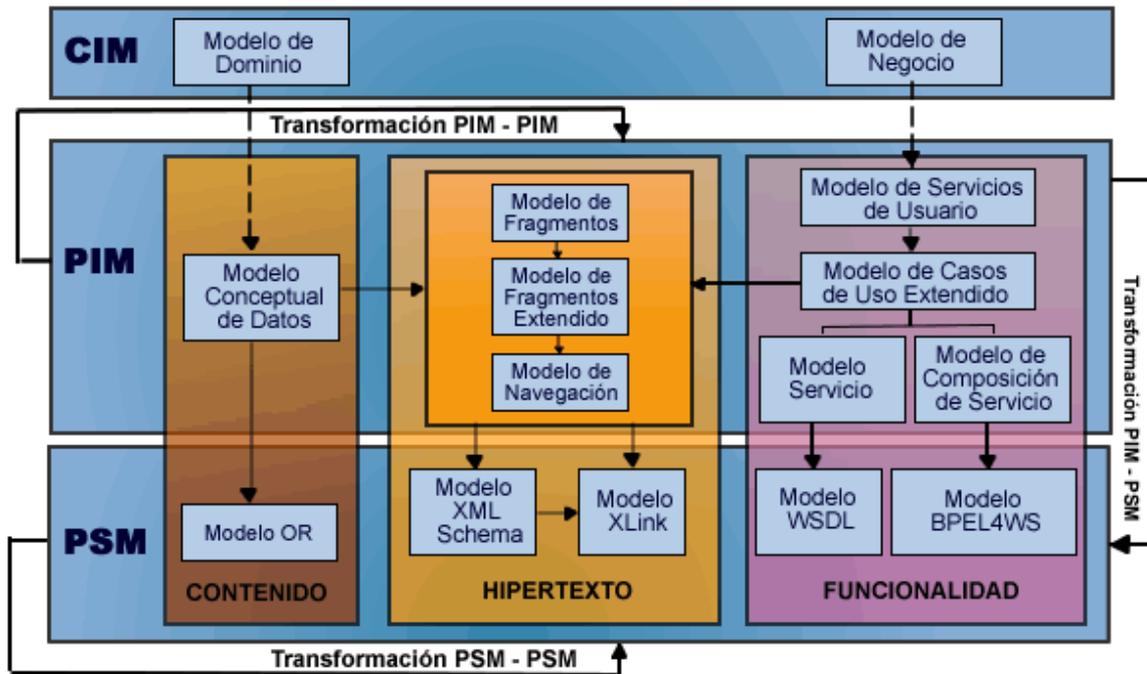


Figura 3.15: Proceso de desarrollo de MIDAS. Modelos Específicos de la Plataforma

Modelo Objeto Relacional.-

Este es el modelo final del aspecto de contenido, Para generar este modelo se debe tomar todas las clases del modelo conceptual de datos refinado y transformarlos en tipos de objetos. Además se deben mantener las relaciones que existen entre ellos. Es un modelo lógico de datos.

El modelo Objeto - Relacional o modelo Relacional Extendido es el camino más "directo" que corrige los problemas del modelo Relacional. El modelo de datos "Extended Relational" es usado en sistemas basados en el modelo de datos relacional incorporando objetos, procedimientos, versiones y nuevas capacidades.

No existe un sólo modelo objeto-relacional, hay una diversidad de estos modelos, cuyas características dependen del camino, y el grado en el cual las extensiones son realizadas. Todos los modelos forman parte del modelo relacional básico (tablas, relaciones y SQL), todos tienen como base el mismo

concepto de objeto, y todos tienen la capacidad de almacenar datos y procedimientos en la base de datos.

Según [Marcos, 2005]; SQL:2003, el modelo Objeto-Relacional tiene las siguientes extensiones (nuevas características) con respecto al modelo Relacional propuesto en el estándar SQL:1992:

- Extensiones Objeto-Relacionales:
 - Tipos Definidos por el Usuario
 - Tipos fila y referencia
 - Tipos colección
 - Soporte de objetos grandes (LOBs)
- Triggers
- Procedimientos almacenados y funciones definidas por el usuario
- Consultas recursivas
- Extensiones OLAP.

Además plantea como beneficios del modelo Objeto-Relacional la capacidad de extensibilidad, el poder de expresión, la reusabilidad, la integración, etc.

- Extensibilidad: Necesidad de nuevas capacidades de los DBMS¹⁰:
 - Nuevos tipos de datos que representen de mejor manera el dominio de la aplicación (dinero, imagen, línea, etc.).
 - Nuevas operaciones para soportar el comportamiento de los tipos.
- Integración: modelos relacional y orientado a objetos en un solo lenguaje.
- Poder de expresión: Necesidad de soportar objetos y relaciones complejas.
- Reusabilidad: Compartir librerías de tipos existentes.
- Nuevas consultas: Mejores capacidades de representación y filtrado.

¹⁰ Data Base Management System: Sistema de administración de Base de Datos

En [Marcos, 2005] se muestra de forma clara cuales son las nuevas características que soporta el modelo Objeto-Relacional, referenciadas en SQL: 2003:

- Large Objects (LOBs).
 - Binary
 - Character
- Tipos Definidos por el Usuario (UDTs).
 - Distinct types
 - Structured types
- Constructores de tipo.
 - Row types.
 - Referenced types
- Tipos colección.
 - Arrays
- Jerarquías de tablas y de vistas.
- Métodos, funciones y procedimiento definidos por el usuario.

En la *Figura 3.16* se muestra una comparación entre los tipos de modelos Relacional y Objeto-Relacional

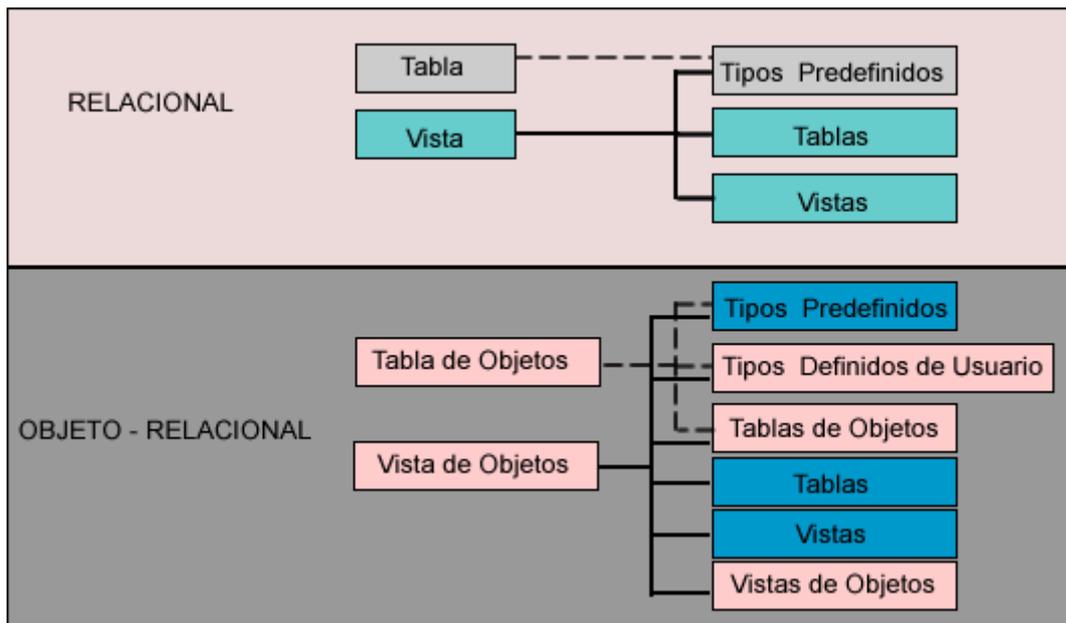


Figura 3.16: Modelo Relacional y Modelo Objeto-Relacional

Modelo XML Schema.-

Permite modelar lógicamente los fragmentos. XML Schema es un documento que permite definir el contenido y la estructura de un tipo de documento XML, al describir los elementos y atributos que puede contener un documento además de la forma en la que se pueden definir dentro de una estructura jerárquica de un documento válido [Vela, Marcos, 2005].

Para generar un modelo XML Schema se propone utilizar UML, puesto que XML Schema no posee su propia notación gráfica. Pero debido a que este lenguaje no permite representar directamente los esquemas, es necesario realizar una extensión al mismo mediante bloques de construcción por medio de un conjunto de estereotipos, valores etiquetados y restricciones.

Una extensión de UML estaría compuesto por una breve descripción, la lista y descripción de los estereotipos, valores etiquetados y restricciones, y un conjunto

de reglas que permitan determinar si un modelo está bien construido, es decir si es válido o no.

- Para realizar XML Schemas, se debe tomar en cuenta lo siguiente:
 - Para elegir el elemento estereotipado se utiliza el criterio <<ELEMENT>> como estereotipo debido a que está definido explícitamente en el XML Schema.
 - Los atributos de los tipos ELEMENT son considerados atributos estereotipados de las clases que representan el tipo ELEMENT.
 - Para los atributos que tengan un enlace a otro tipo ELEMENT se utilizarán los estereotipos IDREF o IDREFS (son referencias a palabras ID definidas por otros atributos. IDREF es una palabra simple e IDREFS es una lista de palabras separadas por espacios. Es decir en IDREF el valor del atributo debe contener un puntero a algún otro elemento, referenciado mediante el valor de un atributo de tipo ID.)
 - Cada tipo <<ELEMENT>> tiene que estar enlazado con una asociación de <<Compose>> o <<BelongTo>> con otro <<ELEMENT>>, <<complexType>> o <<simpleType>>.
 - Un atributo de tipo <<IDREF>> o <<IDREFS>> en un <<ELEMENT>> implica una asociación de <<Compose>> con otro <<ELEMENT>>.
 - Un <<ELEMENT>> puede contener una colección de atributos.

El tipo *ELEMENT* permite la representación de un elemento de XML Schema. Su formato consiste en: el nombre del elemento, su tipo, y el número mínimo y máximo de ocurrencias.

Los *atributos*, pertenecen a un tipo <<ELEMENT>> y solo a una clase <<ELEMENT>>. Su formato contiene: el nombre del atributo, su tipo, la restricción (si es requerido u opcional), y el valor (ya sea por defecto o fijo).

El tipo <<*complexType*>>, es un metamodelo del tipo clase y esta formado por varios elementos. Debe estar relacionado a través de una asociación <<Compose>> con los elementos que forman el tipo complejo.

El tipo <<*simpleType*>>, también es un metamodelo del tipo clase que está derivado de un tipo predefinido o un *simpleType* previamente definido. Debe estar relacionado por medio de una asociación <<Compose>> con el elemento o atributo que lo usa. Su formato consta del Tipo base y las restricciones propias del tipo base.

Una asociación *Compose*, es un tipo especial de asociación unidireccional que enlaza una clase <<ELEMENT>> con la clase referenciada por un atributo IDREF/IDREFS o un tipo simple o complejo que lo sea utilizado por el <<ELEMENT>>. Si el atributo es de tipo IDREFS, la flecha estará rellena. Si se trata de un atributo IDREF, la flecha estará sin rellenar. La dirección de la asociación se representará con la flecha al final de la clase, que se referencia por el tipo ELEMENT. Hay que tomar en cuenta que sólo se puede usar para unir dos tipos <<ELEMENT>>, uno de los cuales debe tener un atributo cuyo tipo sea IDREF/S, que haga referencia a otro tipo ELEMENT.

Una asociación *BelongTo*, también es un tipo especial de asociación unidireccional que enlaza una clase <<ELEMENT>> pero con la clase que representa a su elemento padre. La parte de la asociación con el rombo representa el elemento padre. En el extremo opuesto pueden aparecer el mínimo número y/o máximo de ocurrencias. Un número máximo de ocurrencias ilimitado se representará mediante una N. Esta asociación sólo se puede usar para unir dos <<ELEMENTS>>, donde uno es el padre del otro.

A continuación se presenta un ejemplo para comprender de mejor manera los conceptos mencionados anteriormente:

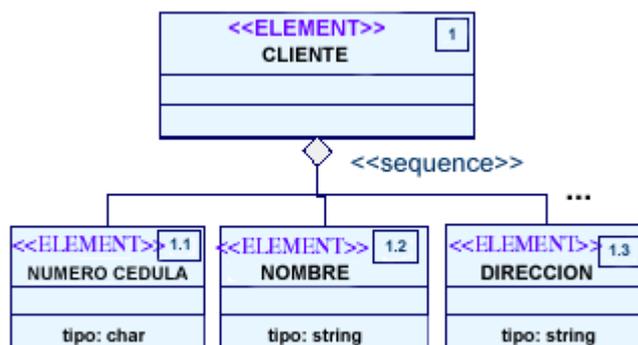


Figura 3.17: Ejemplo de un Modelo XML Schemas

Modelo X-Link.-

XLink o Lenguaje de Enlaces (vínculos) XML, es una recomendación del World Wide Web Consortium¹¹, es un lenguaje que permite incluir elementos en documentos XML para crear vínculos entre recursos XML, es decir, permite crear elementos de XML que describen relaciones cruzadas entre documentos, imágenes y archivos de la red.

¹¹ W3C: <http://www.w3.org/>

Dicho de otra forma, XLink, describe un camino estándar para añadir hipervínculos en un archivo XML. Es un mecanismo de vinculación a otros documentos XML.

Así este lenguaje permite realizar lo siguiente:

- Crear vínculos bidireccionales.
- Crear una relación de vínculos entre varios documentos.
- Agregar a un vínculo información acerca del mismo.
- Crear y describir vínculos a documentos en multitud de ubicaciones.

Cualquier elemento XML puede ser un enlace XLink, así como también cualquier elemento puede ser enlazado por un XLink. Un elemento será un enlace XLink por el simple hecho de adicionarle una serie de *atributos* que existen dentro del espacio de nombres (namespace) xlink, definido por la norma "http://www.w3.org/XML/XLink/1.0" o superior.

Ejemplo:

```
<ejemplo xmlns:xlink="http://www.w3.org/XML/XLink/1.0">
  <enlace xlink:type="simple"
    xlink:href="http://www.w3.org/TR/xlink/"> La página de XLink
  </enlace>
</ejemplo>
```

Entre los atributos que pueden acompañar a un XLink pueden estar:

- Show: con este atributo se puede referir cómo se verá el resultado del hecho de seguir el vínculo. Sus posibles valores son:

- Replace: reemplaza el documento actual por el recurso al que apunta el enlace, en otras palabras, abre el enlace en la misma ventana.
- New: abre una nueva ventana del navegador con el documento destino del enlace.
- Parsed: el contenido del texto apuntado se incluye en lugar del enlace y se procesa como si fuera parte del mismo documento de origen.

Ejemplo:

```

<ejemplo xmlns:xlink="http://www.w3.org/XML/XLink/1.0">
  <enlace xlink:type="simple"
    xlink:href="http://www.w3.org/TR/xlink"
    xlink:show="replace"> Carga la página de XLink
  </enlace>
  ...
  <nuevav-enlace xlink:type="simple"
    xlink:href="http://www.espe.edu.ec/"
    xlink:show="new">
    Abre en una nueva ventana la página de la ESPE
  </nuevav-enlace>
  ...
  <incluir xlink:type="simple"
    xlink:href="informacion.xml"
    xlink:show="parsed">
    Incluye información
  </incluir>
  ...
</ejemplo>

```

- Actuate: Permite mostrar cuando se debe proceder a buscar el destino al que apunta un enlace. Sus posibles valores son:
 - User: cuando el usuario presione o dé alguna orden para seguir el vínculo.
 - Auto: el vínculo se sigue automáticamente.

XLink también dispone de un tipo de enlaces conocidos como vínculos extendidos que pueden tener más de dos recursos por vincular. Estos recursos se especifican con el elemento Locator.

Ejemplo:

```

<ejemplo xmlns:xlink="http://www.w3.org/XML/XLink/1.0">
...
<enlaces xlink:type="extended">
  <locator xlink:type="locator"
    xlink:href="http://www.espe.edu.ec"
    xlink:show="replace"
    xlink:actuate="user"/>
  <locator xlink:type="locator"
    xlink:href="http://www.epn.edu.ec"
    xlink:show="new"
    xlink:actuate="user"/>
  <locator xlink:type="locator"
    xlink:href="http://www.espol.edu.ec"
    xlink:show="new"
    xlink:actuate="auto"/>
  <locator xlink:type="locator"
    xlink:href="http://www.puce.edu.ec"
    xlink:show="replace"
    xlink:actuate="auto"/>
  Universidades
</enlaces>
...
</ejemplo>

```

Modelo WSDL.-

El lenguaje de descripción de servicios Web WSDL¹² fue propuesto por el W3C. Permite modelar la interfaz de un Servicio Web en formato XML; el mismo que define: operaciones, mensajes, protocolos y localizaciones. Además permite separar la descripción *abstracta* (tipos de puerto o interfaces) y los detalles *concretos* que se manejan, como los enlaces que se dan a un protocolo de red o a los formatos de mensajes.

El modelo WSDL es una extensión a UML, que incluye estereotipos, valores etiquetados y restricciones.

WSDL proporciona la información necesaria al cliente para interactuar con el servicio Web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de la red (por ejemplo: SOAP sobre HTTP).

WSDL utiliza la notación basada en XML Schema para describir formatos de mensajes, que lo hace neutral respecto del lenguaje de programación, además de estar basado en estándares, lo que lo hace apropiado para describir interfaces de servicios Web XML y accesibles muchas plataformas y lenguajes de programación. Además de detallar el contenido de los mensajes, WSDL precisa dónde está disponible el servicio y qué protocolo de comunicaciones utilizar para hablar con el servicio. Esto significa que WSDL define todo lo que necesita un programa para interactuar con un servicio Web XML.

WSDL define los servicios como colecciones de puntos finales de red o puertos. En WSDL, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos [Christensen, 2000].

¹² WSDL: Web Services Description Language

WSDL define los servicios de red utilizando los siguientes elementos:

- **Types:** provee definiciones del tipo de datos que utiliza el sistema de tipos para detallar el intercambio de mensajes.
- **Message:** representa la definición abstracta y escrita de los datos que se transmiten. Un mensaje está compuesto de partes lógicas que están asociadas a una definición dentro del sistema de tipos.
- **Operation:** constituye la descripción abstracta de una acción que es permitida por el servicio.
- **Port Type:** es un conjunto de operaciones abstractas permitidas por uno o más puntos finales. Cada operación hace referencia a un mensaje entrante y a mensajes salientes.
- **Binding:** define la especificación del protocolo y del formato de datos para las operaciones y mensajes determinados en un portType establecido.
- **Port:** es el único punto final de comunicación que especifica la dirección para un enlace. Está definido como la combinación de un enlace y una dirección de red.
- **Service:** es un conjunto de puntos finales (puertos) relacionados.

La *Figura 3.18* muestra la relación entre los diferentes elementos de WSDL:

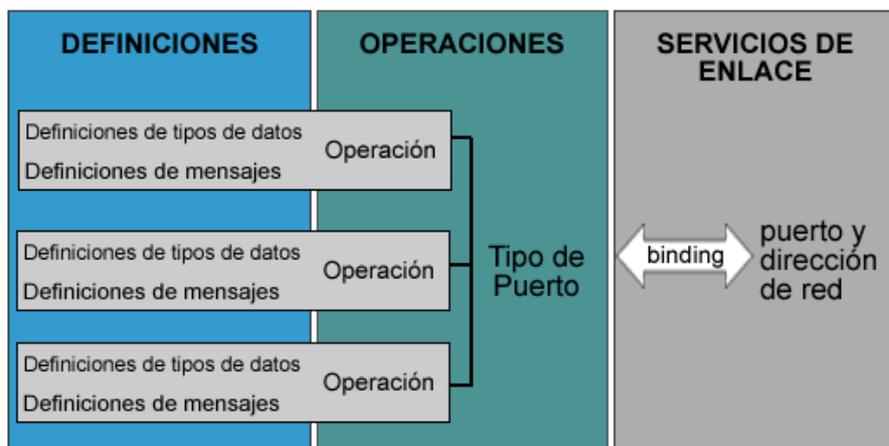


Figura 3.18: Relación entre los elementos de WSDL

El ejemplo muestra un documento XSDL con todos sus elementos:

```
<?xml version="1.0"?>
<definitions name="Cotizacion"
targetNamespace="http://www.ejemplo.com/cotizacion.wsdl"
  xmlns:tns="http://www.ejemplo.com/cotizacion.wsdl"
  xmlns:xsd1="http://www.ejemplo.com/cotizacion.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types> <schema targetNamespace="http://www.ejemplo.com/cotizacion.xsd"
    xmlns="http://www.w3.org/1999/XMLSchema">
    <element name="PrecioComercioRequest"> <complexType>
      <all> <element name="Teletipo" type="string"/></all>
    </complexType> </element>
    <element name="PrecioComercio"> <complexType>
      <all> <element name="Precio" type="float"/> </all>
    </complexType> </element>
  </schema> </types>
  <message name="GetLastInputPrecioComercio">
    <part name="body" element="xsd1: PrecioComercio"/> </message>
  <message name="GetLastOutputPrecioComercio">
    <part name="body" element="xsd1:ResultadoPrecioComercio"/> </message>
  <portType name="CotizacionPortType"> <operation name="GetLastPrecioComercio">
    <input message="tns:GetLastInputPrecioComercio"/>
    <output message="tns:GetLastOutputPrecioComercio"/>
  </operation> </portType>
  <binding name="CotizacionSoapBinding" type="tns:CotizacionPortType">
    <soap: binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastPrecioComercio">
      <soap:operation soapAction="http://www.ejemplo.com/GetLastPrecioComercio"/>
    <input> <soap:body use="literal" namespace="http://www.ejemplo.com/cotizacion.xsd"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/> </input>
    <output> <soap:body use="literal" namespace="http://www.ejemplo.com/cotizacion.xsd"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/> </output>
    </operation>
  </binding>
```

```
<service name="CotizacionService">
  <documentation>Servicio de Ejemplo</documentation>
  <port name="CotizacionPort" binding="tns:CotizacionBinding">
    <soap:address location="http://www.ejemplo.com/cotizacion"/>
  </port>
</service>
```

Modelo BPEL4WS.-

Business Process Execution Language for Web Services permite especificar procesos complejos desarrollados como composiciones de servicios Web y luego ejecutarlos automáticamente. Además proporciona una infraestructura *puente* (middleware) poderosa para facilitar la ejecución de instancias de procesos de negocio que pueden llegar a tener una larga duración.

Es importante que los BPEL4WS se construyan y revisen a un nivel profundo en la etapa de diseño, y prever posibles fallos como: problemas en la red, en los sistemas remotos, servicios que no respondan, etc.; para que estos procesos respondan adecuadamente frente a este tipo de percances.

Debido a que BPEL4WS es una notación XML para especificar tanto *coordinaciones* como *composiciones*, es necesario describir en qué consisten cada uno de estos dos conceptos:

- *Coordinación.-* (procesos abstractos) es la especificación de interacciones complejas que se dan entre diferentes servicios Web: como las relaciones entre las operaciones, o la coordinación entre los servicios Web para realizar tareas complejas.

- *Composición de servicios.*- (procesos ejecutables) se describe como la implementación de servicios Web complejos, es decir que internamente llaman a otros servicios Web para realizar una determinada tarea.

Un proceso BPEL4WS especifica flujo de actividades, papeles de entidades que llaman o dan servicio Web al proceso, tipos de puertos propuestos e información de correlación (relacionar mensajes con las distintas instancias del proceso). Los procesos que son ejecutables se los puede correr de forma automática a través de un *motor BPEL4WS*.

El modelo BPEL4WS permite estructurar varias cosas como:

- Actividades
- Variables
- Composición jerárquica mediante servicios Web.
- Mecanismos para manejar fallos y compensaciones.

Las actividades (composición secuencial o concurrente, ejecución condicional y bucles) corren bajo el entorno que da un *scope*, el cual las contiene además de los datos. Si existen actividades que se encuentran contenidas en el mismo *scope*, estas comparten su entorno.

Los *scopes* pueden formar una jerarquía de *scopes*, ya que estos pueden contener a su vez a otros *scopes*, en esta jerarquía el proceso es la raíz.

Los datos se almacenan en variables, y solo las actividades que consten en la jerarquía del *scope* podrán acceder a los datos.

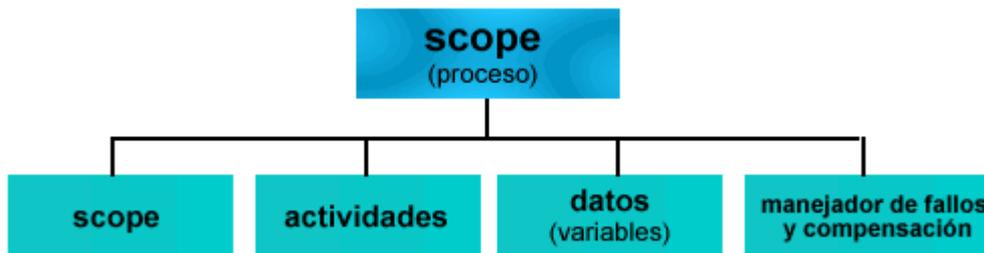


Figura 3.19: Jerarquía de Scopes

Existen también los manejadores de fallos y compensación que permiten controlar las excepciones, su mecanismo es parecido al utilizado en Java. El proceso consiste en que una actividad en ejecución detecta un fallo y notifica la ocurrencia de un error, este fallo puede ser capturado y tratado ya sea en su propio scope o en el inmediato superior. Y finalmente el mecanismo de compensación ejecutará el comportamiento transaccional que se debe ejecutar, es decir, por ejemplo, que si un scope relacionado no pudo terminar debido a un error, entonces se debe deshacer todas las actividades realizadas anteriormente en el scope en el que se propagó el fallo.

3.3- La tecnología .NET

3.3.1- Introducción

Microsoft .NET¹³ es una plataforma de software que integra información, personas, sistemas y hardware. .NET vincula una gran diversidad de tecnologías que van desde negocios a tecnologías de uso personal, de dispositivos móviles a servidores empresariales, permitiendo el acceso a información relevante, en el momento que sea requerida.

¹³ Punto NET: New Enhanced Technology

Esta plataforma ha sido desarrollada tomando como base los estándares de Servicios Web XML que permiten simplificar la escritura de los sistemas que admitan la interconexión e interoperabilidad entre varios de ellos.

El objetivo de esta tecnología es el de permitir que los sistemas y aplicaciones, nuevas o existentes, enlacen sus datos y operaciones, con independencia del hardware utilizado, del sistema operativo usado, o del lenguaje de programación empleado para su creación.

Los Servicios Web XML constituyen al mismo tiempo una metodología y una capa de transporte para el paso de información entre sistemas operativos, dispositivos y redes diferentes.

.NET ofrece la capacidad de desarrollar, implementar, administrar y utilizar soluciones informáticas enlazadas a través de Servicios Web XML¹⁴, de una forma segura y rápida. Estos sistemas persiguen que las organizaciones y el acceso a la información se integren de manera más veloz y ágil.

3.3.2- Web Services XML

Los Servicios Web son la más innovadora tecnología de desarrollo en el Web, que usan tecnologías programables y reutilizables que aprovechan la flexibilidad de Internet. Con esta tecnología se puede tener un sinnúmero de aplicaciones conectadas en red, que se estén ejecutando o no en diferentes plataformas, que entreguen información a todos sus usuarios.

¹⁴ XML: Extended Mark-up Language, lenguaje de marcado extendido, propuesto por W3C (World Wide Web Consortium)

Los Servicios Web XML tienen como base un conjunto de estándares abiertos, que incluyen XML, SOAP¹⁵, WSDL¹⁶ y UDDI¹⁷, los cuales son administrados por el World Wide Web Consortium (W3C).

Al utilizar la tecnología .NET se aprovecharán en gran medida las ventajas que ofrecen los Servicios Web y lo que ellos engloban.

3.3.3- El .NET Framework

.NET Framework (Marco de trabajo .NET) es un entorno que permite construir y ejecutar Servicios Web XML. Es un componente integral de Windows que tiene la capacidad de ser un modelo de programación y un ambiente de ejecución para:

- Aplicaciones Cliente – Servidor,
- Aplicaciones Web, y
- Servicios Web XML.

El .NET Framework está diseñado para cumplir los siguientes objetivos:

- Proveer un ambiente de programación orientada a objetos, en el que el código de los objetos se pueda:
 - Almacenar y ejecutar de forma local,
 - Ejecutar de forma local pero distribuida en Internet, o
 - Ejecutar de forma remota;

Utilizando las herramientas que provee el Visual Studio .NET.

- Suministrar un entorno de ejecución que:

¹⁵ SOAP: Simple Object Access Protocol, protocolo de comunicaciones para los Servicios Web XML

¹⁶ WSDL: Web Services Description Language, lenguaje de descripción de Servicios Web

¹⁷ UDDI: Universal Discovery Description and Integration, páginas amarillas de los Servicios Web, dónde encontrar los Servicios Web.

- Reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Impulse la ejecución segura del código.
- Disminuya e incluso elimine los problemas de rendimiento de los ambientes en los que se utilizan secuencias de comandos o intérpretes de comandos.
- Cimentar la comunicación en estándares (como SOAP) para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.
- Procurar al desarrollador una experiencia coherente entre tipos de aplicaciones muy diferentes, por ejemplo Aplicaciones Windows y Web.

La *Figura 3.20* muestra en resumen la Arquitectura global del .NET Framework.

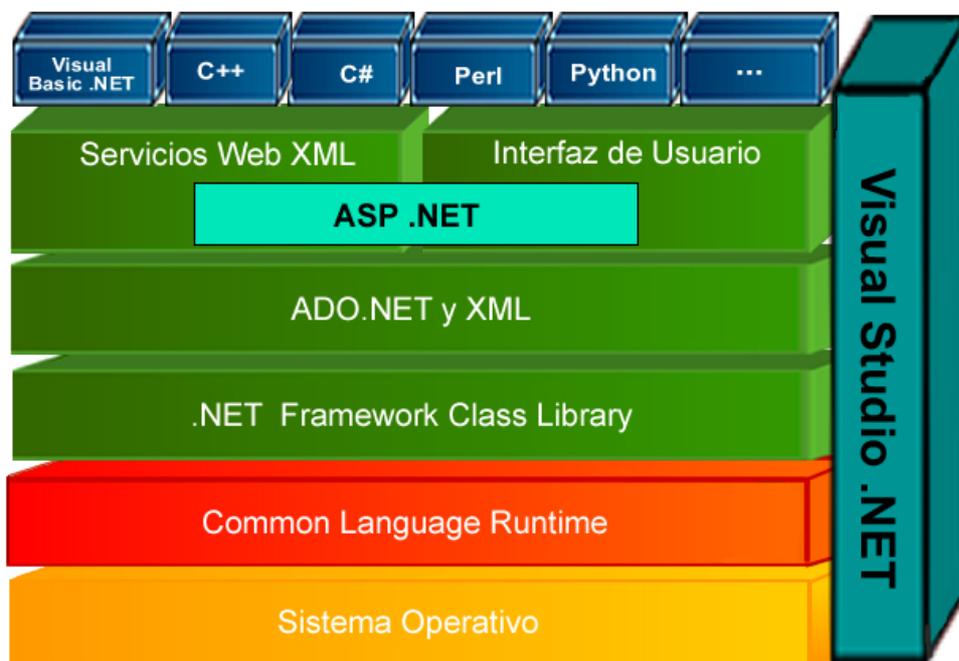


Figura 3.20: Arquitectura de .NET

El marco de trabajo de .NET tiene dos componentes principales:

- El motor de tiempo de ejecución: Common Language Runtime (CLR), y
- La biblioteca de clases de .NET: Framework Class Library (FCL).

Common Language Runtime es el núcleo del .NET Framework, es el fundamento de la tecnología, ya que se usa en el desarrollo de los componentes y es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes que soporta el Framework. El motor de tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales como el control de los recursos del sistema (administración de memoria, administración de subprocesos e interacción remota), a la vez que emplea una precisa seguridad a los tipos y otras formas de especificación del código que fomentan su seguridad y solidez. El código destinado al CLR se denomina *código administrado*, a diferencia del resto de código, que se conoce como *código no administrado*.

La biblioteca de clases, es una rica colección orientada a objetos de clases, interfaces tipos reutilizables que se pueden usar para desarrollar aplicaciones que van desde Windows Forms (tradicional herramientas de interfaz gráfica de usuario GUI) o de línea de comandos hasta las más recientes aplicaciones basadas en ASP.NET, como los Web Forms y los servicios Web XML.

En la *Figura 3.21* se muestra la relación de Common Language Runtime y la Biblioteca de Clases con el sistema y las aplicaciones en conjunto.



Figura 3.21: .NET Framework en contexto¹⁸.

Características de Common Language Runtime

Common Language Runtime (CLR) es el motor de ejecución de las aplicaciones .Net. Administra la memoria del sistema, ejecución de código, comprobación de la seguridad del código, ejecución de subprocessos, compilación, entre otros servicios del sistema. Estas características son propias del *código administrado* que se ejecuta en CLR.

Con relación hacia aspectos de seguridad, los componentes administrados reciben grados de “confianza” diferentes, en función de una serie de factores

¹⁸ Imagen tomada de: Guía del desarrollador de .NET Framework (MSDN)

entre los que se incluye su origen que puede ser Internet, red corporativa o host local.

El CLR impone seguridad en el acceso al código, las características de seguridad del motor de tiempo de ejecución permiten que el software legítimo implementado en la red sea muy variado.

Adicionalmente el CLR demuestra la solidez del código mediante la implementación de una infraestructura estricta de comprobación de tipos y código llamada Sistema de Tipos Común (CTS, Common Type System).

El entorno administrado del CLR controla automáticamente la disposición de los objetos, administra las referencias a éstos y los libera cuando ya no se utilizan. Esta administración automática de la memoria soluciona dos errores muy frecuentes en las aplicaciones: la pérdida de memoria y las referencias no válidas a la memoria.

En el CLR el código administrado nunca se interpreta, lo que mejora el rendimiento. Usando el denominado compilador JIT (Just-In-Time) permite ejecutar todo el código administrado en el lenguaje máquina nativo del sistema en el que se ejecuta.

La interoperabilidad entre el código administrado y no administrado¹⁹ permite que los programadores continúen utilizando componentes COM y DLL que se requieran.

Biblioteca de clases de .NET Framework

El conjunto de clases de .NET Framework es una rica colección de clases, interfaces, tipos reutilizables que se integran estrechamente con CLR y que optimizan el desarrollo de aplicaciones.

¹⁹ Podría llamarse a esto la interacción entre software actual y software “antiguo”

La colección de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado puede derivar funciones. El .NET Framework posee un sistema de tipos universal (CTS). Este sistema permite que el desarrollador pueda interactuar con la biblioteca de clases .NET con los creados por él mismo. De esta forma se aprovechan las ventajas propias de la programación orientada a objetos, como la herencia de clases predefinidas para crear nuevas clases, o el polimorfismo de clases para modificar o ampliar funcionalidades de clases ya existentes.

Del conjunto de clases de .NET Framework, se pueden obtener tipos predefinidos para desarrollar los siguientes tipos de aplicaciones y servicios:

- Aplicaciones de consola
- ADO.NET para conectar las aplicaciones a bases de datos.
- Aplicaciones GUI de Windows
- Servicios de Windows
- Aplicaciones de ASP.NET
- Servicios Web XML

Código administrado en el servidor

ASP.NET es la parte de .NET Framework dedicada al desarrollo de aplicaciones para el Web. Pero ASP.NET no es sólo un host de CLR, se trata de una completa arquitectura para el desarrollo de sitios Web y objetos distribuidos en Internet mediante código administrado. Los Web Forms y los Servicios Web XML utilizan IIS (Servicios de Internet Information Server) y ASP.NET como mecanismos de publicación de las aplicaciones y ambos disponen de un conjunto de clases compatibles en .NET Framework.

Los servicios Web XML, son componentes distribuidos de aplicaciones de servidor similares a los sitios Web comunes, que permiten la comunicación a través de Internet entre diferentes equipos y distintos sistemas. Sin embargo, a diferencia de las aplicaciones basadas en el Web, los componentes de servicios Web XML no tienen interfaz de usuario y no están orientados a un browser. En su lugar, los servicios Web XML tienen componentes de software reutilizables diseñados para que los usen otras aplicaciones, que pueden ser: aplicaciones cliente tradicionales, aplicaciones basadas en el Web o, incluso, otros servicios Web XML.

3.3.4- Visual Basic.NET

Visual Basic .NET es un lenguaje de programación RAD (Rapid Application Development), que proporciona a los desarrolladores una herramienta productiva y poderosa para resolver los desafíos actuales en el desarrollo de aplicaciones para Windows y la Web. Es un ambiente óptimo para construir, desarrollar y ejecutar los Servicios Web XML.

En principio el equipo de desarrollo de herramientas de Microsoft, y específicamente de Visual Basic, se planteo como objetivos “armar” una nueva versión de Visual Basic, que fuera compatible con el nuevo entorno de desarrollo que proponía Microsoft (.Net Framework) y modificar aquellas cosas que se venían usando desde versiones anteriores (Visual Basic 4, 5 y 6) y que quedaron obsoletas y en muchos casos dificultaban una programación adecuada y acorde a estas nuevas tecnologías.

Se tomaron en cuenta algunos criterios para mejorar el lenguaje, que se pueden sintetizar así:

- Hacer el lenguaje aún más simple de utilizar que el mismo VB 6.
- Facilitar el camino del aprendizaje.
- Corregir cuestiones de diseño: que el entorno de desarrollo no tenga características particulares en algunos elementos (que el modelo de programación sea común en todos los casos – lenguajes –).
- Agregar características de orientación a objetos completas.
- Vincular al lenguaje directamente con el .Net Framework.
- Restringir los cambios a lo estrictamente necesario para no alejarse del lenguaje original.

Visual Basic .Net se relaciona con el .Net Framework, básicamente utilizando el Motor de Ejecución Común (Common Language Runtime - CLR) accediendo a todos los servicios de la plataforma .Net, tiene por lo tanto interoperabilidad entre lenguajes. Eso significa que Visual Basic puede llamar código u objetos escritos en C#²⁰ o en C++, etc.; inclusive puede heredar de cualquiera de estos lenguajes o estos pueden heredar desde Visual Basic, e incluso si es necesario se puede manejar interoperabilidad con objetos COM.

Visual Basic respetando lo que el .Net Framework indica, implementa y organiza su código a través de la creación de Namespaces²¹. El .Net Framework establece a partir de System, un conjunto de Namespaces jerárquico para determinar distintos elementos de la funcionalidad. De la misma forma se debe

²⁰ Se lee Ce Sharp.

²¹ Namespace es un mecanismo de ordenamiento del código, debiendo este ser estructurado de forma jerárquica.

respetar esta estructura y generar Namespaces propios para mantener organizado el código.

Un Assemblies es el conjunto de datos que describe toda la funcionalidad de un conjunto de código dado. Esto es el momento en que se compila el código y se genera la DLL (Biblioteca de Carga Dinámica) o el EXE (Archivo ejecutable), se genera además de forma automática el Assemblies correspondiente a este conjunto de código. Ese Assemblies puede estar distribuido en uno o más archivos, dependiendo de la forma como se arme un proyecto. Ese archivo o archivos contienen lo que se denomina la "Metadata"²².

La Metadata establece una descripción de si mismo, los métodos que expone, los tipos que implementa, que otros Assemblies necesita (referencias externas), que version tiene, que mecanismos de seguridad, etc.

Estos son principalmente los factores que toma en cuenta Visual Basic .Net para interactuar adecuadamente con el .Net Framework.

Para la creación de aplicaciones Web y Servicios Web XML, Visual Basic .Net presenta características que hacen de este lenguaje una conjunción ideal para usarlo con la metodología propuesta.

Con el Diseñador de Web Forms y el Diseñador XML, el desarrollador puede escoger entre utilizar las características "inteligentes" del IDE para edición de código (Microsoft IntelliSense) y la capacidad para completar etiquetas; o bien, elegir el editor WYSIWYG (lo que ve es lo que se imprime) para crear aplicaciones Web interactivas arrastrando y colocando diferentes elementos en los formularios Web.

²² Descripción de todo lo que expone y todo lo que es un conjunto de código.

Siguiendo ciertos pasos se puede diseñar, programar, depurar y distribuir eficaces servicios Web XML que reduzcan el tiempo de programación al tener acceso y encapsular los procesos de la organización desde cualquier plataforma, que es uno de los objetivos de los Servicios Web XML.

CAPITULO IV

DESARROLLO DEL SISTEMA BASYS

4.1- Metodología MIDAS

4.1.1- Análisis del sistema

Especificación de Requerimientos de Software según IEEE 830

Introducción

Propósito

La presente Especificación de Requerimientos de Software (ERS) del Sistema de Administración de competencias de Básquetbol “Basys” se ha elaborado para ser usada como guía durante el desarrollo del sistema y después de su implementación. Esta ERS define que es lo que el software puede y no puede hacer, así como también los requerimientos tecnológicos mínimos necesarios para el buen funcionamiento del sistema; esto se refiere a las necesidades de Hardware y Software.

Esta ERS puede ser consultada para obtener información sobre la Administración, Funcionamiento y Mantenimiento de Basys así como cualquier otra referencia acerca del sistema. Además en caso de que se requiera realizar mejoras o modificaciones, esta ERS servirá como guía para que cualquier otro ingeniero de software y/o desarrollador conozca el sistema.

La ERS de Basys está dirigida a todo el personal que tiene acceso al sistema, como son el administrador del sistema y el personal calificado como usuario del mismo.

Alcance

- *Identificación del sistema:* BASYS (Sistema de Básquetbol) es un sistema que se implementará para realizar de manera sencilla, rápida y segura la

gestión de los procesos referentes a las competencias de Básquetbol que realiza la Facultad de Educación Física, Deportes y Recreación de la ESPE.

- *Objetivo del Sistema:* automatizar los procesos de control de los campeonatos de básquetbol, equipos, jugadores, y estadísticas de todos y cada uno de los partidos jugados y obtener reportes ágiles y precisos.
- *Módulos a considerar*
 - *Módulo de Seguridad de acceso al sistema.* Mediante el uso de usuarios y contraseñas se controlará el ingreso al sistema en base a perfiles. Se deberá determinar a que módulos puede acceder un usuario.
 - *Módulo de Campeonatos.* Se podrán registrar todas las competencias en la disciplina de baloncesto que la FEFDER quiera administrar, sean estas internas o en las cuales participen las diferentes selecciones de la ESPE.
 - *Módulo de Equipos.* Se mantendrá el control de todos los equipos que han jugado partidos y se encuentran registrados en campeonatos administrados por Basys, sean estos las selecciones de básquetbol de la ESPE, sus rivales u otros equipos.
 - *Módulo de Jugadores.* Se administrará la información más relevante de todos y cada uno de los jugadores registrados (incluyendo la foto en el caso de disponer de esta) en el sistema, y sus respectivas estadísticas.
 - *Módulo de Estadísticas.* Se recogerán estadísticas por jugador y por equipo, en todos y cada uno de los partidos jugados, registrados en

Basys. Además se entregará los reportes necesarios por jugador, equipo y campeonato.

- *Definiciones, Acrónimos y Abreviaturas:*

Tabla 4.1: Tabla de definiciones de acrónimos y abreviaturas

Término	Detalle
Basys	Sistema de Básquetbol desarrollado para la Facultad de Educación Física, Deportes y recreación de la ESPE.
Usuario	Persona quien mediante un nombre de usuario y contraseña accede al sistema con su respectivo perfil.
Perfil	Conjunto de tareas específicas (páginas del sistema) a las que puede acceder un usuario, dependiendo de sus permisos
Alias	Username, nombre de usuario para acceso al sistema
IEEE	Institute of Electrical & Electronics Engineers. Instituto de Ingenieros Eléctricos y Electrónicos.
ERS	Especificación de Requerimientos del Sistema
SIW	Sistemas de Información Web
GUI	Interfaz Gráfica de usuario
GridView	Vistas Web, espacios de visualización de información.
FEFDER	Facultad de Educación Física, Deportes y Recreación de la ESPE
ESPE	Escuela Politécnica del Ejército
IIS	Internet Information Server
XML	Extended Markup Language
UML	Unified Modeling Language

- *Referencias:*

- *IEEE Recommended Practice for Software Requirements Specifications Sponsor: Software Engineering Standards Committee of the IEEE Computer Society Standard IEEE-830-1998.*
- *FIBA. Federación Internacional de Básquetbol.*
- *Expertos de la FEFDER. MsC. Patricio Ponce, entrenador de selecciones de Básquetbol de la ESPE.*

Visión General

La presente ERS muestra una Descripción General acerca del sistema, la cual describe los factores generales que afectan al producto y sus requerimientos sin la necesidad de un nivel profundo de detalle, lo que permitirá entender con facilidad la totalidad del sistema. Más adelante estarán expuestos todos los Requerimientos Específicos, detallados lo suficiente para permitir a los Ingenieros de Software diseñar un sistema que satisfaga dichos requerimientos, y realizar las pruebas que ratifiquen que el sistema cumple con los mismos y con las normas y estándares de calidad que requiere un sistema como Basys.

La ERS de Basys está dividida en tres temas generales, estos son:

1. Introducción.
2. Descripción General.
3. Requerimientos Específicos.

Descripción General

A continuación se detallan los factores que afectan al producto y a sus requerimientos.

Perspectiva del Producto

Basys es un sistema informático que será desarrollado para la FEFDER. El sistema está orientado a la administración de los campeonatos de básquetbol, equipos, jugadores, y estadísticas de partidos.

Interfaces del Sistema

El sistema es totalmente independiente y autónomo, y será desarrollado utilizando una arquitectura n-Capas, utilizando Servicios Web XML. Será diseñado siguiendo la innovadora metodología ágil para el desarrollo de SIW, Midas.

Basys será desarrollado sobre la plataforma Windows, la base de datos a utilizarse es SQL Server 2005 Express Edition, la GUI será desarrollada para ser utilizada en un browser (Tecnología Web).

Interfaces de Usuario

La GUI será diseñada para su correcta y eficiente ejecución en entorno Web. La interacción entre la aplicación y el usuario se realizará mediante páginas ASPX.

Se implementarán diferentes tipos de elementos para manipular la información:

- Ingresos de información a través de formularios Web típicos.
- Validadores de información en los formularios, para verificar la completitud de los datos.
- Salidas mediante Vistas Web (GridView), que mostrarán la información requerida por el usuario.
- Botones para ejecutar los diferentes procesos de mantenimiento de Basys, como guardar, modificar, eliminar y consultar.
- Despliegue de reportes y capacidades de exportación a otros formatos e impresión en papel.

Interfaces de Hardware

En este punto se especifican las características lógicas de cada interfaz entre el software y el hardware en el cual se puede ejecutar la aplicación. Para que el sistema funcione correctamente el equipo que será usado como servidor debe disponer de la siguiente tecnología mínima:

- Pentium IV de 1.4 GHz.
- Memoria RAM de 256 MB.
- Disco Duro de 30 GB.
- Tarjeta de Red 10/100 mbps.
- Monitor, mouse, teclado, CD-ROM.

Las características mínimas de el (los) computador(es) cliente (Host) que requiere la FEFDER son:

- Pentium III de 700 MHz.
- Memoria RAM de 128 MB.
- Disco Duro de 20 GB.
- Tarjeta de Red 10/100 mbps.
- Monitor, mouse, teclado.

Interfaces de Software

Basys deberá contar el siguiente software de base en el equipo servidor:

- Sistema Operativo: Windows 2000, 2000 Server, XP, 2003 Server.
- Servidor Web Internet Information Server version 5.0 o superior
- Base de Datos: SQL Server 2005 Express Edition

Interfaces de Comunicaciones

Para que Basys funcione correctamente, los equipos donde se ejecutará el sistema deben funcionar en un ambiente de red TCP/IP.

Limitaciones de memoria

Para un funcionamiento aceptable del sistema se recomienda 128 MB en RAM en el(los) equipo(s) cliente(s) y 256 MB mínimo en el equipo servidor.

Funciones del Producto

Las funciones que debe realizar el producto están agrupadas en los siguientes módulos:

a) *Funciones del Módulo de Seguridad de acceso al sistema.*

- Consultar todos los usuarios existentes
- Consultar un usuario
- Ingresar un usuario nuevo
- Modificar un usuario existente
- Eliminar un usuario
- Cambiar la contraseña del usuario
- Ingresar el perfil de un usuario
- Consultar el perfil de un usuario
- Modificar el perfil del Usuario
- Eliminar el perfil de un usuario
- Validar el dígito auto verificador de la cédula del usuario
- Validar el ingreso de un usuario al sistema
- Validar el nombre de los usuarios
- Validar el perfil de acceso al sistema de un usuario

b) *Funciones del Módulo de Campeonatos.*

- Consultar todos los campeonatos existentes
- Consultar un campeonato
- Ingresar un campeonato nuevo
- Modificar un campeonato existente
- Eliminar un campeonato
- Validar el nombre de los campeonatos

c) *Funciones del Módulo de Equipos.*

- Consultar todos los equipos existentes
- Consultar un equipo
- Ingresar un equipo nuevo
- Modificar un equipo existente
- Eliminar un equipo
- Validar el nombre de los equipos

d) *Funciones del Módulo de Jugadores.*

- Consultar todos los jugadores existentes
- Consultar un jugador
- Ingresar un jugador nuevo
- Modificar un jugador existente
- Eliminar un jugador
- Validar el dígito auto verificador de la cédula del jugador
- Validar el nombre de los jugadores

e) Funciones del Módulo de Estadísticas.

Sub Módulo Partido

- Consultar todos los partidos existentes
- Consultar un partido
- Ingresar un partido nuevo
- Modificar un partido existente
- Consultar los equipos del partido
- Ingresar un equipo nuevo del partido
- Modificar un equipo existente, del partido

Sub Módulo Información del Partido

- Iniciar un partido nuevo
- Iniciar un cuarto de tiempo
- Cargar los datos de un partido existente
- Obtener el marcador de un partido
- Iniciar el periodo de estadísticas de un jugador en el partido
- Cambiar a un jugador por otro jugador
- Actualizar una asistencia de balón en jugador y equipo
- Actualizar una falta cometida en jugador y equipo
- Actualizar una falta recibida en jugador y equipo
- Actualizar lanzamiento de corta distancia convertido, en jugador y equipo
- Actualizar lanzamiento de corta distancia fallado, en jugador y equipo
- Actualizar lanzamiento de larga distancia convertido, en jugador y equipo

- Actualizar lanzamiento de larga distancia fallado, en jugador y equipo
- Actualizar una pérdida de balón en jugador y equipo
- Actualizar puntos anotados en jugador y equipo
- Actualizar rebote defensivo en jugador y equipo
- Actualizar rebote ofensivo en jugador y equipo
- Actualizar tiro libre convertido en jugador y equipo
- Actualizar tiro libre fallado en jugador y equipo
- Actualizar violación cometida en jugador y equipo

Sub Módulo Estadísticas

- Iniciar las estadísticas de un equipo
- Iniciar las estadísticas de un jugador
- Consultar un campo de estadísticas del equipo
- Consultar un campo de estadísticas del jugador
- Consultar un campo de periodo
- Consultar un campo de resumen
- Calcular las estadísticas de un equipo
- Calcular las estadísticas de un jugador
- Calcular el control de balón del equipo y jugadores
- Calcular la efectividad del equipo y jugadores
- Calcular las faltas del equipo y jugadores
- Calcular la posesión del balón del equipo y jugadores
- Calcular los rebotes defensivos del equipo y jugadores
- Calcular los rebotes ofensivos del equipo y jugadores
- Calcular los tiros de campo del equipo y jugadores

- Calcular los tiros libres del equipo y jugadores
- Normalizar los datos del equipo y jugadores

Características de Usuarios

Los permisos de acceso al sistema se dan por módulo, con lo cual se tienen dos grupos de usuarios:

- Administrador: Que tiene acceso a todos los módulos de Basys, se encarga de gestionar la seguridad de acceso al sistema. Es quien entrega alias y contraseñas de ingreso a Basys a los Usuarios Generales, puede definir todas las configuraciones que estime pertinentes.
- Usuarios Generales: se encargan de proporcionar los datos del sistema, es decir, de alimentar la base de datos de Basys. Pueden tener acceso a todos los módulos, menos al módulo de seguridad del sistema. Es importante aclarar que si se desea que un Usuario tenga acceso solo a consultas, solo debe tener permisos de ingreso al módulo de Reportes.

El sistema permanece en interacción con el usuario durante los siguientes procesos:

- Ingreso de los usuarios al sistema.
- Administración de Usuarios.
- Consulta y modificación de Perfiles de usuario.
- Administración de Campeonatos.
- Administración de Jugadores.
- Administración de Equipos.

- Administración de datos generales de Partido
- Ingreso de datos de los partidos (Estadísticas por jugador).
- Elaboración de reportes estadísticos.

El sistema realiza los siguientes procesos de forma automática (sin el conocimiento del usuario):

- Ingreso y eliminación de perfiles de usuario.
- Obtención del perfil de los usuarios al momento de ingresar al sistema.
- Aplicación de las fórmulas para estadísticas.
- Obtención automática de estadísticas.

Requisitos Específicos

En esta sección de la ERS se muestran todos los requisitos de Basys, con un nivel de detalle suficiente que permite a los Ingenieros de Software, diseñar el sistema de tal forma que cumpla con todos los requerimientos de la FEFDER, utilizando a Midas como metodología de desarrollo; además de probar a los Auditores de software que el sistema satisface esos requisitos.

También es importante la comprensión de esta sección por parte de los responsables de la FEFDER (usuarios finales), para que puedan determinar si el sistema será diseñado de acuerdo a sus necesidades.

Interfaces Externas

Basys es un sistema nuevo, autónomo, completamente independiente de otros, por lo que no existen interfaces con otros sistemas.

Requisitos funcionales

A continuación se detallan las funciones de Basys que van a ser implementadas para que cumplan con todos los requisitos especificados por la Facultad de Educación Física, Deportes y Recreación de la ESPE.

i. Funciones del Módulo de Seguridad de acceso al sistema.

1) Consultar todos los usuarios existentes

- Introducción: Despliega todos los usuarios que se han registrado en el sistema, en la Vista Web de la pantalla Usuarios.
- Nombre de la Función: *ConsultarTodosUsuarios()*
- Entradas: No se contempla ninguna entrada.
- Proceso: Selecciona los campos código, nombre, alias y mail de todos los registros de la tabla Usuario. Los ordena por nombre.
- Salida: Devuelve los registros obtenidos de la tabla Usuario.

2) Consultar un usuario

- Introducción: Despliega todos los datos de un usuario, dado su código, en el Formulario Web de la pantalla Usuarios.
- Nombre de la Función: *ConsultarUsuario()*
- Entradas: Código del usuario.
- Proceso: Trae de la base de datos, todos los datos de un usuario, dado su código.
- Salida: Devuelve el registro obtenido.

3) Ingresar un usuario nuevo

- Introducción: Añade un nuevo usuario a la base de datos, con su respectivo perfil de acceso al sistema.

- Nombre de la Función: *IngresarUsuario()*
- Entradas: Todos los datos del usuario a ser ingresados.
- Proceso: Valida que el alias entregado no se repita con otro ya existente. Valida el número de cédula del usuario. Inserta los datos consignados en la base. Inserta el perfil del usuario denegándole el acceso a todos los módulos del sistema.
- Salida: Éxito o fracaso en la inserción.

4) Modificar un usuario existente

- Introducción: Modifica uno o más datos de un usuario existente.
- Nombre de la Función: *ModificarUsuario()*
- Entradas: Datos del usuario a ser modificados.
- Proceso: Valida que si se desea modificar el alias de usuario, este no se repita con otro ya existente. Valida el número de cédula del usuario, de ser requerida su modificación. Modifica los datos del usuario.
- Salida: Éxito o fracaso en la modificación.

5) Eliminar un usuario

- Introducción: Elimina físicamente de la base de datos, el registro de un usuario, dado su código.
- Nombre de la Función: *EliminarUsuario()*
- Entradas: Código del usuario.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos primero el perfil del usuario y luego el registro del usuario que coincida con el código dado.
- Salida: Éxito o fracaso en la eliminación.

6) Cambiar la contraseña del usuario

- Introducción: El usuario modifica su contraseña de acceso al sistema. Es recomendable realizar una modificación de contraseña de forma periódica.
- Nombre de la Función: *CambiarContraseña()*
- Entradas: El código del usuario y la nueva contraseña de acceso.
- Proceso: Pregunta por la contraseña anterior, pide la nueva contraseña dos veces para evitar los errores en la digitación de los datos. Modifica la contraseña anterior por la nueva contraseña.
- Salida: Éxito o fracaso en el cambio de contraseña.

7) Ingresar el perfil de un usuario

- Introducción: Añade un nuevo perfil de usuario a la base de datos, se realiza cada vez que se añade un nuevo usuario. En la primera vez se deniega el permiso de acceso a todos los módulos. Posteriormente se modificará esto dándole el acceso a los módulos que necesite dicho usuario.
- Nombre de la Función: *IngresarPerfil()*
- Entradas: Código del usuario y sus permisos de acceso a los módulos del sistema.
- Proceso: Inserta el perfil para el usuario nuevo que se ha ingresado.
- Salida: Éxito o fracaso en la inserción.

8) Consultar el perfil de un usuario

- Introducción: Despliega todos los datos del perfil de acceso al sistema de un usuario, dado su código, en el Formulario Web de Modificación de Perfil de Usuarios.

- Nombre de la Función: *ConsultarPerfil()*
- Entradas: Código del usuario.
- Proceso: Trae de la base de datos, todos los datos del perfil de acceso al sistema de un usuario, dado su código.
- Salida: Devuelve el registro obtenido.

9) Modificar el perfil de un usuario

- Introducción: Modifica los permisos de acceso de un usuario al sistema.
- Nombre de la Función: *ModificarPerfil()*
- Entradas: Código del usuario y perfiles de acceso a ser modificados.
- Proceso: Modifica los permisos de acceso del usuario que sean requeridos, denegándole o permitiéndole el acceso a uno u otro módulo del sistema según sea el caso.
- Salida: Éxito o fracaso en la modificación.

10) Eliminar el perfil de un usuario

- Introducción: Elimina físicamente de la base de datos, el perfil de un usuario, dado su código.
- Nombre de la Función: *EliminarPerfil()*
- Entradas: Código del usuario.
- Proceso: Elimina de la base de datos el registro de perfil del usuario, que coincida con el código dado.
- Salida: Éxito o fracaso en la eliminación.

11) Validar el dígito auto verificador de la cédula del usuario

- Introducción: Verifica que una cédula dada sea correcta, utilizando el algoritmo provisto por el Registro Civil.

- Nombre de la Función: *ValidarCedula()*
- Entradas: Cédula a ser validada.
- Proceso: Verifica que el décimo dígito de la cédula (dígito auto verificador), sea correcto utilizando los 9 dígitos anteriores, comprobando si una cédula dada es o no correcta.
- Salida: Verdadero si la cédula es correcta y Falso si no lo es.

12) Validar el ingreso de un usuario al sistema

- Introducción: Valida que el Nombre de usuario (alias) y la contraseña ingresada, sean de un usuario registrado del sistema, permitiéndole el ingreso al mismo.
- Nombre de la Función: *ValidarIngreso()*
- Entradas: Alias y Contraseña del usuario.
- Proceso: Verifica el alias y la contraseña provistos, en la base de datos, si existen y son correctos, es decir pertenecen a un usuario del sistema, busca el código del usuario para posteriormente obtener su perfil de acceso al sistema.
- Salida: Si son correctos los datos, devuelve el código del usuario, si no lo son, devuelve fracaso en la ejecución de la función.

13) Validar el nombre de los usuarios

- Introducción: Valida que cuando se ingrese o modifique un nuevo nombre de usuario (alias), este no se repita con otro ya existente y perteneciente a otro usuario, es decir, el alias es único para los usuarios.
- Nombre de la Función: *ValidarNombre()*
- Entradas: El alias del usuario a validar.

- Proceso: Verifica en la base de datos el alias provisto, buscando si existe y pertenece a otro usuario del sistema, o si es nuevo (si no existe).
- Salida: Éxito cuando no existe, fracaso si el alias ya lo tiene otro usuario.

14) Validar el perfil de acceso al sistema de un usuario

- Introducción: Valida el perfil de acceso al sistema de un usuario, dado su código, es decir los módulos a los que tiene acceso el usuario.
- Nombre de la Función: *ValidarPerfil()*
- Entradas: Código del usuario.
- Proceso: Trae de la base de datos, todos los campos de perfil (módulos) a los cuales tiene acceso un usuario, dado su código.
- Salida: Devuelve el registro obtenido.

ii. Funciones del Módulo de Campeonatos.

15) Consultar todos los campeonatos existentes

- Introducción: Despliega todos los campeonatos que se han registrado en el sistema, en la Vista Web de la pantalla Campeonato.
- Nombre de la Función: *ConsultarTodosCampeonatos ()*
- Entradas: Nombre del campeonato.
- Proceso: Selecciona todos los campos de los registros de la tabla Campeonato que coincidan con el parámetro nombre del campeonato. Los ordena por nombre.

- Salida: Devuelve los registros obtenidos de la tabla Campeonato.

16)Consultar un campeonato

- Introducción: Despliega todos los datos de un campeonato, dado su código, en el Formulario Web de la pantalla Campeonato.
- Nombre de la Función: *ConsultarCampeonato()*
- Entradas: Código del campeonato.
- Proceso: Trae de la base de datos, todos los datos de un campeonato, dado su código.
- Salida: Devuelve el registro obtenido.

17)Ingresar un campeonato nuevo

- Introducción: Añade un nuevo campeonato a la base de datos.
- Nombre de la Función: *IngresarCampeonato()*
- Entradas: Todos los datos del campeonato a ser ingresados.
- Proceso: Valida que el nombre del campeonato no se repita con otro ya existente. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

18)Modificar un campeonato existente

- Introducción: Modifica uno o más datos de un campeonato existente.
- Nombre de la Función: *ModificarCampeonato()*
- Entradas: Datos del campeonato a ser modificados.
- Proceso: Valida que si se desea modificar el nombre del campeonato, este no se repita con otro ya existente. Modifica los datos del campeonato.
- Salida: Éxito o fracaso en la modificación.

19) Eliminar un campeonato

- Introducción: Elimina físicamente de la base de datos, el registro de un campeonato, dado su código.
- Nombre de la Función: *EliminarCampeonato()*
- Entradas: Código del campeonato.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del campeonato que coincida con el código dado.
- Salida: Éxito o fracaso en la eliminación.

20) Validar el nombre de los campeonatos

- Introducción: Valida que cuando se ingrese o modifique el nombre de un campeonato, este no se repita con otro ya existente y perteneciente a otro campeonato, es decir, el nombre es único para cada campeonato.
- Nombre de la Función: *ValidarNombre()*
- Entradas: El nombre del campeonato a validar.
- Proceso: Verifica en la base de datos el nombre provisto, buscando si existe y pertenece a otro campeonato, o si es nuevo (si no existe).
- Salida: Éxito cuando no existe, fracaso si el nombre ya lo tiene otro campeonato.

iii. Funciones del Módulo de Equipos.

21) Consultar todos los equipos existentes

- Introducción: Despliega todos los equipos que se han registrado en el sistema, en la Vista Web de la pantalla Equipo.
- Nombre de la Función: *ConsultarTodosEquipos ()*

- Entradas: Nombre del equipo.
- Proceso: Selecciona todos los campos de los registros de la tabla Equipo que coincidan con el parámetro nombre del equipo. Los ordena por nombre.
- Salida: Devuelve los registros obtenidos de la tabla Equipo.

22)Consultar un equipo

- Introducción: Despliega todos los datos de un equipo, dado su código, en el Formulario Web de la pantalla Equipo.
- Nombre de la Función: *ConsultarEquipo()*
- Entradas: Código del equipo.
- Proceso: Trae de la base de datos, todos los datos de un equipo, dado su código.
- Salida: Devuelve el registro obtenido.

23)Ingresar un equipo nuevo

- Introducción: Añade un nuevo equipo a la base de datos.
- Nombre de la Función: *IngresarEquipo()*
- Entradas: Todos los datos del equipo a ser ingresados.
- Proceso: Valida que el nombre del equipo no se repita con otro ya existente. Inserta los datos consignados en la base.
- Salida: Éxito o fracaso en la inserción.

24)Modificar un equipo existente

- Introducción: Modifica uno o más datos de un equipo existente.
- Nombre de la Función: *ModificarEquipo()*
- Entradas: Datos del equipo a ser modificados.

- Proceso: Valida que si se desea modificar el nombre del equipo, este no se repita con otro ya existente. Modifica los datos del equipo.
- Salida: Éxito o fracaso en la modificación.

25) Eliminar un equipo

- Introducción: Elimina físicamente de la base de datos, el registro de un equipo, dado su código.
- Nombre de la Función: *EliminarEquipo()*
- Entradas: Código del equipo.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del equipo que coincida con el código dado.
- Salida: Éxito o fracaso en la eliminación.

26) Validar el nombre de los equipos

- Introducción: Valida que cuando se ingrese o modifique el nombre de un equipo, este no se repita con otro ya existente y perteneciente a otro equipo, es decir, el nombre es único para cada equipo.
- Nombre de la Función: *ValidarNombre()*
- Entradas: El nombre del equipo a validar.
- Proceso: Verifica en la base de datos el nombre provisto, buscando si existe y pertenece a otro equipo, o si es nuevo (si no existe).
- Salida: Éxito cuando no existe, fracaso si el nombre ya lo tiene otro equipo.

iv. **Funciones del Módulo de Jugadores.**

27) Consultar todos los jugadores existentes

- Introducción: Despliega todos los jugadores que se han registrado en el sistema, en la Vista Web de la pantalla Jugador.
- Nombre de la Función: *ConsultarTodosJugadores ()*
- Entradas: Nombre del jugador.
- Proceso: Selecciona todos los campos de los registros de la tabla Jugador que coincidan con el parámetro nombre del jugador. Los ordena por nombre.
- Salida: Devuelve los registros obtenidos de la tabla Jugador.

28) Consultar un jugador

- Introducción: Despliega todos los datos de un jugador, dado su código, en el Formulario Web de la pantalla Jugador.
- Nombre de la Función: *ConsultarJugador()*
- Entradas: Código del jugador.
- Proceso: Trae de la base de datos, todos los datos de un jugador, dado su código.
- Salida: Devuelve el registro obtenido.

29) Ingresar un jugador nuevo

- Introducción: Añade un nuevo jugador a la base de datos.
- Nombre de la Función: *IngresarJugador()*
- Entradas: Todos los datos del jugador a ser ingresados.
- Proceso: Valida que el nombre del jugador no se repita con otro ya existente. Valida el número de cédula del jugador. Inserta los datos consignados en la base.

- Salida: Éxito o fracaso en la inserción.

30) Modificar un jugador existente

- Introducción: Modifica uno o más datos de un jugador existente.
- Nombre de la Función: *ModificarJugador()*
- Entradas: Datos del jugador a ser modificados.
- Proceso: Valida que si se desea modificar el nombre del jugador, este no se repita con otro ya existente. Valida el número de cédula del jugador, de ser requerida su modificación. Modifica los datos del jugador.
- Salida: Éxito o fracaso en la modificación.

31) Eliminar un jugador

- Introducción: Elimina físicamente de la base de datos, el registro de un jugador, dado su código.
- Nombre de la Función: *EliminarJugador()*
- Entradas: Código del jugador.
- Proceso: Luego de confirmar la eliminación, elimina de la base de datos el registro del jugador que coincida con el código dado.
- Salida: Éxito o fracaso en la eliminación.

32) Validar el dígito auto verificador de la cédula del jugador

- Introducción: Verifica que una cédula dada sea correcta, utilizando el algoritmo provisto por el Registro Civil.
- Nombre de la Función: *ValidarCedula()*
- Entradas: Cédula a ser validada.

- Proceso: Verifica que el décimo dígito de la cédula (dígito auto verificador), sea correcto utilizando los 9 dígitos anteriores, comprobando si una cédula dada es o no correcta.
- Salida: Verdadero si la cédula es correcta y Falso si no lo es.

33) Validar el nombre de los jugadores

- Introducción: Valida que cuando se ingrese o modifique el nombre de un jugador, este no se repita con otro ya existente y perteneciente a otro jugador, es decir, el nombre debe ser único para cada jugador.
- Nombre de la Función: *ValidarNombre()*
- Entradas: El nombre del jugador a verificar.
- Proceso: Verifica en la base de datos el nombre provisto, buscando si existe y pertenece a otro jugador, o si es nuevo (si no existe).
- Salida: Éxito cuando no existe, fracaso si el nombre ya lo tiene otro jugador.

v. Funciones del Módulo de Estadísticas. Sub Módulo Partido

34) Consultar todos los partidos existentes

- Introducción: Despliega todos los partidos que se han registrado en el sistema, en la Vista Web de la pantalla Partido.
- Nombre de la Función: *ConsultarTodosPartidos()*
- Entradas: Nombre del partido.
- Proceso: Selecciona todos los campos de los registros de la tabla Partido que coincidan con el parámetro nombre del partido. Los ordena por nombre.

- Salida: Devuelve los registros obtenidos de la tabla Partido.

35) Consultar un partido

- Introducción: Despliega todos los datos de un partido, dado su código, en el Formulario Web de la pantalla Partido.
- Nombre de la Función: *ConsultarPartido()*
- Entradas: Código del partido.
- Proceso: Trae de la base de datos, todos los datos de un partido, dado su código.
- Salida: Devuelve el registro obtenido.

36) Ingresar un partido nuevo

- Introducción: Registra un nuevo partido en la base de datos.
- Nombre de la Función: *IngresarPartido()*
- Entradas: Todos los datos del partido a ser ingresados.
- Proceso: Inserta los datos consignados en la tabla Partido de la base de datos, luego añade dos registros (1 para cada equipo que pertenece al partido), para el ingreso de las estadísticas de dichos equipos.
- Salida: Éxito o fracaso en la inserción.

37) Modificar un partido existente

- Introducción: Modifica uno o más datos de un partido existente.
- Nombre de la Función: *ModificarPartido()*
- Entradas: Datos del partido a ser modificados.
- Proceso: Modifica los datos del partido. Valida que si se desea modificar uno de los equipos que juegan el partido, se realice la modificación también para las estadísticas de dichos equipos.

- Salida: Éxito o fracaso en la modificación.

38) Consultar los equipos del partido

- Introducción: Despliega los datos de los dos equipos que intervienen en un partido, dado el código del partido, en el Formulario Web de la pantalla Partido.
- Nombre de la Función: *ConsultarEquipoPartido()*
- Entradas: Código del partido.
- Proceso: Trae de la base de datos, los registros pertenecientes a los equipos que juegan un partido, dado el código del partido.
- Salida: Devuelve los registros obtenidos.

39) Ingresar un equipo nuevo del partido.

- Introducción: Cuando se ingresa un partido nuevo, se crea un registro para el ingreso de estadísticas (cabecera), para cada equipo que interviene en el partido, esta función crea ese registro en la base de datos.
- Nombre de la Función: *IngresarEstadísticas()*
- Entradas: Código del partido y código del equipo.
- Proceso: Inserta los datos consignados en la base de datos.
- Salida: Éxito o fracaso en la inserción.

40) Modificar un equipo existente, del partido

- Introducción: Cuando se modifica un partido existente y se modifica uno o los dos equipos que participan en el mismo, se modifica también el registro para el ingreso de estadísticas, para cada equipo modificado, esta función modifica ese registro.
- Nombre de la Función: *ModificarEstadísticas()*

- Entradas: Código del equipo a ser modificado.
- Proceso: Modifica los datos del equipo, si este se ha modificado.
- Salida: Éxito o fracaso en la modificación.

vi. Funciones del Módulo de Estadísticas. Sub Módulo Información del Partido

41) Iniciar un partido nuevo

- Introducción: Inicializa con cero todos los datos del equipo que van a ser ingresados una vez que empiece el partido.
- Nombre de la Función: *IniciarPartido()*
- Entradas: Código de estadísticas del equipo.
- Proceso: Ingresa con 0 en la base de datos el detalle de las estadísticas que se van a tomar para un equipo en el partido.
- Salida: Éxito o fracaso en la inserción.

42) Iniciar un cuarto de tiempo (periodo)

- Introducción: Un partido de baloncesto se juega en cuatro cuartos reglamentarios, esta función se encarga de actualizar el cuarto (1ro., 2do, 3ro. o 4to. periodo) que se está jugando.
- Nombre de la Función: *IniciarCuarto()*
- Entradas: Periodo actual.
- Proceso: Actualiza en la base de datos, el periodo que está jugando un equipo en el partido.
- Salida: Éxito o fracaso en la actualización.

43) Iniciar el periodo de estadísticas de un jugador en el partido

- Introducción: Inicializa con cero todos los datos de información del partido de los jugadores, que van a disputar el partido. Estos datos van a ser ingresados una vez que empiece el partido y en el transcurso del mismo
- Nombre de la Función: *IniciarPeriodoJugador()*
- Entradas: Datos del jugador a inicializar.
- Proceso: Ingresa con 0 en la base de datos el detalle de las estadísticas (información del partido) que se van a tomar para un jugador, en el partido.
- Salida: Éxito o fracaso en la inserción.

44) Cargar los datos de un partido existente

- Introducción: Despliega todos los datos estadísticos (información del partido) de un jugador, cuando ya ha sido registrado en un partido, dado su código, para modificarlos en el Formulario Web de la pantalla Información del Partido.
- Nombre de la Función: *CargarPartido()*
- Entradas: Código de estadísticas del jugador.
- Proceso: Trae de la base de datos, todos los datos de información del partido de un jugador, dado su código de estadísticas.
- Salida: Devuelve el registro obtenido.

45) Obtener el marcador de un partido

- Introducción: Despliega en los casilleros de marcador de la pantalla de Información del Partido, el puntaje total de los dos equipos que intervienen en un partido, dado el código del partido

- Nombre de la Función: *MarcadorPartido()*
- Entradas: Código del partido.
- Proceso: Trae de la base de datos, el puntaje total de los equipos que juegan un partido, dado el código del partido.
- Salida: Devuelve los registros obtenidos.

46)Cambiar a un jugador por otro jugador

- Introducción: El baloncesto se juega en equipos de 5 jugadores. Los datos de los 5 jugadores que está en cancha va a mostrarse en la pantalla de Información del partido, cuando se realice un cambio de jugadores, esta función será encargada de proceder a la modificación de dichos jugadores en el sistema.
- Nombre de la Función: *CambiarJugador()*
- Entradas: Datos del jugador que entra al campo de juego.
- Proceso: Modifica el estado del jugador (esta jugando o no), luego trae de la base de datos todos los datos de información del partido del jugador que entra al campo de juego.
- Salida: Devuelve el registro obtenido.

47)Actualizar una asistencia de balón en jugador y equipo

- Introducción: Suma o resta el número de asistencias de balón por jugador y equipo.
- Nombre de la Función: *ActualizarAsistencia ()*
- Entradas: Nuevo número de asistencias, código del jugador, código del equipo, y código de la estadística del equipo.

- Proceso: Modifica el campo Asistencia, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Asistencia, del equipo.
- Salida: Éxito o fracaso en la actualización.

48) Actualizar una falta cometida en jugador y equipo

- Introducción: Suma o resta el número de faltas cometidas por jugador y equipo.
- Nombre de la Función: *ActualizarFaltaCometida ()*
- Entradas: Nuevo número de faltas cometidas, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Faltas Cometidas, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Faltas Cometidas, del equipo.
- Salida: Éxito o fracaso en la actualización.

49) Actualizar una falta recibida en jugador y equipo

- Introducción: Suma o resta el número de faltas recibidas por jugador y equipo.
- Nombre de la Función: *ActualizarFaltaRecibida ()*
- Entradas: Nuevo número de faltas recibidas, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Faltas Recibidas, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Faltas Recibidas, del equipo.
- Salida: Éxito o fracaso en la actualización.

50)Actualizar lanzamiento de corta distancia convertido, en jugador y equipo

- Introducción: Suma o resta el número de lanzamientos de 2 puntos anotados, por jugador y equipo.
- Nombre de la Función: *ActualizarLanzamientoCortaConvertido ()*
- Entradas: Nuevo número de lanzamientos de 2 puntos anotados, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Lanzamiento de corta distancia anotado, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Lanzamiento de corta distancia anotado, del equipo.
- Salida: Éxito o fracaso en la actualización.

51)Actualizar lanzamiento de corta distancia fallado, en jugador y equipo

- Introducción: Suma o resta el número de lanzamientos de 2 puntos fallados, por jugador y equipo.
- Nombre de la Función: *ActualizarLanzamientoCortaFallado ()*
- Entradas: Nuevo número de lanzamientos de 2 puntos fallados, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Lanzamiento de corta distancia fallado, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Lanzamiento de corta distancia fallado, del equipo.

- Salida: Éxito o fracaso en la actualización.

52)Actualizar lanzamiento de larga distancia convertido, en jugador y equipo

- Introducción: Suma o resta el número de lanzamientos de 3 puntos anotados, por jugador y equipo.
- Nombre de la Función: *ActualizarLanzamientoLargaConvertido ()*
- Entradas: Nuevo número de lanzamientos de 3 puntos anotados, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Lanzamiento de larga distancia anotado, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Lanzamiento de larga distancia anotado, del equipo.
- Salida: Éxito o fracaso en la actualización.

53)Actualizar lanzamiento de larga distancia fallado, en jugador y equipo

- Introducción: Suma o resta el número de lanzamientos de 3 puntos fallados, por jugador y equipo.
- Nombre de la Función: *ActualizarLanzamientoLargaFallado ()*
- Entradas: Nuevo número de lanzamientos de 3 puntos fallados, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Lanzamiento de larga distancia fallado, de Información del partido del jugador, con el nuevo valor, luego

modifica de la misma manera el campo Lanzamiento de larga distancia fallado, del equipo.

- Salida: Éxito o fracaso en la actualización.

54)Actualizar una pérdida de balón en jugador y equipo

- Introducción: Suma o resta el número de pérdidas de balón por jugador y equipo.
- Nombre de la Función: *ActualizarPerdidaDeBalon ()*
- Entradas: Nuevo número de pérdidas de balón, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Pérdidas de balón, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Pérdidas de balón, del equipo.
- Salida: Éxito o fracaso en la actualización.

55)Actualizar puntos anotados en jugador y equipo

- Introducción: Suma o resta el número de puntos anotados por jugador y equipo.
- Nombre de la Función: *ActualizarPuntos ()*
- Entradas: Nuevo número de puntos anotados, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Puntos, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo puntos, del equipo.
- Salida: Éxito o fracaso en la actualización.

56) Actualizar rebote defensivo en jugador y equipo

- Introducción: Suma o resta el número de rebotes defensivos por jugador y equipo.
- Nombre de la Función: *ActualizarReboteDefensivo ()*
- Entradas: Nuevo número de rebotes defensivos, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Rebotes defensivos, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Rebotes defensivos, del equipo.
- Salida: Éxito o fracaso en la actualización.

57) Actualizar rebote ofensivo en jugador y equipo

- Introducción: Suma o resta el número de rebotes ofensivos por jugador y equipo.
- Nombre de la Función: *ActualizarReboteOfensivo ()*
- Entradas: Nuevo número de rebotes ofensivos, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Rebotes ofensivos, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Rebotes ofensivos, del equipo.
- Salida: Éxito o fracaso en la actualización.

58) Actualizar tiro libre convertido en jugador y equipo

- Introducción: Suma o resta el número de tiros libres convertidos por jugador y equipo.
- Nombre de la Función: *ActualizarTiroLibreConvertido ()*

- Entradas: Nuevo número de tiros libres convertidos, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Tiros libres convertidos, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Tiros libres convertidos, del equipo.
- Salida: Éxito o fracaso en la actualización.

59) Actualizar tiro libre fallado en jugador y equipo

- Introducción: Suma o resta el número de tiros libres fallados por jugador y equipo.
- Nombre de la Función: *ActualizarTiroLibreFallado ()*
- Entradas: Nuevo número de tiros libres fallados, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Tiros libres fallados, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Tiros libres fallados, del equipo.
- Salida: Éxito o fracaso en la actualización.

60) Actualizar violación cometida en jugador y equipo

- Introducción: Suma o resta el número de violaciones cometidas por jugador y equipo.
- Nombre de la Función: *ActualizarViolacion ()*
- Entradas: Nuevo número de violaciones cometidas, código del jugador, código del equipo, y código de la estadística del equipo.
- Proceso: Modifica el campo Violaciones cometidas, de Información del partido del jugador, con el nuevo valor, luego modifica de la misma manera el campo Violaciones cometidas, del equipo.

- Salida: Éxito o fracaso en la actualización.

vii. Funciones del Módulo de Estadísticas. Sub Módulo Estadísticas

61) Iniciar las estadísticas de un equipo

- Introducción: Inicializa con cero todos los datos estadísticos del equipo que van a ser calculados mediante las fórmulas de efectividad una vez concluido el partido.
- Nombre de la Función: *IniciarEstadisticasEquipo()*
- Entradas: Código de estadísticas del equipo.
- Proceso: Ingresa con 0 en la base de datos un registro que será destinado para guardar el cálculo de las estadísticas.
- Salida: Éxito o fracaso en la inserción.

62) Iniciar las estadísticas de un jugador

- Introducción: Inicializa con cero todos los datos estadísticos de los jugadores, los mismos que van a ser calculados mediante las fórmulas de efectividad una vez concluido el partido.
- Nombre de la Función: *IniciarEstadisticasJugador()*
- Entradas: Código de estadísticas del equipo, código del jugador.
- Proceso: Ingresa con 0 en la base de datos un registro que será destinado para guardar el cálculo de las estadísticas del jugador.
- Salida: Éxito o fracaso en la inserción.

63) Consultar un campo de estadísticas del equipo

- Introducción: Devuelve un campo cualquiera de la tabla donde se guardan las estadísticas del equipo que se van a calcular mediante las fórmulas de efectividad.

- Nombre de la Función: *ConsultarCampoEEQ ()*
- Entradas: Nombre del campo, Código de estadísticas del equipo.
- Proceso: Trae de la base de datos el valor del campo solicitado de las estadísticas del equipo, dado su código de estadísticas.
- Salida: Devuelve el valor obtenido.

64) Consultar un campo de estadísticas del jugador

- Introducción: Devuelve un campo cualquiera de la tabla donde se guardan las estadísticas del jugador que se van a calcular mediante las formulas de efectividad.
- Nombre de la Función: *ConsultarCampoEJU ()*
- Entradas: Nombre del campo, Código de estadísticas del jugador.
- Proceso: Trae de la base de datos el valor del campo solicitado de las estadísticas del jugador, dado su código de estadísticas.
- Salida: Devuelve el valor obtenido.

65) Consultar un campo de periodo

- Introducción: Devuelve un campo cualquiera de la tabla donde se guarda la Información del partido del jugador.
- Nombre de la Función: *ConsultarCampoPeriodo ()*
- Entradas: Nombre del campo, código de estadísticas del jugador, código del jugador.
- Proceso: Trae de la base de datos el valor del campo solicitado de Información del partido del jugador, dado su código.
- Salida: Devuelve el valor obtenido.

66) Consultar un campo de resumen

- Introducción: Devuelve un campo cualquiera de la tabla donde se guarda la Información del partido del equipo.
- Nombre de la Función: *ConsultarCampoResumen ()*
- Entradas: Nombre del campo, código de estadísticas del equipo.
- Proceso: Trae de la base de datos el valor del campo solicitado de Información del partido del equipo, dado su código de estadísticas.
- Salida: Devuelve el valor obtenido.

67) Calcular las estadísticas de un equipo

- Introducción: Esta función llama a las funciones para el cálculo de las estadísticas del equipo mediante las fórmulas de efectividad.
- Nombre de la Función: *EEQ ()*
- Entradas: Código de estadísticas del equipo.
- Proceso: Realiza las llamadas a las funciones: Calcular la efectividad, Calcular los tiros de campo, Calcular los tiros libres, Calcular el control de balón, Calcular las faltas, Calcular los rebotes defensivos, Calcular los rebotes ofensivos, Calcular la posesión del balón y Normalizar los datos.
- Salida: Mensaje de éxito o fracaso en la ejecución de las funciones.

68) Calcular las estadísticas de un jugador

- Introducción: Esta función llama a las funciones para el cálculo de las estadísticas del jugador mediante las fórmulas de efectividad.
- Nombre de la Función: *EJU ()*
- Entradas: Código de estadísticas del jugador, código del jugador.

- Proceso: Realiza las llamadas a las funciones: Calcular la efectividad, Calcular los tiros de campo, Calcular los tiros libres, Calcular el control de balón, Calcular las faltas, Calcular los rebotes defensivos, Calcular los rebotes ofensivos, Calcular la posesión del balón y Normalizar los datos de las estadísticas del jugador.
- Salida: Mensaje de éxito o fracaso en la ejecución de las funciones.

69)Calcular la efectividad del equipo y jugadores

- Introducción: Calcula la efectividad del equipo y de los jugadores en un partido de baloncesto aplicando la fórmula del Tendex.
- Nombre de la Función: *CalcularEfectividad ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula del Tendex para efectividad. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

70)Calcular el control de balón del equipo y jugadores

- Introducción: Calcula el control de balón del equipo y de los jugadores en un partido de baloncesto.
- Nombre de la Función: *CalcularControlBalon ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula para

calcular el Control del Balón. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.

- Salida: Éxito o fracaso en el cálculo.

71)Calcular las faltas del equipo y jugadores

- Introducción: Calcula las faltas del equipo y de los jugadores en un partido de baloncesto.
- Nombre de la Función: *CalcularFaltas ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula para calcular las Faltas. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

72)Calcular la posesión del balón del equipo y jugadores

- Introducción: Calcula la posesión de balón del equipo y de los jugadores en un partido de baloncesto.
- Nombre de la Función: *CalcularPosesionBalon ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula para calcular la Posesión de Balón. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

73)Calcular los rebotes defensivos del equipo y jugadores

- Introducción: Calcula el porcentaje de rebotes defensivos del equipo y de los jugadores en un partido de baloncesto.
- Nombre de la Función: *CalcularReboteDefensivo ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula para calcular los rebotes defensivos. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

74)Calcular los rebotes ofensivos del equipo y jugadores

- Introducción: Calcula el porcentaje de rebotes ofensivos del equipo y de los jugadores en un partido de baloncesto.
- Nombre de la Función: *CalcularReboteOfensivo ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula para calcular los rebotes ofensivos. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

75)Calcular los tiros de campo del equipo y jugadores

- Introducción: Calcula los tiros de campo del equipo y de los jugadores en un partido de baloncesto.

- Nombre de la Función: *CalcularTirosCampo ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula para calcular los tiros de campo. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

76)Calcular los tiros libres del equipo y jugadores

- Introducción: Calcula los tiros libres del equipo y de los jugadores en un partido de baloncesto.
- Nombre de la Función: *CalcularTirosLibre ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.
- Proceso: Consulta los datos que necesita de las tablas de Información del Partido, luego con esos datos aplica la fórmula para calcular los tiros libres. Posteriormente guarda el resultado obtenido en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

77)Normalizar los datos del equipo y jugadores

- Introducción: Normaliza los resultados obtenidos con el cálculo de las fórmulas de efectividad, a 100 posesiones de balón.
- Nombre de la Función: *CalcularNormalizar ()*
- Entradas: Código de estadísticas de equipo y jugador, código del jugador.

- Proceso: Consulta los datos que necesita para normalizar de las tablas de Estadísticas y de Información del Partido, luego normaliza a 100 posesiones de balón a los siguientes datos: tiros de campo, tiros libres, control de balón y faltas. Posteriormente guarda los resultados obtenidos en el registro de estadísticas de jugador y equipo.
- Salida: Éxito o fracaso en el cálculo.

Frecuencia de uso

El motor de base de datos a usarse es SQL Server 2005 Express Edition, las tablas llamadas de mantenimiento son:

- Usuario
- Perfil
- Equipo
- Campeonato
- Jugador

Sobre estas tablas se realizan las operaciones básicas de ingresos, consultas, modificaciones, búsquedas y eliminaciones de forma periódica, es decir se manejan transacciones en una cantidad soportada con holgura por el motor de base de datos.

Las demás tablas son llamadas transaccionales porque soportan mayor carga de operaciones, es decir se presentan accesos permanentes a la base de datos, los mismos que son controlados por grupos de conexiones para evitar cuellos de botella y no permitir la disminución del rendimiento del sistema.

Capacidad de acceso

SQL Server 2005 Express Edition, permite el manejo de 20 usuarios concurrentes por procesador, es decir BasyS tiene la capacidad de aceptar 20 usuarios conectados al mismo tiempo, pero un número indeterminado de usuarios que se conectan en tiempos diferentes, dependiendo de las necesidades del cliente.

Requisitos de Rendimiento

Basys es una aplicación Web, que deberá manejar por lo menos una terminal Web (Browser), pudiendo aumentar el número de terminales de acuerdo a las necesidades de la FEFDER hasta 20, que es el número soportado por la base de datos y el servidor Web.

El número simultáneo de usuarios conectados al sistema dependerá de la cantidad de terminales (host cliente) y de los perfiles de usuario, teniendo en cuenta los siguientes parámetros:

- Un solo usuario para el perfil Administrador para mantener la integridad de los datos y centralizar la administración del sistema.
- Los tiempos de respuesta deseados deben estar en el rango de 0.2 a 4 segundos dependiendo de la complejidad de los procesos.

Restricciones de Diseño

Este punto está dirigido principalmente al diseño de las funciones y de la base de datos tomando en cuenta los siguientes condicionamientos:

- Dentro del diseño de la base de datos, los atributos serán nombrados manteniendo las tres primeras letras de la tabla a la que pertenecen, seguido por el nombre del atributo, con su primera letra mayúscula.
- En el programa, los nombres de las variables empiezan con tres letras que denotan el tipo de dato del cual se declaran, seguido del nombre de la variable, con su primera letra en mayúscula, por ejemplo:
 - strVariable
 - intVariable
 - datVariable
 - lonVariable

- Para identificar a los controles, sus nombres comienzan con las tres primeras letras en minúsculas del control seguida del nombre, donde la primera letra es mayúscula, por ejemplo:
 - btnNombre (Botones de Control)
 - txtNombre (Cajas de Texto)
 - lstNombre (Listas desplegables)

Atributos del Sistema de Software

- *Fiabilidad:* La veracidad y verificación de los datos que ingresen al sistema será de absoluta responsabilidad de quienes digiten dicha información, el sistema tendrá rutinas de validación en todos los casos posibles. Basys debe permitir guardar datos de manera completa y correcta en la base de datos, además de ello debe ofrece seguridad y confidencialidad.
- *Mantenimiento:* En la medida que transcurre el tiempo, el sistema puede quedar obsoleto, sea esto por el avance de la tecnología, las modificaciones en las reglas del básquetbol, o las investigaciones en el campo de la evaluación a los jugadores de este deporte, por lo tanto es importante y conveniente mantener una política de actualizaciones periódica, para que no se tenga que rehacer por completo la parte de funcionalidad del sistema, a mediano plazo.
- *Seguridad:* El SIW posee diferentes perfiles de usuario, lo cual permite la implementación de un ambiente de seguridad y el reforzamiento de las restricciones en el acceso a la base de datos.
- *Portabilidad:* El sistema debe estar instalado en un servidor con tecnología Microsoft (Windows), pero al ser desarrollado con una estructura funcional basada en XML Web Services, los mismos pueden ser invocados desde cualquier plataforma que soporte esta tecnología.

Determinación de costos del sistema

Para determinar el costo de desarrollo de un sistema, se debe tomar en cuenta el valor de los recursos humanos, materiales y equipos que se han utilizado para desarrollar el sistema de información Web.

Al momento de definir el costo del proyecto se determinan los valores del equipo de trabajo, es decir de las personas que trabajan en el desarrollo, de los equipos utilizados y del valor de las licencias del software de desarrollo. Estos datos se observan en la *Tabla 4.2* y en la *Tabla 4.3*:

Tabla 4.2: Costos de desarrollo del SIW. Recursos Humanos

Etapas	Tiempo de duración (meses)	Número de personas	Costo mensual por persona	Costo Total
Análisis y levantamiento de requerimientos.	1	2	300	600
Diseño del sistema	3	2	300	1800
Construcción del sistema	4	2	300	2400
Pruebas del sistema	1	2	300	600
Total				5400

Tabla 4.3: Costos de hardware y software de desarrollo

Cant.	Equipos y Software	Costo unitario	Costo total
1	PC de escritorio (Pentium IV 1.4 GHz. RAM de 256 MB. 30 GB de disco. Tarjeta de red, monitor, mouse, teclado, CD-ROM).	650	650.0
1	Computador portátil (Pentium IV 1.4 GHz. RAM de 256 MB. 30 GB de disco. Tarjeta de red, monitor, mouse, teclado, CD-ROM).	1100	1100.0
1	Microsoft Visual Studio 2005 Standard Edition	382.50	382.5
1	Microsoft SQL Server 2005 Express Edition	0	0
1	Rational XDE Developer Evaluation Version	0	0
Total			2132.5

Costos de personal adicional

Tabla 4.4: Costos de personal adicional

Cargo	Tiempo (meses)	Número de personas	Costo mensual por persona	Costo Total
Experto en básquetbol. Entrenador de selecciones de la ESPE	1	1	250	250
Total				250

Costos generales

Tabla 4.5: Costos generales

Tipo	Tiempo (meses)	Costo mensual	Costo Total
Suministros, materiales y otros gastos	9	20	180
Conexión a Internet	8	19	152
Costos de teléfono por conexión a Internet	8	25	200
Total			532

Costo total del proyecto

Tabla 4.6: Resumen de costos del proyecto

Costos	Valor
Costo de desarrollo del SIW	5400.00
Costos de hardware y software de desarrollo	2132.50
Costos de personal adicional	250.00
Costo generales	532.00
Costo total de desarrollo del proyecto	8314.50

Aplicando la fórmula que se muestra a continuación, se determinará el valor estimado que debe tener cada copia de Basys:

$$V_c = \frac{(V_d + G_a) * (1 + 0.006)^{T_{vs}}}{T_{vs} * N_{ca}} + C_e$$

En donde:

- Vc: Valor mínimo por unidad del sistema.
- Vd: Valor de desarrollo del proyecto.
- Ga: Ganancia anual deseada.
- Tvs: Tiempo de vida del SIW.
- Nca: Número de unidades promedio anuales a venderse.
- Ce: Costo de elaboración de cada unidad.

