



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS**

**TEMA: DESARROLLO E IMPLEMENTACIÓN DE UN
SISTEMA DE GESTIÓN ACADÉMICA, PARA EL CENTRO
TRANSFERENCIA Y DESARROLLO TECNOLÓGICO ESPE-
CECAI.**

AUTOR: PACHECO GUALLASAMIN RINA DEL CONSUELO

**DIRECTOR: ING. GERMÁN, ÑACATO
CODIRECTOR: ING. ARROYO, RUBÉN**

SANGOLQUÍ, JUNIO 2015

CERTIFICADO

Ing. Germán Ñacato

Ing. Rubén Arroyo

CERTIFICAN

Que el trabajo titulado “Sistema de Gestión Académica para el Centro de Transferencia y Desarrollo Tecnológico ESPE-CECAI”, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecida por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

El mencionado trabajo consta de un documento empastado y dos discos compactos.

Autorizan a Rina del Consuelo Pacheco Guallasamin a Ing. Director, en su calidad de Director de Carrera.

Sangolquí, junio del 2015.



Ing. Germán Ñacato
DIRECTOR



Ing. Rubén Arroyo
CODIRECTOR

AUTORÍA DE RESPONSABILIDAD

Rina del Consuelo Pacheco Guallasamin

DECLARO QUE:

El proyecto denominado: **Sistema de Gestión Académica para el Centro de Transferencia y Desarrollo Tecnológico ESPE-CECAI**, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan en los pies de página correspondientes, cuyas fuentes se incorporan en la biografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo de este contenido, veracidad y alcance del proyecto de grado en mención.

Sangolquí, junio del 2015



RINA DEL CONSUELO PACHECO GUALLASAMIN

AUTORIZACIÓN

Yo, Rina del Consuelo Pacheco Guallasamin

Autorizo a la UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, la publicación, en la biblioteca virtual de la Institución del trabajo “SISTEMA DE GESTIÓN ACADÉMICA PARA EL CENTRO DE TRANSFERENCIA Y DESARROLLO TECNOLÓGICO ESPE-CECAI”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, junio del 2015



RINA DEL CONSUELO PACHECO GUALLASAMIN

DEDICATORIA

Principalmente quiero dedicar este trabajo a Dios, por haberme regalado el don de la vida, por ser mi fortaleza en mis momentos de debilidad y por brindarme una vida llena de mucho aprendizaje, experiencia, felicidad y permitirme el haber llegado hasta este momento tan importante de mi formación profesional.

Gracias Dios por estar al lado mío en cada paso que doy cuidándome y dándome mucha fortaleza para seguir en pie y lograr realizar mi sueño como profesional, un sueño que creí distante y ahora en día lo estoy logrando.

A mis padres Eduardo y Angelita, porque con su amor, trabajo y sacrificio de todos estos años creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, de mi vida, y porque el orgullo que sienten por mí, fue lo que me hizo llegar hasta esta instancia final de mis estudios.

También se la dedico a mí amado esposo Jorge, compañero inseparable de cada jornada. El representó gran esfuerzo y tesón en momentos de decline y cansancio impulsando al logro de mis sueños y éxitos profesionales.

A mi hija Stefany quien ha sido mi mayor motivación para nunca rendirme en mis estudios y para poder llegar a ser un ejemplo para ella.

A mi angelito del cielo, que sé que en todo momento me cuida y donde quiera que se encuentre sé que va estar feliz de verme culminando mis estudios.

A mi hermano Patricio quien siempre ha estado junto a mí brindándome su apoyo incondicional.

A mi amigo Carlitos y Carla que de una u otra manera estuvieron pendientes con su apoyo llenándome de sabiduría y conocimiento.

A todos en general por darme el tiempo para realizarme profesionalmente.

AGRADECIMIENTO

A Dios

Por estar siempre conmigo en los buenos y malos momentos de mi existencia.

A mis Padres

Eduardo: Detrás de este logro estás tú papito, tu apoyo y cariño. Porque sé que después de todo este tiempo transcurrido aun confías en mí. Gracias por darme la oportunidad de hacer realidad este sueño compartido, por alentarme a hacer lo que quiero y ser como soy. Te adoro.

Angelita: Si hay algo que sé hacer bien es por ti, y cuando llega la recompensa por un esfuerzo no puedo dejar de recordar tu cercanía, complicidad, devoción tu ejemplo y admiración que siento por ti ya que eres mi mejor amiga y mi aliada, gracias por todo el apoyo brindado en esta tesis y por ser el pilar fundamental en mi vida. Que esta sea la recompensa a tantos años de entrega, desvelos, apoyo: Estamos juntas. Te quiero con todo mi corazón.

A mi Esposo Jorge W.B.

Gracias por tu infinita paciencia, por tu tierna compañía y tu inagotable apoyo, por compartir tu vida con la mía y mis logros, de esta tesis también son tuyos. Te amo.

A mi Stefy

Porque a pesar de ser tan pequeña me das tu tiempo valioso para cumplir mis logros además por ser la inspiración de mi vida.

A mi Hermano

Porque me has estado impulsando a continuar con mis sueños y anhelos además de ser mí ejemplo.

Amis amigos

A Carlitos y Karla, por haberme tenido la paciencia necesaria y por motivarme a seguir adelante en los momentos de desesperación.

A mis Directores de Tesis, Ing. Germán Ñacato e Ing. Rubén Arroyo

Agradezco la confianza, apoyo y dedicación de su tiempo valioso además por haber compartido conmigo sus conocimientos y sobre todo su amistad.

ÍNDICE DE CONTENIDO

CERTIFICADO	ii
AUTORÍA DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
RESUMEN.....	xvii
SUMMARY	xviii
 CAPÍTULO 1	
INTRODUCCIÓN	1
1.1 Reseña Histórica de Innovativa.....	1
1.2 Justificación.....	2
1.2.1 Planteamiento del Problema.....	2
1.2.2 Descripción de la Solución Propuesta.....	2
1.3 Alcance.....	3
1.4 Objetivos	5
1.4.1 Objetivo General	5
1.4.2 Objetivos Específicos.....	5
 CAPÍTULO 2	
MARCO TEÓRICO	6
2.1 Introducción	6
2.1.1 Conceptos	6
2.1.2 Desarrollo de software	6
2.1.3 Gestión Académica	7
2.1.4 Metodología.	7
2.2 Extreme Programming	8
2.2.1 Definición.....	8
2.2.2 Objetivo de la Programación Extrema	9
2.2.3 Características esenciales de la metodología	9
2.2.3.1 Simplicidad.....	9
2.2.3.2 Comunicación.....	10

2.2.3.3	Retroalimentación(feedback)	10
2.2.3.4	Coraje o valentía.....	11
2.2.3.5	Respeto	11
2.2.4	Otras características fundamentales de XP	11
2.2.5	Fases de la Metodología.....	12
2.2.5.1	Fase 1: Planificación del proyecto.	14
2.2.5.1.1	Historias de usuario.-.....	14
2.2.5.1.2	Plan de Entrega (Release plan).-	14
2.2.5.1.3	Iteraciones (Iteration Plan).....	15
2.2.5.1.4	Velocidad del proyecto.....	15
2.2.5.1.5	Programación en pareja.....	15
2.2.5.1.6	Reuniones diarias (Stant Up Meeting)	15
2.2.5.2	Fase 2: Diseño.	16
2.2.5.2.1	Diseños simples.....	16
2.2.5.2.2	Glosarios de términos.....	16
2.2.5.2.3	Riesgos	16
2.2.5.2.4	Funcionalidad extra	16
2.2.5.2.5	Refactorizar	16
2.2.5.2.6	Tarjetas C.R.C.	16
2.2.5.3	Fase 3: Codificación.....	17
2.2.5.3.1	Disponibilidad del Cliente.....	17
2.2.5.3.2	Programación dirigida por las pruebas.....	17
2.2.5.3.3	Programación por Parejas.....	18
2.2.5.3.4	Integraciones Permanentes.....	18
2.2.5.4	Fase 4: Pruebas.....	18
2.2.5.5	Fase 5: Producción	18
2.2.5.6	Fase 6: Mantenimiento	19
2.2.5.7	Fase 7: Muerte del Proyecto.....	19
2.3	Open Source	19
2.3.1	Definición.....	19
2.3.2	Características de Open Source.....	20
2.3.3	Formas de Obtener Software Libre	20

2.3.4	Ventajas	20
2.3.5	Desventajas.....	20
2.4	El Lenguaje Unificado de Modelado (UML).....	20
2.4.1	Definición de UML	21
2.4.1.1	Vistas:.....	22
2.4.1.2	Diagramas:	22
2.4.1.2.1	Casos de Uso.-.....	23
2.4.1.2.2	Diagrama de Secuencia	23
2.4.1.2.3	Diagrama de Colaboración.....	24
2.4.1.2.4	Diagrama de Estados	25
2.4.1.2.5	Diagrama de Actividad.....	26
2.4.1.2.6	Diagrama de Clases	27
2.4.1.2.7	Diagrama de Objetos	27
2.4.1.2.8	Diagrama de Componentes	28
2.4.1.2.9	Diagrama de Implementación	28
2.4.1.3	Símbolos o Elementos de modelo:.....	29
2.4.1.4	Reglas o Mecanismos generales:.....	29
2.4.2	Fases del Desarrollo de un Sistema.....	29
2.4.2.1	Análisis.....	29
2.4.2.2	Diseño.....	29
2.4.2.3	Programación	30
2.4.2.4	Pruebas	30
2.5	Mapa Mental	30
2.5.1	Definición.....	30
2.5.2	Características de los Mapas Mentales.....	31
2.5.3	Ventajas	32
2.6	J2EE	32
2.6.1	Definición.....	32
2.6.2	Arquitectura de Aplicación Web J2EE	32
2.7	Herramientas de Desarrollo.....	34
2.7.1	Net Beans	34
2.7.2	Maven.....	34