Se toman en cuenta las siguientes consideraciones:

- Se estima que el SIW no producirá una ganancia anual, porque de momento no se prevén realizar actualizaciones.
- Se esperaría producir al menos 5 copias anuales.
- Tomando en cuenta el tiempo de vida promedio de un SIW, y la utilidad de Basys, se tiene como tiempo de vida 8 años.
- El costo de elaboración de cada unidad implica: impresión y encuadernación del manual de usuario y manual de instalación; grabación de CDs de instalación. Este costo asciende a \$7.00

Al reemplazar los valores obtenidos en la fórmula dada, se obtiene:

$$V_C = \frac{(8314,5 + 0) * (1 + 0,006)^8}{8 * 5} + 7$$

El valor mínimo por unidad que se desprende del cálculo es **\$ 214.88**

4.1.2- Diseño del Sistema

Modelos Independientes de Computación – CIM

Están conformados por dos modelos:

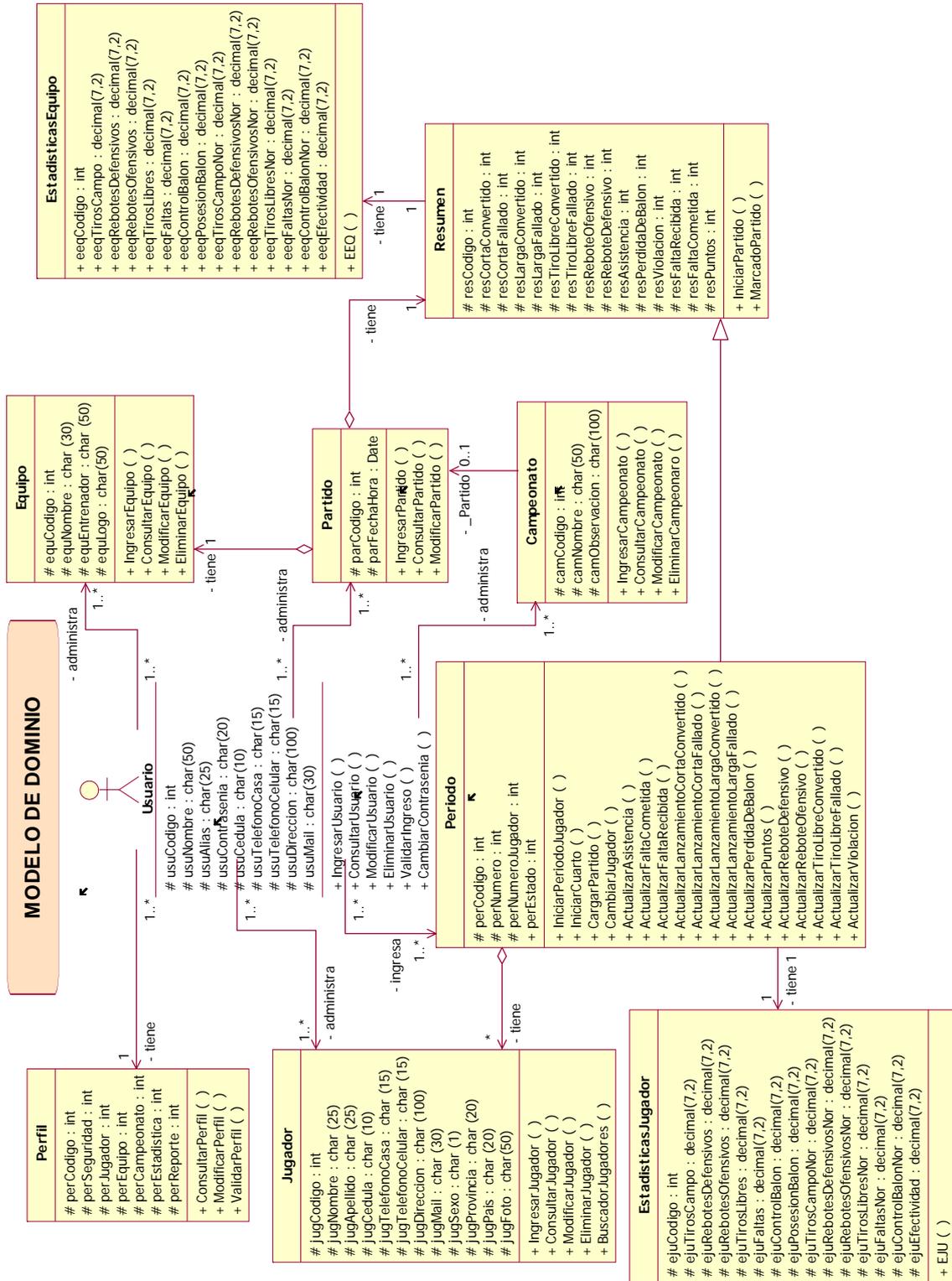


Figura 4.2: Modelo de Dominio

En el modelo de Negocio se muestran las funciones del sistema a nivel macro y su interacción con el Usuario del sistema.

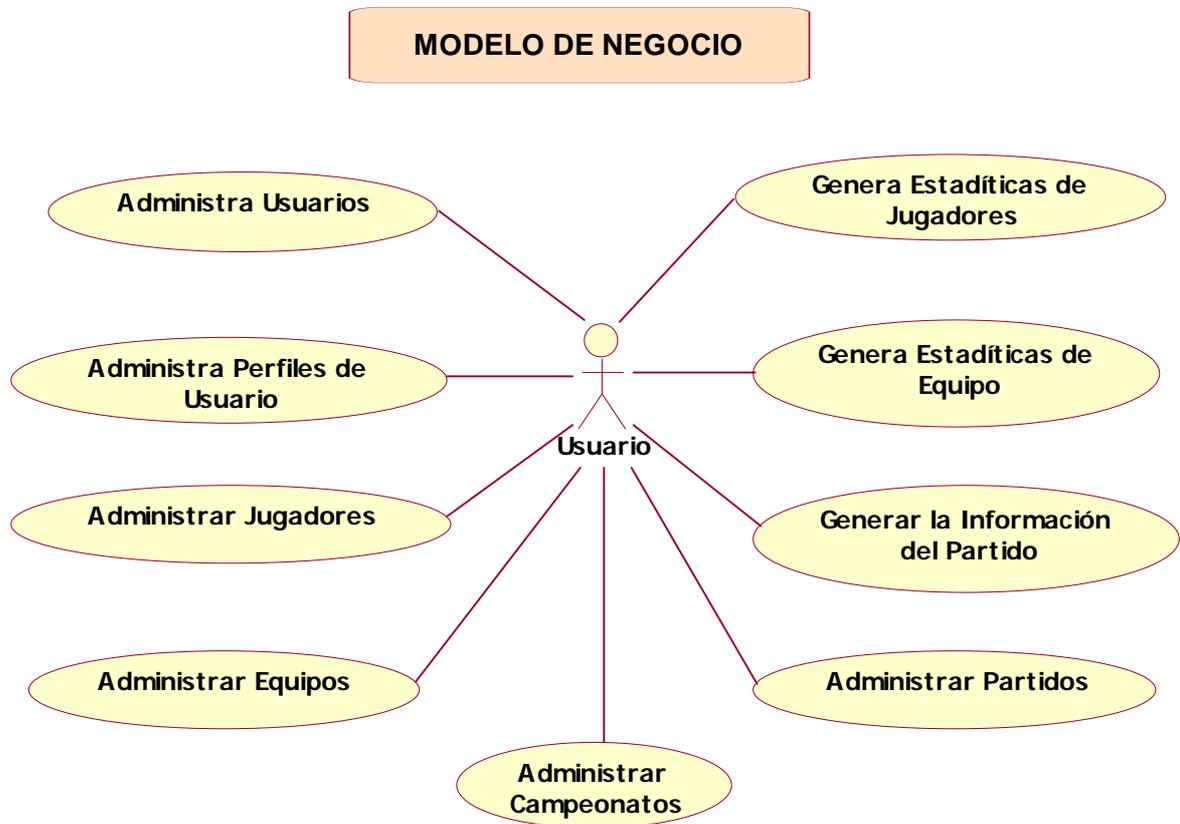
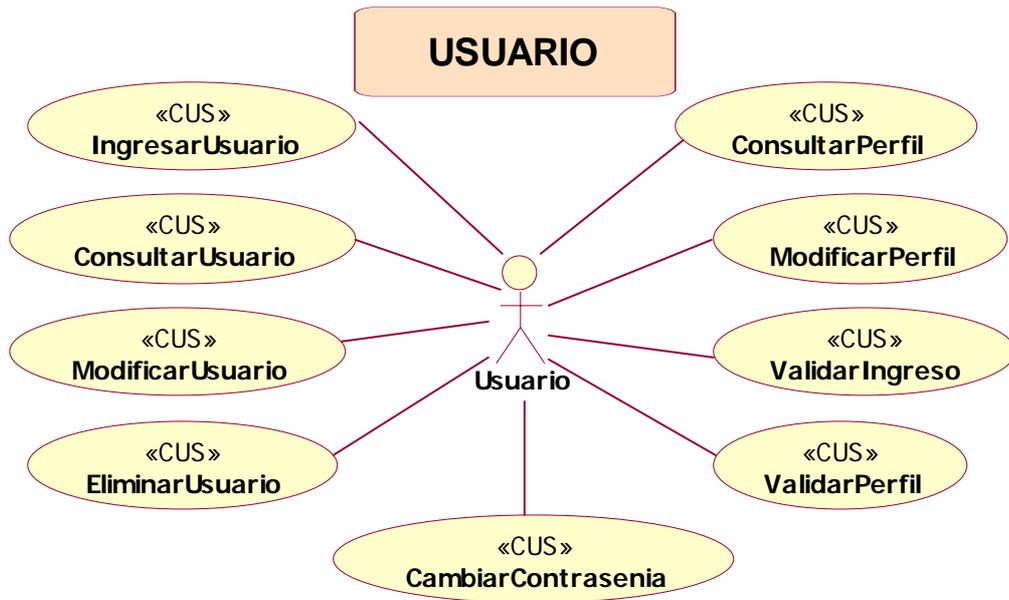


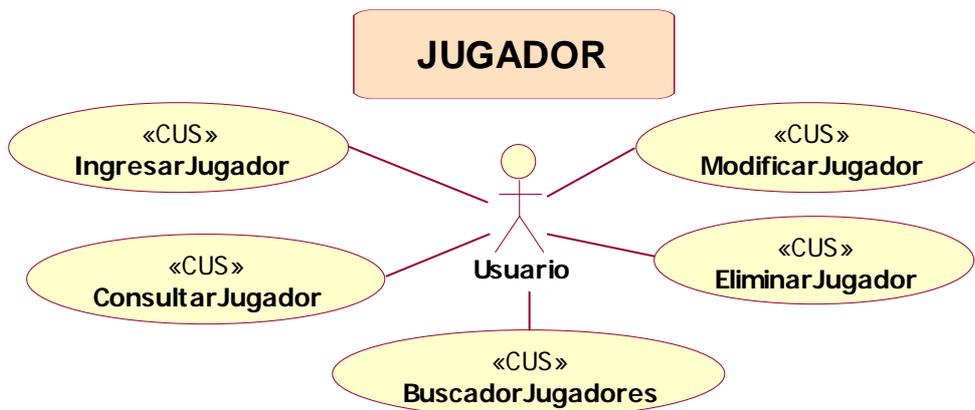
Figura 4.3: Modelo de Negocio

En el aspecto de funcionalidad se comienza con la elaboración del Modelo de Servicios de usuario que nos permite representar los servicios que el usuario requiere del sistema.

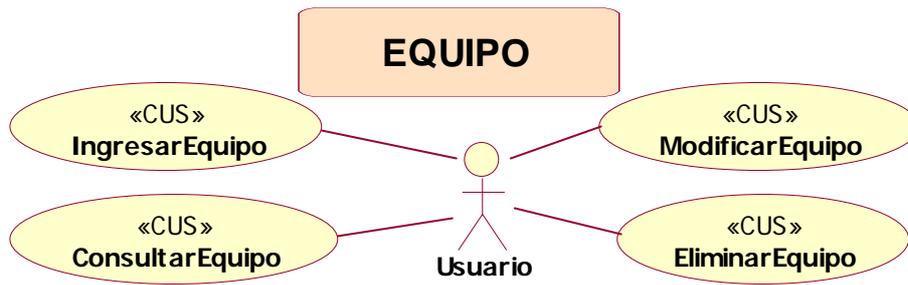
Modelo de Servicios de Usuario – Usuario



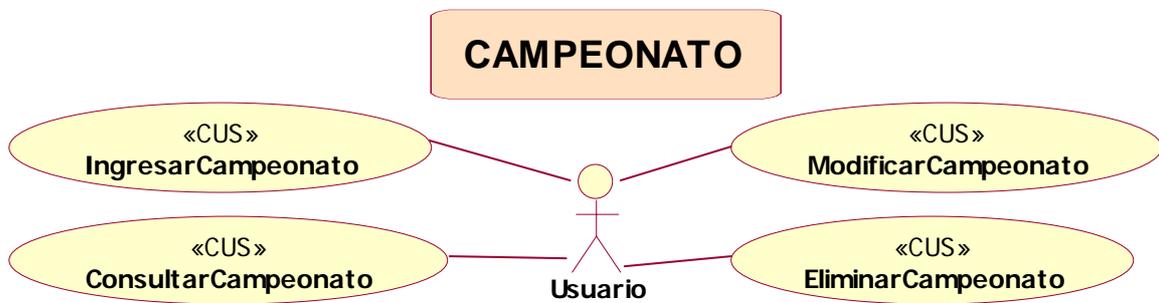
Modelo de Servicios de Usuario – Jugador



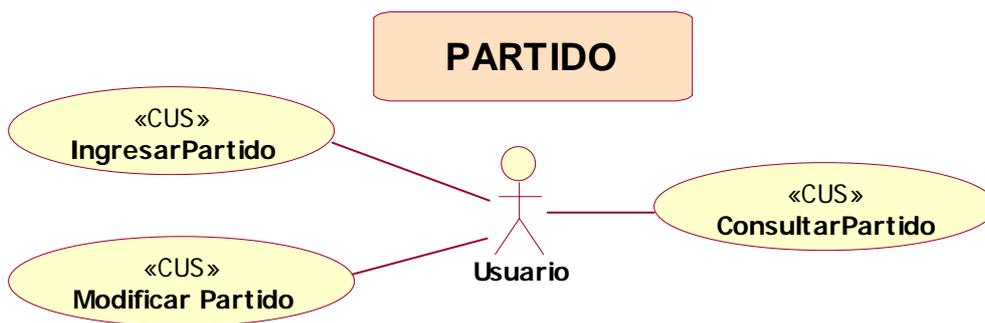
Modelo de Servicios de Usuario – Equipo



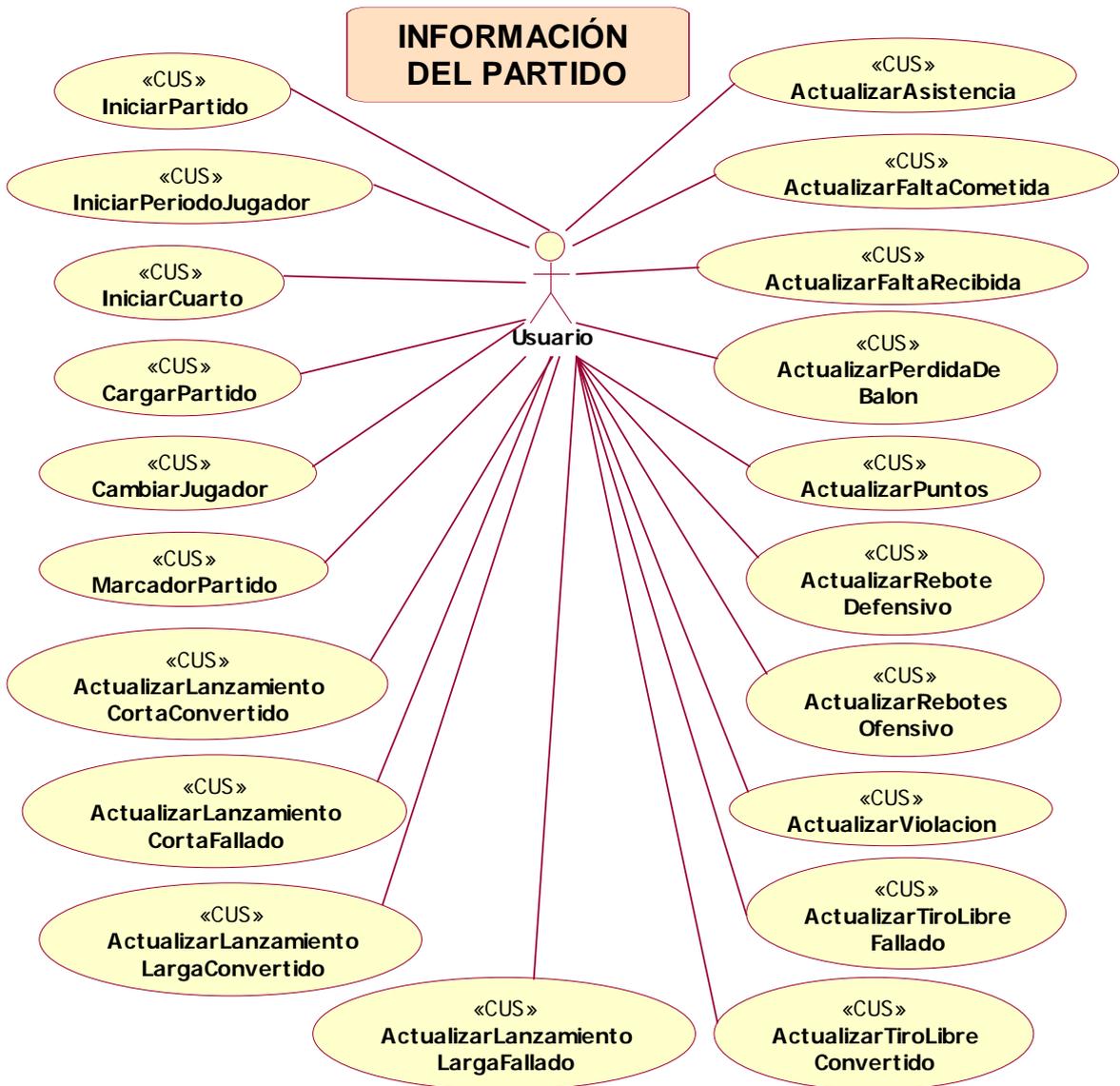
Modelo de Servicios de Usuario – Campeonato



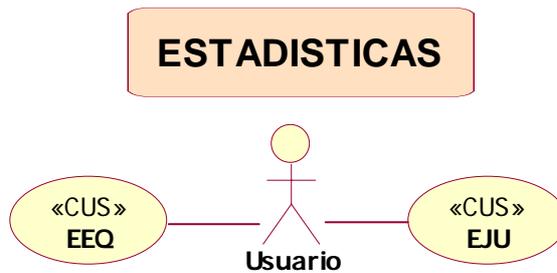
Modelo de Servicios de Usuario – Partido



Modelo de Servicios de Usuario – Información del Partido

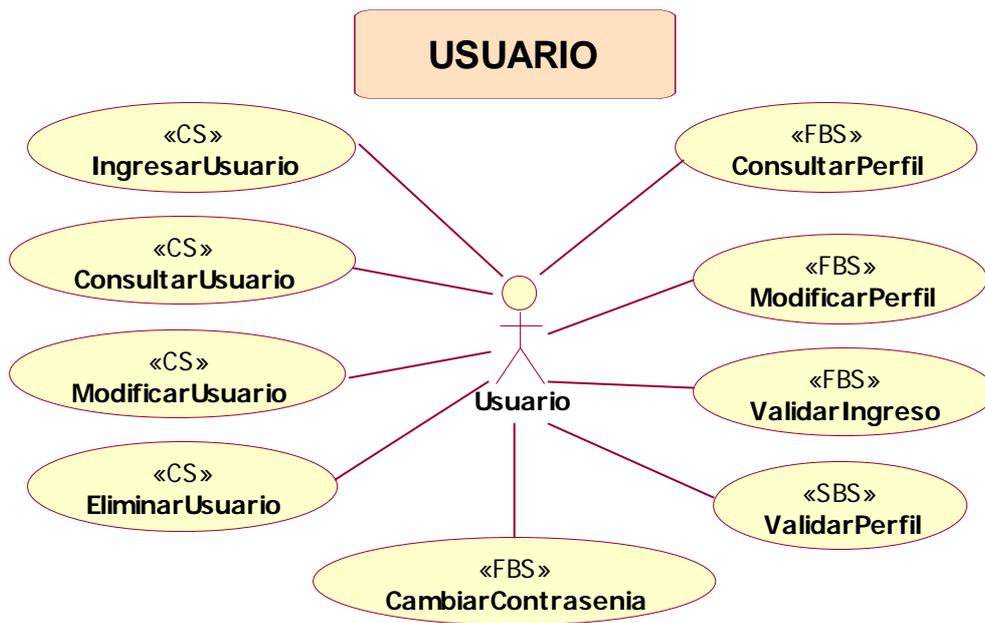


Modelo de Servicios de Usuario – Estadísticas

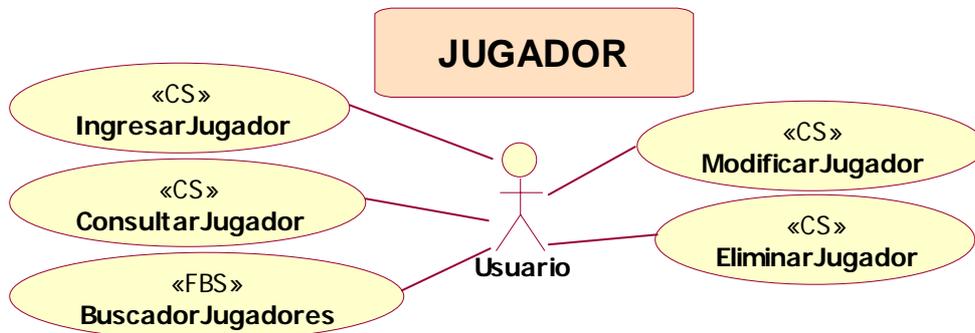


Luego de definir los servicios de manera general se debe ahora identificar si el servicio es compuesto <<CS>>, básico funcional <<FBS>> o básico estructural <<SBS>>, lo que da origen a un modelo intermedio al cual se lo va a llamar Modelo de casos de uso más servicios de usuario, como se muestra a continuación:

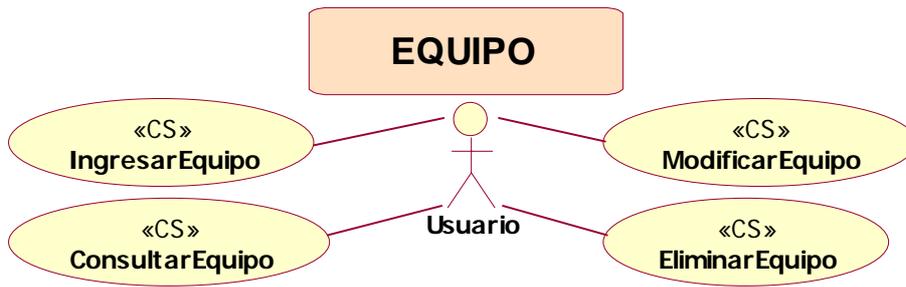
Modelo de casos de uso más servicios de usuario – Usuario



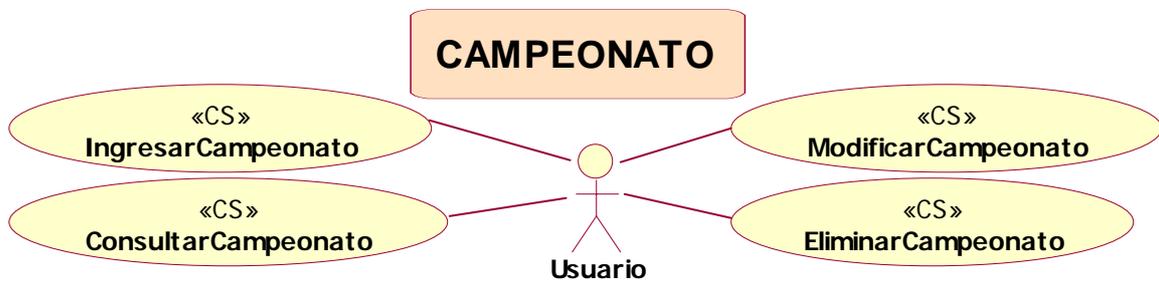
Modelo de casos de uso más servicios de usuario – Jugador



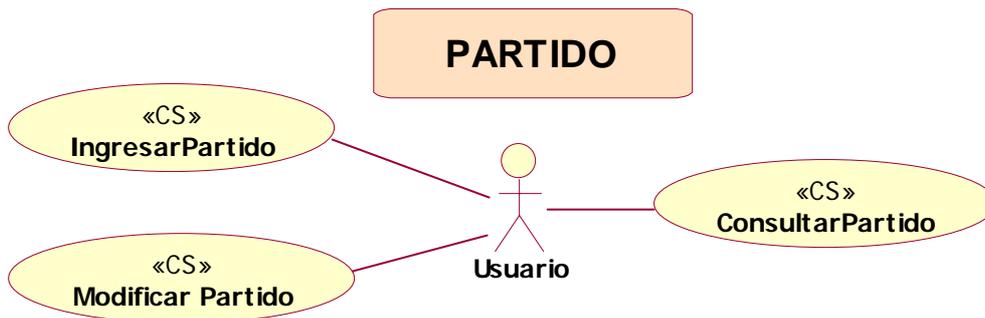
Modelo de casos de uso más servicios de usuario – Equipo



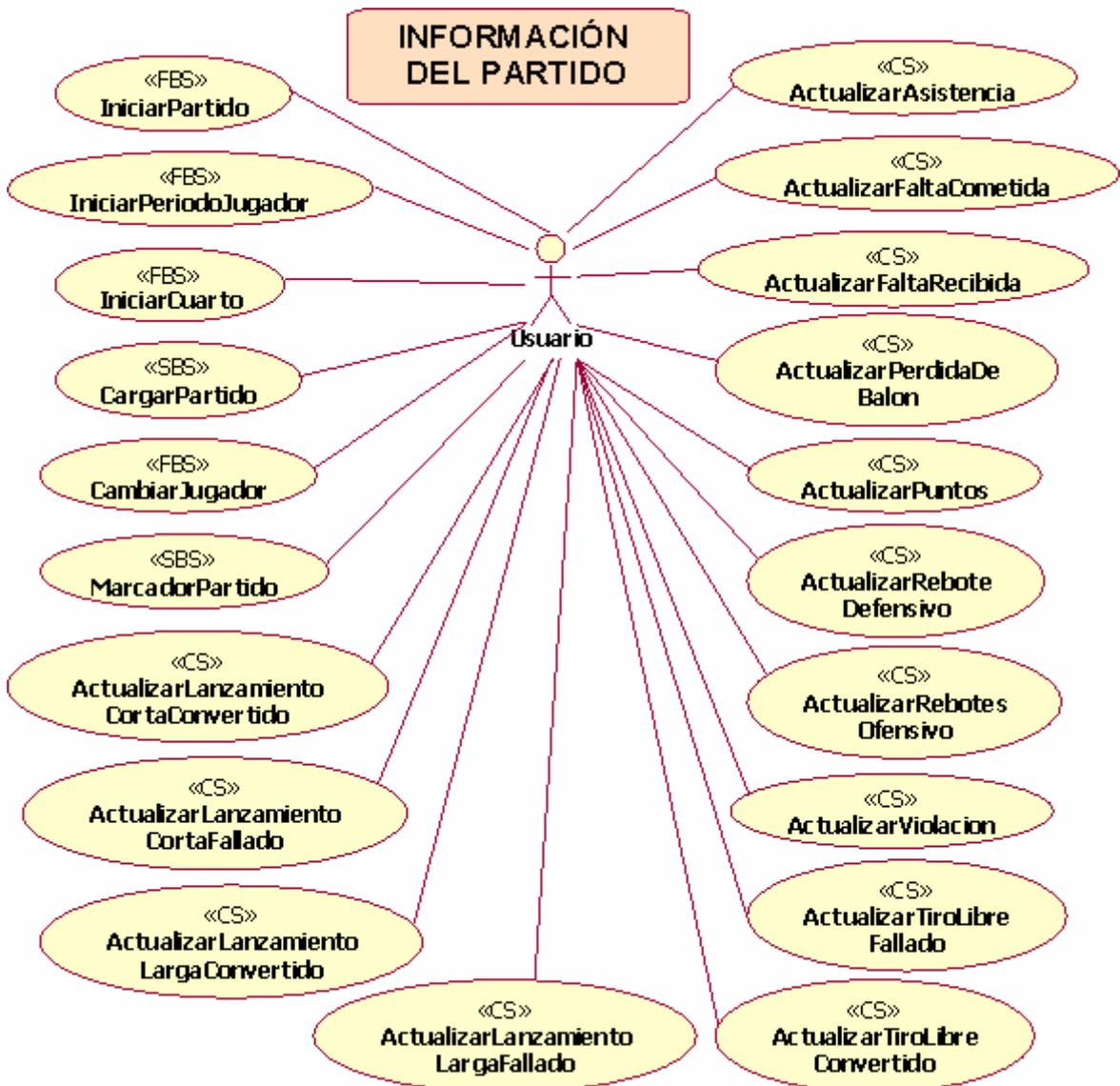
Modelo de casos de uso más servicios de usuario – Campeonato



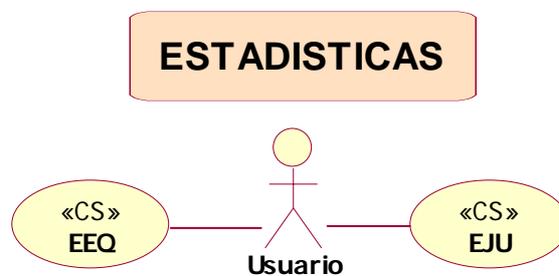
Modelo de casos de uso más servicios de usuario – Partido



Modelo de casos de uso más servicios de usuario – Información del Partido



Modelo de casos de uso más servicios de usuario – Estadísticas



Finalmente para obtener el Modelo de Casos de uso extendido se debe añadir las relaciones “include” y “extend”:

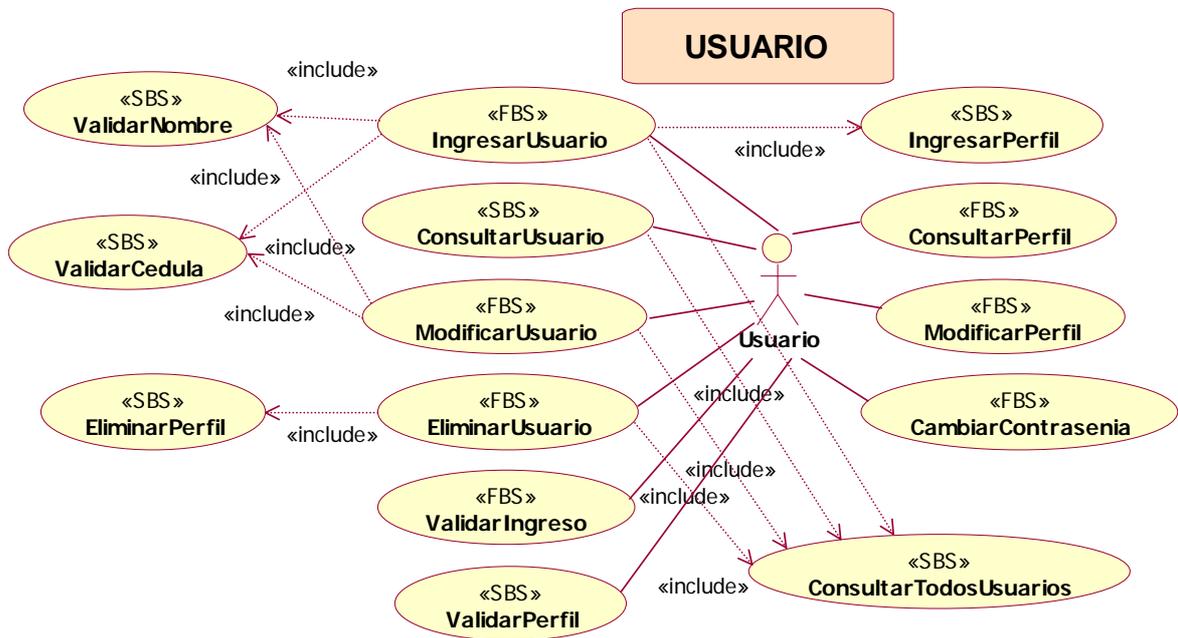


Figura 4.5: Modelo de casos de uso extendido – Usuario

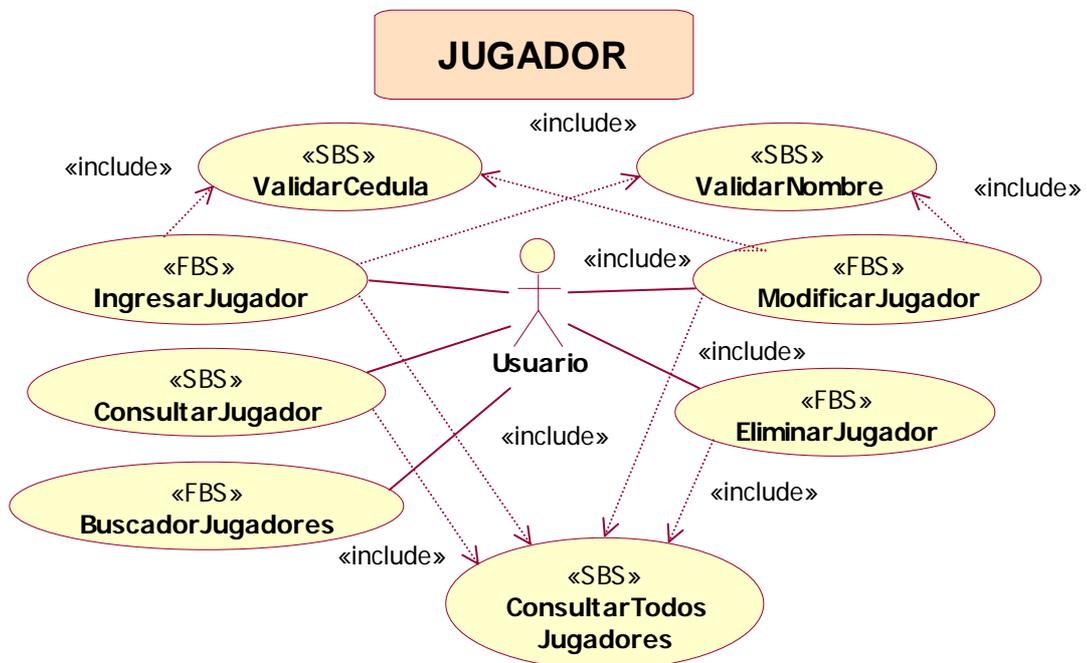


Figura 4.6: Modelo de casos de uso extendido – Jugador

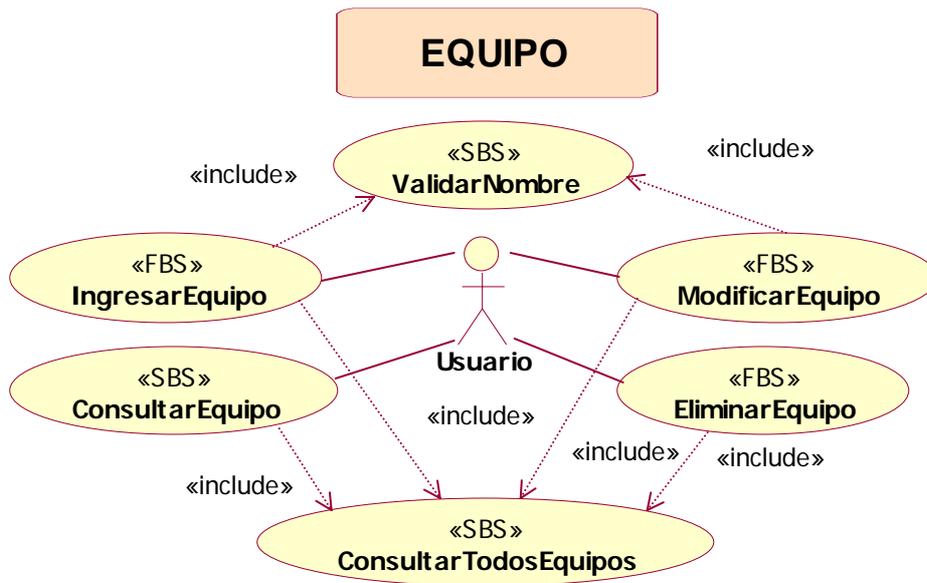


Figura 4.7: Modelo de casos de uso extendido – Equipo

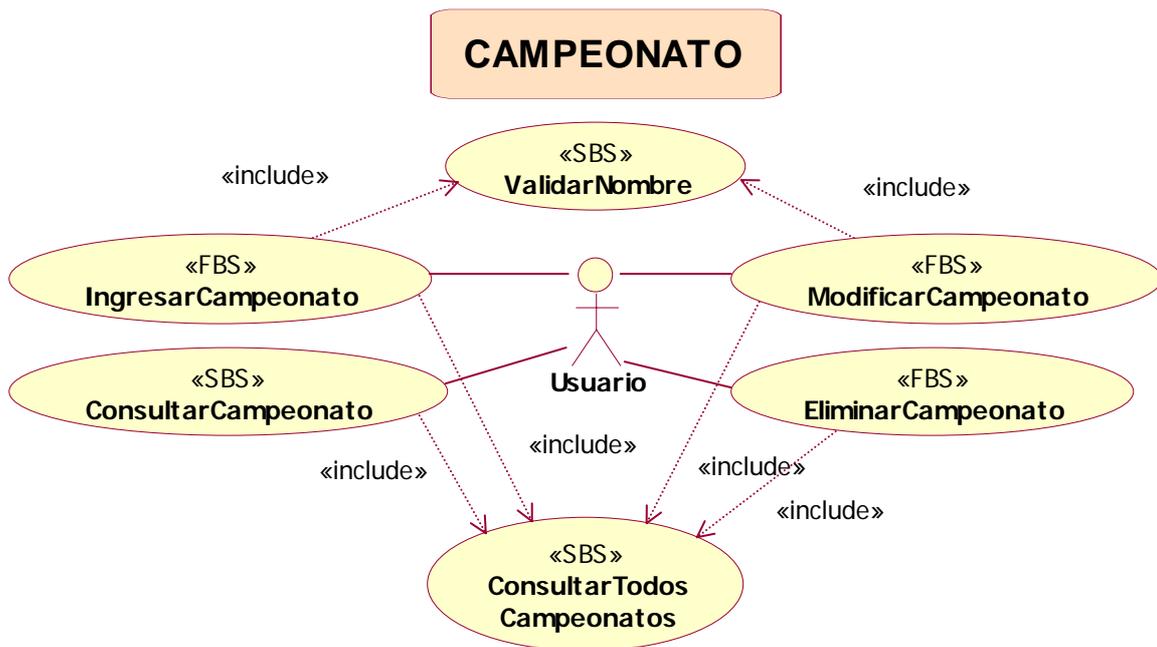


Figura 4.8: Modelo de casos de uso extendido – Campeonato

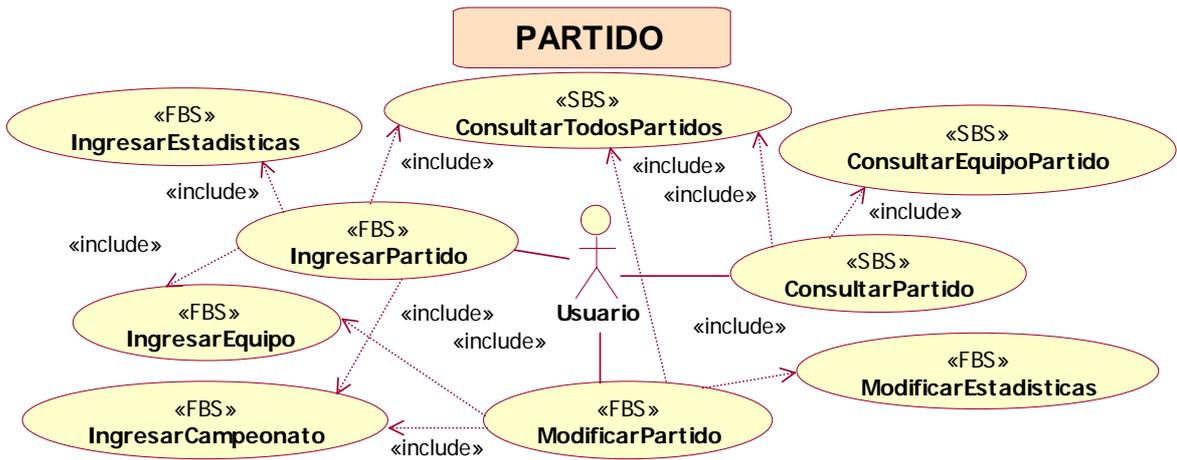


Figura 4.9: Modelo de casos de uso extendido – Partido

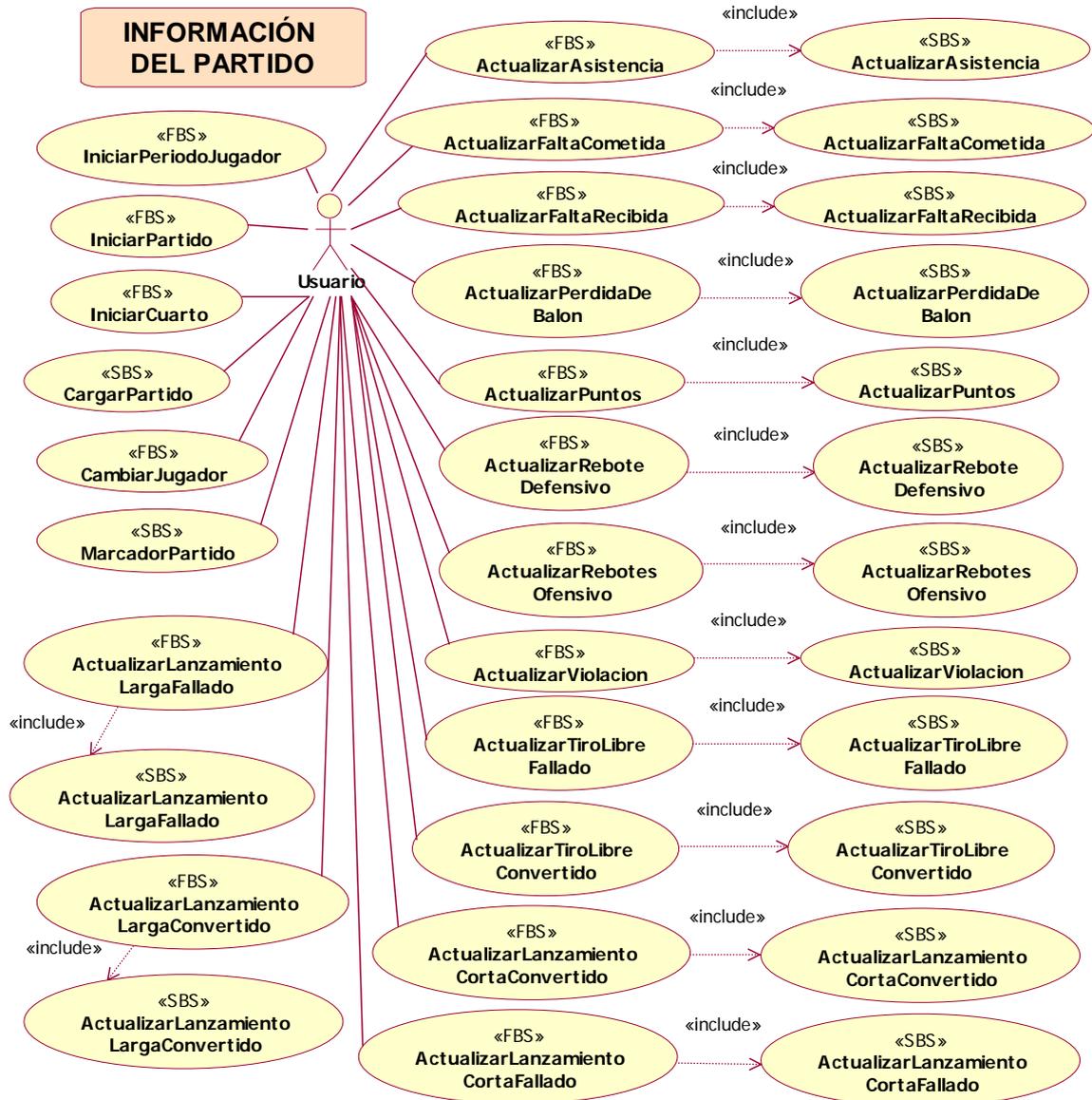


Figura 4.10: Modelo de casos de uso extendido – Información del Partido

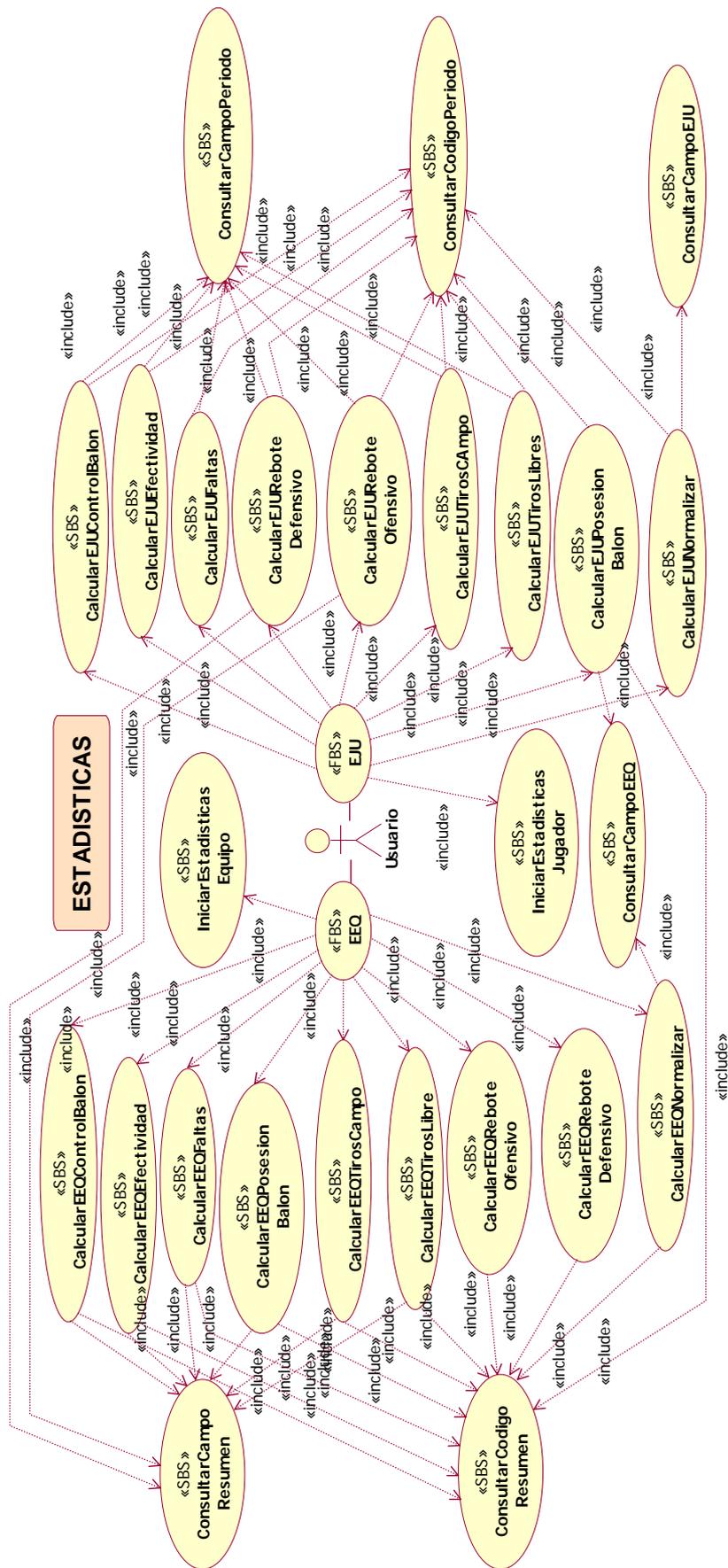
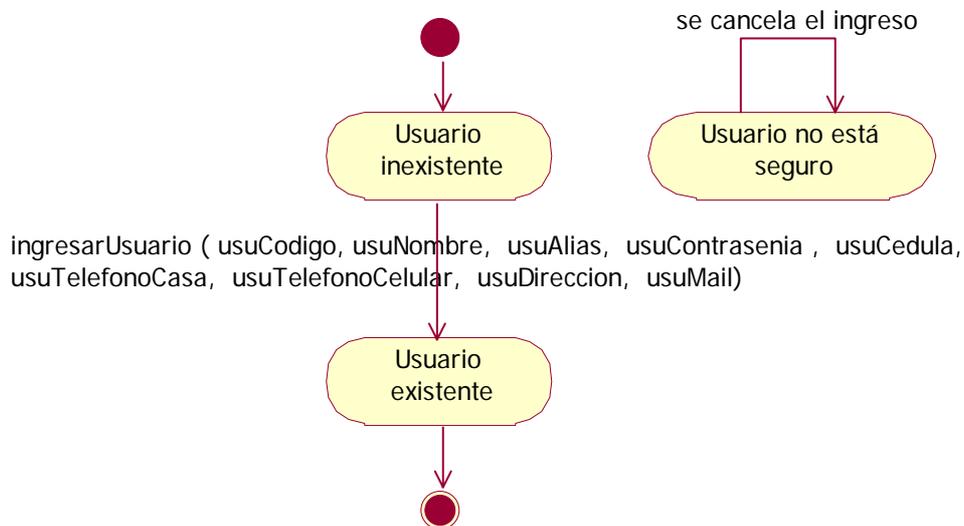
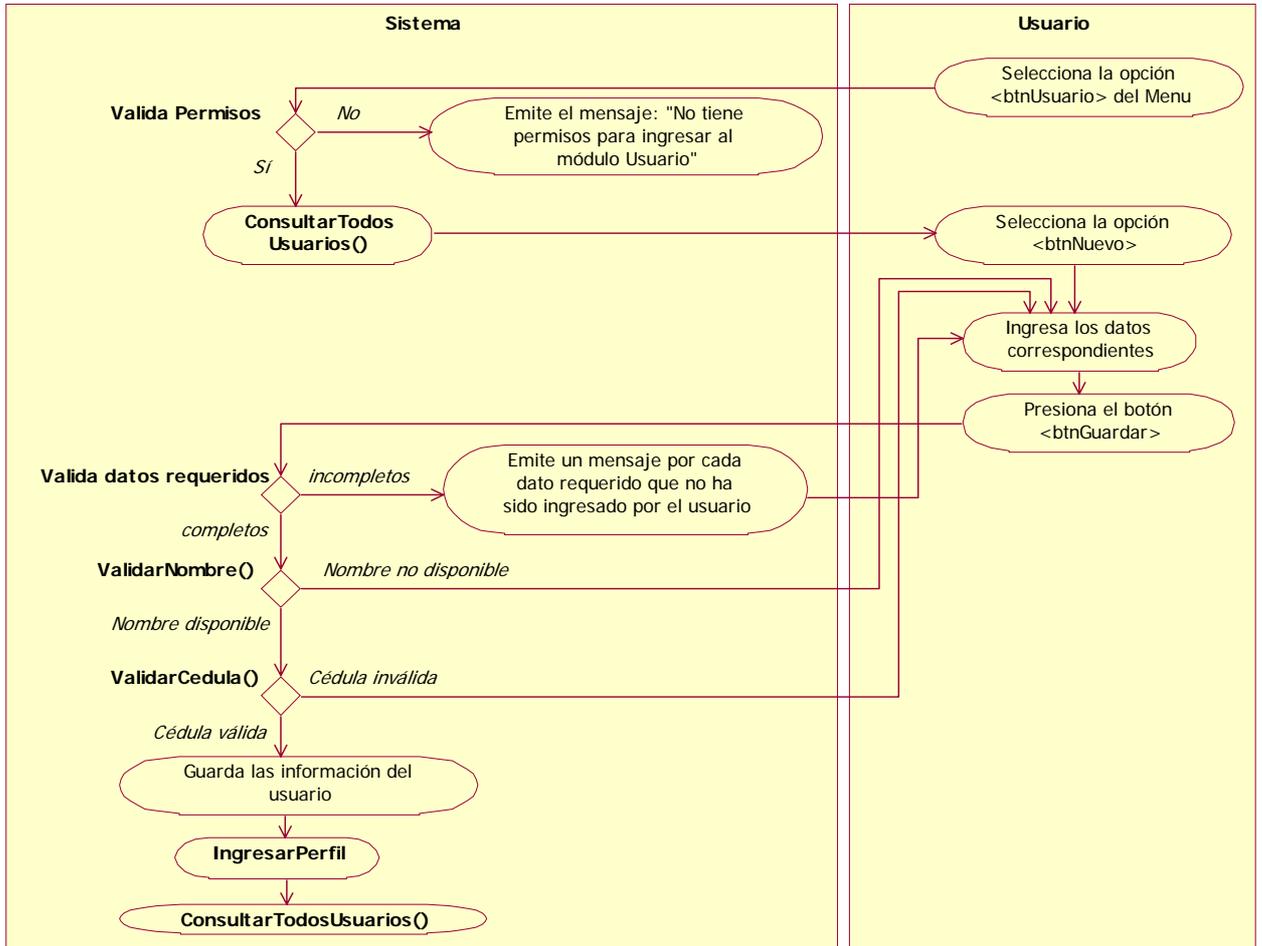


Figura 4.11: Modelo de casos de uso extendido – Estadísticas

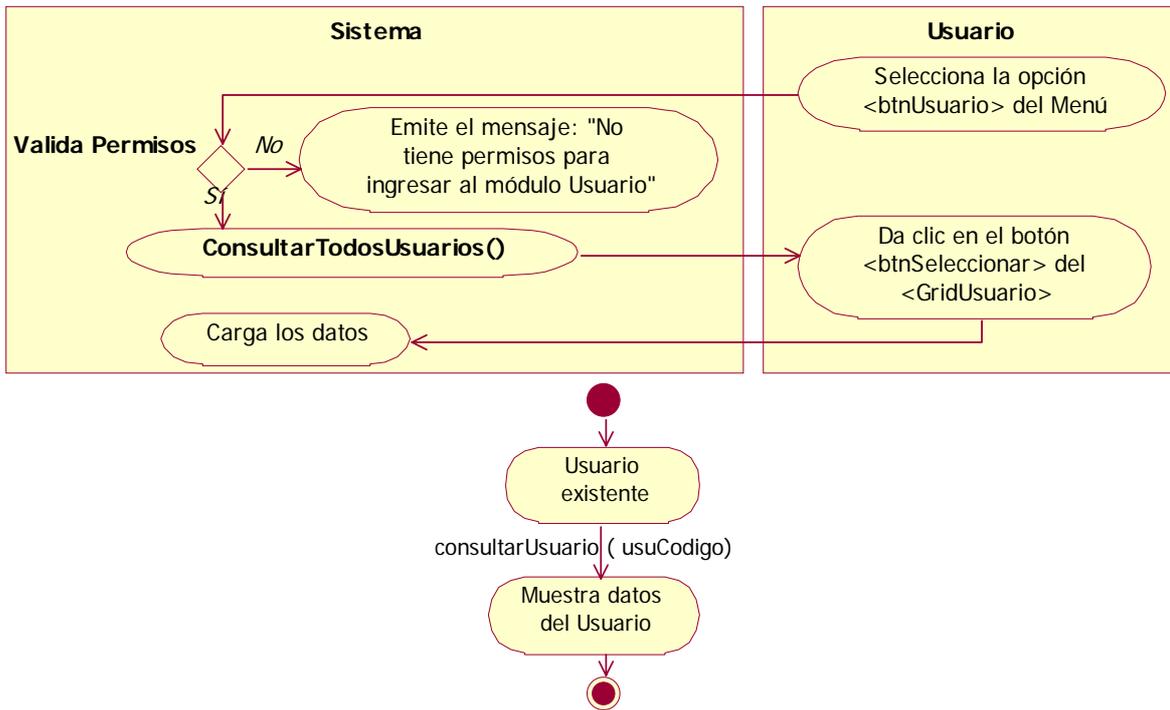
A continuación se presenta el Modelo de Composición de Servicios, el cual permite conocer la secuencia de cómo se ejecutará el servicio de usuario.

- Paquete Usuario

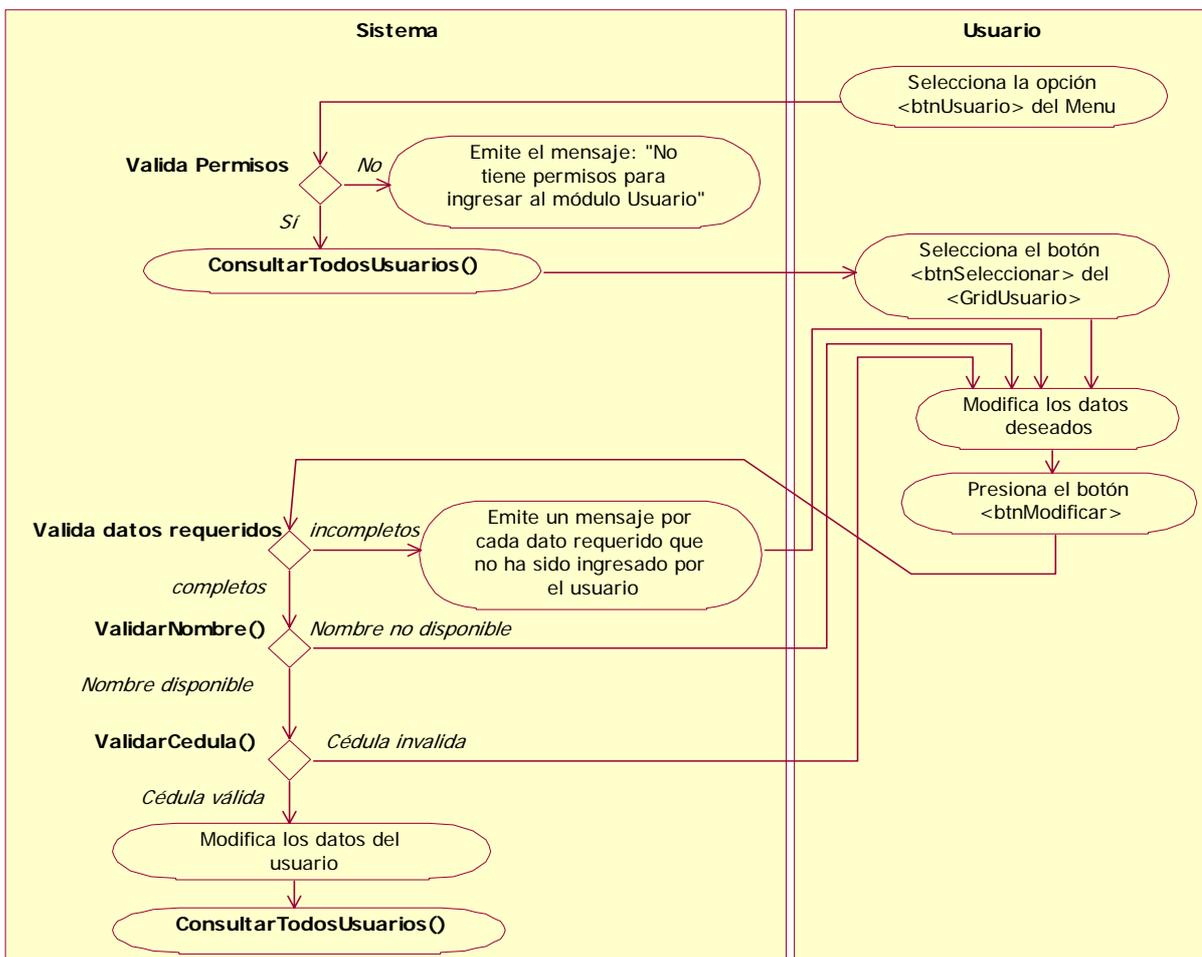
Modelo de composición de servicios – Ingresar Usuario

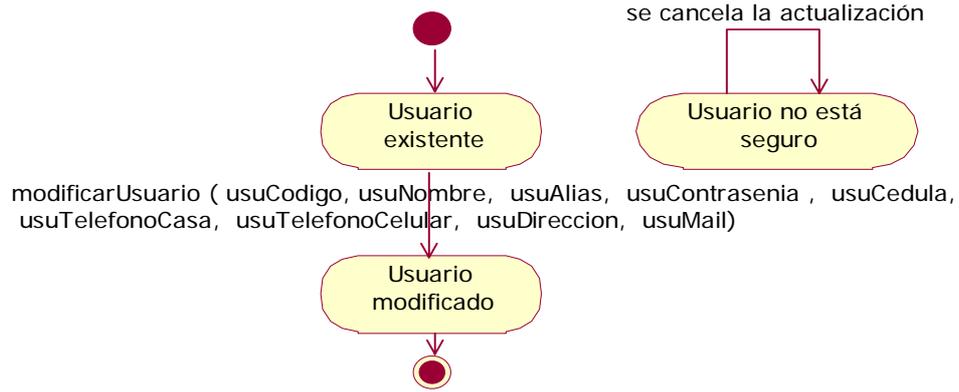


Modelo de composición de servicios – Consultar Usuario

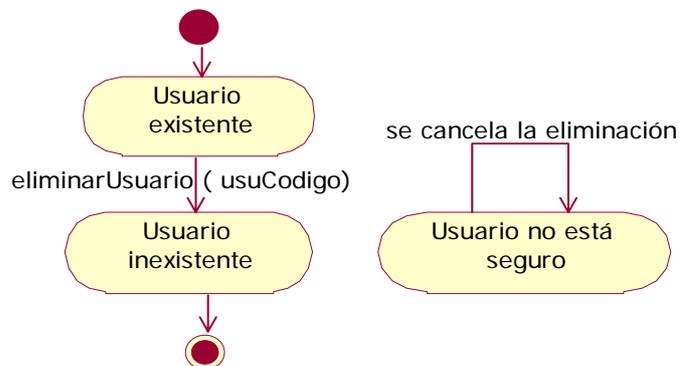
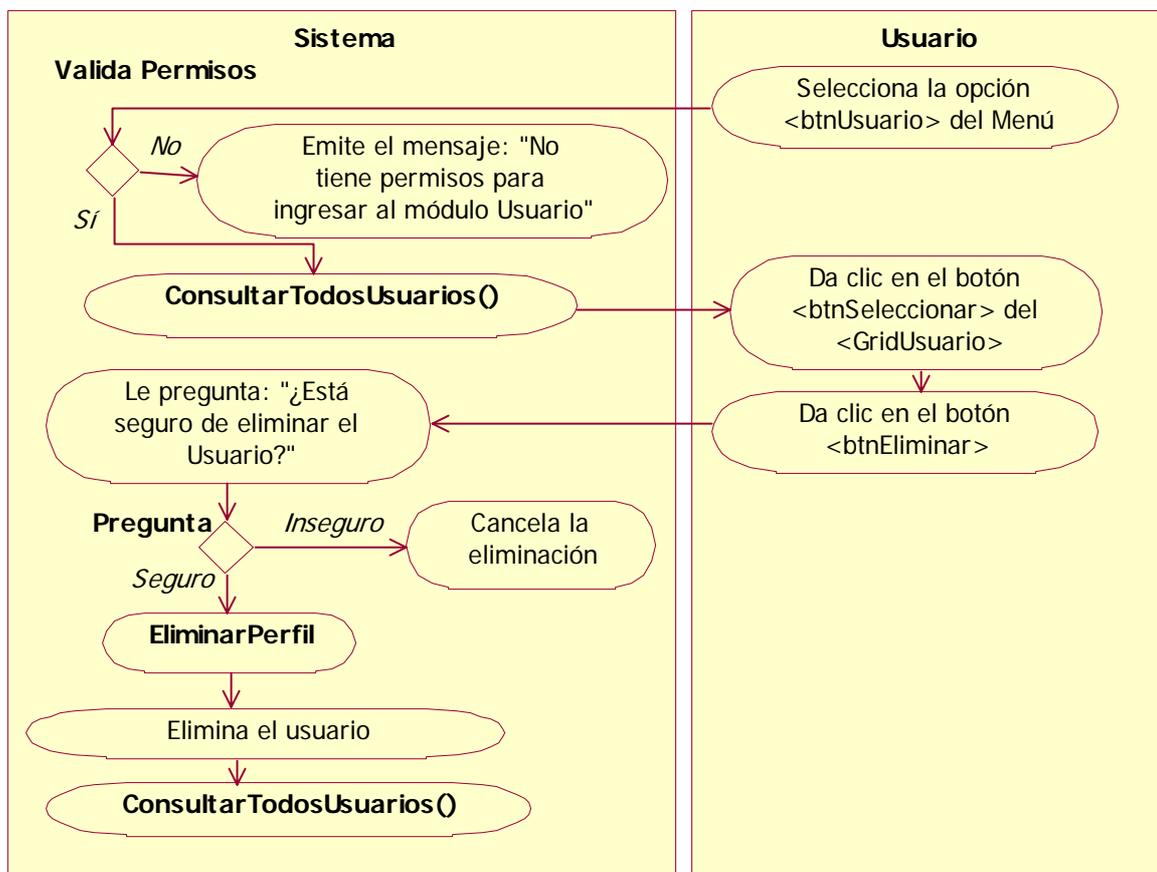


Modelo de composición de servicios – Modificar Usuario

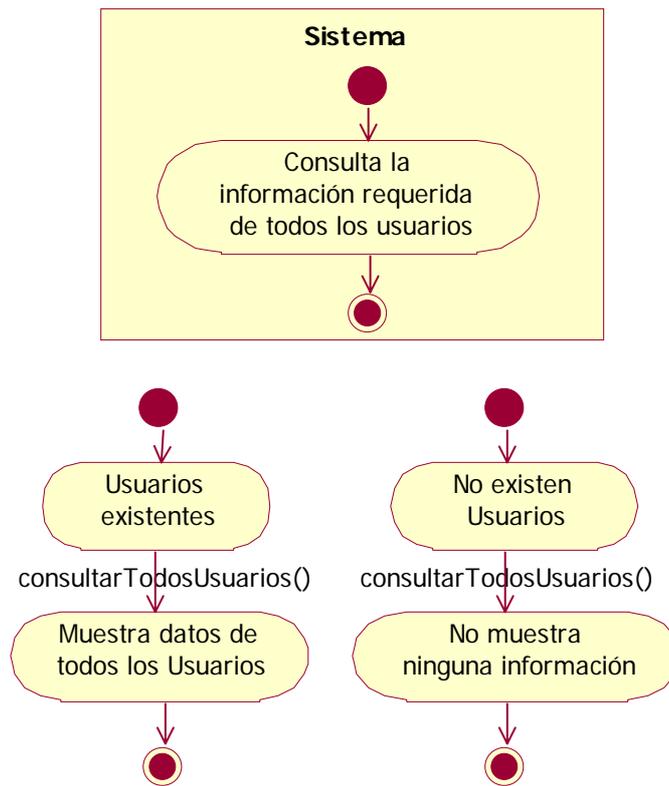




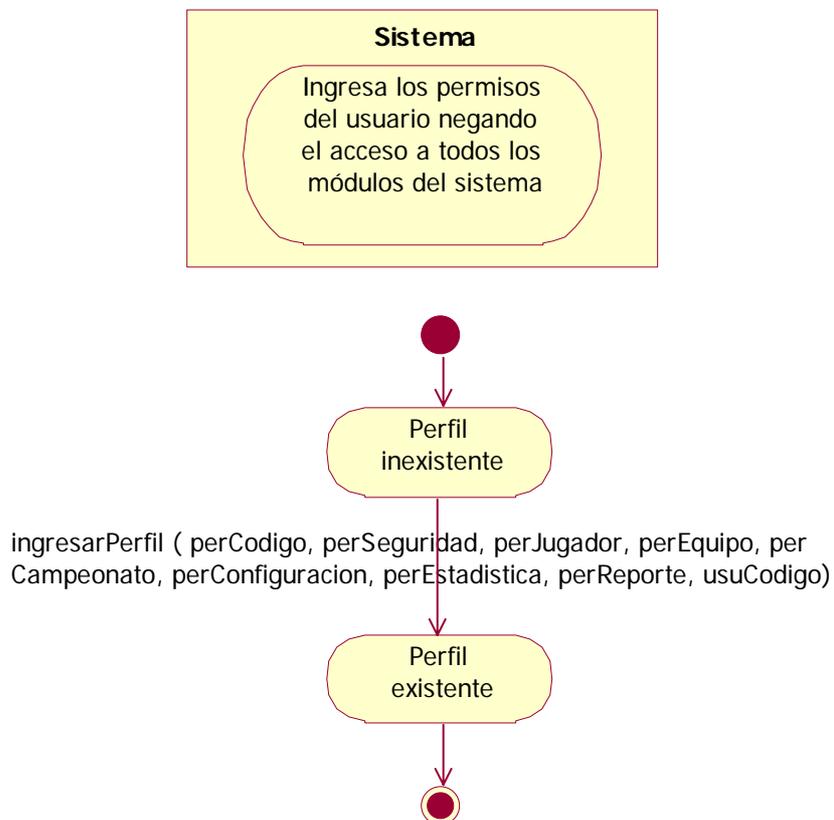
Modelo de composición de servicios – Eliminar Usuario



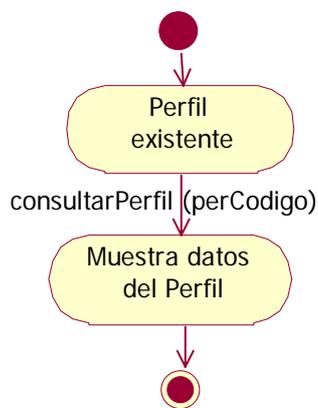
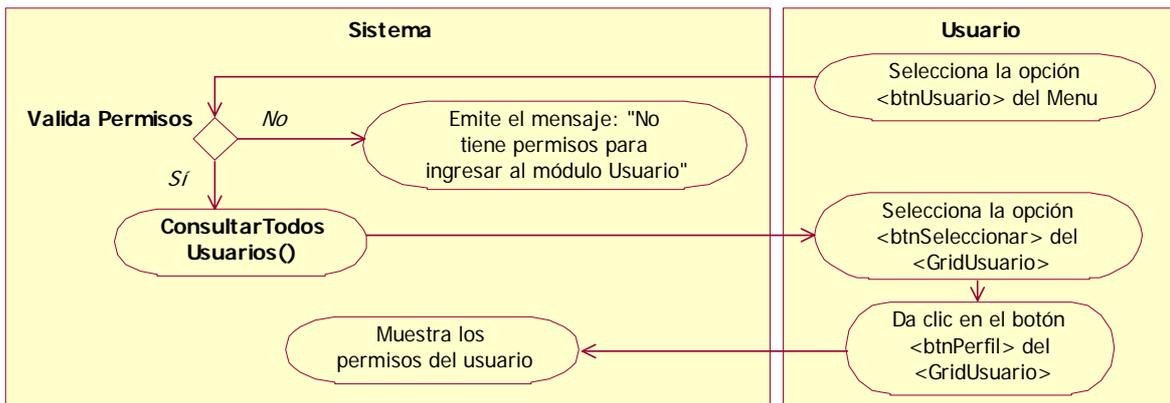
Modelo de composición de servicios – Consultar Todos los Usuarios



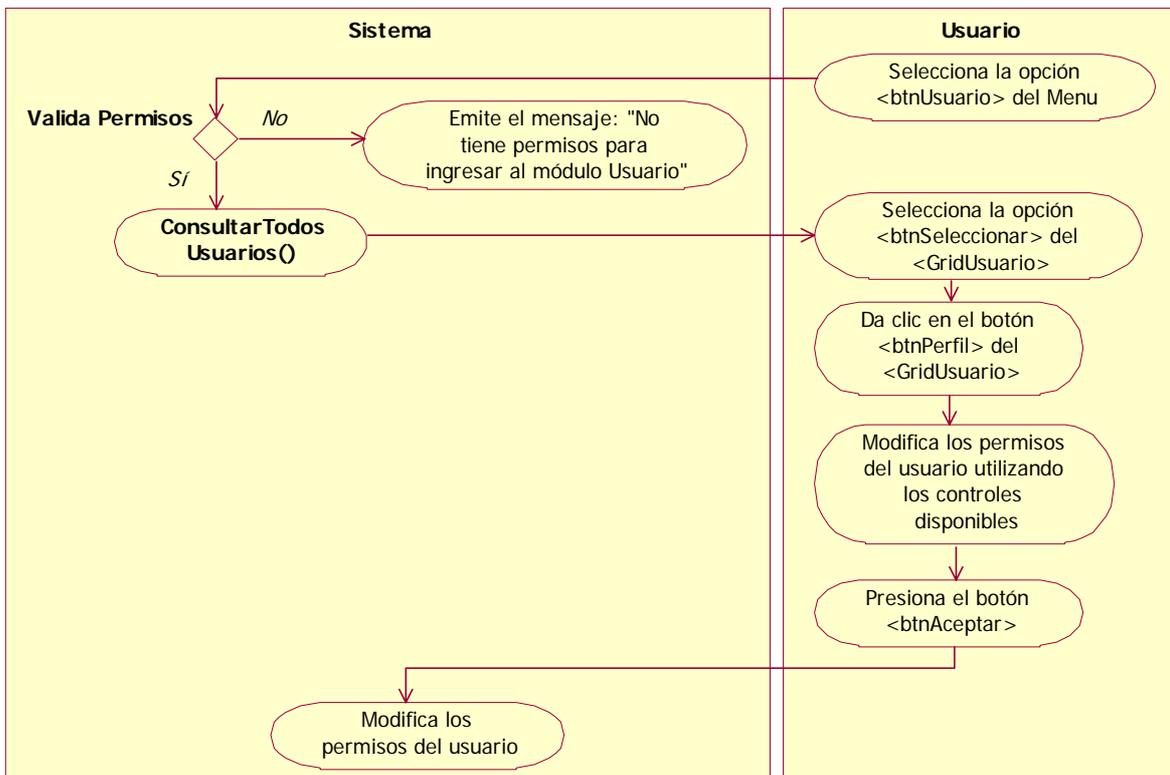
Modelo de composición de servicios – Ingresar Perfil

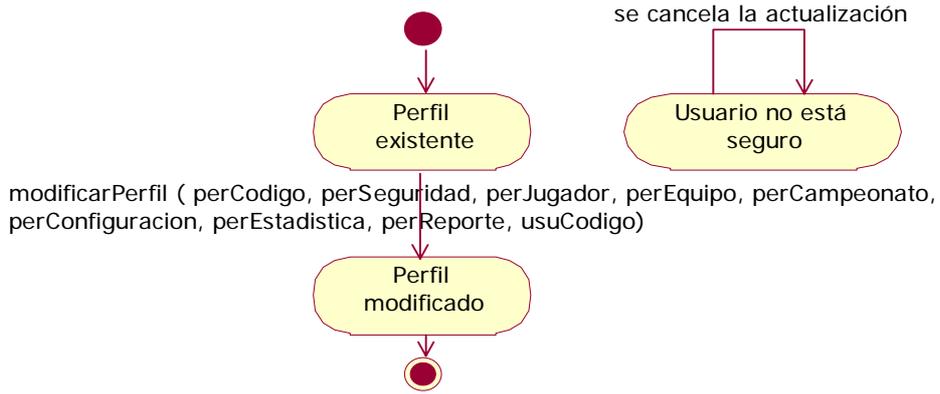


Modelo de composición de servicios – Consultar Perfil

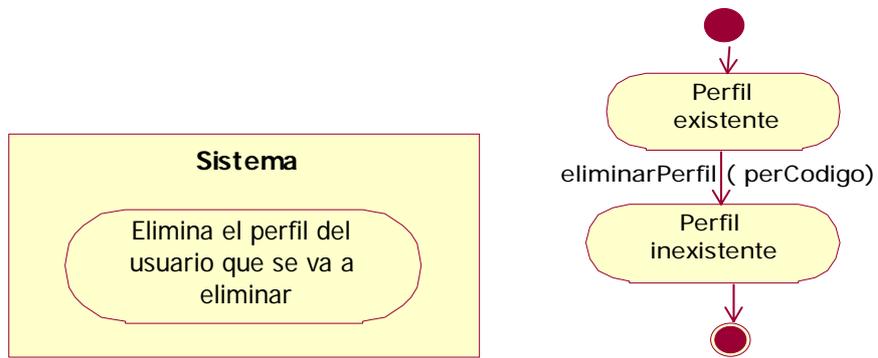


Modelo de composición de servicios – Modificar Perfil

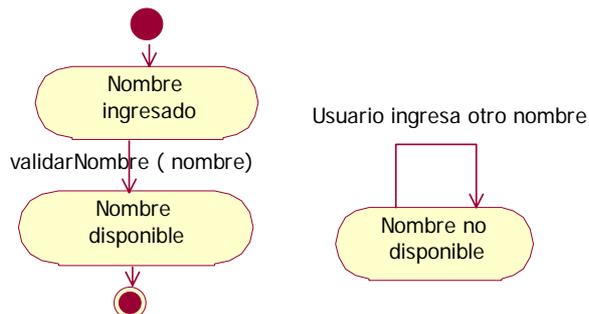
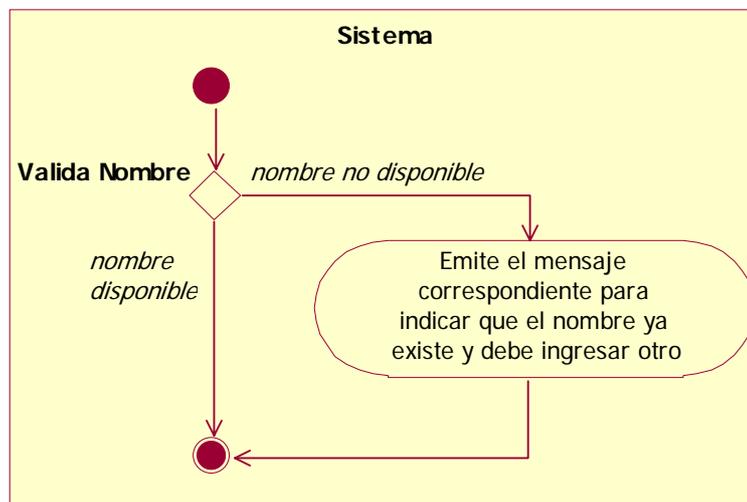




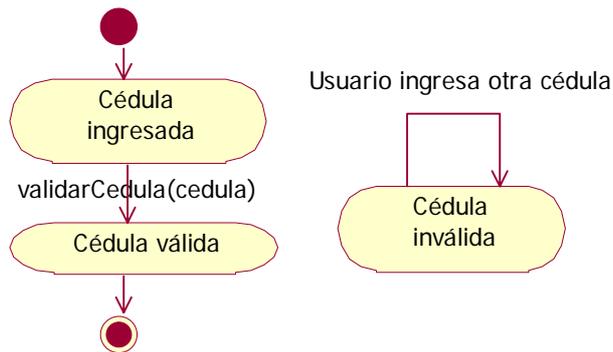
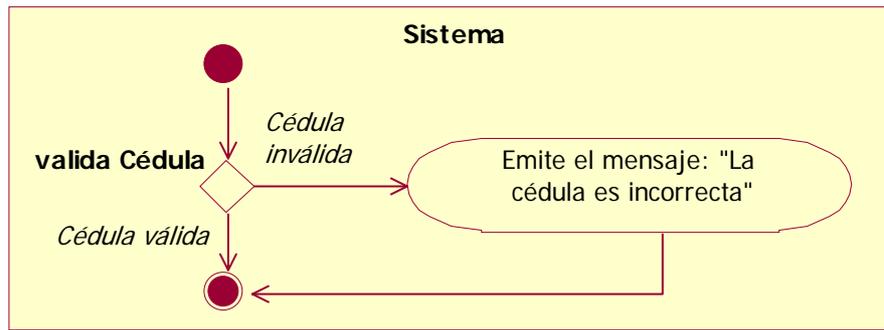
Modelo de composición de servicios – Eliminar Perfil



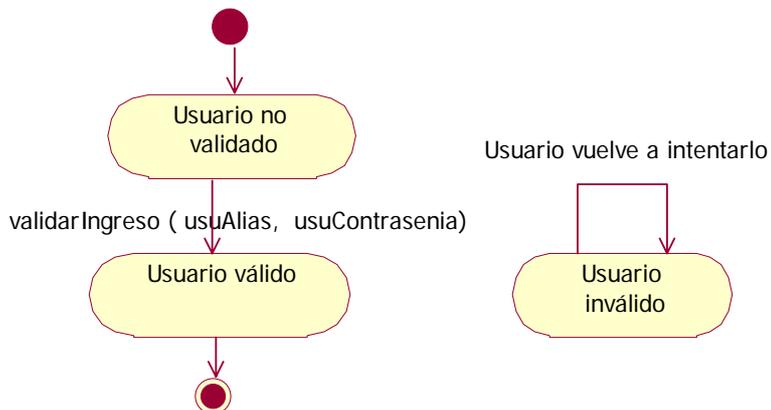
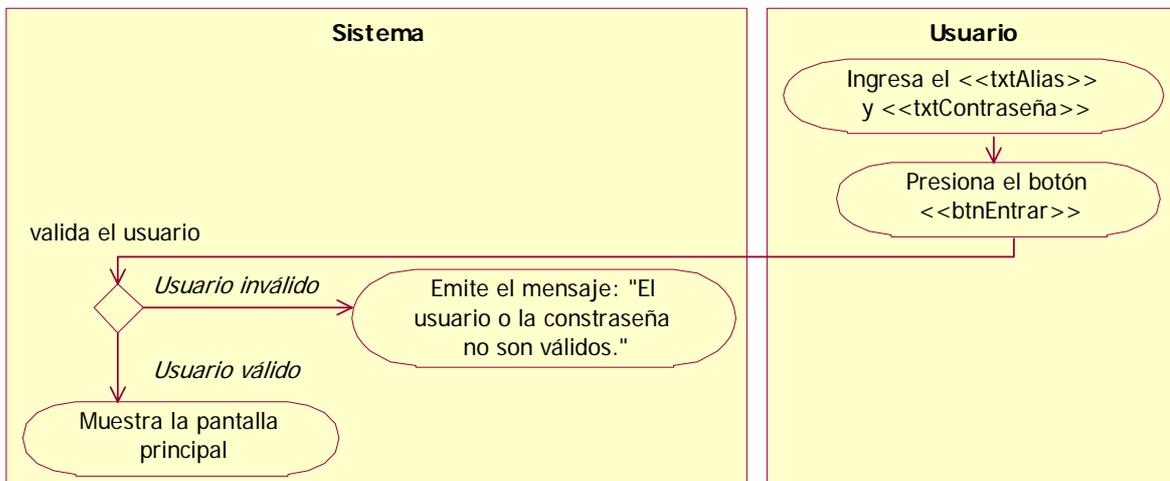
Modelo de composición de servicios – Validar Nombre



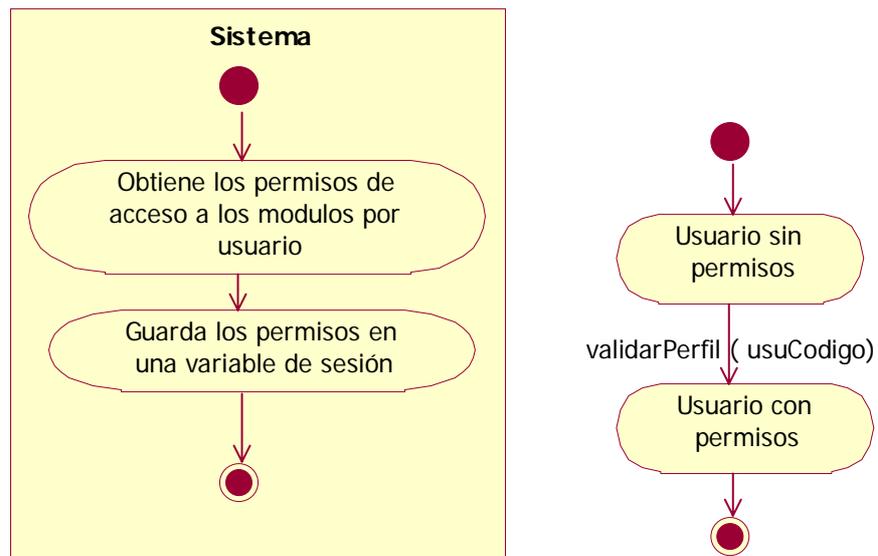
Modelo de composición de servicios – Validar Cédula



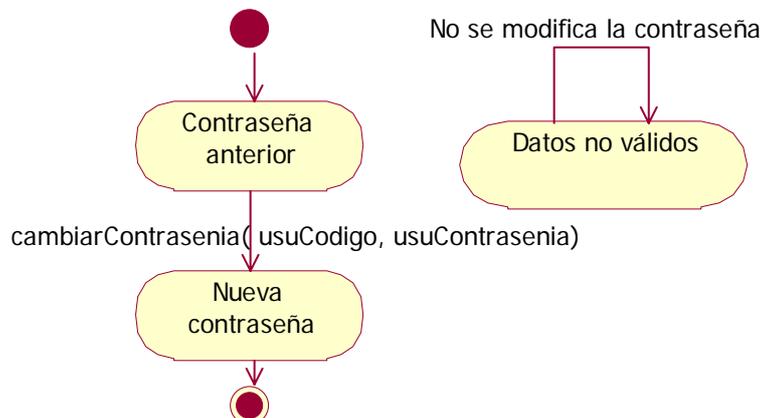
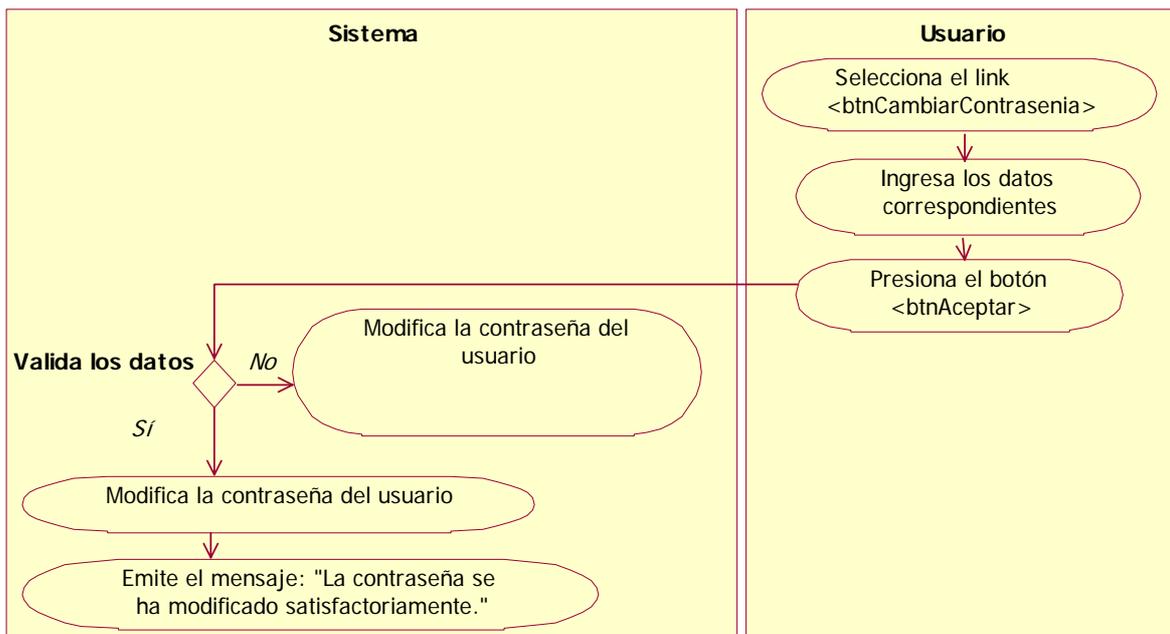
Modelo de composición de servicios – Validar Ingreso



Modelo de composición de servicios – Validar Perfil

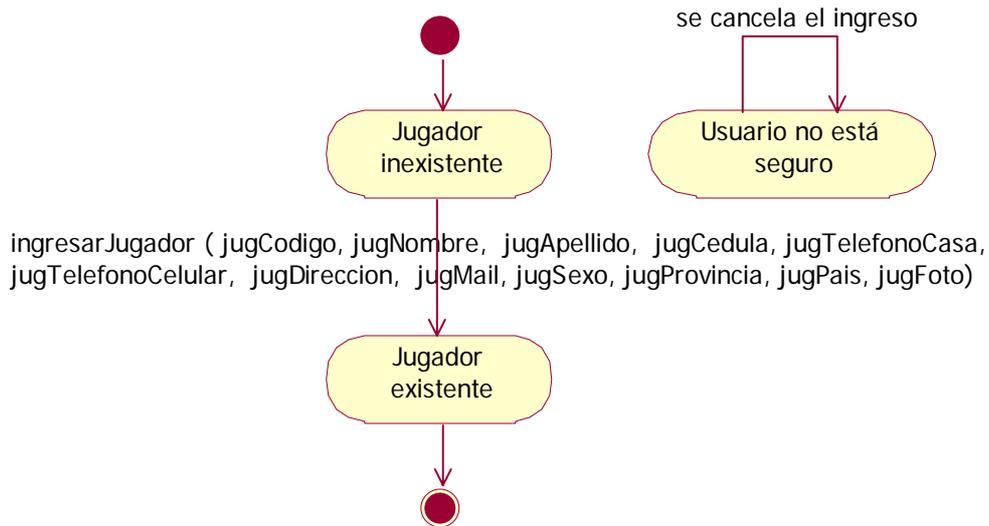
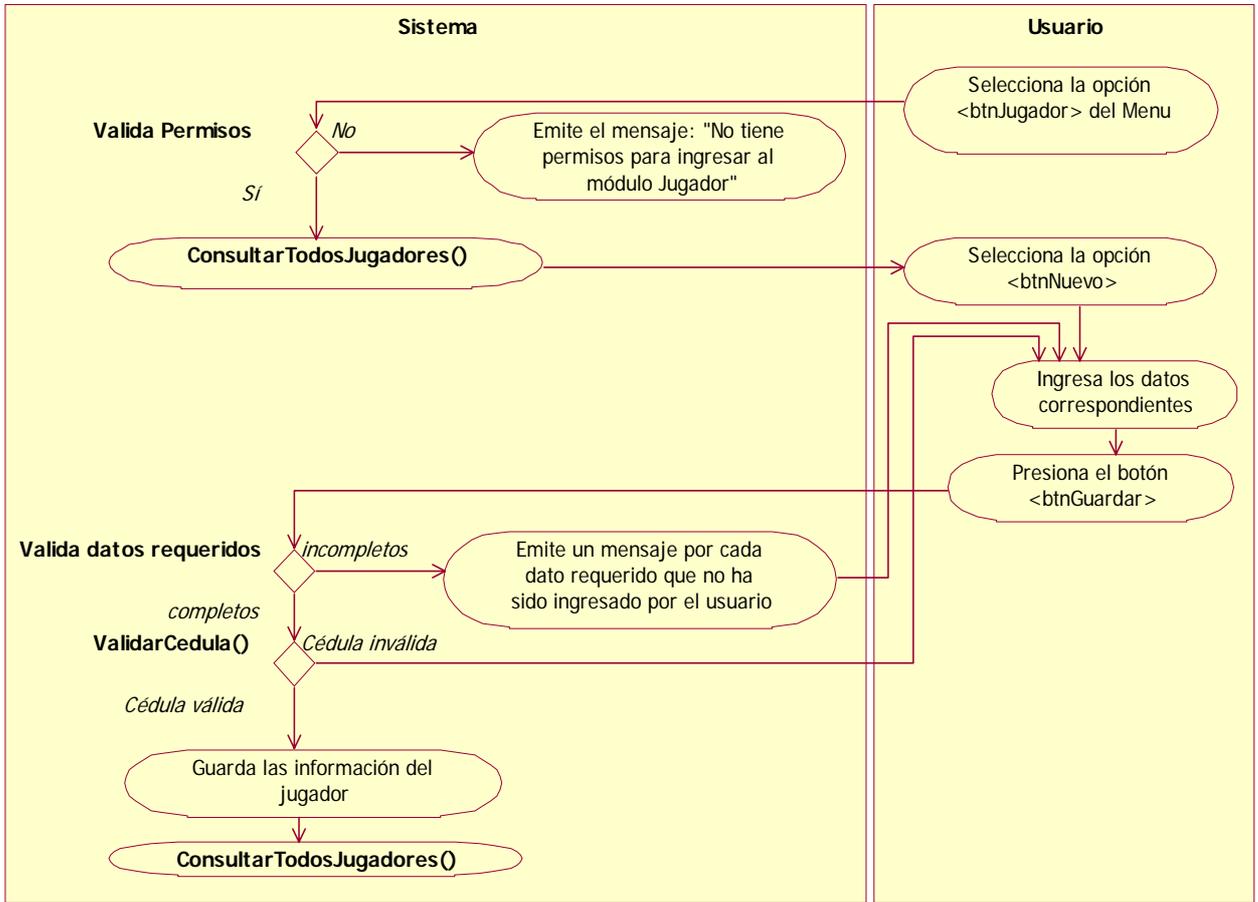


Modelo de composición de servicios – Cambiar Contraseña

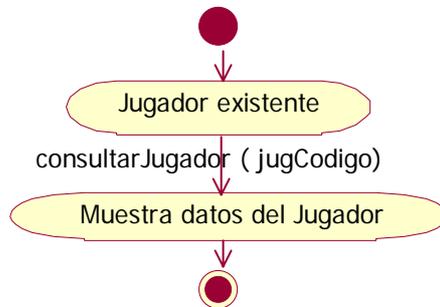
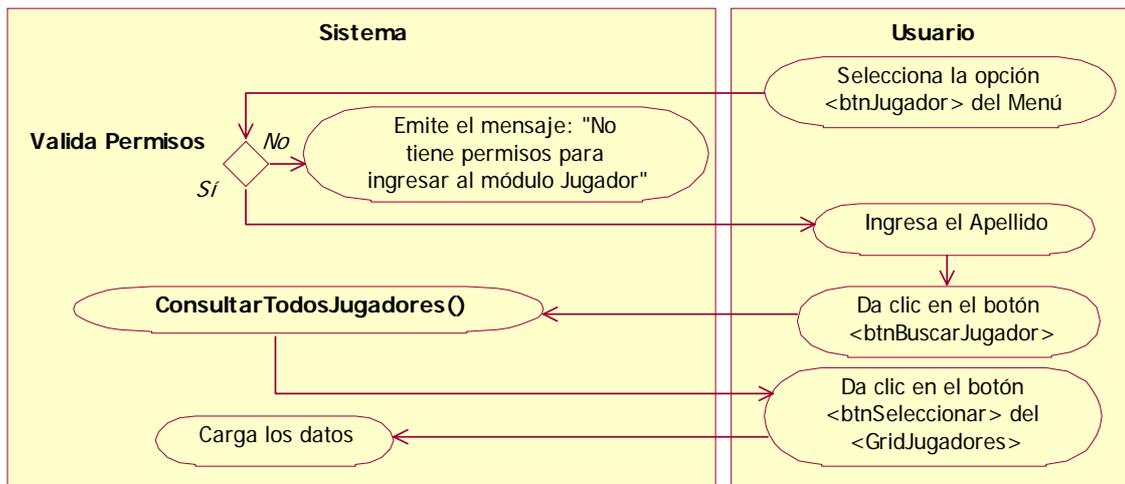


- Paquete Jugador

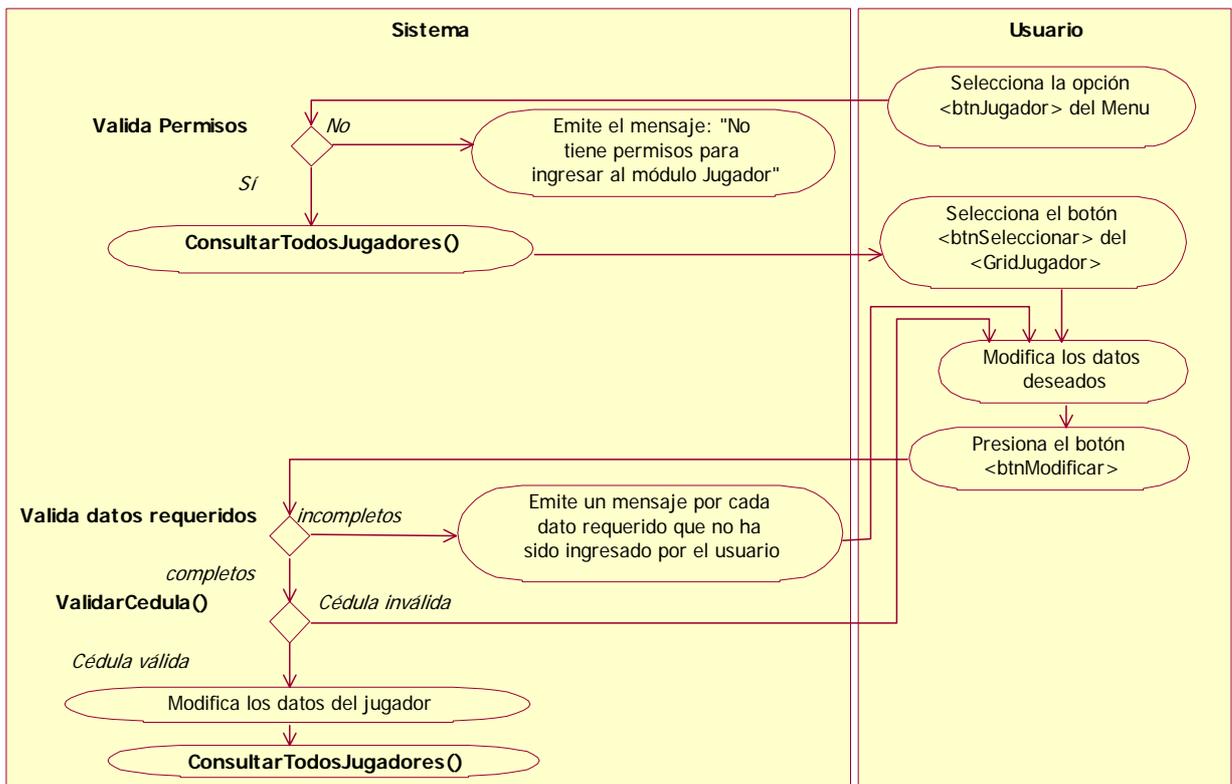
Modelo de composición de servicios – Ingresar Jugador

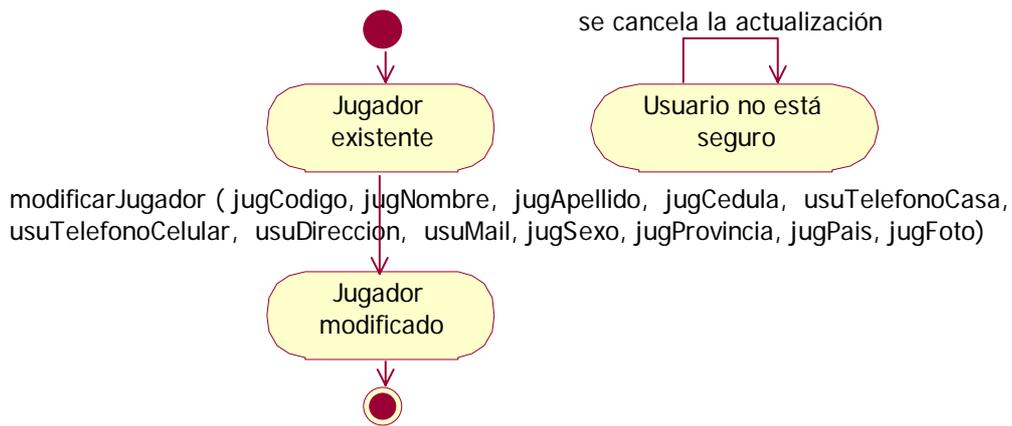


Modelo de composición de servicios – Consultar Jugador

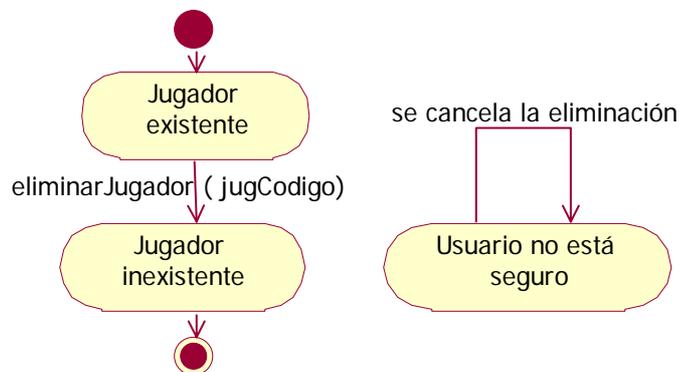
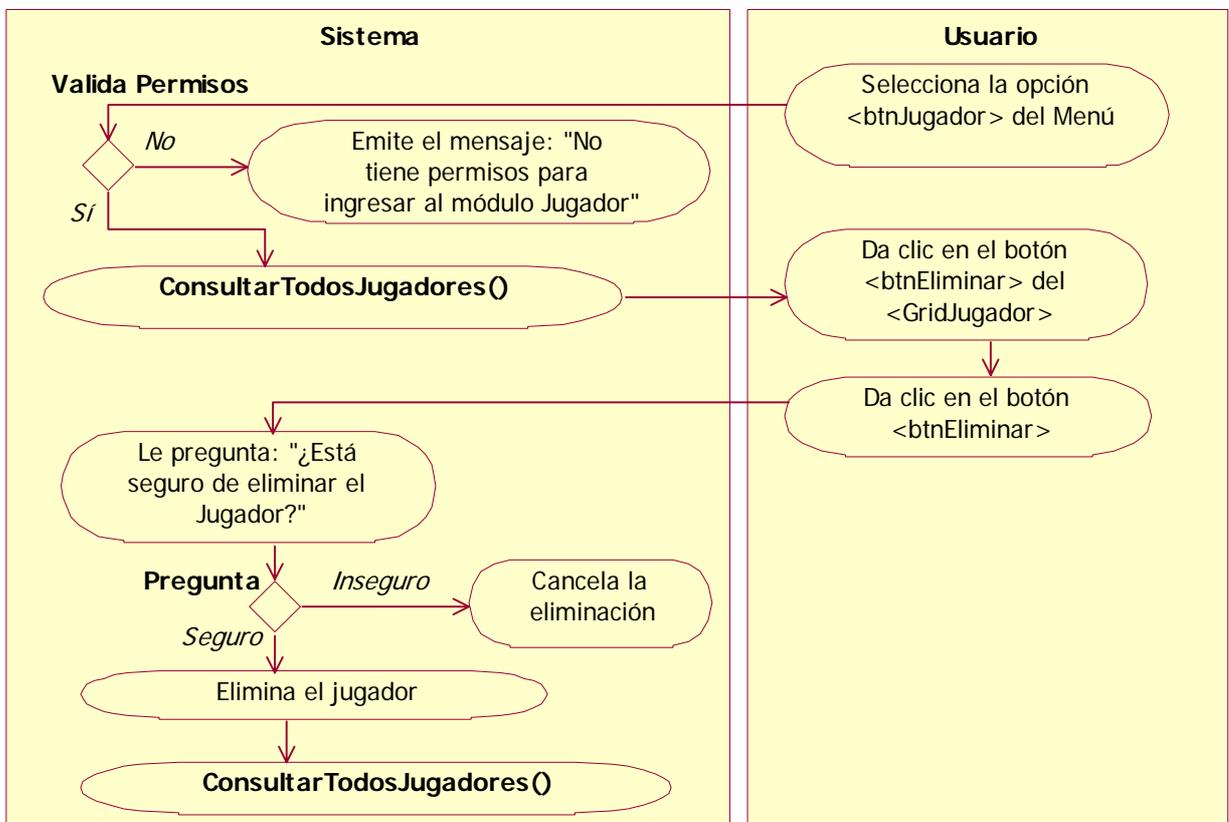


Modelo de composición de servicios – Modificar Jugador

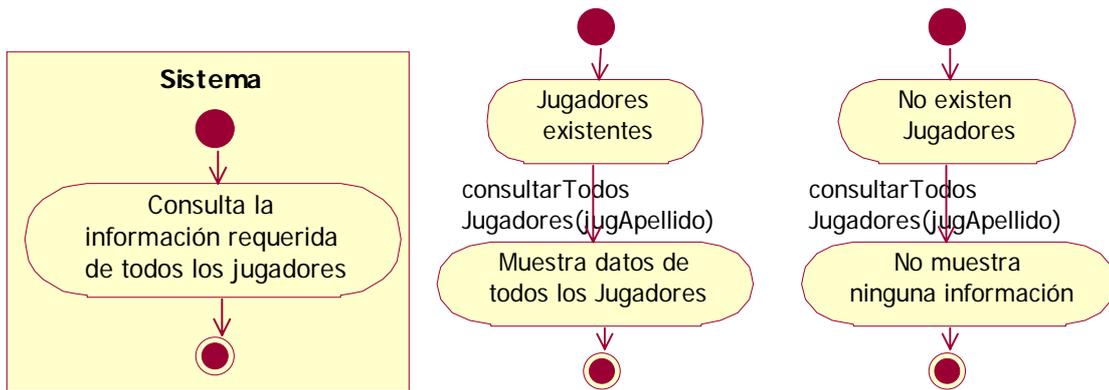




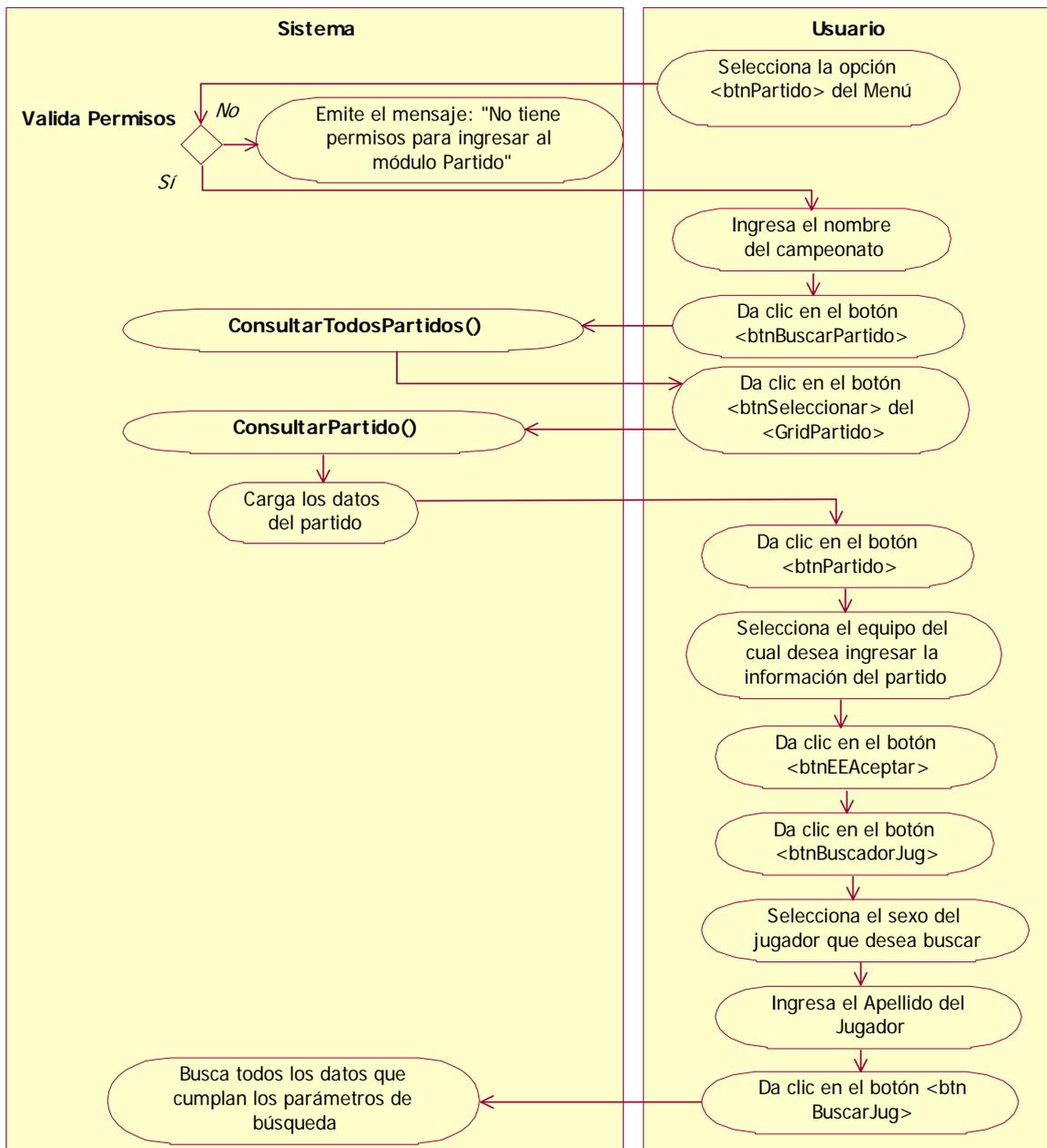
Modelo de composición de servicios – Eliminar Jugador

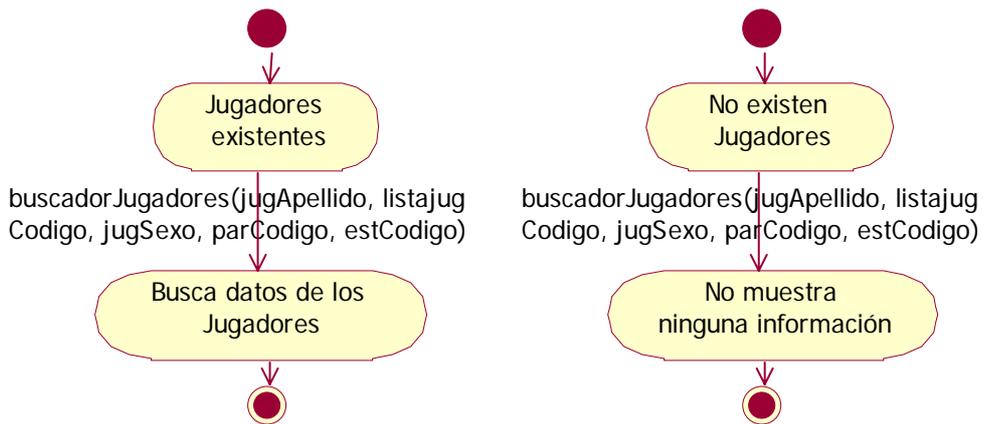


Modelo de composición de servicios – Consultar Todos los Jugadores



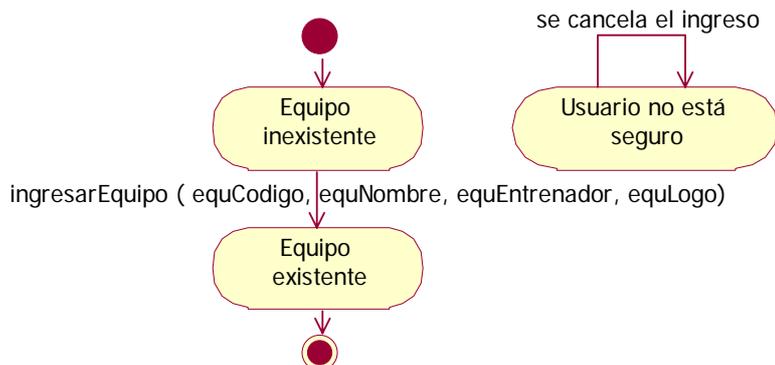
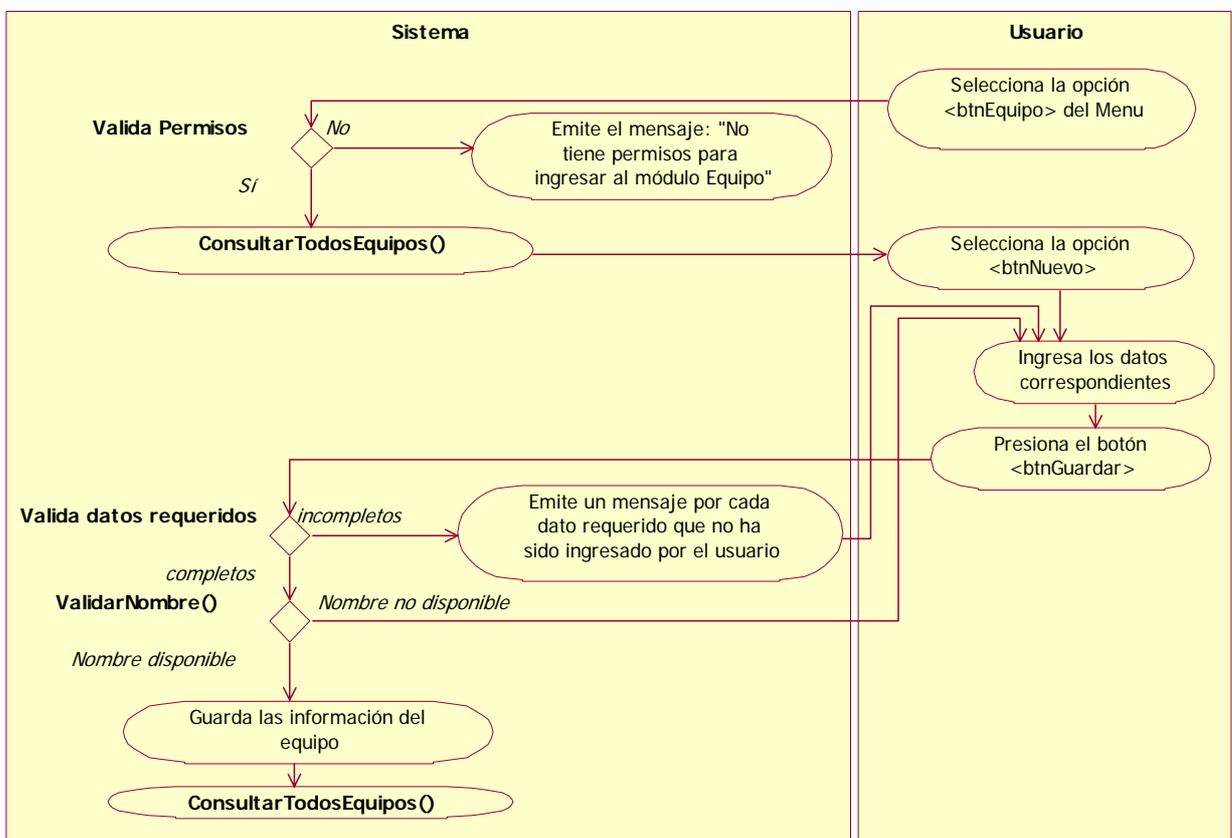
Modelo de composición de servicios – Buscador Jugadores



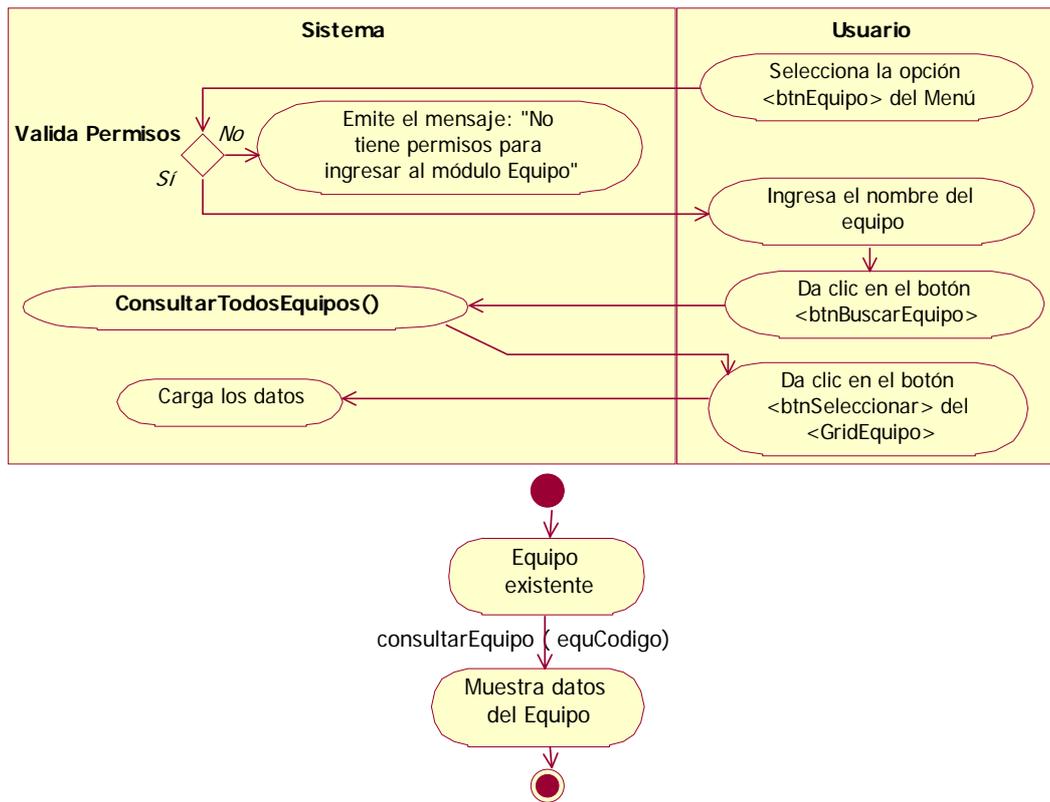


- Paquete Equipo

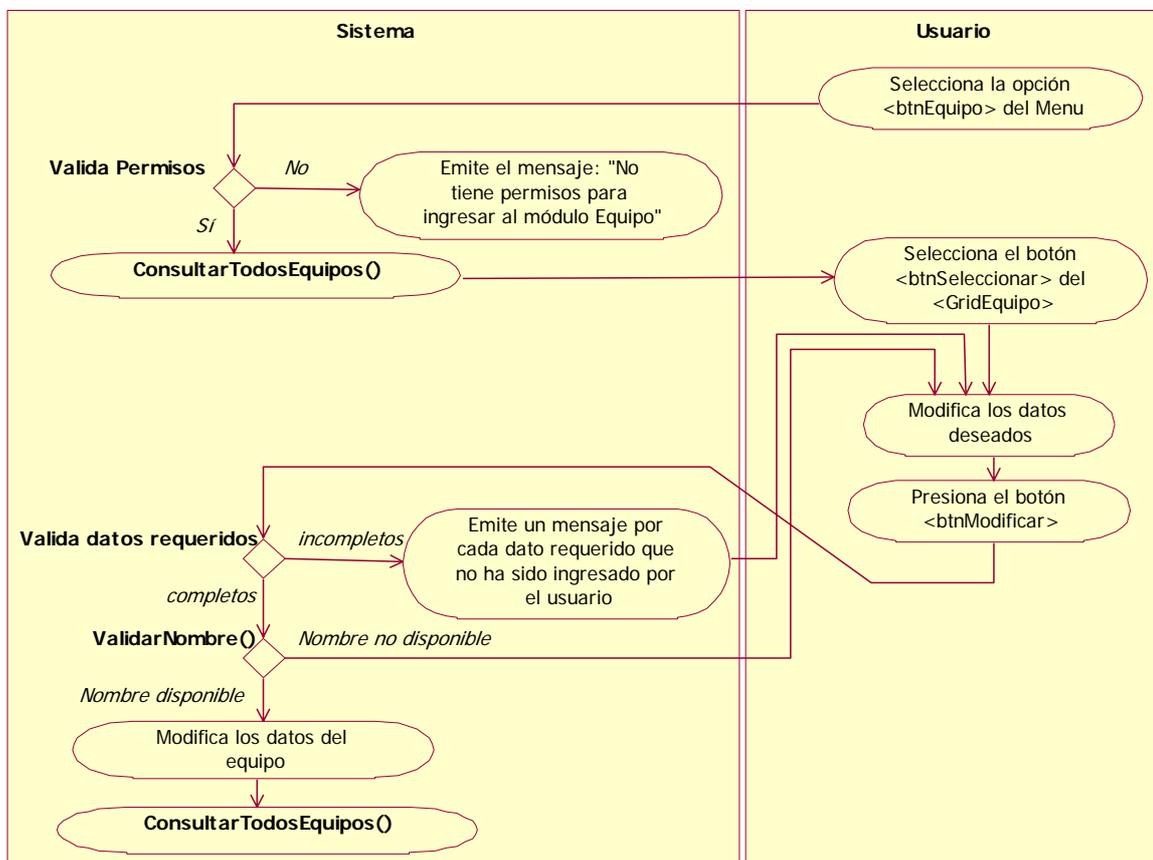
Modelo de composición de servicios – Ingresar Equipo

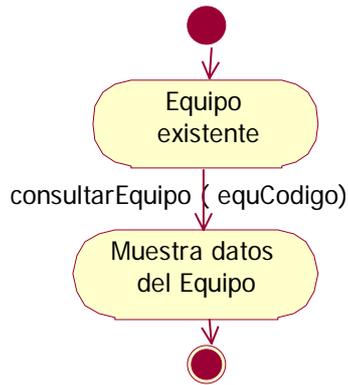


Modelo de composición de servicios – Consultar Equipo

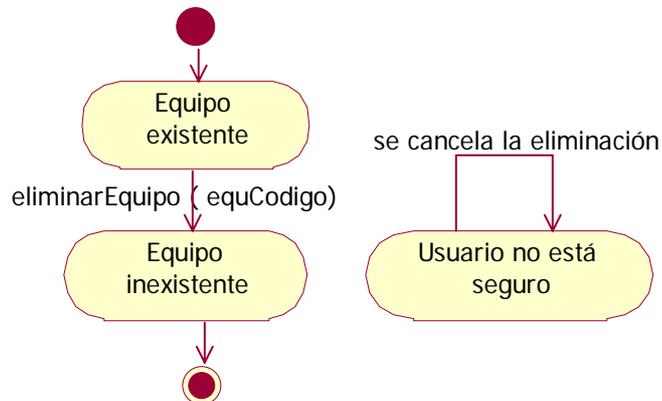
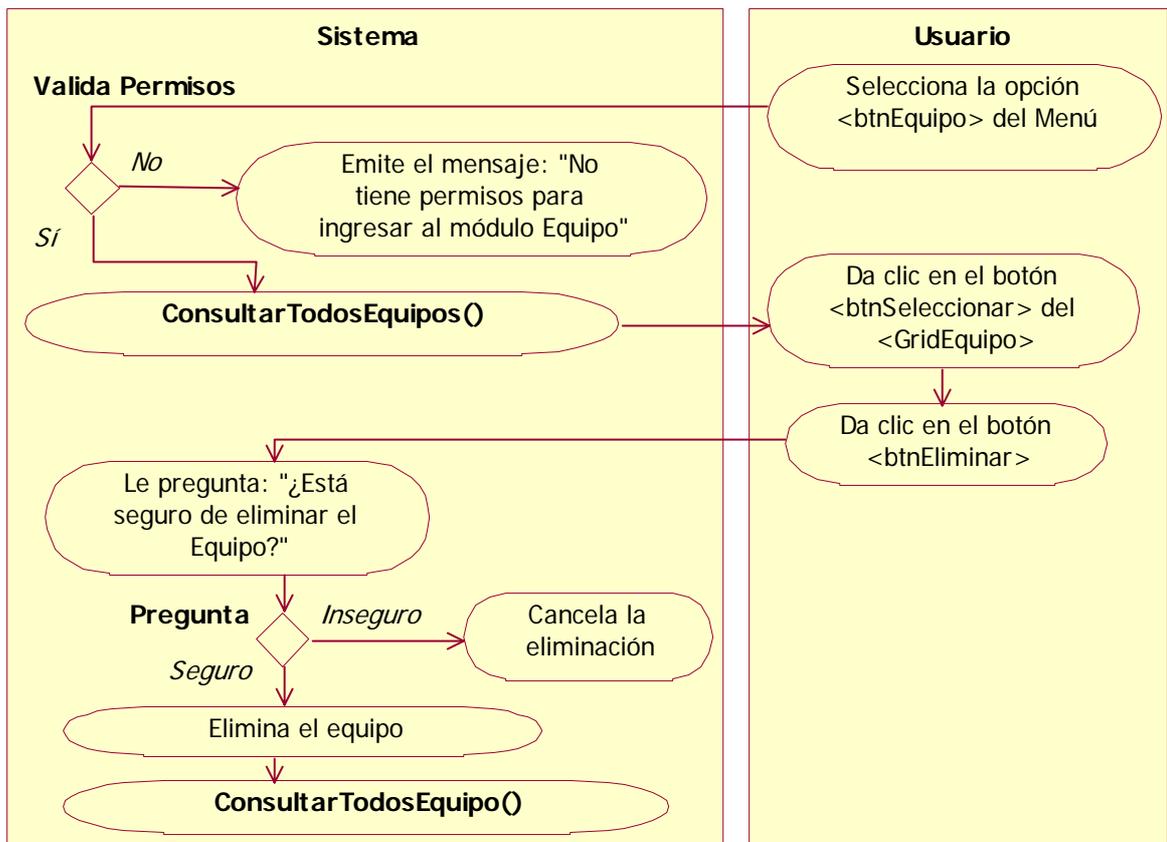


Modelo de composición de servicios – Modificar Equipo

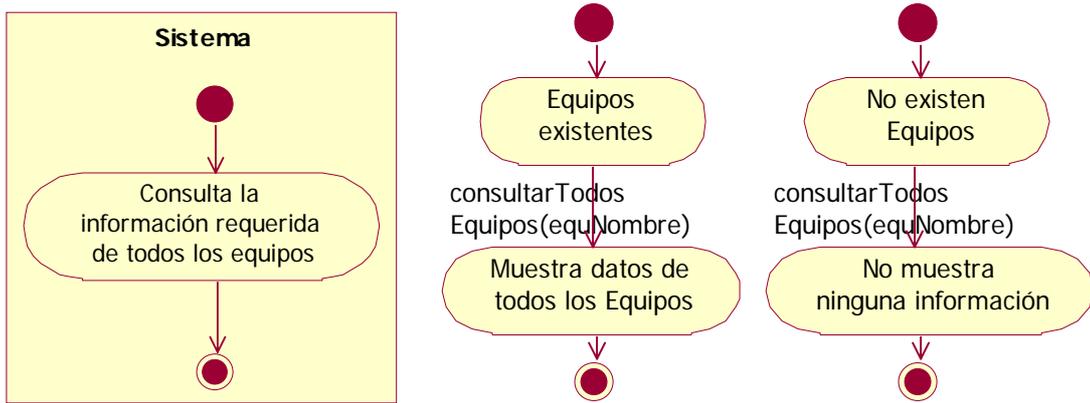




Modelo de composición de servicios – Eliminar Equipo

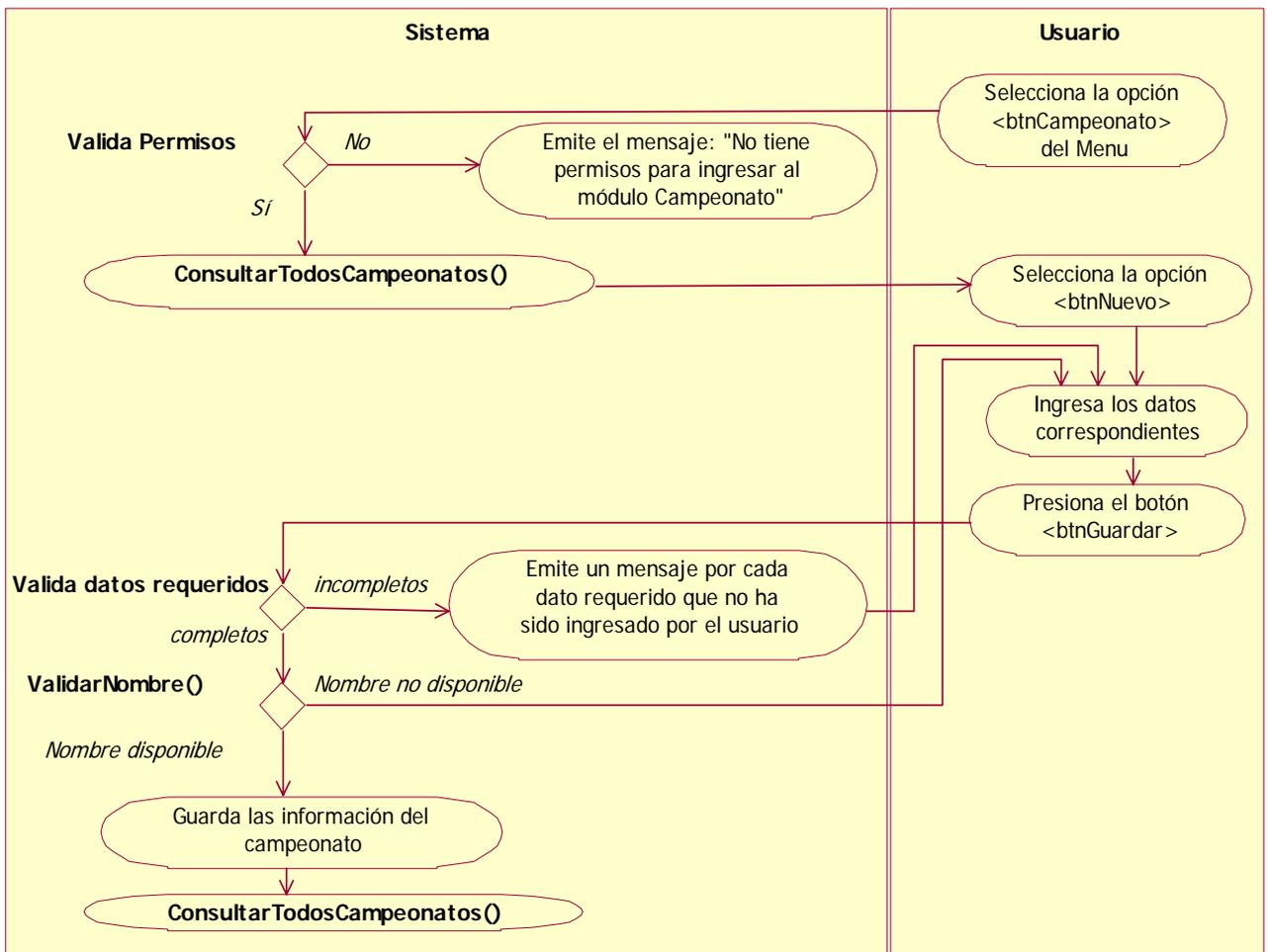


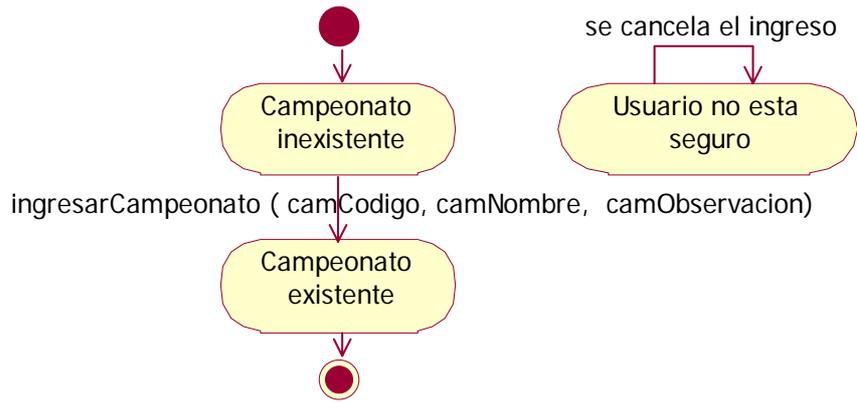
Modelo de composición de servicios – Consultar Todos los Equipos



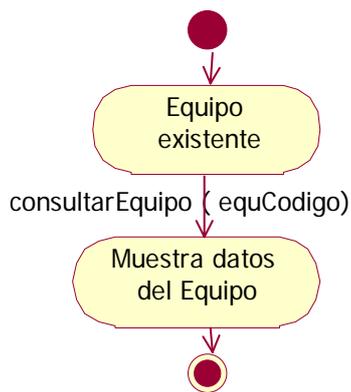
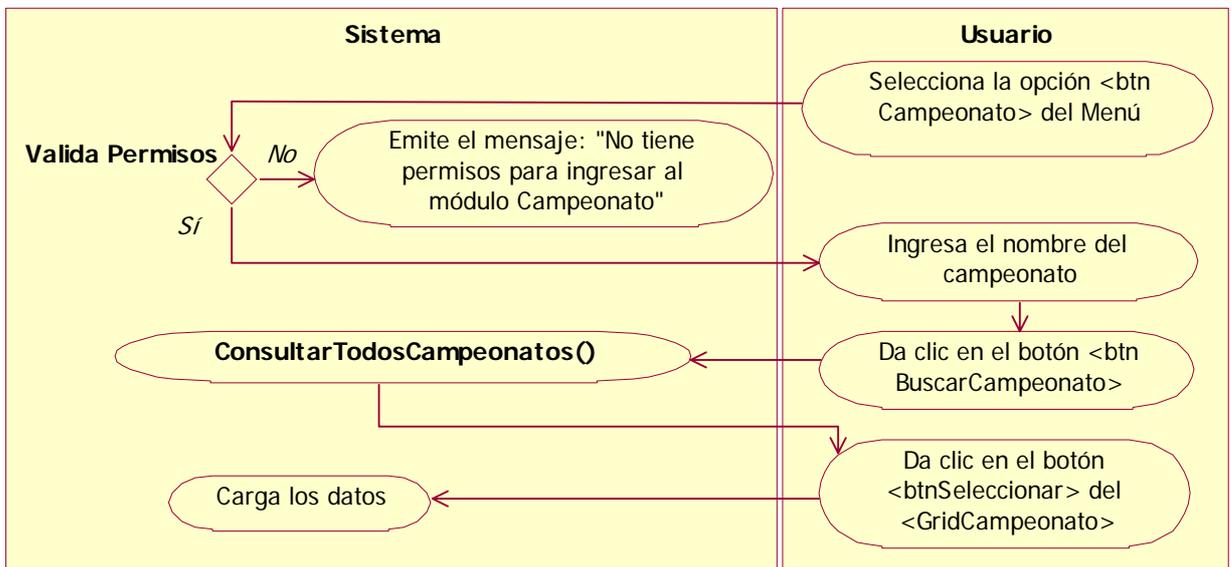
- Paquete Campeonato

Modelo de composición de servicios – Ingresar Campeonato

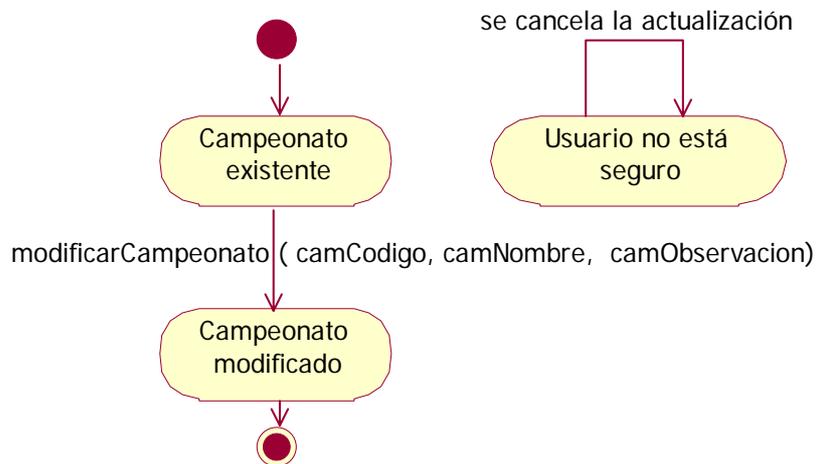
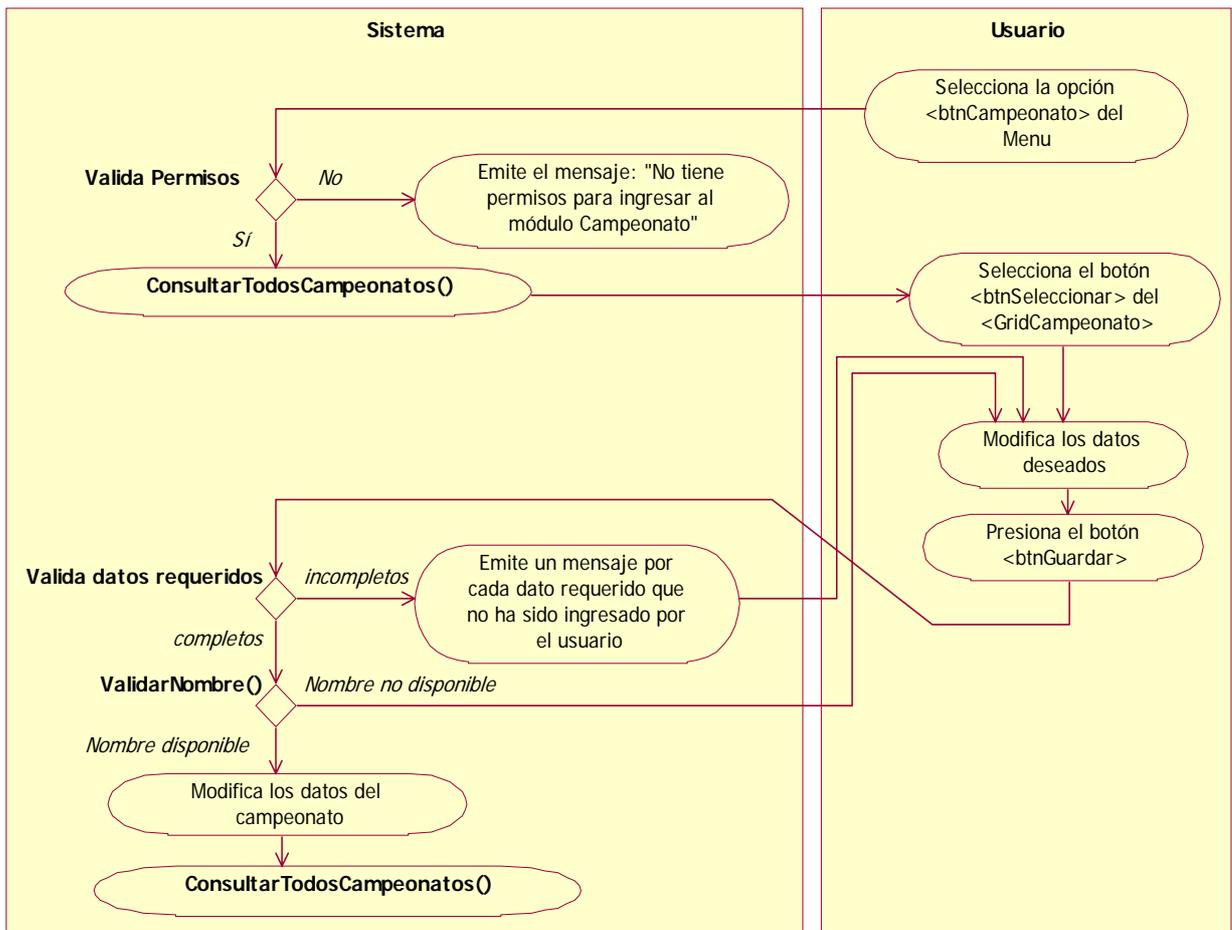




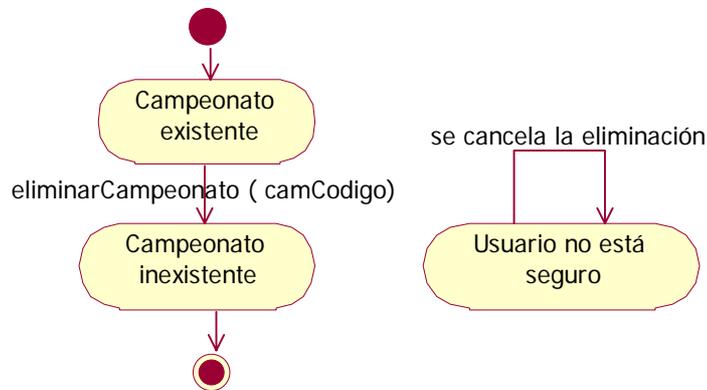
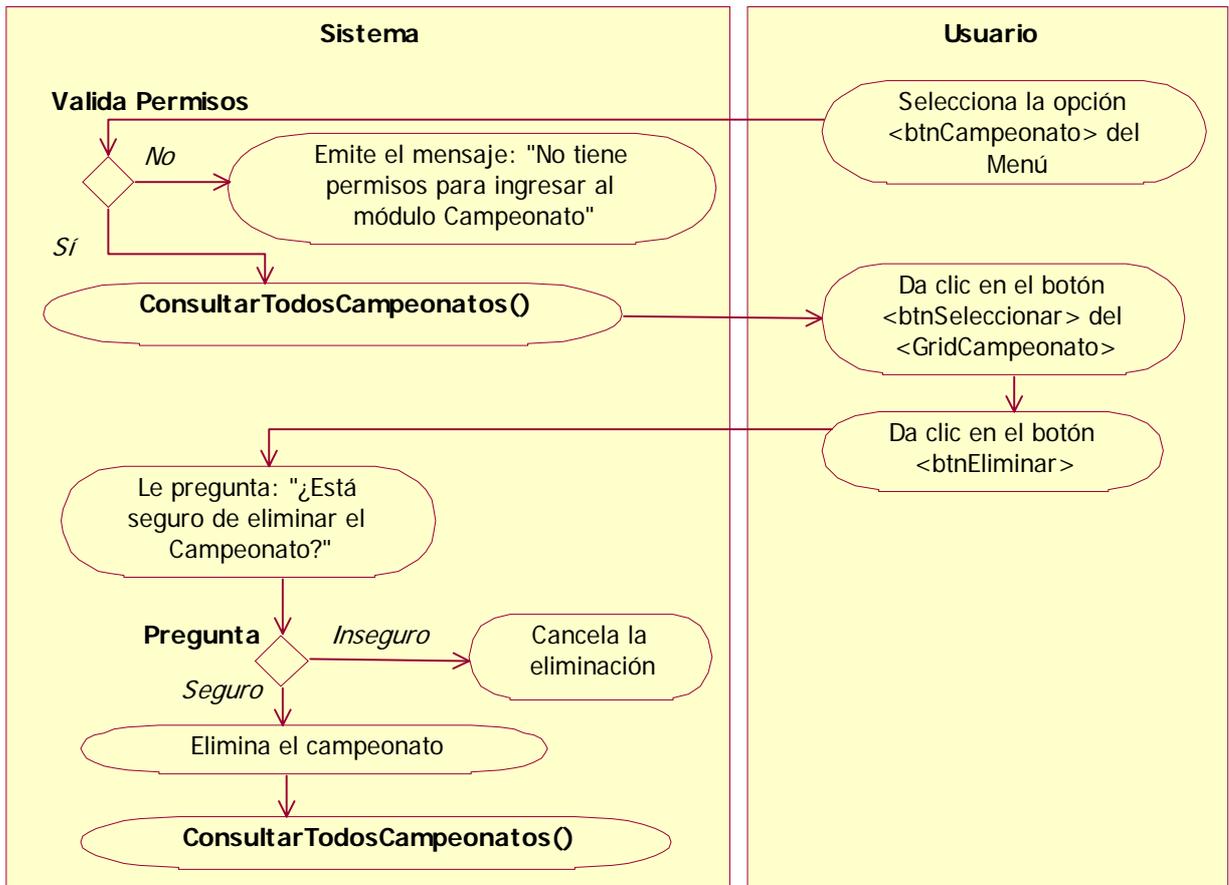
Modelo de composición de servicios – Consultar Campeonato



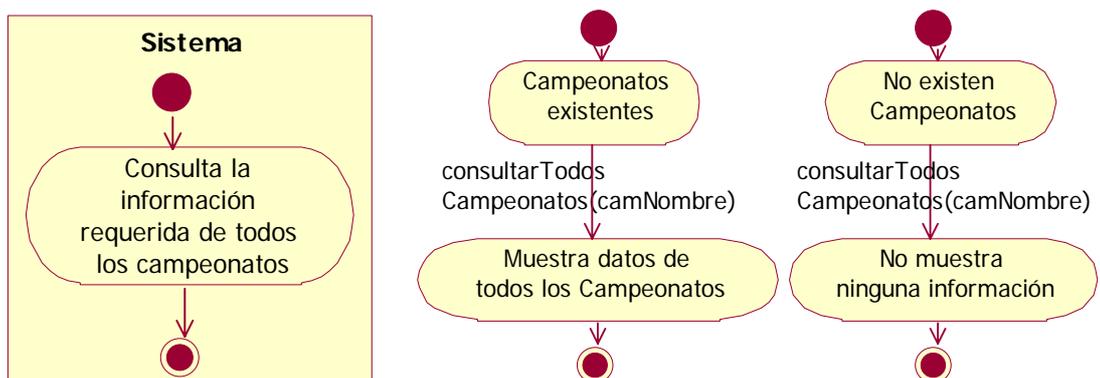
Modelo de composición de servicios – Modificar Campeonato



Modelo de composición de servicios – Eliminar Campeonato

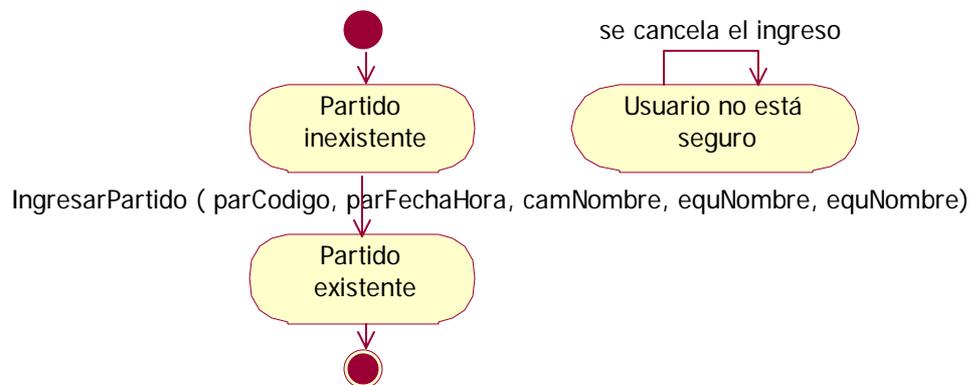
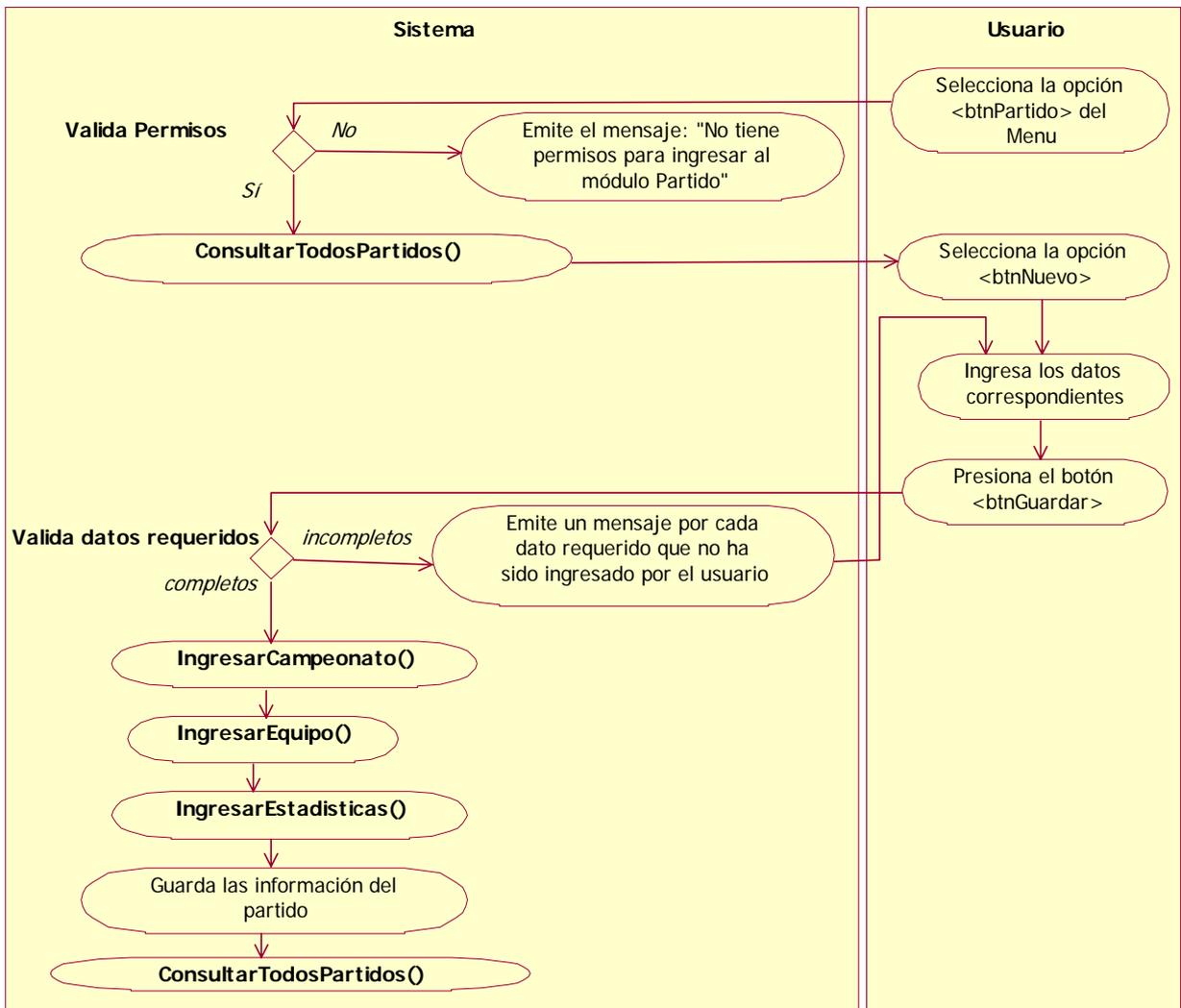


Modelo de composición de servicios –Consultar Todos Campeonatos

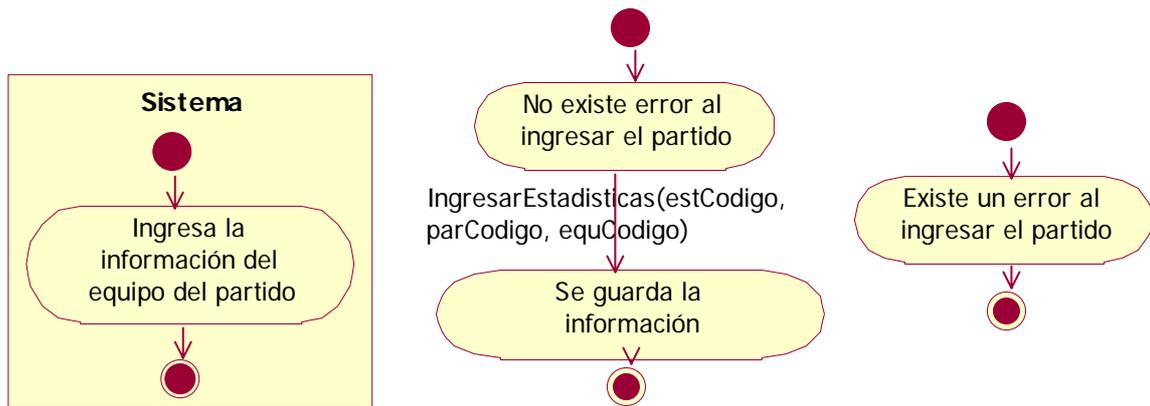


- Paquete Partido

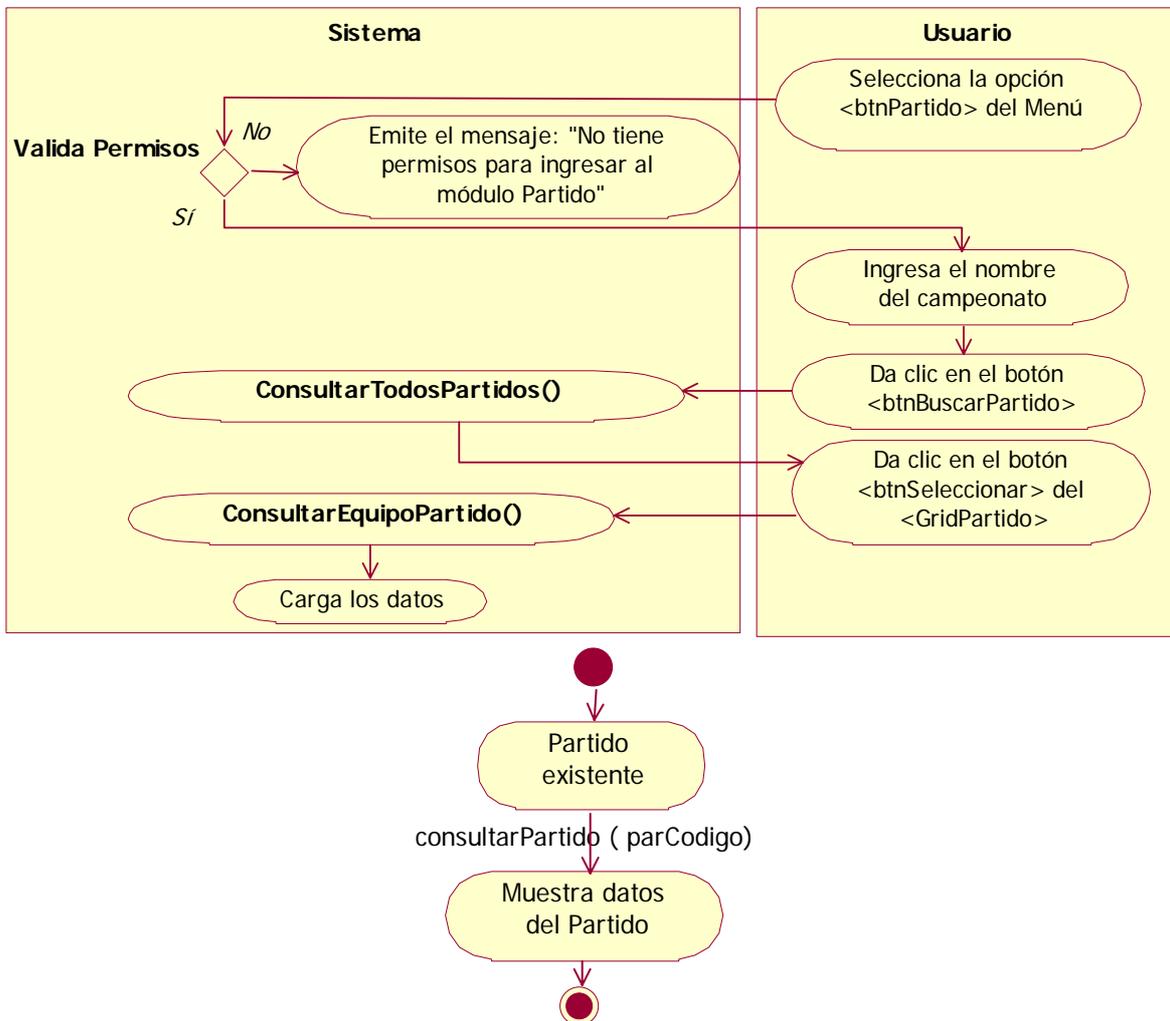
Modelo de composición de servicios –Ingresar Partido



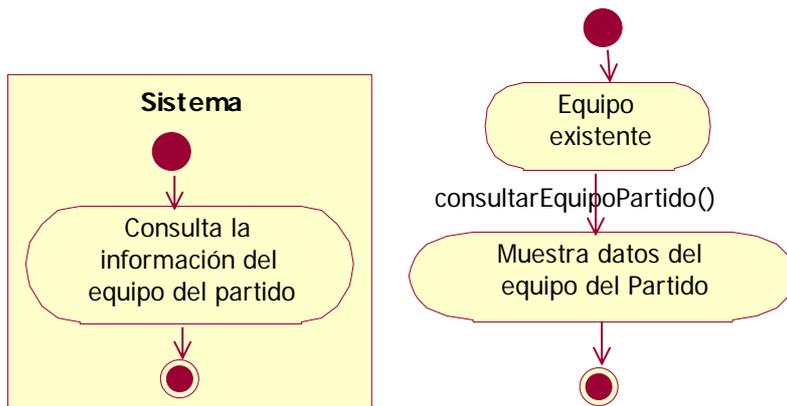
Modelo de composición de servicios –Ingresar Estadísticas



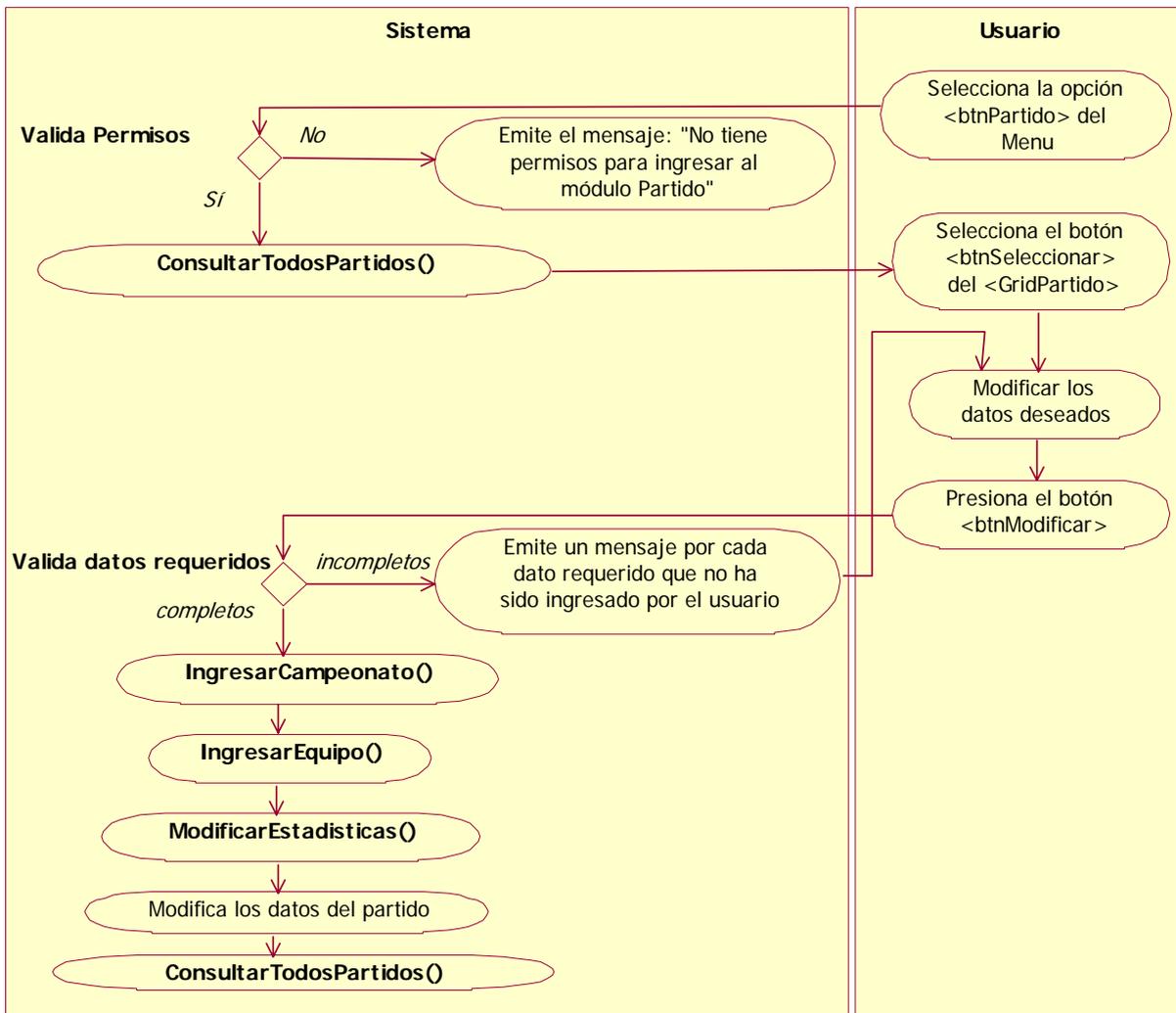
Modelo de composición de servicios –Consultar Partido

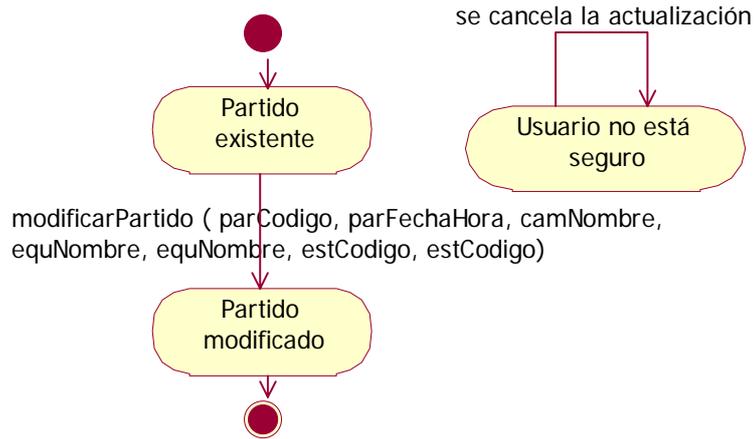


Modelo de composición de servicios –Consultar Equipo Partido

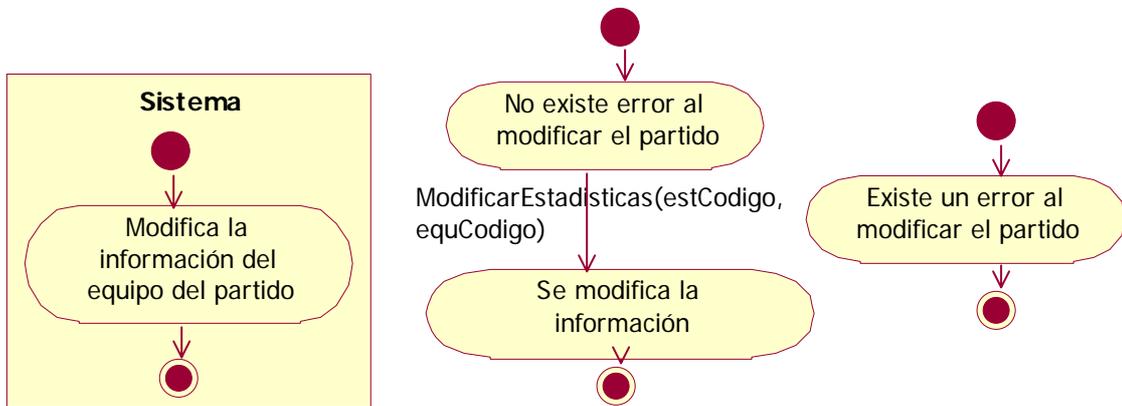


Modelo de composición de servicios –Modificar Partido

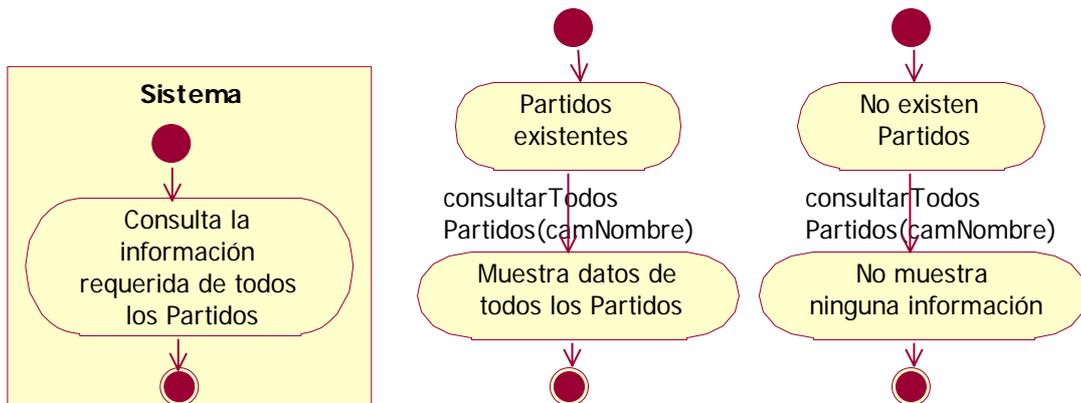




Modelo de composición de servicios –Modificar Estadísticas

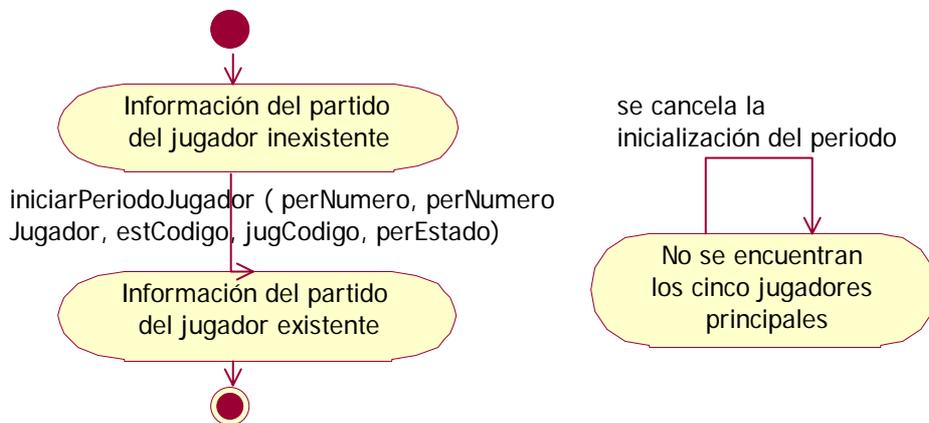
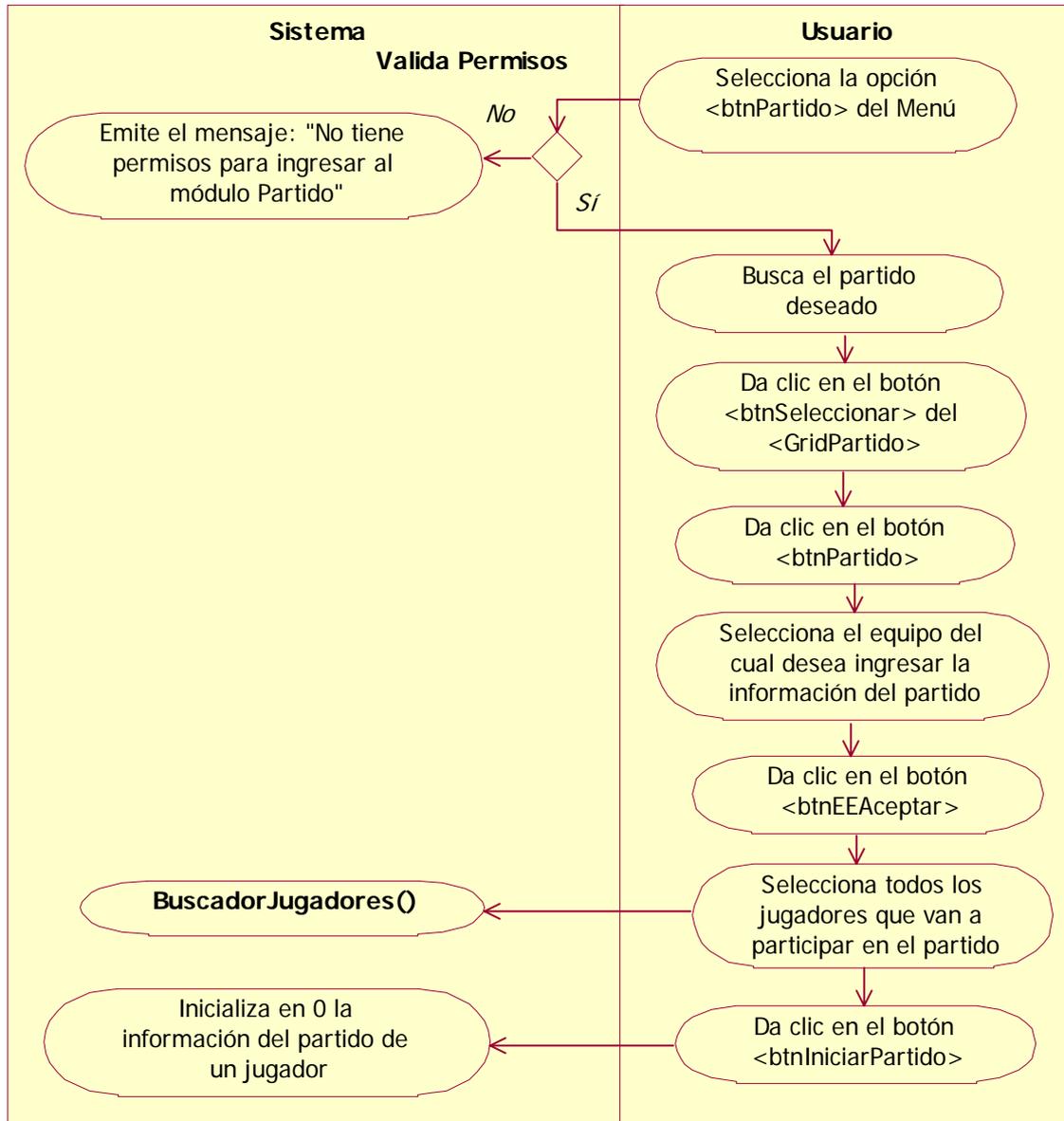


Modelo de composición de servicios –Consultar Todos Partidos

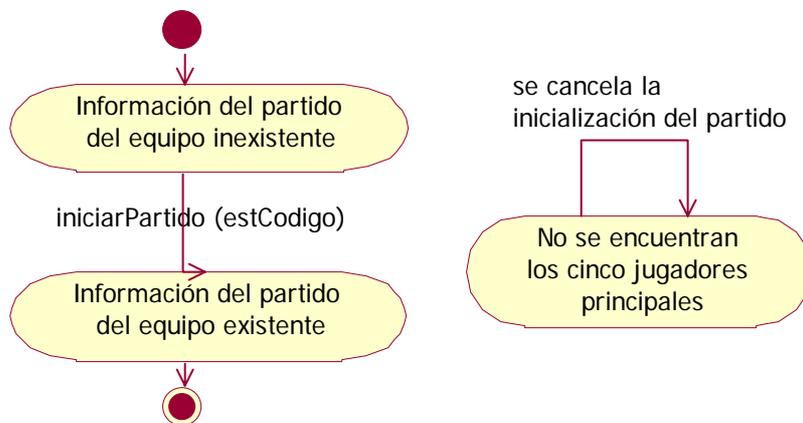
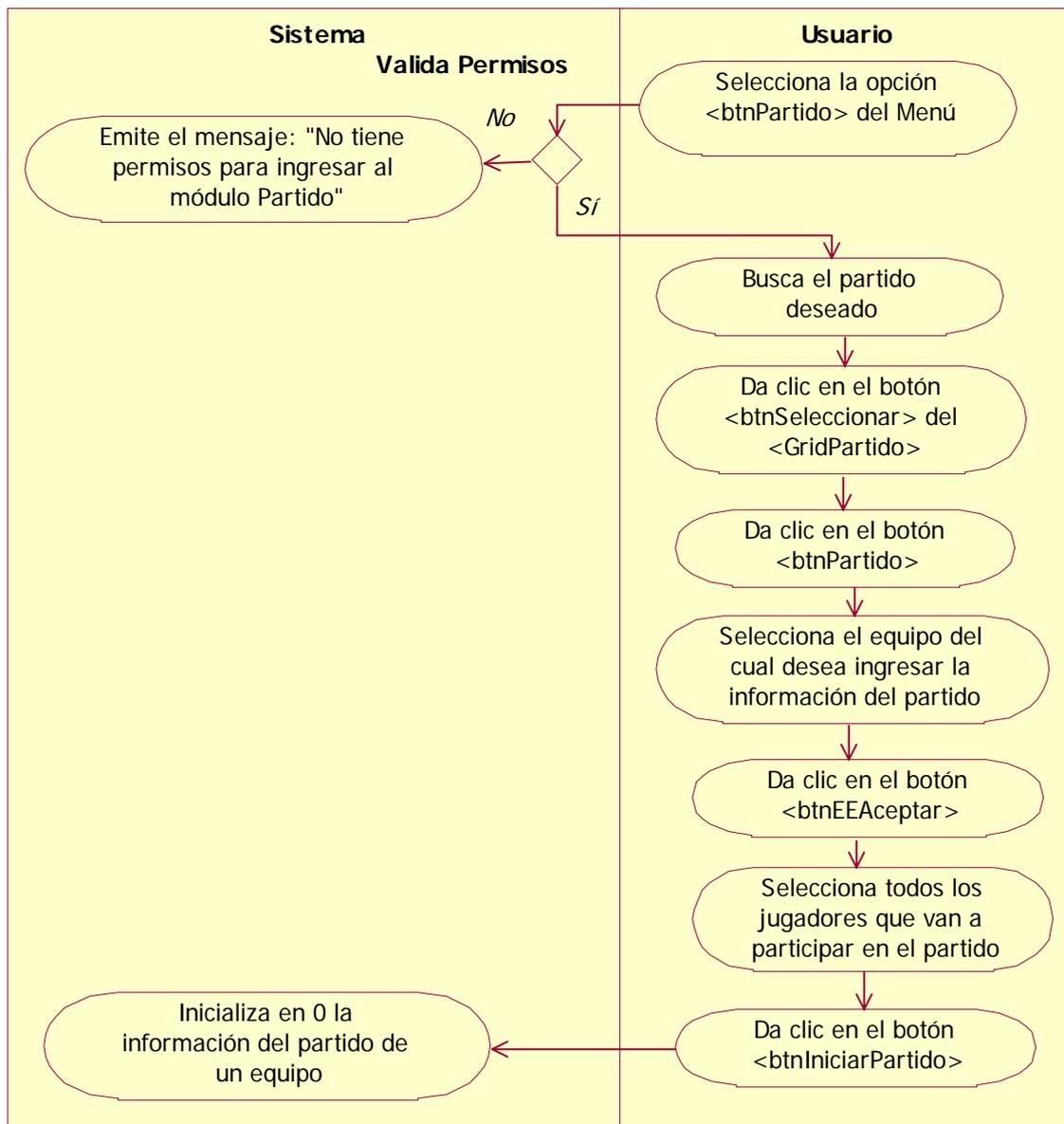


- Paquete Información del partido

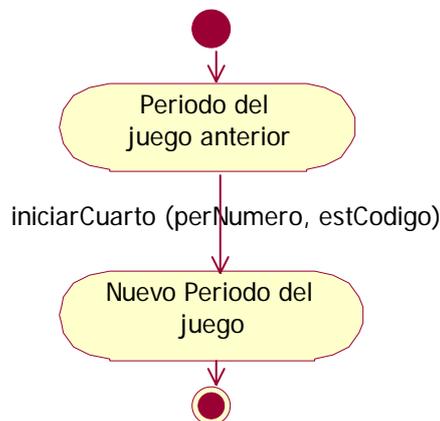
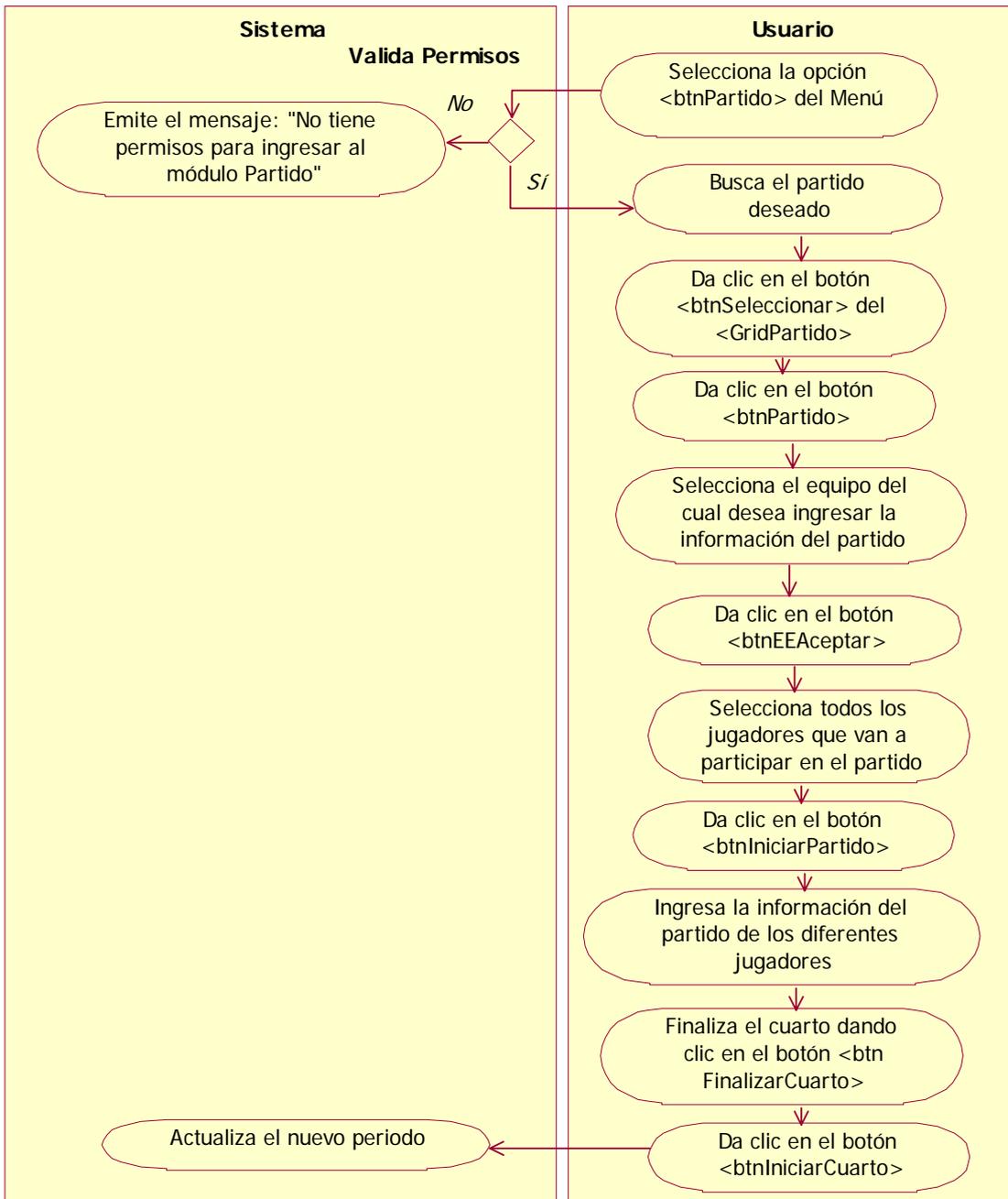
Modelo de composición de servicios – Iniciar Periodo Jugador



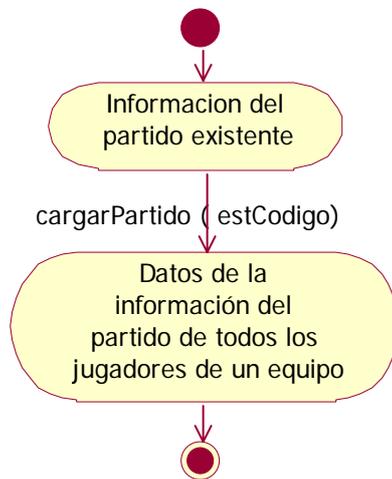
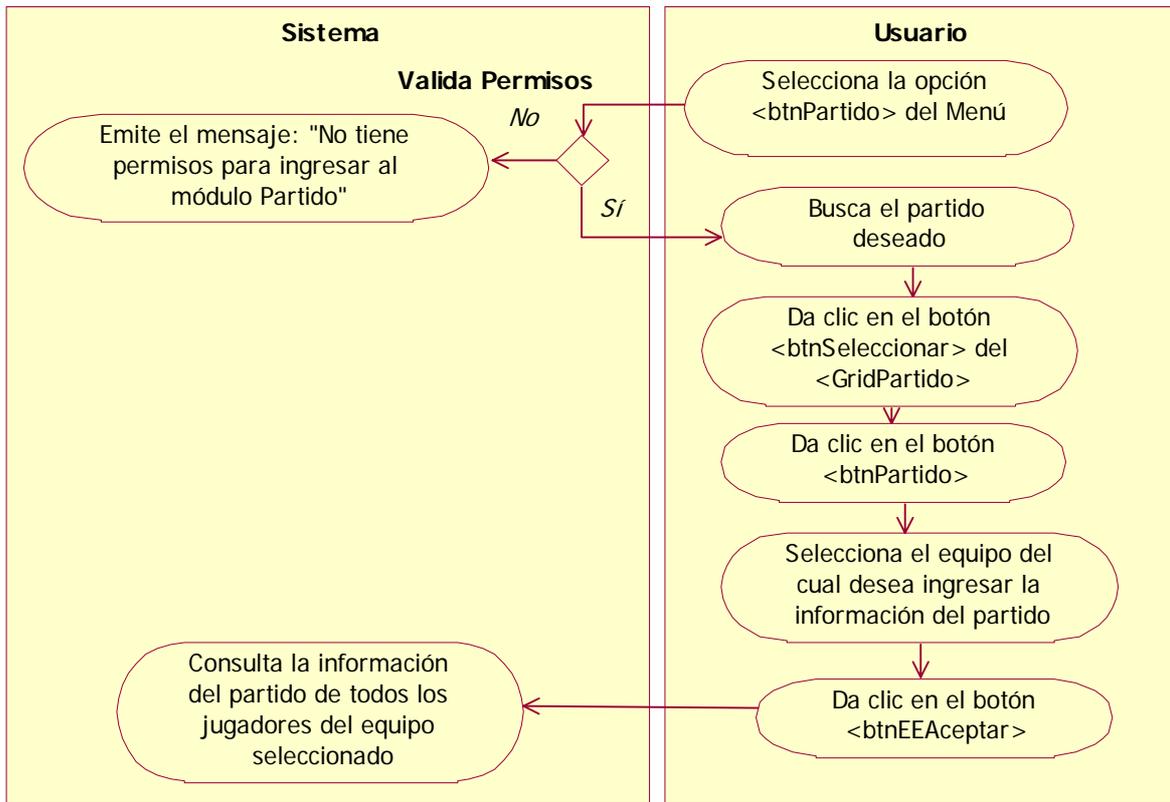
Modelo de composición de servicios – Iniciar Partido



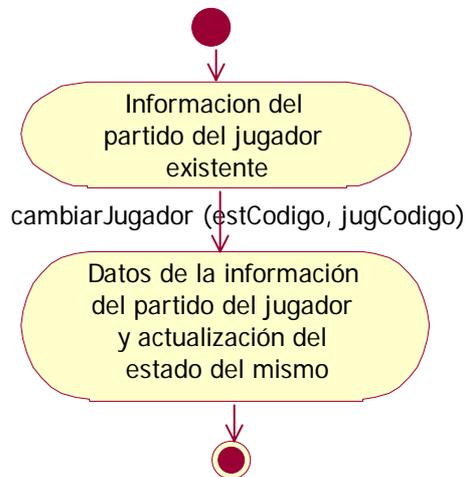
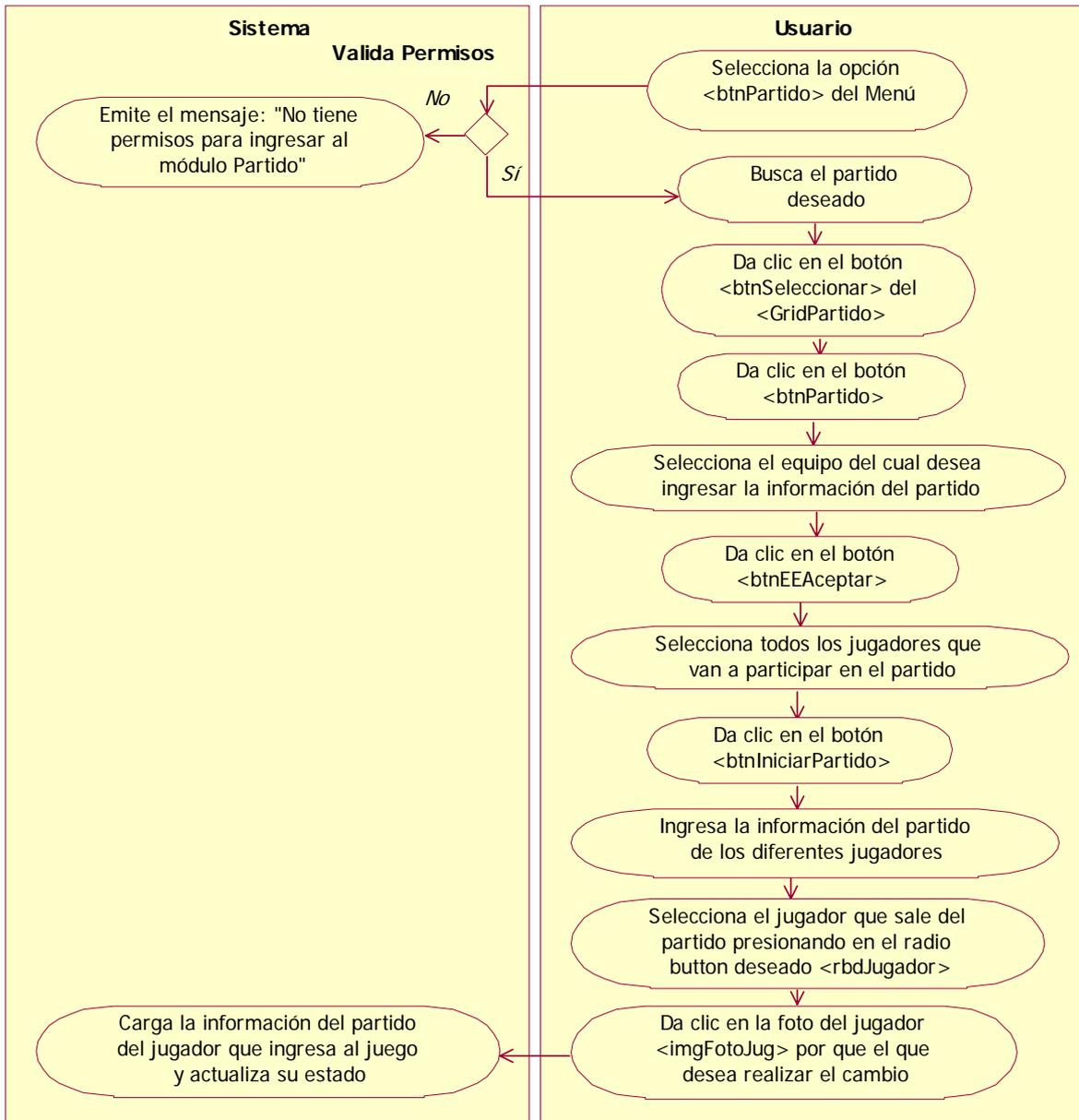
Modelo de composición de servicios – Iniciar Cuarto