2.7.2.1	Ciclo de vida.....	36
2.7.3	Postgres	36
2.7.3.1	Características Generales	39
2.7.3.2	Limitaciones	39
2.7.3.3	Ventajas	39
2.7.4	Power Designer	40
2.7.4.1	Técnicas de Modelamiento.....	40
2.7.4.2	Beneficios.....	43
2.7.4.3	Ambientes en PowerDesigner:	43
2.7.4.3.1	Arquitectura de Datos.....	43
2.7.4.3.2	Arquitectura Empresarial	44
2.7.4.3.3	Arquitectura de la Información	44
CAPÍTULO 3		
ANÁLISIS Y DISEÑO.....		46
3.1	Identificación de Requerimientos Funcionales	46
3.2	Descripción del Problema	46
3.3	Propuesta de Solución	46
3.3.1	Casos de Uso	47
3.3.2	Diagramas de Secuencia.....	75
3.3.3	Diagrama de Base de Datos	77
3.3.4	Diccionario de Datos	79
3.3.4.1	Listado de Tablas	79
3.3.5	Arquitectura del Sistema	90
3.3.5.1	Arquitectura de Despliegue.....	90
3.3.5.2	Arquitectura de Desarrollo	91
CAPÍTULO 4		
IMPLEMENTACIÓN Y PRUEBA		92
4.1	Implementación	92
4.1.1	Instalación y configuración del Sistema.....	92
4.1.1.1	Instalación del JDK	92
4.1.1.2	Instalación de Postgres	94
4.1.1.3	Instalación de Glassfish.....	127

4.1.1.3.1	Instalación de Glassfish.....	127
4.1.1.3.2	Configuración de Glassfish.....	128
4.1.1.4	Despliegue del Sistema.....	132
4.2	Seguimiento y ejecución del sistema de Gestión Académica.....	133
CAPÍTULO 5		
CONCLUSIONES Y RECOMENDACIONES.....		134
5.1	Conclusiones.....	134
5.2	Recomendaciones.....	134
BIBLIOGRAFÍA.....		136

ÍNDICE DE TABLAS

Tabla 1	Administrador de Estado Civil.....	49
Tabla 2	Administrar Región.....	50
Tabla 3	Administrar Nivel Académico	50
Tabla 4	Administrar Forma de Pago	51
Tabla 5	Administrar Tipo de Cliente.....	52
Tabla 6	Administrar Documento.....	53
Tabla 7	Administrar Perfil.....	55
Tabla 8	Administrar Usuario.....	55
Tabla 9	Administrar Usuario por Centro.....	56
Tabla 10	Administrar Cambio de Clave.....	57
Tabla 11	Administrar Curso.....	59
Tabla 12	Administrar Profesor.....	60
Tabla 13	Centro de Capacitación	61
Tabla 14	Registrar Planificación	62
Tabla 15	Actualiza Planificación	63
Tabla 16	Activar Inscripción.....	64
Tabla 17	Revisar Inscripción.....	66
Tabla 18	Actualizar datos de alumno.....	67
Tabla 19	Visualizar Cursos	69
Tabla 20	Visualizar Información.....	70
Tabla 21	Visualizar Información.....	71
Tabla 22	Registrar Inscripción	72
Tabla 23	Visualizar Nota.....	73
Tabla 24	Visualizar Asistencia.....	74
Tabla 25	Listado de Tablas	79
Tabla 26	Asistencia	80
Tabla 27	Aula	80
Tabla 28	Beneficiario	80
Tabla 29	Beneficiario	81
Tabla 30	Cantón	81
Tabla 31	Centro	81

Tabla 32	Curso	82
Tabla 33	Civil.....	82
Tabla 34	Fecha descripción.....	82
Tabla 35	Forma Pago	82
Tabla 36	Inscripción.....	83
Tabla 37	Módulo	84
Tabla 38	Nivel Académico.....	84
Tabla 39	Notas.....	84
Tabla 40	Pagos	85
Tabla 41	Parroquia	85
Tabla 42	Perfil	85
Tabla 43	Persona	86
Tabla 44	Persona Centro	87
Tabla 45	Plan aula	87
Tabla 46	Planificación.....	88
Tabla 47	Profesor Curso.....	88
Tabla 48	Provincia.....	88
Tabla 49	Títulos.....	89
Tabla 50	Asistencia	89
Tabla 51	Notas.....	89
Tabla 52	Características del Servidor.....	92

ÍNDICE DE FIGURAS

Figura 1.	Características de la Metodología	9
Figura 2	Fases de Extreme Programming.....	12
Figura 3	Fases de la Metodología de Xtreme Programming	13
Figura 4	Modelo de Tarjeta CRC	17
Figura 5	Modelo UML.....	21
Figura 6	Casos de Uso	23
Figura 7	Casos de Uso	23
Figura 8	Diagramas de Secuencia.....	24
Figura 9	Representación de una instancia de la Clase.....	24
Figura 10	Representación de Múltiples instancias de la Clase.....	24
Figura 11	Diagrama de Colaboración.....	25
Figura 12	Diagrama de Estados.....	26
Figura 13	Diagrama de Actividad.....	26
Figura14	Diagrama de Clases	27
Figura 15	Diagrama de Objetos	27
Figura 16	Diagrama de Componentes	28
Figura 17	Diagrama de Implementación	28
Figura 18	Mapa Mental	30
Figura 19	Ideas de un Mapa Mental	31
Figura 20	Representación de J2EE.....	33
Figura 21	Postgress.....	37
Figura 22	Modelo Conceptual	40
Figura 23	Modelo Físico.....	41
Figura 24	Data Mapping Editor.....	41
Figura 25	Data Movement Modeling	42
Figura 26	XML Modeling	42
Figura 27	Diagrama de Casos de Uso de Parámetros del Sistema	48
Figura 28	Diagrama de Casos de Uso de Seguridad del Sistema	54
Figura 29	Diagrama de Casos de Uso de Estudios del Sistema	58
Figura 30	Diagrama de Casos de Uso de Revisar Inscripción.....	65
Figura 31	Diagrama de Casos de Uso de Visualizar Información.....	68

Figura 32	Diagrama de Secuencia de Planificación	75
Figura 33	Diagrama de Secuencia de Inscripciones	76
Figura 34	Diagrama de Modelo Lógico.....	77
Figura 35	Diagrama de Modelo Físico	78
Figura 36	Servidor de Base de Datos	90
Figura 37	Servidor de Aplicaciones Glassfish.....	91
Figura 38	Ejecución del Instalador	93
Figura 39	Finalización de la Instalación.....	93
Figura 40	Instalador de Postgres.....	94
Figura 41	Especificación de Usuario y Clave.....	94
Figura 42	Especificación del Puerto	95
Figura 43	Configuración Regional y Lenguaje de la Base de Datos.....	95
Figura 44	Pantalla de Inicio de Instalación.....	96
Figura 45	Pantalla de Administrador de Postgres	96
Figura 46	Pantalla de Creación de Base de Datos	97
Figura 47	Pantalla de Ingreso del Nombre de la Base de Datos.....	97
Figura 48	Creación de la Estructura de la Base de Datos.....	98
Figura 49	Ejecución del Script	98
Figura 50	Pantalla de descompresión del Archivo Glassfish	127
Figura 51	Pantalla de Consola de Comandos	127
Figura 52	Pantalla de Configuración del Dominio.....	128
Figura 53	Pantalla del Comando Create Domain	128
Figura 54	Pantalla de descompresión del archivo postgres	129
Figura 55	Pantalla de Acceso a la Aplicación	129
Figura 56	Pantalla de ingreso al Sistema.....	129
Figura 57	Pantalla de Acceso al Resources JDBC	130
Figura 58	Pantalla del Pool de Conexión	130
Figura 59	Pantalla de Configuración del Pool de Conexión.....	131
Figura 60	Pantalla de Visualización del Pool de Conexión.....	131
Figura 61	Pantalla de Verificación de la Conexión.....	131
Figura 62	Pantalla de Creación del Data Source	132
Figura 63	Pantalla de despliegue del Sistema.....	132

Figura 64	Pantalla de Visualización de Aplicaciones.....	133
Figura 65	Acceso al Sistema de Gestión Académica	133