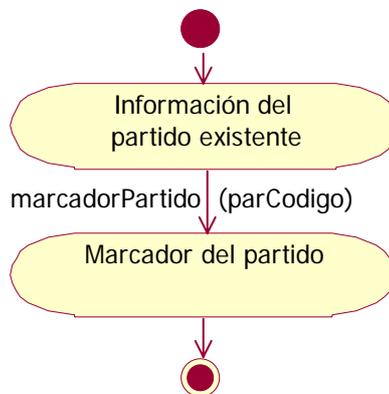
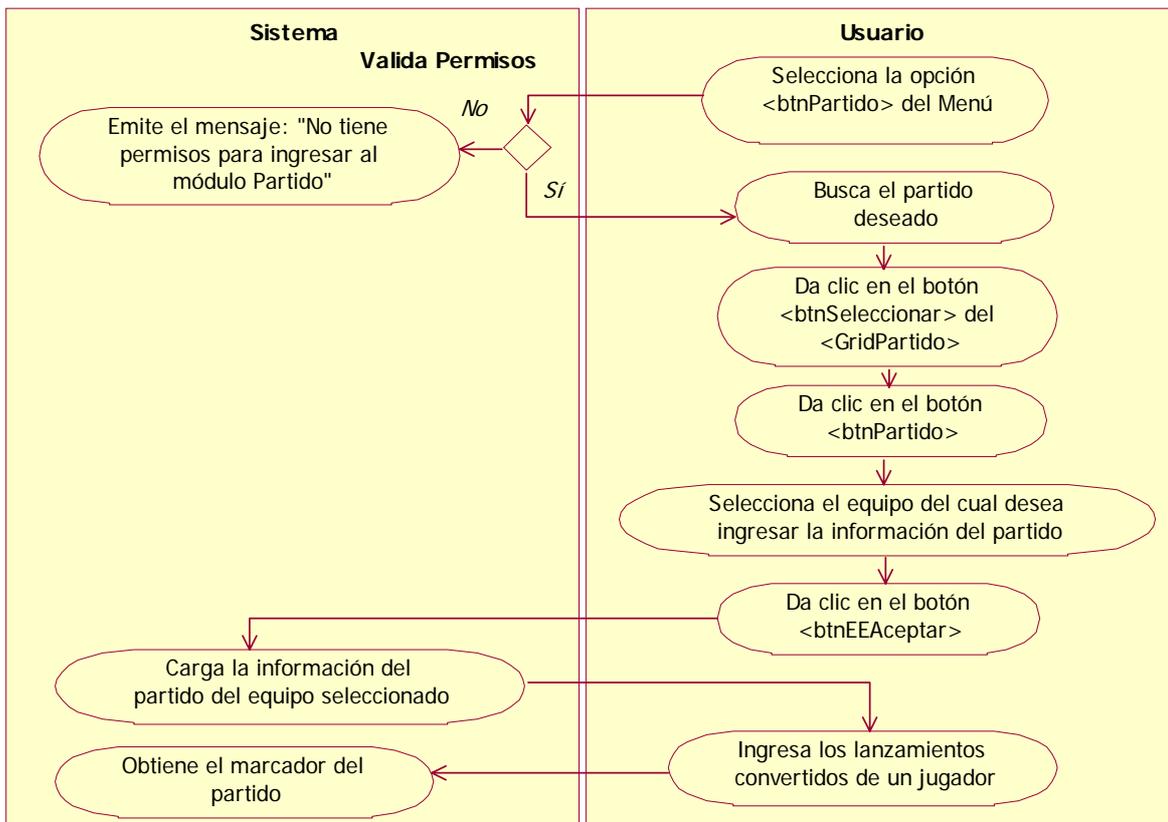
Modelo de composición de servicios – Cargar Partido



Modelo de composición de servicios – Cambiar Jugador

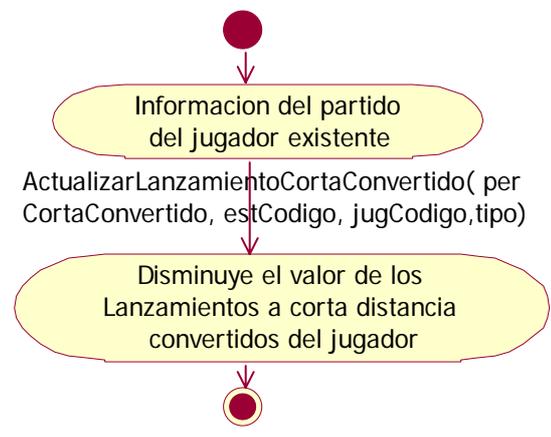
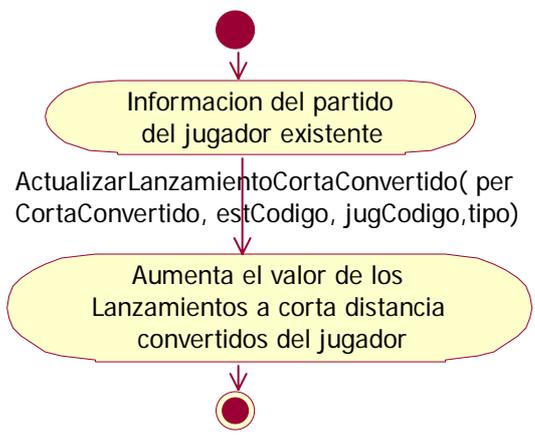
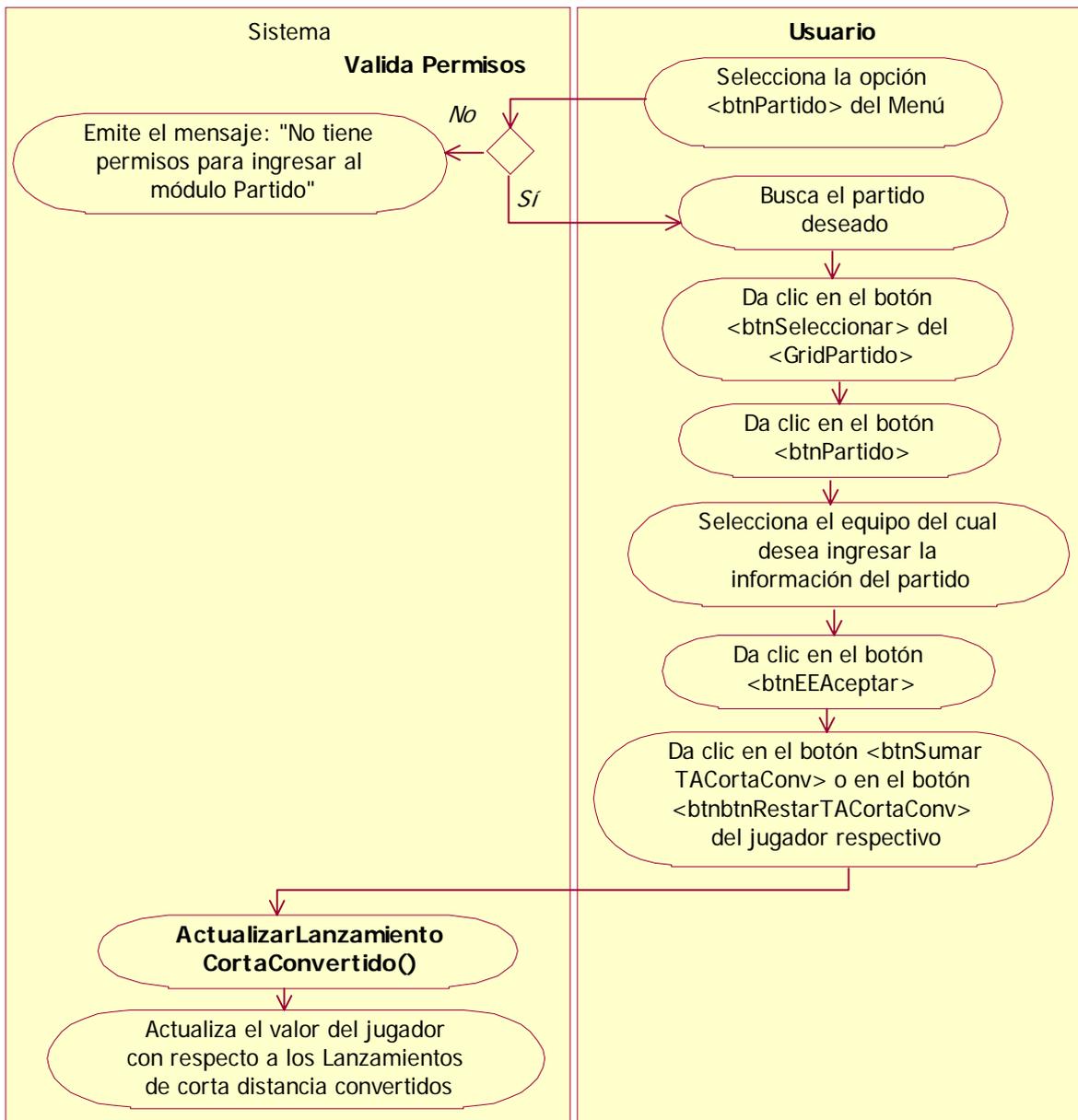


Modelo de composición de servicios – Marcador Partido



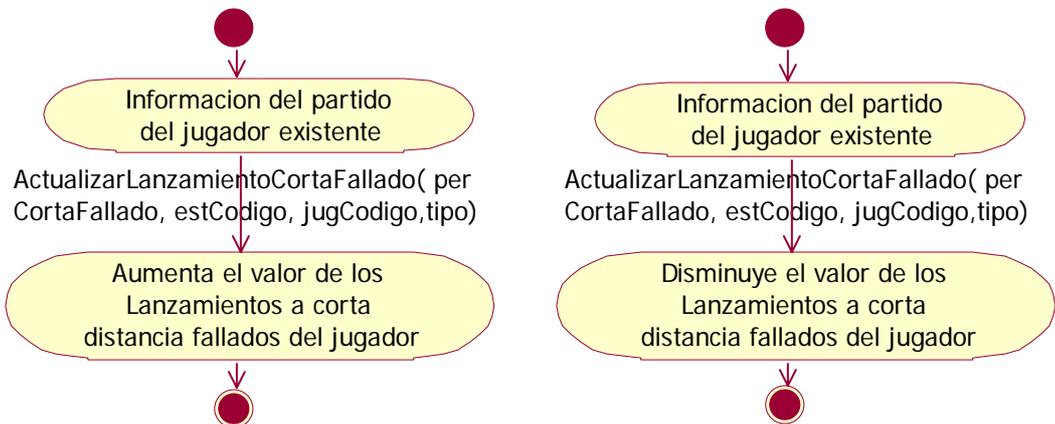
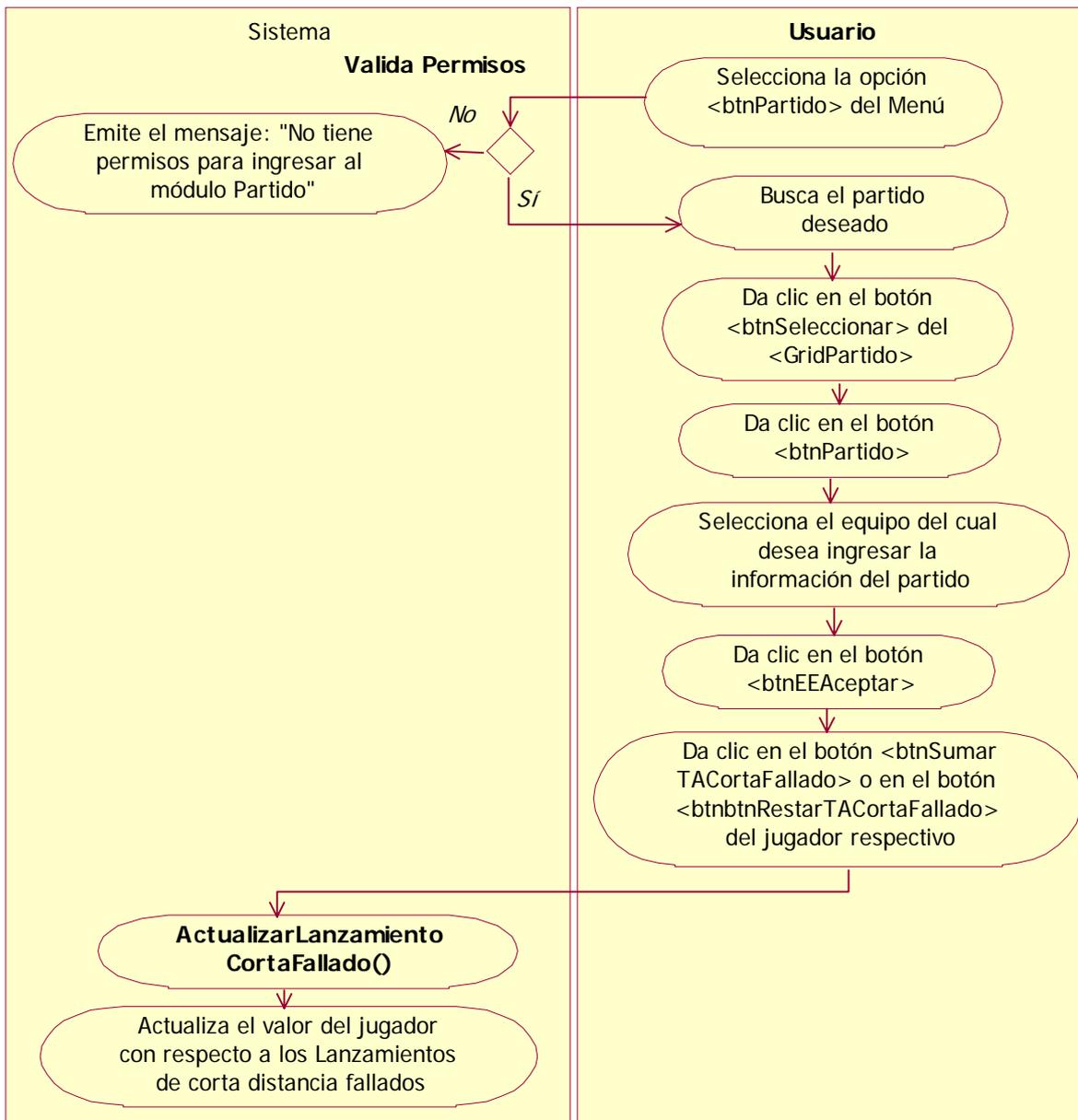
Modelo de composición de servicios –

Actualizar Lanzamiento Corta Convertido (Periodo)



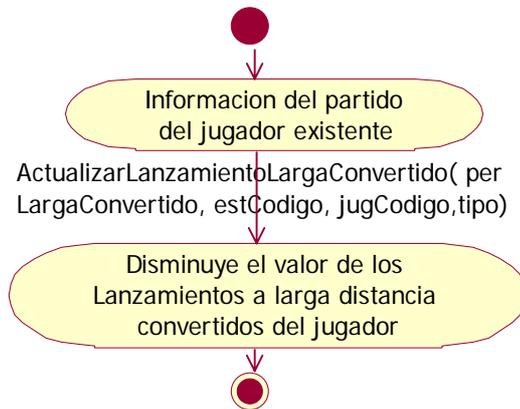
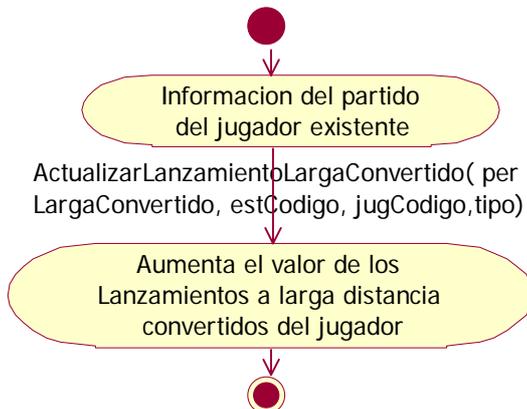
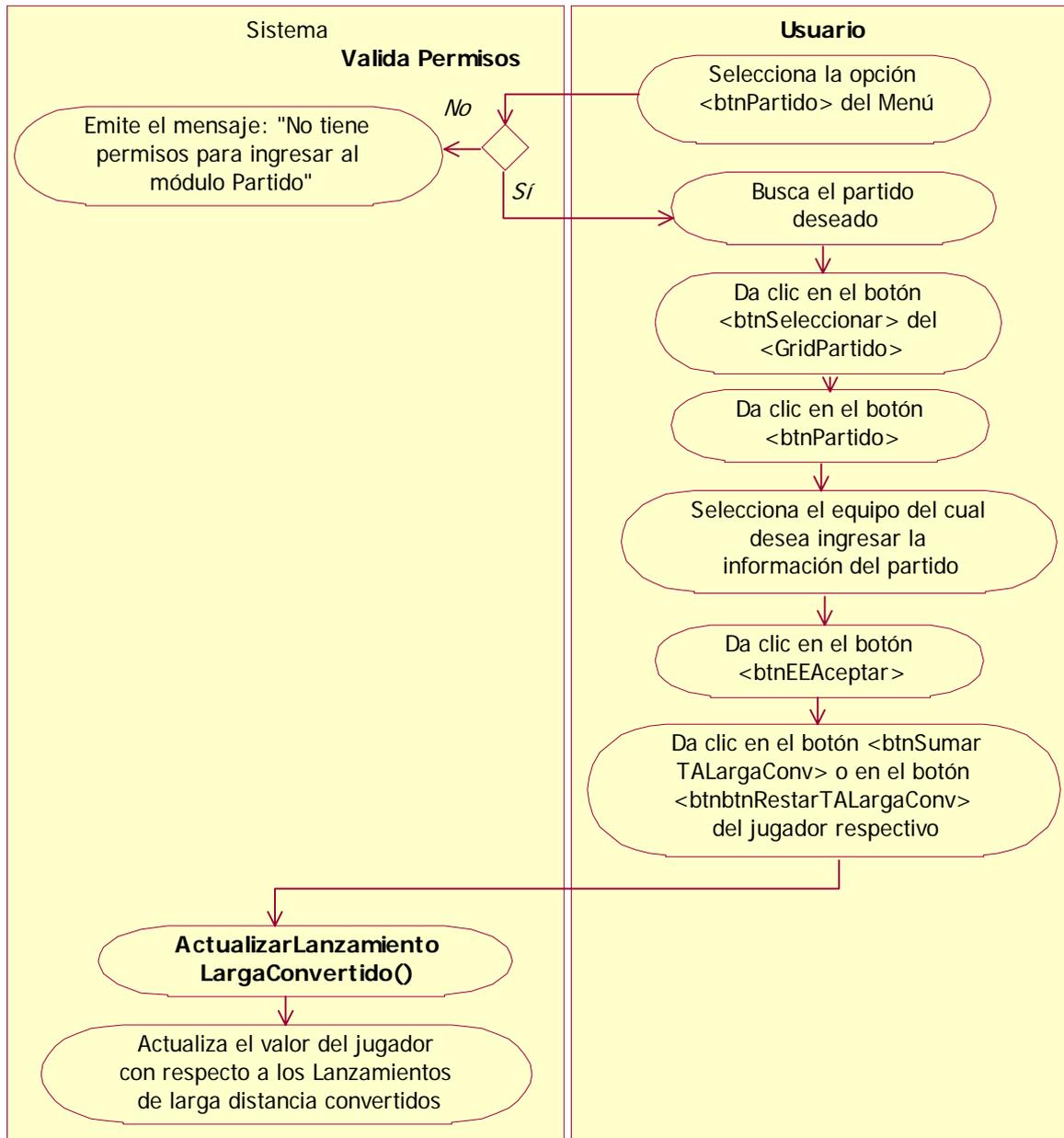
Modelo de composición de servicios –

Actualizar Lanzamiento Corta Fallado (Periodo)



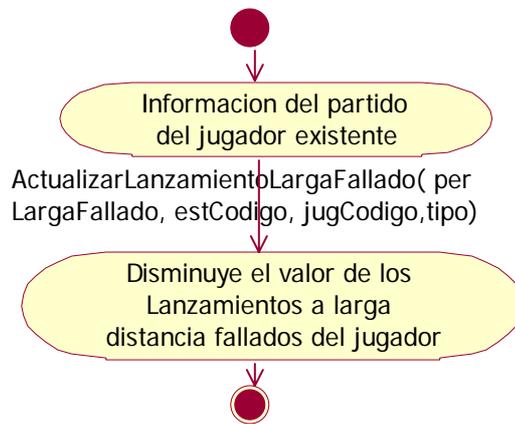
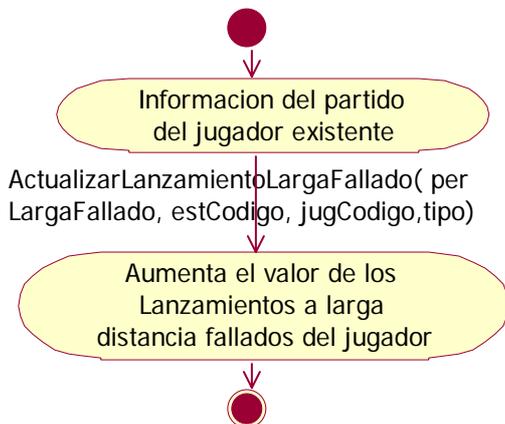
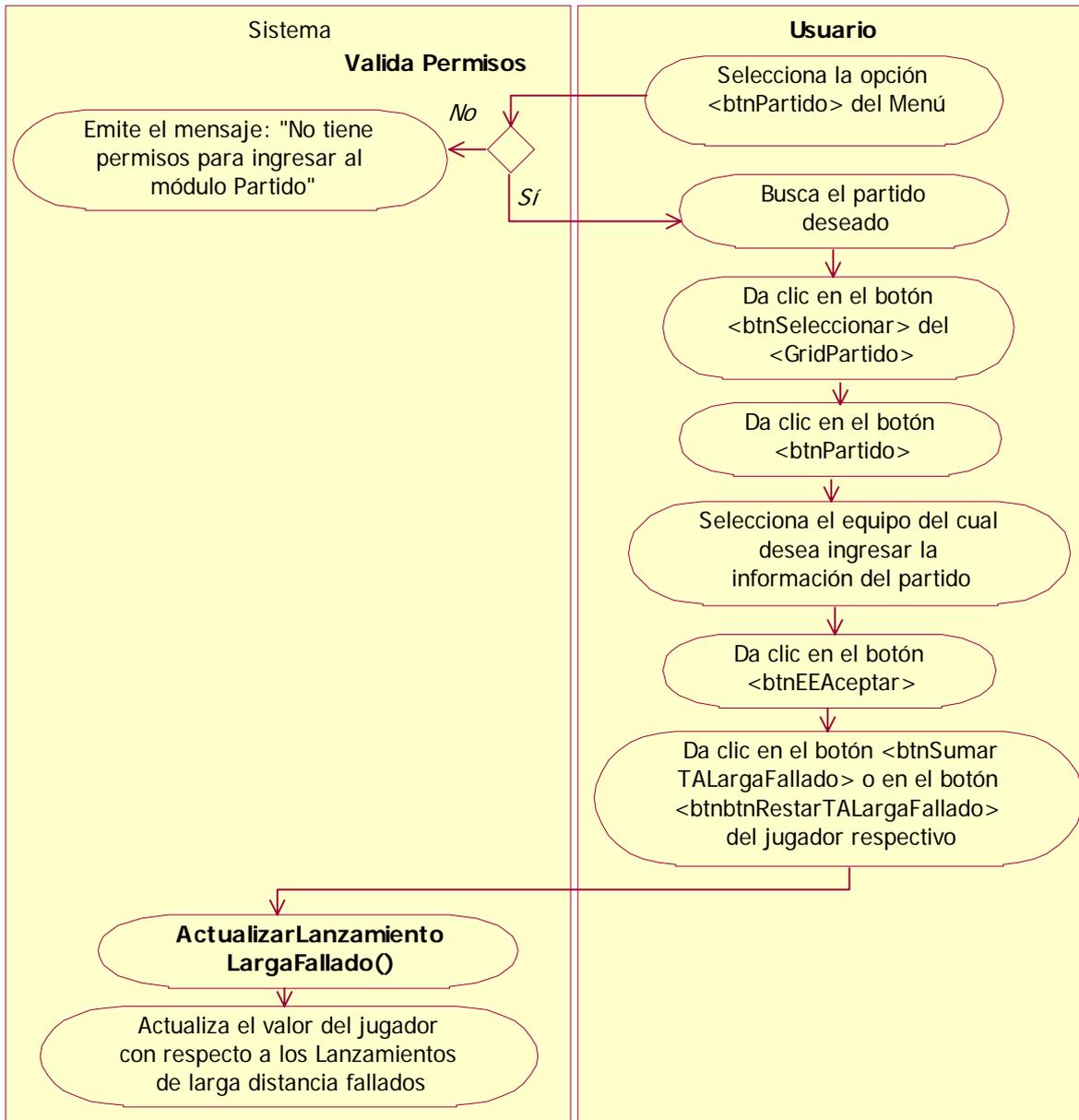
Modelo de composición de servicios –

Actualizar Lanzamiento Larga Convertido (Periodo)



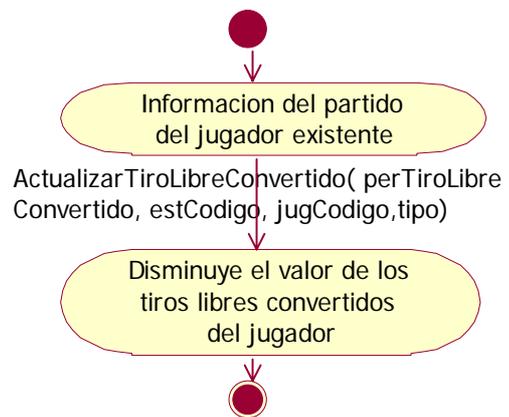
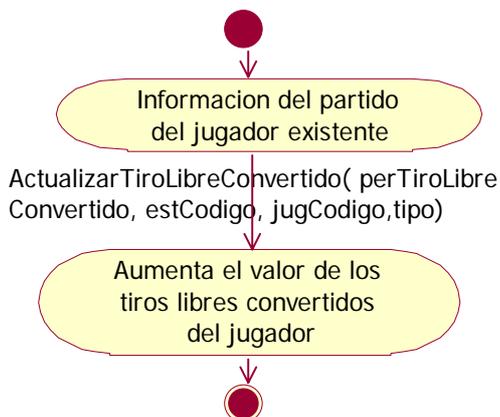
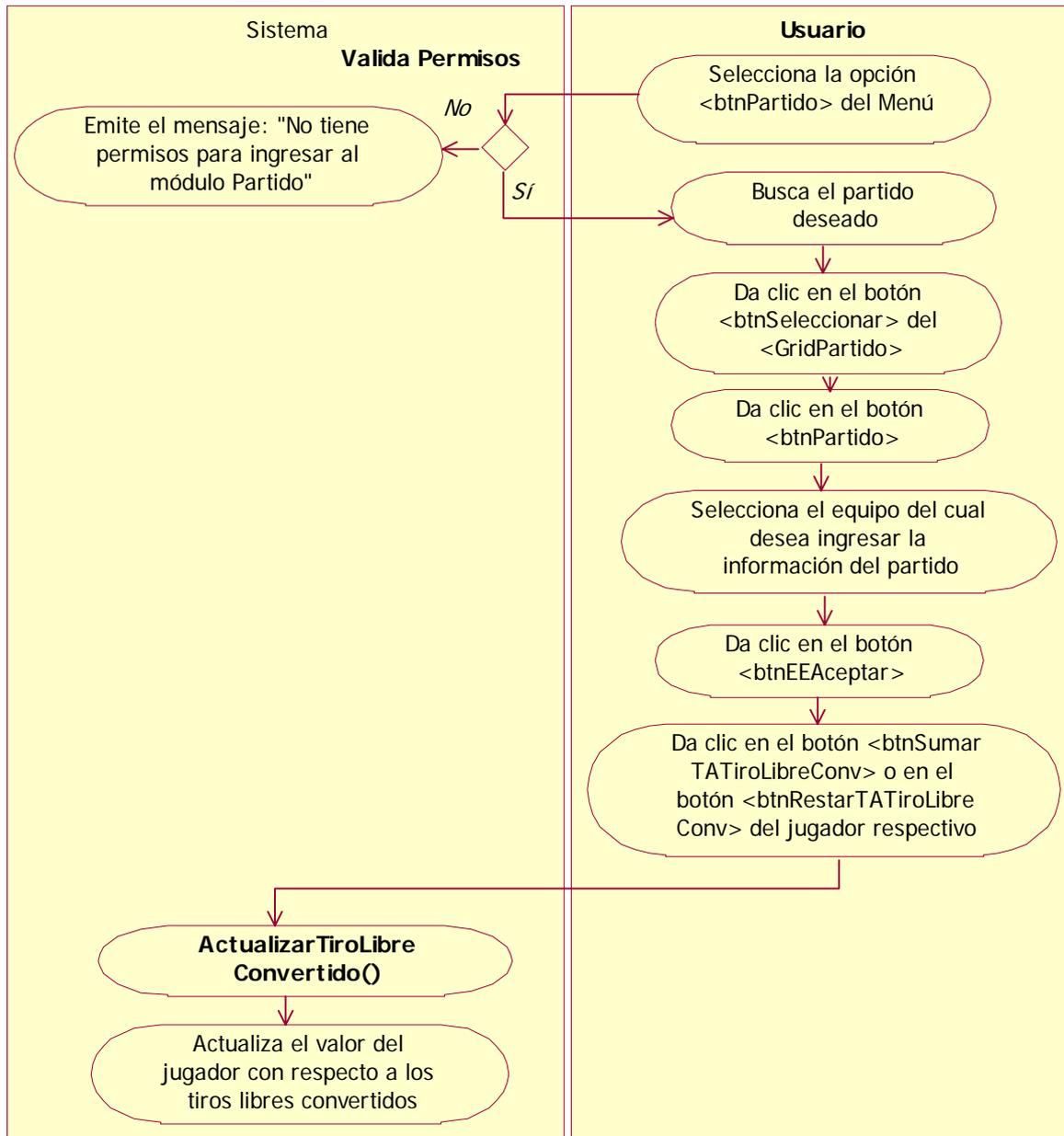
Modelo de composición de servicios –

Actualizar Lanzamiento Larga Fallado (Periodo)



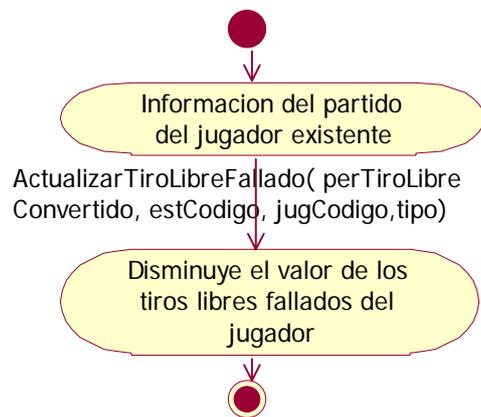
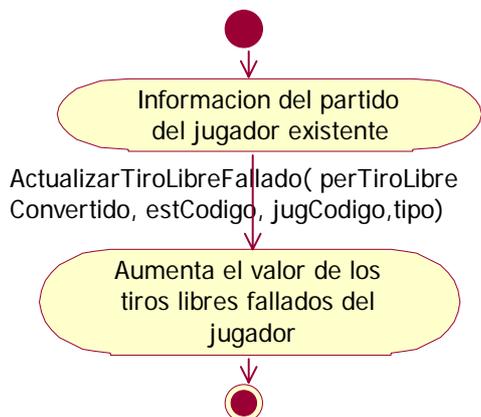
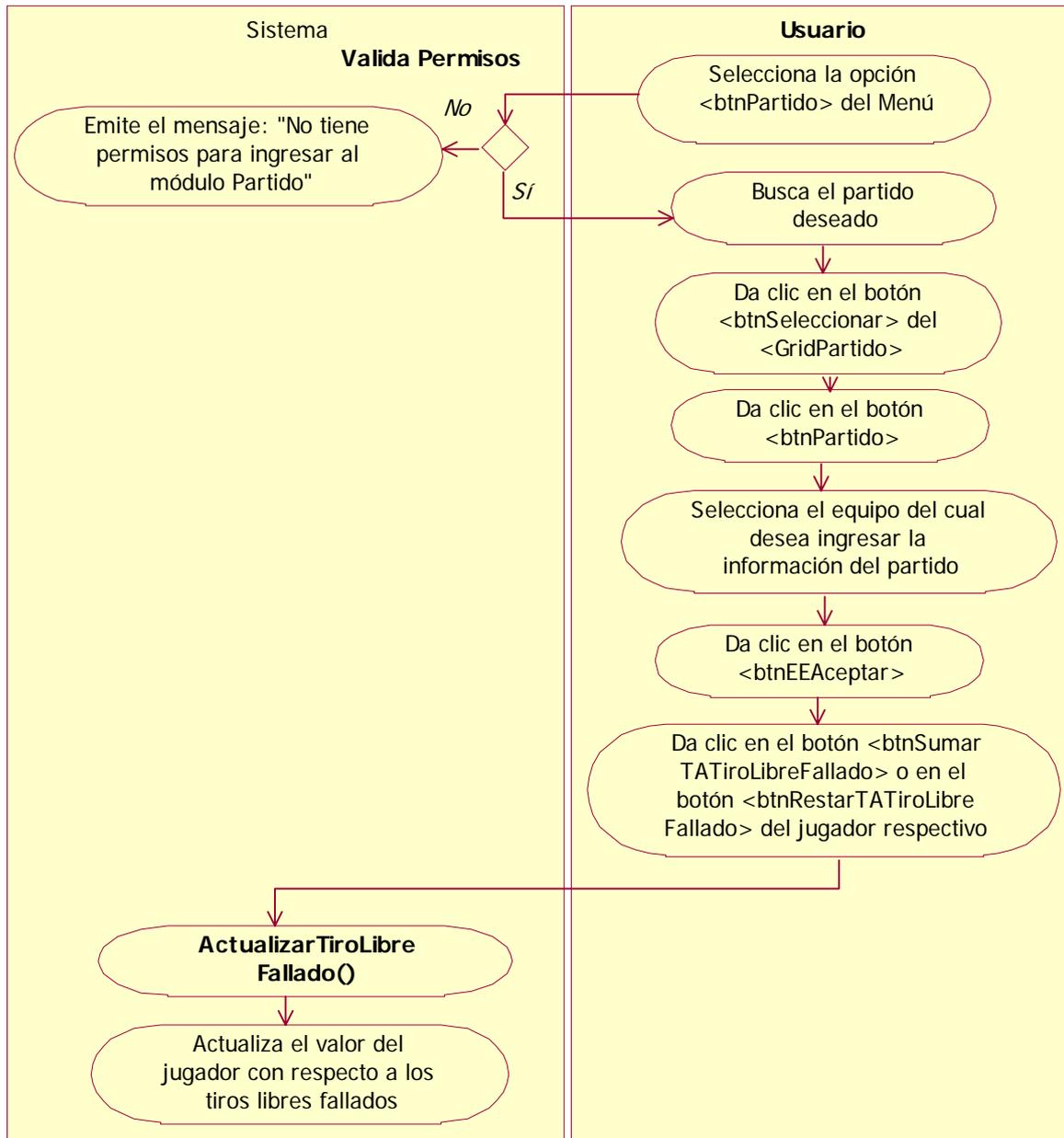
Modelo de composición de servicios –

Actualizar Tiro Libre Convertido (Periodo)



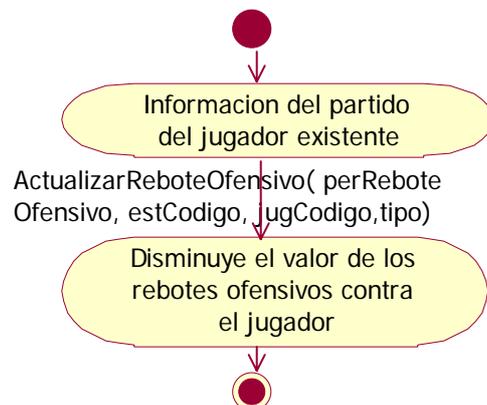
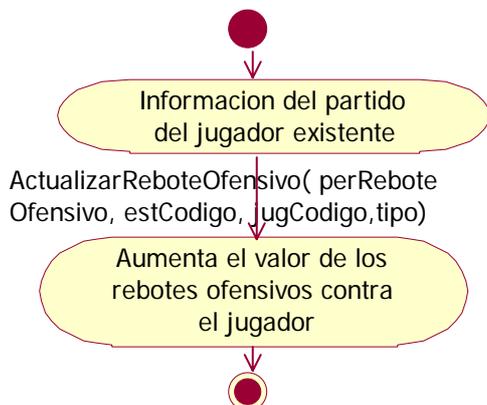
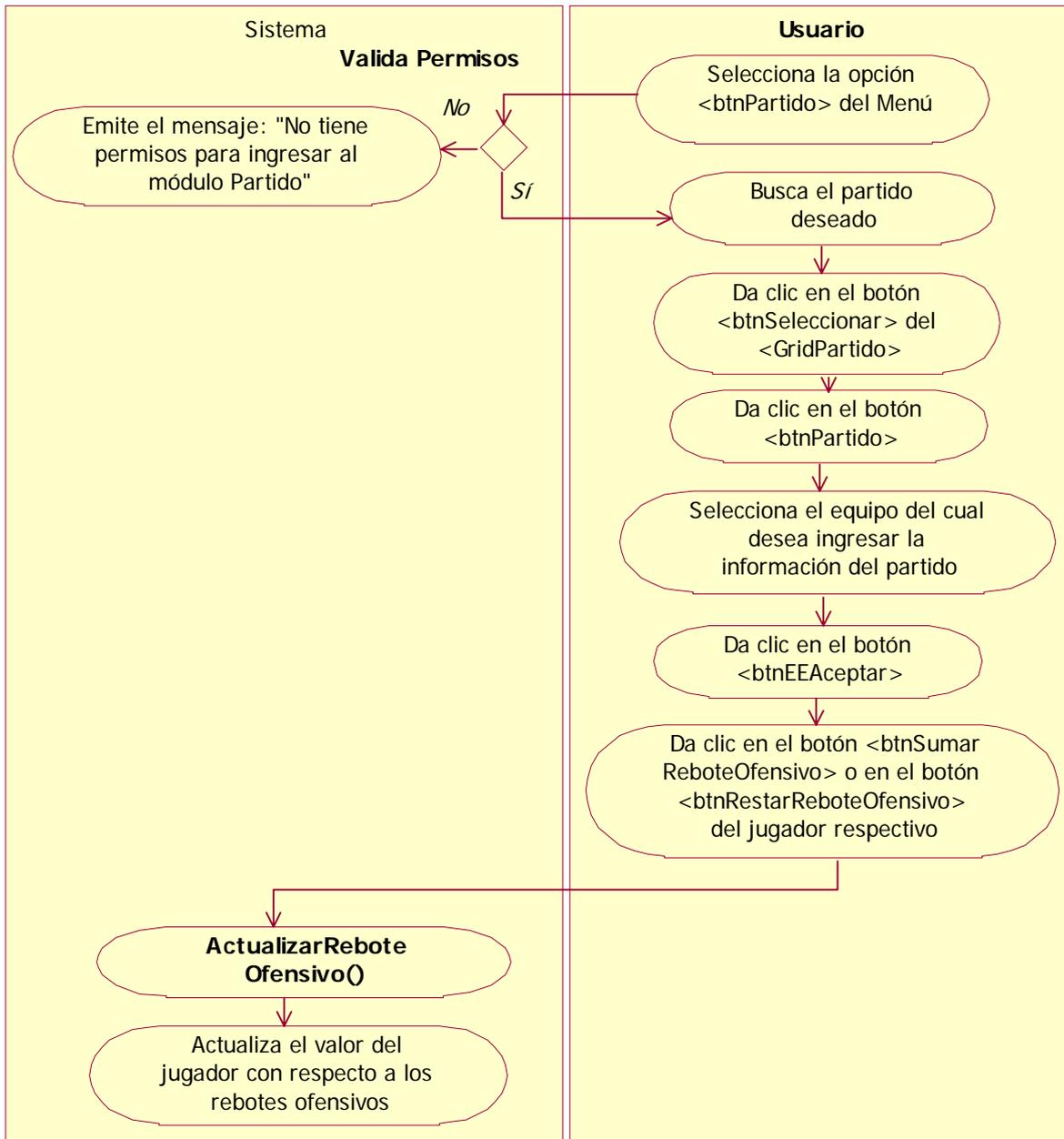
Modelo de composición de servicios –

Actualizar Tiro Libre Fallado (Periodo)



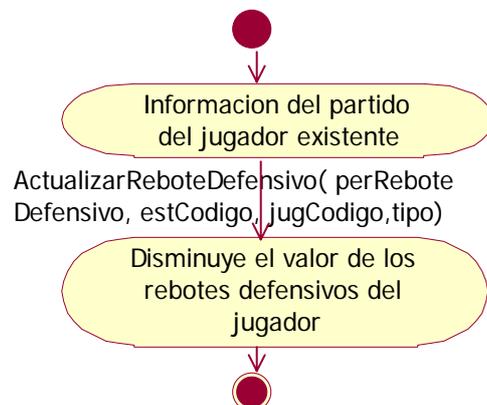
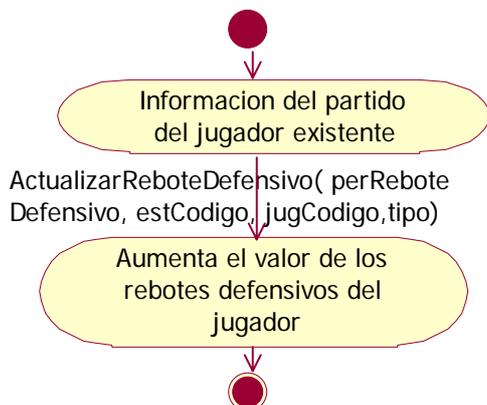
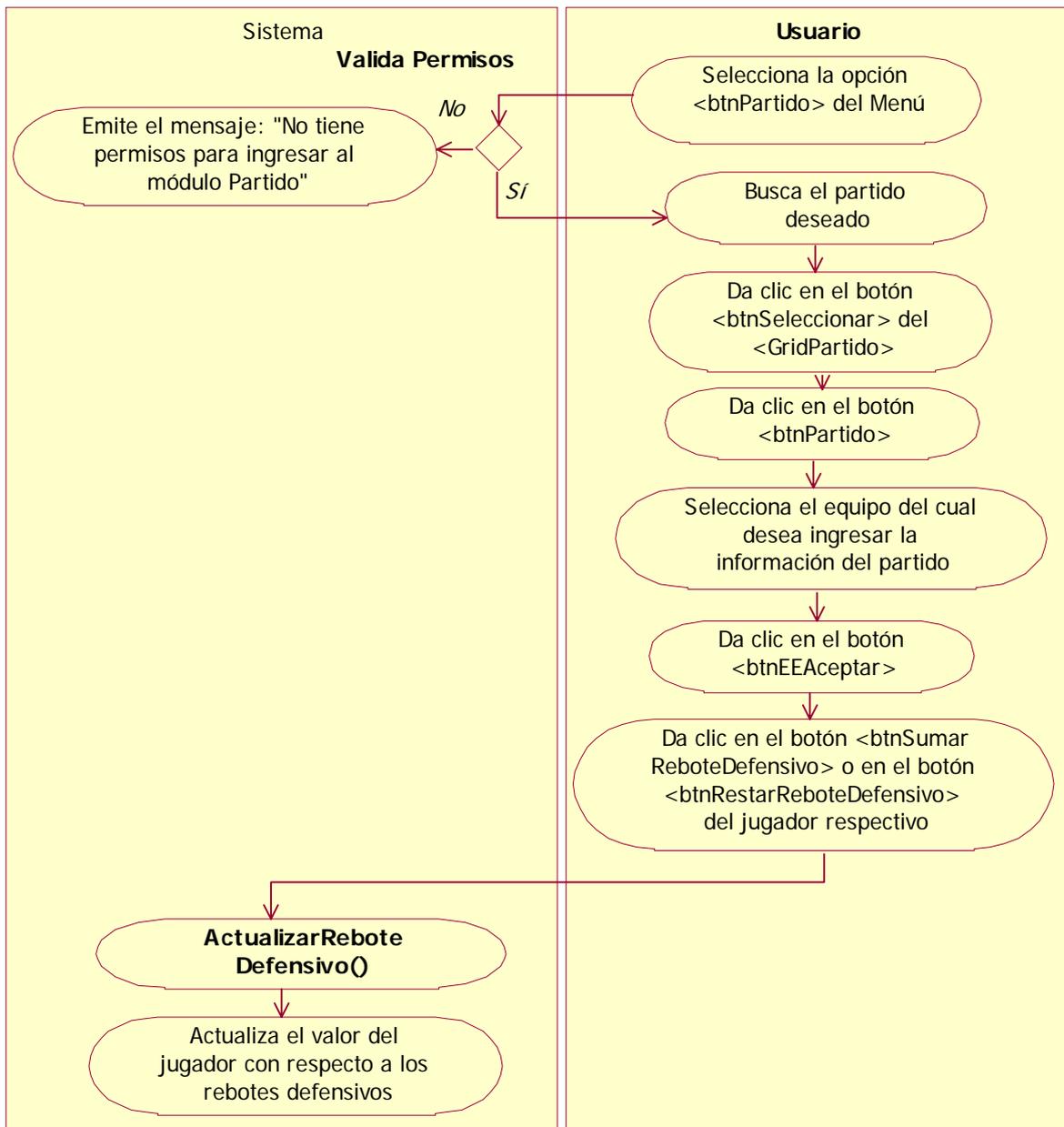
Modelo de composición de servicios –

Actualizar Rebote Ofensivo (Periodo)



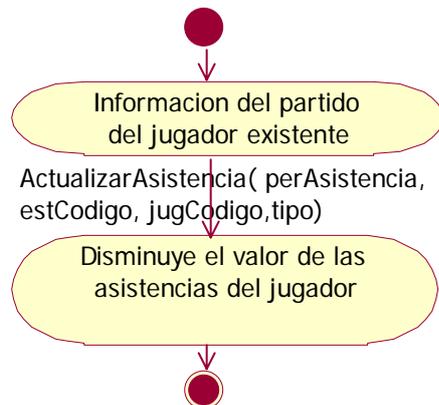
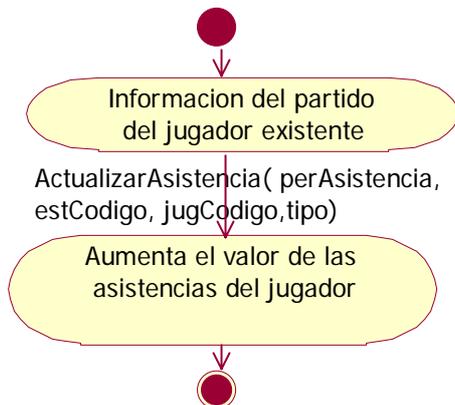
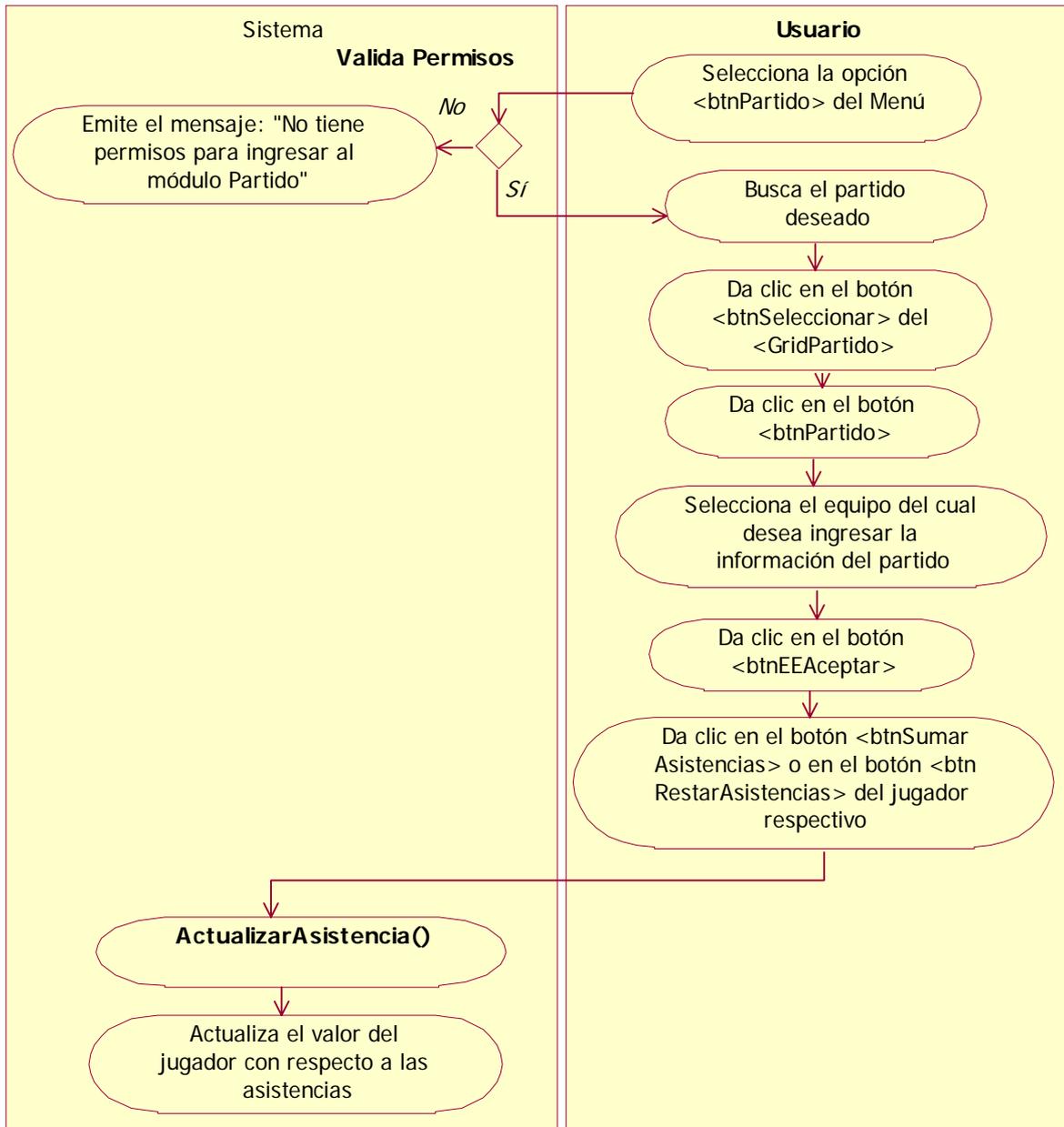
Modelo de composición de servicios –

Actualizar Rebote Defensivo (Periodo)



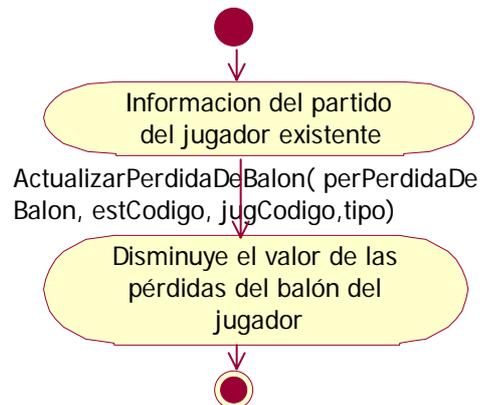
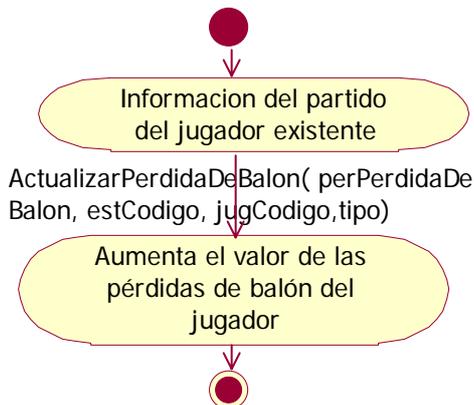
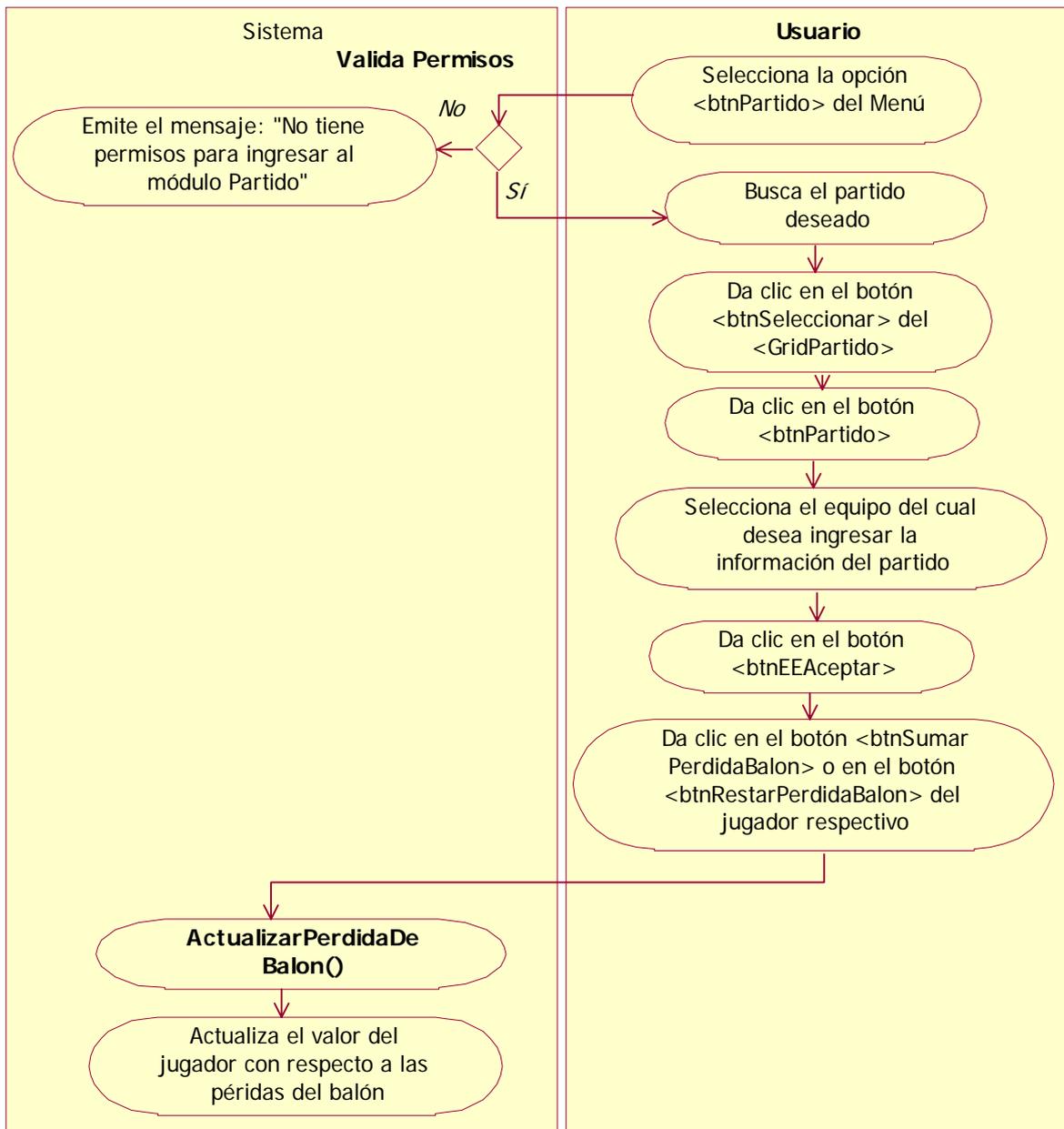
Modelo de composición de servicios –

Actualizar Asistencia (Periodo)



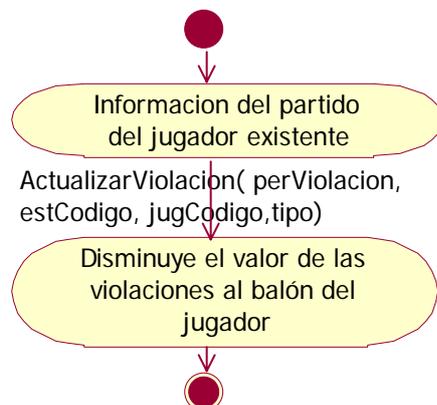
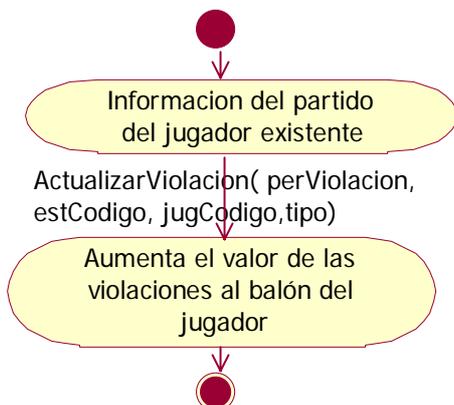
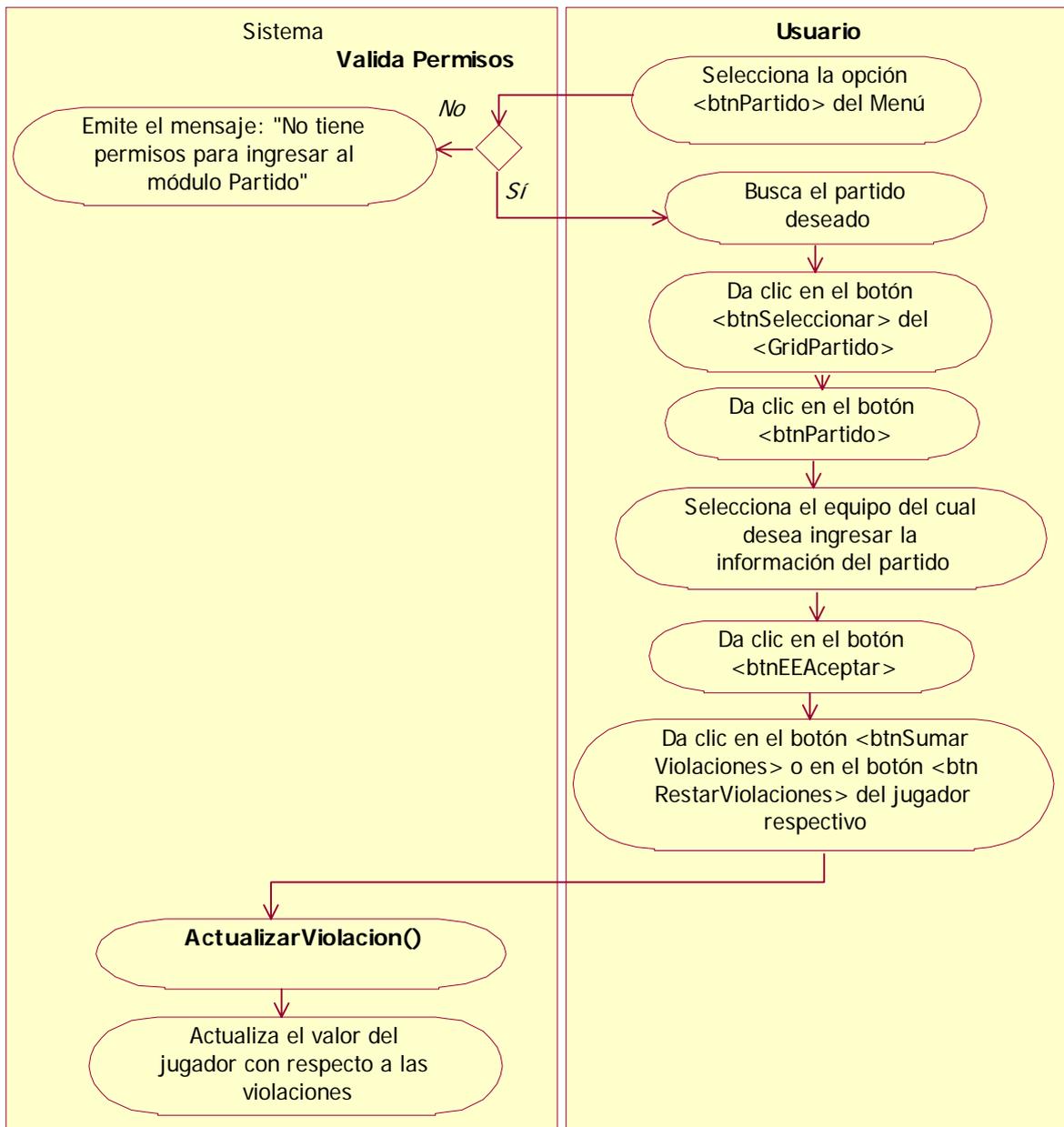
Modelo de composición de servicios –

Actualizar Perdida de Balón (Periodo)



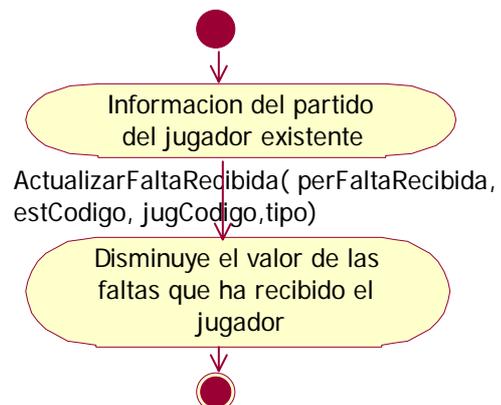
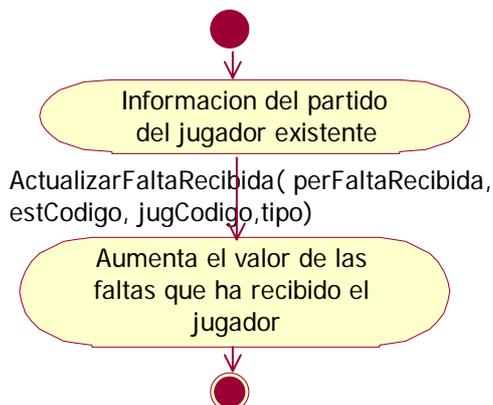
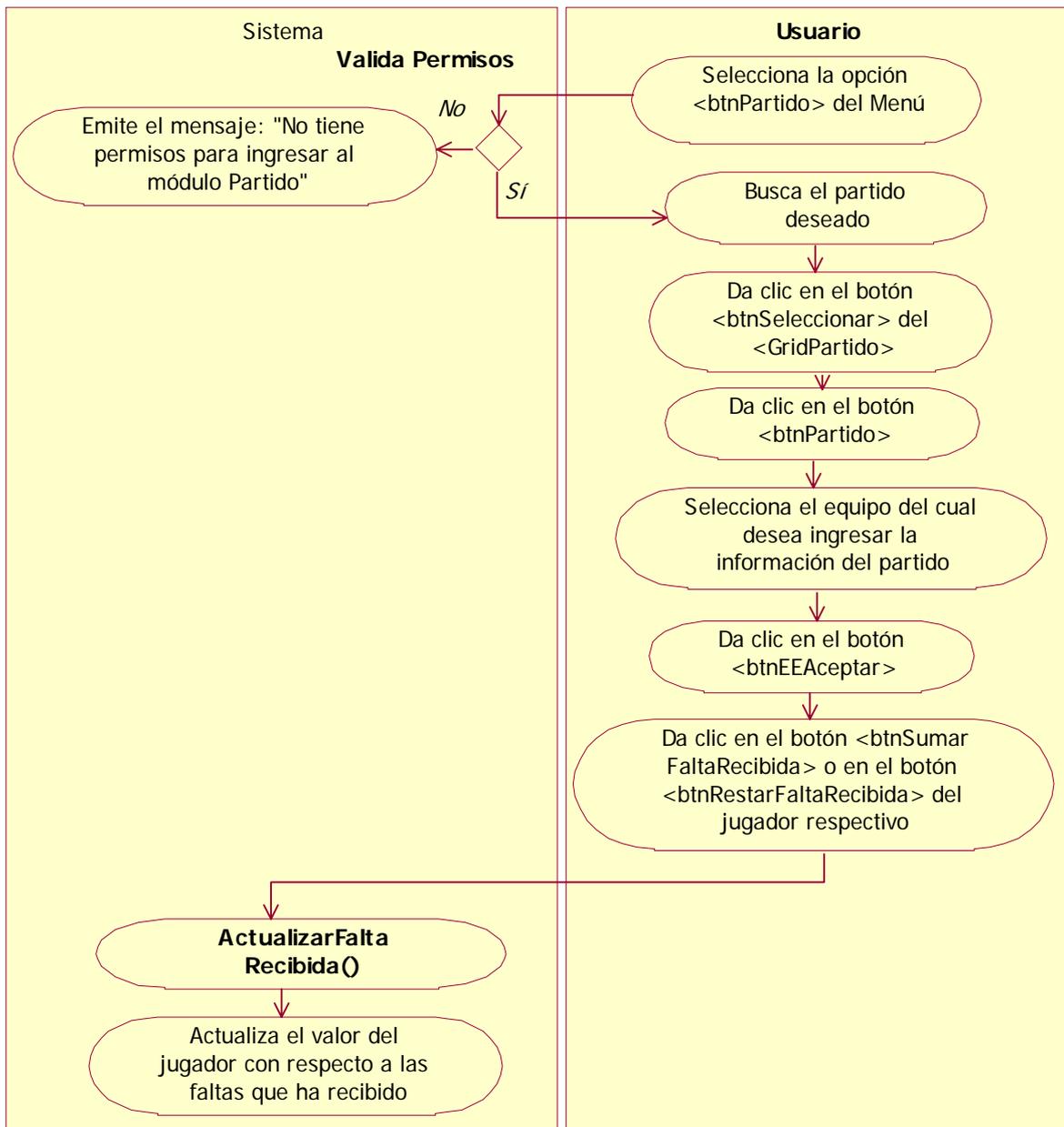
Modelo de composición de servicios –

Actualizar Violación (Periodo)



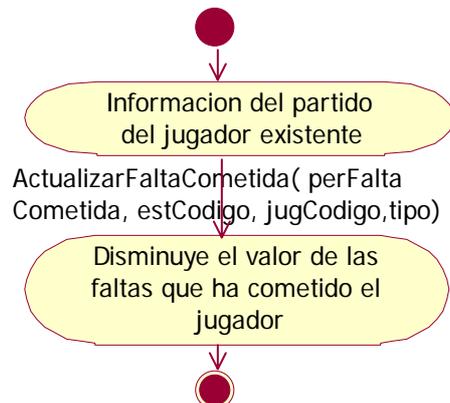
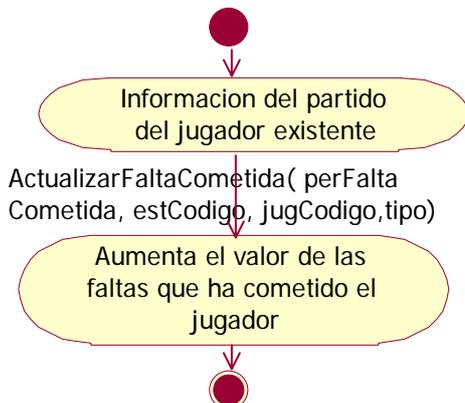
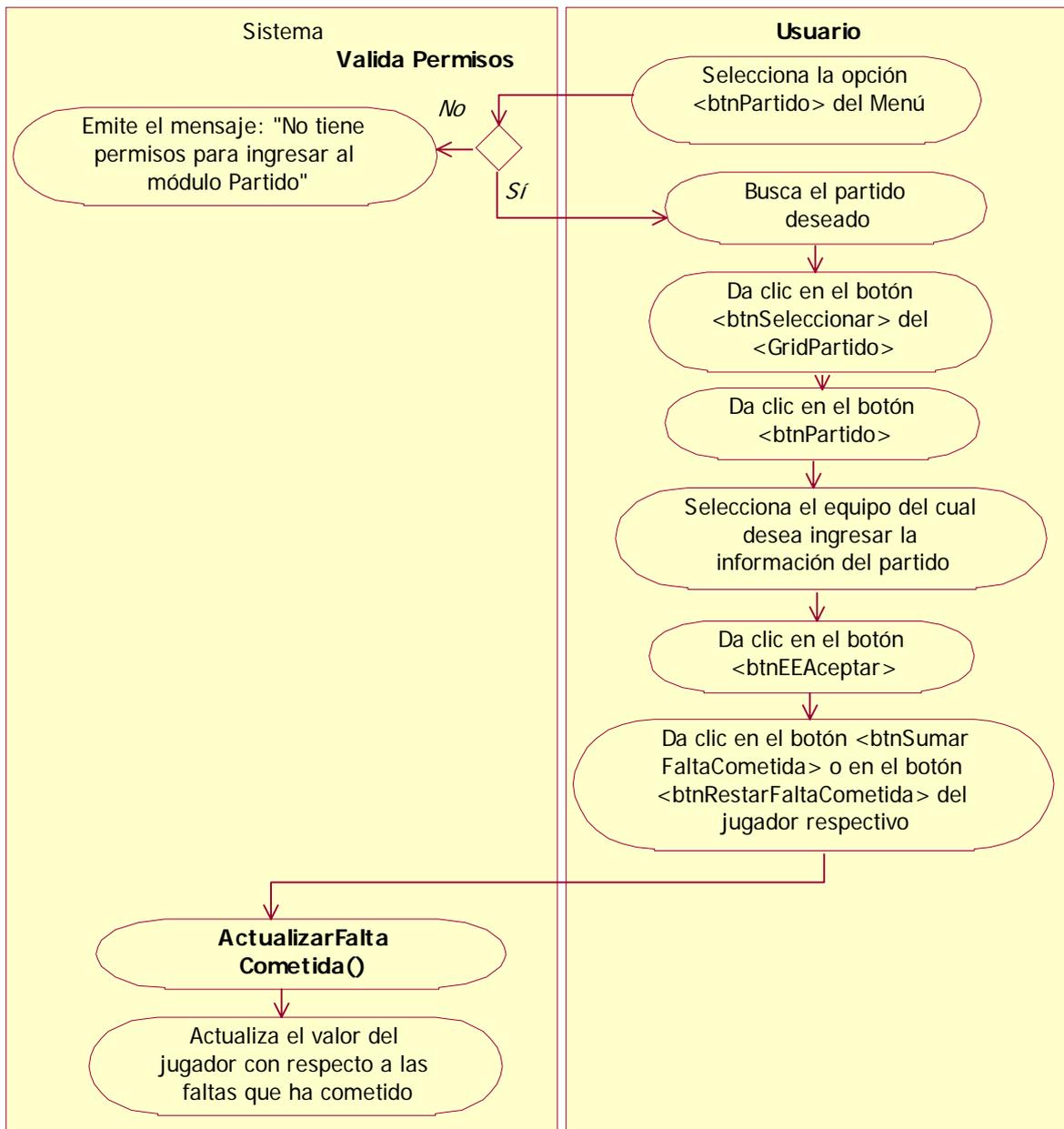
Modelo de composición de servicios –

Actualizar Falta Recibida (Periodo)

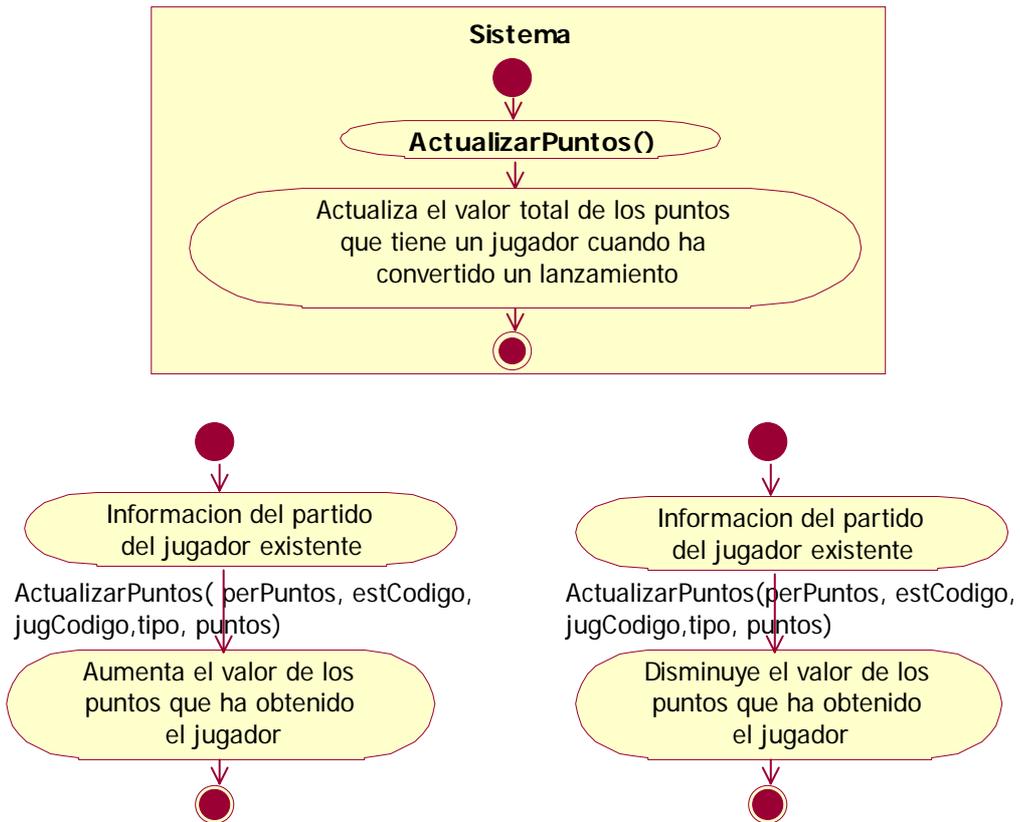


Modelo de composición de servicios –

Actualizar Falta Cometida (Periodo)

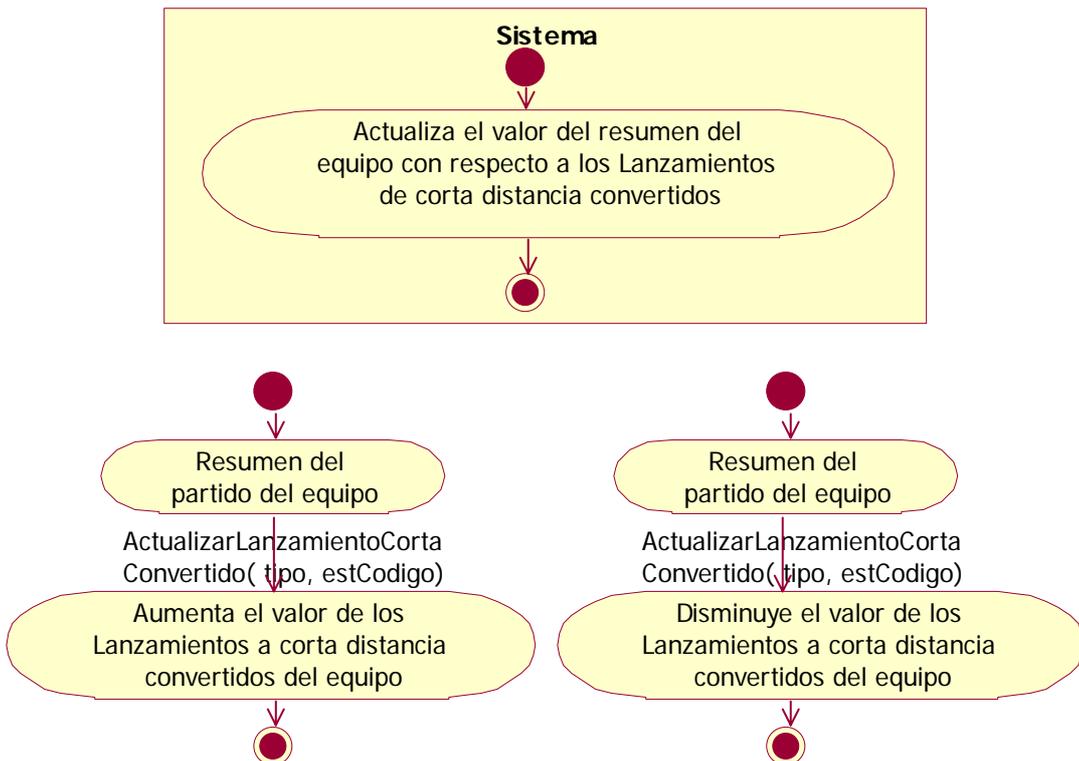


Modelo de composición de servicios – Actualizar Puntos (Periodo)



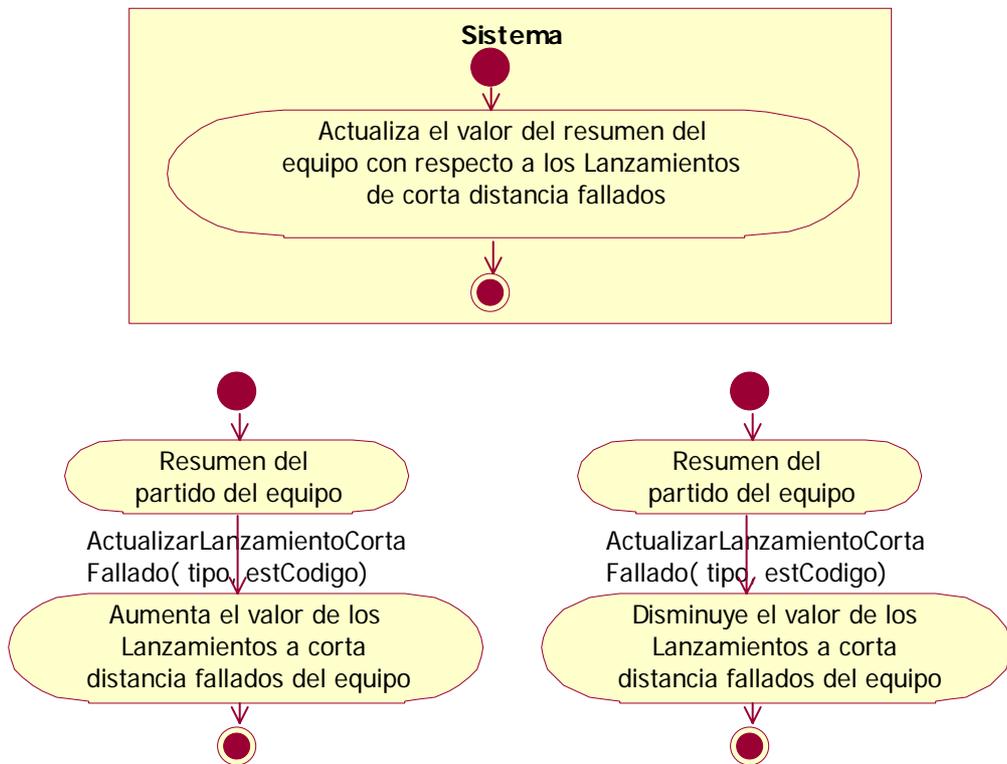
Modelo de composición de servicios –

Actualizar Lanzamiento Corta Convertido (Resumen)



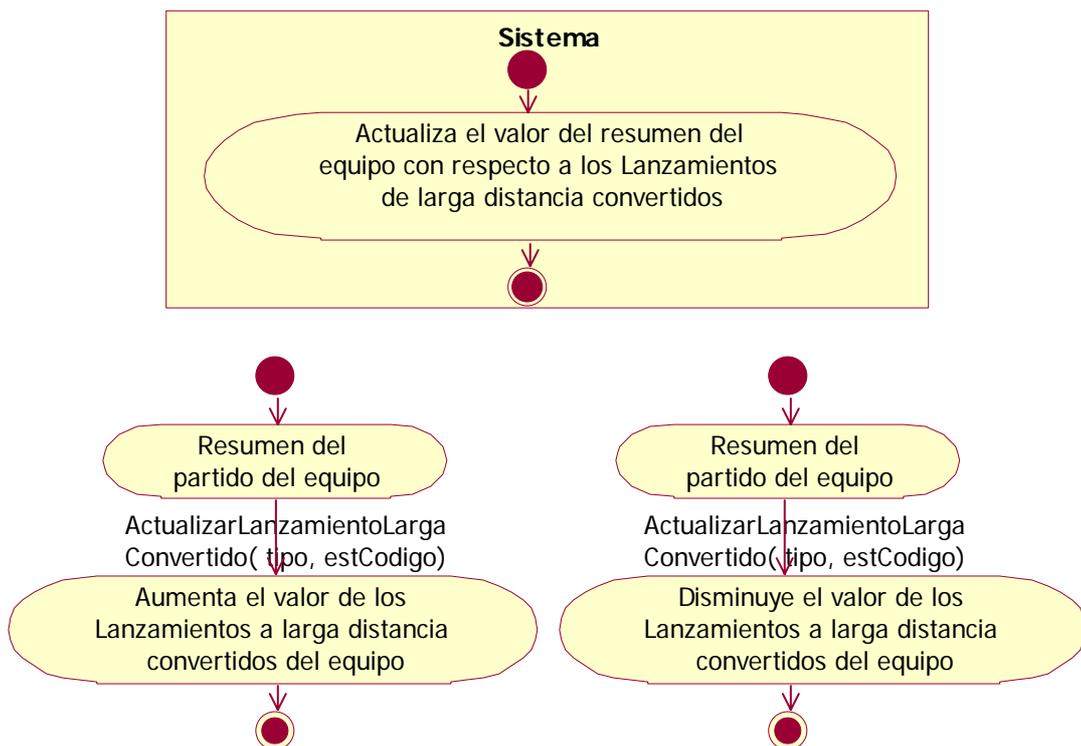
Modelo de composición de servicios –

Actualizar Lanzamiento Corta Fallado (Resumen)



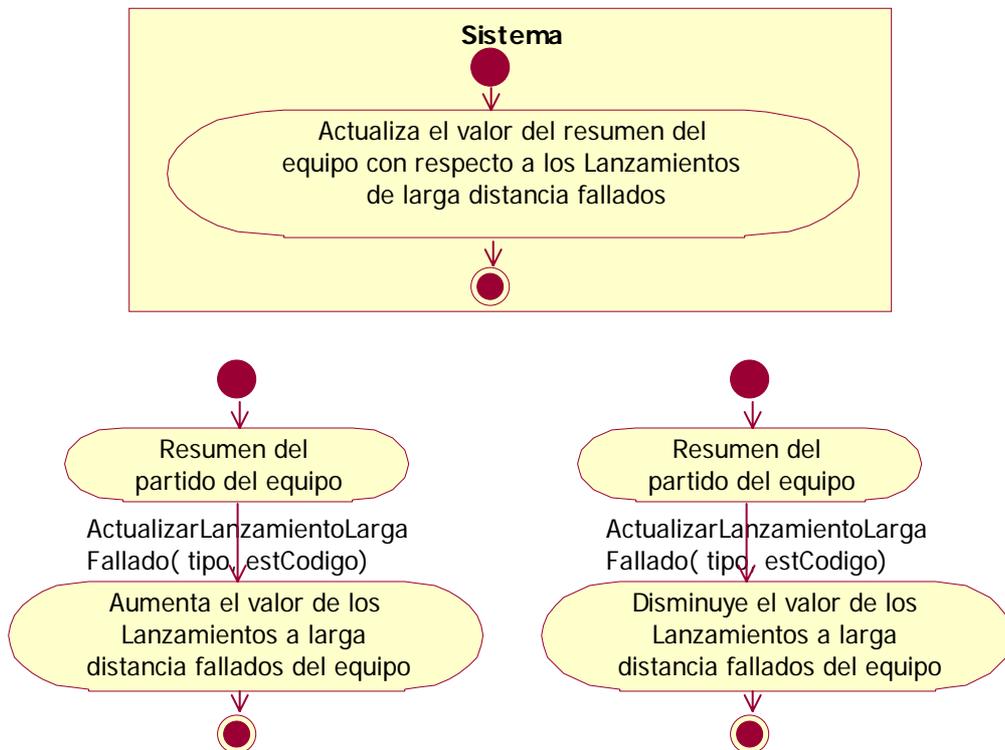
Modelo de composición de servicios –

Actualizar Lanzamiento Larga Convertido (Resumen)



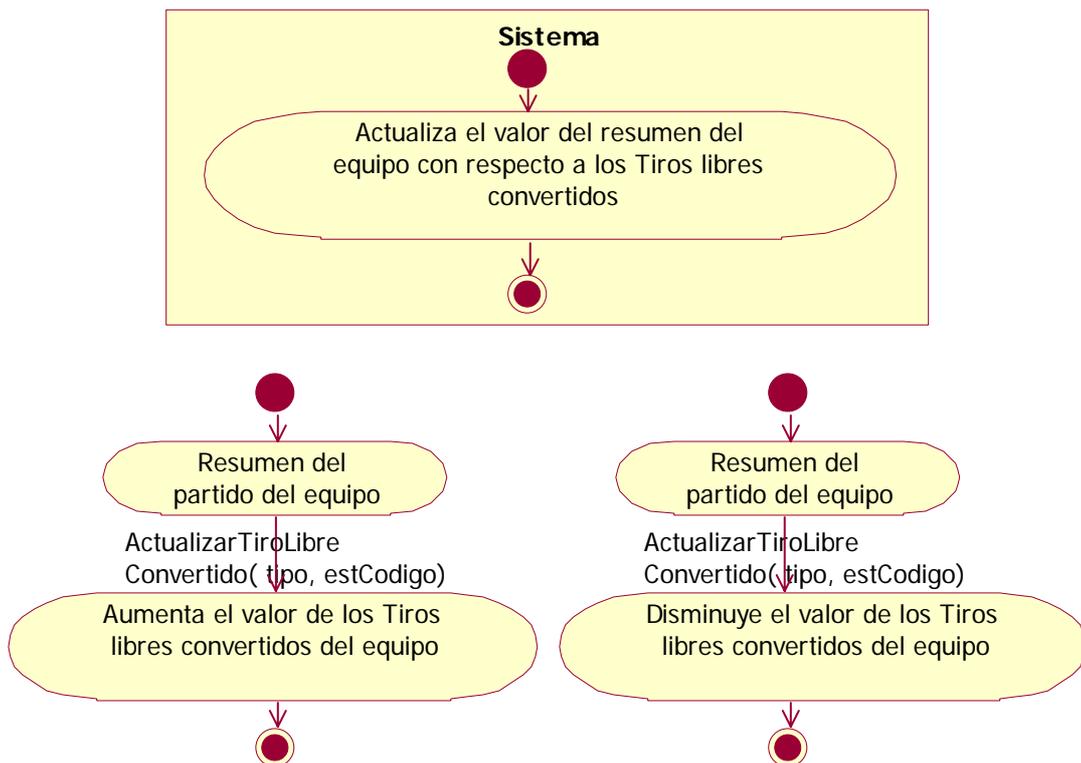
Modelo de composición de servicios –

Actualizar Lanzamiento Larga Fallado (Resumen)



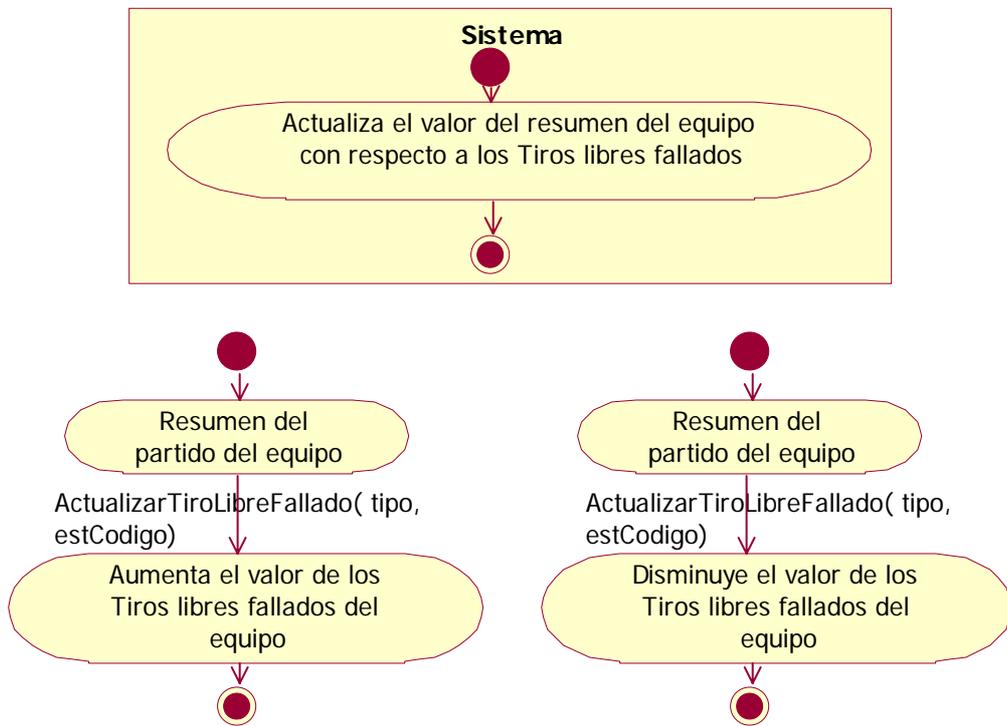
Modelo de composición de servicios –

Actualizar Tiro Libre Convertido (Resumen)



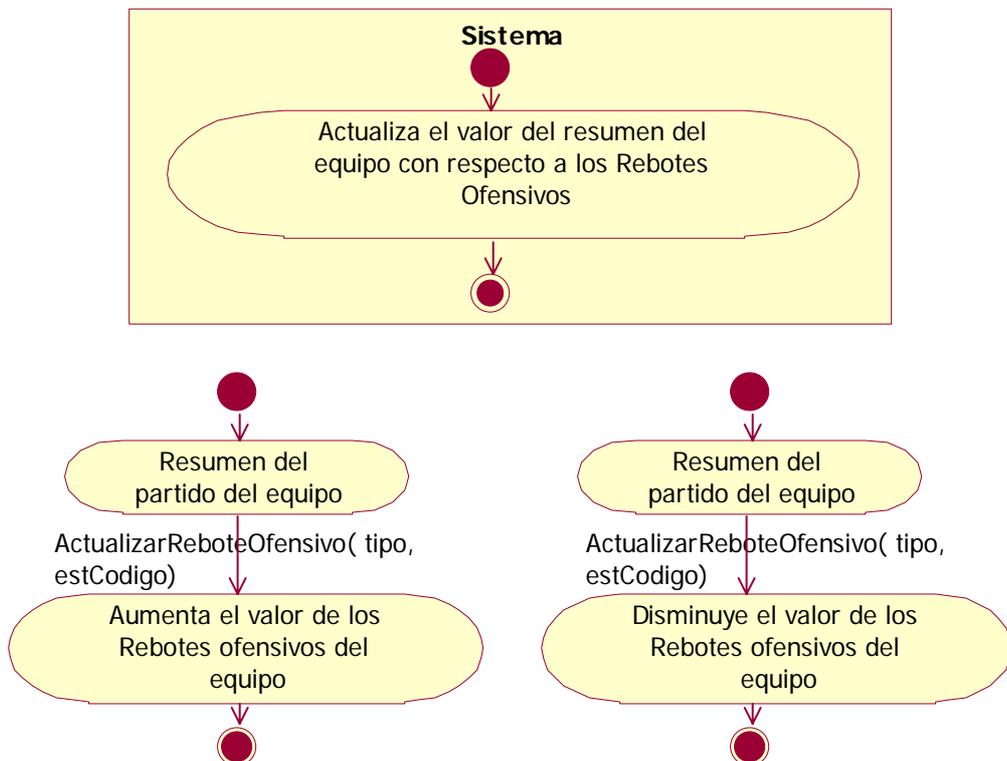
Modelo de composición de servicios –

Actualizar Tiro Libre Fallado (Resumen)