RESUMEN

INNOVATIVA (CTT-ESPE-CECAI), Centro de Transferencia y Desarrollo Tecnológico de la UFA-(ESPE); es una Institución de Derecho Público, cuya misión es transferencia de tecnología mediante el desarrollo de proyectos y prestación de servicios de capacitación, actualmente la gestión académica de los centros de capacitación y consolidación de la información lo realizan en hojas electrónicas, lo que no ha permitido que la Unidad obtenga información en tiempo real limitando la toma de decisiones oportuna. Se ha visto la necesidad de contar con herramientas tecnológicas que permitan registrar la información en línea de los alumnos, consolidarla en tiempo real y mantener un historial mediante la Implementación de un Sistema de Gestión Académica en ambiente web, optimizando los procesos manuales como control de matrículas, calificaciones, asistencia, emisión de certificados, etc. Se aplicó la metodología de desarrollo denominada Extreme Programming, por su rapidez y agilidad en el desarrollo de productos de software. Las herramientas de desarrollo que se utilizaron son de última tecnología y alineadas a las políticas gubernamentales en el uso de herramientas Open Source por lo que se utilizó NetBeans como framework de desarrollo, para el almacenamiento de datos Postgress, finalmente Glasfish como Servidor de Aplicaciones. Los resultados obtenidos en el CTT-ESPE-CECAI, exclusivamente en el Área de Capacitación Tecnológica tanto para la planta Central como los centros de Capacitación fue la obtención de información consolidada de los centros, control de los alumnos sus calificaciones, certificados, etc. adicionalmente se obtuvo organizar y estandarizar procesos.

Palabras Clave:

- **METODOLOGÍA XP,**
- **OPEN SOURCE,**
- **JAVANETBEANS.**

SUMMARY

INNOVATIVA (CTT-ESPE-CECAI) Transfer Center and Technological Development of the UFA (ESPE); is a public institution whose mission is technology transfer through the project development and delivery of training, currently the academic management of training centers and consolidation of information is done in spreadsheets, which has not allowed it acquires information in real time limiting timely decisions. It has been the need for technological tools to record information online students, consolidate real-time and keep a history by implementing a Management System Academic Web environment, optimizing manual processes and control plates , grades, attendance, issuing of certificates, etc. Development methodology called Extreme Programming, for its speed and agility in the development of applied software products. Development tools that were used are the latest technology and aligned to government policies on the use of Open Source tools so as NetBeans development framework used for the storage of data Postgress finally Glasfish as Application Server. The results of the CTT-ESPE-CECAI exclusively in the area of technology training for both the Central plant as training centers was to obtain consolidated information centers, students control their qualifications, certificates, etc. additionally obtained organize and standardize processes.

KeyWords:

- **METODOLOGÍA XP,**
- **OPEN SOURCE,**
- **JAVANETBEANS**

CAPÍTULO 1

INTRODUCCIÓN

1.1 Reseña Histórica de Innovativa

INNOVATIVA (CTT-ESPE-CECAI), Centro de Transferencia y Desarrollo Tecnológico de la Universidad de la Fuerzas Armadas ESPE; es una Institución de Derecho Público, con autonomía administrativa y financiera, cuya misión es transferencia de tecnología mediante el desarrollo de proyectos y prestación de servicios de capacitación.

En concordancia con la misión el CTT posee la Unidad Capacitación tecnológica cuya actividad se ha enmarcado en transferir conocimientos en áreas administrativas y técnicas, respondiendo así a las necesidades y requerimientos de la sociedad en este ámbito para todo el país.

En la actualidad es considerado como el Centro de Capacitación Informática presencial más grande, se ha venido ampliando los servicios de capacitación, a otras áreas: Administrativas, Financiera, Diseño Gráfico, Programación, entre otros. A partir de agosto de 2010, se ha transformado en la Unidad de Capacitación y Productividad.

La Unidad de Capacitación Tecnológica al momento administra 20 centros de capacitación distribuidos en todo el país, a los cuales acuden diariamente 15 alumnos a tomar los programas de capacitación que brinda esta unidad.

“**INNOVATIVA** es la nueva imagen corporativa del Centro de Transferencia y Desarrollo Tecnológico CTT ESPE-CECAI, el cual fue creado el 24 de junio de 2003 como una unidad adscrita a la ESPE con autonomía administrativa y financiera, con la finalidad de ejecutar proyectos de investigación, innovación, transferencia de tecnología y capacitación integral, contribuyendo de esta manera con el desarrollo sustentable del Ecuador.” (Centro de Desarrollo Tecnológico ESPE CECAI, 2003)

Toda institución que brinda servicios educativos debe ofrecer servicios de calidad a sus estudiantes, tanto en el nivel académico como el nivel de servicios institucionales tales como: herramientas para inscripciones remotas y facilidad en la consulta de notas y horarios. Por lo que el presente proyecto plantea el desarrollo informático de un software que provee al personal administrativo, instructores y alumnos del Centro de

Transferencias de Tecnologías en el manejo de notas, cursos, materias, administración de instructores, alumnos y generación de certificados.

En una primera etapa se inicia con la identificación de las necesidades informáticas del Centro de Transferencia de Tecnologías, para identificar que procesos y áreas deben ser automatizados y la forma como el sistema informático debe almacenar o generar datos.

En una segunda etapa se desarrolla el sistema informático acorde a las necesidades previamente identificadas, para lo cual se empleará una metodología de desarrollo acorde a las características del software.

En la etapa final del proyecto se implementará el sistema informático desarrollado en el ambiente tecnológico del Centro de Transferencias de Tecnologías.

1.2 Justificación

1.2.1 Planteamiento del Problema

En la actualidad toda institución debe estar orientada a mejorar los servicios hacia el cliente, el Centro de Transferencia y Desarrollo Tecnológico ESPE-CECAI no debe ser ajeno a este cambio. En este momento todas las actividades de registro de instructores, estudiantes, horarios, cursos, notas e impresión de certificados de aprobación de los cursos son generados de forma manual, debido a que muchas de las personas que desean inscribirse en algún curso deben realizarlo de manera presencial en las instalaciones de los centros o subcentros de capacitación del Centro de Transferencias de Tecnologías de Información dispersas a nivel nacional, cada centro y subcentro maneja su propia información de manera independiente y la información no se encuentra centralizada lo que hace imposible que se puedan generar reportes estadísticos para la toma de decisiones a nivel gerencial.

1.2.2 Descripción de la Solución Propuesta

Para solucionar estos inconvenientes se propone el desarrollo de un sistema informático que provea de los servicios de inscripción de alumnos, registro instructores, materias, horarios, notas y generación de certificados de aprobación, los cuales podrán ser accedidos empleando internet y permitirá la generación de reportes debido que toda la información a nivel nacional estará centralizada en los equipos informáticos del Centro de Transferencias de Tecnologías de Información.

La arquitectura a emplearse será J2EE, la misma que permite el desarrollo de sistemas informáticos en ambiente Web.

1.3 Alcance

Desarrollar e Implementar un Sistema de Gestión Académico Web, que permita el registro del Estudiante, Instructores, Cursos, Horarios, Notas y el posterior despliegue del listado de aprobados y la impresión de certificados del área del CECAI, el mismo que estará centralizado en el servidor principal del CTT.

El Sistema de Gestión Académica tendrá los siguientes Módulos:

Parametrización

- Creación de Estado Civil
- Creación de Región
- Creación de Nivel Académico
- Creación de Formas de Pago
- Creación de Tipos de Cliente
- Creación de Descuentos

Módulo de Seguridad

- Definición de Perfil
- Definición de usuarios
- Usuario por Centro
- Cambio de Clave

Estudios

- Curso
- Profesor
- Centro de Capacitación
- Planificación
- Revisar Inscripciones
- Actualizar Datos del Alumno

Administración de Cursos

Creación de Cursos

- Registro de Contenidos
- Asignación de horarios
- Asignación de Instructores
- Ubicación

Creación de Profesores

- Estudios
- Asignación de Perfil

Creación de Centros de Capacitación

- Publicación de Curso
- Fecha Inicio de Curso
- Fecha Finalización de Curso
- Registro de Estudiantes
- Asignación de Instructores
- Matriculación en el Curso

Planificación

- Asigno Centro de Capacitación
- Asigno Profesor
- Asigno Aula

Revisar Inscripción

- Elije Curso
- Anula Inscripción
- Realiza Pago
- Cambia tipo de Alumno

Inicio de Curso

- Registro de Asistencia de Estudiantes
- Registro de Notas
- Publicación de Resultados

Reportes

- Generación e impresión de Certificados
- Reporte Económico por cursos
- Reporte de Inscritos por curso
- Reporte Aprobados por Curso
- Reporte de Notas de alumno
- Reporte de Asistencia de Alumno

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar e Implementar un Sistema de Gestión Académico en ambiente web, para el control de matrícula, cursos, alumnos, instructores, calificaciones de los estudiantes y emisión de certificados para el área del CECAI.

1.4.2 Objetivos Específicos

- Levantar y analizar requerimientos para el desarrollo de la aplicación de tal manera que cubra las diferentes consideraciones metodológicas y técnicas.
- Identificar los procedimientos de control de matrícula y calificaciones para generar reportes que agilicen los procesos manuales
- Diseñar el sistema de información para el control de matrícula y calificaciones.
- Implementar el sistema de gestión académica en ambiente web utilizando la metodología de desarrollo de Programación Extrema.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Introducción

Este capítulo contiene conceptos, la tecnología que se utilizará y metodología que se aplicará para el desarrollo del Sistema de Gestión Académica para el Centro De Transferencia y desarrollo ESPE-CECAI.

2.1.1 Conceptos

Software.- el software es el conjunto de instrucciones responsable de que el hardware realice su tarea además mediante las computadoras se emplean para manipular datos.

Tipos de Software

- ***Software del sistema:*** Es un conjunto de programas que administran los recursos de la computadora.
- ***Software de aplicaciones:*** Programas que son escritos por los usuarios para realizar una tarea específica en la computadora.
- ***Software de usuario final:*** Es el software que permiten el desarrollo de algunas aplicaciones directamente por los usuarios finales, el software del usuario final con frecuencia tiene que trabajar a través del software de aplicación y finalmente a través del software del sistema.
- ***El software integrado:*** está formado por un paquete de aplicaciones que pueden compartir datos, o por lo menos intercomunicar sus informaciones.
- ***El software libre:*** es el que otorga libertad a los usuarios para modificar, copiar, compartir, adaptar y mejorar los programas.
- ***Software libre*** - Representan al conjunto de programas en el que los usuarios disponen de plena libertad para copiarlo, compartirlo y modificarlo.

2.1.2 Desarrollo de software

Aplica principios, técnicas, herramientas y métodos para la construcción, implementación, instalación, mantenimiento y gestión de sistemas de información, es un programa que permite el desarrollo de aplicaciones, algunos de estos son java, visual basic, C++, etc. además es una estructura aplicada al desarrollo de un producto y se la conoce por IDE (Integrated Development Environment, por sus siglas en inglés).

El desarrollo de software es la realización sistemática de las actividades de planeación, diseño, codificación, pruebas, lanzamiento de productos de software nuevos, cumpliendo con los requisitos especificados y con las normativas de seguridad de información.

El desarrollo del software y la programación son uno de los pilares fundamentales de la informática y al cual dedican muchas horas de esfuerzos las empresas, colegios, academias y universidades.

Conforme avanza la tecnología, aparece nuevas soluciones, nuevas formas de programación, nuevos lenguajes y herramientas que intentan realizar el trabajo del desarrollador un poco más fácil.

En el mercado se intenta aportar las mejores soluciones por lo cual en la actualidad la sociedad se encuentra en una época de transición, que se encamina hacia un nuevo estilo de programación basada en estándares.

2.1.3 Gestión Académica

Son las acciones que realizan las organizaciones o instituciones académicas en función de su conocimiento. Se encarga de las tareas administrativas y académicas relacionadas con profesores y alumnos, así como de las tareas de soporte, tanto de las áreas de la institución como del servicio de la docencia que se imparte en sus diversos espacios.

2.1.4 Metodología.

Todo desarrollo de software tiene su riesgo y dificultad de controlar, si no se lleva una metodología adecuada, las más conocidas son:

- La metodología RUP es la más adaptable para proyectos de largo plazo.
- La metodología XP en cambio, se recomienda para proyectos de corto plazo.
- La metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.

Lo más importante antes de elegir la metodología que se debe aplicar para implementar el software, es determinar el alcance que tendrá el proyecto y luego ver cuál es la que más se acomoda a la aplicación y a las necesidades.

La metodología de desarrollo que se aplicará al proyecto de Gestión Académica para el Centro de Transferencia y Desarrollo Tecnológico ESPE-CECAI es Extreme Programming. Las herramientas de desarrollo que se utilizaron son:

- UML (Lenguaje Unificado de Modelamiento)
- Mapa Mental
- J2EE
- NetBeans
- Maven
- Postgress
- Power Designer

Son herramientas de fácil uso en las aplicaciones de desarrollo, además de tener un código fácil de mantener y legible para los diseñadores. Con el uso de herramientas Open Source, se utilizó NetBeans como framework de desarrollo, para el almacenamiento de datos Postgress y finalmente Glasfish como Servidor de Aplicaciones. Para el modelamiento de datos y procesos Lógico y Físico se utilizó Power Designer.

2.2 Extreme Programming

2.2.1 Definición

La programación extrema o *Xtreme Programming* (XP), fue formulada por Kent Beck en el año de 1990, es la metodología de desarrollo de la ingeniería de software más destacada, proporciona procesos ágiles e interpersonales para el éxito del desarrollo de software. La programación extrema se diferencia de las metodologías tradicionales porque pone más énfasis en la adaptabilidad que en la previsibilidad. XP puede adaptarse a los cambios en cualquier punto de la vida del proyecto y durante el ciclo de vida del software. Se basa en la realimentación continua entre el cliente y el equipo de trabajo, fomentan una comunicación fluida entre los participantes, con suficiente feedback para permitirle al equipo ver en dónde están y ajustar las prácticas a su situación única.

2.2.2 Objetivo de la Programación Extrema

- Satisfacer las necesidades del cliente
- Brindar al cliente o usuario el software que el necesita y cuando lo necesita.
- Dar respuestas rápidas a las necesidades del cliente incluso cuando los cambios sean al final del software.
- Potenciar al máximo el trabajo en grupo.
- Jefes clientes y desarrolladores forman parte del equipo de trabajo y están involucrados en el desarrollo del software.

2.2.3 Características esenciales de la metodología

A continuación se observa en la figura 1, donde se muestra las características de la metodología.

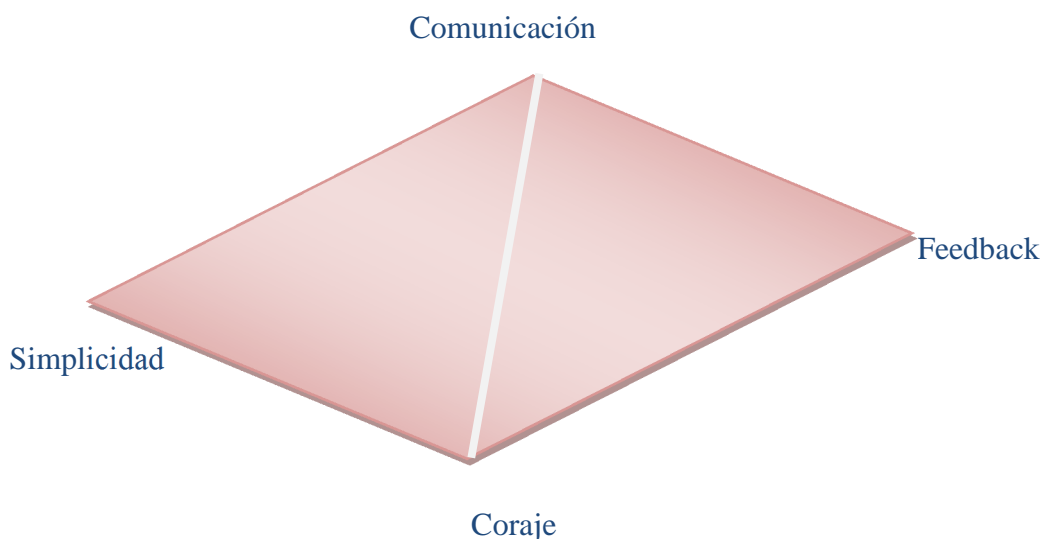


Figura 1. Características de la Metodología

2.2.3.1 Simplicidad

La simplicidad es la base de la programación extrema, se simplifica el diseño, para agilizar el desarrollo y facilitar el mantenimiento. Es necesaria e imprescindible la refactorización del código a medida que va creciendo el software.

La simplicidad debe aplicarse a la documentación, autodocumentado el código, eligiendo adecuadamente nombres de variables, métodos y clases. Cuando se aplica la simplicidad se asegura que mientras más grande sea el proyecto, todo el equipo conocerá mejor el sistema completo.

Un objetivo prioritario de *Xtreme Programming* es el de mantener las cosas simples.

- Pasar los test o pruebas.
- El código releva sus intenciones.
- Eliminar toda la duplicación.
- Reducir el número de elementos.

2.2.3.2 Comunicación

La comunicación es uno de los factores clave, necesarios para el éxito de los proyectos de software. Los programadores se comunican mejor con el código cuando más simple es, caso contrario hay que lograr hacerlo entendible para los desarrolladores que están inmiscuidos. Las pruebas unitarias y el código autodocumentado es otra forma importante de comunicación.

Los programadores se comunican constantemente gracias a la programación por parejas. La comunicación con el cliente debe ser fluida ya que el cliente forma parte del equipo de desarrollo además es quien decide qué características tienen prioridad y siempre debe estar disponible para solucionar dudas.

Un proyecto de software siempre se realiza en equipo, es fundamental que los canales de comunicación sean amplios.

Comunicación entre:

- Cliente y Equipo de desarrollo.
- Entre miembros del equipo.
- Entre programadores y testers.

Es fundamental tratar de eliminar barreras, aplicando la comunicación directa.

2.2.3.3 Retroalimentación(feedback)

La retroalimentación es necesaria en todas y cada una de las escalas ya que dimensiona el proyecto.