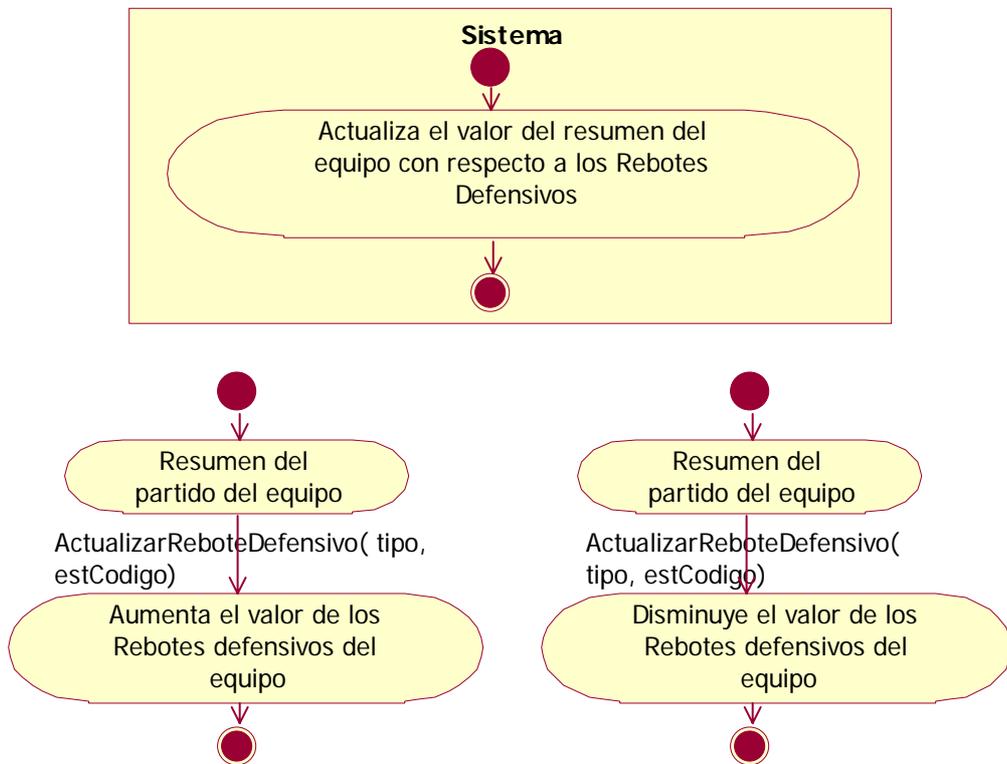
Modelo de composición de servicios –

Actualizar Rebote Ofensivo (Resumen)



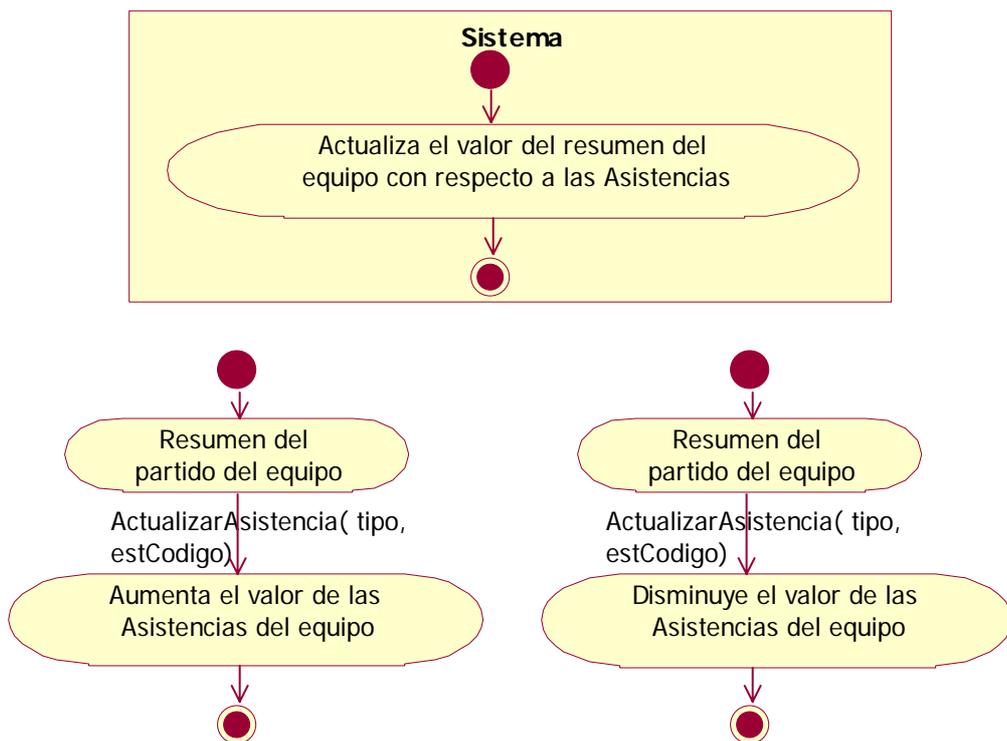
Modelo de composición de servicios –

Actualizar Rebote Defensivo (Resumen)



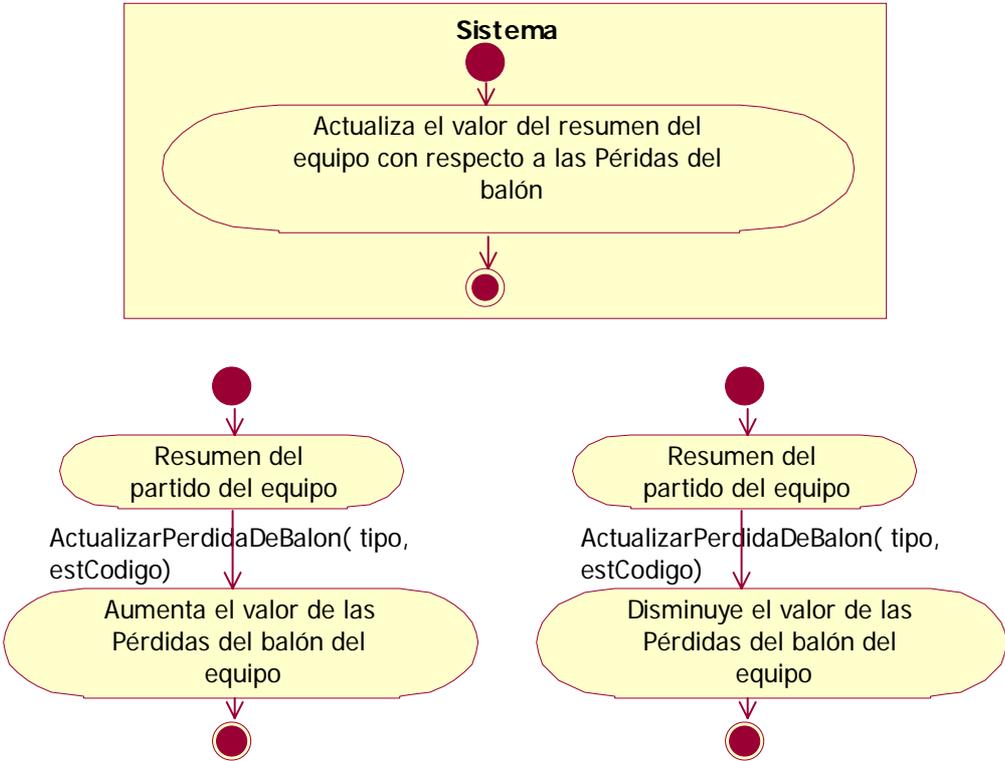
Modelo de composición de servicios –

Actualizar Asistencia (Resumen)



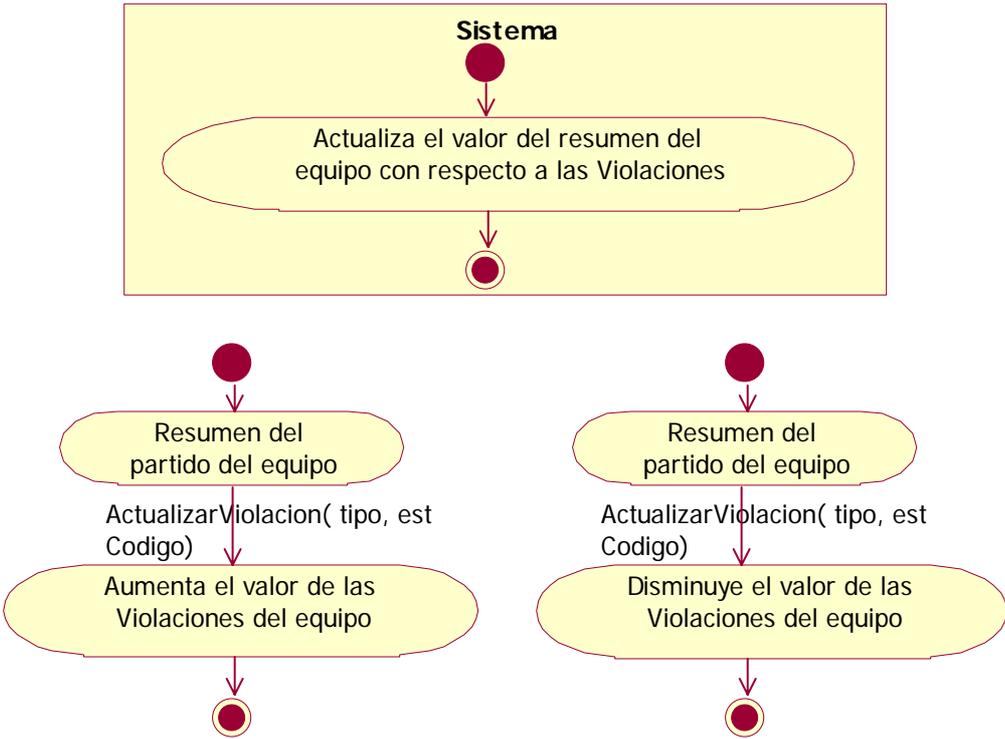
Modelo de composición de servicios –

Actualizar Perdida de Balón (Resumen)



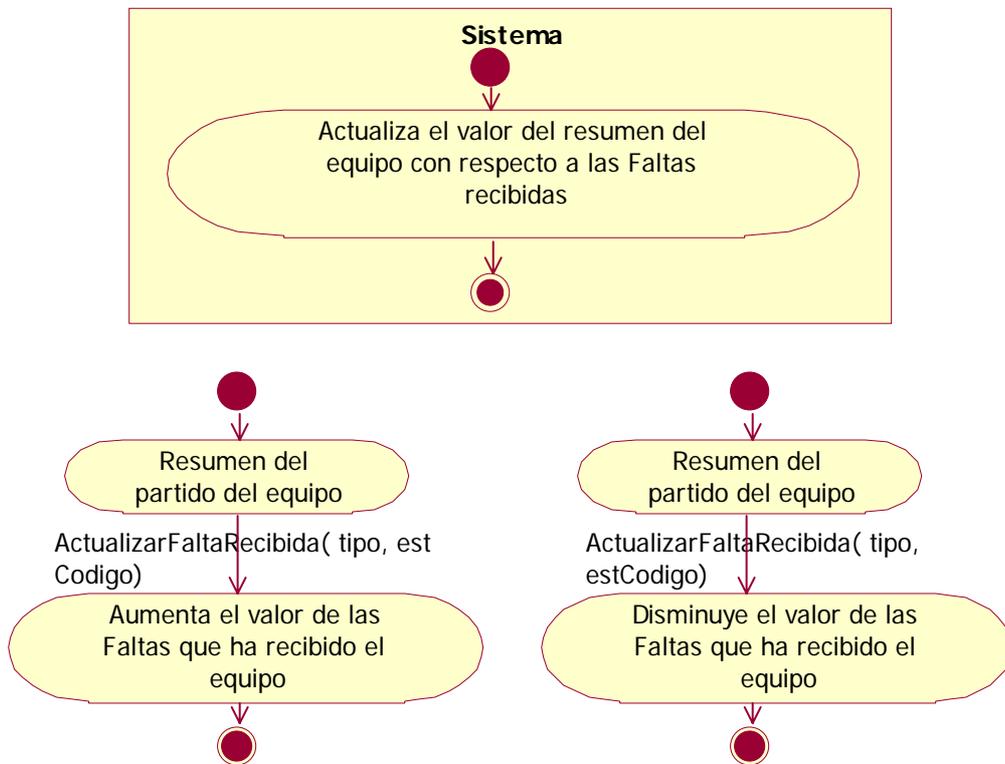
Modelo de composición de servicios –

Actualizar Violación (Resumen)



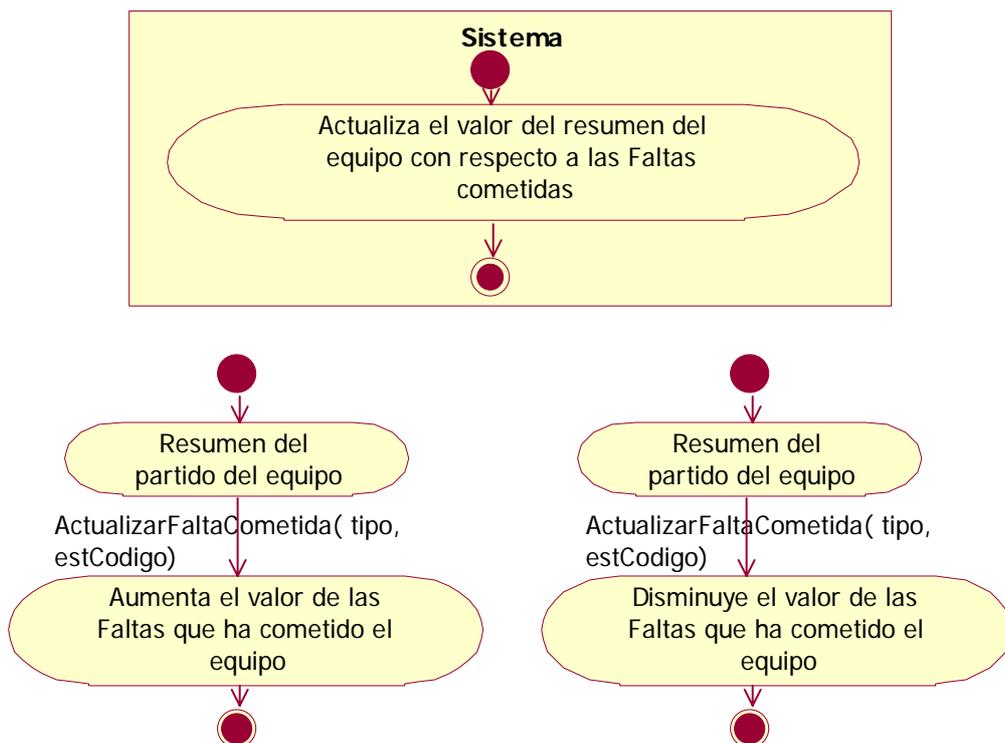
Modelo de composición de servicios –

Actualizar Falta Recibida (Resumen)



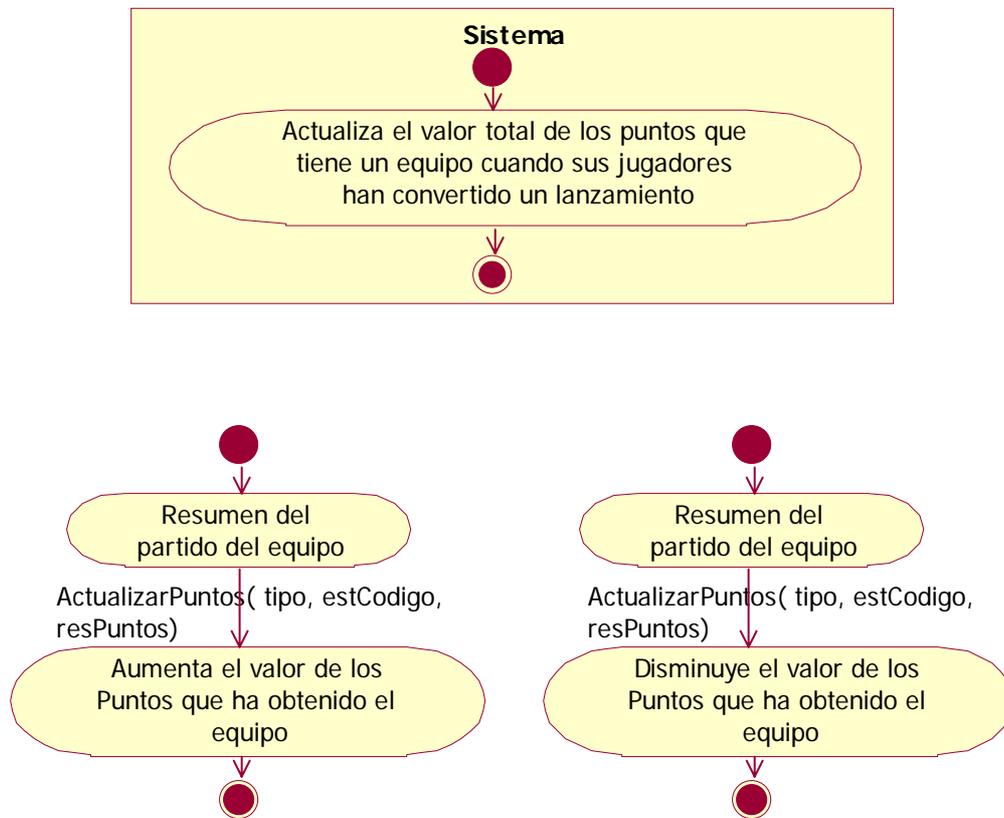
Modelo de composición de servicios –

Actualizar Falta Cometida (Resumen)



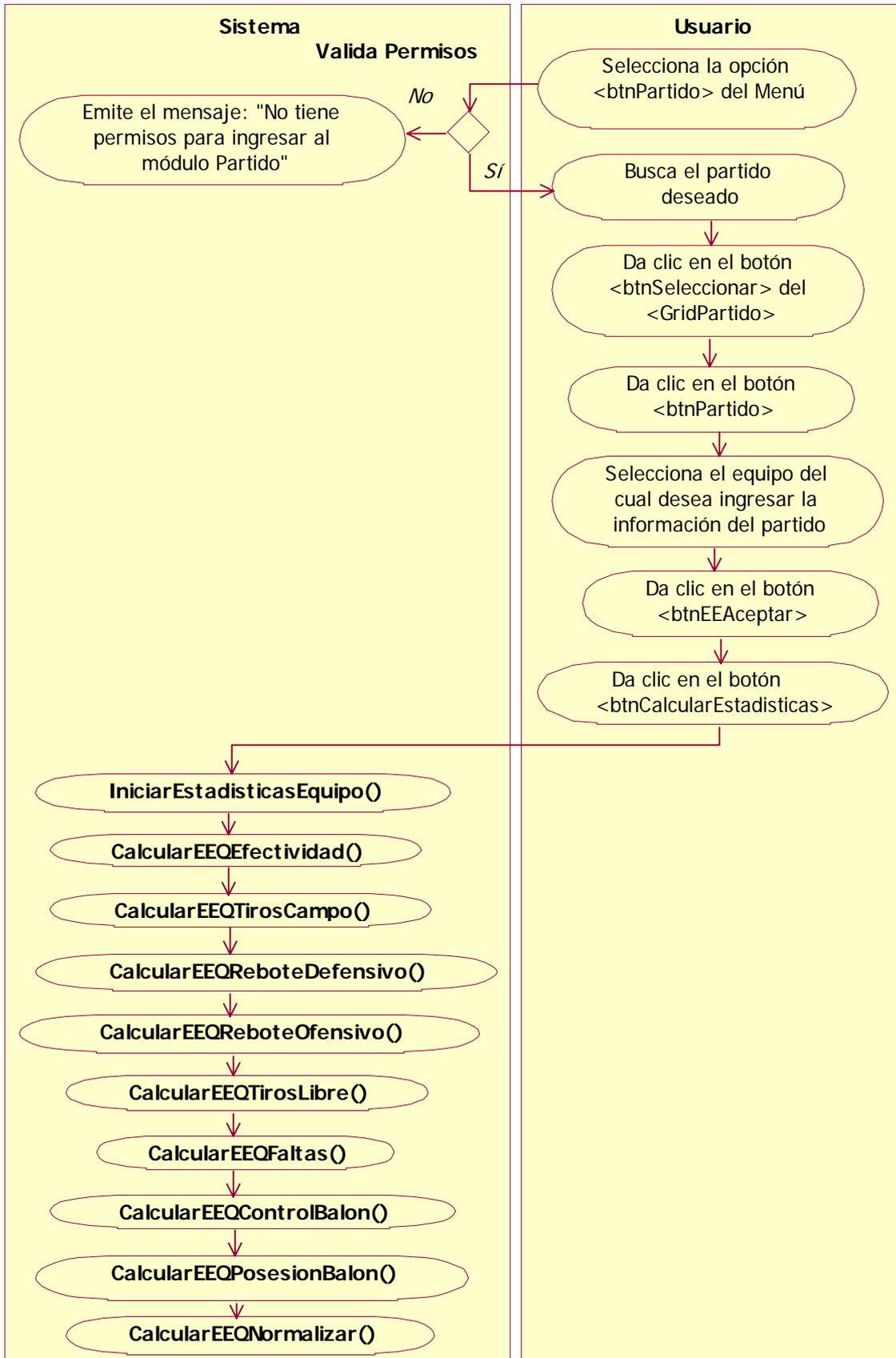
Modelo de composición de servicios –

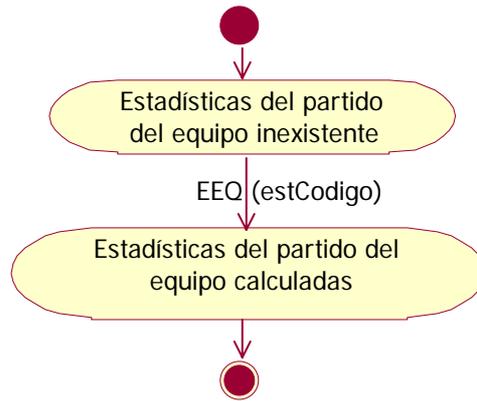
Actualizar Puntos (Resumen)



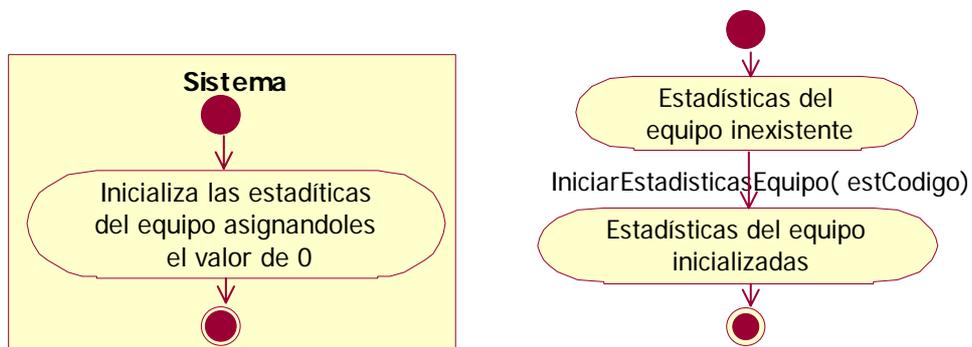
- Paquete Estadística

Modelo de composición de servicios – EEQ

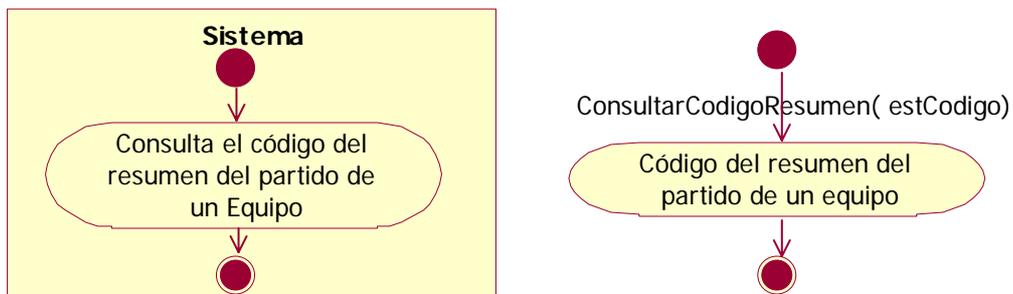




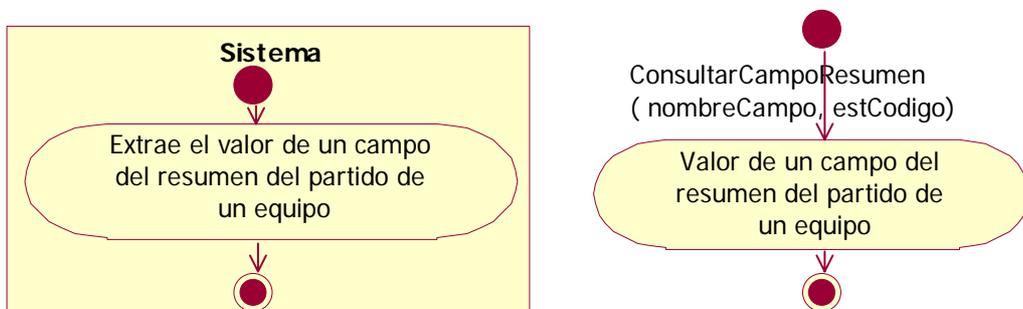
Modelo de composición de servicios – Iniciar Estadísticas Equipo



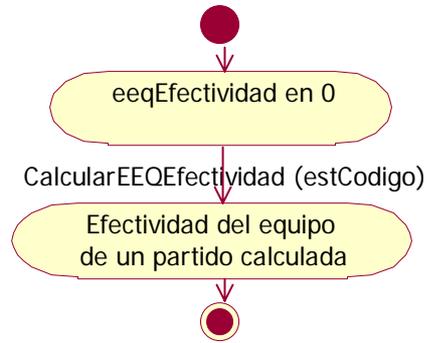
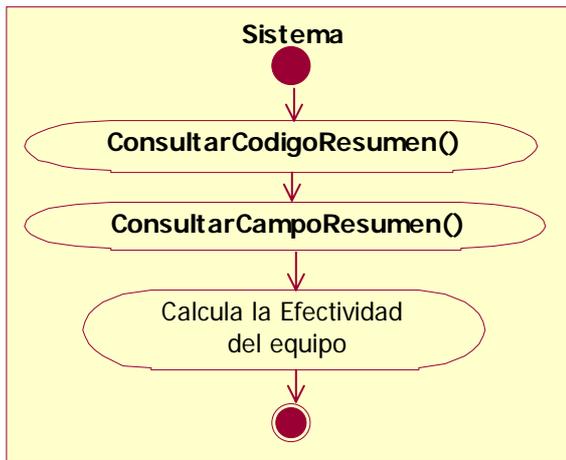
Modelo de composición de servicios – Consultar Código Resumen



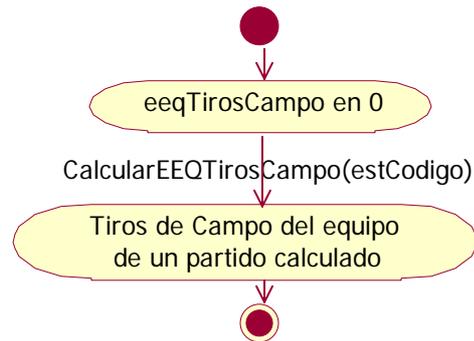
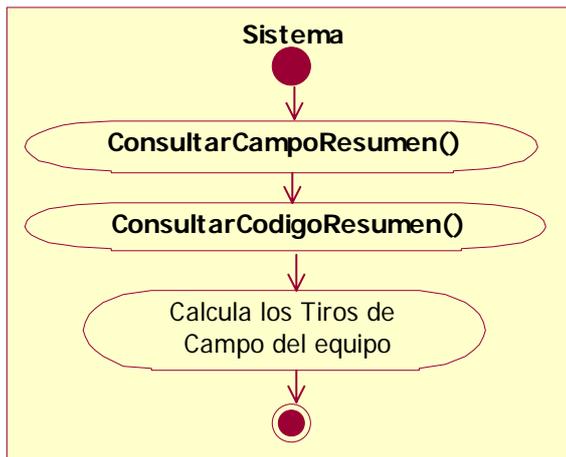
Modelo de composición de servicios – Consultar Campo Resumen



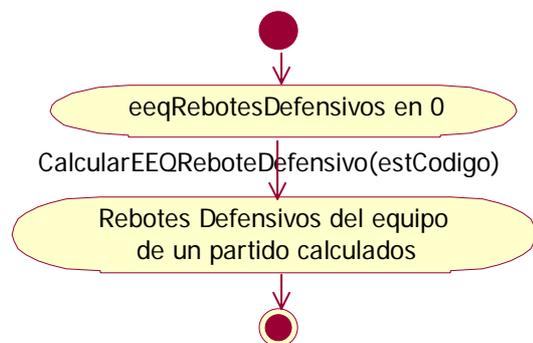
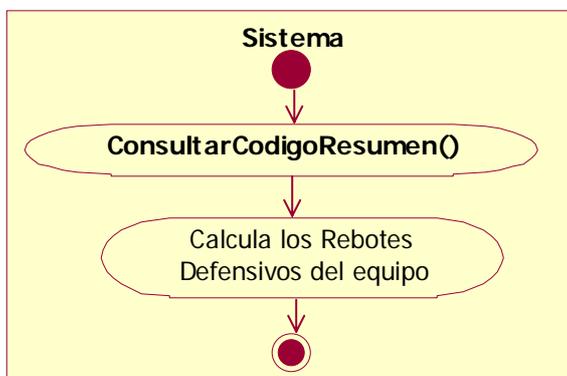
Modelo de composición de servicios – CalcularEEQ Efectividad



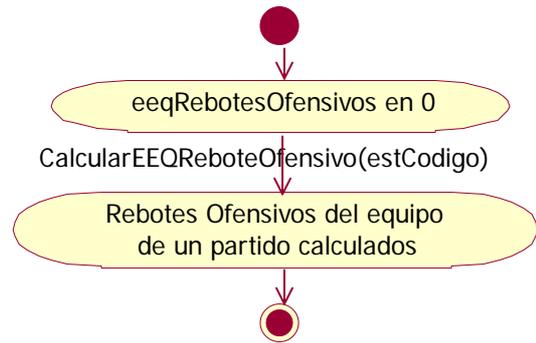
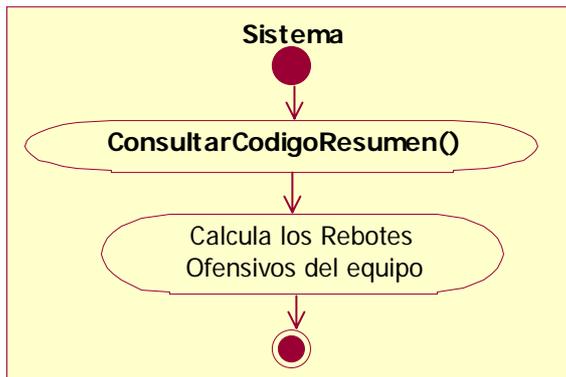
Modelo de composición de servicios – CalcularEEQ Tiros Campo



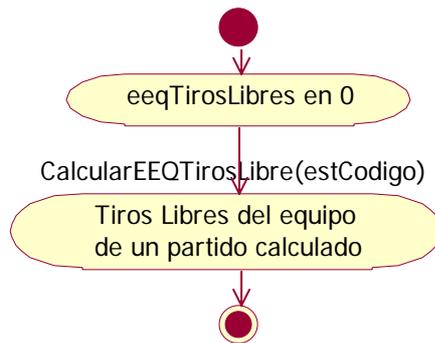
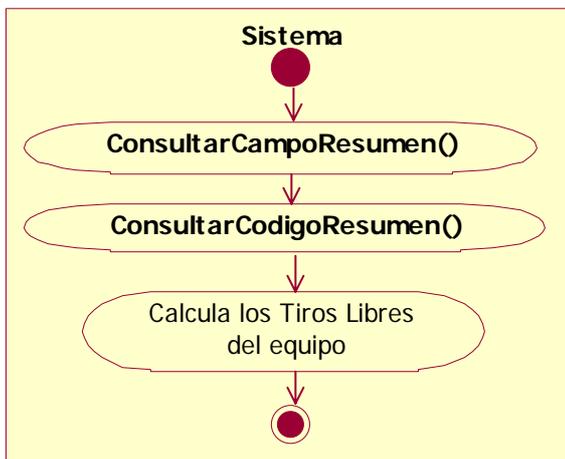
Modelo de composición de servicios – CalcularEEQ Rebote Defensivo



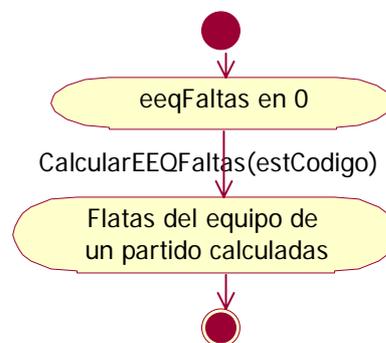
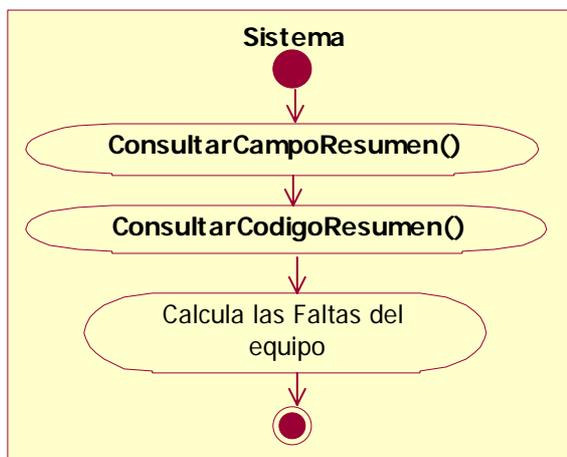
Modelo de composición de servicios – CalcularEEQ Rebote Ofensivo



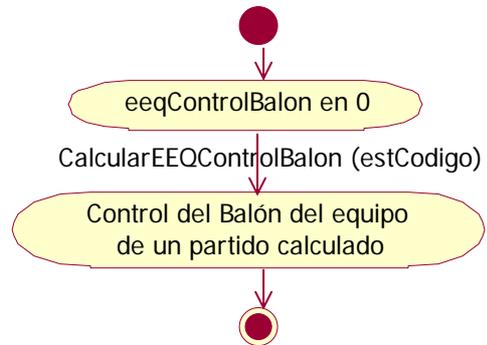
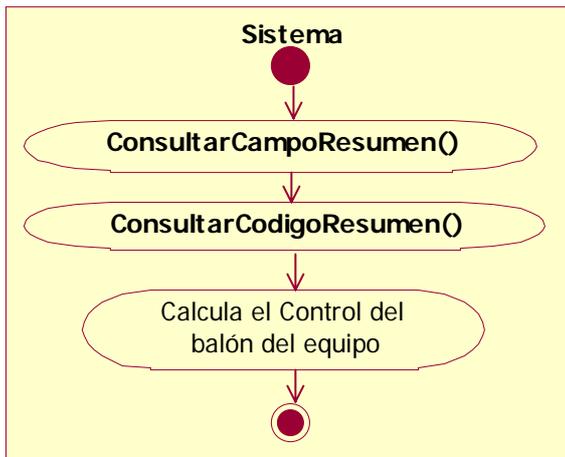
Modelo de composición de servicios – CalcularEEQ Tiros Libres



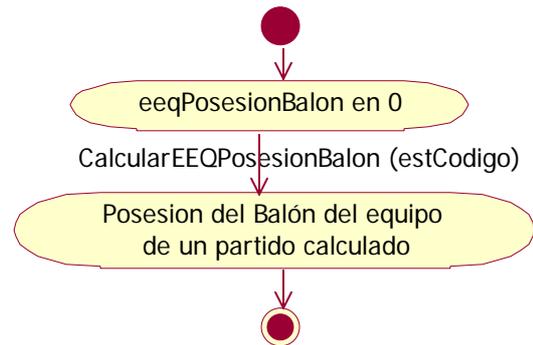
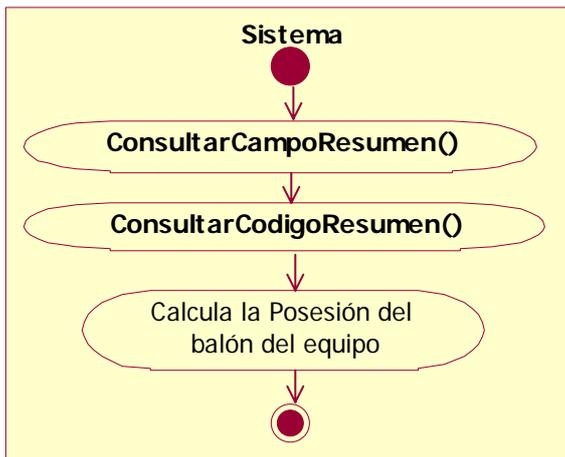
Modelo de composición de servicios – CalcularEEQ Faltas



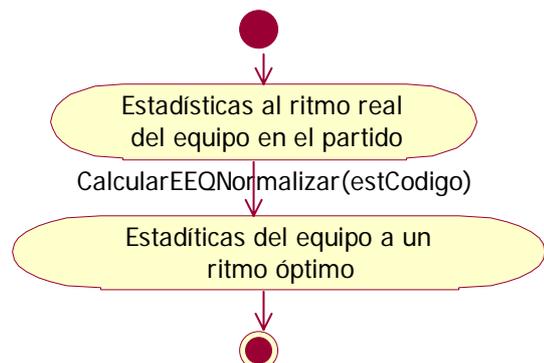
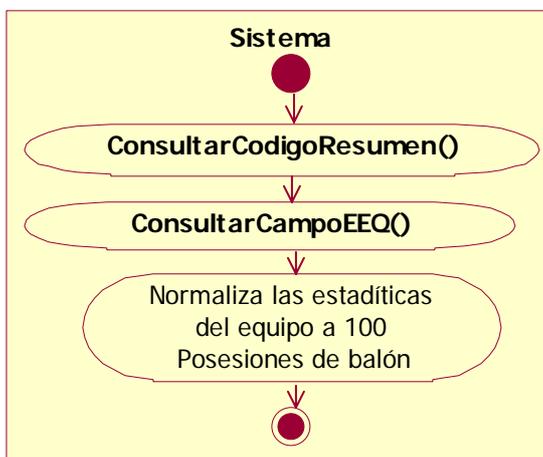
Modelo de composición de servicios – CalcularEEQ Control Balón



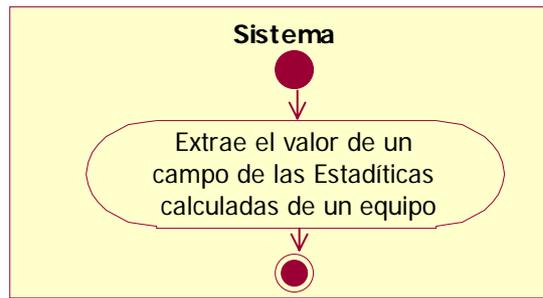
Modelo de composición de servicios – CalcularEEQ Posesión Balón



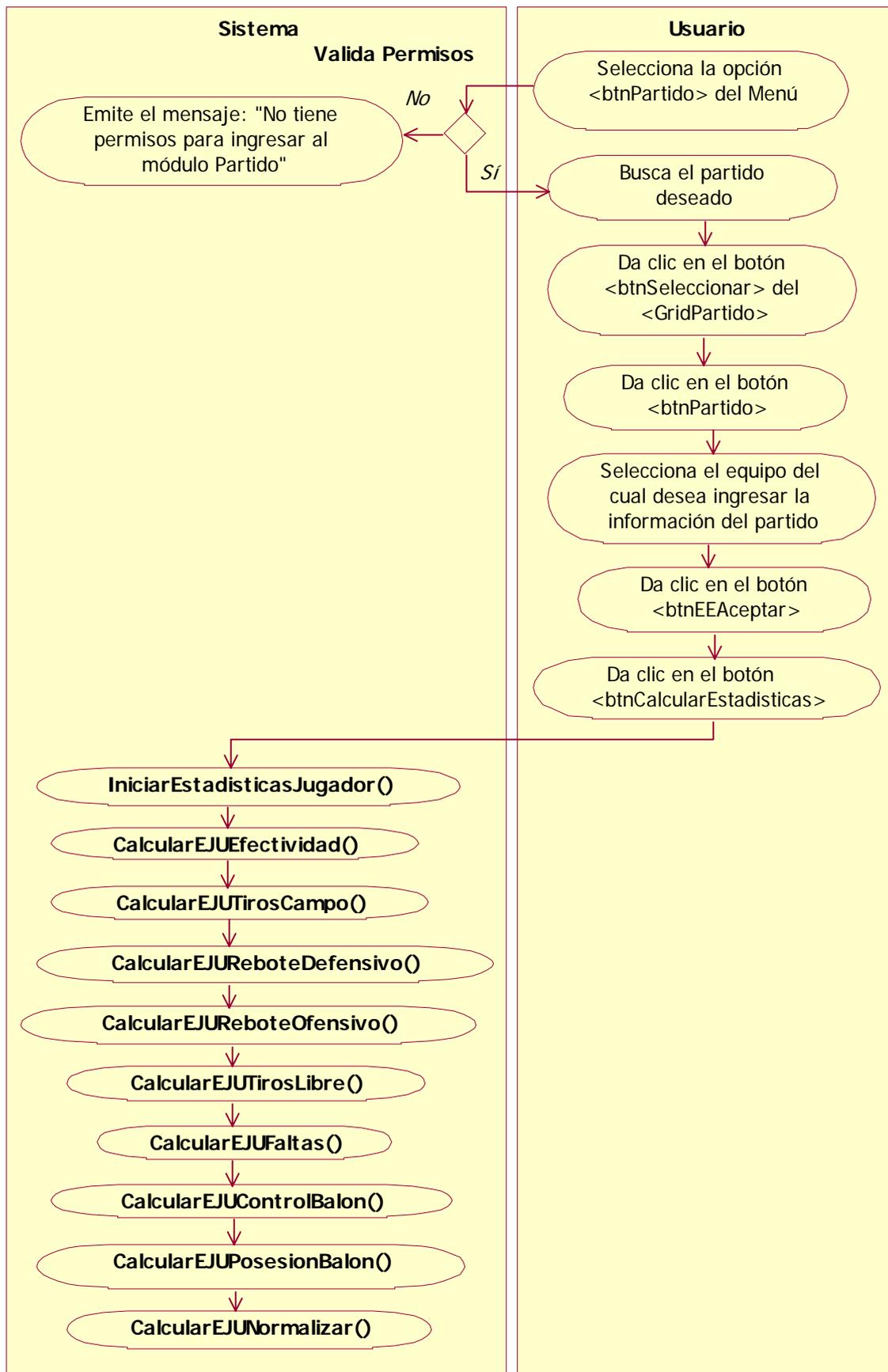
Modelo de composición de servicios – CalcularEEQ Normalizar

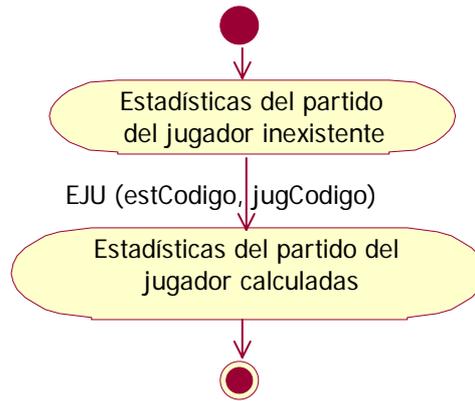


Modelo de composición de servicios – Consultar Campo EEQ

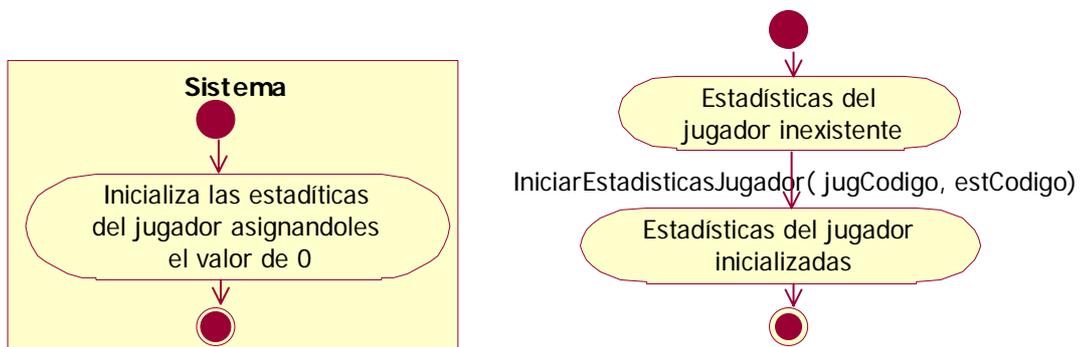


Modelo de composición de servicios – EJU

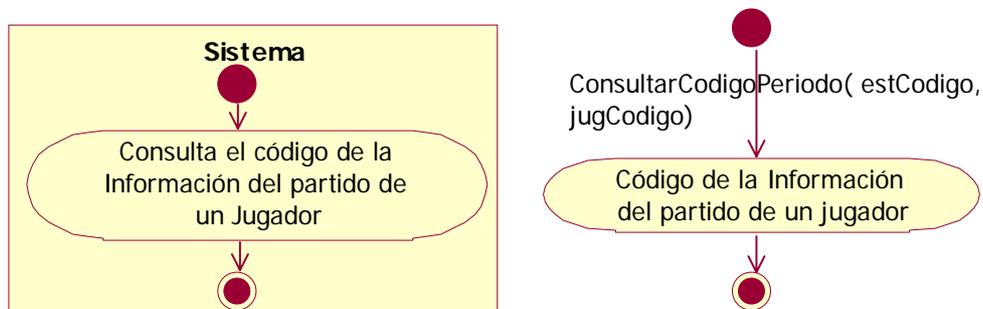




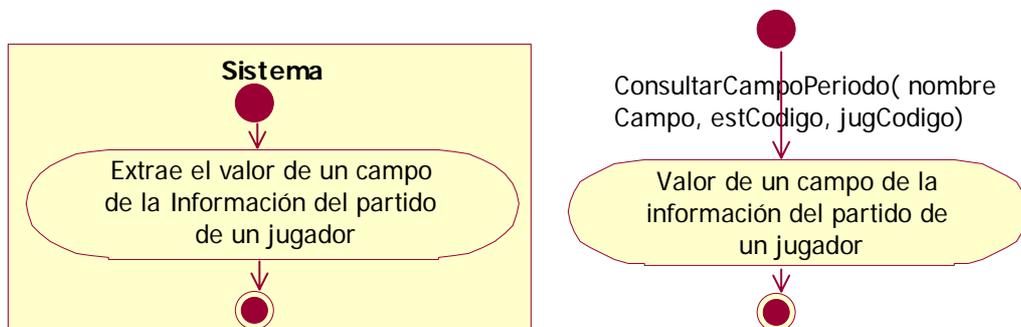
Modelo de composición de servicios – Iniciar Estadísticas Jugador



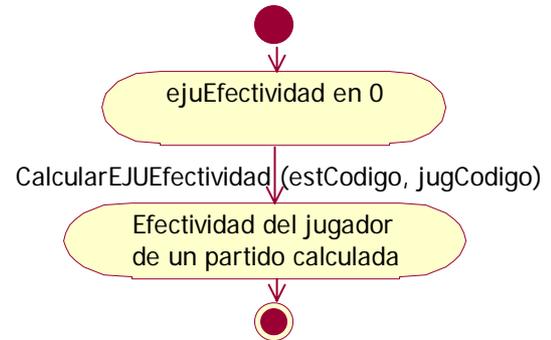
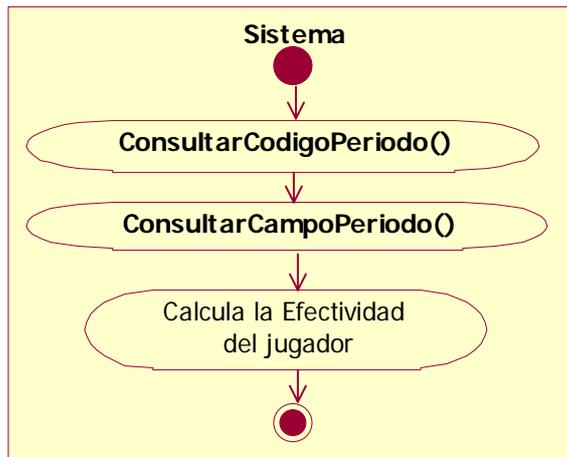
Modelo de composición de servicios – Consultar Código Periodo



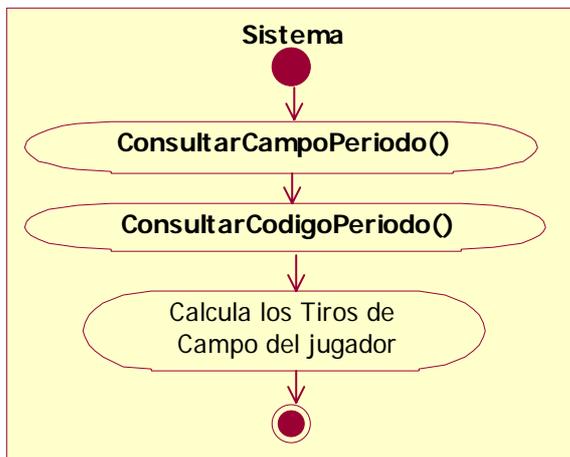
Modelo de composición de servicios – Consultar Campo Periodo



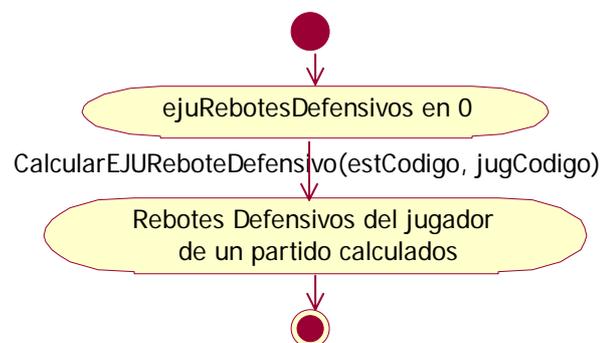
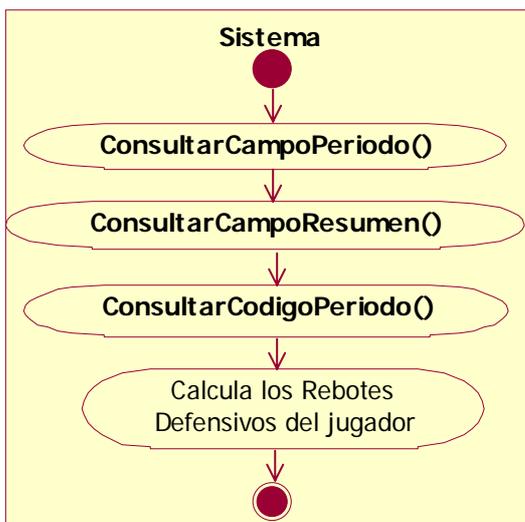
Modelo de composición de servicios – CalcularEJU Efectividad



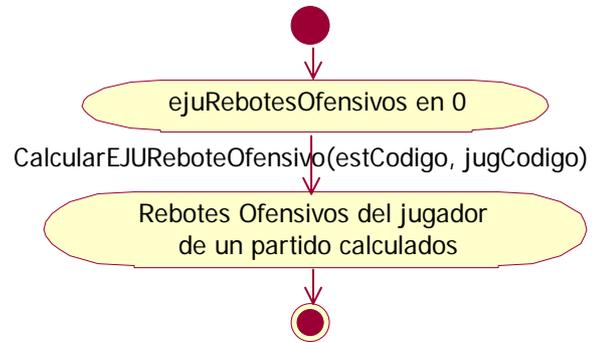
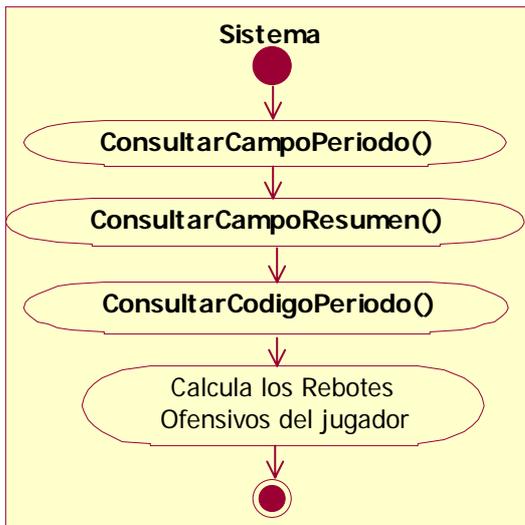
Modelo de composición de servicios – CalcularEJU Tiros Campo



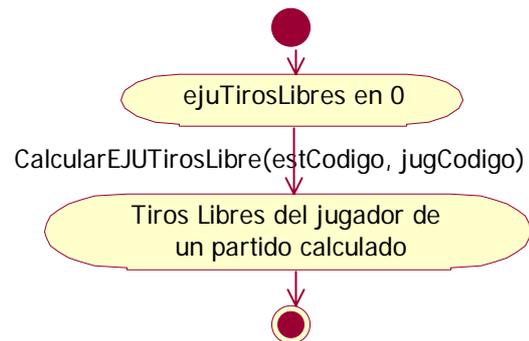
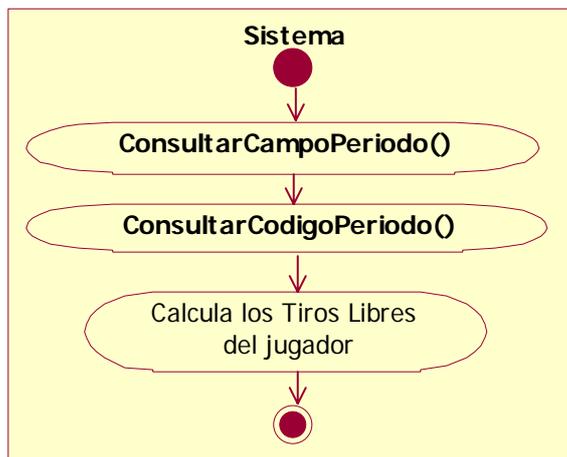
Modelo de composición de servicios – CalcularEJU Rebote Defensivo



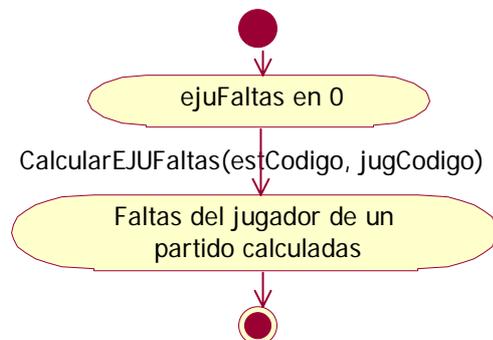
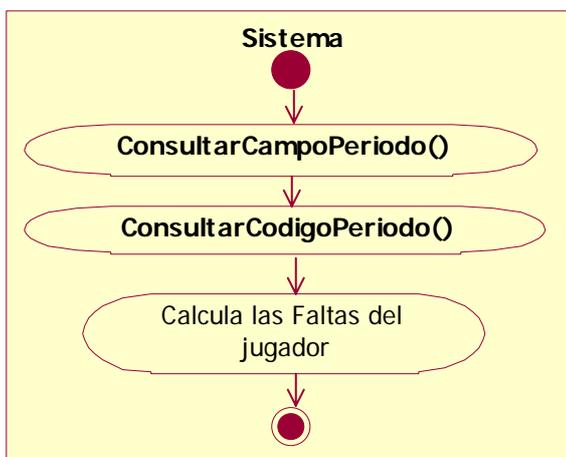
Modelo de composición de servicios – CalcularEJU Rebote Ofensivo



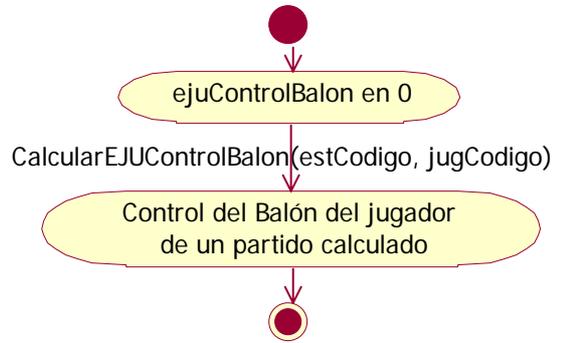
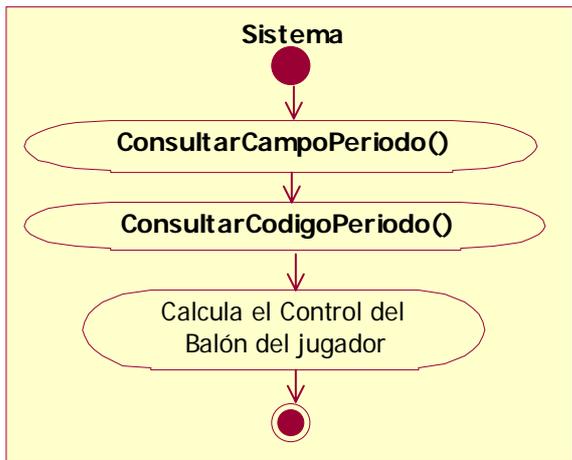
Modelo de composición de servicios – CalcularEJU Tiros Libres



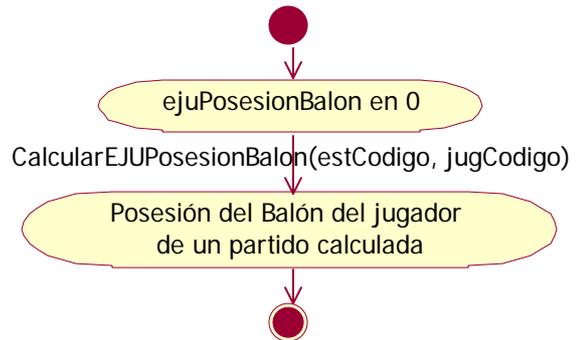
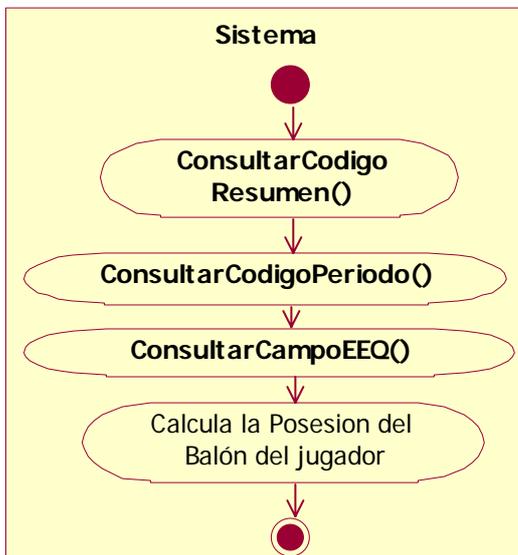
Modelo de composición de servicios – CalcularEJU Faltas



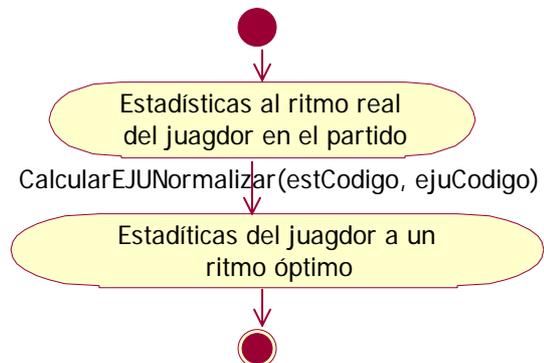
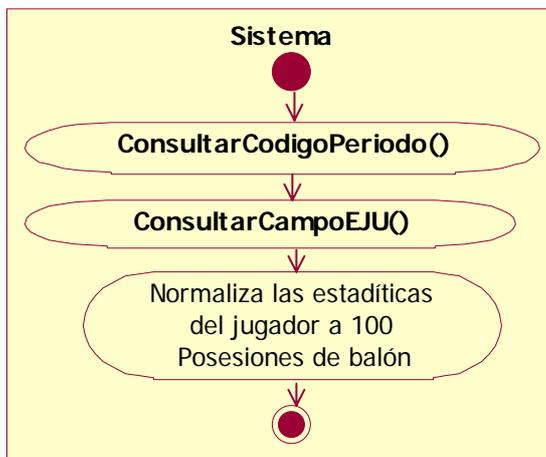
Modelo de composición de servicios – CalcularEJU Control Balón



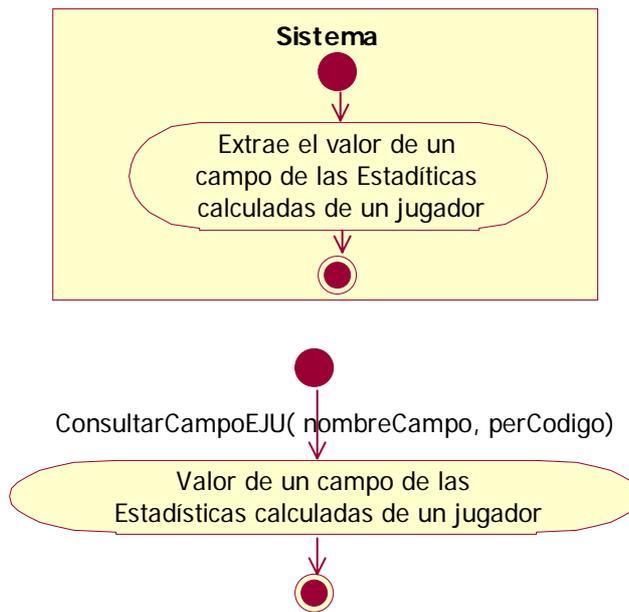
Modelo de composición de servicios – CalcularEJU Posesión Balón



Modelo de composición de servicios – CalcularEJU Normalizar



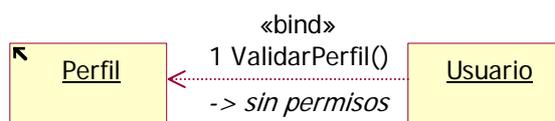
Modelo de composición de servicios – Consultar Campo EJU



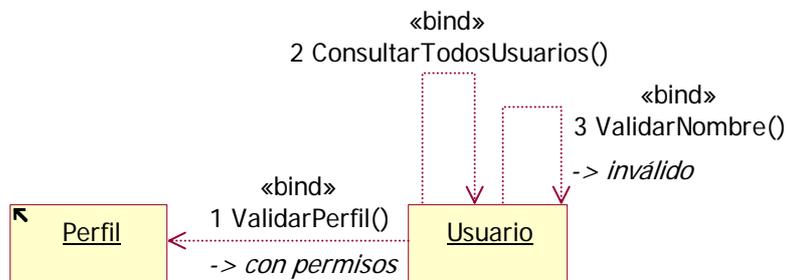
Para modelar cada uno de los servicios conceptuales que ofrece el sistema de información Web se recurre al modelo de servicios, los cuales son:

- Paquete Usuario
 - Ingresar usuario

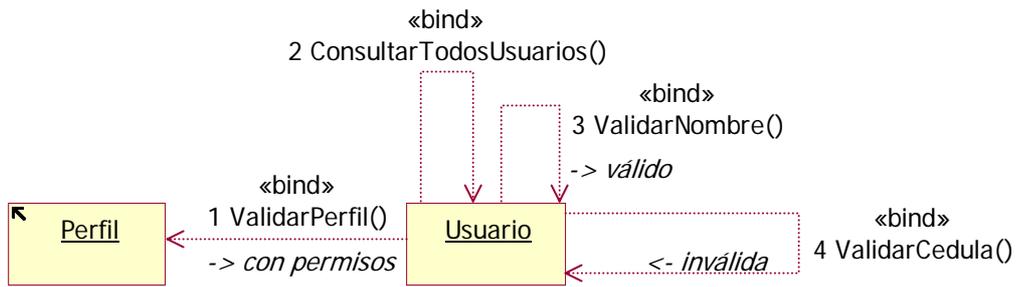
Modelo de Servicios – Ingresar Usuario Sin permisos



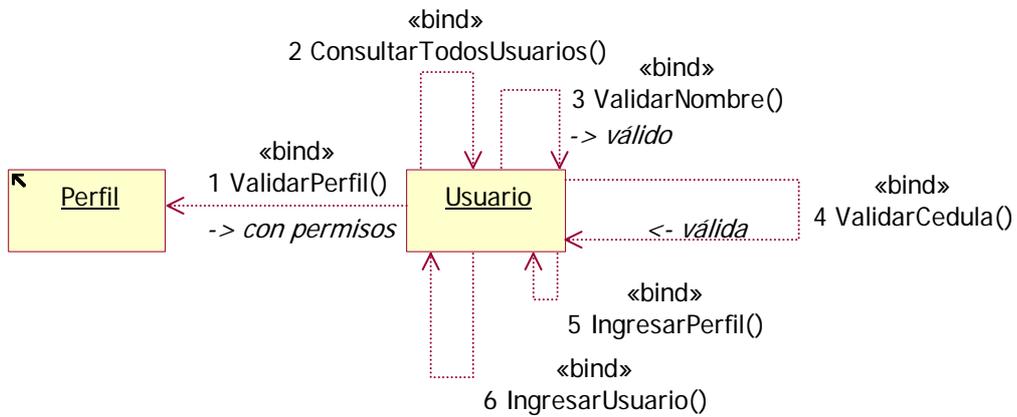
Modelo de Servicios – Ingresar Usuario Nombre Inválido



Modelo de Servicios – Ingresar Usuario Cedula Inválida

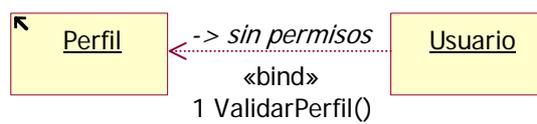


Modelo de Servicios – Ingresar Usuario

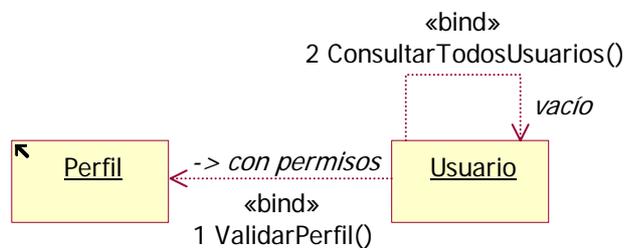


- Consultar usuario

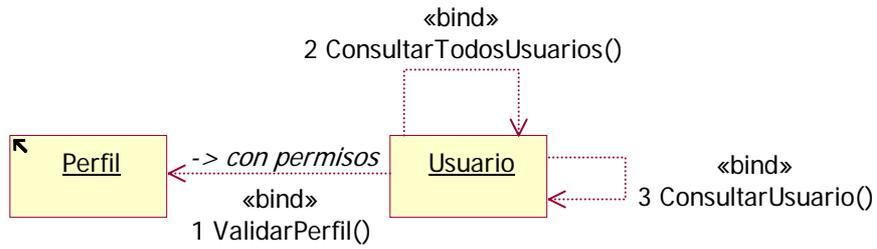
Modelo de Servicios – Consultar Usuario Sin permisos



Modelo de Servicios – Consultar Usuario Sin usuarios



Modelo de Servicios – Consultar Usuario

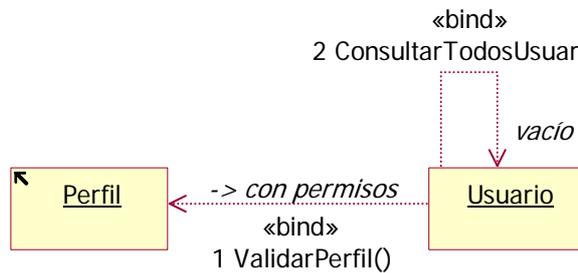


- Modificar usuario

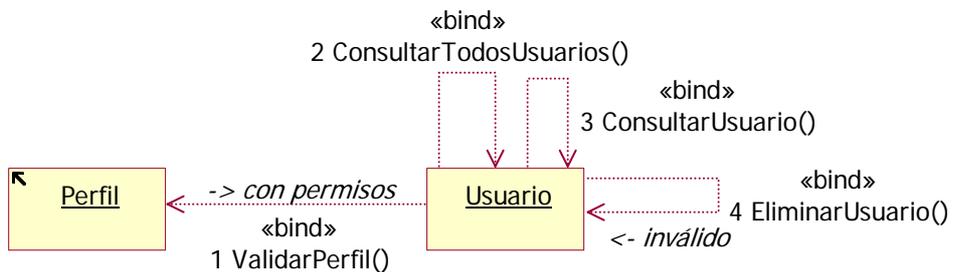
Modelo de Servicios – Modificar Usuario Sin permisos



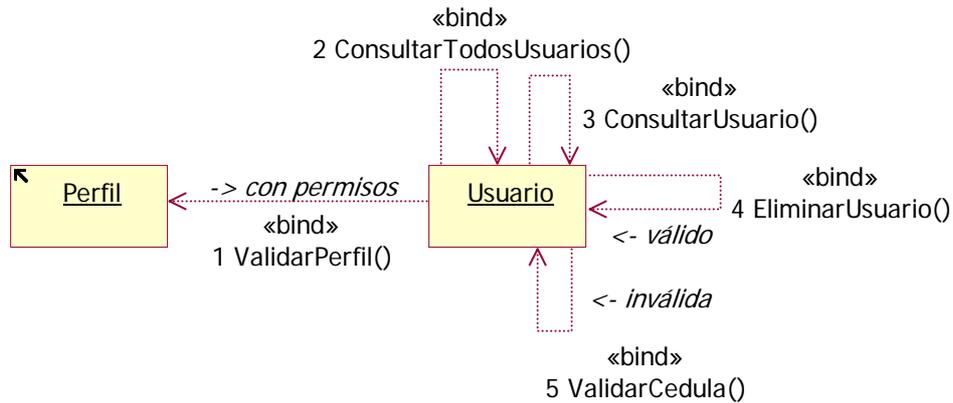
Modelo de Servicios – Modificar Usuario Sin usuarios



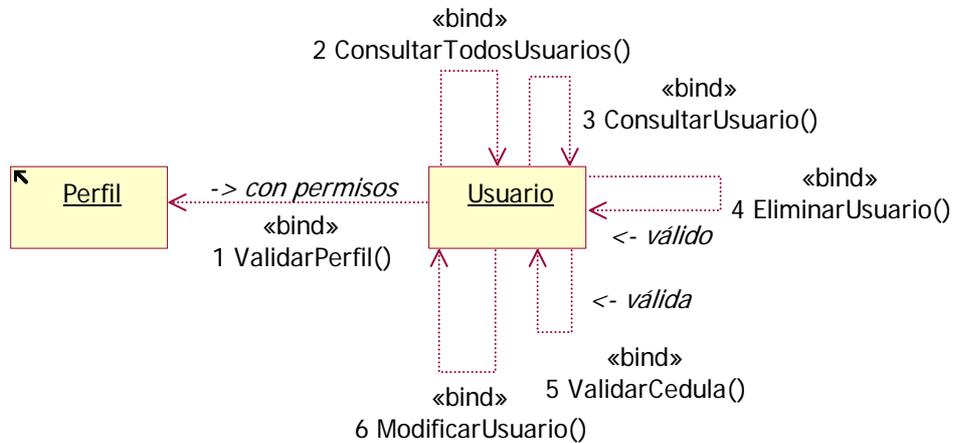
Modelo de Servicios – Modificar Usuario Nombre Inválido



Modelo de Servicios – Modificar Usuario Cedula Inválida

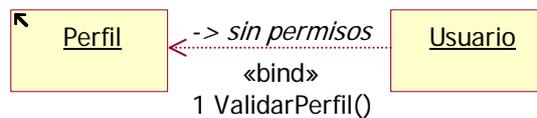


Modelo de Servicios – Modificar Usuario

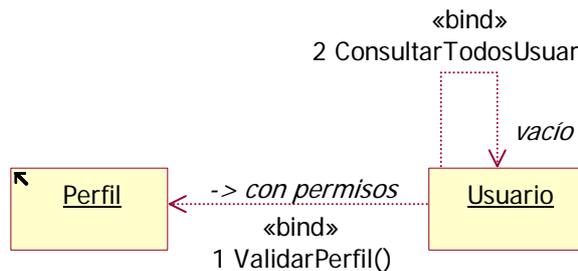


- Eliminar usuario

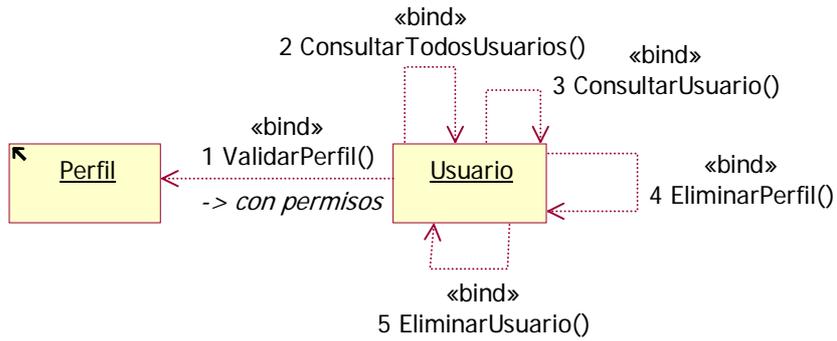
Modelo de Servicios – Modificar Usuario Sin permisos



Modelo de Servicios – Eliminar Usuario sin usuarios



Modelo de Servicios – Eliminar Usuario

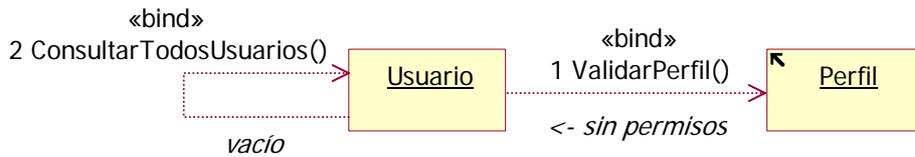


- Consultar perfil

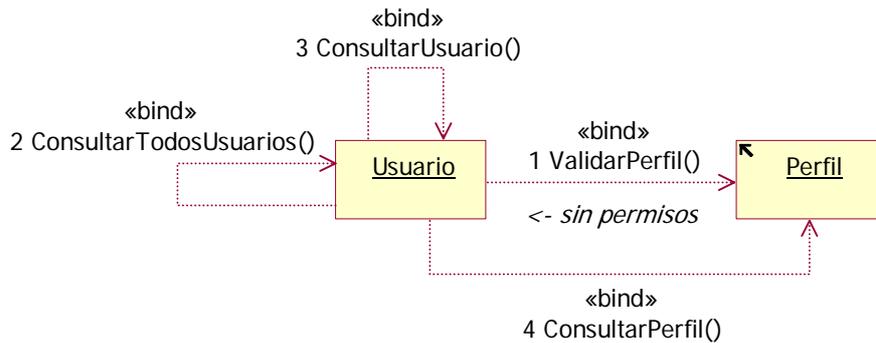
Modelo de Servicios – Consultar Perfil sin permisos



Modelo de Servicios – Consultar Perfil sin usuarios



Modelo de Servicios – Consultar Perfil

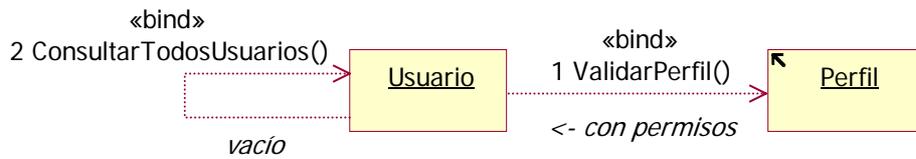


- Modificar perfil

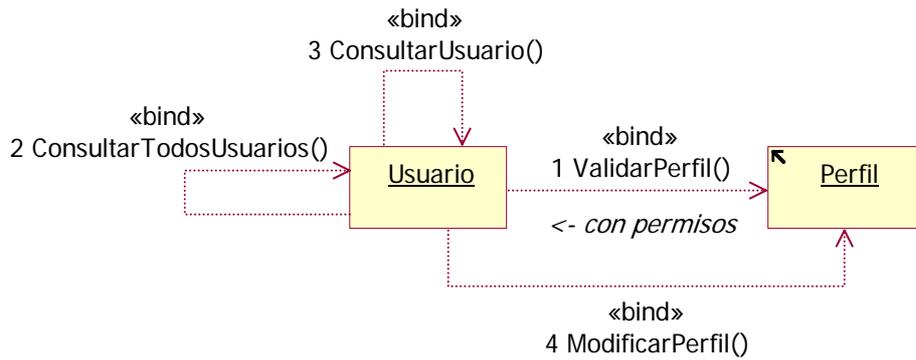
Modelo de Servicios – Modificar Perfil sin permisos



Modelo de Servicios – Modificar Perfil sin usuarios

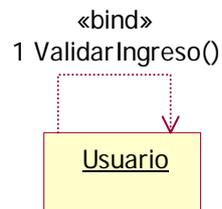


Modelo de Servicios – Modificar Perfil



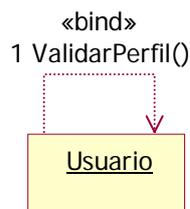
- Validar ingreso

Modelo de Servicios – Validar ingreso



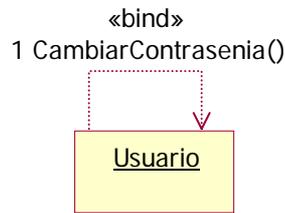
- Validar perfil

Modelo de Servicios – Validar Perfil



- Cambiar contraseña

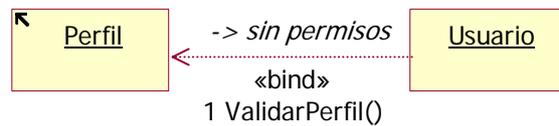
Modelo de Servicios – Cambiar Contraseña



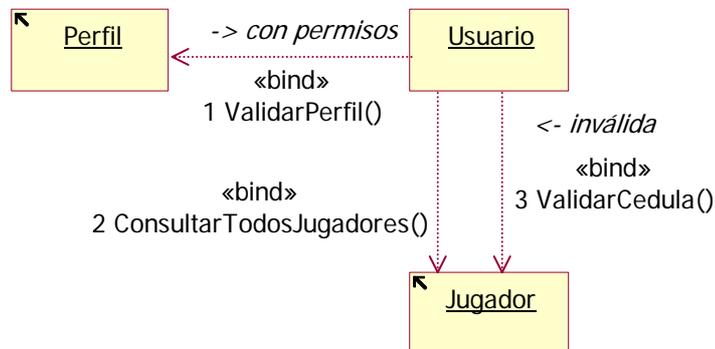
- Paquete Jugador

- Ingresar Jugador

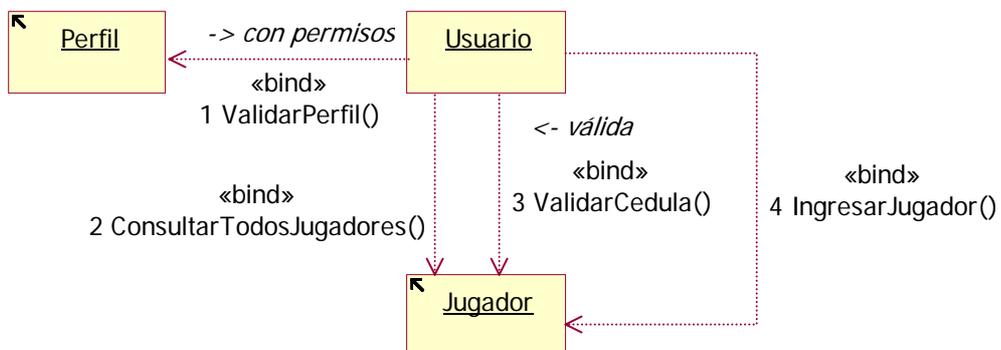
Modelo de Servicios –Ingresar Jugador sin permisos



Modelo de Servicios –Ingresar Jugador Cédula inválida

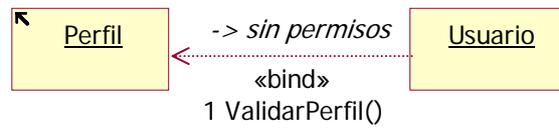


Modelo de Servicios –Ingresar Jugador

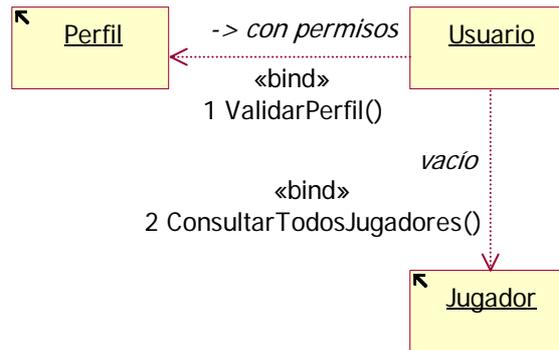


- Consultar Jugador

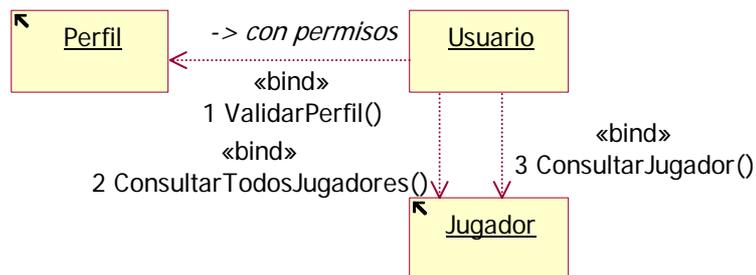
Modelo de Servicios – Consultar Jugador sin permisos



Modelo de Servicios – Consultar Jugador sin jugadores

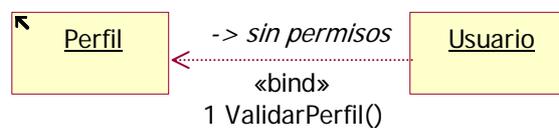


Modelo de Servicios – Consultar Jugador

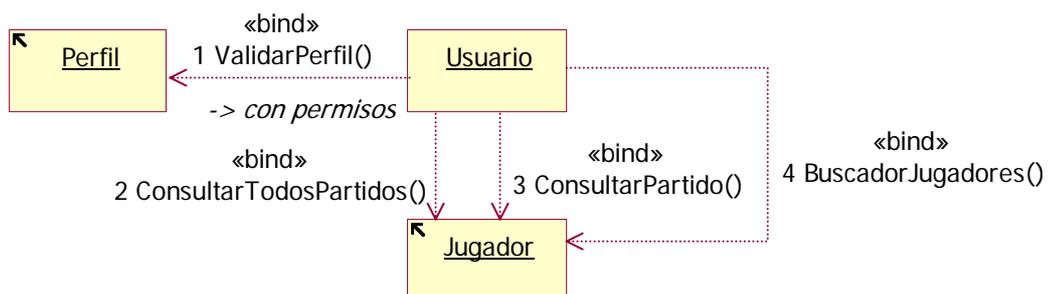


- Buscador Jugadores

Modelo de Servicios – Buscador Jugadores sin permiso

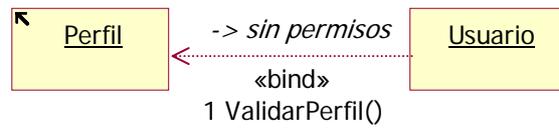


Modelo de Servicios – Buscador Jugadores

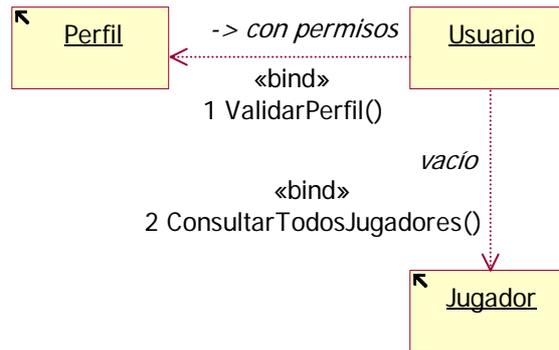


- Modificar Jugador

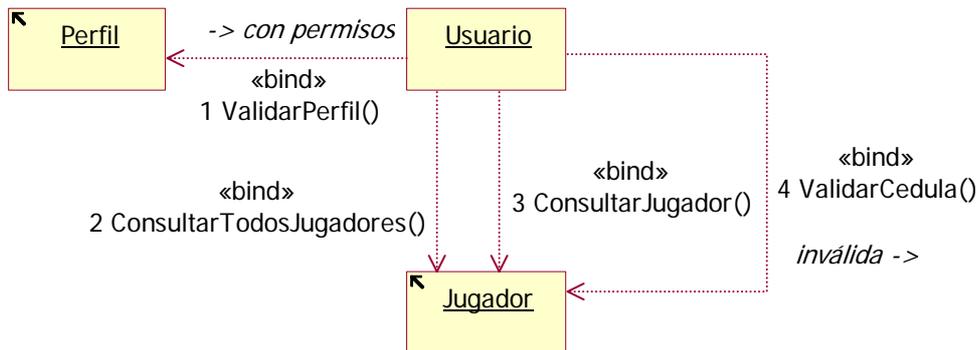
Modelo de Servicios – Modificar Jugador sin permisos



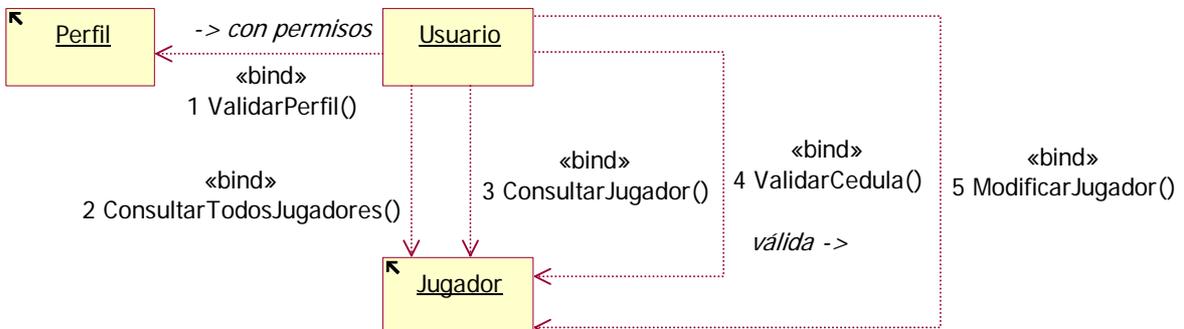
Modelo de Servicios – Modificar Jugador sin jugadores



Modelo de Servicios – Modificar Jugador Cédula inválida

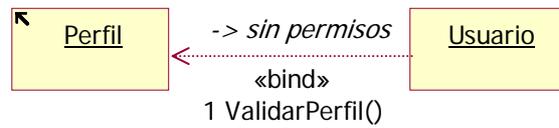


Modelo de Servicios – Modificar Jugador

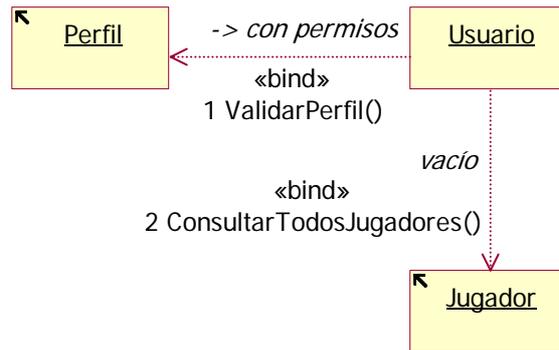


- Eliminar Jugador

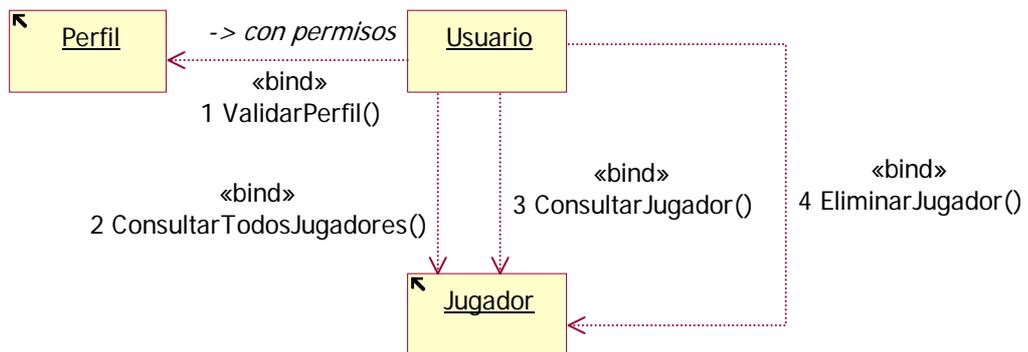
Modelo de Servicios – Eliminar Jugador sin permisos



Modelo de Servicios – Eliminar Jugador sin jugadores



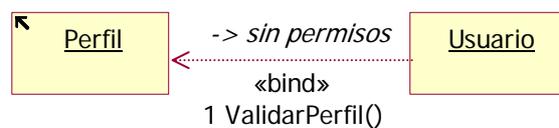
Modelo de Servicios – Eliminar Jugador



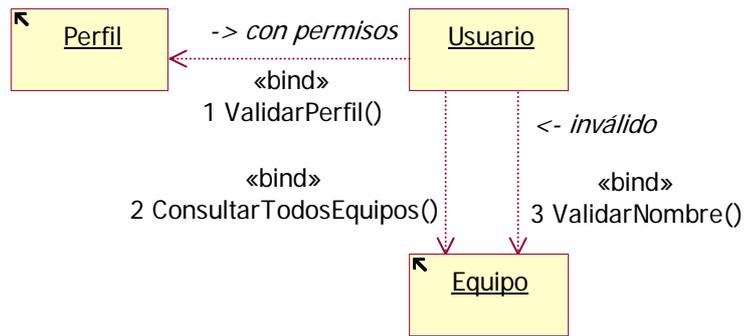
- Paquete Equipo

- Ingresar Equipo

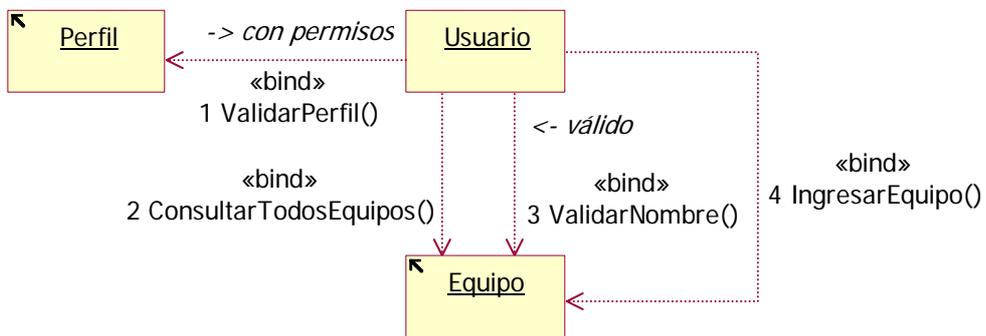
Figura 4.175: Modelo de Servicios –Ingresar Equipo sin permisos



Modelo de Servicios –Ingresar Equipo Nombre inválido

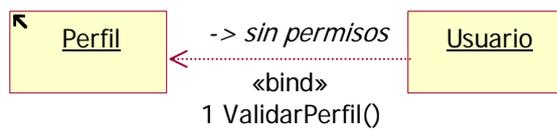


Modelo de Servicios –Ingresar Equipo

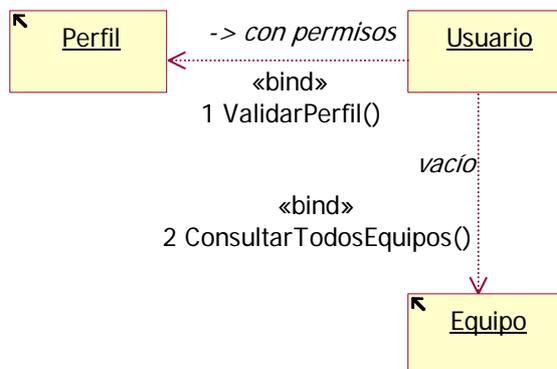


- Consultar Equipo

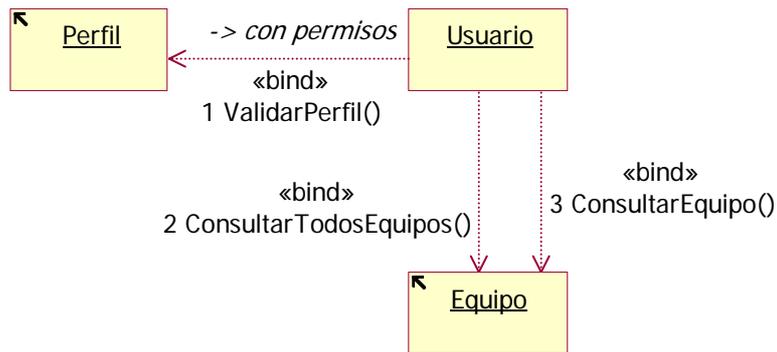
Modelo de Servicios –Consultar Equipo sin permisos



Modelo de Servicios –Consultar Equipo sin equipos

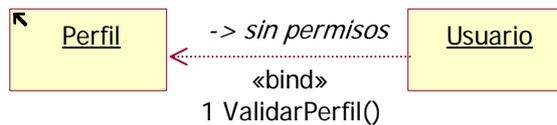


Modelo de Servicios – Consultar Equipo

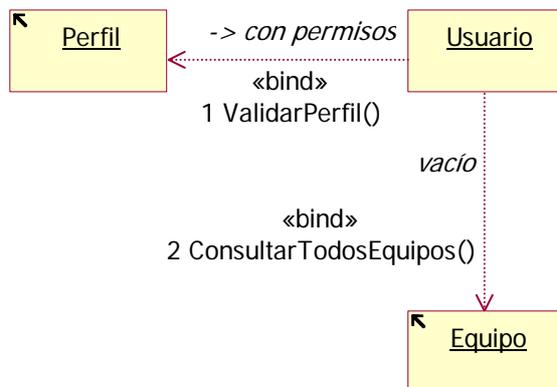


○ Modificar Equipo

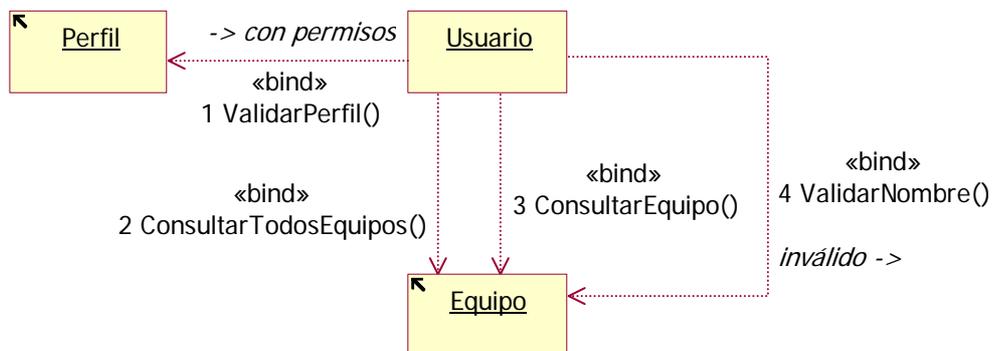
Modelo de Servicios – Modificar Equipo sin permisos



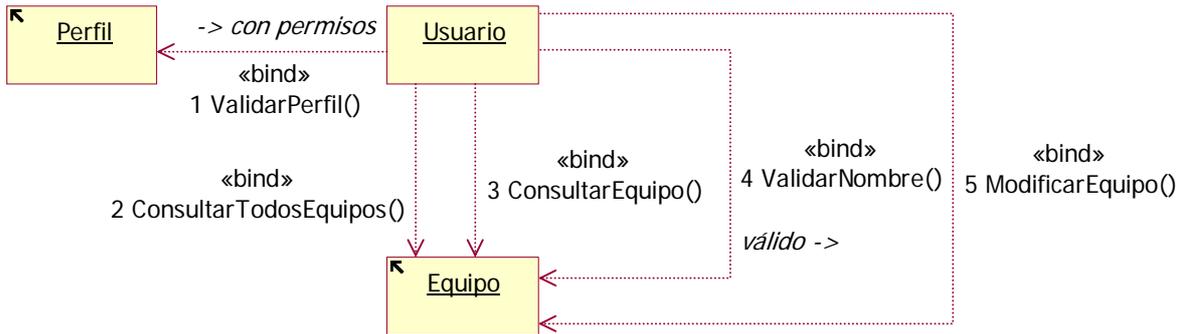
Modelo de Servicios – Modificar Equipo sin Equipos



Modelo de Servicios – Modificar Equipo Cédula inválida

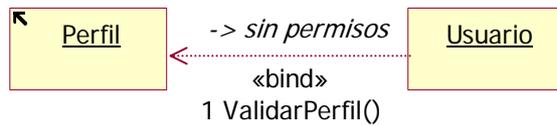


Modelo de Servicios – Modificar Equipo

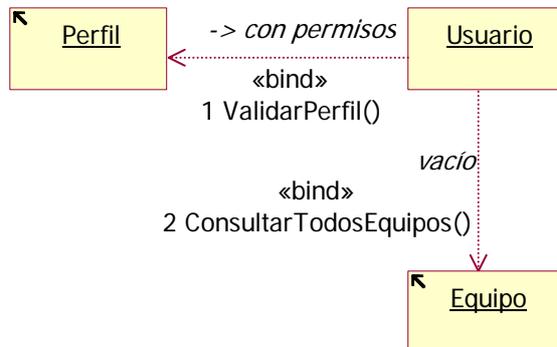


○ Eliminar Equipo

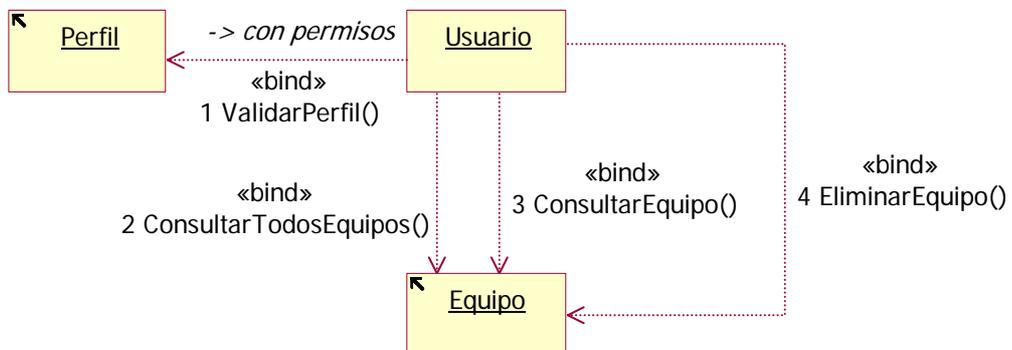
Modelo de Servicios – Eliminar Equipo sin permisos



Modelo de Servicios – Eliminar Equipo sin Equipos

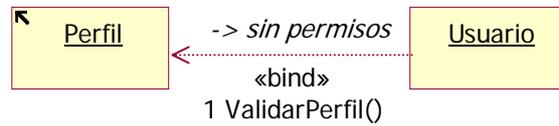


Modelo de Servicios – Eliminar Equipo

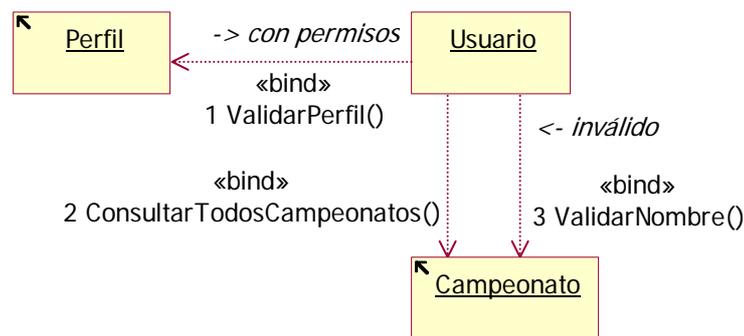


- Paquete Campeonato
 - Ingresar Campeonato

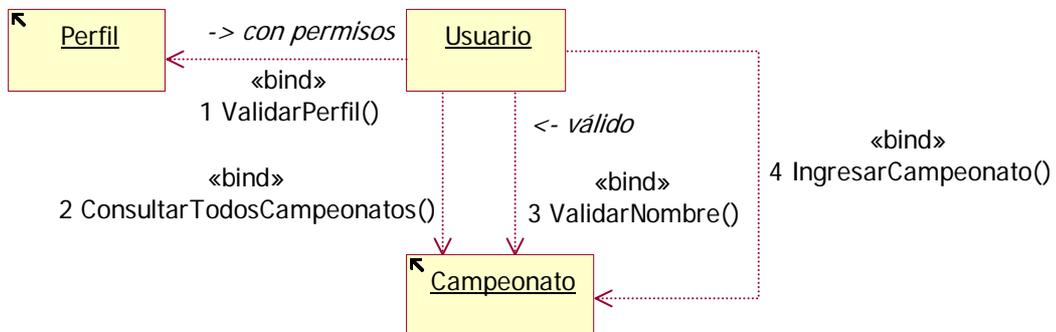
Modelo de Servicios –Ingresar Campeonato sin permisos



Modelo de Servicios –Ingresar Campeonato Nombre inválido

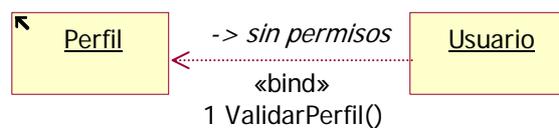


Modelo de Servicios –Ingresar Campeonato

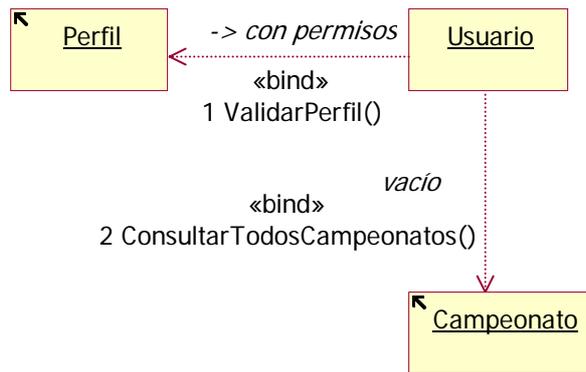


- Consultar Campeonato

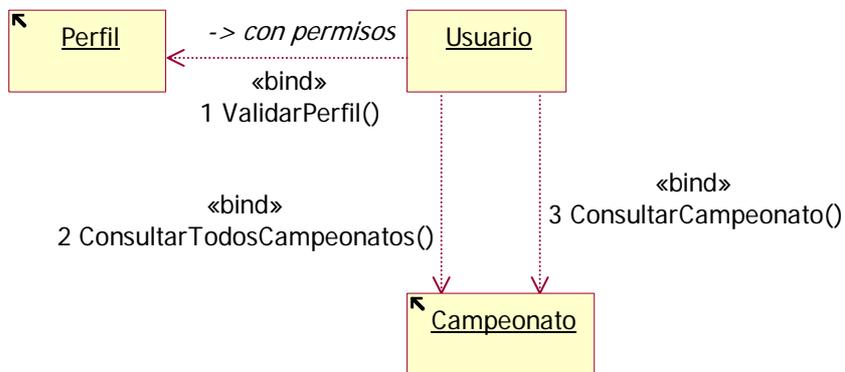
Modelo de Servicios –Consultar Campeonato sin permisos



Modelo de Servicios –Consultar Campeonato sin Campeonatos

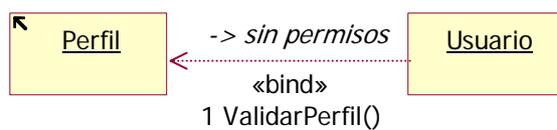


Modelo de Servicios – Consultar Campeonato

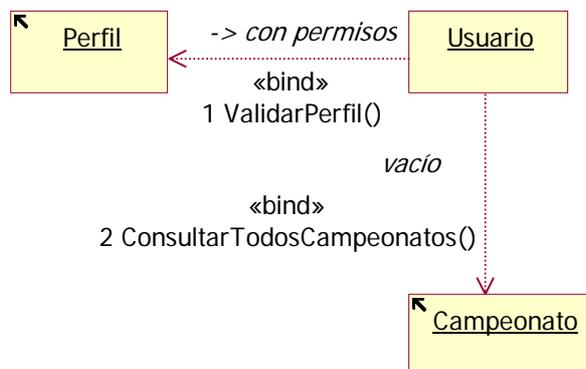


- Modificar Campeonato

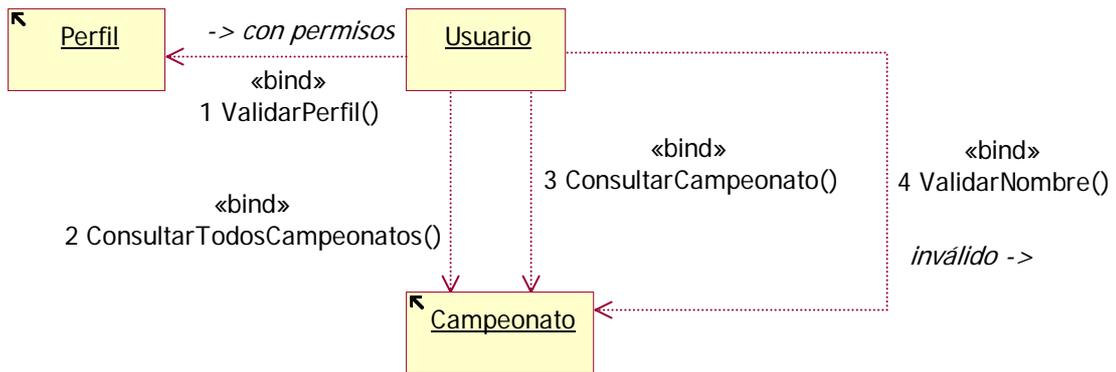
Modelo de Servicios –Modificar Campeonato sin permisos



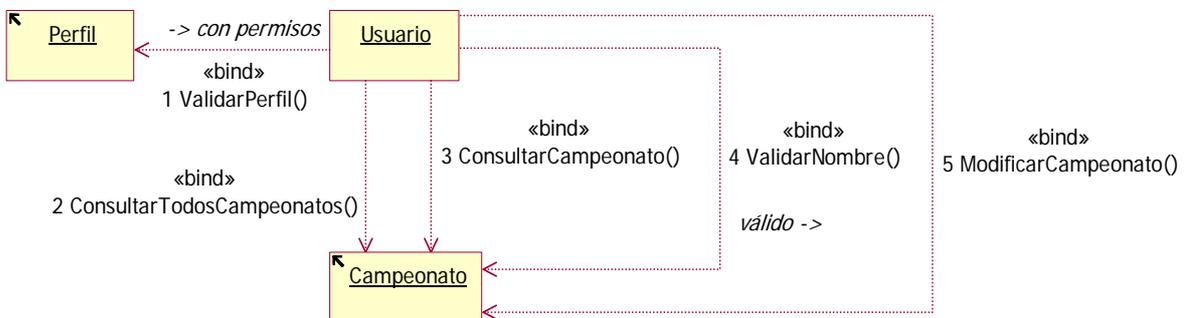
Modelo de Servicios –Modificar Campeonato sin Campeonatos



Modelo de Servicios – Modificar Campeonato Cédula inválida

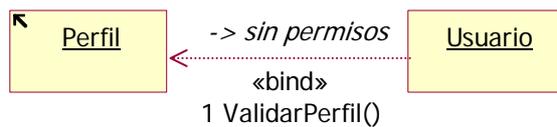


Modelo de Servicios – Modificar Campeonato

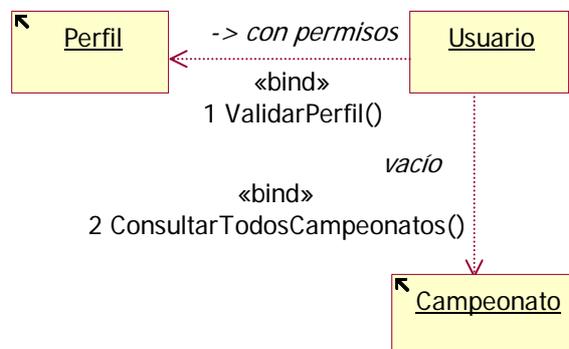


- Eliminar Campeonato

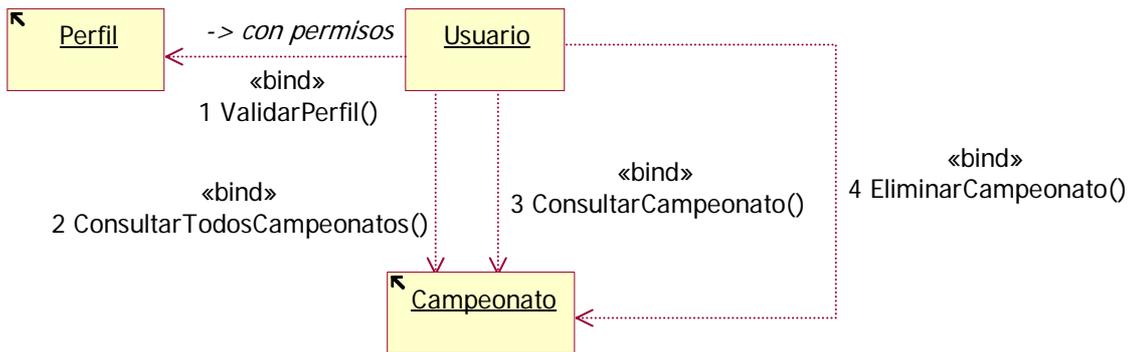
Modelo de Servicios – Eliminar Campeonato sin permisos



Modelo de Servicios – Eliminar Campeonato sin Campeonatos



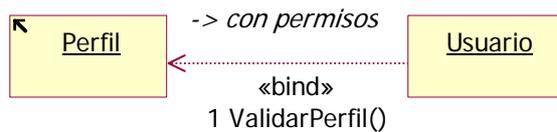
Modelo de Servicios – Eliminar Campeonato



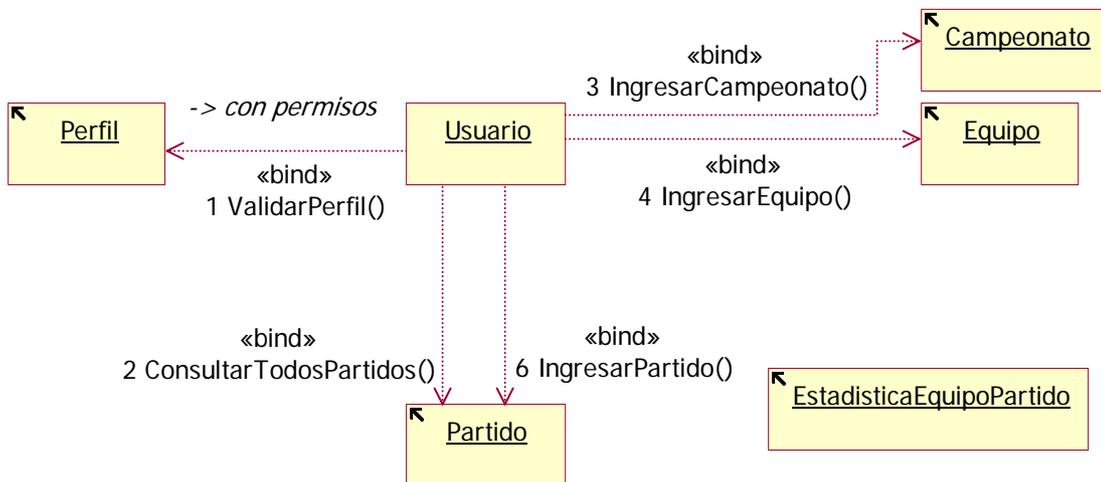
- Paquete Partido

- Ingresar Partido

Modelo de Servicios – Ingresar Partido sin permisos



Modelo de Servicios – Ingresar Partido

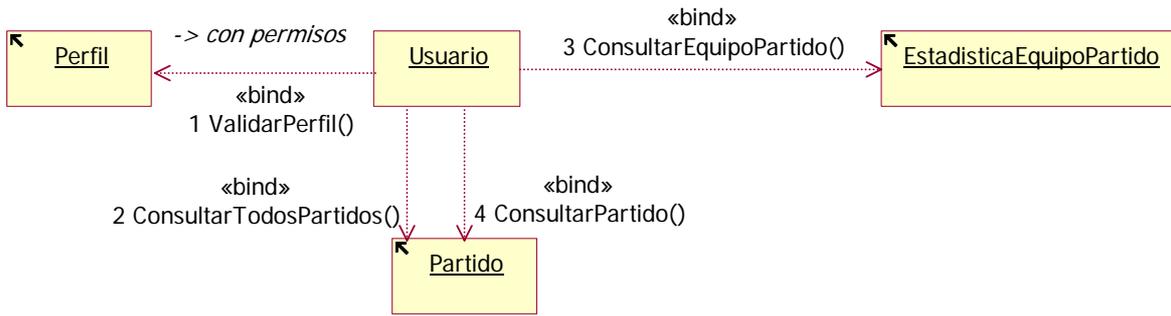


- Consultar Partido

Modelo de Servicios – Consultar Partido sin permisos

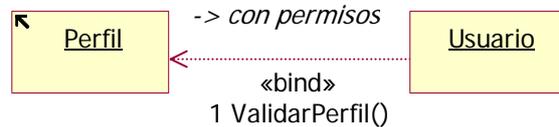


Modelo de Servicios – Consultar Partido

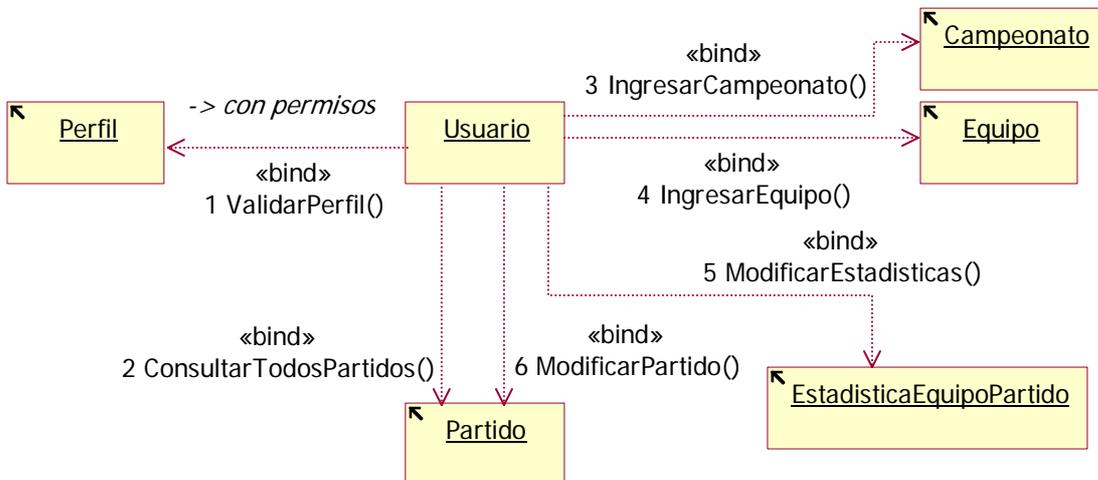


○ Modificar Partido

Modelo de Servicios – Modificar Partido sin permisos



Modelo de Servicios – Modificar Partido



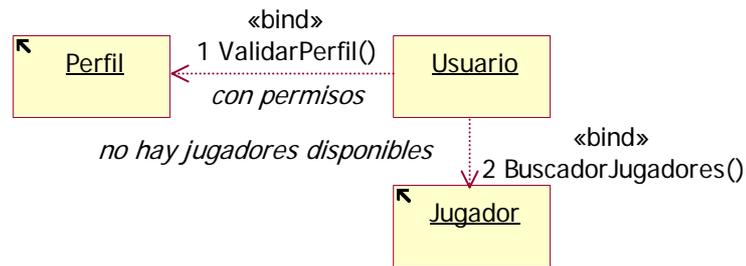
● Paquete Información del Partido

○ Iniciar Periodo Jugador

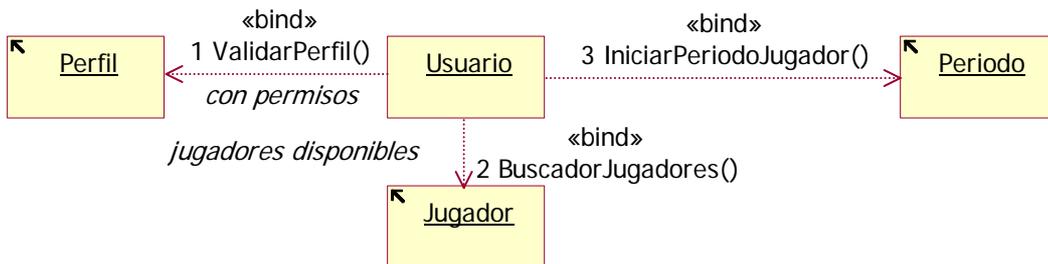
Modelo de Servicios – Iniciar Periodo Jugador sin permisos



Modelo de Servicios – Iniciar Periodo Jugador sin jugadores



Modelo de Servicios – Iniciar Periodo Jugador

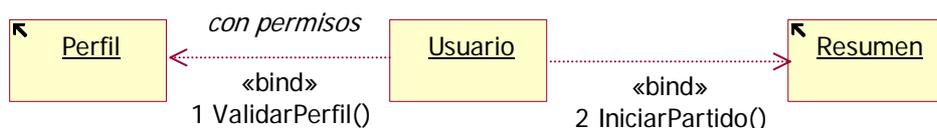


o Iniciar Partido

Modelo de Servicios – Iniciar Partido sin permisos



Modelo de Servicios – Iniciar Partido

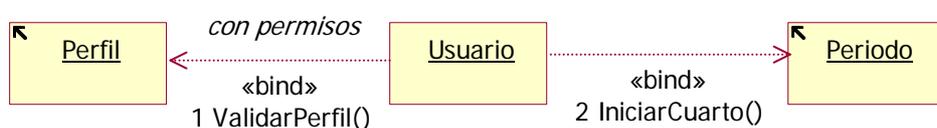


o Iniciar Cuarto

Modelo de Servicios – Iniciar Cuarto sin permisos

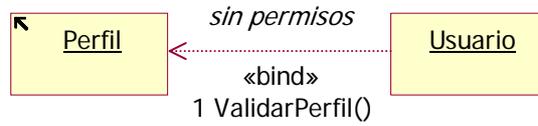


Modelo de Servicios – Iniciar Cuarto

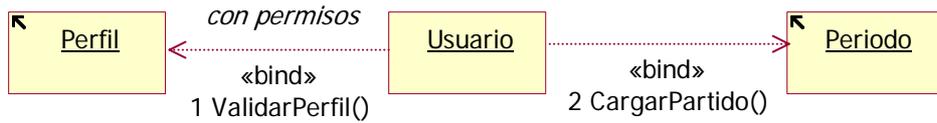


- Cargar Partido

Modelo de Servicios – Cargar Partido sin permisos



Modelo de Servicios – Cargar Partido

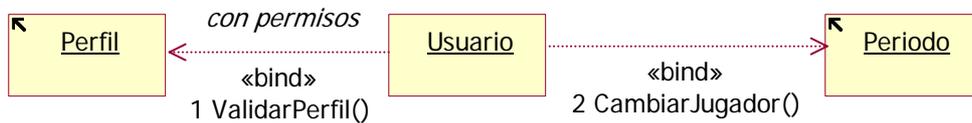


- Cambiar Jugador

Modelo de Servicios – Cambiar Jugador sin permisos



Modelo de Servicios – Cambiar Jugador

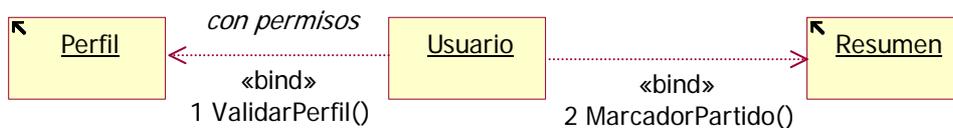


- Marcador Partido

Modelo de Servicios – Marcador Partido sin permisos



Modelo de Servicios – Marcador Partido

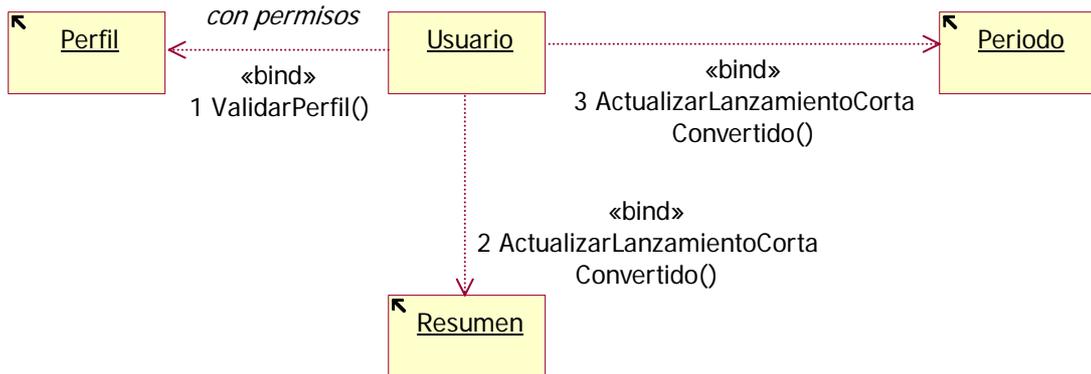


- Actualizar Lanzamiento Corta Convertido

Modelo de Servicios – Actualizar Lanzamiento Corta Convertido sin permisos



Modelo de Servicios – Actualizar Lanzamiento Corta Convertido

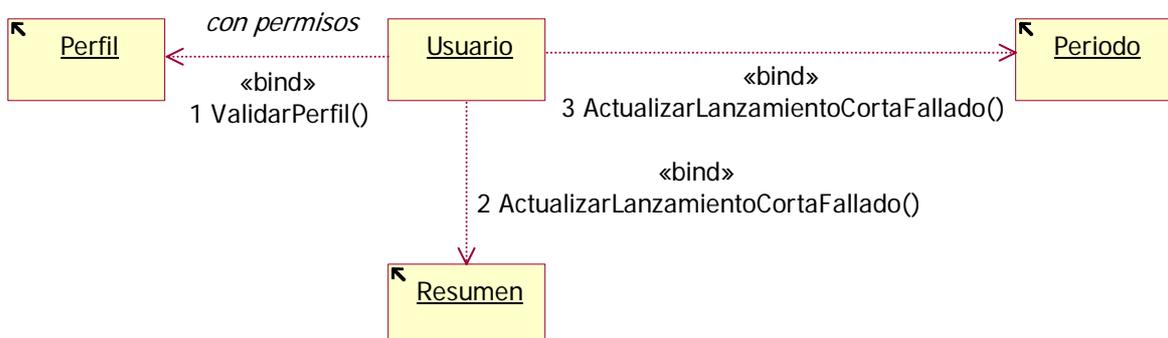


- Actualizar Lanzamiento Corta Fallado

Modelo de Servicios – Actualizar Lanzamiento Corta Fallado sin permisos



Modelo de Servicios – Actualizar Lanzamiento Corta Fallado

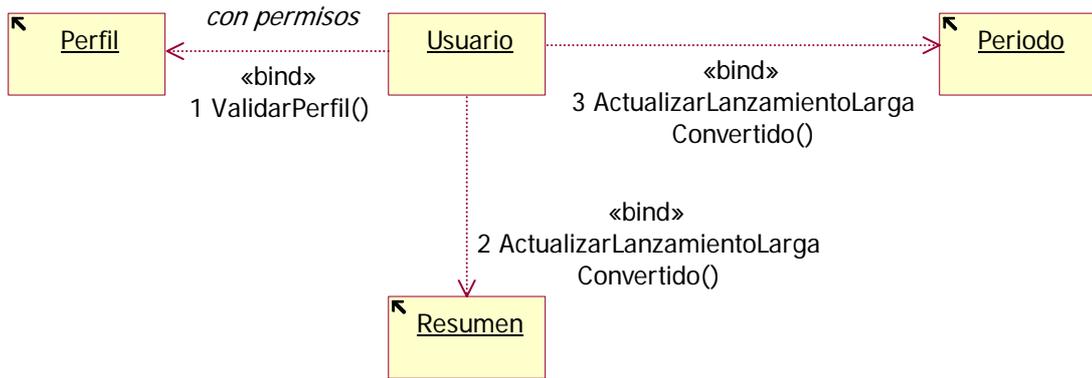


- Actualizar Lanzamiento Larga Convertido

Modelo de Servicios – Actualizar Lanzamiento Larga Convertido sin permisos



Modelo de Servicios – Actualizar Lanzamiento Larga Convertido

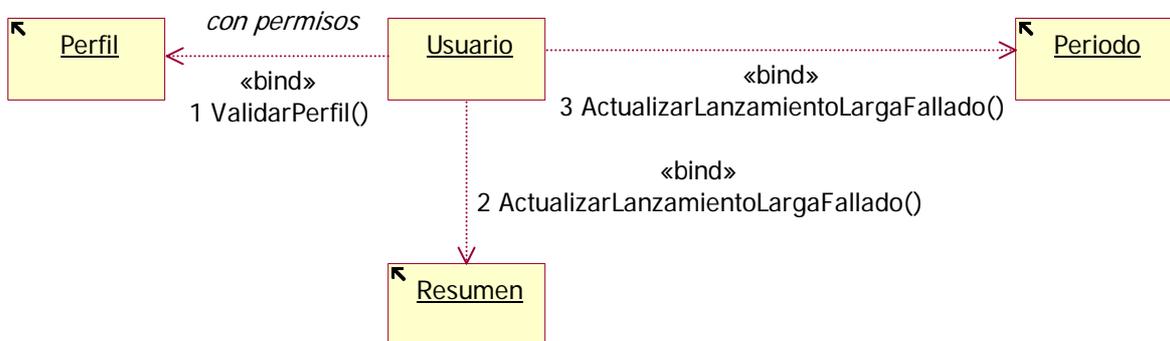


- Actualizar Lanzamiento Larga Fallado

Modelo de Servicios – Actualizar Lanzamiento Larga Fallado sin permisos



Modelo de Servicios – Actualizar Lanzamiento Larga Fallado

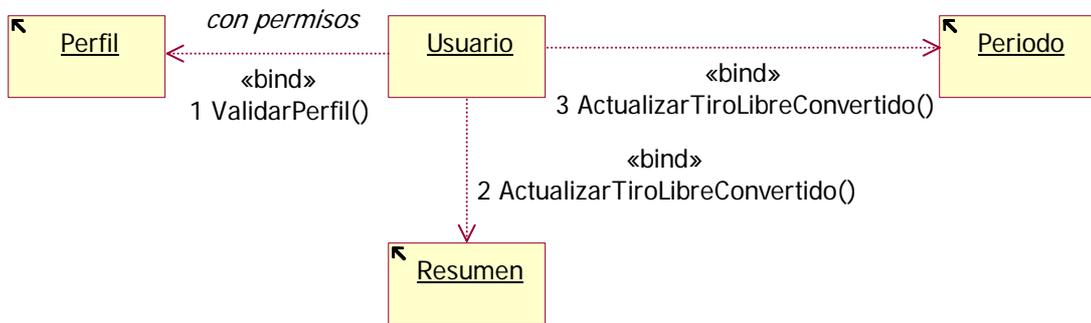


- Actualizar Tiro Libre Convertido

Modelo de Servicios – Actualizar Lanzamiento Tiro Libre Convertido sin permisos



Modelo de Servicios – Actualizar Lanzamiento Tiro Libre Convertido

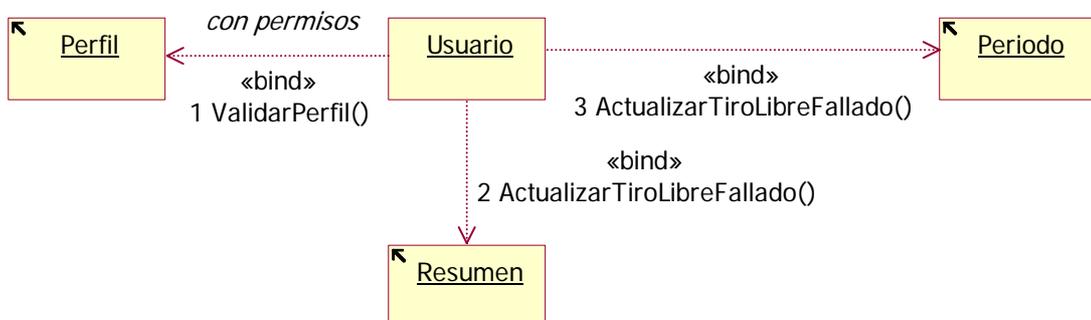


- Actualizar Tiro Libre Fallado

Modelo de Servicios – Actualizar Lanzamiento Tiro Libre Fallado sin permisos

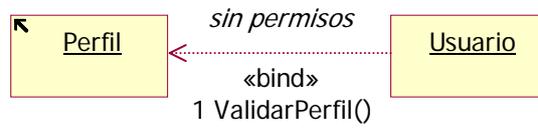


Modelo de Servicios – Actualizar Lanzamiento Tiro Libre Fallado

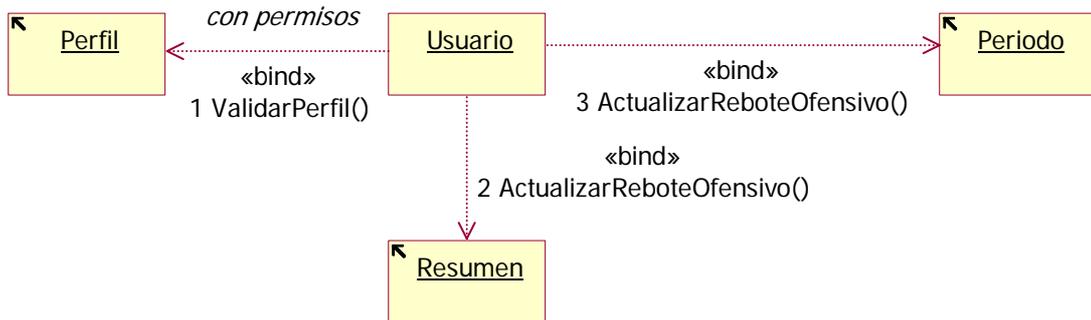


- Actualizar Rebote Ofensivo

Modelo de Servicios – Actualizar Rebote Ofensivo sin permisos



Modelo de Servicios – Actualizar Rebote Ofensivo

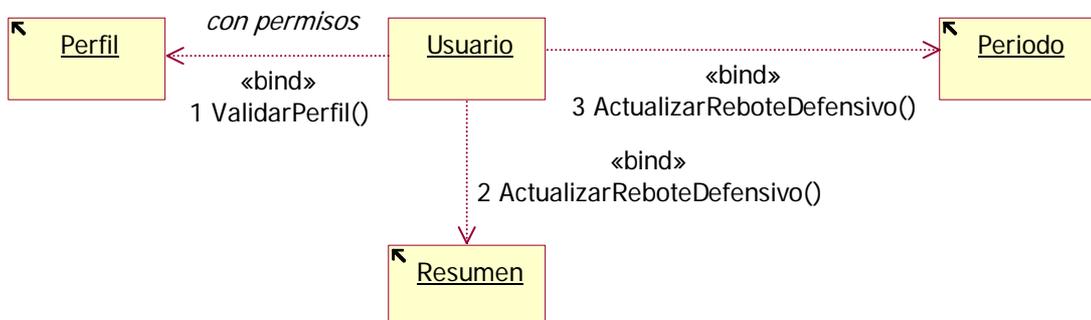


- Actualizar Rebote Defensivo

Modelo de Servicios – Actualizar Rebote Defensivo sin permisos



Modelo de Servicios – Actualizar Rebote Defensivo

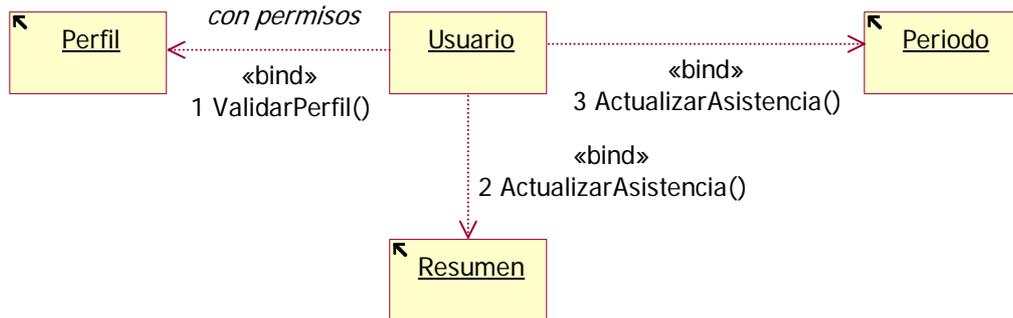


- Actualizar Asistencia

Modelo de Servicios – Actualizar Asistencia sin permisos



Modelo de Servicios – Actualizar Asistencia

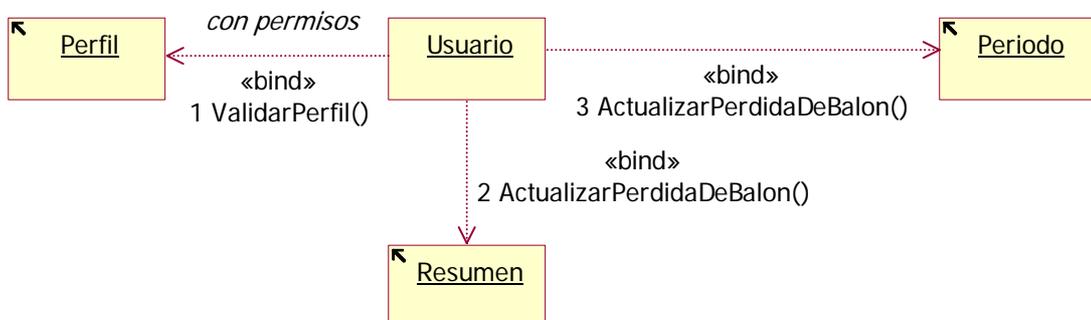


- Actualizar Perdida de Balón

Modelo de Servicios – Actualizar Pérdida de Balón sin permisos



Modelo de Servicios – Actualizar Pérdida de Balón

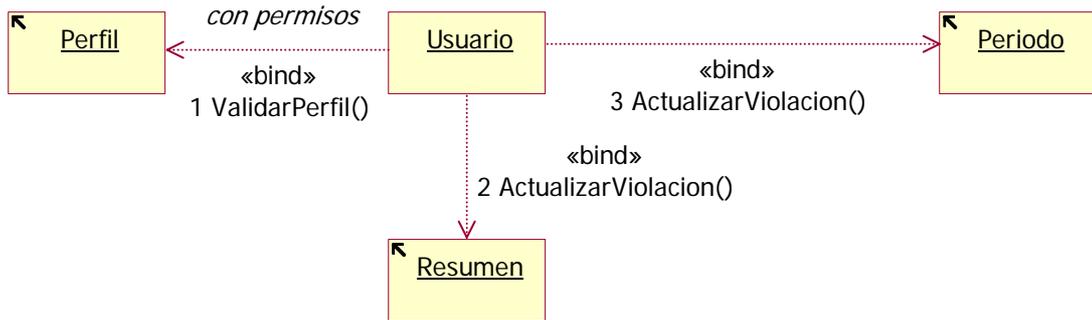


- Actualizar Violación

Modelo de Servicios – Actualizar Violación sin permisos



Modelo de Servicios – Actualizar Violación

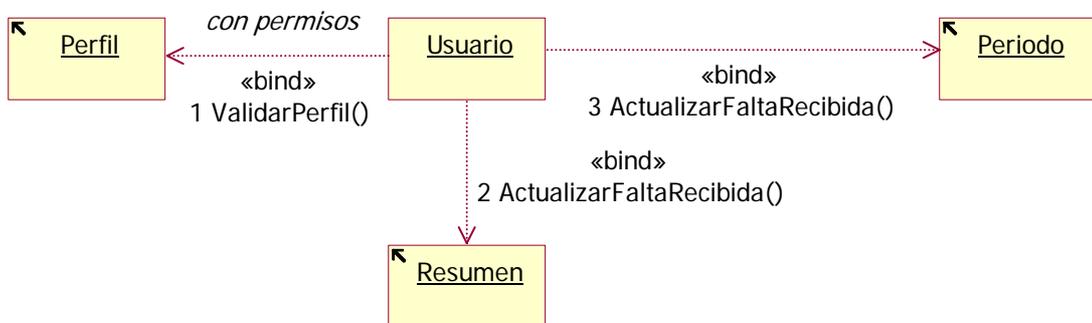


- Actualizar Falta Recibida

Modelo de Servicios – Actualizar Falta Recibida sin permisos



Modelo de Servicios – Actualizar Falta Recibida

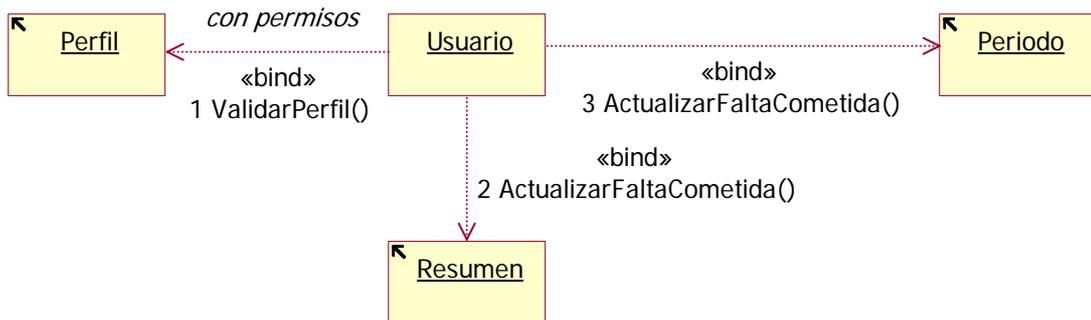


- Actualizar Falta Cometida

Modelo de Servicios – Actualizar Falta Cometida sin permisos



Modelo de Servicios – Actualizar Falta Cometida

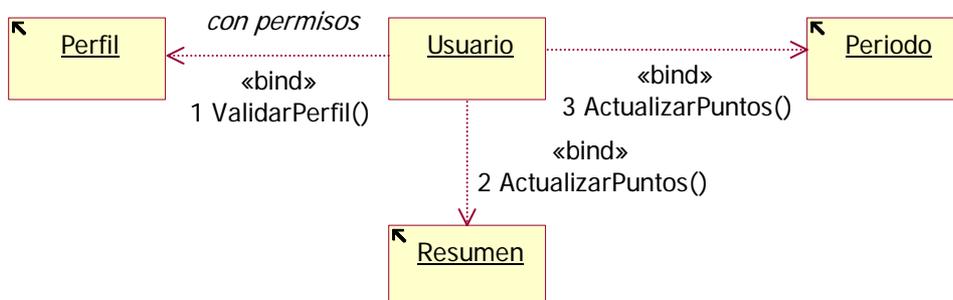


- Actualizar Puntos

Modelo de Servicios – Actualizar Puntos sin permisos



Modelo de Servicios – Actualizar Puntos



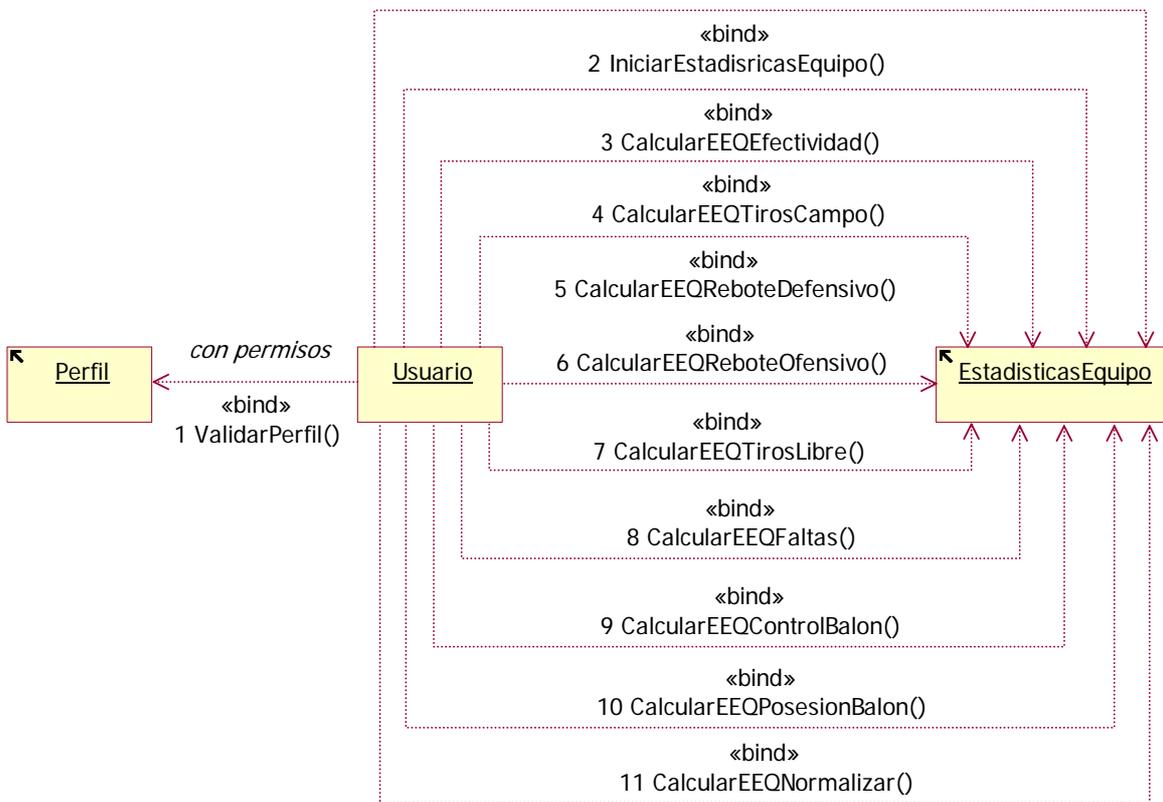
- Paquete Estadísticas

- EEQ

Modelo de Servicios – EEQ sin permisos

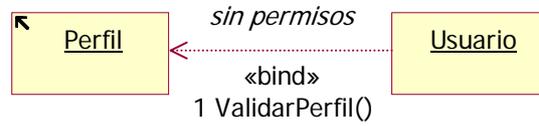


Modelo de Servicios – EEQ

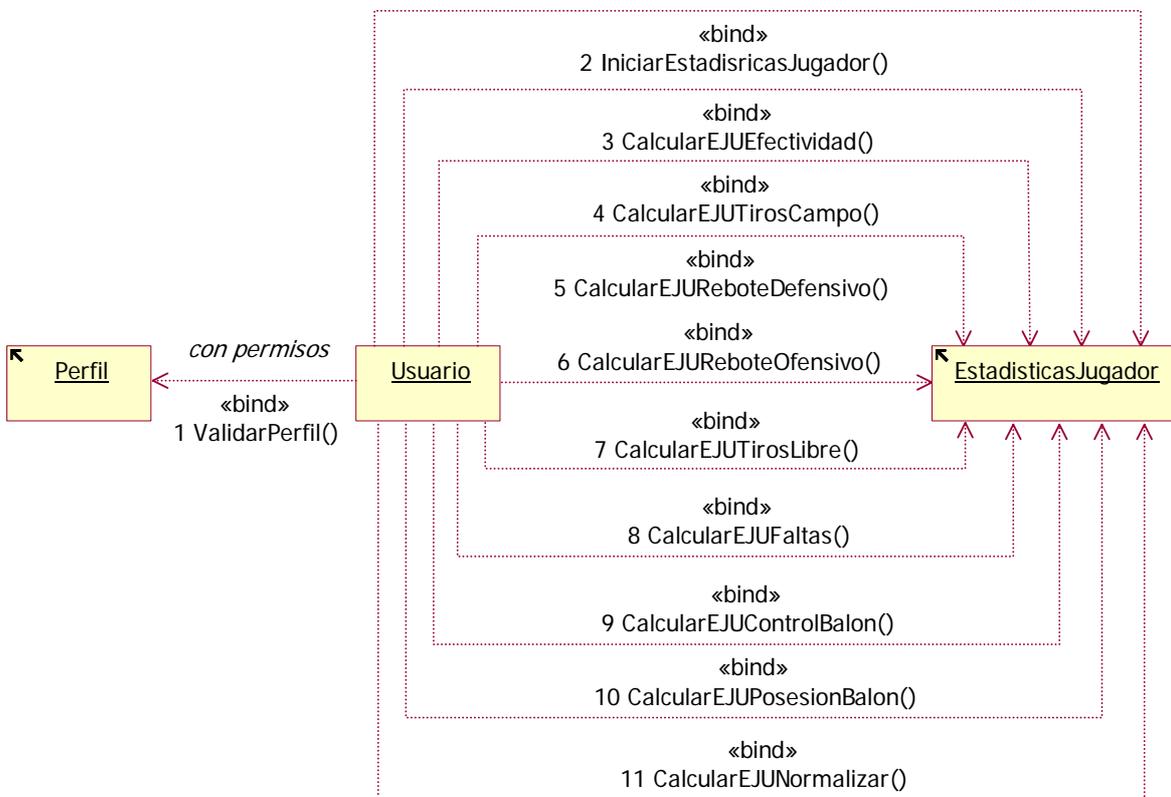


o EJU

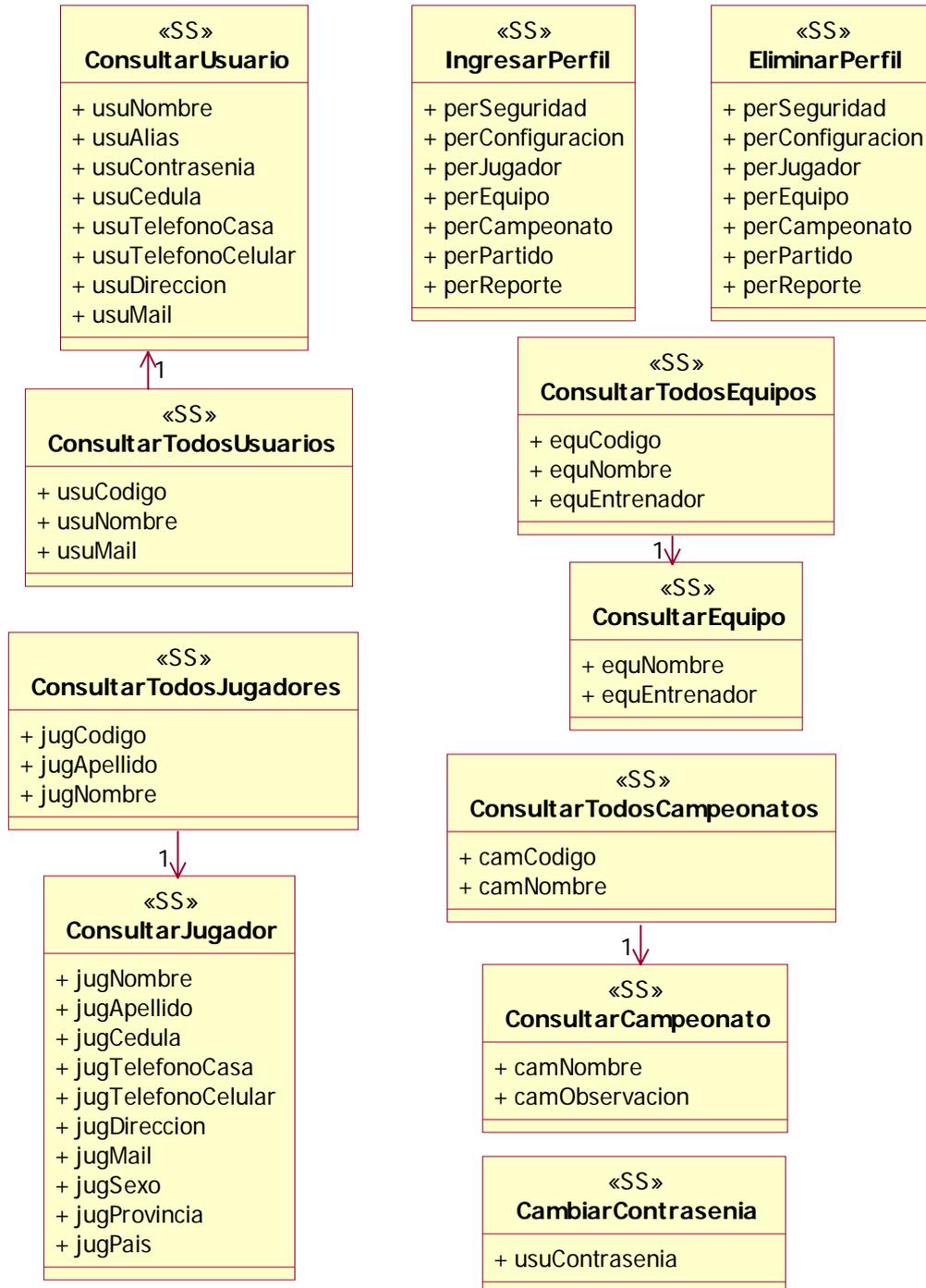
Modelo de Servicios – EJU sin permisos



Modelo de Servicios – EJU



En el aspecto de Hipertexto MIDAS propone la elaboración de tres modelos, los cuales son:



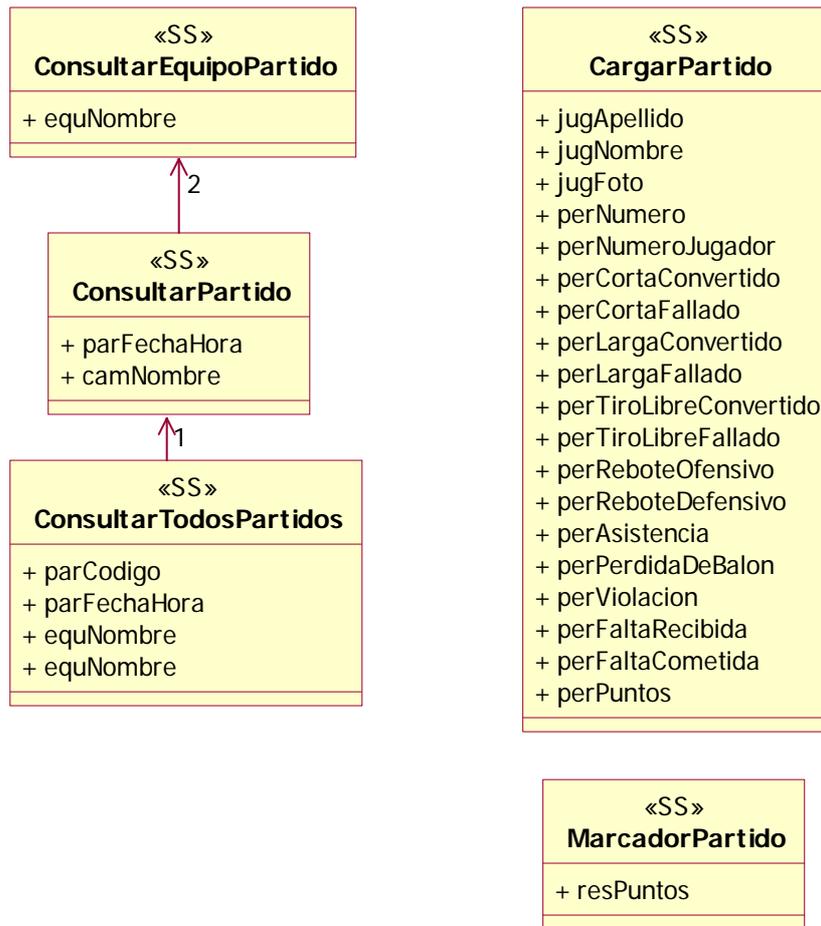


Figura 4.12: Modelo de Fragmentos

Modelos Especificos de la Plataforma – PSM

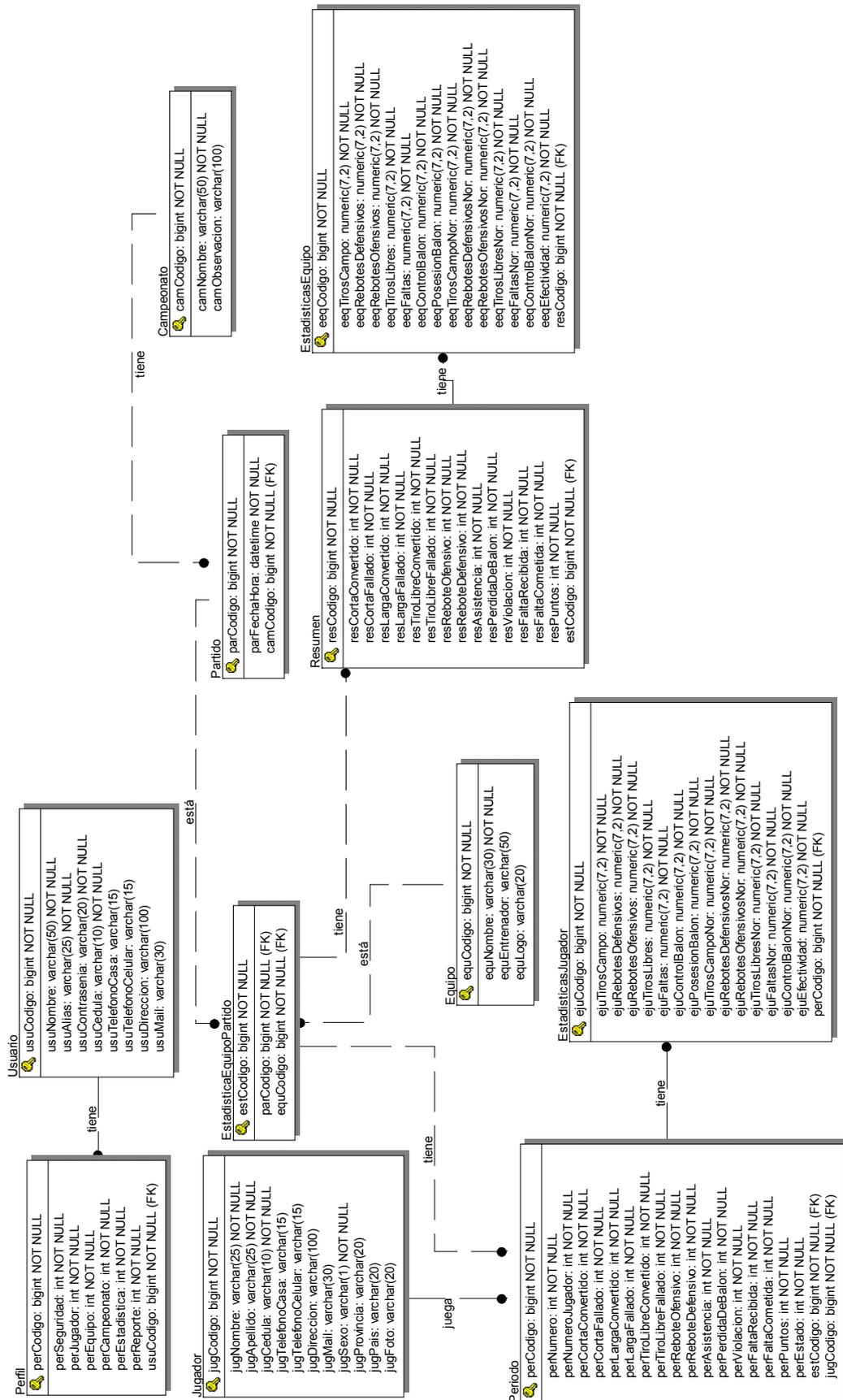


Figura 4.15: Modelo Objeto Relacional

El modelo presentado en la *Figura 4.15* representa el último modelo del aspecto de contenido en la metodología Midas.

Este es el que va a ser generado en la base de datos creando los esquemas, tablas, campos y relaciones que va a utilizar el sistema.

Modelo XML Schemas

El modelo específico de la plataforma XML Schemas, modela los fragmentos conceptuales producto de la codificación y compilación de los fragmentos funcionales y estructurales. Los siguientes modelos del SIW permiten diferenciar claramente los slice funcionales y estructurales.

Web Service Usuario

```
<s:element name="ValidarIngreso">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="strAlias" type="s:string"
      />
      <s:element minOccurs="0" maxOccurs="1" name="strContrasenia"
      type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

Web Service Campeonato

```
<s:element name="ConsultarCampeonato">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="lngCodigo" type="s:long" />
    </s:sequence>
  </s:complexType>
</s:element>
```

Web Service Equipo

```
<s:element name="ModificarEquipo">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="codigo" type="s:long" />
      <s:element minOccurs="0" maxOccurs="1" name="nombre" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

```

<s:element minOccurs="0" maxOccurs="1" name="entrenador"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="logo" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>

```

Web Service Jugador

```

<s:element name="IngresarJugador">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="nombre" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="apellido" type="s:string"
/>
<s:element minOccurs="0" maxOccurs="1" name="cedula" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="teca" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="tece" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="direccion" type="s:string"
/>
<s:element minOccurs="0" maxOccurs="1" name="mail" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="sexo" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="provincia" type="s:string"
/>
<s:element minOccurs="0" maxOccurs="1" name="pais" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="foto" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>

```

Web Service Partido

```

<s:element name="ModificarPartido">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="codigo" type="s:long" />
<s:element minOccurs="0" maxOccurs="1" name="fechaHora"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="nombreCampeonato"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="NombreEquipoL"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="NombreEquipoV"
type="s:string" />
<s:element minOccurs="1" maxOccurs="1" name="codigoEstEquLocal"
type="s:long" />
<s:element minOccurs="1" maxOccurs="1" name="codigoEstEquVisitante"
type="s:long" />
</s:sequence>
</s:complexType>
</s:element>

```

Web Service Periodo

```
<s:element name="ActualizarLanzamientoCortaConvertido">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="valor" type="s:int" />
      <s:element minOccurs="1" maxOccurs="1" name="estCodigo" type="s:long"
/>
      <s:element minOccurs="1" maxOccurs="1" name="jugCodigo" type="s:long"
/>
      <s:element minOccurs="1" maxOccurs="1" name="tipo" type="s:int" />
    </s:sequence>
  </s:complexType>
</s:element>
```

Web Service Resumen

```
<s:element name="MarcadorPartido">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="codigoPartido"
type="s:long" />
    </s:sequence>
  </s:complexType>
</s:element>
```

Web Service Estadística

```
<s:element name="EJU">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="estCodigo" type="s:long"
/>
      <s:element minOccurs="1" maxOccurs="1" name="jugCodigo" type="s:long"
/>
    </s:sequence>
  </s:complexType>
</s:element>
```

Modelo WSDL

El modelo WSDL muestra la interfaz de un Servicio Web en formato XML, se han extraído las partes más importantes del documento WSDL del Servicio Web Equipo tomando en cuenta las funciones de mantenimiento (consultar, ingresar, modificar, eliminar), para comprender el alcance del modelo.

Web Service Equipo

Cabecera WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
```

Tipos WSDL para intercambio de mensajes

```
<wsdl:types>
  <s:schema elementFormDefault="qualified"
targetNamespace="http://tempuri.org/">
    <s:element name="ConsultarTodosEquipos">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="nombre" type="s:string"
/>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="ConsultarEquipo">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="lngCodigo"
type="s:long" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="IngresarEquipo">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="nombre" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="entrenador"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="logo" type="s:string"
/>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="ModificarEquipo">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="codigo" type="s:long"
/>
          <s:element minOccurs="0" maxOccurs="1" name="nombre" type="s:string"
/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
</wsdl:types>
```

```

    <s:element minOccurs="0" maxOccurs="1" name="entrenador"
type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="logo" type="s:string"
/>
  </s:sequence>
</s:complexType>
</s:element>
<s:element name="EliminarEquipo">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="codigo" type="s:long"
/>
    </s:sequence>
  </s:complexType>
</s:element>
</s:schema>
</wsdl:types>

```

Mensajes WSDL, datos que se transmiten

```

  <wsdl:message name="ConsultarTodosEquiposSoapIn">
<wsdl:part name="parameters" element="tns:ConsultarTodosEquipos" />
</wsdl:message>
<wsdl:message name="ConsultarTodosEquiposSoapOut">
<wsdl:part name="parameters" element="tns:ConsultarTodosEquiposResponse"
/>
</wsdl:message>
  <wsdl:message name="ConsultarEquipoSoapIn">
<wsdl:part name="parameters" element="tns:ConsultarEquipo" />
</wsdl:message>
<wsdl:message name="ConsultarEquipoSoapOut">
<wsdl:part name="parameters" element="tns:ConsultarEquipoResponse" />
</wsdl:message>
  <wsdl:message name="IngresarEquipoSoapIn">
<wsdl:part name="parameters" element="tns:IngresarEquipo" />
</wsdl:message>
<wsdl:message name="IngresarEquipoSoapOut">
<wsdl:part name="parameters" element="tns:IngresarEquipoResponse" />
</wsdl:message>
  <wsdl:message name="ModificarEquipoSoapIn">
<wsdl:part name="parameters" element="tns:ModificarEquipo" />
</wsdl:message>
<wsdl:message name="ModificarEquipoSoapOut">
<wsdl:part name="parameters" element="tns:ModificarEquipoResponse" />
</wsdl:message>
  <wsdl:message name="EliminarEquipoSoapIn">
<wsdl:part name="parameters" element="tns:EliminarEquipo" />
</wsdl:message>
<wsdl:message name="EliminarEquipoSoapOut">
<wsdl:part name="parameters" element="tns:EliminarEquipoResponse" />
</wsdl:message>

```

Operaciones WSDL, acciones permitidas por el servicio Web

```
<wsdl:portType name="EquipoSoap">
  <wsdl:operation name="ConsultarTodosEquipos">
<wsdl:input message="tns:ConsultarTodosEquiposSoapIn" />
<wsdl:output message="tns:ConsultarTodosEquiposSoapOut" />
</wsdl:operation>
  <wsdl:operation name="ConsultarEquipo">
<wsdl:input message="tns:ConsultarEquipoSoapIn" />
<wsdl:output message="tns:ConsultarEquipoSoapOut" />
</wsdl:operation>
  <wsdl:operation name="IngresarEquipo">
<wsdl:input message="tns:IngresarEquipoSoapIn" />
<wsdl:output message="tns:IngresarEquipoSoapOut" />
</wsdl:operation>
  <wsdl:operation name="ModificarEquipo">
<wsdl:input message="tns:ModificarEquipoSoapIn" />
<wsdl:output message="tns:ModificarEquipoSoapOut" />
</wsdl:operation>
  <wsdl:operation name="EliminarEquipo">
<wsdl:input message="tns:EliminarEquipoSoapIn" />
<wsdl:output message="tns:EliminarEquipoSoapOut" />
</wsdl:operation>
</wsdl:portType>
```

WSDL Binding y Acciones SOAP, protocolo y formato de datos

```
<wsdl:binding name="EquipoSoap" type="tns:EquipoSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ConsultarTodosEquipos">
<soap:operation soapAction="http://tempuri.org/ConsultarTodosEquipos"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
  <wsdl:operation name="ConsultarEquipo">
<soap:operation soapAction="http://tempuri.org/ConsultarEquipo"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
  <wsdl:operation name="IngresarEquipo">
<soap:operation soapAction="http://tempuri.org/IngresarEquipo"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
  <wsdl:operation name="ModificarEquipo">
```

```

<soap:operation soapAction="http://tempuri.org/ModificarEquipo"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
  <wsdl:operation name="EliminarEquipo">
<soap:operation soapAction="http://tempuri.org/EliminarEquipo"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="EquipoSoap12" type="tns:EquipoSoap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="ConsultarTodosEquipos">
<soap12:operation soapAction="http://tempuri.org/ConsultarTodosEquipos"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
  <wsdl:operation name="ConsultarEquipo">
<soap12:operation soapAction="http://tempuri.org/ConsultarEquipo"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
  <wsdl:operation name="IngresarEquipo">
<soap12:operation soapAction="http://tempuri.org/IngresarEquipo"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
  <wsdl:operation name="ModificarEquipo">
<soap12:operation soapAction="http://tempuri.org/ModificarEquipo"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>

```

```

</wsdl:operation>
  <wsdl:operation name="EliminarEquipo">
<soap12:operation soapAction="http://tempuri.org/EliminarEquipo"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Equipo">
<wsdl:port name="EquipoSoap" binding="tns:EquipoSoap">
<soap:address location="http://localhost/Basys/Funcionalidad/Equipo.asmx"
/>
</wsdl:port>
  <wsdl:port name="EquipoSoap12" binding="tns:EquipoSoap12">
    <soap12:address
location="http://localhost/Basys/Funcionalidad/Equipo.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Modelo BPEL4WS

La definición de cabeceras del modelo BPEL para servicios Web, representa el paso más importante al determinar el camino de los Servicios Web para evitar fallos, y en el caso de que se produzcan y sean de hardware (Servidor de Servicios Web en Internet), realiza una redirección a las funciones para servidores de respaldo en el caso de que hubieren.

Web Service Usuario: Cabecera BPEL

```

<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/">
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Usuario.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Usuario.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
    <soap address="http://localhost/Basys/Funcionalidad/Usuario.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:UsuarioSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
      <soap address="http://localhost/Basys/Funcionalidad/Usuario.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:UsuarioSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
    </discovery>

```

Web Service Campeonato: Cabecera BPEL

```
<?xml version="1.0" encoding="utf-8"?>
<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/" >
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Campeonato.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Campeonato.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/Basys/Funcionalidad/Campeonato.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:CampeonatoSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
  <soap address="http://localhost/Basys/Funcionalidad/Campeonato.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:CampeonatoSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

Web Service Equipo: Cabecera BPEL

```
<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/" >
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Equipo.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Equipo.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/Basys/Funcionalidad/Equipo.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:EquipoSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
  <soap address="http://localhost/Basys/Funcionalidad/Equipo.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:EquipoSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

Web Service Jugador: Cabecera BPEL

```
<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/" >
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Jugador.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Jugador.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/Basys/Funcionalidad/Jugador.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:JugadorSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
  <soap address="http://localhost/Basys/Funcionalidad/Jugador.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:JugadorSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

Web Service Partido: Cabecera BPEL

```
<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/" >
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Partido.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Partido.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/Basys/Funcionalidad/Partido.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:PartidoSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
  <soap address="http://localhost/Basys/Funcionalidad/Partido.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:PartidoSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

Web Service Periodo: Cabecera BPEL

```
<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/" >
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Periodo.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Periodo.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/Basys/Funcionalidad/Periodo.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:PeriodoSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
  <soap address="http://localhost/Basys/Funcionalidad/Periodo.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:PeriodoSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

Web Service Resumen: Cabecera BPEL

```
<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/" >
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Resumen.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Resumen.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/Basys/Funcionalidad/Resumen.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:ResumenSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
  <soap address="http://localhost/Basys/Funcionalidad/Resumen.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:ResumenSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

Web Service Estadística: Cabecera BPEL

```
<discovery xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/disco/" >
  <contractRef
ref="http://localhost/Basys/Funcionalidad/Estadistica.asmx?wsdl"
docRef="http://localhost/Basys/Funcionalidad/Estadistica.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
  <soap address="http://localhost/Basys/Funcionalidad/Estadistica.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:EstadisticaSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
  <soap address="http://localhost/Basys/Funcionalidad/Estadistica.asmx"
xmlns:q2="http://tempuri.org/" binding="q2:EstadisticaSoap12"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
</discovery>
```

4.1.3- Construcción del sistema

Basys es un sistema de información Web N capas, que ha sido desarrollado en su totalidad utilizando las herramientas que provee Microsoft Visual Studio 2005. Se ha utilizado el lenguaje Visual Basic .Net, tanto para el desarrollo de los XML Web Services, como del Sitio Web (Aplicación Servidor). Como motor de base de datos se utiliza SQL Server 2005.

Generación de la Base de Datos

En el SQL Server Management Studio se crea una nueva base de datos con el nombre "BASYS", en esta base de datos en blanco se va a subir las entidades, relaciones y demás objetos de la base de datos del sistema.

En la herramienta Case utilizada para la generación del Modelo Objeto – Relacional, se realiza una conexión directa con la base de datos a través de un origen de datos ODBC, y se genera la base de datos en SQL Server 2005, como se puede apreciar en la *Figura 4.16*

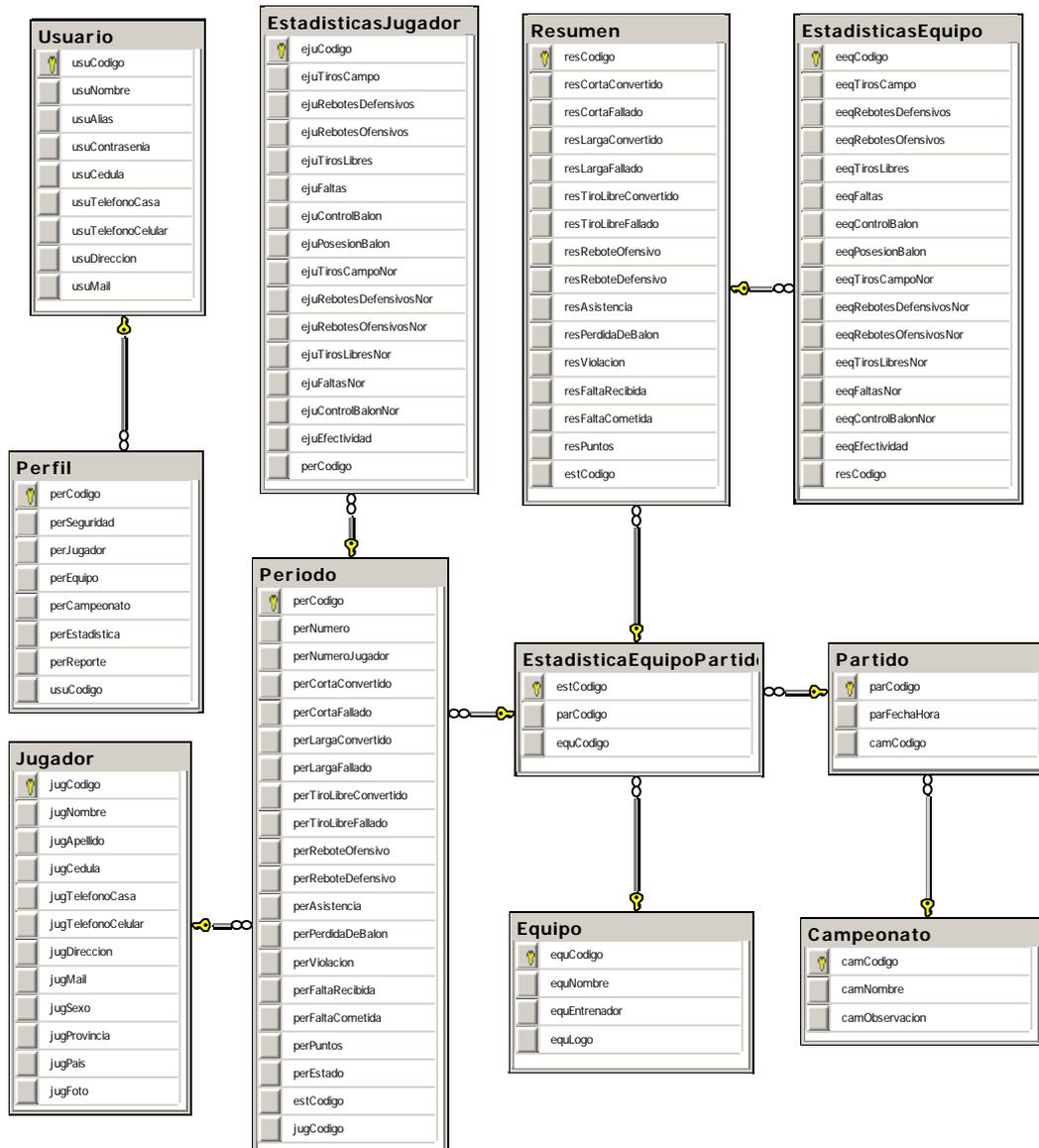


Figura 4.16: Modelo Objeto-Relacional en SQL Server

Diseño de la Interfaz de Usuario

En primer lugar se elaboran las artes del sistema, como imágenes, fondos, botones y logotipos en una herramienta de edición de imágenes, se debe tomar en cuenta que se han utilizado imágenes JPEG y GIF como formatos de archivos de imagen, ya que por su tamaño son convenientes cuando un cliente (Browser) realiza una petición de los mismos al Servidor Web (IIS), mejorando el desempeño final del sistema.

Posteriormente con las artes terminadas, se procede a acoplarlas a los controles que provee la herramienta Web Developer del Visual Studio 2005, generando la interfaz de usuario de todas las páginas Web (.ASPX) a las cuales los usuarios van a tener acceso.

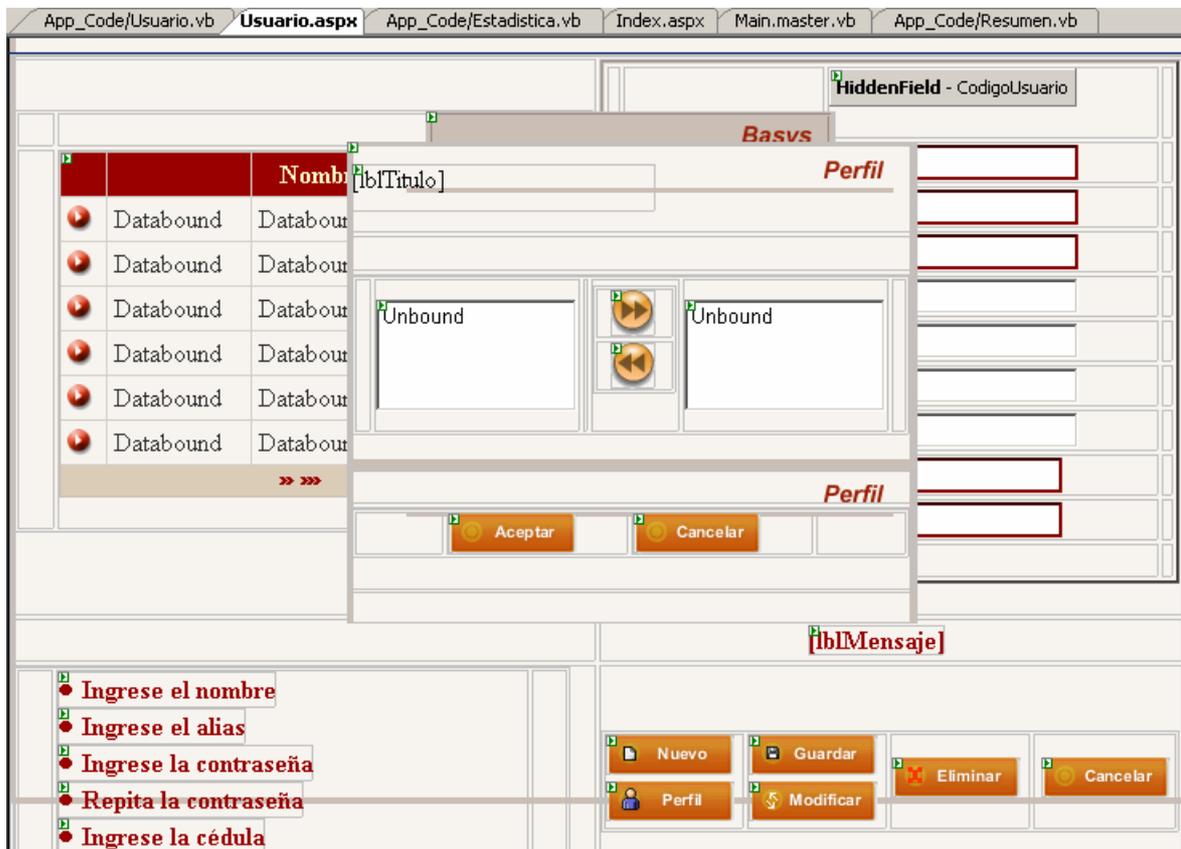


Figura 4.17: Pantalla de Usuarios en Vista Diseño

Estructuración del Sistema

Siguiendo la metodología MIDAS estudiada en el presente documento, se puede apreciar en la *Figura 4.18* que la solución del sistema Basys consta de dos proyectos:

- Funcionalidad: que aloja a los Servicios Web XML.
- Hipertexto: que contiene al sitio Web servidor.

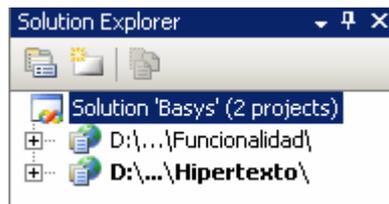


Figura 4.18: Proyectos de la solución Basys

Proyecto Funcionalidad

El proyecto Funcionalidad de la Solución Basys, contiene los Servicios Web, dentro de los cuales se define la lógica del negocio, es decir aquí se codifican todas las funciones que tiene el SIW.

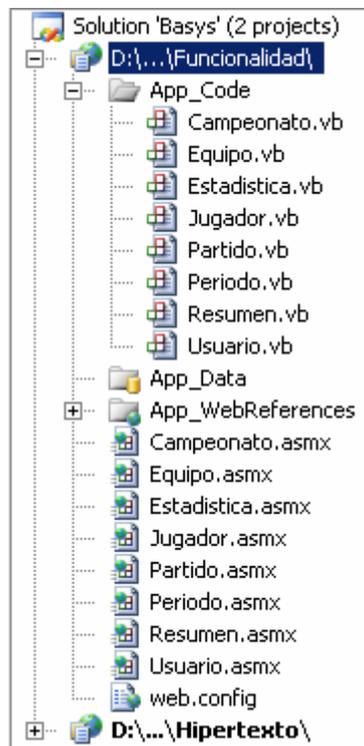


Figura 4.19: Proyecto Funcionalidad

La extensión de los archivos de servicios Web XML es .ASMX en los que se encuentra la definición del Servicio Web como tal, pero el código se introduce en archivos .VB.

El proyecto funcionalidad dispone de 8 servicios Web que contienen todas las funciones de Basys los mismos que están agrupados de forma lógica en los siguientes módulos:

- Módulo de seguridad: Usuario.aspx
- Módulo de campeonatos: Campeonato.aspx
- Módulo de equipos: Equipo.aspx
- Módulo de Jugadores: Jugador.aspx
- Módulo de Estadísticas: Estadística.aspx, Partido.aspx, Periodo.aspx y Resumen.aspx

Estructura del servicio Web

Como se puede apreciar a continuación, se muestra la definición del servicio Web (.ASMX).

```
<%@ WebService Language="vb" CodeBehind="~/App_Code/Usuario.vb"
Class="Usuario" %>
```

La definición del Servicio Web en el archivo .VB:

```
Imports System.Web
Imports System.Web.Services
Imports System.Web.Services.Protocols

<WebService(Namespace:="http://tempuri.org/")> _
<WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Public Class Usuario
    Inherits System.Web.Services.WebService
End Class
```

Luego se declaran constantes o variables de configuración de aplicación

```
Private DSN As String =
System.Configuration.ConfigurationManager.ConnectionStrings("BasysConnect
ionString").ToString
```

Posterior a esto se escribe el código de los servicios de usuario o Métodos Web (Funciones en Servicios Web) propiamente dichos, detallando toda la lógica de negocio, en donde se puede incluir llamadas a otros métodos del mismo Servicio Web o haciendo referencias a métodos ubicados en otros Servicios Web:

```

<WebMethod()> Public Function ConsultarUsuario(ByVal lngCodigo As Long)
As Data.DataSet
    Dim objcon As New System.Data.SqlClient.SqlConnection(DSN)
    Dim strsql As String
    strsql = "Select * from Usuario where usuCodigo=" & lngCodigo
    Dim objsda As New System.Data.SqlClient.SqlDataAdapter(strsql,
objcon)
    Dim objdas As New Data.DataSet()
    objsda.Fill(objdas, "Usuario")
    Return objdas
End Function

```

Proyecto Hipertexto

El proyecto Hipertexto de la Solución Basys, contiene el sitio Web, es decir las páginas HTML, ASPX, imágenes, hojas de estilo y demás archivos que van a ser solicitados por el cliente Web (browser). En este proyecto se define la interfaz de usuario, se hace referencias a los Servicios Web para realizar llamadas a sus funciones y se hacen las validaciones pertinentes a nivel de formulario.

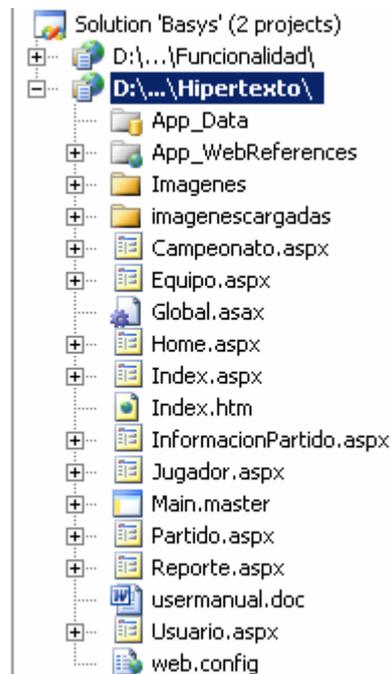


Figura 4.20: Proyecto Hipertexto

La extensión de los archivos que contienen las validaciones a nivel de formulario y la interfaz de usuario (código HTML y ASP) tienen una extensión

.ASPX, mientras que los archivos con todo el código y llamadas a servicios Web asociados a estos llevan una extensión .ASPX.VB y tienen el mismo nombre.

Referencias Web

Para acceder y utilizar los métodos Web codificados en los servicios Web y que se encuentran en el Proyecto Funcionalidad, se deben crear referencias Web a los mismos, colocándoles un nombre, en este caso las referencias llevan el prefijo "ws" para identificarlas, existe un número de referencias igual al número de servicios Web creados en Funcionalidad.

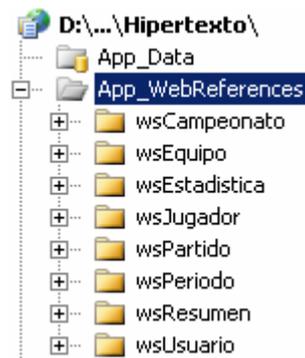


Figura 4.21: Referencias Web

Al crearse las referencias Web en el archivo de configuración del proyecto se crean vínculos a las mismas, para ubicarlas, este vínculo puede ser cualquier dirección de Internet en la cual se encuentren los servicios Web:

```
<appSettings>
  <add key="wsJugador.Jugador"
value="http://localhost/Basys/Funcionalidad/Jugador.aspx" />
  <add key="ReportServer" value="http://localhost/ReportServer" />
  <add key="ReportDirectory" value="/ReportesBasys/" />
  <add key="wsCampeonato.Campeonato"
value="http://localhost/Basys/Funcionalidad/Campeonato.aspx" />
  <add key="wsEquipo.Equipo"
value="http://localhost/Basys/Funcionalidad/Equipo.aspx" />
  <add key="wsPartido.Partido"
value="http://localhost/Basys/Funcionalidad/Partido.aspx" />
  <add key="wsPeriodo.Periodo"
value="http://localhost/Basys/Funcionalidad/Periodo.aspx" />
  <add key="wsResumen.Resumen"
value="http://localhost/Basys/Funcionalidad/Resumen.aspx" />
```

```

    <add key="wsUsuario.Usuario"
value="http://localhost/Basys/Funcionalidad/Usuario.aspx"/>
    <add key="wsEstadistica.Estadistica"
value="http://localhost/Basys/Funcionalidad/Estadistica.aspx"/>
</appSettings>

```

Luego se procede a codificar las validaciones y llamadas a los métodos Web. Las funciones del sitio Web están asociadas siempre a un objeto, el que puede ser un control de usuario como un botón, una vista Web o una lista desplegable.

```

Protected Sub GridUsuario_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles GridUsuario.SelectedIndexChanged
    paraEditar()
    Dim objCliente As New wsUsuario.Usuario
    Dim objds As New Data.DataSet
    Dim objfila As Data.DataRow
    objds =
objCliente.ConsultarUsuario(CType(Trim(GridUsuario.SelectedRow.Cells(2).Text),
Long))
    objfila = objds.Tables("Usuario").Rows(0)
    Me.CodigoUsuario.Value = objfila.Item("usuCodigo").ToString
    Me.txtNombre.Text = objfila.Item("usuNombre").ToString
    Me.txtAlias.Text = objfila.Item("usuAlias").ToString
    Me.txtContrasenia1.Text = objfila.Item("usuContrasenia").ToString
    Me.txtContrasenia2.Text = objfila.Item("usuContrasenia").ToString
    Me.txtCedula.Text = objfila.Item("usuCedula").ToString
    Me.txtTelefonoCasa.Text = objfila.Item("usuTelefonoCasa").ToString
    Me.txtTelefonoCelular.Text=objfila.Item("usuTelefonoCelular").ToString
    Me.txtDireccion.Text = objfila.Item("usuDireccion").ToString
    Me.txtMail.Text = objfila.Item("usuMail").ToString
    Me.btnEliminar.Enabled = True
    Me.btnModificar.Enabled = True
End Sub

```

4.1.4- Pruebas del sistema

La planificación y ejecución de las pruebas, representa una fase fundamental en el proceso de desarrollo de software, ya que son un elemento importante para garantizar la calidad del sistema, y mucho más si se trata de un SIW, ya que estos residen en una red, trabajan con sistemas operativos diferentes, navegadores Web distintos, protocolos de comunicación y plataformas de hardware que hacen mucho más trascendentes a las pruebas del SIW.