Evaluación del sistema: Se realizan pruebas unitarias, o pruebas periódicas de integración, los programadores siempre tienen información directa del estado del sistema después de la implementación de los cambios realizados.

La retroalimentación del cliente: Las pruebas funcionales o pruebas de aceptación, están escritas por el cliente y los probadores. Obtendrán información

concreta sobre el estado actual de su sistema. Esta opinión está prevista una vez cada dos o tres semanas por lo que el cliente puede dirigir fácilmente el desarrollo.

Evaluación del equipo: Cuando los clientes vienen con nuevas exigencias en la planificación del equipo eso da una estimación del tiempo que se tardará en poner en práctica.

El código es una fuente de retroalimentación. Por ejemplo, las pruebas unitarias informan sobre el estado del código y la ejecución de esta permite descubrir fallos debidos a cambios en el código del software. El cliente está constantemente viendo el producto y dando feedback es decir retroalimentando.

2.2.3.4 Coraje o valentía

Para que esta práctica funcione, se requiere coraje del cliente y los desarrolladores. La agresividad es la actitud hacia la implementación del sistema es decir se cambia la estructura del código siempre que sea necesario. Tratar rápidamente con el cliente los desajustes de agendas para decidir qué partes y cuándo se va a entregar.

2.2.3.5 Respeto

Los miembros del equipo respetan su trabajo los unos a otros, porque siempre están luchando por la alta calidad en el producto, buscan el diseño óptimo y eficiente para la solución a través de la refactorización del código.

2.2.4 **Otras características fundamentales de XP**

- “Desarrollo iterativo e incremental que realiza pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas incluyendo pruebas de regresión.
- Programación en parejas, las tareas de desarrollo se llevan a cabo por dos personas, de esta manera el código es revisado y discutido mientras se escribe.” **Según** (Beck, K. , 2000)
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nuevas funcionalidades se debe hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad sin modificar su comportamiento

2.2.5 Fases de la Metodología

La Programación extrema se basa en prueba y errores para obtener un software funcional, además es orientada hacia quien produce (usuario o cliente), y usa el software ya que el cliente participa en todas las fases de la creación del software, además los requisitos pueden cambiar. En la integración de grupos pequeños ya sea de 2 o 12 personas que trabajan en parejas es necesario aplicar las siguientes etapas, tal y como se muestra en la figura 2:

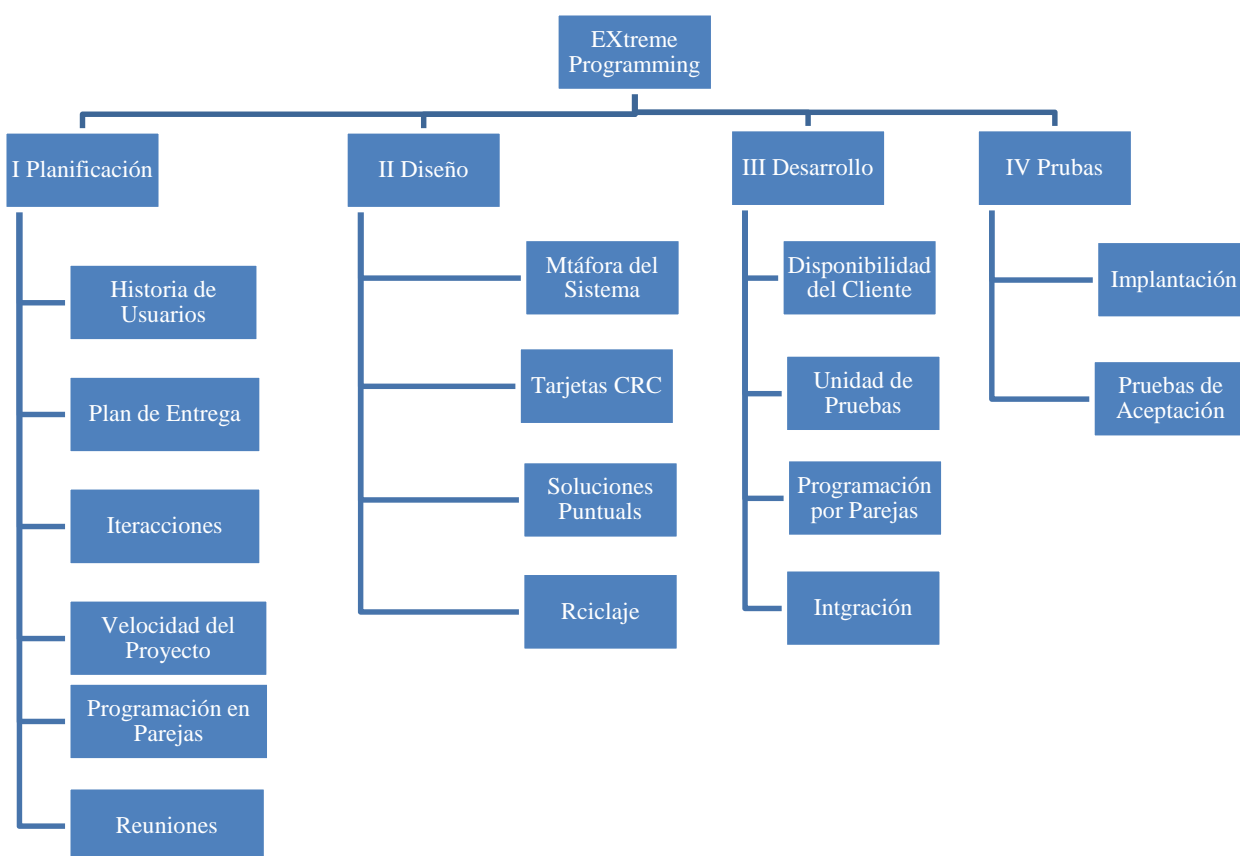


Figura 2. Fases de Extreme Programming

Tomado de (W. Quishpe, J. Tipan, 2011)

FASES DE LA METODOLOGIA XTREME PROGRAMMING

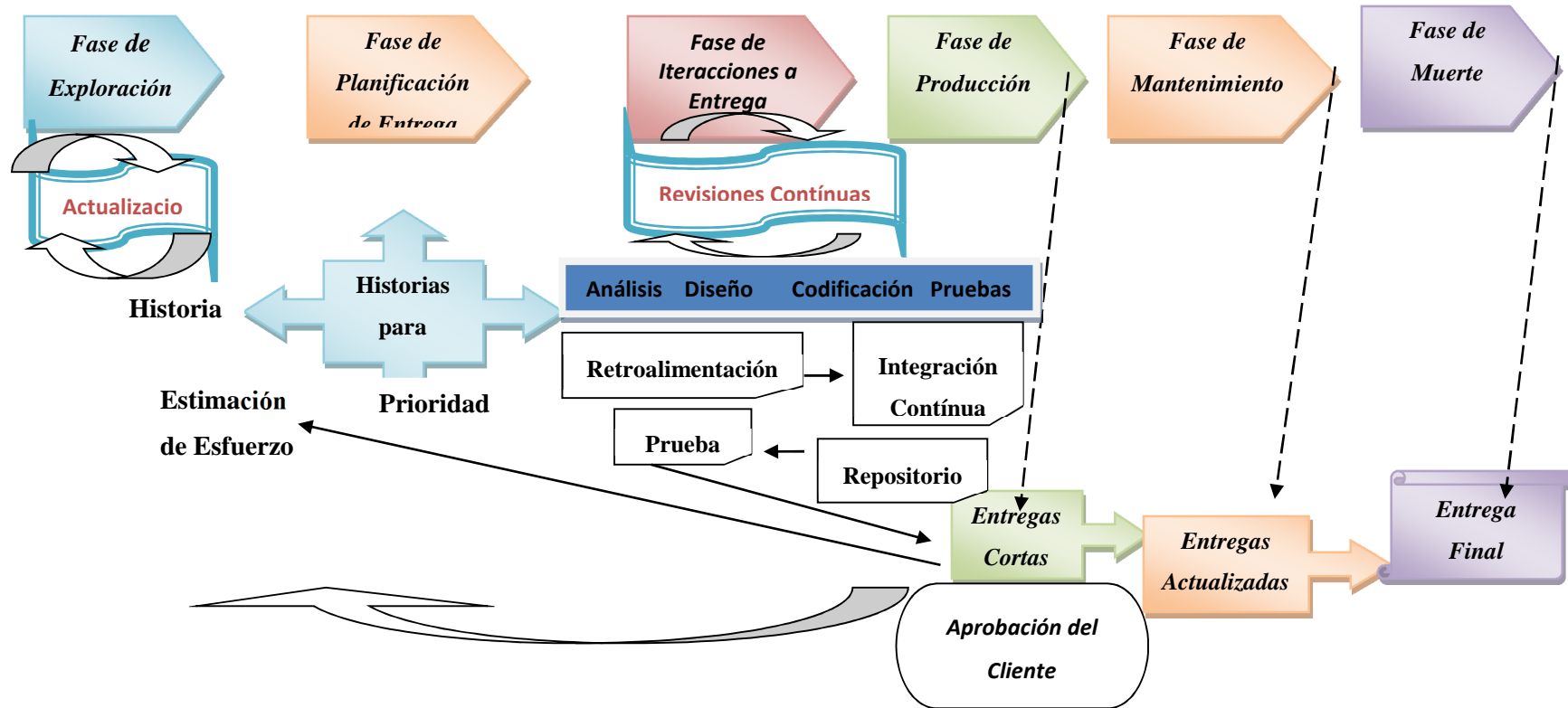


Figura 3. Fases de la Metodología de Xtreme Programming

Tomada de (W. Quishpe, J. Tipan, 2011)

2.2.5.1 Fase 1: Planificación del proyecto.

En esta fase el cliente establece la prioridad de cada historia de usuario, los programadores realizan una estimación del esfuerzo necesario de cada historia, determinando un cronograma de la primera entrega la cual deberá obtenerse en no más de tres meses. Esta fase dura unos pocos días.

La planificación se puede realizar basándose en el alcance del proyecto y el tiempo. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar.

Al planificar según alcance del sistema, se divide la suma de puntos de las historias de Usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

2.2.5.1.1 Historias de usuario.-

Las historias de usuario son escritas por el cliente y tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucha insistencia en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo, es también utilizado en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

2.2.5.1.2 Plan de Entrega (Release plan).-

Una vez que se tienen definidas las historias de usuario es necesario una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales, la prioridad con la que serán implementadas en cada versión del programa.

Después de un "Release plan" tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir, el tiempo que tardarán en desarrollarse y publicarse

las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado.

2.2.5.1.3 Iteraciones (*Iteration Plan*)

Todo proyecto que siga la metodología X.P. se debe dividir en iteraciones de aproximadamente 3 semanas de duración. Al inicio de cada iteración los clientes deben seleccionar las necesidades de usuario definidas en el "Release planning" mismas que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior.

Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

2.2.5.1.4 Velocidad del proyecto

Representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias del usuario que se pueden implementar en una iteración; de esta forma, se sabrá el número de historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto se controlará que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Plan" o libración de la planificación.

2.2.5.1.5 Programación en pareja

La programación en parejas incrementa el rendimiento y la calidad del software desarrollado. El trabajo en pareja involucra trabajo en equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño eficiente y de gran calidad.

2.2.5.1.6 Reuniones diarias (*Stant Up Meeting*)

Es necesario que se expongan los problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo debe tener voz y voto esto inmiscuye a desarrolladores y usuarios.

2.2.5.2 Fase 2: Diseño.

La metodología XP hace esencial énfasis en los diseños claros y simples además de la retroalimentación. Los conceptos más importantes de diseño en esta metodología son los siguientes:

2.2.5.2.1 Diseños simples

Es necesario conseguir un diseño fácilmente entendible e implementable que a la larga cueste menos tiempo y esfuerzo a la hora de desarrollar un software.

2.2.5.2.2 Glosarios de términos

Usar una correcta especificación en los nombres de métodos y clases, ayudará a comprender el diseño y proporcionará la reutilización del código del software.

2.2.5.2.3 Riesgos

De haber inconvenientes durante el diseño, es necesario utilizar una pareja de desarrolladores que se encarguen de investigar y reducir ese problema.

2.2.5.2.4 Funcionalidad extra

No se deben añadir funcionalidades extras al programa aunque crea que en un futuro será utilizada. Es importante tomar en consideración que sólo el 10% de la misma se utilizara, lo que implica que el desarrollo funcional extra es un desperdicio de tiempo y recursos a la larga.

2.2.5.2.5 Refactorizar

Es necesario mejorar y modificar la estructura de la codificación ya creada sin alterar la funcionalidad. Debe revisar códigos para procurar optimizar el funcionamiento.

2.2.5.2.6 Tarjetas C.R.C.

El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse en el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica

Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad, tal y como se muestra en la figura 4.

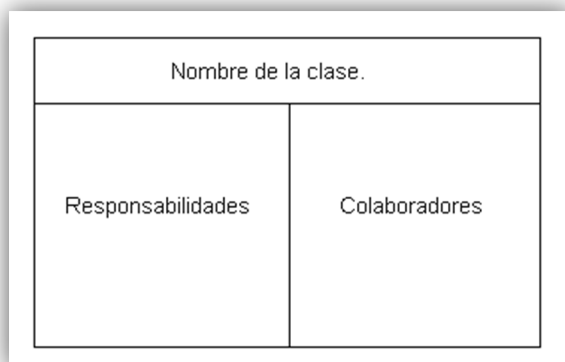


Figura 4. Modelo de Tarjeta CRC

Autor: (H. Cervantes, 1990)

2.2.5.3 Fase 3: Codificación.

2.2.5.3.1 *Disponibilidad del Cliente*

El cliente es una parte muy importante del equipo de desarrollo; su presencia es indispensable en las distintas fases de X.P. ya que a la hora de codificar una historia de usuario su presencia es aún más necesaria.

Antes de desarrollar una historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple con la funcionalidad. La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

X.P sugiere un modelo de trabajo usando repositorios de código dónde las parejas de programadores publican sus códigos implementados y corregidos junto a los test que deben pasar, de esta forma los programadores que necesiten códigos trabajarán siempre con las últimas versiones.

2.2.5.3.2 *Programación dirigida por las pruebas*

Se basa en lograr un código limpio que funcione, la idea es que los requisitos sean traducidos a pruebas unitarias, de este modo, cuando las pruebas pasen se garantizará que el software cumple con los requisitos que se han establecido. Lo primero que se escribe son los test que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas.

2.2.5.3.3 Programación por Parejas

Al trabajar en pares se minimizan los errores y se logran mejores diseños. El producto obtenido es de mejor calidad.

2.2.5.3.4 Integraciones Permanentes

Es importante que los desarrolladores trabajen siempre con la última versión puesto que el realizar cambios o mejoras sobre versiones antiguas causan graves problemas, retrasan al proyecto y demandan más tiempo de desarrollo. Es por eso que XP promueve a la publicación de nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores, solo una pareja de desarrolladores puede integrar su código a la vez.

2.2.5.4 Fase 4: Pruebas.

Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que se vayan implementando en la fase de desarrollo en lugar de al final de la fase de desarrollo.

Se tienen las siguientes pruebas:

Pruebas unitarias

Todos los módulos deben pasar las pruebas unitarias antes de ser liberados o publicados. Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. (J. Joskowicz , 2008)

Detección y corrección de errores

Cuando se encuentra un error debe ser corregido inmediatamente, y se deben tener precauciones de que no vuelvan a ocurrir. Es importante volver a realizar pruebas para verificar que el error haya sido resuelto.

Pruebas de Aceptación

Otra prueba clave es las pruebas de aceptación de los clientes en cuanto a su funcionalidad del software.

2.2.5.5 Fase 5: Producción

La fase de producción requiere de pruebas adicionales además de revisiones de rendimiento antes de que el sistema sea entregado al cliente.

Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que el tiempo que toma cada iteración sea más corto, de tres a una semana. Es necesario documentar las ideas que han sido propuestas y las sugerencias para su posterior implementación

2.2.5.6 Fase 6: Mantenimiento

Una vez que la primera versión se encuentre en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto es necesario del soporte para el cliente.

La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

2.2.5.7 Fase 7: Muerte del Proyecto

Cuando el cliente ya no tiene más historias para ser incluidas en el Sistema y a la vez el usuario está satisfecho. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

2.3 Open Source

2.3.1 Definición

El software Open Source garantiza a cualquier persona el derecho de usar, modificar y redistribuir el código libremente.

Código abierto (en ingleses open source) es el término con el que se conoce al software distribuido y de desarrollo libre. El Código Abierto permite a los programadores leer, modificar y redistribuir el código fuente de un programa a la brevedad posible. La licencia de código abierto permite explícitamente:

- Utilizar un programa para cualquier propósito sin limitaciones.
- Distribuir copias del programa (no se paga por la licencia).

El modelo de desarrollo del software de código abierto se basa en el conocimiento compartido hacia otras personas así se obtendrá como gran ejemplo, el Internet y otras herramientas que lo hicieron posible (tcp/ip, UNIX, C).

2.3.2 Características de Open Source

- Fiabilidad y seguridad del producto resultante, es más confiable y efectivo por el trabajo simultáneo de los programadores además de la corrección de errores por parte de los programadores.
- Rapidez de desarrollo debido a la variedad de herramientas y librerías disponibles.
- Relación con el usuario: el programador puede definir mejor las necesidades del cliente.
- Libre: es de libre distribución y sin restricciones.

2.3.3 Formas de Obtener Software Libre

A través de copias en CD, revistas especializadas, o comprándolos en tiendas de computación, etc.

A través de Internet: por medio de FTP, sitios Web, canales de chat, foros de noticias, programas de intercambio de archivos, etc.

2.3.4 Ventajas

- El usuario no paga por la licencia de uso del programa.
- El proveedor cobra los servicios que presta.
- Al disponer del código fuente, cualquier persona puede ofrecer soporte, desarrollo u otro tipo de servicio para el software.
- Los datos generados son accesibles, sin obligar al cliente a invertir en licencias.