Desarrollo del protocolo de pruebas

Las pruebas a realizarse al SIW Basys contemplan las estrategias de Unidad, Integración, Aceptación y Prueba general del sistema en la cual se verificarán los siguientes parámetros:

- Recuperación
- Seguridad
- Resistencia(Stress)
- Rendimiento

Planificación de Prueba de Unidad

Tabla 4.7: Plan de pruebas de Unidad en Inicio de Sesión

Parámetro	Valores
Responsable	Andrea E. Puertas
Recursos	Hardware: Servidor Web y de Base de datos, PC cliente. Software: Sitio Web publicado, Web services publicados, código fuente del SIW.
Tiempo	3 días, 1 hora por día
Técnica de Prueba	Caja Blanca Caja Negra
Función del SIW	Validar el ingreso de un usuario al sistema
Alcance	<i>Caja Blanca:</i> Tablas comparativas explicando el comportamiento del SIW en procesos críticos como: Inicio de sesión y validaciones para el ingreso al sistema. <i>Caja Negra:</i> Identificar las clases de equivalencia

	para valorar el comportamiento del SIW en: Autenticación de usuarios, validaciones para el ingreso y mensajes del sistema.
Diseño de los casos de prueba	<i>Caja Blanca:</i> Identificación de lógica funcional y parámetros. <i>Caja Negra:</i> Valores extremos, fáciles e ilegales.

Tabla 4.8: Plan de pruebas de Unidad en actualización de puntos anotados en jugador y equipo

Parámetro	Valores
Responsable	Francisco J. Robayo
Recursos	Hardware: Servidor Web y de Base de datos, PC cliente. Software: Sitio Web publicado, Web services publicados, código fuente del SIW.
Tiempo	3 días, 1 hora por día
Técnica de Prueba	Caja Blanca Caja Negra
Función del SIW	Actualizar puntos anotados en jugador y equipo
Alcance	<i>Caja Blanca:</i> Tablas comparativas explicando el comportamiento del SIW en procesos críticos como: inicio de tarea, validaciones para la actualización, actualización de datos y finalización de tarea. <i>Caja Negra:</i> Identificar las clases de equivalencia para valorar el comportamiento del SIW en: actualización de información, validaciones para la actualización y mensajes del sistema.
Diseño de los casos de prueba	<i>Caja Blanca:</i> Identificación de lógica funcional y parámetros. <i>Caja Negra:</i> Valores extremos, fáciles e ilegales.

Planificación de Prueba de Integración

Tabla 4.9: Plan de pruebas de Integración

Parámetro	Valores
Responsable	Andrea E. Puertas, Francisco J. Robayo
Recursos	Hardware: Servidor Web y de Base de datos, PC cliente. Software: Sitio Web publicado, Web services publicados.
Tiempo	1 días, 3 horas.

Función del SIW	Todo el sistema
Alcance	Manejo completo del sistema por una persona con conocimientos básicos sobre manejo de páginas Web, previa capacitación.

Planificación de Prueba de Aceptación

Tabla 4.10: Plan de pruebas de Aceptación

Parámetro	Valores
Responsable	Andrea E. Puertas, Francisco J. Robayo, MS. Patricio Ponce, entrenador de la FEFDER
Recursos	Hardware: Servidor Web y de Base de datos, PC cliente. Software: Sitio Web publicado, Web services publicados.
Tiempo	1 día, 4 horas
Función del SIW	Todo el sistema
Alcance	Manejo completo del sistema por el usuario final, en este caso el MS. Patricio Ponce, entrenador de selecciones de básquetbol de la ESPE, previa capacitación.

Planificación de Pruebas Generales del sistema

Tabla 4.11: Plan de pruebas generales

Parámetro	Valores
Responsable	Andrea E. Puertas, Francisco J. Robayo
Recursos	Hardware: Servidor Web y de Base de datos, PC cliente. Software: Sitio Web publicado, Web services publicados.
Tiempo	1 día, 4 horas
Técnica de Prueba	Recuperación Seguridad Stress Rendimiento
Función del SIW	Todo el sistema
Alcance	Manejo completo del sistema por parte de los desarrolladores, sometiendo al sistema a condiciones extremas.

Ejecución del protocolo de pruebas

Pruebas de Unidad

Las pruebas de unidad se realizaron a todo el sistema para detectar y corregir errores, a continuación se muestran documentadas dos funciones específicas probadas con esta estrategia.

Prueba de Unidad en Inicio de Sesión

Las siguientes tablas muestran el proceso de la prueba de unidad en la función Validar el ingreso de un usuario al sistema.

Tabla 4.12: Prueba de unidad en Inicio de Sesión (Caja Blanca)

Proceso	“Validar el ingreso de un usuario al sistema”
Requerimiento	Ingreso de un usuario al sistema. Si se cumple con las condiciones, se ingresará al sistema correctamente.
Definición de datos	Alias de usuario y contraseña. Los dos argumentos pueden tomar cualquier valor y pueden tener cualquier extensión.
Descripción del proceso	El proceso para permitir el ingreso del usuario al sistema es recuperar de la base de datos la información respectiva (alias y contraseña) y compararlos con los datos que se ingresan en el formulario Web de Login. Si se verifica que los datos ingresados son iguales a los datos recuperados, se permite el inicio de sesión e ingreso del usuario al sistema.
Comportamiento	La tabla 4.13 describe el comportamiento del sistema de acuerdo a los datos ingresados.

Tabla 4.13: Prueba de unidad, validación de datos (Caja Blanca)

Parámetro 1	Ingresó parámetro válido	Parámetro 2	Ingresó parámetro válido	Error	Ingreso al sistema
Alias de usuario	Si	Contraseña	Si	No	Si
Alias de usuario	Si	Contraseña	No	Si	No
Alias de usuario	No	Contraseña	Si	Si	No
Alias de usuario	No	Contraseña	No	Si	No

Para realizar la prueba de Caja Negra se toman en cuenta los casos de prueba.

Tabla 4.14: Prueba de unidad en Inicio de Sesión (Caja Negra)

Proceso	“Validar el ingreso de un usuario al sistema”
Requerimiento	Ingreso de un usuario al sistema. Si se cumple con las condiciones, se ingresará al sistema correctamente.
Definición de datos	Alias de usuario y contraseña. Los dos argumentos pueden tomar cualquier valor y pueden tener cualquier extensión.
Definición de Casos de Prueba	Valores Fáciles: El alias y la contraseña existen. Valores Extremos: El alias y la contraseña no existen
Descripción del proceso	El proceso para permitir el ingreso del usuario al sistema es recuperar de la base de datos la información respectiva (alias y contraseña) y compararlos con los datos que se ingresan en el formulario Web de Login. Si se verifica que los datos ingresados son iguales a los datos recuperados, se permite el inicio de sesión e ingreso del usuario al sistema.
Comportamiento	La tabla 4.15 describe el comportamiento del formulario Web de acuerdo a los datos ingresados.

Tabla 4.15: Prueba de unidad, validación de datos (Caja Negra)

Ingresó Alias de usuario	Ingresó Contraseña	Alias Válido	Contraseña Válida	Error	Ingreso al sistema
Si	Si	Si	Si	No	Si
Si	Si	Si	No	Si	No
Si	Si	No	Si	Si	No
Si	Si	No	No	Si	No
Si	No	Si	Si	Si	No
Si	No	Si	No	Si	No
Si	No	No	Si	Si	No
Si	No	No	No	Si	No
No	Si	Si	Si	Si	No
No	Si	Si	No	Si	No
No	Si	No	Si	Si	No
No	Si	No	No	Si	No

No	No	Si	Si	Si	No
No	No	Si	No	Si	No
No	No	No	Si	Si	No
No	No	No	No	Si	No

Conclusión: Al observar las tablas precedentes se puede concluir que el comportamiento de la función es el adecuado para el sistema. Es decir solo permite el ingreso del usuario al sistema cuando los dos argumentos (alias de usuario y contraseña) son válidos y concuerdan con la información grabada en la base de datos.

Prueba de Unidad en Actualización de puntos anotados en jugador y equipo

Las siguientes tablas muestran el proceso de la prueba de unidad en la función Actualizar puntos anotados en jugador y equipo.

Tabla 4.16: Prueba de unidad en Actualización de puntos anotados en jugador y equipo (Caja Blanca)

Proceso	“Actualizar puntos anotados en jugador y equipo”
Requerimiento	Actualización de puntos anotados en jugador y equipo. Si se cumplen las condiciones, se actualizarán los datos correctamente.
Definición de datos	Nuevo valor de puntos anotados (Tipo integer) Código de estadísticas de equipo (Tipo long) Código del jugador(Tipo long)
Descripción del proceso	El proceso para permitir la actualización de puntos anotados en jugador y equipo es enviar a base de datos la información respectiva (valor de puntos, código de estadísticas de equipo, código del jugador), actualizar dichos datos primero en información de equipo y luego en información de jugador.
Comportamiento	La tabla 4.17 describe el comportamiento del sistema de acuerdo a los datos ingresados.

Tabla 4.17: Prueba de unidad, detalle del proceso (Caja Blanca)

Parám 1	Ingresó parám. válido	Parám 2	Ingresó parám. válido	Parám 3	Ingresó parám. válido	Error	Actualizó datos
Puntos	Si	Estadísticas Equipo	Si	Código Jugador	Si	No	Si
Puntos	Si	Estadísticas Equipo	Si	Código Jugador	No	Si	No
Puntos	Si	Estadísticas Equipo	No	Código Jugador	Si	Si	No
Puntos	Si	Estadísticas Equipo	No	Código Jugador	No	Si	No
Puntos	No	Estadísticas Equipo	Si	Código Jugador	Si	Si	No
Puntos	No	Estadísticas Equipo	Si	Código Jugador	No	Si	No
Puntos	No	Estadísticas Equipo	No	Código Jugador	Si	Si	No
Puntos	No	Estadísticas Equipo	No	Código Jugador	No	Si	No

Para realizar la prueba de Caja Negra se toman en cuenta los casos de prueba.

Tabla 4.18: Actualización de puntos anotados en jugador y equipo (Caja Negra)

Proceso	“Validar el ingreso de un usuario al sistema”
Requerimiento	Actualización de puntos anotados en jugador y equipo. Si se cumplen las condiciones, se actualizarán los datos correctamente.
Definición de datos	Nuevo valor de puntos anotados (Tipo integer) Código de estadísticas de equipo (Tipo long) Código del jugador(Tipo long)
Definición de Casos de Prueba	Valores Fáciles: Los valores consignados existen y son válidos. Valores Extremos: Los valores consignados no existen.
Descripción del proceso	El proceso para permitir la actualización de puntos anotados en jugador y equipo es enviar a base de datos la información respectiva (valor de puntos, código de estadísticas de equipo, código del jugador), actualizar dichos datos primero en información de equipo y luego en información de jugador.

Comportamiento	La tabla 4.19 describe el comportamiento del formulario Web de acuerdo a los datos ingresados.
-----------------------	--

Tabla 4.19: Prueba de unidad, detalle del proceso (Caja Negra)

Ingresó Puntos Válidos	Ingresó Estadísticas Equipo Válido	Código Jugador Válido	Ingreso manual de datos	Error	Ingreso al sistema
Si	Si	Si	No	No	Si
Si	Si	No	No	Si	No
Si	No	Si	No	Si	No
Si	No	No	No	Si	No
No	Si	Si	No	Si	No
No	Si	No	No	Si	No
No	No	Si	No	Si	No
No	No	No	No	Si	No

Conclusión: Se puede verificar que los controles implementados en el sistema para la actualización de la información del partido tanto de jugadores como de equipos, hacen que se minimicen los errores por parte del usuario del sistema consiguiendo datos correctos y completos que se consignan a la BD.

Pruebas de Integración

Las pruebas de integración se realizaron a todo el sistema, una vez concluida la construcción del mismo en su totalidad, se incluyó a personas con conocimientos básicos de manejo de páginas Web e Internet, que desconocían el objetivo del sistema, es decir que se intentó únicamente validar y verificar el funcionamiento en conjunto del sistema y buscar opiniones fuera de clientes y desarrolladores.

Conclusión: Se puede verificar que el sistema posee una interfaz amigable al usuario, lo que verifica la usabilidad del mismo, se determina que los módulos de seguridad, jugador, equipo, campeonato y estadísticas cumplen su objetivo. Se

comprueba que el sistema tiene buen rendimiento y que los reportes consiguen extraer la información requerida por el cliente y que presentan funcionalidades extras.

Pruebas de Aceptación

Para la realización de las pruebas de integración, se contó con la presencia del MS. Patricio Ponce, entrenador de las selecciones de baloncesto de la ESPE y profesor de la FEFDER, como cliente y usuario final del sistema.

Se valida que el sistema cumpla con las funciones descritas en la especificación de requerimientos de software, es decir se revisa la funcionalidad del sistema.

Tabla 4.20: Formulario de revisión de funcionalidad

Formulario de revisión de funcionalidad				
Analista: Desarrolladores y personal de la ESPE				
Revisión de Estándares				
Actividad	Si	No	No Aplica	Observaciones
¿El formulario realiza la función que se requiere?	X			
¿Los datos del formulario Web cambian de acuerdo a las acciones realizadas?	X			
¿Es rápido y fácil el manejo de los formularios Web?	X			La página se refresca en cada acción
Cuando se cambia el valor de un campo de entrada, ¿se modifican los cambios de despliegue?	X			
Los campos que hacen referencia a datos de otra tablas ¿Tienen cada uno sus listas de valores?	X			
¿Las listas de valores son lentas para recuperar la información?		X		
¿El tiempo de respuesta es el adecuado?	X			
¿El orden de navegación de los			X	

campos es el correcto?				
Cuándo se realiza una acción equivocada, ¿existen mensajes de alerta?	X			
Si el reporte requiere de mucho tiempo, ¿se le alerta al usuario sobre el retardo de la respuesta?	X			
Si un campo es destinado a ingresar un determinado tipo de variable, ¿se valida su ingreso?	X			De acuerdo al tipo de dato

Conclusión: El SIW Basys cumple y sobrepasa las expectativas del usuario final, ya que se han implementado funcionalidades adicionales a las pedidas por el cliente para mejorar el desempeño del sistema como tal.

Pruebas Generales del Sistema

Las pruebas generales determinan la efectividad del sistema, ya que toman en cuenta los aspectos de recuperación, seguridad, resistencia (stress) y rendimiento, verificando tanto la funcionalidad como la existencia de errores que se pueden haber pasado por alto el momento del desarrollo o que no pueden estar relacionados con la programación sino más bien con el entorno del sistema.

Prueba de Recuperación

En la prueba de recuperación se verifica la capacidad que tiene el SIW para su recuperación ante su uso excesivo, o si es tolerante a los fallos que pueda producir el sistema operativo.

Para la evaluar la recuperación del SIW, es importante aclarar que por la naturaleza del mismo, esta recuperación tiene que ser manual.

Tabla 4.21: Prueba de recuperación

Prueba	Validación	Observaciones
Se reinicia o se apaga el servidor mientras se está usando Basys	Los datos en el sistema permanecen incorruptibles	Será necesario reiniciar el sistema en el equipo cliente para volver a utilizarlo

Prueba de Seguridad

La prueba de seguridad del SIW tiene por objetivo verificar la fiabilidad de ingreso al sistema, niveles de acceso (perfiles) y seguridad de la información.

Tabla 4.22: Prueba de seguridad

Prueba	Validación	Observaciones
Ingreso al sistema	Verificada en pruebas de Unidad	Ver tablas 4.11 y 4.12
Niveles de acceso	Los usuarios tienen perfiles de acceso por módulo. Solo el administrador del sistema posee permisos de acceso al módulo de seguridad.	
Seguridad de información	Se tiene el respaldo que la información se encuentra almacenada en un servidor de base de datos.	Administrada por el DBMS SQL Server 2005

Prueba de Resistencia (stress) y Rendimiento

La prueba de stress del SIW tiene por objetivo verificar la potencialidad del sistema, cuando está sujeto a condiciones extremas, para verificar cuanto tiempo duraría antes de colapsar en el caso que se diere.

Tabla 4.23: Prueba de stress

Prueba	Validación	Observaciones
Se ejecutan en el servidor 10 programas en primer plano (incluido Basys) y 23 en segundo plano (Incluido el servidor de Base de Datos)	El SIW disminuye su rendimiento y sus tiempos de respuesta crecen	Evitar forzar al servidor

La prueba de rendimiento consiste en verificar los recursos del sistema operativo que consume el SIW.

Con los administradores de procesos del sistema operativo se puede verificar que entre el servidor de base de datos (SQL Server) y el Servidor Web (IIS), consumen el 35% de los recursos de un servidor con las características mínimas requeridas por Basys.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1- Conclusiones

- La metodología de desarrollo de Sistemas de Información Web que propone MIDAS permite el diseño y construcción de Aplicaciones Web sólidas, que cumplen con las características de calidad, seguridad, funcionalidad y usabilidad que necesitan los usuarios de la Web.
- La metodología MIDAS aprovecha en gran medida la robusta estructura de la Arquitectura Dirigida por Modelos MDA y los principios fundamentales de metodologías ágiles como *eXtreme Programming XP*, enfocando el proceso de desarrollo a los Sistemas de Información Web que poseen características que difieren mucho de los sistemas tradicionales.
- El desarrollo de los CIM, PIM y PSM, que utilizan el lenguaje UML como base, permiten modelar el contenido, la lógica del negocio, y el hipertexto del sistema de información Web; para tener una idea clara y precisa al momento de implementarlo, garantizando su óptimo desarrollo.
- Uno de los modelos que aportan de manera significativa tanto a los desarrolladores como a los usuarios finales, es el Modelo de Navegación Extendido, puesto que permite visualizar de manera global

y sencilla el funcionamiento y las diferentes opciones que brinda el sistema.

- El SIW Basys desarrollado siguiendo MIDAS, provee herramientas de gestión de jugadores, equipos y campeonatos de básquetbol así como también tiene la capacidad de obtener estadísticas de los partidos, entregando así información de suma importancia a los expertos del baloncesto, para que sea usada en beneficio del desarrollo del mismo.
- Para asegurar el óptimo funcionamiento y garantizar la calidad del sistema Basys, se han realizado pruebas de rendimiento, resistencia, interfaz y funcionalidad.
- Se han cubierto las expectativas de la Facultad de Educación Física, Deportes y Recreación, y se ha entregado una herramienta que permitirá mejorar el análisis del rendimiento de los jugadores de baloncesto de la ESPE.

5.2- Recomendaciones

- Es recomendable definir estándares para los nombres de variables, funciones, controles y objetos antes de realizar la construcción de un sistema, puesto que esto ayudará al desarrollador a estructurar un código mucho más limpio que al momento de realizar pruebas y depuraciones facilite su trabajo.
- En el desarrollo de las materias de la rama de Ingeniería de Software que se imparten en la ESPE se debe poner énfasis en el estudio de la Arquitectura Dirigida por Modelos MDA, ya que aporta valiosos conceptos al desarrollo de software, brindando al estudiante o profesional que profundiza su análisis una visión clara de los requerimientos del mundo actual.
- Es importante involucrar al usuario final del sistema en todas las etapas de desarrollo de software, ya que solo así se conseguirá un cliente satisfecho, un producto que cumpla sus requerimientos y cubra sus expectativas y necesidades.
- La Facultad de Sistemas como parte del pénsum obligatorio de estudios debería incluir una materia que se dedique únicamente al estudio de la Ingeniería Web, metodologías y herramientas de desarrollo, ya que no se le dedica el tiempo suficiente a este tipo específico de sistemas y en

la actualidad ocupan una gran parte del mercado de software a nivel mundial.

- Promover como tema de tesis, el desarrollo de una herramienta CASE basada en la metodología MIDAS, que facilite y mejore el modelamiento de Sistemas de Información Web, ya que en la actualidad se deben acoplar los modelos que proponen las metodologías orientadas al desarrollo Web, a las herramientas existentes.

BIBLIOGRAFÍA

Referencias Bibliográficas

- PRESSMAN, Roger S., Ingeniería del Software. Un enfoque práctico, Editorial McGraw-Hill. España 2002. 5ta. Edición.
- PLATT, David S., Introducing Microsoft .NET, Microsoft Press. Redmond, Washington 2002. 2da. Edición.
- RICHTER, Jeffrey; BALENA, Francisco. Applied Microsoft .Net framework Programming in Microsoft Visual Basic .Net, Microsoft Press 2003.
- CHRISTENSEN, Erik; CURBERA, Francisco; MEREDITH, Greg; WEERAWARANA, Sanjiva. Lenguaje de descripción de servicios Web (WSDL) 1.0, Ariba 2000, International Business Machines Corporation, Microsoft Corporation
- GOLDBARG, Charles F.; PRESCOD Paul. The XML handbook, Editorial Prentice Hall PTR. 2000.
- EISENBERG, A.; MELTON, J.; KULKARNI, K.; MICHELS J. SQL: 2003 has been published, ACM SIGMOD Record Volume 33, March 2004.
- FREEDMAN, ALAN, Diccionario de Computación Bilingüe, Editorial McGraw-Hill. Colombia 1996, 7ma. Edición.

Referencias Electrónicas

- MARCOS, Esperanza; CÁCERES, Paloma. 2002, MIDAS: A methodology for Web information systems development, Rey Juan Carlos University, Kybele Research Group
Disponibile en: <http://kybele.escet.urjc.es/>
- MARCOS, Esperanza; CÁCERES, Paloma. Hacia un proceso metodológico dirigido por modelos para el desarrollo ágil de sistemas de información Web, Grupo de Investigación KYBELE, Universidad Rey Juan Carlos. Madrid España
Disponibile en: <http://kybele.escet.urjc.es/>
- MARCOS, Esperanza; CÁCERES, Paloma. Del Modelo de Casos de Uso al Modelo de Navegación, Grupo de Investigación KYBELE, Universidad Rey Juan Carlos. Madrid España
Disponibile en:
<http://www.spc.org.pe/ideas2004/skin.php?idioma=es&source=html/133/index.html>
- MARCOS, Esperanza; CÁCERES, Paloma. Modelado Navegacional desde una Perspectiva Orientada a Servicios de Usuario, Grupo de Investigación KYBELE, Universidad Rey Juan Carlos. Madrid España
Disponibile en: <http://kybele.escet.urjc.es/>

- SIEGEL, Jon; The OMG Staff Strategy Group. Developing in OMG's Model-Driven Architecture, Object Management Group White Paper. November, 2001.

Disponible en: <http://www.omg.org/mda/>

- World Wide Web Consortium

Disponible en: www.w3c.org

- Proyecto HELDU, 2005. OMG Y MDA, 11 Julio de 2005.

Disponible en: <http://www.esi.es/en/Projects/Heldu/helduOMG&MDA.html>

- Servicios Web XML

Disponible en:

www.w3c.es/Eventos/ServiciosWeb/presentaciones/Presentacion_JSWeb_Marcos_Lopez.pdf

- Desarrollo Orientado a Objetos con UML

Disponible en: <http://www.clikear.com/manuales/uml/introduccion.asp>

- The Microsoft .NET Framework Community.

Disponible en: <http://www.gotdotnet.com/>

- Federación Internacional de Básquetbol Asociado

Referencias a:

- Bradshaw, D. (1984) Motivation Through Basketball Statistics. Coaching Review

- Swalgin, K. (1994) The Basketball Evaluation System: A Scientific Approach To Player Evaluation. In: J. Krause (Ed.) Coaching Basketball. National Association Of Basketball Coaches. Masters Press, New York.
- Brown, D. (1991) Scouting And Statistics. Dale Brown Enterprises, New York.
- Heeren, D. (1994) The Basketball Abstract. Masters Press, Indianapolis.
- Sampaio, A. (1998). Los indicadores estadísticos más determinantes en el resultado final en los partidos de básquetbol.

Disponibles en: www.fiba.com y www.fibaamericas.com

GLOSARIO DE TÉRMINOS

- **.Net:** New Enhanced Technology, Nueva tecnología mejorada para el desarrollo de sistemas.
- **.NET Framework:** marco de trabajo de .NET.
- **ASP:** Active Server Pages, Páginas activas de servidor.
- **ASPX:** Active Server Pages extended, Páginas activas de servidor extendidas. Extensión para las nuevas páginas ASP.
- **ASMX:** Extensión para los archivos de Servicios Web XML.
- **BASYS:** Basketball System, Sistema de Básquetbol desarrollado para la Facultad de Educación Física, Deportes y recreación de la ESPE.
- **BPEL4WS:** Business Process Execution Language for Web Services. Lenguaje de ejecución de procesos de negocio para Web Services.
- **CLR:** Common Language Runtime, Motor de ejecución de las aplicaciones .Net.
- **CIM:** Modelos Independientes de Computación.
- **CS:** Servicios de uso compuestos.
- **CUS:** Servicios de usuario conceptuales.
- **DBMS:** Data Base Management System, Sistema de administración de Base de Datos.
- **ERS:** Especificación de Requerimientos del Sistema.
- **FBS:** Servicios de uso básico funcional.
- **FCL:** Framework Class Library.
- **FEFDER:** Facultad de Educación Física, Deportes y Recreación de la ESPE.

- **FIBA:** Federación Internacional de Básquetbol.
- **FS:** Fragmentos funcionales.
- **GUI:** Interfaz Grafica de usuario.
- **HTML:** Hypertext Markup Language, Lenguaje de marcado de hipertexto.
- **HTTP:** Hypertext Transfer Protocol, Protocolo de transferencia de hipertextos.
- **IEEE:** Institute of Electrical & Electronics Engineers. Instituto de Ingenieros Eléctricos y Electrónicos.
- **IIS:** Internet Information Server, Servidor Web para plataformas Windows.
- **JPEG:** Join Photographic Expert Group. Grupo unido de expertos en fotografía. (Extensión de archivos gráficos).
- **MDA:** Model Driven Architecture, Arquitectura dirigida por modelos.
- **NBA:** National Basketball Association, Liga Profesional de Baloncesto de Norte América.
- **OMG:** Object Management Group.
- **PIM:** Modelos Independientes de la Plataforma.
- **PSM:** Modelos Específicos de la Plataforma.
- **RAD:** Rapid Application Development, Desarrollo rápido de aplicaciones.
- **SBS:** Servicios de uso básico estructural.
- **SIW:** Sistema de Información Web.
- **SOAP:** Simple Object Access Protocol, protocolo de comunicaciones para los Servicios Web XML.
- **SQL:** Structured Query Language, Lenguaje de consulta estructurado.
- **SS:** Fragmentos estructurales.

- **Tendex:** coeficiente de evaluación del rendimiento (efectividad) de los jugadores.
- **UML:** Unified Modeling Language, Lenguaje de modelado unificado.
- **VB:** Visual Basic.
- **W3C:** World Wide Web Consortium.
- **WSDL:** Web Services Description Language, Lenguaje de descripción de servicios Web
- **X-link:** Lenguaje de Enlaces (vínculos) XML.
- **XML:** Extended Markup Language, Lenguaje de marcado extendido.
- **XP:** eXtreme Programming, Programación extrema.

ANEXOS

ANEXO A
MANUAL DE INSTALACIÓN

INDICE

MANUAL DE INSTALACIÓN	300
INTRODUCCIÓN	300
REQUERIMIENTOS MÍNIMOS DE INSTALACIÓN	300
<i>Requerimiento de Plataforma</i>	300
<i>Requerimientos de Bases de datos</i>	301
<i>Requerimientos de Infraestructura</i>	301
INSTALADOR DE BASYS V1.0	302
<i>Prerrequisitos para la instalación</i>	302
Levantamiento del IIS	302
.Net Framework 3.0	306
Windows Installer 3.1	308
<i>Instalación de MS SQL Server 2005</i>	310
<i>Configuración de la Base de Datos Basys</i>	321
Subir el backup de la base de datos	321
<i>Reportes de Basys</i>	326
<i>Basys Funcionalidad</i>	328
Personalización del archivo de configuración	331
<i>Basys Hipertexto</i>	332

ANEXO A

MANUAL DE INSTALACIÓN

INTRODUCCIÓN

El presente manual hace referencia a la instalación del Sitio Web, Servicios Web y Base de datos del sistema para la administración de estadísticas de los partidos de básquetbol que la Facultad de Educación Física, Deportes y Recreación de la ESPE desee obtener.

Este manual está orientado ha Administradores del Sitio Web BASYS, ó para personas con alto conocimiento de informática. Es necesario para el administrador tener conocimientos de Administración de Bases de Datos.

REQUERIMIENTOS MÍNIMOS DE INSTALACIÓN

Los requerimientos mínimos son importantes ya que de ellos depende el buen funcionamiento de la aplicación. Para lo cual aquí se hará énfasis en la plataforma, conexión a la base de datos y otros tópicos significativos. Es necesario tener en cuenta que deben ser cumplidos a cabalidad todos los pasos para garantizar un óptimo desempeño del sistema.

Requerimiento de Plataforma

BASYS es un aplicativo creado en base a la plataforma .Net de Microsoft, por lo que se considera necesario un servidor que corra la siguiente plataforma:

- Windows Vista, 2000 SP²³ 4, 2003 SP 1, Windows XP SP 3.
- Servidor Web Internet Information Server versión 5.0 o superior

Cabe señalar que las políticas de seguridad con que se manejen este tipo aplicaciones, dependerán del administrador de aplicaciones local.

²³ SP: Service Pack, Paquete de Servicio (Conjunto de actualizaciones provistas por Microsoft).

Requerimientos de Bases de datos

BASYS trabaja con la base de Datos SQL Server 2005 Express Edition, por lo que se considera necesario tener instalado como mínimo:

- SQL Server Management Studio (como administrador del motor de Base de Datos).
- Reporting Services (para la generación y administración de reportes del sistema).

Cabe señalar que las políticas de seguridad y contingencia con que se manejen este tipo de DBMS²⁴ dependerán del administrador de aplicaciones local.

Requerimientos de Infraestructura

Con respecto a la infraestructura y con referencia al Hardware es necesario cumplir con los siguientes requerimientos mínimos:

Tabla A.1: Requerimientos mínimos de Hardware

Dispositivo	Requerimiento mínimo
Procesador	Pentium IV de 1.4 GHz
Espacio en Disco	30 GB
Memoria	256 MB
Tarjeta de Red	10/100 mbps
CD ROM	32x o superior
Externos	Monitor, mouse, teclado

²⁴ DBMS: Data Base Management System (Sistemas manejadores de base de datos)

INSTALADOR DE BASYS V1.0

Para realizar la instalación del sistema Basys v1.0 se dispone de una página Web (*Figura A.1*) que contiene el manual y los instaladores necesarios para llevar a cabo la implantación del mismo.



Figura A.1: Menú de Instalación

Prerrequisitos para la instalación

Levantamiento del IIS

Puesto que Basys es una aplicación Web es necesario instalarla en un Servidor Web que corra Windows, por lo tanto se debe levantar IIS²⁵. Para lo cual se deben realizar los siguientes pasos:

1. Ingresar al *Panel de Control* como se muestra en la *Figura A.2* a continuación:

²⁵ IIS: Internet Information Server (Servidor Web de Microsoft)



Figura A.2: Abrir Panel de control

2. Seleccionar el acceso directo *Agregar o quitar programas*:



3. En la ventana que se despliega en el lado derecho se encuentran las diferentes acciones que se puede realizar, seleccionar la opción *Agregar o quitar componentes de Windows*:



4. Seguidamente se muestra una ventana que permite seleccionar los componentes de Windows que se desean instalar. En este caso puntual se debe seleccionar el check box *Servicios de Internet Information Server (IIS)*. Luego presionar el botón *Detalles* para seleccionar todas las opciones que se presentan, finalmente se da clic en el botón *Siguiente* o *Next*.

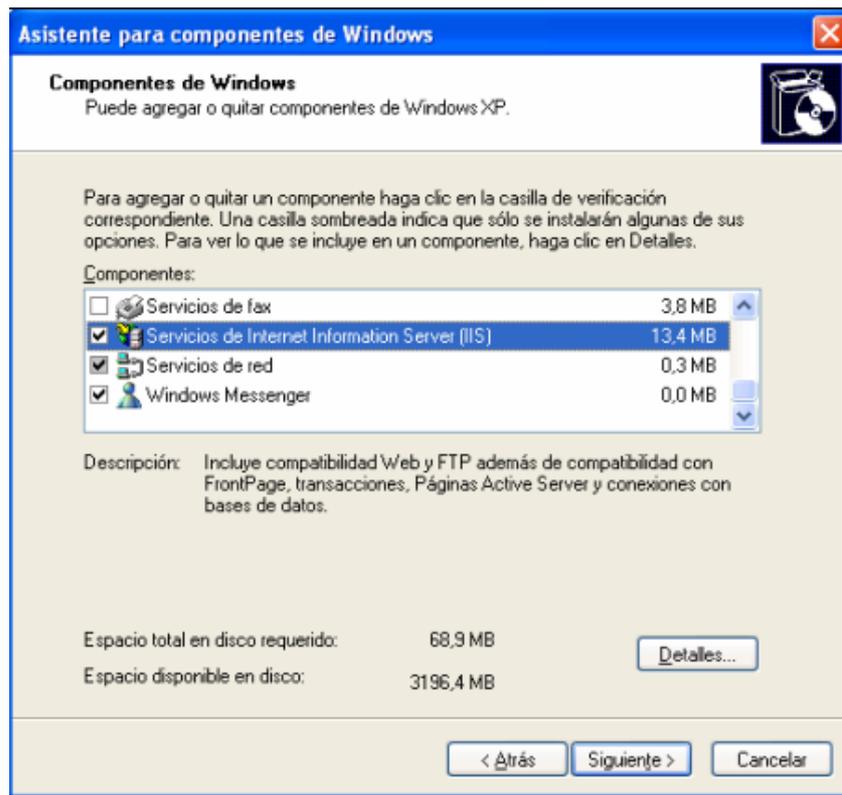


Figura A.3: Seleccionar el componente IIS

5. El sistema solicita el CD del sistema operativo del ordenador en el que se está trabajando, en este caso es Windows XP Professional Service Pack 2. Presionar el botón *Aceptar* o *Accept*.

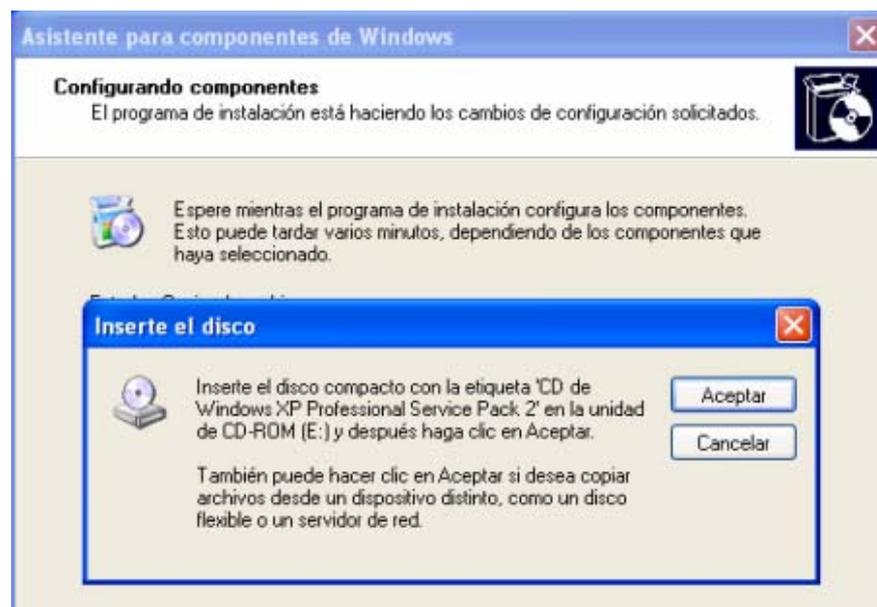


Figura A.4: Ingresar el CD del sistema operativo actual

6. El instalador empieza por copiar los archivos que necesita para realizar la instalación y luego instala los Servicios de Internet Information Server

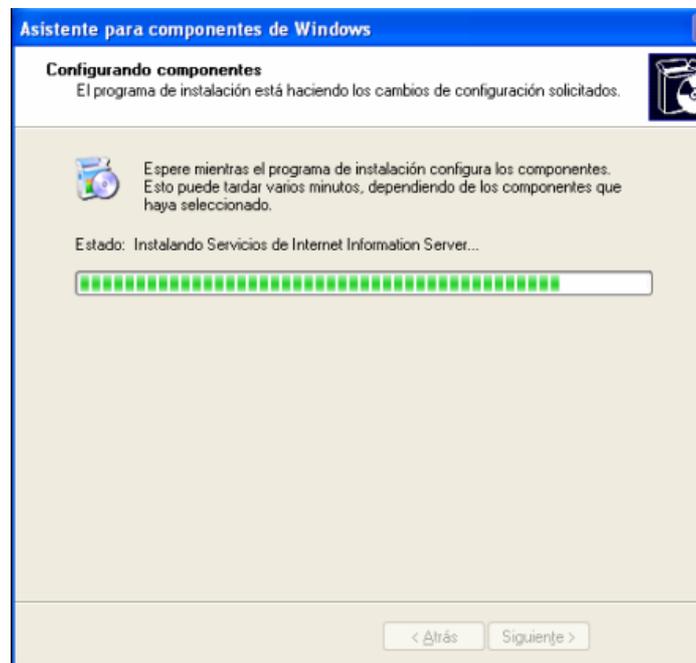


Figura A.5: Instalación del IIS

7. Finalización de la instalación del componente IIS de Windows. Presionar el botón *Finalizar* o *Finish*.



Figura A.6: Finalización de la instalación

.Net Framework 3.0

El .Net Framework es un marco de trabajo necesario para la ejecución del sistema Basys. Para instalarlo se debe acceder a la segunda opción del menú de instalación y seguir los siguientes pasos:

1. Bienvenida del instalador del .NET Framework 2.0. Seleccionar el botón *Siguiente* o *Next*.

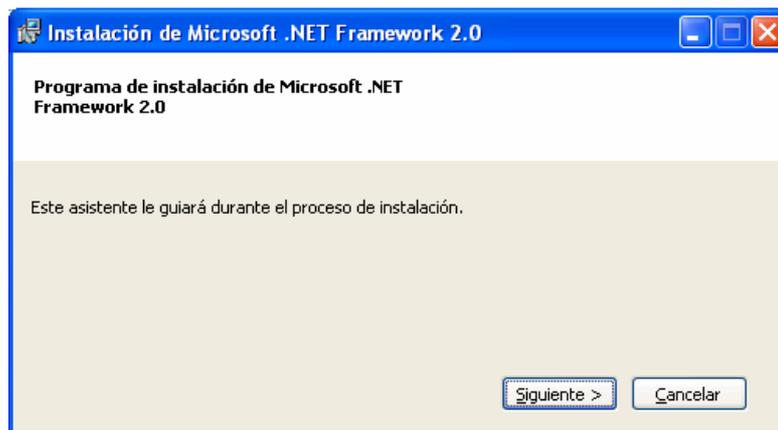


Figura A.7: Bienvenida del instalador

2. Seguidamente se presenta el contrato de licencia del software, en el que debemos seleccionar el check box *Aceptando los términos del contrato*, y presionar el botón *Instalar* o *Install*

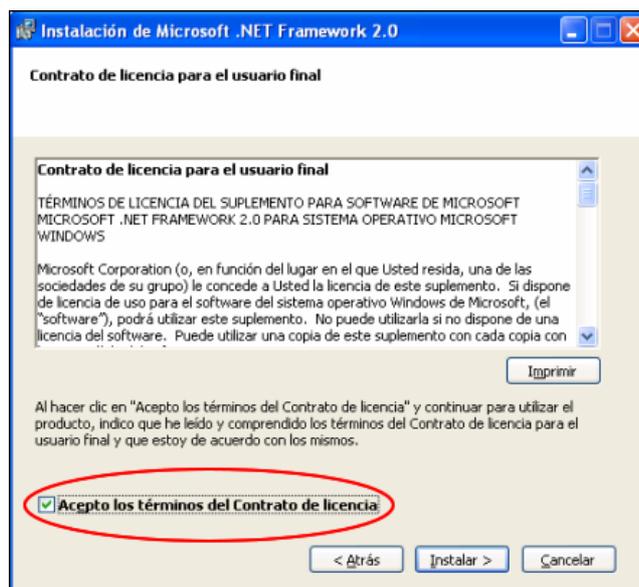


Figura A.8: Aceptación de los términos del contrato

3. A continuación se muestra el progreso de la instalación:

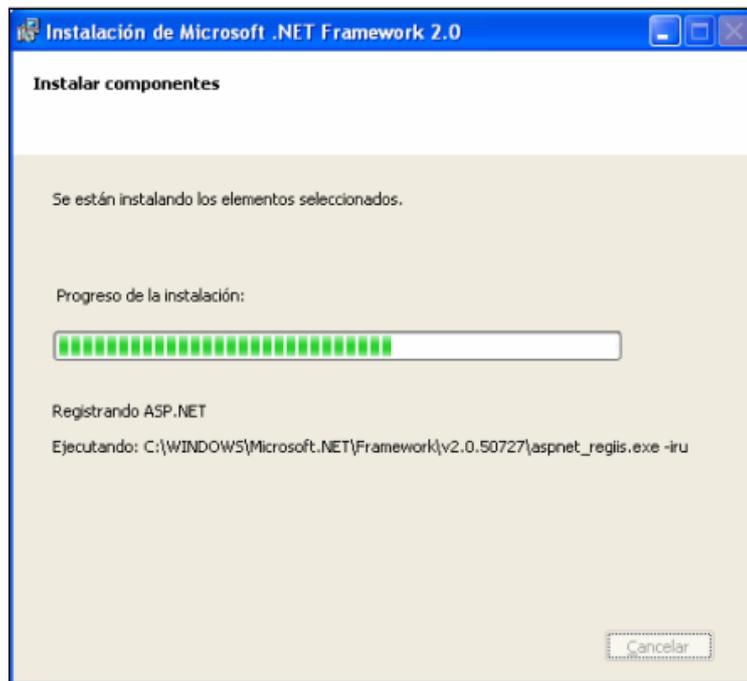


Figura A.9: Instalación del .NET Framework 2.0

4. Instalación del .Net Framework 2.0 finalizada correctamente.

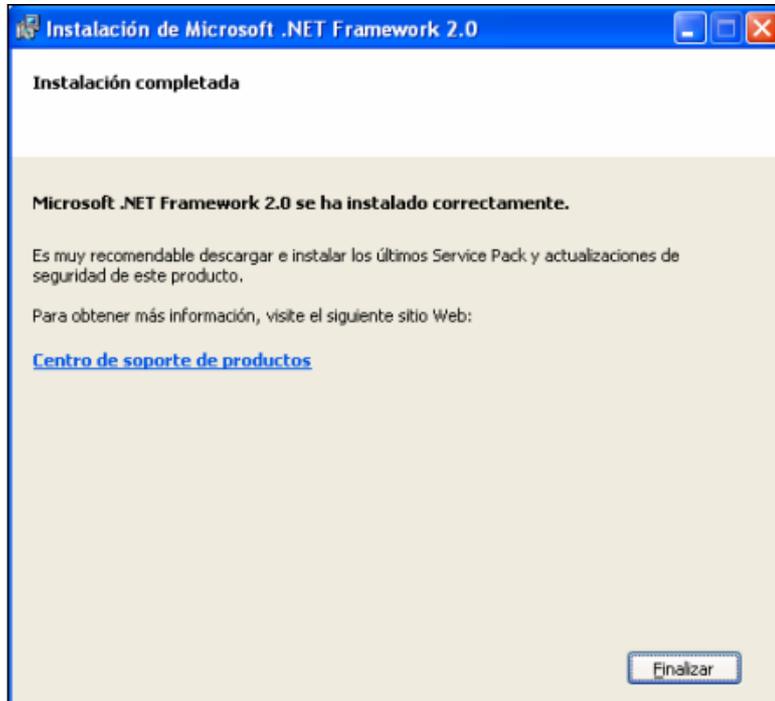


Figura A.10: Finalización de la instalación

Windows Installer 3.1

Es necesario realizar la instalación del Windows Installer 3.1 para realizar la instalación del MS SQL Server 2005 que se muestra en el siguiente apartado.

1. Bienvenida del instalador del Windows Installer 3.1. Presionar el botón *Siguiente* o *Next*.



Figura A.11: Bienvenida del instalador

2. Presentación del contrato de la licencia del software, en el cual se debe seleccionar la opción *Acepto* y presionar el botón *Siguiente* o *Next*

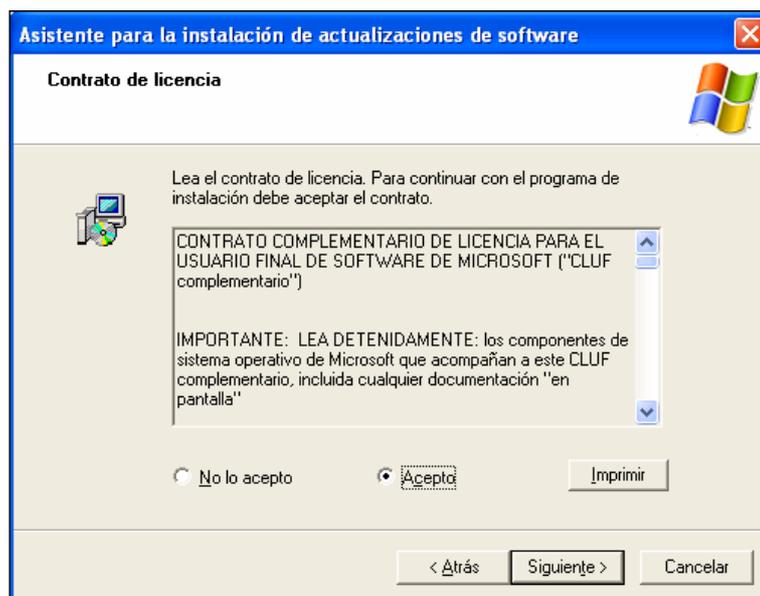


Figura A.12: Aceptación de los términos del contrato de licencia

3. A continuación se muestra el progreso de la instalación:

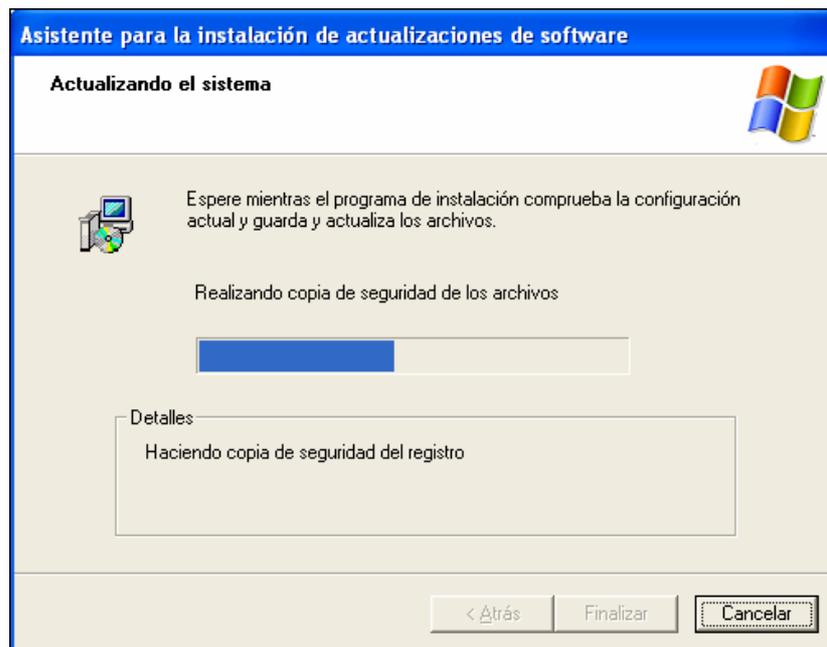


Figura A.13: Progreso de la instalación

4. La instalación del Windows Installer 3.1 ha finalizado correctamente.

Presionar el botón *Finalizar* o *Finish*.



Figura A.14: Finalización de la instalación

Instalación de MS SQL Server 2005

MS SQL Server 2005 es el motor de base de datos que utiliza el sistema Basys. Es necesario recalcar la importancia de tener en óptimas condiciones las conexiones a la base de datos. A continuación se detallarán los pasos necesarios para realizar la instalación de la base de datos SQL Server 2005 Standard Edition:

1. Para comenzar el proceso de instalación, se debe insertar el DVD de SQL Server 2005 en la unidad de DVD. Si la característica de ejecución automática de la unidad de DVD no inicia el programa de instalación, desplácese hasta la carpeta raíz del DVD e inicie *splash.hta*.

En el cuadro de diálogo de ejecución automática, presionar en el botón *Ejecutar* para que se muestre la ventana de opciones de instalación:

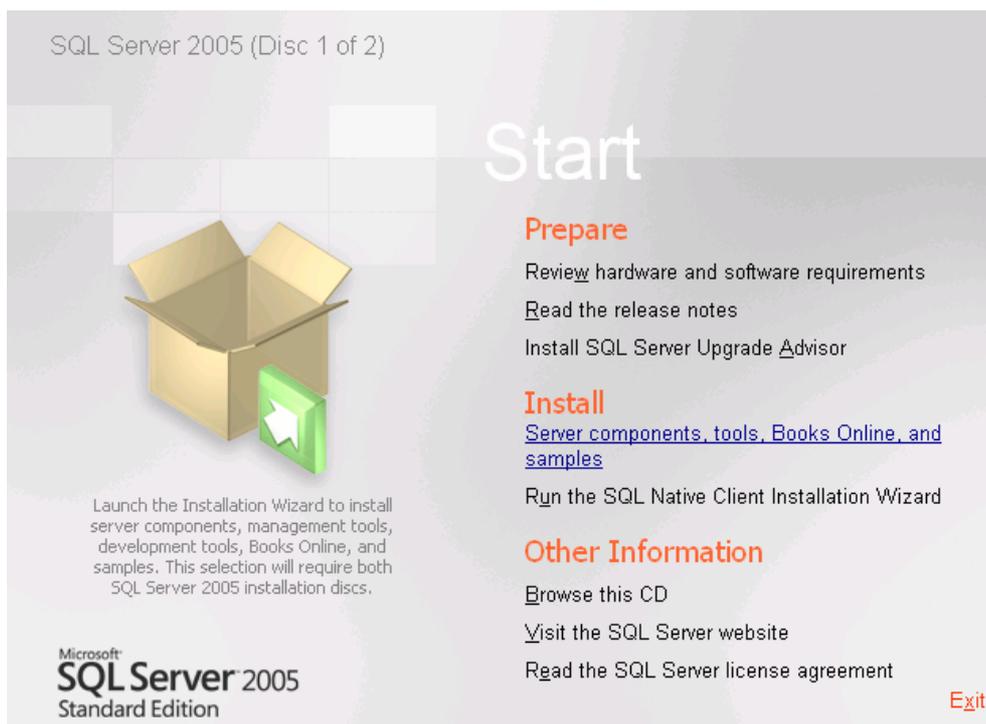


Figura A.15: Opciones que se pueden instalar

2. En la página Contrato de licencia para el usuario final, presione el checkbox para aceptar los términos y las condiciones de la licencia, y luego se da clic en el botón *Siguiente* o *Next*.

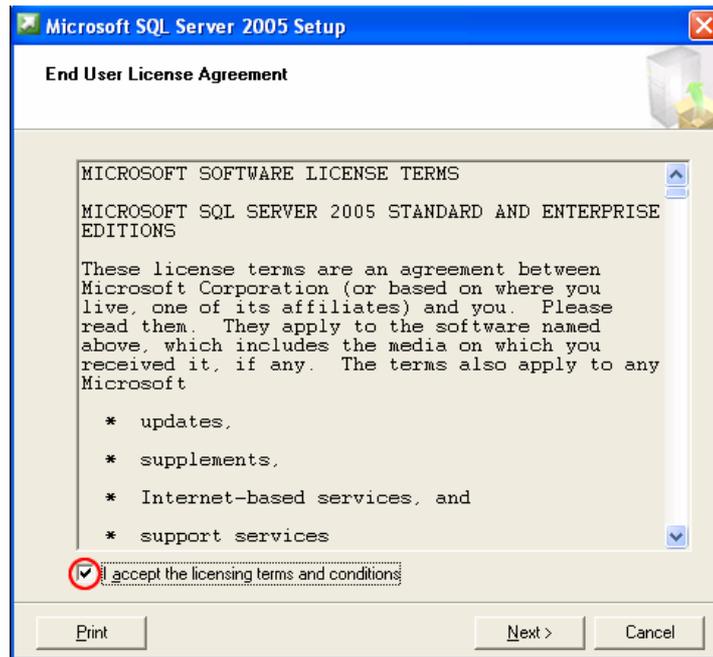


Figura A.16: Pantalla de Contrato de Licencia

3. En la página de Actualización de componentes de SQL Server, el programa de instalación instalará el software que precisa SQL Server 2005. Para comenzar el proceso de actualización de componentes, se debe presionar el botón *Instalar* o *Install*.

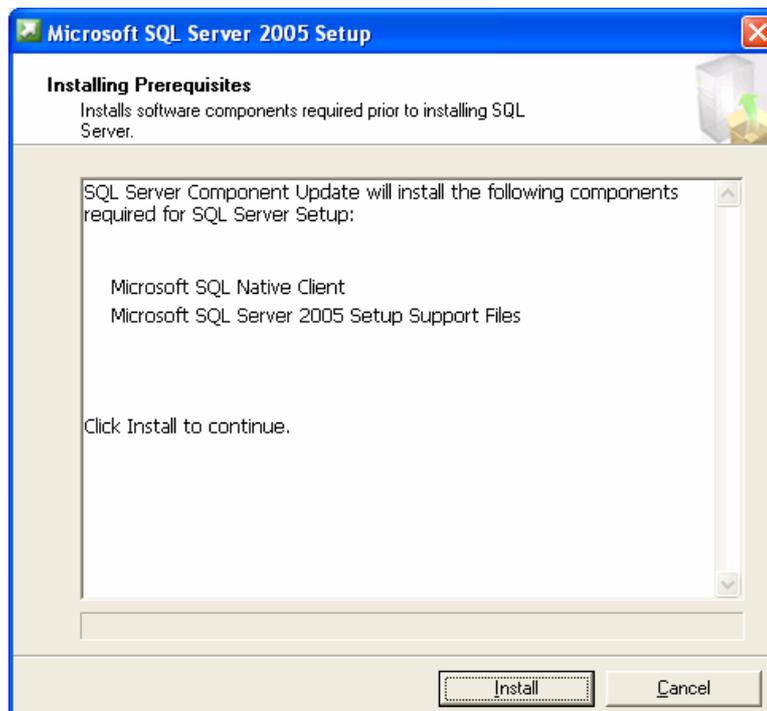


Figura A.17: Pantalla de de actualización de componentes

4. A continuación se muestra el progreso de la actualización de componentes.

Una vez finalizada la actualización presionar el botón *Siguiente* o *Next*.



Figura A.18: Progreso de la instalación

5. En la página de bienvenida del Asistente para la instalación de SQL Server, presionar el botón *Siguiente* o *Next* para continuar.



Figura A.19: Pantalla de Bienvenida a la instalación

6. En la página Comprobación de configuración del sistema (SCC), se examina el equipo en el que se va a realizar la instalación para detectar posibles condiciones que podrían bloquear el programa de instalación. Para interrumpir la exploración, presionar el botón *Detener* o *Stop*. Para continuar con el programa de instalación una vez finalizada la exploración de SCC, presionar el botón *Siguiente* o *Next*.

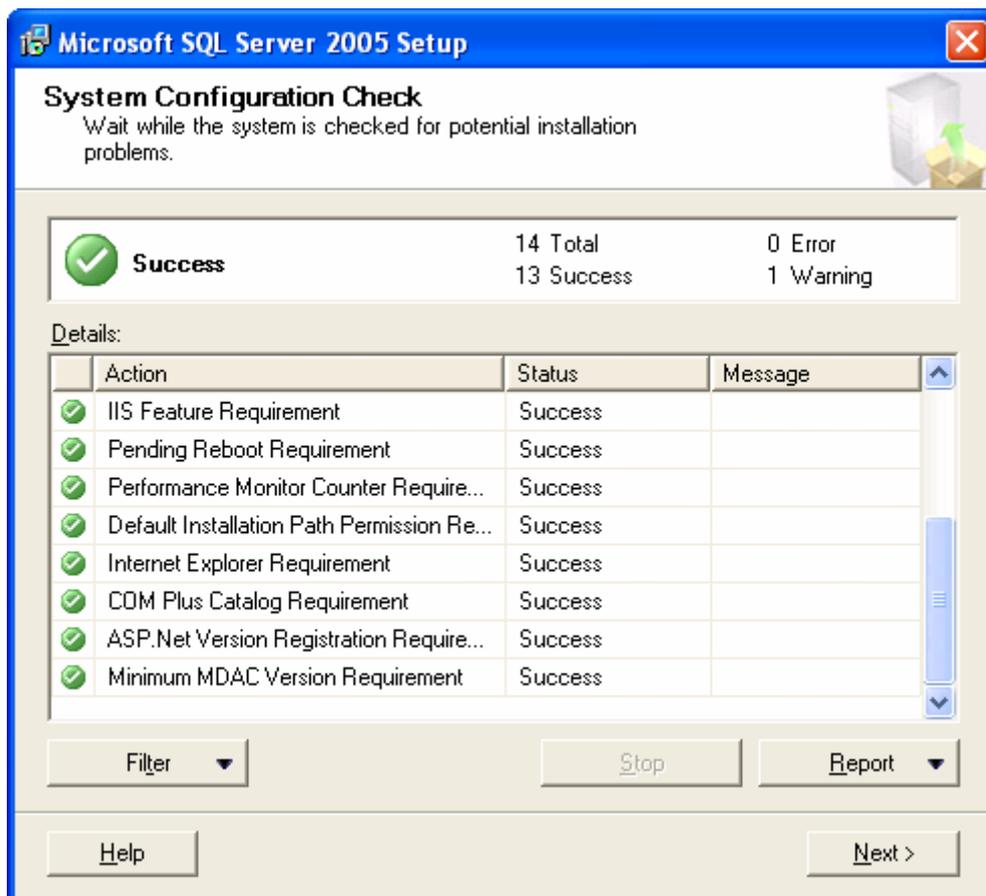


Figura A.20: Pantalla de Comprobación de Configuración del sistema

7. En la página Información de registro, escriba la información correspondiente en los campos *Nombre* o *Name* y *Compañía* o *Company*. Presione el botón *Siguiente* o *Next*.

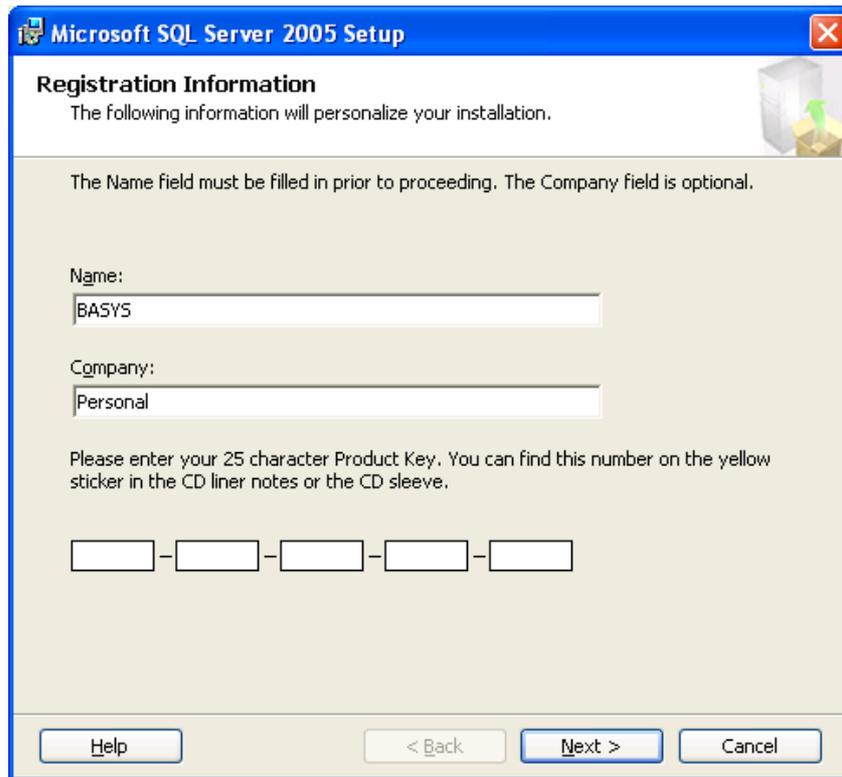


Figura A.21: Pantalla de registro

8. En la página Componentes para instalar, presione el botón *Avanzadas* o *Advanced* para seleccionar los componentes necesarios.

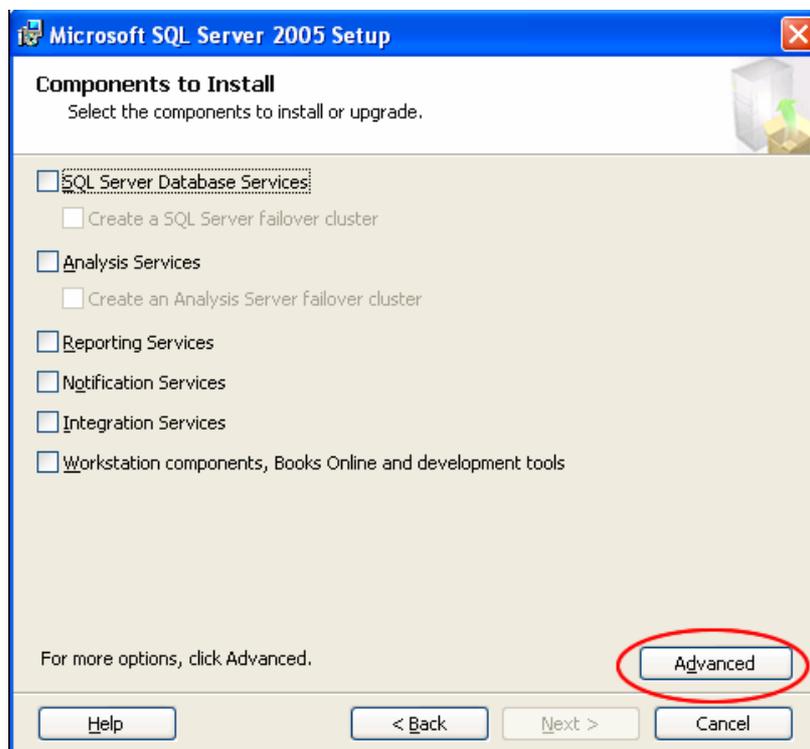


Figura A.22: Instalación de Componentes

9. En la selección de los componentes se deben seleccionar las subopciones siguientes: *Database Services*, *Reporting Services* y *Client Components*. Para lo cual se debe seleccionar el componente y presionar con el mouse en la flecha de color negro y escoger la opción *Entire feature will be installed on local hard drive*, como se muestra en la *Figura A.23*. Finalmente se debe presionar el botón *Siguiente* o *Next*.

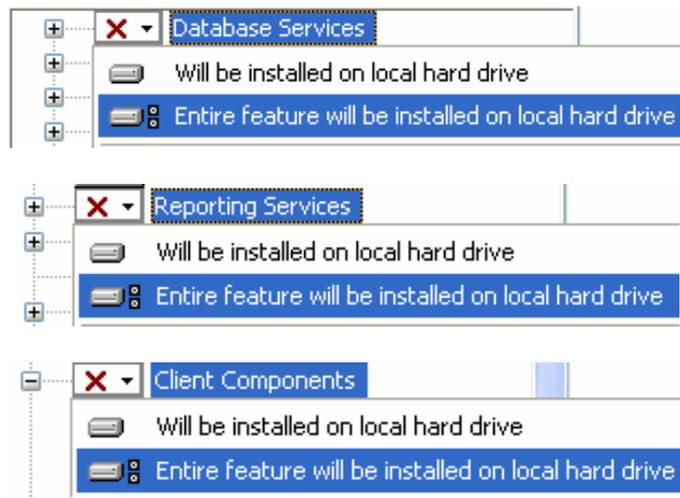


Figura A.23: Selección de las características de los componentes

10. En la página Nombre de instancia, seleccionar la opción *Default instante* o *Instancia por defecto* y dar clic en el botón *Siguiente* o *Next*.

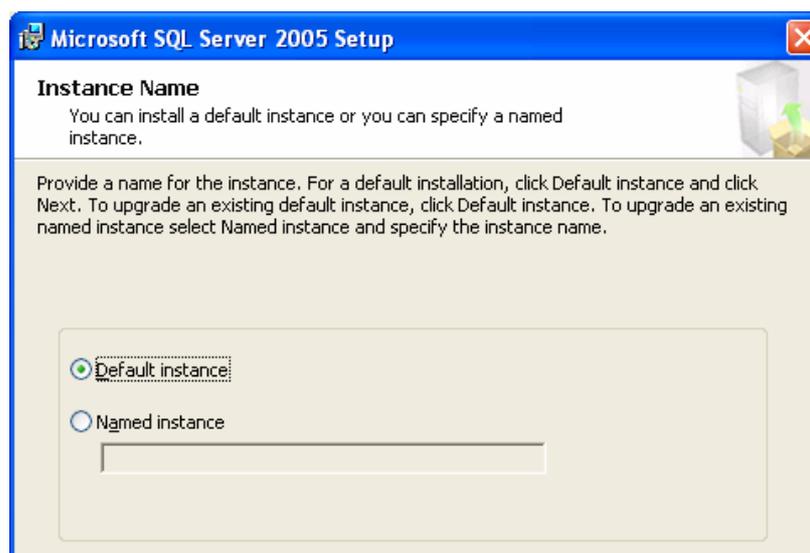


Figura A.24: Nombre de Instancia

11. En la página Cuenta de servicio, seleccionar la opción *Use the built-in System account* y en la lista desplegable la opción *Local System*, activar los check box de *SQL Server* y *Reporting Services*. Presionar el botón *Siguiente* o *Next*.

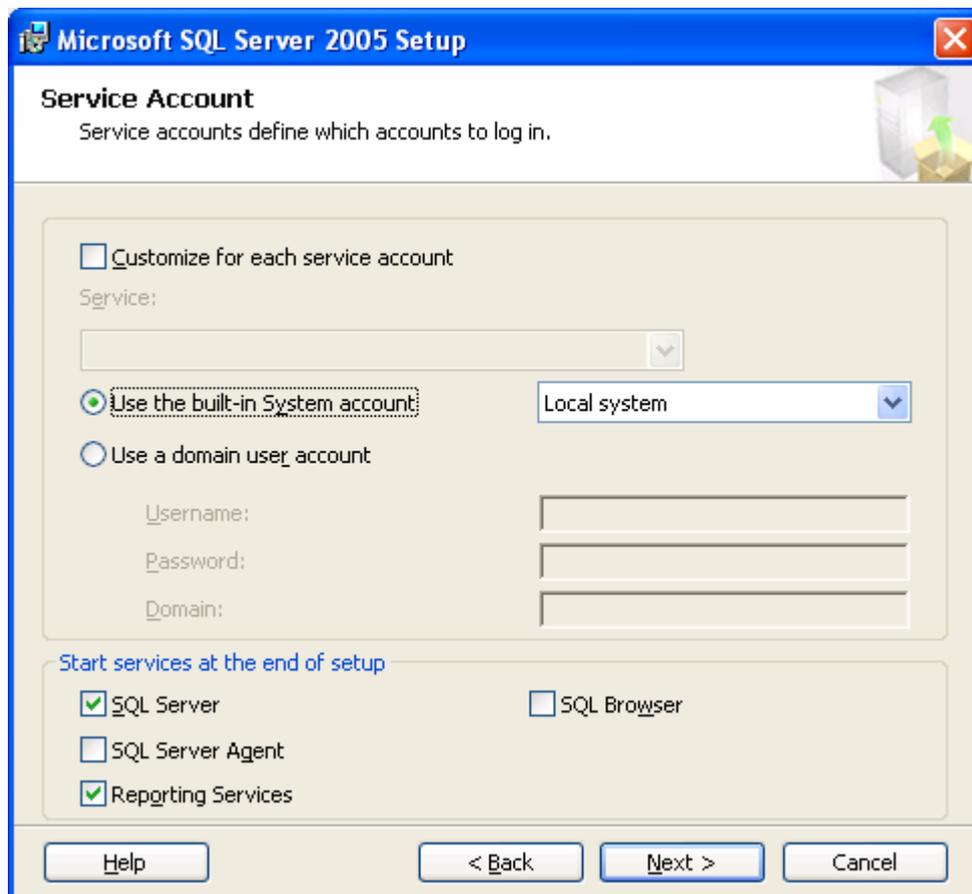


Figura A.25: Pantalla para Cuenta de Servicio

12. En la página Modo de autenticación, seleccionar el modo de autenticación *Mixed Mode*. Luego escribir y confirmar una contraseña segura para el inicio de sesión del usuario sa. Presionar el botón *Siguiente* o *Next*.

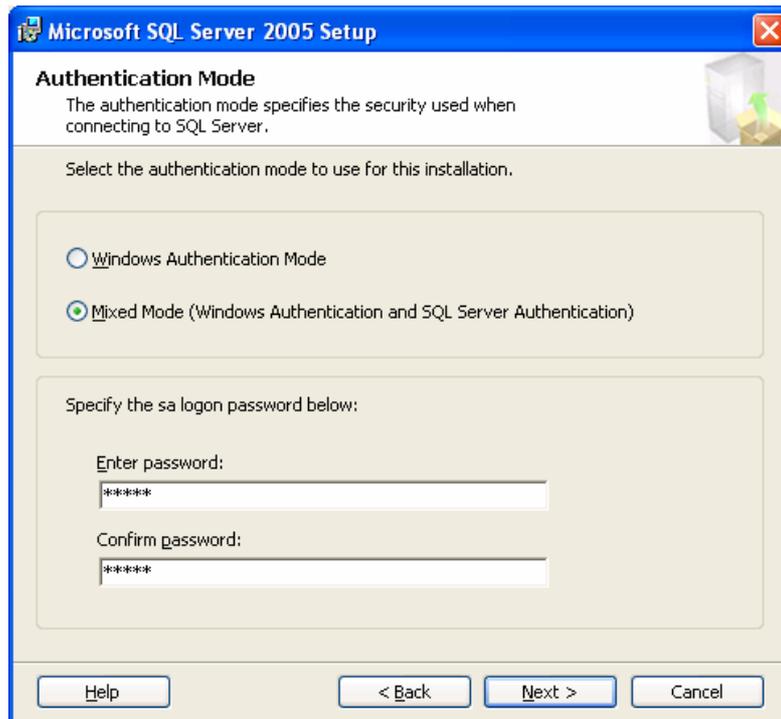


Figura A.26: Selección de Modo de Autenticación

13. En la página Configuración de intercalación, verificar que se encuentre activadas las opciones que se muestran en la *Figura A.27* a seguir:

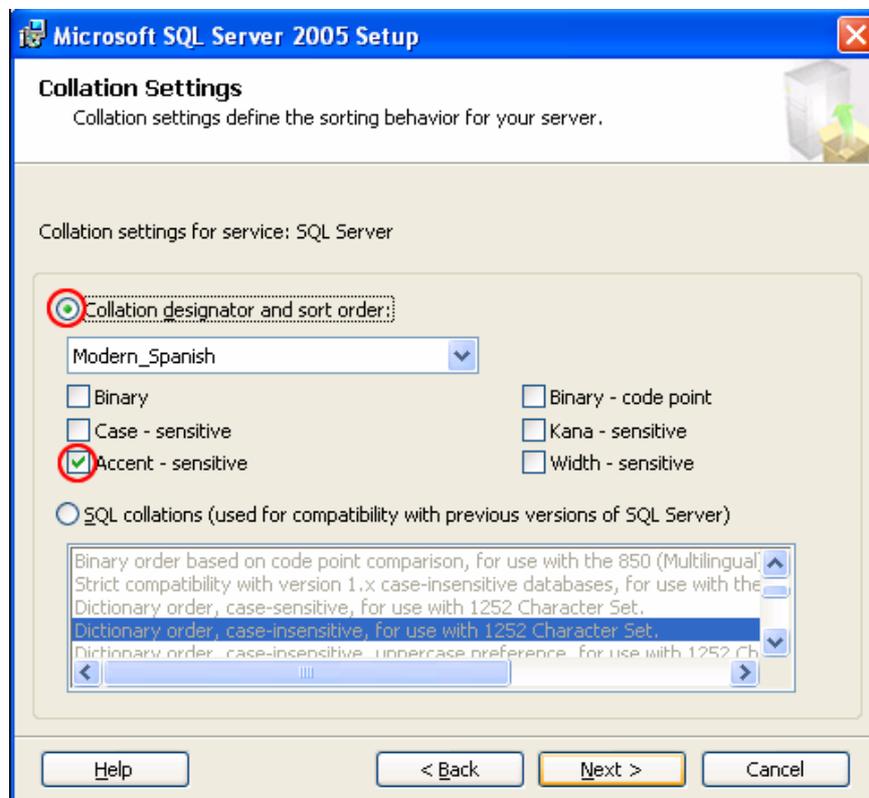


Figura A.27: Pantalla de configuración de intercalación

14. En la página Opciones de instalación del Servidor de reportes mantener la opción *Install the default configuration*. Presionar el botón *Siguiente* o *Next*.

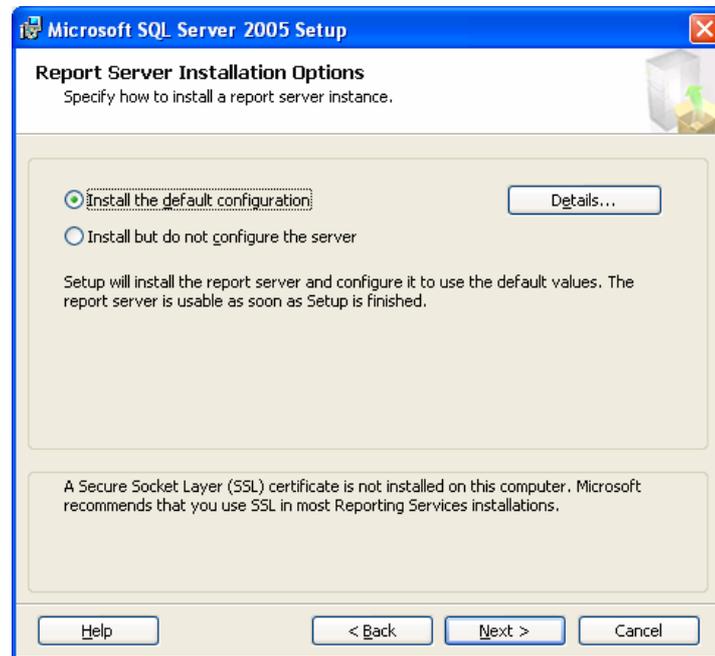


Figura A.28: Página de configuración de las opciones del servidor de informes

15. En la página Informes de errores, desactivar los check box para deshabilitar los informes de errores y presionar el botón *Siguiente* o *Next*.

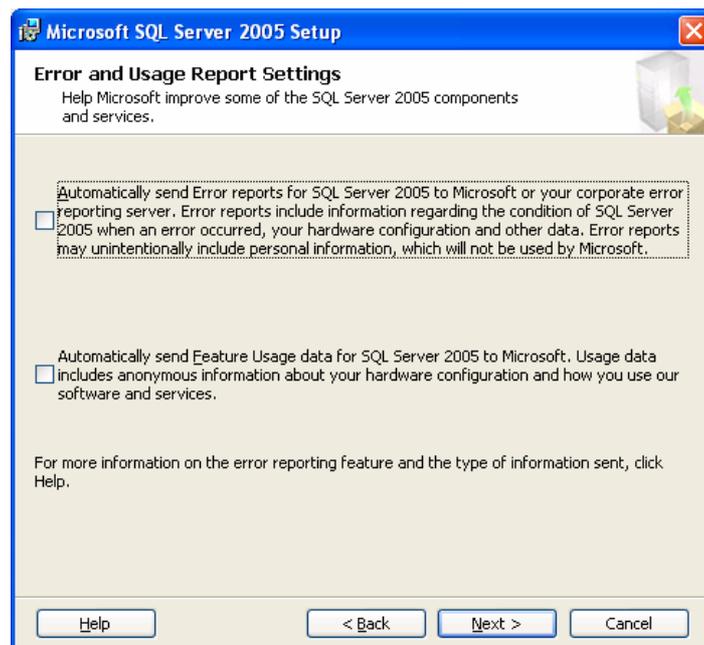


Figura A.29: Pantalla de Informe de Errores

16. En la página Progreso de la instalación, se puede supervisar el progreso de la instalación a medida que continúa el programa de instalación. Una vez finalizada la instalación presionar el botón *Siguiente* o *Next*.

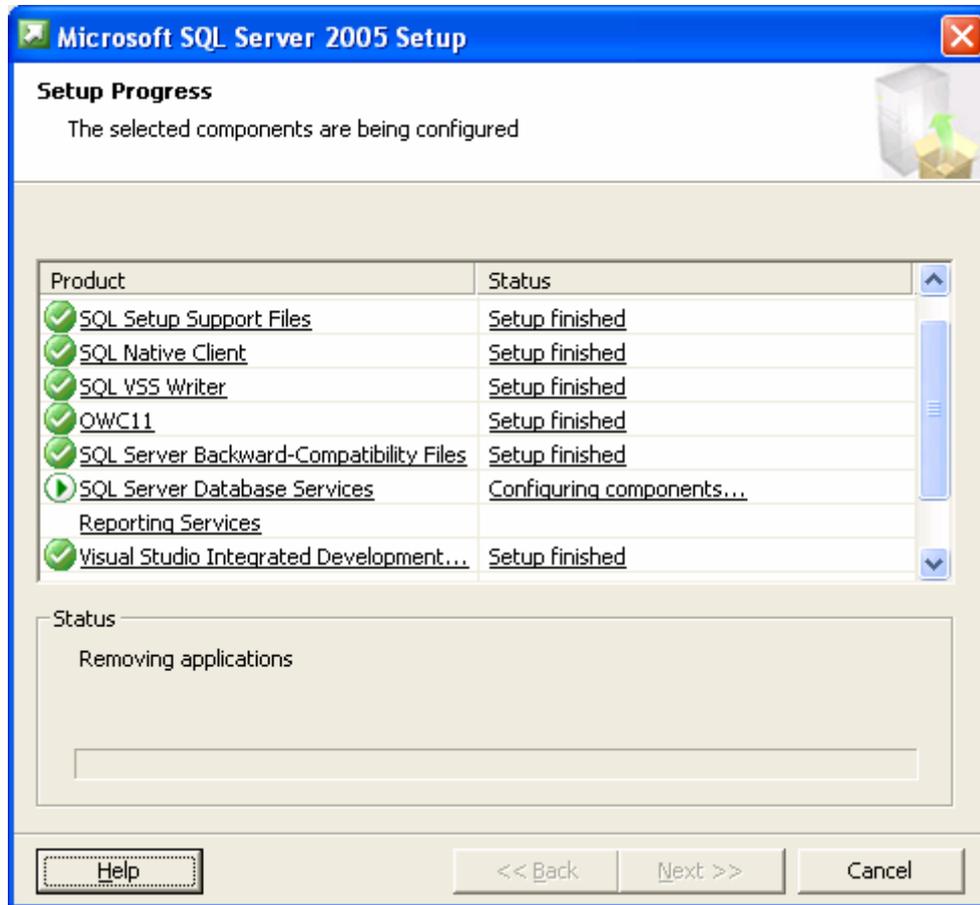


Figura A.30: Pantalla de progreso de la instalación

17. En la página Finalización del Asistente para la instalación de Microsoft SQL Server puede hacer clic en el vínculo disponible para ver el registro del resumen del programa de instalación. Para salir del Asistente para la instalación de SQL Server, presionar el botón *Finalizar* o *Finish*.



Figura A.31: Pantalla de finalización de instalación

Nota: es recomendable reiniciar el equipo para continuar con la instalación de Basys.

Configuración de la Base de Datos Basys

Una vez instalado el motor de base de datos, es necesario restaurar el repositorio de datos que utiliza el sistema Basys, el cual contiene las diferentes tablas que utiliza el mismo con sus respectivas relaciones; para lo cual se utiliza el archivo de respaldo²⁶ “*BD_BasysInicial.bak*”.

Subir el backup de la base de datos

El archivo para restaurar la base de datos de Basys se encuentra en el CD de instalación dentro de la carpeta denominada *BackupBaseDeDatos*. (Figura A.32).

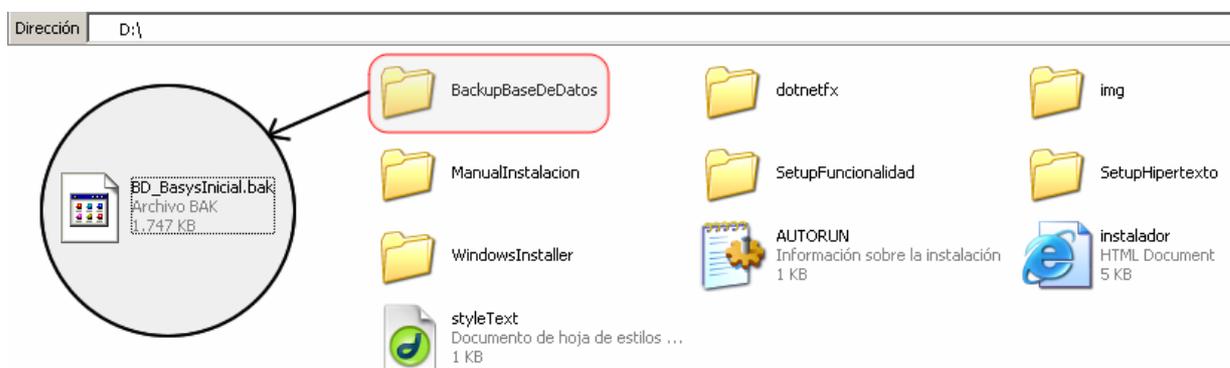


Figura A.32: Localización del backup de la base de datos

Para restaurar el backup se deben seguir los siguientes pasos:

1. Abrir el Administrador de SQL Server, para lo cual se debe seleccionar el *Menú Inicio - Microsoft SQL Server 2005 - SQL Server Management Studio*



Figura A.33: SQL Server Management Studio

²⁶ Un archivo de respaldo (*.bak*) de SQL Server se refiere a un backup que contiene la información base de una base de datos.

2. Conectarse al servidor de Base de datos (*Figura A.34*), en el cual se debe ingresar el usuario (*sa*) y contraseña definido en el proceso de la instalación de SQL Server.



Figura A.34: Conexión al servidor de base de datos

3. En la parte superior izquierda de la consola principal de SQL Server se encuentra un árbol que contiene los recursos disponibles por el administrador. Seleccionar la carpeta *Databases* y presionar el botón derecho del Mouse, en el submenú que se despliega seleccionar la opción *New Database...*

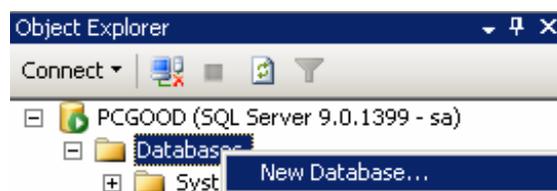


Figura A.35: Creación de una base de datos

4. A continuación se presenta una ventana de creación de la base de datos en la cual, en el campo *Database name* o *Nombre Base de datos* se escribe *Basys* y se presiona el botón *OK*.

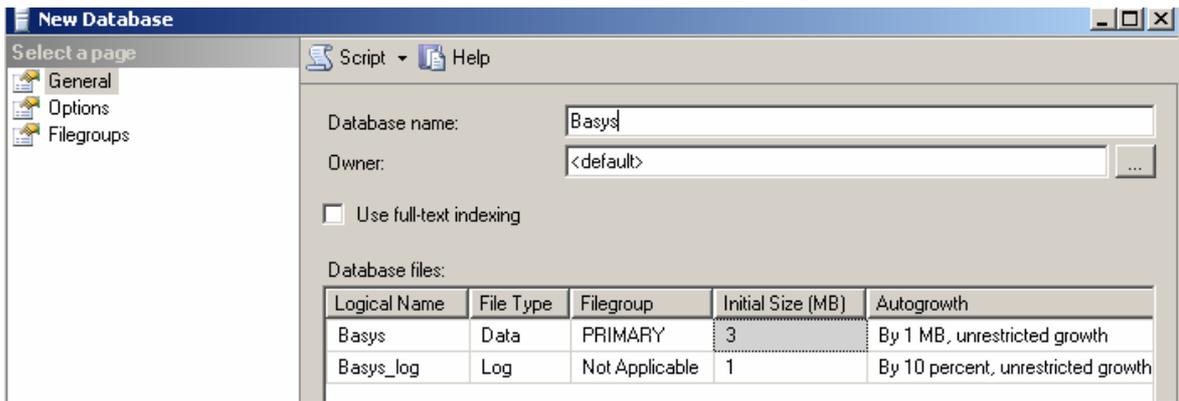


Figura A.36: Información de la nueva base de datos

5. En el árbol aparecerá la Base de Datos creada, al presionar clic derecho sobre ella se accede a *Taks* o *Tareas*, donde se presentan varias opciones de las cuales seleccionamos *Restore* o *Restaurar*, y finalmente en el submenú final se selecciona la opción *Database* o *Base de Datos*.

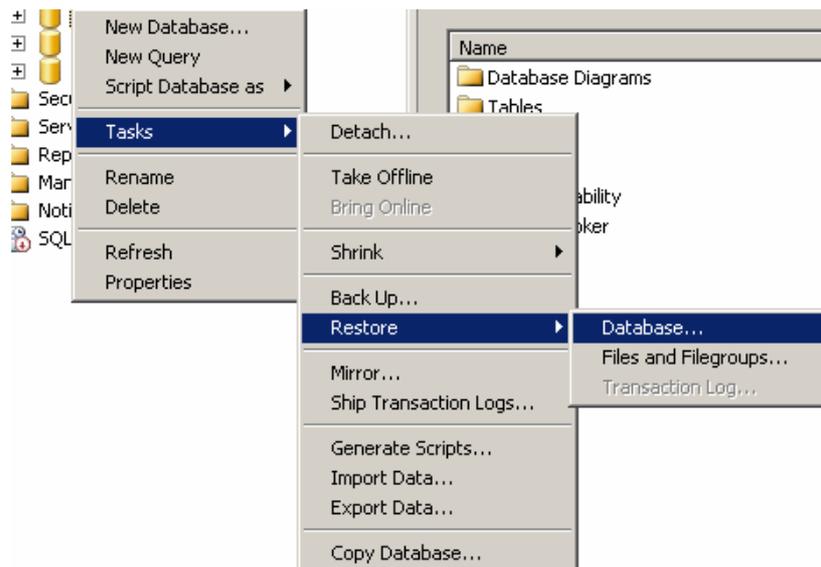


Figura A.37: Acceder a la opción de restauración

6. En la pantalla de opciones de restauración se debe seleccionar la opción *From device* y presionar el botón que se encuentra en el lado derecho  para especificar el archivo de restauración que contiene los datos iniciales y la estructura de la base de datos. En la ventana para la especificación del

medio se debe seleccionar el dispositivo que contiene la base de datos, en la cual se debe elegir *File* y presionar el botón *Add*.

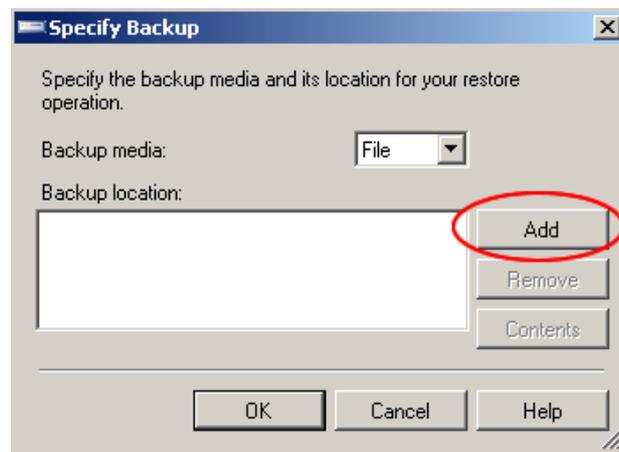


Figura A.38: Selección del medio de restauración

7. Seguidamente se debe ingresar a la localización del backup antes mencionada y seleccionar el archivo *BD_BasysInicial.bak*, y presionar el botón *OK*. (Figura A.39). Se regresa a la ventana de especificación del medio de restauración y se presiona *OK*. (Figura A.40)

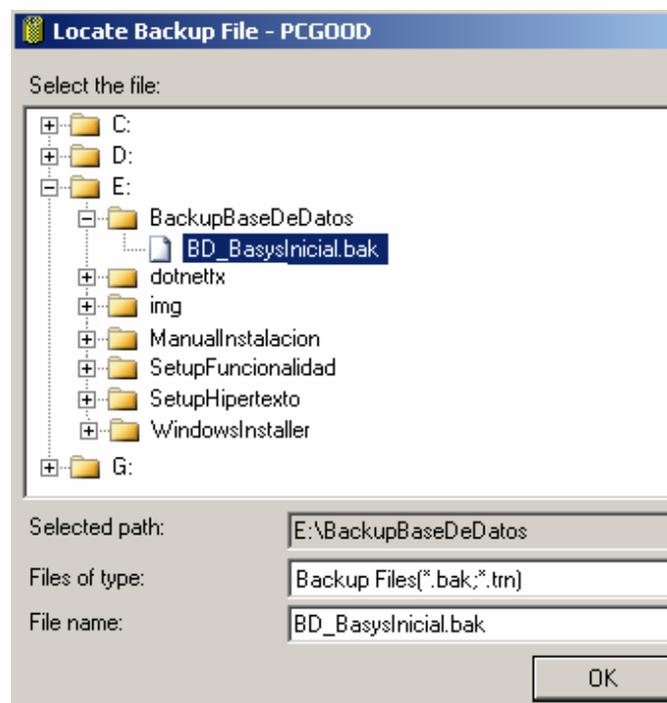


Figura A.39: Selección del archivo de restauración

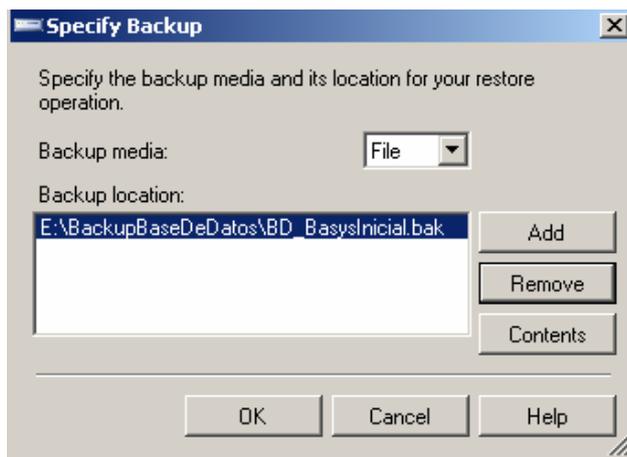


Figura A.40: Selección del medio de restauración completada

8. En la parte superior izquierda de la ventana principal de la restauración de la base de datos se encuentran dos opciones *General* y *Options*. En la vista de la opción *General* se debe seleccionar el check box de la base de datos que se desea restaurar, como se muestra en la *Figura A.41* a seguir:

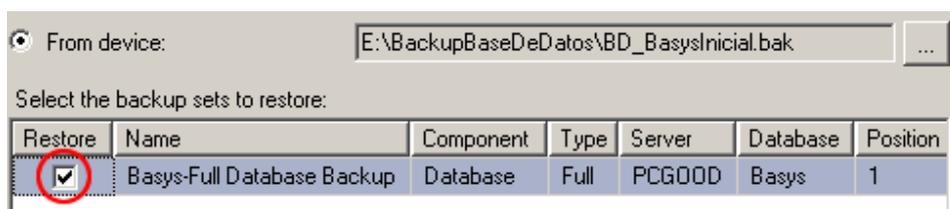


Figura A.41: Selección de la restauración a ejecutarse

9. En la vista *Options*, se debe seleccionar la primera opción que permite sobrescribir los datos existentes y finalmente se presiona el botón *OK*.

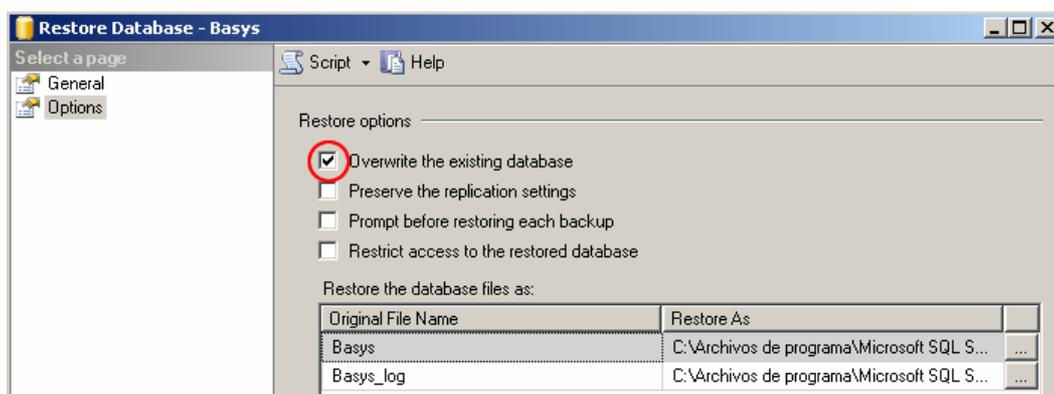


Figura A.42: Selección de sobre escritura de la base de datos

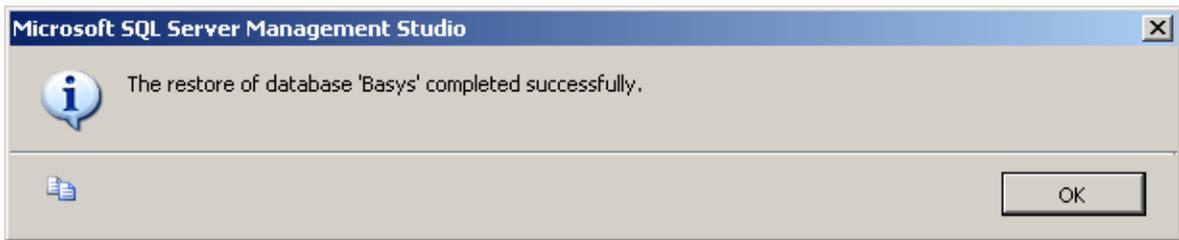


Figura A.43: Restauración de la base de datos exitosa

Reportes de Basys

Cuando se realizó la instalación de la base de datos se instaló un modulo denominado Business Intelligence el cual permite la generación de reportes utilizando la herramienta Reporting Services.