2.3.5 Desventajas

- No existen garantías.
- La curva de aprendizaje en estos ambientes tiene una pendiente elevada.
- Existencia de poca documentación o documentación confusa.
- El costo de capacitación es muy alto.
- Su licencia no se puede vender en costos altos por lo que es gratis.

2.4 El Lenguaje Unificado de Modelado (UML)

En todas las disciplinas de la Ingeniería son importantes los modelos ya que describen el aspecto y conducta de "*algo*". Ese "*algo*" puede existir, estar desarrollado o recién en planeación. Es este el momento donde los diseñadores, pueden incluir áreas tales como funcionalidad, performance y confiabilidad.

El modelado sirve para grandes y pequeños sistemas, sin embargo entre más grande y más complejo es el sistema, es importante utilizar modelado por una simple razón: "El hombre hace modelos de sistemas complejos porque no puede entenderlos en su totalidad".

2.4.1 Definición de UML

UML es el lenguaje de modelado de sistemas de software, respaldado por el OMG (Object Management Group). Es un lenguaje gráfico que permite visualizar, especificar, construir y documentar el desarrollo de software de un sistema en un reducido número de diagramas, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y esquemas de bases de datos.

UML es un "lenguaje de modelado" por símbolos utilizado por muchas metodologías, que describen métodos o procesos, es utilizado para definir un sistema, detallar artefactos en el sistema y para documentar.

Existen diferencias importantes entre un método y un lenguaje de modelado.

Un *método* es una forma explícita de estructurar el pensamiento y acciones de cada individuo, además dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

Un modelo es expresado en un *lenguaje de modelado* que consiste de vistas, diagramas, símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas, tal y como se muestra en la figura 5.

UML

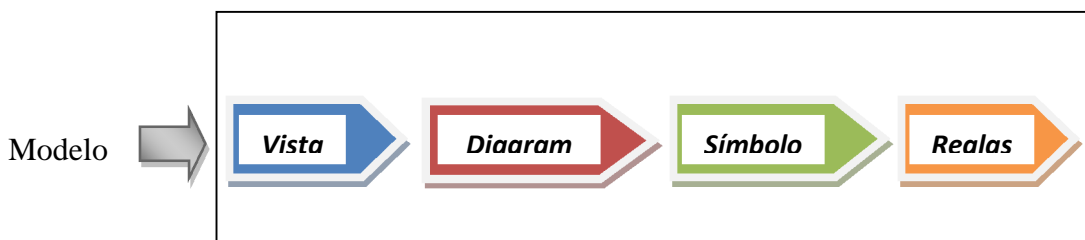


Figura 5. Modelo UML

2.4.1.1 Vistas:

Es una abstracción de un número de diagramas que muestran una forma completa del sistema. Las diferentes vistas que UML tiene son:

- **Vista Use-Case:** Muestra la funcionalidad del sistema como la perciben los actores externos.
- **Vista Lógica:** Muestra cómo se diseña la funcionalidad dentro del sistema, en términos de la estructura estática y la conducta dinámica del sistema.
- **Vista de Componentes:** Muestra la organización de los componentes de código.
- **Vista Concurrente:** Muestra la concurrencia en el sistema, direccionando los problemas con la comunicación y sincronización que están presentes en un sistema concurrente.
- **Vista de Distribución:** muestra la distribución del sistema en la arquitectura física con computadoras y dispositivos llamados *nodos*.

2.4.1.2 Diagramas:

Son gráficas que describen el contenido de una vista. UML tiene nueve tipos de diagramas que son utilizados para proveer todas las vistas de un sistema.

UML ofrece nueve diagramas:

- Diagramas de Casos de Uso para modelar los procesos
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado modela el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema

2.4.1.2.1 Casos de Uso.-

Es una estructura que ayuda a trabajar con los Usuarios determinando la forma en que se usara un software además de definir los requerimientos. Describen qué es lo que debe hacer el sistema, pero no cómo. Los actores y casos de uso son descritos en un diagrama use-case. Cada caso de uso es definido en texto este especifica los requerimientos del cliente: lo que él (o ella) espera del sistema sin considerar la funcionalidad que se implementará, tal y como se muestra en la figura 6 y 7.

Ejemplo: Como usar un producto de modo que cumpla con las necesidades requeridas.



Figura 6. Casos de Uso

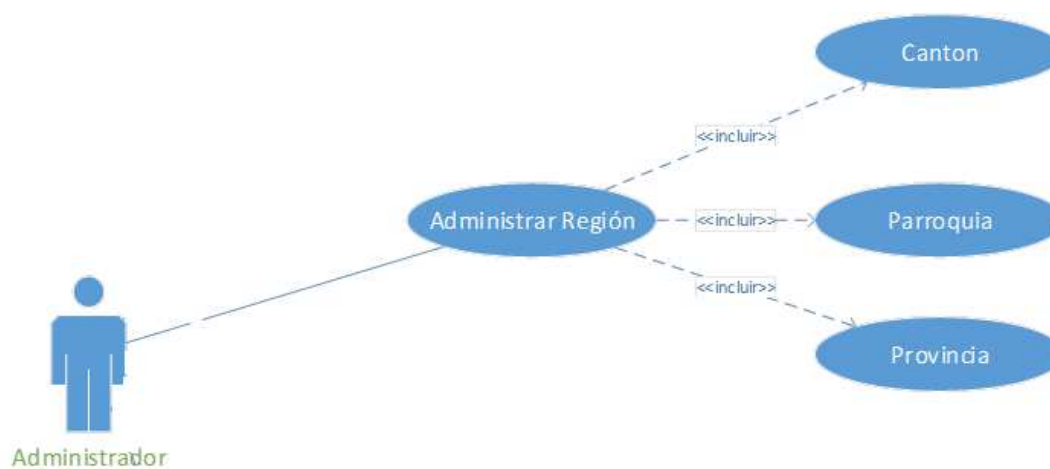


Figura 7. Casos de Uso

2.4.1.2.2 Diagrama de Secuencia

Muestra el intercambio de mensajes en un momento dado y en el tiempo de los distintos flujos de una aplicación. En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta.

El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros, tal y como se muestra en la figura 8.

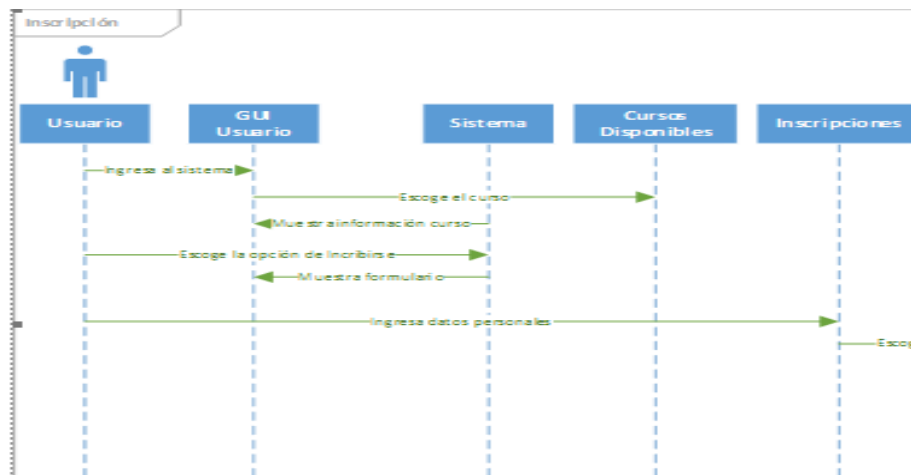


Figura 8. Diagramas de Secuencia

Mensaje simple: se usa cuando se transfiere el control de un objeto a otro.

Mensaje Síncrono: el objeto espera la respuesta al mensaje antes de continuar su flujo.



Mensaje asíncrono: El flujo continúa sin esperar la respuesta al mensaje enviado.

Se debe diagramar la secuencia de un caso de uso

2.4.1.2.3 Diagrama de Colaboración

Describe la colaboración o iteraciones entre un conjunto de objetos

Representación de los objetos

- Una instancia de la clase, tal y como se muestra en la figura 9.

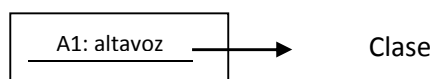


Figura 9. Representación de una instancia de la Clase

- Múltiples instancias de la clase, tal y como se muestra en la figura 10.



Figura 10. Representación de Múltiples instancias de la Clase

- Los mensajes representan el Cliente (Envía mensaje) Servidor (Recibe Mensaje) entre los objetos, tal y como se muestra en la figura 11.

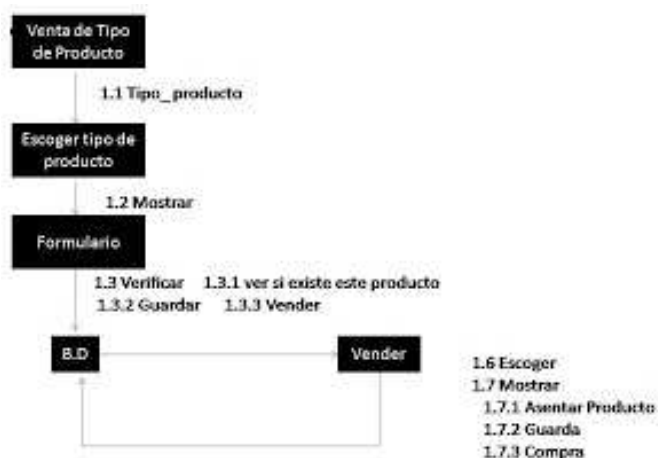


Figura 11. Diagrama de Colaboración

Autor: (Luis, 2009)

2.4.1.2.4 Diagrama de Estados

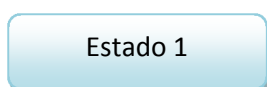
Expresa los posibles estados de una empresa.