Es necesario subir estos reportes en el servidor de la aplicación, para lo cual se deben seguir los siguientes pasos:

1. Nos ubicamos en el Menú Inicio – Microsoft SQL Server 2005 y seleccionamos el utilitario SQL Server Business Intelligence Development Studio



Figura A.44: Abrir el SQL Server Business Intelligence Development Studio

2. Abrimos el proyecto de reportes de Basys, para lo cual damos clic en el botón *Abrir Archivo*, como se muestra en la Figura A.45 a continuación:

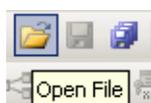


Figura A.45: Abrir archivo

3. Debe seleccionar la unidad de CD, luego abrir la carpeta *ReportesBasys* en el cual se encuentra el proyecto de reportes denominado *ReportesBasys.sln*, selecciona dicho archivo y presiona el botón Aceptar.

(Figura A.46)

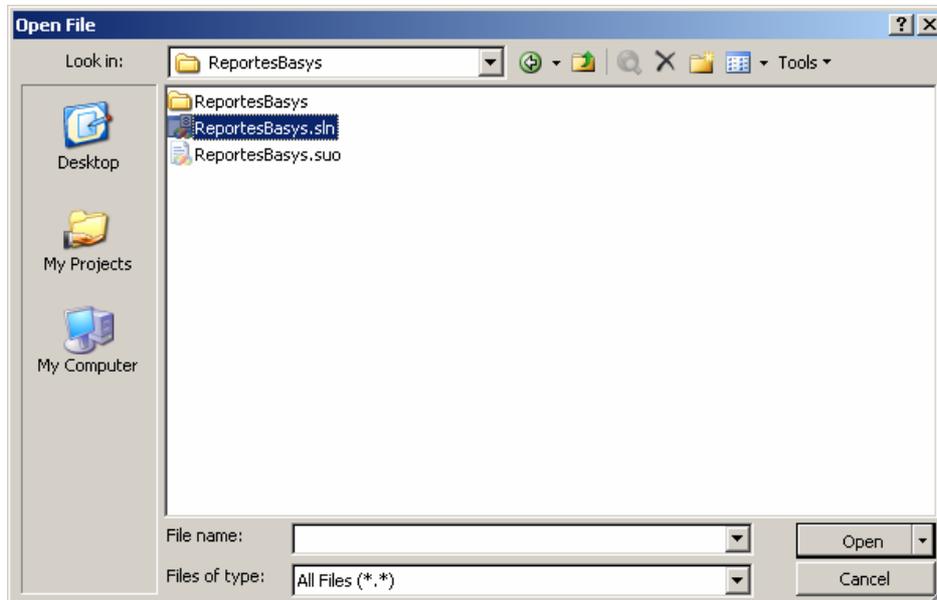


Figura A.46: Seleccionar el proyecto de reportes

4. Finalmente se selecciona el proyecto *ReportesBasys* y se presiona el botón derecho del Mouse, seguidamente se presenta un submenú en el cual se debe seleccionar la opción *Deploy*. Esta acción suele tardar algunos minutos dependiendo de la velocidad del procesador del equipo.

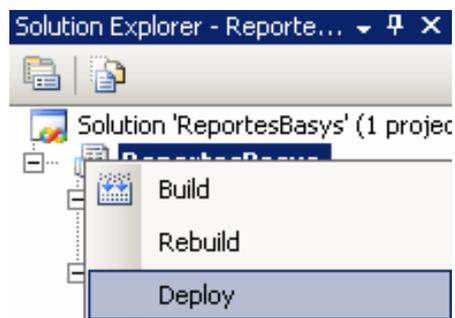


Figura A.47: Subir los reportes de Basys

Basys Funcionalidad

La Funcionalidad de Basys corresponde a los XML Web Services que utiliza el Sitio Web y es el que provee la lógica de la aplicación, por lo que su instalación previa es fundamental.

Corresponde a la sexta opción del menú de instalación, y para proceder a su instalación se debe dar clic en dicha opción. A continuación se visualizará un Wizard que le ayudará a realizar una instalación exitosa.

1. Primeramente se despliega una ventana que permite descargarse el archivo o ejecutarlo desde su ubicación actual. En este paso se debe presionar el botón *Ejecutar*.

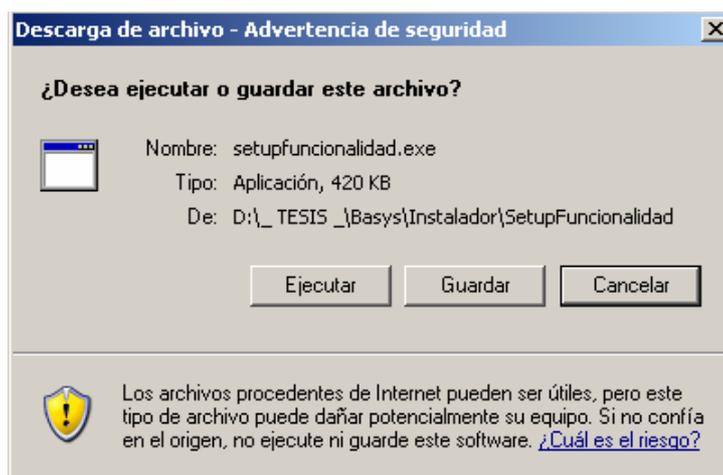


Figura A.48: Descarga de archivo

2. En la ventana siguiente debe presionar el botón *Ejecutar*.



Figura A.49: Verificación del fabricante

3. Seguidamente se presenta la bienvenida del instalador, en la cual seleccionamos el botón *Next* o *Siguiente*.

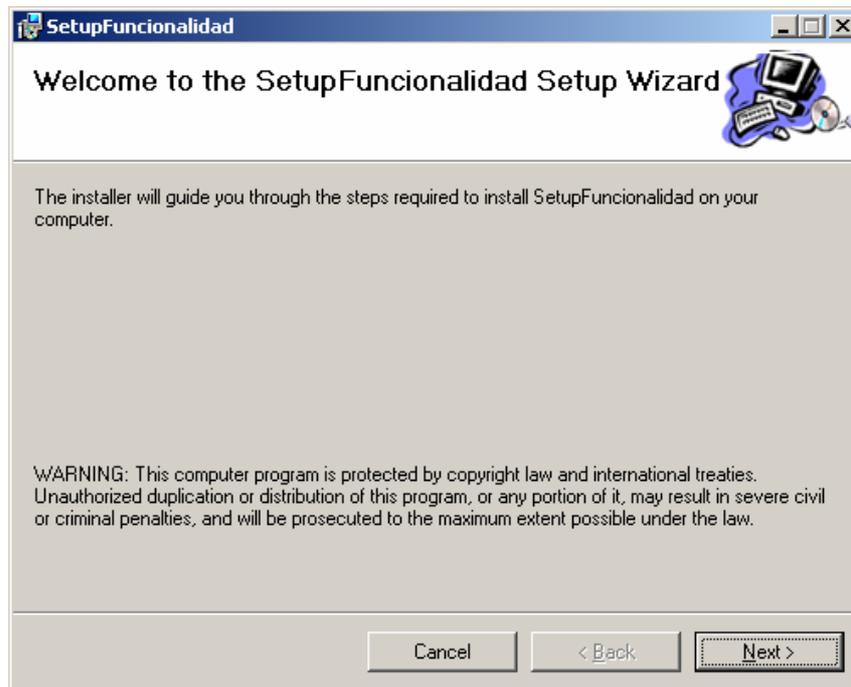


Figura A.50: Bienvenida del SetupFuncionalidad

4. Se presenta el nombre del sitio y el directorio virtual a crearse, se deben dejar los valores por defecto y presionar el botón *Next* o *Siguiente*.

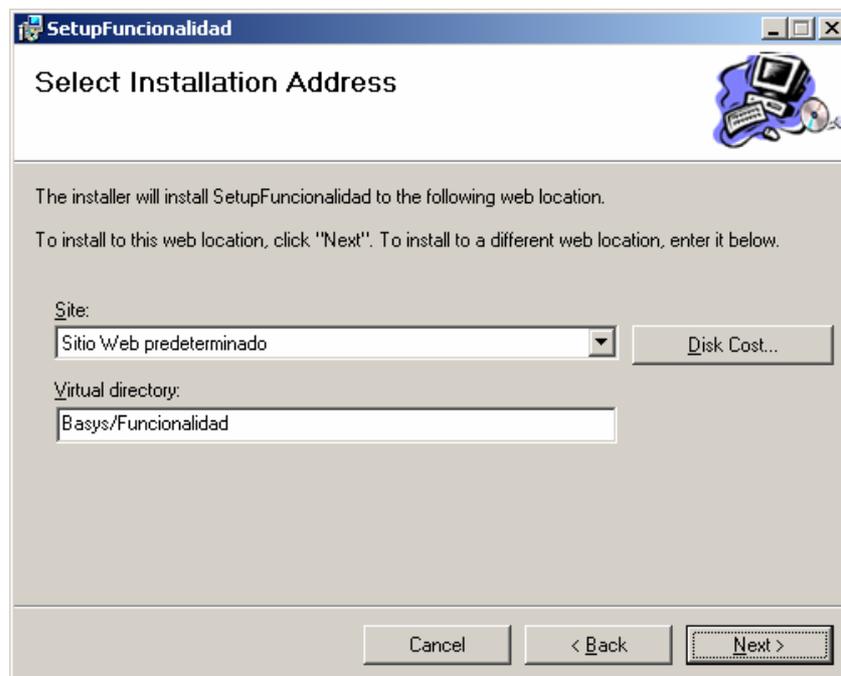


Figura A.51: Información de la Funcionalidad

5. Una vez determinada la información anterior, el instalador se encuentra listo para realizar la instalación. Para empezar con ella se debe seleccionar el botón *Next* o *Siguiente*.



Figura A.52: Confirmación de la instalación

6. A continuación se presenta una barra del progreso de la instalación.

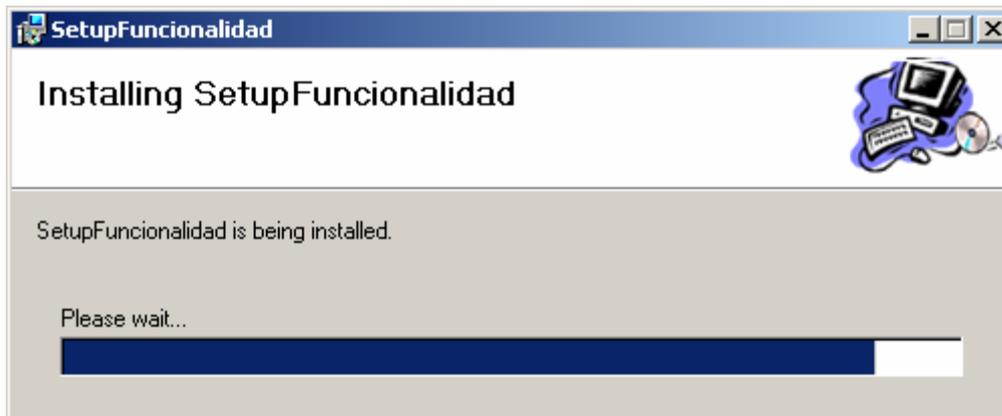


Figura A.53: Instalando la funcionalidad del sistema

7. Una vez finalizada la instalación de la funcionalidad del sistema, se debe cerrar el Wizard presionando el botón *Close* o *Cerrar*.

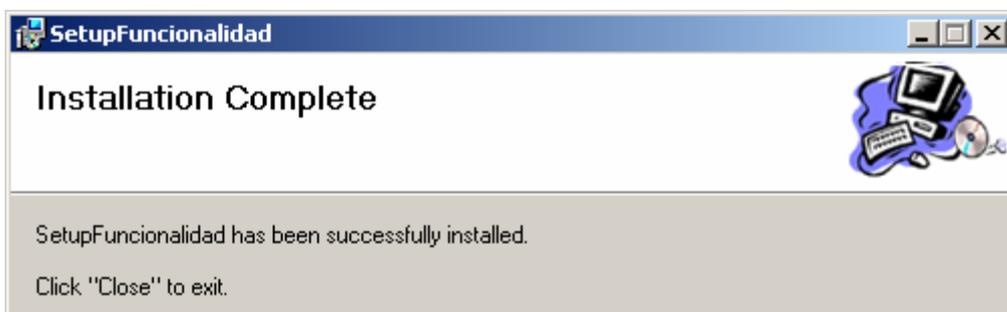


Figura A.54: Instalación completada

Personalización del archivo de configuración

Una vez realizada la instalación utilizando el Wizard es necesario configurar el archivo de configuración para especificar el nombre del servidor donde se encuentra instalada la Base de datos. Para lo cual se debe acceder al directorio virtual creado el cual se encuentra en la siguiente dirección:

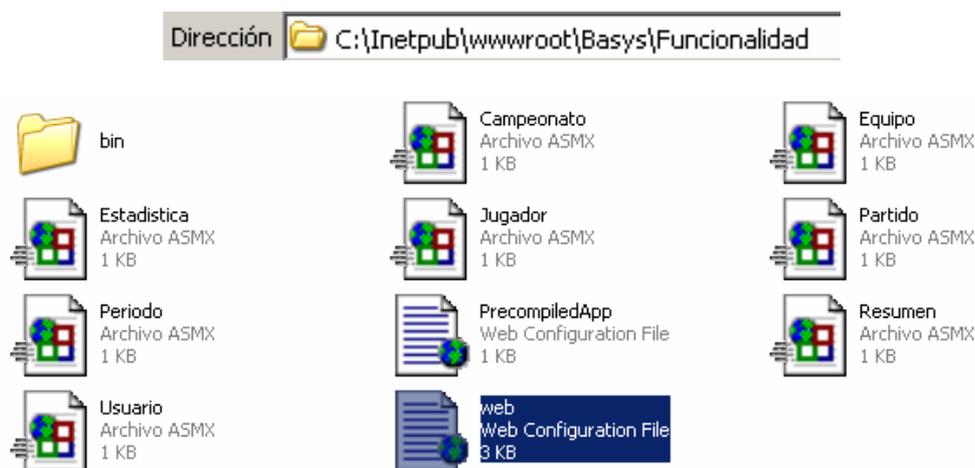


Figura A.55: Contenido del directorio virtual

En donde se encuentra el archivo *web.config* el cual se debe abrirlo con un editor de texto como el *notepad* como se muestra en la *Figura A.56* a seguir:



Figura A.56: Editar el archivo de configuración

En el apartado `<configuration>`, `<connectionStrings>` se encuentra la cadena de conexión a la base de datos, siendo esta la siguiente:

```
<add name="BasysConnectionString"
connectionString="Password=Contraseña;Persist Security Info=True;User
ID=sa;Initial Catalog=Basys;Data Source=Nombre_servidor"/>
```

Se deben realizar dos cambios en el archivo de configuración:

1. La contraseña para conectarse a la base de datos.- Para lo cual en el lugar donde se encuentra **Contraseña** se debe especificar la contraseña que se definió en la instalación.
2. El nombre del servidor.- de igual manera donde se encuentra **Nombre_servidor** se debe especificar el nombre del servidor donde se encuentra instalada la base de datos.

Basys Hipertexto

El Hipertexto de Basys corresponde al Sitio Web de Basys al cual van a acceder los usuario para utilizar el sistema.

Corresponde a la séptima opción del menú de instalación, y para proceder a su instalación se debe dar clic en dicha opción. A continuación se visualizará un Wizard que le ayudará a realizar una instalación exitosa.

1. Primeramente se despliega una ventana que permite descargarse el archivo o ejecutarlo desde su ubicación actual. En este paso se debe presionar el botón Ejecutar.

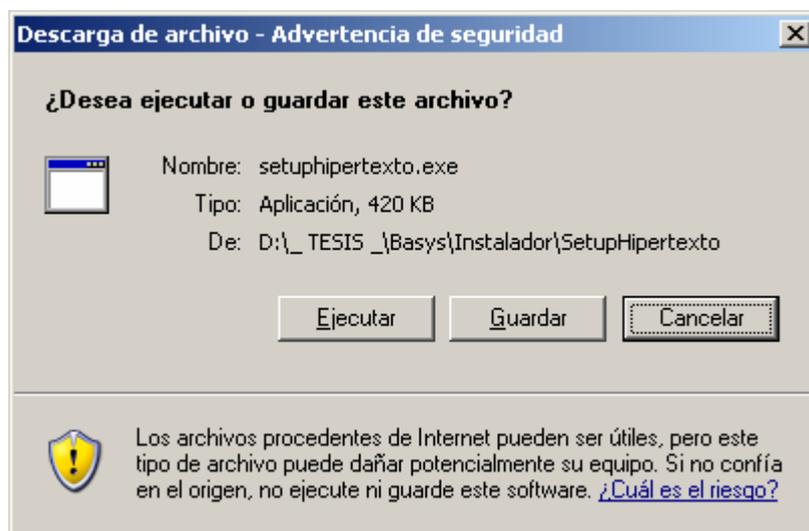


Figura A.57: Descarga de archivo

2. En la ventana siguiente debe presionar el botón *Ejecutar*.

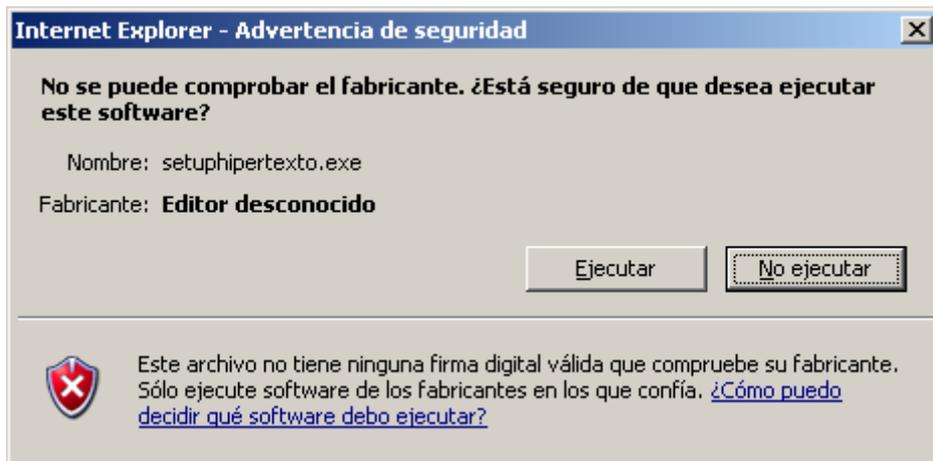


Figura A.58: Verificación del fabricante

3. Seguidamente se presenta la bienvenida del instalador, en la cual seleccionamos el botón *Next* o *Siguiente*.



Figura A.59: Bienvenida del SetupHipertexto

4. Se presenta el nombre del sitio y el directorio virtual a crearse, se deben dejar los valores por defecto y presionar el botón *Next* o *Siguiente*.

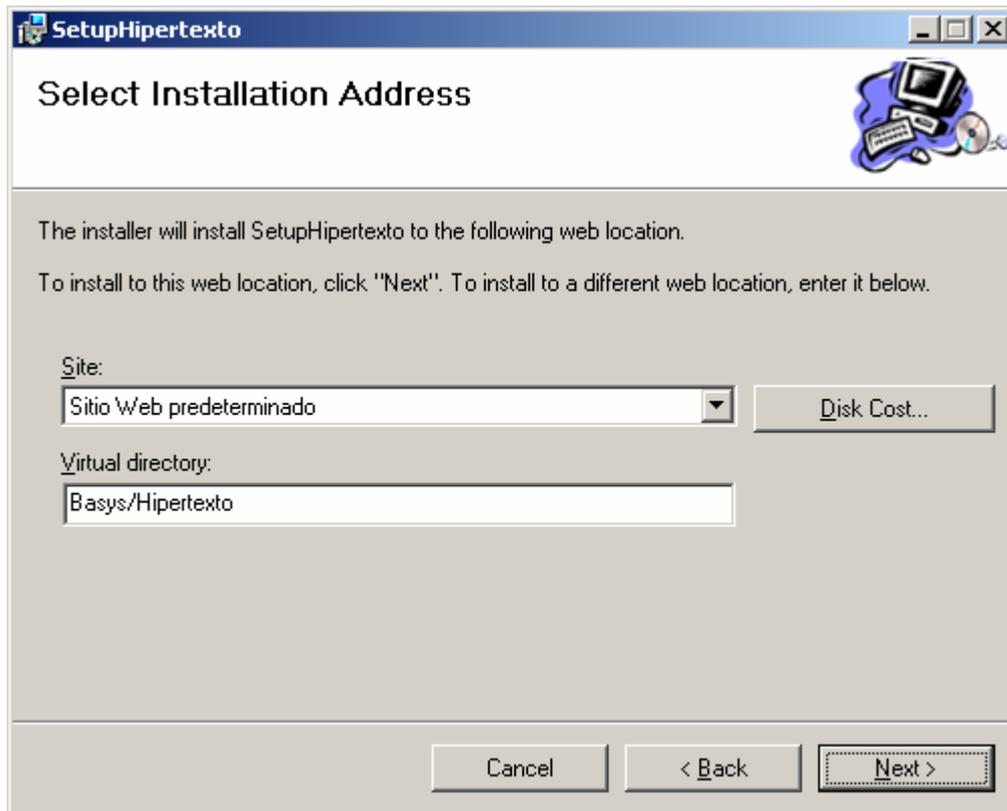


Figura A.60: Información de la Funcionalidad

5. Una vez determinada la información anterior, el instalador se encuentra listo para realizar la instalación. Para empezar con ella se debe seleccionar el botón *Next* o *Siguiente*.

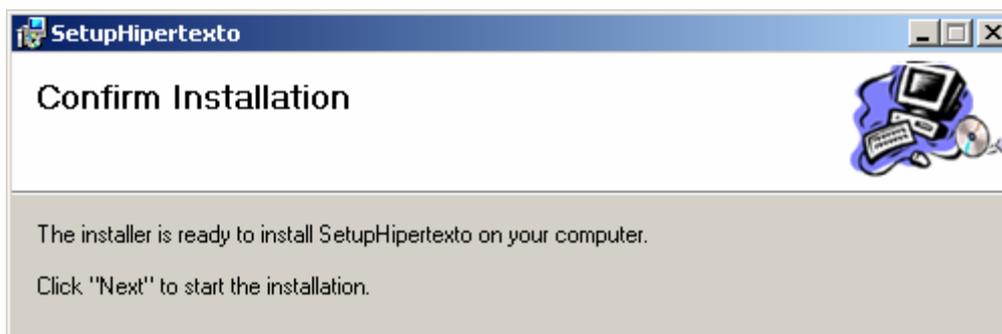


Figura A.61: Confirmación de la instalación

6. A continuación se presenta una barra del progreso de la instalación.

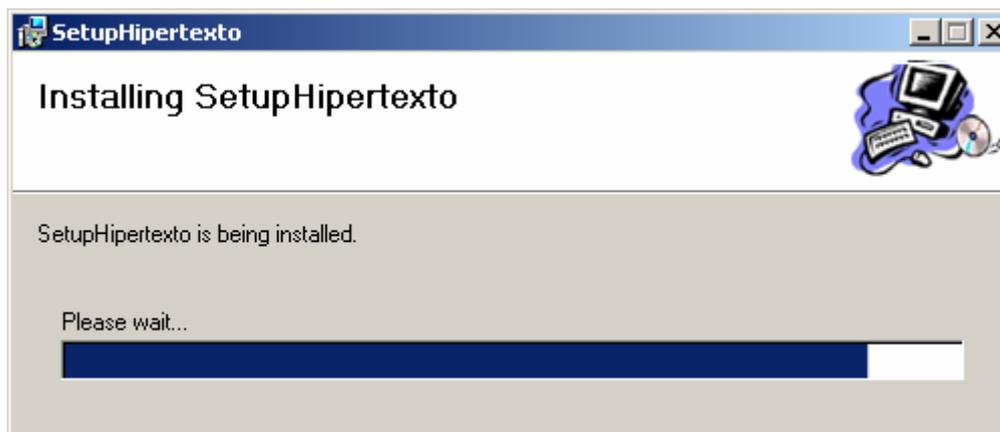


Figura A.62: Instalando la funcionalidad del sistema

7. Una vez finalizada la instalación del hipertexto del sistema, se debe cerrar el Wizard presionando el botón *Close* o *Cerrar*.

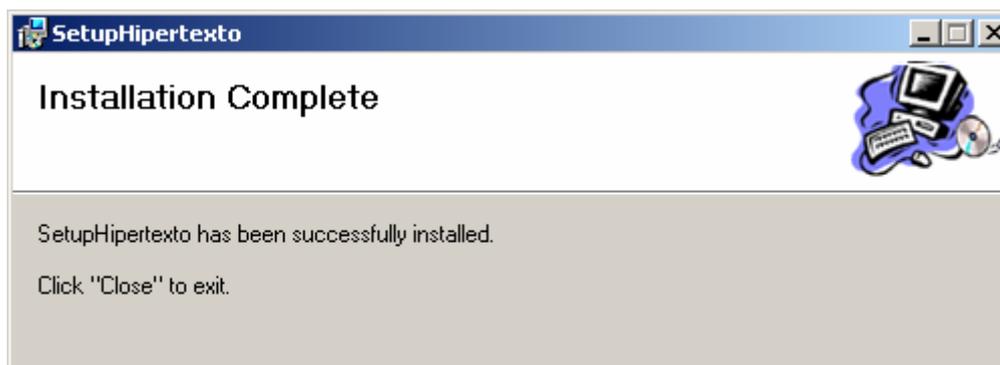


Figura A.63: Instalación completada

Para verificar que el sistema corra sin ninguna novedad se debe abrir una ventana de Internet Explorer y digitar la dirección IP o el nombre del servidor seguido del nombre del directorio virtual *Basys* y el nombre del Sitio Web *Hipertexto*, como se muestra a continuación:



Figura A.64: Dirección Web para ejecutar el sistema Basys

ANEXO B
MANUAL DE USUARIO

INDICE

MANUAL DE USUARIO	338
INTRODUCCIÓN	338
FUNCIONAMIENTO	339
Características comunes en Basys	339
Campos Obligatorios	339
Campos NO Obligatorios	339
Restricciones en la eliminación de datos	340
Desplazamiento de la información	340
AMBIENTES DE BASYS	341
Salir de la aplicación	344
Cambio de contraseña	344
NAVEGANDO POR BASYS	345
Opción Usuario	345
Ingresar un nuevo Usuario	346
Seleccionar un Usuario	347
Modificar un Usuario	348
Eliminar un Usuario	348
Asignar el Perfil a un Usuario	348
Opción Jugador	350
Ingresar un nuevo Jugador	351
Buscador de Jugadores	352
Seleccionar un Jugador	353
Modificar un Jugador	353
Eliminar un Jugador	353
Opción Equipo	354
Ingresar un nuevo Equipo	355
Buscador de Equipos	356
Seleccionar un Equipo	356
Modificar un Equipo	356
Eliminar un Equipo	356
Opción Campeonato	357
Ingresar un nuevo Campeonato	358
Buscador de Campeonatos	359
Seleccionar un Campeonato	359
Modificar un Campeonato	359
Eliminar un Campeonato	359
Opción Partido	360
Ingresar un nuevo Partido	361
Buscador de Partidos	363
Seleccionar un Partido	363
Modificar un Partido	364
Opción Información del Partido	364
Ingreso de Jugadores e Inicio del Partido	366
Registro de los datos del partido	369
Cambio de un jugador	370
Estados del partido	370
Calcular las estadísticas	371
Opción Reportes	373
Partido	374
Estadísticas del Jugador por Partido	376
Estadísticas del Jugador por Campeonato	377
Estadísticas del Equipo por Campeonato	378
Jugador más valioso	379
Equipo más valioso	380

ANEXO B

MANUAL DE USUARIO

INTRODUCCIÓN

Basys, es un sistema que permite llevar un control real de los datos, mediante el cual se pueden obtener resultados estadísticos precisos y diferentes reportes de acuerdo a las necesidades que presenta el usuario. La información que se administra es sobre: Usuarios, Jugadores, Equipos, Campeonatos y Estadísticas.

El presente documento hace referencia a la manera eficiente de utilizar el Sistema Basys. Su contenido indica la forma de utilizar cada una de las opciones y procesos que permiten administrar los cinco módulos antes propuestos.

De esta manera, Basys es un sistema que busca ayudar en el proceso de:

- *Seguridad:* mediante el uso de usuarios y contraseña se controlará el ingreso al sistema en base a perfiles.
- *Jugadores:* Se llevará información de todos y cada uno de los jugadores registrados para competencias internas o externas.
- *Equipos:* Se mantendrá el control de todos los equipos que quieran registrarse en los campeonatos que organice la FEFDER o de las selecciones de basketball de la ESPE que participen en competencias organizadas por otras instituciones.
- *Campeonatos:* Se podrán registrar todas las competencias en la disciplina de baloncesto que la FEFDER quiera administrar, sean estas internas o en las cuales participen las diferentes selecciones de la ESPE.

- *Estadísticas:* Se recogerán estadísticas por jugador en todos y cada uno de los partidos jugados. Además se entregará los reportes necesarios por jugador, equipo y campeonato.

FUNCIONAMIENTO

Características comunes en Basys

Es importante establecer desde un principio ciertas características comunes que aparecen en las diferentes secciones de la aplicación. Estas características permiten al usuario identificar ciertas acciones que debe realizar. A continuación se detallan las mismas:

Campos Obligatorios

De una forma general, en la hora de ingresar o modificar información, siempre hay al menos un campo que es obligatorio. Es decir, es indispensable que el usuario ingrese esos campos para poder almacenar la información. Para identificar los campos que son obligatorios, las celdas tienen un borde de color marrón como se muestra en la Figura B.1.

The image shows a form field with the label 'Nombre:' on the left and a rectangular input box on the right. The input box has a prominent red border, which, according to the text, indicates that this field is mandatory.

Figura B.1: Campo obligatorio

Campos NO Obligatorios

De una forma general, en la hora de ingresar o modificar información, se puede encontrar campos que no son obligatorios. Es decir, NO es indispensable que el usuario ingrese estos campos para poder almacenar la información. Para

identificar los campos que NO son obligatorios, las celdas no presentan un borde como se muestra en la Figura B.2.

A screenshot of a web form showing a label 'Teléfono Casa:' followed by an empty text input field. The input field does not have a visible border, indicating it is not a required field.

Figura B.2: Campo NO obligatorio

Restricciones en la eliminación de datos

La eliminación de un dato requiere ser buscado previamente y luego presionar en el botón de eliminación para borrar permanentemente el dato seleccionado. Sin embargo, cuando este dato esté siendo usado en otro lugar de la aplicación o esté asignado a otro tipo de dato, entonces la eliminación no podrá ejecutarse y la aplicación mostrará el mensaje de error correspondiente.

Desplazamiento de la información

Todas las opciones tienen en el lado derecho de la ventana un área donde se encuentra la información básica de cada una de las opciones. Cada opción mostrará solo un número determinado de datos en primera instancia. Para poder desplazarse y ver los siguientes datos se deben utilizar los botones que se detallan en la Tabla B.1:

Tabla B.1: Botones de desplazamiento de la información

Botón	Descripción
««	Primera página de la tabla de información
»	Siguiente página de la tabla de información
«	Anterior página de la tabla de información
»»	Última página de la tabla de información

AMBIENTES DE BASYS

La plataforma Web tiene como finalidad la administración de usuarios de la aplicación, jugadores, equipos, campeonatos y estadísticas. En esta sección del documento, se aprenderá como se debe ingresar a la aplicación y se mostrará la pantalla principal que tiene el Administrador.

Al abrir una ventana de Internet Explorer se debe escribir en la barra de direcciones:

http:// <nombre del servidor o dirección IP del servidor>/Basys/Hipertexto/

Por ejemplo como se muestra en la Figura B.3:



Figura B.3: Dirección Web del sistema

A continuación se abrirá una ventana que presenta al sistema Basys, la cual se presenta en la Figura B.4 a seguir:



Figura B.4: Pantalla de presentación de Basys

La segunda ventana que se abre muestra una pantalla en la que se da una información inicial acerca de las bondades que ofrece el sistema y un cuadro en el que se puede ingresar el alias y contraseña del usuario (Figura B.5). Cabe recalcar que todos los usuarios que estén conectados en la intranet tendrán acceso a esta pantalla. Sin embargo, solamente los usuarios activos creados por el administrador que tengan un alias de usuario y contraseña podrán acceder a la pantalla principal del sistema Basys.



Figura B.6: Pantalla de ingreso al sistema

Si el usuario ingresó de forma errónea ya sea el alias de usuario o contraseña, el sistema le presentará el siguiente mensaje: *El usuario o la contraseña no son válidos* (Figura B.7). En tal virtud el usuario deberá volver a intentarlo.



Figura B.7: Usuario o contraseña inválidos

Si se ha ingresado de forma correcta los datos anteriormente mencionados, el usuario ingresará a la pantalla principal del sistema Basys como se muestra en la Figura B.8 en la cual se presentan 5 secciones:



Figura B.8: Pantalla Principal de Basys

A continuación se muestra la Tabla B.2 que describe cada una de las secciones:

Tabla B.2: Secciones de la pantalla principal

Sección	Descripción
1	Menú básico: este menú le permite al usuario ingresar de forma rápida a la pantalla principal, abrir el presente documento (Ayuda) o salir de la aplicación.
2	Menú del sistema: en esta sección de la pantalla principal se muestran seis botones para ingresar a los diferentes módulos del sistema: Usuario, Jugador, Equipo, Campeonato, Partido y Reportes.
3	Cambio de contraseña: aquí se presenta un botón que permite cambiar la contraseña de usuario que ingreso en la aplicación. Esta opción se explicara de manera más detallada a continuación de la tabla.

4	Área de trabajo: aquí se presentara la información que maneja el sistema de acuerdo a la opción seleccionada.
5	Desarrolladores: en la última sección se muestran los nombres de las personas que desarrollaron el presente sistema.

Salir de la aplicación

Si el usuario ya no necesita seguir trabajando en el sistema Basys debe presionar el icono  *Salir*. Es necesario mencionar que si el usuario se encuentra inactivo por mucho tiempo sin finalizar su sesión, el sistema Basys expirará su sesión y cuando el usuario desee realizar alguna acción deberá volver a ingresar su alias de usuario y contraseña.

Cambio de contraseña

Es recomendable que el usuario cambie su contraseña cada cierto período de tiempo por motivos de seguridad, para lo cual deberá estar en la pantalla principal de la aplicación y presionar en el botón que se encuentra en la sección 3. Se desplegará una ventana en la que el usuario actual debe ingresar su contraseña anterior y la nueva contraseña. Una vez verificados los datos debe presionar el botón *Aceptar* para que se realice el cambio respectivo. Si el usuario ya no desea cambiar su contraseña, debe presionar el botón *Cancelar*. Esta ventana se muestra a continuación en la Figura B.9:



Cambiar contraseña

Administrador

Contraseña Anterior:

Contraseña Nueva:

Repita Contraseña:

Figura B.9: Cambio de contraseña

Si la contraseña actual es errónea, no se podrá cambiar la contraseña y el sistema presentara el siguiente mensaje (Figura B.10) para lo cual se debe presionar el botón *Aceptar* y volver a intentarlo. Si el usuario olvido su contraseña, debe acudir al administrador del sistema para que él realice el cambio respectivo.



Figura B.10: Contraseña anterior inválida

Si las nuevas contraseñas ingresadas no coinciden, el sistema presentará el mensaje de la Figura B.11, se debe presionar el botón *Aceptar* y volver a intentarlo.

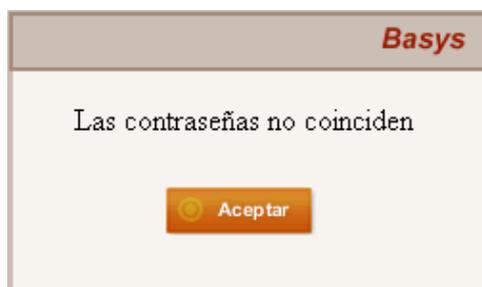


Figura B.11: Nuevas contraseñas no coinciden

NAVEGANDO POR BASYS

Opción Usuario

La opción usuario le permite al administrador del sistema administrar los usuarios con sus respectivos perfiles, que pueden ingresar a la aplicación. Cuando se ingresa en esta opción se puede crear, consultar, modificar o eliminar a los usuarios de la aplicación. Además permite asignar el perfil que va a tener un usuario para determinar a que opciones puede tener acceso.

Para ingresar en esta opción (Figura B.12) debe presionar sobre el botón *Usuario* del *Menú del Sistema*.



Figura B.13: Administración de Usuarios

A continuación se muestra la Tabla B.3 que describe cada una de las secciones:

Tabla B.3: Secciones de la pantalla Usuarios

Sección	Descripción
1	Administración de Usuarios: esta sección presenta todos los usuarios registrados en la aplicación. Y dispone de un botón para seleccionar un usuario para su visualización, modificación, eliminación o para asignarle su perfil.
2	Información del usuario: en esta sección se presentan todos los campos para ingresar o visualizar la información de un usuario.
3	Botones de control: esta sección presenta diferentes botones para realizar las diferentes acciones encargadas de manipular la información de los usuarios.

Ingresar un nuevo Usuario

Para crear un nuevo usuario se deben seguir los siguientes pasos:

1. Presionar el botón *Nuevo* que se encuentra en la sección 3 de la pantalla de usuarios. Al realizar esto los campos de la sección 2 se habilitarán.

2. Ingresar la información del usuario: nombre, alias, cédula, teléfono casa, teléfono celular, dirección, mail, contraseña y repetir la contraseña. Los campos nombre, alias, cédula, contraseña y repetir la contraseña son obligatorios.
3. Si no se ingresan todos los campos obligatorios y se presiona el botón *Guardar*, en la parte inferior izquierda de la sección 1 aparecerán mensajes indicando los campos que no han sido ingresados, como se muestra en la Figura B.14:

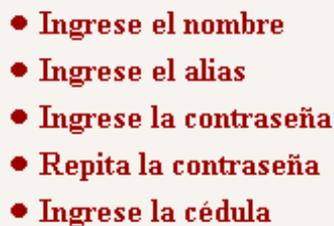
- 
- **Ingrese el nombre**
 - **Ingrese el alias**
 - **Ingrese la contraseña**
 - **Repita la contraseña**
 - **Ingrese la cédula**

Figura B.14: Campos obligatorios no ingresados

4. Si la cédula ingresada es incorrecta el sistema mostrará el mensaje “*La cédula es incorrecta*” en la parte inferior de la sección 2 de la pantalla de usuarios.
5. Una vez verificados los datos del usuario ingresados se procede a presionar el botón *Guardar* que se encuentra en la sección 3 de la pantalla usuarios.
6. Si no desea guardar la información del nuevo usuario debe presionar el botón *Cancelar* de la sección Botones de Control.

Seleccionar un Usuario

En la sección 1 de la pantalla de usuarios se presenta información básica de los usuarios: nombre (alias) y mail. En la parte izquierda de dicha información se

presenta el icono  que permite seleccionar un usuario para visualizar su información en la sección 2.

Modificar un Usuario

Para modificar la información de un usuario se debe seleccionar primeramente el usuario a ser modificado, seguidamente se realizan los cambios respectivos en la sección 2 y se presiona el botón *Modificar* de la sección 3.

Eliminar un Usuario

Para eliminar un usuario se debe seleccionar primeramente el usuario a ser eliminado y presionar en botón *Eliminar* de la sección 3 de la pantalla usuarios. El sistema presentará el mensaje de la Figura B.15 para verificar que se encuentre seguro de realizar dicha acción. Si se encuentra seguro debe presionar el botón *Aceptar*, caso contrario debe presionar el botón *Cancelar*.

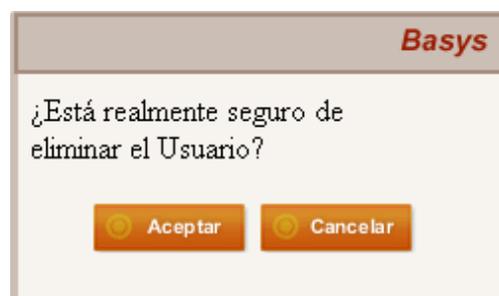


Figura B.15: Ventana de verificación de la eliminación del usuario

Asignar el Perfil a un Usuario

El nivel de seguridad que permite la aplicación se encuentra al nivel de las opciones que presenta el Menú del Sistema. Para asignar el perfil de un usuario se deben seguir los siguientes pasos:

1. Seleccionar el usuario al que se desea asignar o modificar un perfil en la sección 1.

2. Presionar el botón *Perfil* que se encuentra en la sección 3 de la pantalla de usuarios.
3. A continuación se presenta una ventana (Figura B.16) para administrar el perfil que va a tener el usuario.



Figura B.16: Ventana de administración del Perfil del usuario

A continuación se muestra la Tabla B.4 que describe las secciones:

Tabla B.4: Secciones de la ventana Perfil

Sección	Descripción
1	Perfiles disponibles: muestra todas las opciones a las que puede tener acceso el usuario.
2	Botones de administración de perfiles: permite agregar o quitar las opciones a las que puede tener acceso el usuario.
3	Perfiles asignados: despliega las opciones a las que puede acceder el usuario
4	Botones de control: acepta o cancela la operación.

4. Para agregar una opción, se debe seleccionar una de las opciones que se encuentran en la sección 1 y presionar el botón  de la sección 2. Esta nueva opción se agregará en la sección 3 de la ventana de Perfil. En cambio para quitar una opción, se debe hacer de forma inversa, es decir, seleccionar una de las opciones que se encuentran en la sección 3 y

- presionar el botón  de la sección 2. Esta opción se agregará en la sección 1 de la ventana de Perfil.
5. Una vez agregadas o quitadas la opciones a las que tiene acceso el usuario se procede a presionar el botón *Aceptar* del la sección 4 de la ventana Perfil.
 6. Si no desea realizar ningún cambio, debe presionar el botón *Cancelar* de la sección 4 de la ventana Perfil.

Opción Jugador

La opción jugador le permite al usuario administrar la información de todos los jugadores que van a participar ya sea en competencias internas como externas. Para ingresar en esta opción debe presionar el botón *Jugador* del *Menú del Sistema*. Si el usuario activo no posee el perfil que le permita tener acceso a esta opción, se mostrará el mensaje correspondiente en la sección 3 de la Pantalla Principal, como se muestra en la Figura B.17:



Figura B.17: No tiene permisos para ingresar a la opción Jugador

En el caso que el usuario activo si posea acceso a la opción Jugador se presentará la pantalla que se muestra en la Figura B.18 a seguir:



Figura B.18: Administración de Jugadores

A continuación se muestra la Tabla B.5 que describe cada una de las secciones:

Tabla B.5: Secciones de la pantalla Jugador

Sección	Descripción
1	Administración de Jugadores: esta sección presenta todos los jugadores registrados en la aplicación. Y dispone de un botón para seleccionar un jugador para su visualización, modificación o eliminación. En la parte superior la aplicación muestra la foto del jugador si la tuviere. Debajo de ella se encuentra un buscador.
2	Información del jugador: en esta sección se presentan todos los campos para ingresar o visualizar la información de un jugador.
3	Botones de control: esta sección presenta diferentes botones para realizar las diferentes acciones encargadas de manipular la información de los jugadores.

Ingresar un nuevo Jugador

Para crear un nuevo jugador se deben seguir los siguientes pasos:

1. Presionar el botón *Nuevo* que se encuentra en la sección 3 de la pantalla de jugadores. Al realizar esto los campos de la sección 2 se habilitarán.

2. Ingresar la información del jugador: nombre, apellido, cédula, teléfono casa, teléfono celular, dirección, mail, sexo, país, provincia, y foto. Los campos nombre, apellido, cédula y sexo son obligatorios.
3. Si no se ingresan todos los campos obligatorios y se presiona el botón *Guardar*, en la parte inferior izquierda de la sección 1 aparecerán mensajes indicando los campos que no han sido ingresados, como se muestra en la Figura B.19:

- **Ingrese el nombre**
- **Ingrese el apellido**
- **Ingrese la cédula**
- **Seleccione el sexo**

Figura B.19: Campos obligatorios no ingresados

4. Si la cédula ingresada es incorrecta el sistema mostrará el mensaje “*La cédula es incorrecta*” en la parte inferior de la sección 2 de la pantalla de usuarios.
5. Una vez verificados los datos del jugador ingresados se procede a presionar el botón *Guardar* que se encuentra en la sección 3 de la pantalla jugadores.
6. Si no desea guardar la información del nuevo jugador debe presionar el botón *Cancelar* de la sección Botones de Control.

Buscador de Jugadores

En la sección 1 se encuentra el campo *Apellido* que permite filtrar los jugadores cuyo apellido coincida total o parcialmente con el dato ingresado. Para lo cual se debe ingresar el parámetro de búsqueda y presionar el botón *Buscar*. 🔍

Seleccionar un Jugador

En la sección 1 de la pantalla de jugadores se presenta información básica de los jugadores: apellido y nombre. En la parte izquierda de dicha información se presenta el icono  que permite seleccionar un jugador para visualizar su información en la sección 2. Además en la parte superior de la sección 1 se presentará la foto del jugador si la tuviere.

Modificar un Jugador

Para modificar la información de un jugador se debe seleccionar primeramente el jugador a ser modificado, seguidamente se realizan los cambios respectivos en la sección 2 y se presiona el botón *Modificar* de la sección 3.

Eliminar un Jugador

Para eliminar un jugador se debe seleccionar primeramente el jugador a ser eliminado y presionar en botón *Eliminar* de la sección 3 de la pantalla jugadores. El sistema presentara el mensaje de la Figura B.20 para verificar que se encuentre seguro de realizar dicha acción. Si se encuentra seguro debe presionar el botón *Aceptar*, caso contrario debe presionar el botón *Cancelar*.

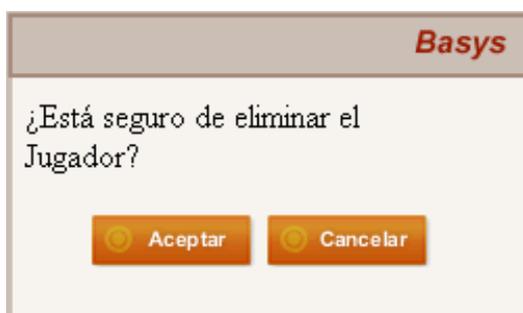


Figura B.20: Ventana de verificación de la eliminación del jugador

Si el jugador ya se encuentra registrado en un partido, el sistema mostrará el mensaje "Error en la eliminación" en la parte inferior de la sección 2.

Opción Equipo

La opción equipo le permite al usuario administrar la información de todos los equipos que van a participar ya sea en competencias internas como externas. Para ingresar en esta opción debe presionar el botón *Equipo* del *Menú del Sistema*. Si el usuario activo no posee el perfil que le permita tener acceso a esta opción, se mostrará el mensaje correspondiente en la sección 3 de la Pantalla Principal, como se muestra en la Figura B.21:



Figura B.21: No tiene permisos para ingresar a la opción Equipo

En el caso que el usuario activo si posea acceso a la opción Equipo se presentará la pantalla que se muestra en la Figura B.22 a seguir:

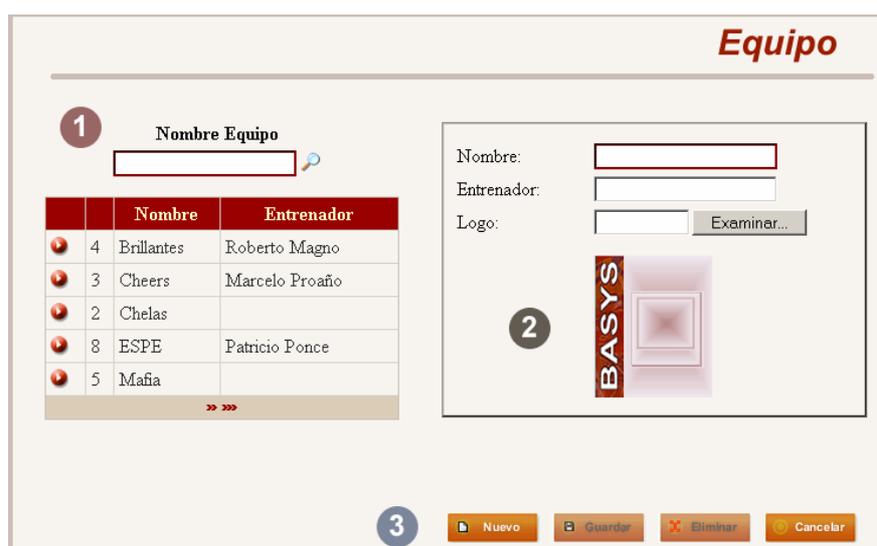


Figura B.22: Administración de Equipos

A continuación se muestra la Tabla B.6 que describe cada una de las secciones:

Tabla B.6: Secciones de la pantalla Equipo

Sección	Descripción
1	Administración de Equipos: esta sección presenta todos los equipos registrados en la aplicación. Y dispone de un botón para seleccionar un equipo para su visualización, modificación o eliminación. Dispone también de un buscador.
2	Información del equipo: en esta sección se presentan todos los campos para ingresar o visualizar la información de un equipo. En la parte inferior la aplicación muestra la foto del equipo si la tuviere
3	Botones de control: esta sección presenta diferentes botones para realizar las diferentes acciones encargadas de manipular la información de los equipos.

Ingresar un nuevo Equipo

Para crear un nuevo equipo se deben seguir los siguientes pasos:

1. Presionar el botón *Nuevo* que se encuentra en la sección 3 de la pantalla de equipos. Al realizar esto los campos de la sección 2 se habilitarán.
2. Ingresar la información del equipo: nombre, entrenador y logo. El campo nombre es obligatorio.
3. Si no se ingresa el nombre del equipo y se presiona el botón *Guardar*, en la parte inferior izquierda de la sección 1 aparecerá el siguiente mensaje “*Ingrese el nombre*”.
4. Una vez verificados los datos del equipo ingresados se procede a presionar el botón *Guardar* que se encuentra en la sección 3 de la pantalla equipos.
5. Si no desea guardar la información del nuevo equipo debe presionar el botón *Cancelar* de la sección Botones de Control.

Buscador de Equipos

En la sección 1 se encuentra el campo *Nombre Equipo* que permite filtrar los equipos cuyo nombre coincida total o parcialmente con el dato ingresado. Para lo cual se debe ingresar el parámetro de búsqueda y presionar el botón *Buscar*. 🔍

Seleccionar un Equipo

En la sección 1 de la pantalla de equipos se presenta información básica de los equipos: nombre y entrenador. En la parte izquierda de dicha información se presenta el icono 📌 que permite seleccionar un equipo para visualizar su información en la sección 2. Además en la parte inferior de la sección 2 se presentará el logo del equipo si lo tuviere.

Modificar un Equipo

Para modificar la información de un equipo se debe seleccionar primeramente el equipo a ser modificado, seguidamente se realizan los cambios respectivos en la sección 2 y se presiona el botón *Modificar* de la sección 3.

Eliminar un Equipo

Para eliminar un equipo se debe seleccionar primeramente el equipo a ser eliminado y presionar en botón *Eliminar* de la sección 3 de la pantalla equipos. El sistema presentara el mensaje de la Figura B.23 para verificar que se encuentre seguro de realizar dicha acción. Si se encuentra seguro debe presionar el botón *Aceptar*, caso contrario debe presionar el botón *Cancelar*.

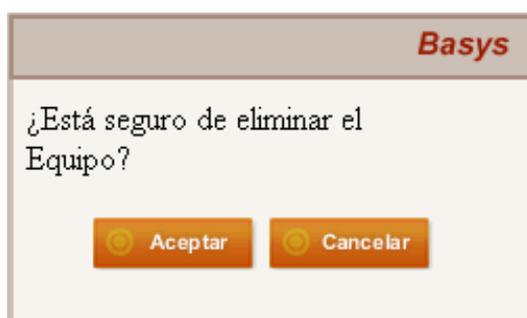


Figura B.23: Ventana de verificación de la eliminación del equipo

Si el equipo ya se encuentra registrado en un partido, el sistema mostrará el mensaje “Error en la eliminación” en la parte inferior de la sección 2 de la pantalla de equipos.

Opción Campeonato

La opción campeonato le permite al usuario administrar la información de todas las competencias sean estas internas o externas. Para ingresar en esta opción debe presionar el botón *Campeonato* del *Menú del Sistema*. Si el usuario activo no posee el perfil que le permita tener acceso a esta opción, se mostrará el mensaje correspondiente en la sección 3 de la Pantalla Principal, como se muestra en la Figura B.24:



Figura B.24: No tiene permisos para ingresar a la opción Campeonato

En el caso que el usuario activo si posea acceso a la opción Campeonato se presentará la pantalla que se muestra en la Figura B.25 a seguir:

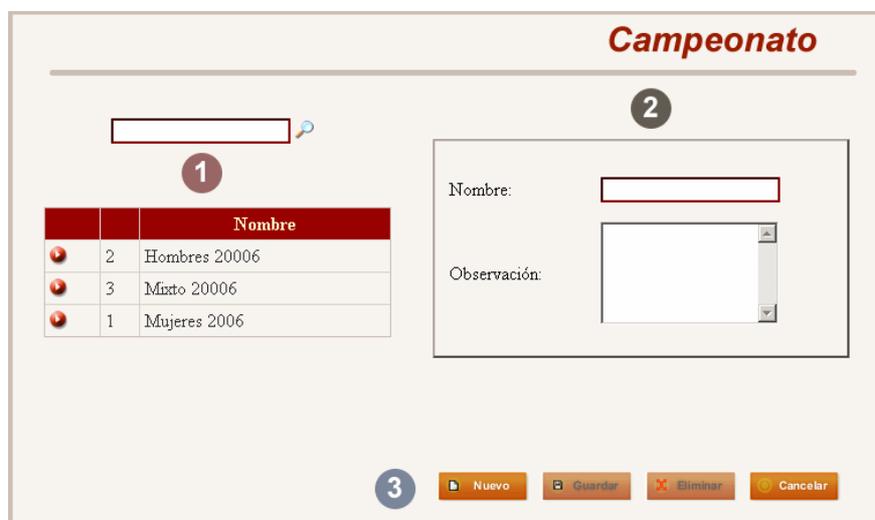


Figura B.25: Administración de Campeonatos

A continuación se muestra la Tabla B.7 que describe cada una de las secciones:

Tabla B.7: Secciones de la pantalla Campeonato

Sección	Descripción
1	Administración de Campeonatos: esta sección presenta todos los campeonatos registrados en la aplicación. Y dispone de un botón para seleccionar un campeonato para su visualización, modificación o eliminación. Dispone también de un buscador.
2	Información del campeonato: en esta sección se presentan todos los campos para ingresar o visualizar la información de un campeonato.
3	Botones de control: esta sección presenta diferentes botones para realizar las diferentes acciones encargadas de manipular la información de los campeonatos.

Ingresar un nuevo Campeonato

Para crear un nuevo campeonato se deben seguir los siguientes pasos:

6. Presionar el botón *Nuevo* que se encuentra en la sección 3 de la pantalla de campeonatos. Al realizar esto los campos de la sección 2 se habilitarán.
7. Ingresar la información del campeonato: nombre y observación. El campo nombre es obligatorio.
8. Si no se ingresa el nombre del campeonato y se presiona el botón *Guardar*, en la parte inferior izquierda de la sección 1 aparecerá el siguiente mensaje "*Ingrese el nombre*".
9. Una vez verificados los datos del campeonato ingresados se procede a presionar el botón *Guardar* que se encuentra en la sección 3 de la pantalla campeonatos.
10. Si no desea guardar la información del nuevo campeonato debe presionar el botón *Cancelar* de la sección Botones de Control.

Buscador de Campeonatos

En la sección 1 se encuentra el campo *Nombre Campeonato* que permite filtrar los campeonatos cuyo nombre coincida total o parcialmente con el dato ingresado. Para lo cual se debe ingresar el parámetro de búsqueda y presionar el botón *Buscar*. 🔍

Seleccionar un Campeonato

En la sección 1 de la pantalla de campeonatos se presentan todos los campeonatos registrados. En la parte izquierda de dicha información se presenta el icono 📌 que permite seleccionar un campeonato para visualizar su información en la sección 2.

Modificar un Campeonato

Para modificar la información de un campeonato se debe seleccionar el campeonato a ser modificado, seguidamente se realizan los cambios respectivos en la sección 2 y se presiona el botón *Modificar* de la sección Botones de Control.

Eliminar un Campeonato

Para eliminar un campeonato se debe seleccionar primeramente el campeonato a ser eliminado y presionar en botón *Eliminar* de la sección 3 de la pantalla campeonatos. El sistema presentara el mensaje de la Figura B.26 para verificar que se encuentre seguro de realizar dicha acción. Si se encuentra seguro debe presionar el botón *Aceptar*, caso contrario debe presionar el botón *Cancelar*.

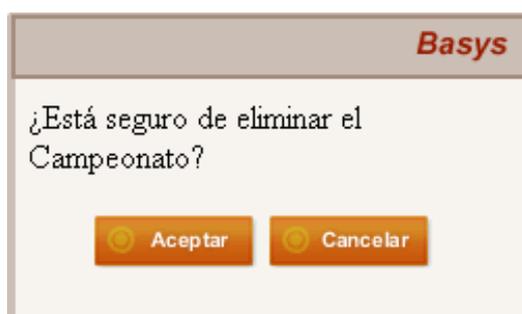


Figura B.26: Ventana de verificación de la eliminación del campeonato

Si el campeonato ya se encuentra registrado en un partido, el sistema mostrará el mensaje “Error en la eliminación” en la parte inferior de la sección 2.

Opción Partido

La opción partido le permite al usuario administrar la información de todos los partidos que se registren en el sistema. Para ingresar en esta opción debe presionar el botón *Partido* del *Menú del Sistema*. Si el usuario activo no posee el perfil que le permita tener acceso a esta opción se mostrará el mensaje correspondiente en la sección 3 de la Pantalla Principal, como se muestra en la Figura B.27:



Figura B.27: No tiene permisos para ingresar a la opción Partido

En el caso que el usuario activo si posea acceso a la opción Partido se presentará la pantalla que se muestra en la Figura B.28 a seguir:

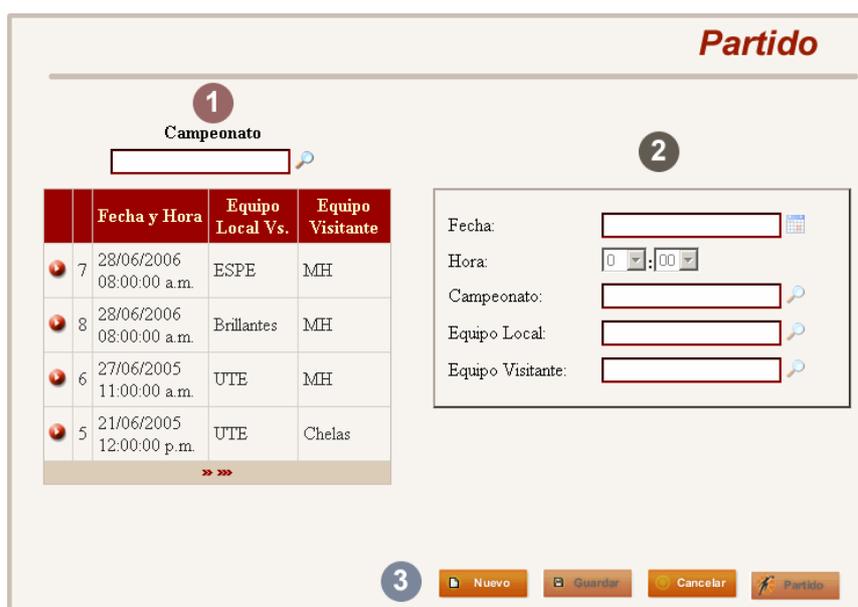


Figura B.28: Administración de Partidos

A continuación se muestra la Tabla B.8 que describe cada una de las secciones:

Tabla B.8: Secciones de la pantalla Partido

Sección	Descripción
1	Administración de Partidos: esta sección presenta todos los partidos registrados en la aplicación. Y dispone de un botón para seleccionar un partido para su visualización, modificación o para registrar la información del mismo. Dispone también de un buscador.
2	Información del partido: en esta sección se presentan todos los campos para ingresar o visualizar la información de un partido. En la parte inferior la aplicación muestra la foto del partido si la tuviere
3	Botones de control: esta sección presenta diferentes botones para realizar las diferentes acciones encargadas de manipular la información de los partidos.

Ingresar un nuevo Partido

Para crear un nuevo partido se deben seguir los siguientes pasos:

1. Presionar el botón *Nuevo* que se encuentra en la sección 3 de la pantalla de partidos. Al realizar esto los campos de la sección 2 se habilitarán.
2. Ingresar la información del partido: fecha, hora, campeonato, equipo local y equipo visitante. Todos los campos son obligatorios.
3. Si no se ingresan los campos obligatorios y se presiona el botón *Guardar*, en la parte inferior izquierda de la sección 1 aparecerán mensajes indicando los campos que no han sido ingresados, como se muestra en la Figura B.29:

- **Ingrese el campeonato**
- **Ingrese el equipo local**
- **Ingrese el equipo visitante**

Figura B.29: Campos obligatorios no ingresados

- a. Para ingresar los campos Campeonato, Equipo Local y Equipo Visitante la aplicación dispone de un buscador de los campeonatos, o equipos que se encuentran registrados en la aplicación. Para lo cual se debe seleccionar el botón *Buscar* y el sistema presentará la ventana que se muestra en la Figura B.30:



Figura B.30: Ventana del Buscador

A continuación se muestra la Tabla B.9 que describe las secciones:

Tabla B.9: Secciones de la ventana del Buscador

Sección	Descripción
1	Búsqueda: está sección dispone de un campo para realizar la búsqueda y un botón para ejecutarla.
2	Información: esta sección presenta la información básica de los campeonatos o equipos que se desean buscar.
3	Botones de control: la sección presenta dos botones que permiten seleccionar o no, un campeonato o equipo.

- b. Para buscar ya sean campeonatos o equipos el usuario debe ingresar el nombre del campeonato o equipo en el campo *Nombre* y presionar el botón *Buscar*.
- c. En la sección 2 se presentarán todos los datos (si no se ha realizado una búsqueda) o las coincidencias con el dato ingresado

en el campo *Nombre*. El usuario debe seleccionar un campeonato o equipo utilizando el botón  y presionar el botón *Aceptar* para que el dato seleccionado se muestre en el campo respectivo de la sección 2 de la pantalla partido. Si el usuario no desea seleccionar un campeonato o equipo debe seleccionar el botón *Cancelar* de la sección 3 de la ventana del Buscador.

4. La pantalla *Partido* le permite al usuario ingresar directamente un campeonato o equipo que no se encuentre registrado en la aplicación. En el caso de que el nombre del campeonato o equipo ingresado sea igual a uno ya registrado el sistema no lo volverá a ingresar.
5. Una vez verificados los datos del partido ingresados se procede a presionar el botón *Guardar* que se encuentra en la sección 3 de la pantalla partidos.
6. Si no desea guardar la información del nuevo partido debe presionar el botón *Cancelar* de la sección Botones de Control.

Buscador de Partidos

En la sección 1 se encuentra el campo *Campeonato* que permite filtrar los partidos que se jugaron en un campeonato cuyo nombre coincida total o parcialmente con el dato ingresado. Para lo cual se debe ingresar el parámetro de búsqueda y presionar el botón *Buscar*. 

Seleccionar un Partido

En la sección 1 de la pantalla de partidos se presenta información básica de los partidos: fecha, hora, equipo local y equipo visitante. En la parte izquierda de dicha información se presenta el icono  que permite seleccionar un partido para visualizar su información en la sección 2.

Modificar un Partido

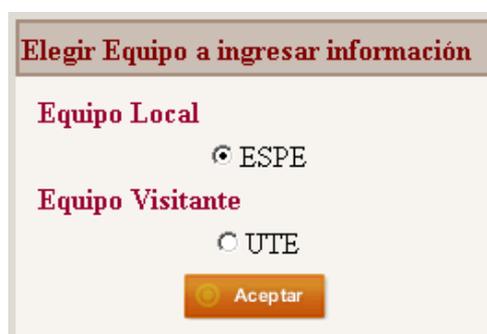
Para modificar la información de un partido se debe seleccionar primeramente el partido a ser modificado, seguidamente se realizan los cambios respectivos en la sección 2 y se presiona el botón *Modificar* de la sección 3.

Opción Información del Partido

Esta opción permite almacenar los tiros al aro (convertidos o fallados), rebotes (defensivos y ofensivos), asistencias, pérdidas de balón, faltas (recibidas y cometidas), y el total de puntos por cada jugador que participe en un determinado partido. Una vez terminado el partido, provee de un proceso para calcular diferentes estadísticas de cada uno de los jugadores y del equipo.

Para ingresar en esta opción, debe seleccionar el partido que desea ingresar la información, de la sección 1 de la ventana de administración de partidos y presionar el botón *Partido* que se encuentra en la sección 3 de la ventana de administración de partidos.

Seguidamente el sistema le presentará una ventana que le permite seleccionar el equipo del cual desea ingresar o modificar su información, para lo cual se debe dar clic en uno de los botones de radio y presionar el botón *Aceptar*, como se muestra en la Figura B.31 a seguir:



The image shows a dialog box with a title bar that reads "Elegir Equipo a ingresar información". Below the title bar, there are two sections. The first section is labeled "Equipo Local" and contains a radio button that is selected (indicated by a small black dot) next to the text "ESPE". The second section is labeled "Equipo Visitante" and contains a radio button that is not selected (indicated by an empty circle) next to the text "UTE". At the bottom of the dialog box, there is an orange button with a white arrow pointing right and the text "Aceptar".

Figura B.31: Seleccionar un equipo

Si se va a ingresar la información de un nuevo partido entonces se deberán registrar los jugadores en iniciar el partido, para poder ingresar los datos respectivos de cada jugador. En el caso de que el partido ya posea datos, el sistema cargará los datos del partido e indicará el periodo en el se encuentra el partido. En cualquiera de los dos casos el sistema presentara la ventana que se muestra en la Figura B.32 como se muestra a continuación:

Información del Partido

1 Período: **1**

Equipo Local: **ESPE**

Fecha: **6/7/2006**

Hora: **9:00**

Campeonato: **ESPE 2006**

0

Equipo Visitante: **UTE**

Nro.	JUGADOR	TIROS AL ARO						REBOTES		A S I S T.	PÉRDIDA DE BALÓN	V I O L A C.	FALTAS		P U N T O S
		Corta		Larga		Tiro Libre		Of.	Def.				Re.	Co.	
		C	F	C	F	C	F								
	<input type="text"/>	0	0	0	0	0	0	0	0	0	0	0	0	0	
	<input type="text"/>	0	0	0	0	0	0	0	0	0	0	0	0	0	
	<input type="text" value="2"/>	0	0	0	0	0	0	0	0	0	0	0	0	0	
	<input type="text"/>	0	0	0	0	0	0	0	0	0	0	0	0	0	
	<input type="text"/>	0	0	0	0	0	0	0	0	0	0	0	0	0	

5

Figura B.32: Venta de Información del Partido

A continuación se muestra la Tabla B.10 que describe las secciones:

Tabla B.10: Secciones de la pantalla Información del Partido

Sección	Descripción
1	Información general del partido: esta sección presenta todos los datos del partido como: Periodo, Fecha, Hora, Campeonato, Equipo Local, Equipo Visitante y Total de los puntos de cada equipo.
2	Información de los jugadores: en esta sección se maneja los jugadores que se encuentran en el partido. La información que se presenta por cada jugador es: Número, Foto y Nombre completo.
3	Botones de control de la información del partido: esta sección dispone de varios botones que permite registrar cada una de las actividades que realicen los jugadores que se encuentren en el juego.
4	Jugadores Suplentes: esta sección consta de los jugadores suplentes que pueden ingresar al partido en cualquier momento.
5	Botones de control: esta sección presenta varios botones para realizar las diferentes acciones encargadas de manejar la información del partido.

Ingreso de Jugadores e Inicio del Partido

Para definir los jugadores que van a ingresar en primera instancia en un partido debemos realizar lo siguiente:

1. En la sección 2 de la ventana de Información del partido, se deben ingresar los cinco jugadores principales del partido, para lo cual se encuentra un botón  *Buscar*, que despliega un buscador de jugadores, como se muestra en la Figura B.33 a continuación:



Figura B.33: Ventana Buscador de Jugadores

A continuación se muestra la Tabla B.11 que describe las secciones:

Tabla B.11: Secciones de la ventana Buscador de Jugadores

Sección	Descripción
1	Parámetros del buscador: esta sección dispone de dos parámetros de búsqueda: Sexo y / o Apellido. Y un botón para ejecutar la búsqueda.
2	Jugadores disponibles: esta sección presenta la información básica de los jugadores que no se encuentran tomados en cuenta para el partido en curso.
3	Número del jugador: esta sección tiene un campo en el cual se debe ingresar el número del jugador que se ha seleccionado en la sección 2.
4	Botones de control: la sección presenta dos botones que permiten seleccionar o no un jugador.

- a. Al abrirse la ventana se despliegan todos los jugadores que no se encuentran en el partido en curso. Si se desea buscar un jugador la sección 1 de la ventana dispone de dos parámetros de búsqueda, el sexo o el Apellido. El usuario debe seleccionar un sexo o ingresar el

apellido o una parte del apellido de un jugador en el campo *Apellido* y presionar el botón *Buscar*. 🏠

- b. En la sección 2 se visualizarán todos los jugadores que cumplan con las condiciones ingresadas, el usuario deberá seleccionar un jugador presionando el botón *Seleccionar* 📌 que se encuentra en la parte izquierda de la misma sección.
- c. Luego debe ingresar el número del jugador que seleccionó y presionar el botón *Aceptar* que se encuentra en la sección 4.
- d. Si el jugador no desea seleccionar un jugador debe seleccionar el botón *Cancelar* que se encuentra en la sección 4.
- e. Puesto que el número del jugador es obligatorio, si no se lo ingresa el sistema presentará la ventana que se muestra en la Figura B.34:

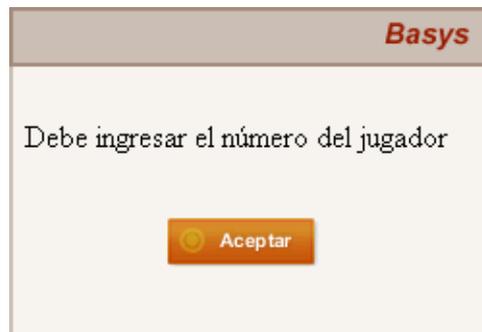


Figura B.34: Número del Jugador

- f. Si el número del jugador ingresado no se encuentra disponible, el sistema presentará el siguiente mensaje (Figura B.35):

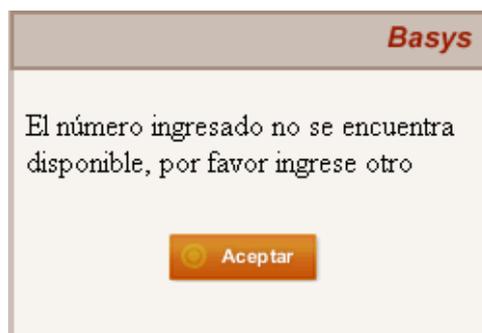


Figura B.35: Número del Jugador no disponible

2. Luego se deben ingresar los jugadores suplentes que se encuentran en la sección 4 de la ventana de información del partido, de la misma forma como se ingresaron los jugadores principales.
3. Una vez ingresados todos los jugadores que pueden participar en el partido se debe seleccionar el botón *Iniciar Partido*  que se encuentra en la sección 5 de la ventana de información del partido. Si no se definieron los cinco jugadores principales, no se podrá iniciar el partido y el sistema emitirá el mensaje que se encuentra en la Figura B.36 a seguir:

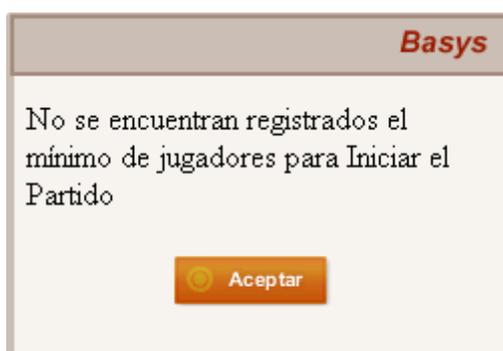


Figura B.36: No se puede iniciar el partido

Registro de los datos del partido

Un partido de basketball consta de 4 periodos, el sistema permite registrar los datos de las acciones que realizan los jugadores en cada uno de sus periodos. Para registrar un suceso del partido de un jugador, el usuario debe ubicarse en la fila del jugador y en la columna de la acción que desea registrar y presionar sobre el botón indicado.

Cada fila de un jugador dispone de botones que incrementan el valor de la acción que se suscitó en el partido, como se muestra en la Figura B.37. Además

se dispone de un botón pequeño de color amarillo que permite disminuir el valor de la acción en el caso de que el usuario haya cometido algún error.



Figura B.37: Esquema por jugador

Cuando el jugador convierta, ya sean en tiros de corta o larga distancia o tiros libres, los puntos del jugador y el puntaje del equipo se incrementarán automáticamente.

Cambio de un jugador

Para realizar un cambio de jugadores, el usuario debe seleccionar en el círculo blanco (botón de radio) que se encuentra en el lado izquierdo del nombre del jugador que desea realizar el cambio, y presionar en la foto del jugador suplente que va a ingresar al partido. Los datos del nuevo jugador se cargarán automáticamente.

Estados del partido

Cuando finalice un cuarto, el usuario deberá seleccionar el botón *Finalizar Cuarto*  que se encuentra en la sección 5 de la ventana de información del partido. Cuando se reanude el partido, es decir comience el siguiente cuarto, el usuario deberá seleccionar el botón *Iniciar Cuarto*  que se encuentra en la sección 5 de la ventana de información del partido. El número del periodo que se encuentra en la sección 1 de la ventana de información del partido se actualizará automáticamente.

Información del Partido															
Periodo:				68		Fecha: 6/7/2006		0							
4		Equipo Local: ESPE		Campeonato: ESPE 2006		Equipo Visitante: UTE									
Nro.	JUGADOR	TIROS AL ARO						REBOTES		A S I S T.	PÉRDIDA DE BALÓN	V I O L A C.	FALTAS		P U N T O S
		Corta		Larga		Tiro Libre		Of.	Def.				Re.	Co.	
		C	F	C	F	C	F								
3	 Robayo Francisco	7	0	4	2	2	0	2	2	0	2	1	2	1	28
5	 Goyes Henry	0	3	3	1	4	0	2	3	3	2	0	1	1	13
7	 Orellana Rene	2	2	3	0	3	2	3	1	2	4	0	0	2	16
4	 Puertas Eduardo	2	0	1	2	0	0	1	3	0	1	1	2	0	7
1	 Hurtado Danny	2	0	0	0	0	0	0	0	0	0	0	0	0	4

Figura B.38: Ejemplo de la información de un partido

Calcular las estadísticas

Una vez que finaliza el segundo o el cuarto periodo del partido, la ventana se deshabilita y el usuario debe seleccionar el botón *Calcular Estadística*  que se encuentra en la sección 1 de la ventana de información del partido. Esta acción se suele demorar un momento y cuando el sistema finaliza de calcular las estadísticas del partido emite la siguiente ventana que se muestra en la Figura B.39, la cual emite un resumen del cálculo de las estadísticas.

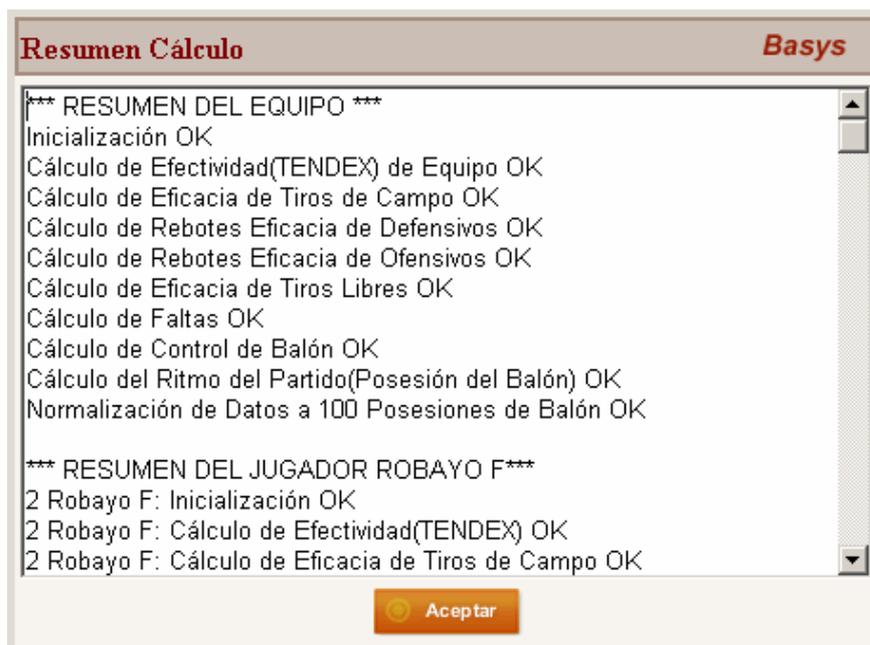


Figura B.39: Resumen del cálculo de las estadísticas

Una vez que el usuario haya revisado que el cálculo se ha realizado sin ningún problema debe seleccionar el botón *Aceptar*. Si ha existido algún error, el usuario debe reportar el error al Administrador del sistema para su verificación.

Finalmente se presenta el botón *Consultar Partido* que permite ingresar de forma rápida al reporte del partido. La creación de los reportes se lo verá a continuación.

Opción Reportes

La opción reportes le permite al usuario emitir diferentes reportes referentes al partido o en base a las estadísticas que se calculan en cada partido. Para ingresar en esta opción debe presionar el botón *Reportes* del *Menú del Sistema*.

Si el usuario activo no posee el perfil que le permita tener acceso a esta opción se mostrará el mensaje correspondiente, en la sección 3 de la Pantalla Principal, como se muestra en la Figura B.40:



Figura B.40: No tiene permisos para ingresar a la opción Reportes

En el caso que el usuario activo si posea acceso a la opción Reportes se presentará la pantalla que se muestra en la Figura B.41 a seguir:



Figura B.41: Menú de los reportes de Basys

Partido

Este reporte presenta la información de un partido específico, en donde se detallan los jugadores con sus respectivos valores obtenidos durante el partido.

Para obtener este reporte el usuario debe seleccionar el link *Partido* y a continuación se presenta una pantalla en donde se deben seleccionar los parámetros de consulta para generar el reporte, como se muestra a continuación en la Figura B.42.

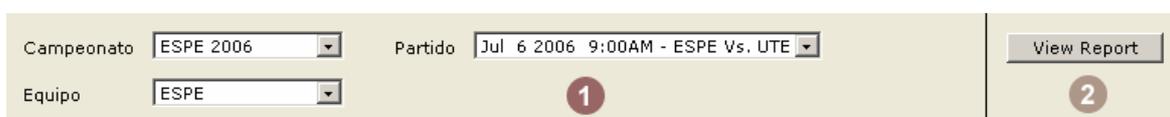


Figura B.42: Parámetros para generar el reporte Partido

En la sección 1 se pueden observar tres parámetros necesarios para generar el reporte Partido, los cuales son:

- **Campeonato:** se despliegan todos los campeonatos registrados en la aplicación.
- **Partido:** muestra los partidos que se han realizado en el campeonato seleccionado.
- **Equipo:** despliega los dos equipos que participaron en el partido.

Para generar el reporte se debe llegar a seleccionar el equipo del que se desea obtener la información del partido y seguidamente se selecciona el botón *View Report* que se encuentra en la sección 2.

Luego se debe esperar un momento mientras se genera el reporte, para lo cual se visualizará el siguiente mensaje que se encuentra en la Figura B.43 a seguir:



Figura B.43: El reporte se está generando

Y luego se presenta en la pantalla, el reporte del partido del equipo seleccionado, como se muestra en la Figura B.44 a continuación:

Facultad de Educación Física, Deportes y Recreación

Información del Partido

Fecha: Thursday, July 06, 2006 Hora: 9:00:00 AM Campeonato: Mujeres 2006
 Equipo Local: ESPE Equipo Visitante: UTE

Periodo: 4 Marcador: 68

Nro.	Jugador	Corta		Larga		Tiro Libre		Rebote		Asistencia	Pérdida	
		C	F	C	F	C	F	Of.	Def.		Balon	V
3	Robayo Francisco	7	0	4	2	2	0	2	2	0	2	
5	Goyes Henry	0	3	3	1	4	0	2	3	3	2	
7	Orellana Rene	2	2	3	0	3	2	3	1	2	4	
4	Puertas Eduardo	2	0	1	2	0	0	1	3	0	1	
1	Hurtado Danny	2	0	0	0	0	0	0	0	0	0	
9	Zambrano Héctor	0	0	0	0	0	0	0	0	0	0	
2	Barraquán José	0	0	0	0	0	0	0	0	0	0	

Más Reportes ...

Figura B.44: Ventana del reporte generado

A continuación se muestra la Tabla B.11 que describe las secciones:

Tabla B.11: Secciones de la ventana del reporte generado

Sección	Descripción
1	Opciones del reporteador: en esta sección se presenta un marco de opciones que permite al usuario navegar por el reporte (recorrer el reporte de página en página), visualizar de mejor maneja el documento; adicionalmente un buscador por palabras, permite seleccionar el formato al que desea ser exportado el reporte como por ejemplo Excel y finalmente un botón para imprimir el mismo.
2	Información del reporte: esta sección presenta la información del reporte generado.
3	Visualizar el menú de los reportes: permite visualizar el menú general de los reportes.

Todos los reportes presentan las mismas opciones mencionadas anteriormente.

Estadísticas del Jugador por Partido

Este reporte presenta las estadísticas de cada jugador en base a los datos obtenidos en el partido. Los parámetros que se necesitan para generar el partido son los mismos que los parámetros del reporte partido. Al generar el reporte se presenta un reporte con los siguientes datos (Figura B.45).

Facultad de Educación Física, Deportes y Recreación						
Estadísticas de los Jugadores por Partido						
Fecha:	Thursday, July 06, 2006	Hora:	9:00:00 AM	Campeonato:	Mujeres 2006	
Equipo Local:	ESPE	Equipo Visitante:	UTE			
Periodo:	4	Marcador:	68		Posesión del Balon:	3
Nro.	Nombre	Tiros Campo	Rebotes Defensivos	Rebotes Ofensivos	Tiros Libres	Faltas
3	Robayo Francisco	13.96	22.22	25.00	5.08	5.0
5	Goyes Henry	5.91	33.33	25.00	10.15	2.0
7	Orellana Rene	6.83	11.11	37.50	4.57	0.0
4	Puertas Eduardo	5.43	33.33	12.50	0.00	0.0
1	Hurtado Danny	2.54	0.00	0.00	0.00	0.0
9	Zambrano Héctor	0.00	0.00	0.00	0.00	0.0
2	Barragán José	0.00	0.00	0.00	0.00	0.0

Figura B.45: Datos del reporte Estadísticas del Jugador por Partido

- **Posesión del balón:** es la medición del ritmo del partido que determina por ejemplo si el partido ha sido más rápido o más lento.
- **Nro.:** el número del Jugador en el partido
- **Nombre:** nombre completo del jugador
- **Tiros de Campo:** expresa la eficacia de un jugador en los tiros de campo.
- **Rebotes Defensivos y Ofensivos:** hace referencia a la eficacia de los jugadores con respecto al equipo, es el porcentaje de rebotes defensivos u ofensivos (número de rebotes).

- **Tiros Libres:** muestra una estadística de eficacia basada en el porcentaje de tiros libres (relación entre convertidos y fallados) y el número de tiros libres convertidos.
- **Faltas:** es la relación entre las faltas recibidas y las faltas cometidas.
- **Control del Balón:** es la relación entre las asistencias y las pérdidas de balón.
- **Efectividad:** es el coeficiente de evaluación de la prestación (eficiencia) de los jugadores en el partido.

Estadísticas del Jugador por Campeonato

Este reporte presenta todas las estadísticas de los jugadores en los partidos que se realizaron en un determinado campeonato. Para generar el reporte es necesario seleccionar un campeonato específico. Además el reporte presenta otro parámetro de consulta que no es obligatorio, el mismo que es un rango de las fechas en las que se efectuaron los partidos, como se muestra a continuación en la Figura B.46:



Campeonato

Fecha inicial NULL

Fecha final NULL

Figura B.46: Parámetros del reporte

Si se desea especificar un rango de fechas, el usuario debe desmarcar el campo *NULL* y presionar en la imagen que presenta un calendario y seleccionar la fecha deseada, como se muestra en la Figura B.47 a continuación:



Figura B.47: Especificar una fecha

El reporte que se genera se encuentra agrupado por partido, y muestra la información de cada equipo, como se muestra a continuación en la Figura B.48:

Facultad de Educación Física, Deportes y Recreación						
Estadísticas de los Jugadores en el Campeonato: ESPE 200						
PARTIDO: Jul 6 2006 9:00AM - ESPE Vs. UTE						
Equipo: ESPE						
Nro.	Nombre Jugador	Tiros Campo	Rebotes Defensivos	Rebotes Ofensivos	Tiros Libres	Faltas
2	Barragán José	0.00	0.00	0.00	0.00	
5	Goyes Henry	5.91	33.33	25.00	10.15	
1	Hurtado Danny	2.54	0.00	0.00	0.00	
7	Orellana Rene	6.83	11.11	37.50	4.57	
4	Puertas Eduardo	5.43	33.33	12.50	0.00	
6	Puertas Vicente	0.00	0.00	0.00	0.00	
8	Robayo Alex	0.00	0.00	0.00	0.00	
3	Robayo Francisco	13.96	22.22	25.00	5.08	

Figura B.48: Reporte Estadísticas de los Jugadores por Campeonato

Estadísticas del Equipo por Campeonato

El siguiente reporte permite obtener las estadísticas de todo el equipo en un campeonato determinado. Este reporte también puede filtrarse opcionalmente por las fechas de los partidos. La información que se muestra es la misma que en el reporte de las estadísticas del jugador, pero en este caso los valores que se

presentan, son los valores totales del equipo. El reporte que se genera corresponde a un reporte como el que se muestra en la Figura B.49:

Facultad de Educación Física, Deportes y Recreación				
Estadísticas de los Equipos en el Campeonato: ESPE 2006				
Partido	Nombre	Tiros Campo	Tiros Libres	Faltas
Jul 6 2006 9:00AM - ESPE Vs. UTE	ESPE	35.10	18.68	
Jul 4 2005 8:00AM - Cheers Vs. Chelas	Cheers	33.47	16.61	

Figura B.49: Reporte Estadísticas de los Equipos por Campeonato

Jugador más valioso

Este reporte presenta una lista de todos los jugadores en un determinado campeonato. La lista está ordenada en forma descendente tomando en cuenta el promedio de efectividad, puntos y asistencias que ha realizado el jugador en el campeonato seleccionado. Este orden determina los jugadores más valiosos por campeonato. Un ejemplo de este reporte se presenta en la Figura B.50:

Facultad de Educación Física, Deportes y Recreación				
ESPE				
Jugador más valioso en el Campeonato: ESPE 2006				
Posición	Nombre del Jugador	Efectividad	Puntos	Asistencias
1	ROBAYO FRANCISCO	28.00	28.00	0.00
2	Del Castillo Gabriela	26.00	28.00	1.00
3	Gárces Elena	18.00	18.00	1.00
4	Goyes Henry	15.00	13.00	3.00
5	Orellana Rene	14.00	16.00	2.00
6	Chicaiza Alexandra	13.00	12.00	1.00
7	Castro Andrea	10.00	13.00	5.00
8	Granja Vale	9.00	10.00	2.00
9	Puertas Eduardo	8.00	7.00	0.00
10	Hurtado Danny	4.00	4.00	0.00
11	Zambrano Héctor	0.00	0.00	0.00

Figura B.50: Reporte Jugador más valioso

Equipo más valioso

De igual manera, este reporte presenta una lista de todos los equipos en un determinado campeonato. La lista está ordenada en forma descendente tomando en cuenta el promedio de efectividad, puntos y asistencias que ha realizado el jugador en el campeonato seleccionado. Este orden determina los equipos más valiosos por campeonato. Un ejemplo de este reporte se presenta a continuación en la Figura B.51:



Facultad de Educación Física, Deportes y Recreación **ESPE**

Equipo más valioso en el Campeonato: ESPE 2006

Posición	Nombre del Equipo	Efectividad	Puntos	Asistencias
1	ESPE	89.00	90.00	3.00
2	Cheers	76.00	81.00	10.00

Figura B.51: Reporte Equipo más valioso

ANEXO C

CARTA DE ACEPTACIÓN DE LA FEFDER

BIOGRAFÍA

DATOS PERSONALES.

Nombre: Puertas Véjar Andrea Elizabeth
Fecha de nacimiento: 22 de Noviembre de 1982
Edad: 23 años
Lugar de nacimiento: Quito – Ecuador
Dirección: Cuba 218 y la Isla
Teléfonos: Domicilio: 250 8706 - 222 6825
Celular: 09-603 6183
Estado civil: Soltera
C.I.: 110331592-3
E-mail: apuertas@mail.com

ESTUDIOS.

Primarios: Sagrados Corazones de Rumipamba
Secundarios: Maristas Los Andes (1er. – 3er. Curso)
Colegio e Instituto Ecuatoriano de Informática
(4to. - 6to. Curso)
Bachiller en Informática y Computación.
Universitarios: Escuela Politécnica del Ejército ESPE
Facultad de Ingeniería de Sistemas e Informática
Egresada

TÍTULOS.

- Bachiller en Ciencias - Especialización Informática.
- Suficiencia en el idioma Inglés. OBM
- Suficiencia en el idioma Inglés. Instituto de idiomas ESPE

MERECIMIENTOS.

- Diploma de excelencia en puntaje total de conducta y aprovechamiento en el Primer Curso. 1995
- Becas de excelencia en puntaje total de conducta y aprovechamiento en Cuarto, Quinto y Sexto Curso. 1998, 1999, 2000.
- Abanderada del Pabellón Nacional del Colegio e Instituto Ecuatoriano de Informática. 2000.
- Mejor egresada del Colegio e Instituto Ecuatoriano de Informática. 2000.
- Tercer Lugar – Categoría Avanzada en el Primer Concurso de Desarrollo de Software, Facultad de Sistemas e Informática de la ESPE. 2003.
- Beca Académica 80% ESPE, 8 semestres consecutivos.

BIOGRAFÍA

DATOS PERSONALES.

Apellidos, Nombres: Robayo Tipán Francisco Javier
Fecha de nacimiento: 3 de Noviembre de 1982.
Edad: 23 años
Lugar de nacimiento: Quito – Ecuador
Dirección: OE1A S38-83 y Huayanay Ñan - Quitumbe
Teléfonos: Domicilio: 2690-407
Celular: 09-600-8934
Estado civil: Soltero
C.I.: 171631546-8
E-mail: frobayo@mail.com
frankjsrc@yahoo.es

ESTUDIOS.

Primarios: Escuela Particular Católica “RAFAEL BUCHELI”
Secundarios: Colegio Militar No. 10 “ABDÓN CALDERÓN” – Quito
Bachiller en Comercio y Administración
Especialidad Computación.
Universitarios: Escuela Politécnica del Ejército ESPE
Facultad de Ingeniería de Sistemas e Informática
Egresado. Culminando Tesis.

TÍTULOS.

- Bachiller en Comercio y Administración Especialidad Computación
- Suficiencia en el idioma Inglés. Instituto de Idiomas ESPE.

MERECIMIENTOS.

- Abanderado del Pabellón de la Ciudad de Quito de la Escuela Particular Católica “RAFAEL BUCHELI”.
- Comandante de curso, de primero a sexto curso del Colegio Militar “ABDÓN CALDERÓN”
- Abanderado del Pabellón Nacional del COMIL 10 “ABDÓN CALDERÓN” y BRIGADIER MAYOR 1999 – 2000
- Mejor Egresado del COMIL 10 “ABDÓN CALDERÓN” promoción 1999 – 2000.
- Beca Académica 80% ESPE, 8 semestres consecutivos

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Andrea E. Puertas V.

Francisco J. Robayo T.

**DECANO DE LA FACULTAD DE INGENIERÍA
DE SISTEMAS E INFORMÁTICA**

Tcrn. de E.M. MARCO QUINTANA

Sangolquí, 30 de Agosto de 2006