Refleja el estado de un objeto en un momento del tiempo, su comportamiento y transacciones entre estados.

Permite representar el ciclo de vida de los objetos, contienen: Los estados, las transacciones, los eventos, las operaciones, acciones y actividades

Estados

Estado Genérico



Estado Inicial

Estado inicial



Estado Final

Estado Final



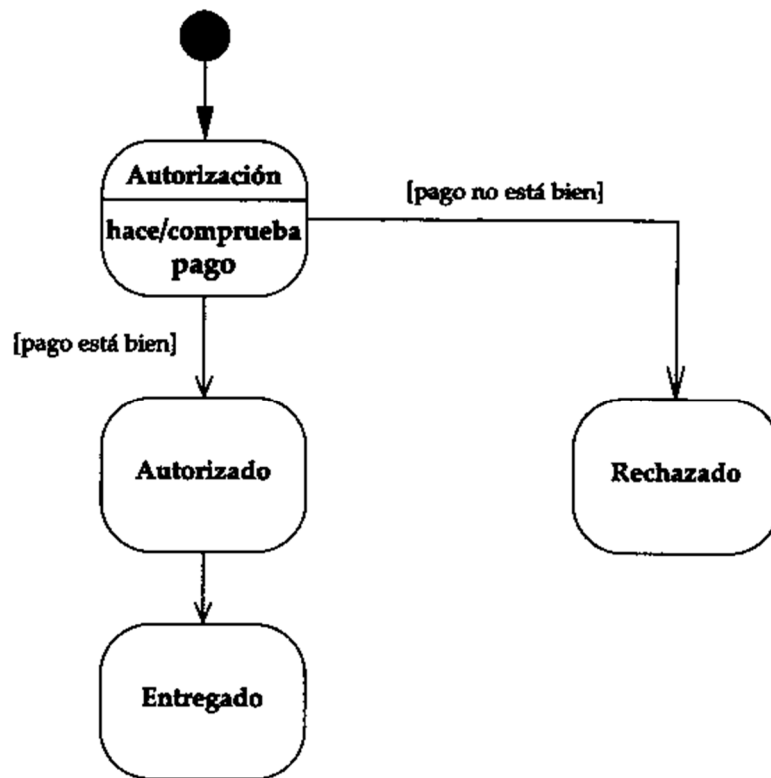


Figura 12. Diagrama de Estados

Autor: (Longman, 1999)

2.4.1.2.5 Diagrama de Actividad

Muestra una visión simplificada de lo que ocurre durante una operación o proceso de un sistema, contienen actividades tal como se muestra en la figura 13.

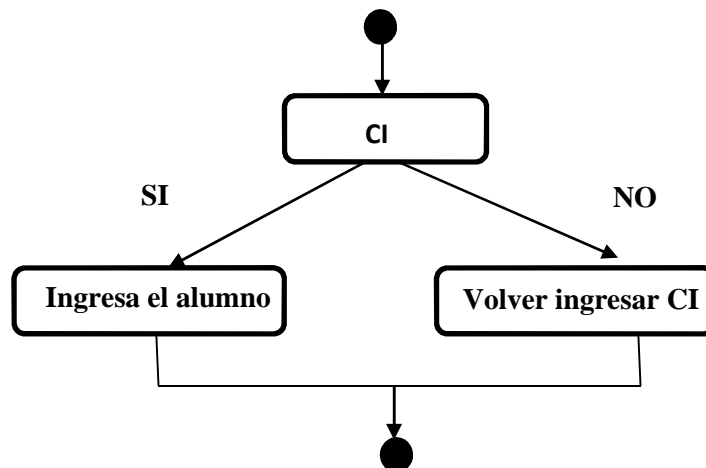


Figura 13. Diagrama de Actividad

2.4.1.2.6 Diagrama de Clases

Se debe pensar en las cosas que nos rodean las mismas que deben poseer atributos para que realicen diferentes acciones, tal y como se muestra en la figura 14.

Una clase tiene atributos y acciones

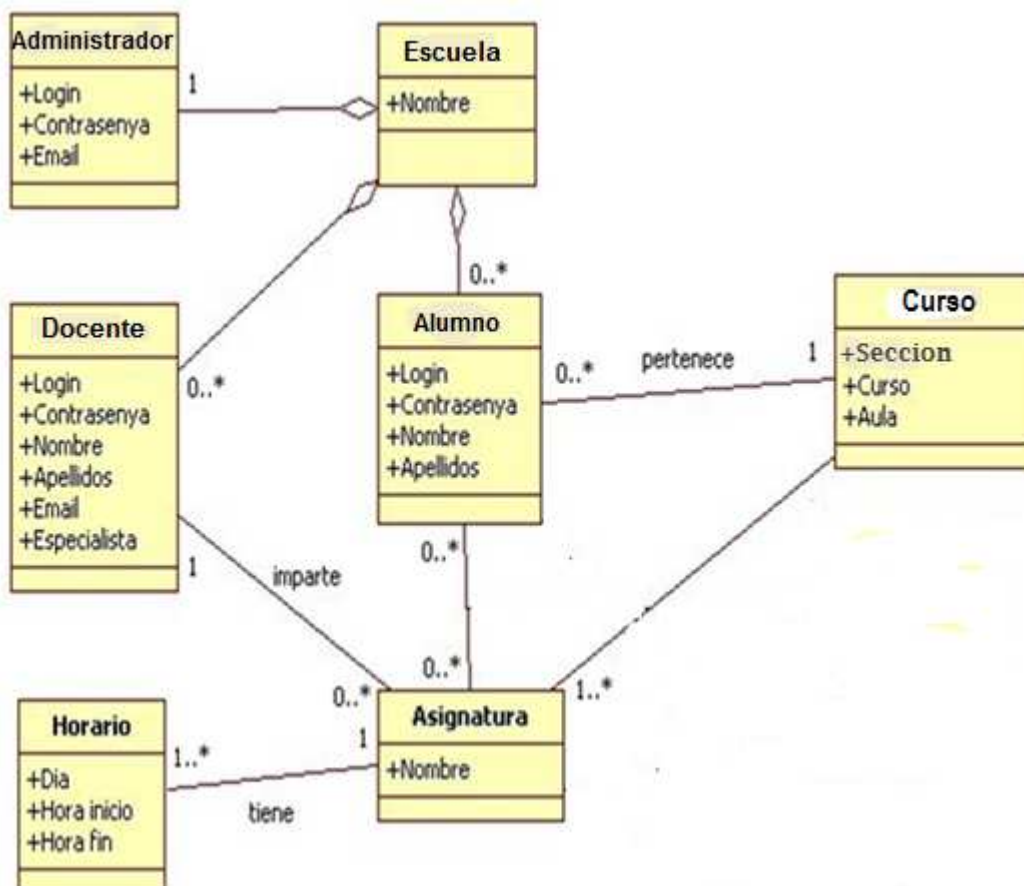


Figura 14. Diagrama de Clases

2.4.1.2.7 Diagrama de Objetos

Un objeto es una instancia de una clase, tal y como se muestra en la figura 15



Figura 15. Diagrama de Objetos

2.4.1.2.8 Diagrama de Componentes

Es la parte física de un sistema, se encuentra en el PC, no en la mente del analista: una tabla, un archivo, un ejecutable, una biblioteca, etc, tal y como se muestra en la figura 16

- Los clientes pueden ver la estructura del sistema finalizado.
- Los desarrolladores contarán con una estructura con la cual trabajar.
- Se logra generar código reutilizable.

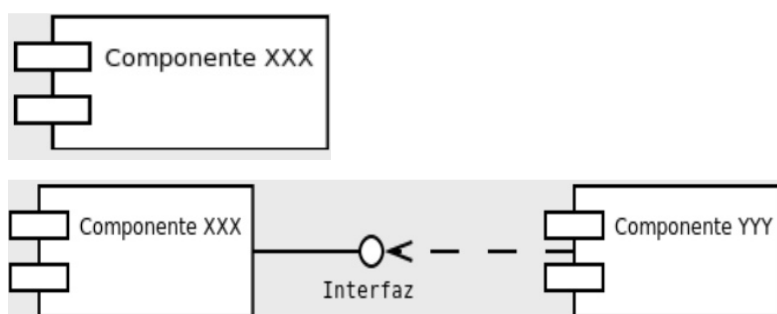


Figura 16. Diagrama de Componentes

2.4.1.2.9 Diagrama de Implementación

Es la representación esquemática de componentes físicos y sus conexiones: comunicación entre servidores, sistemas distribuidos, redes, etc, tal y como se muestra en la figura 17.

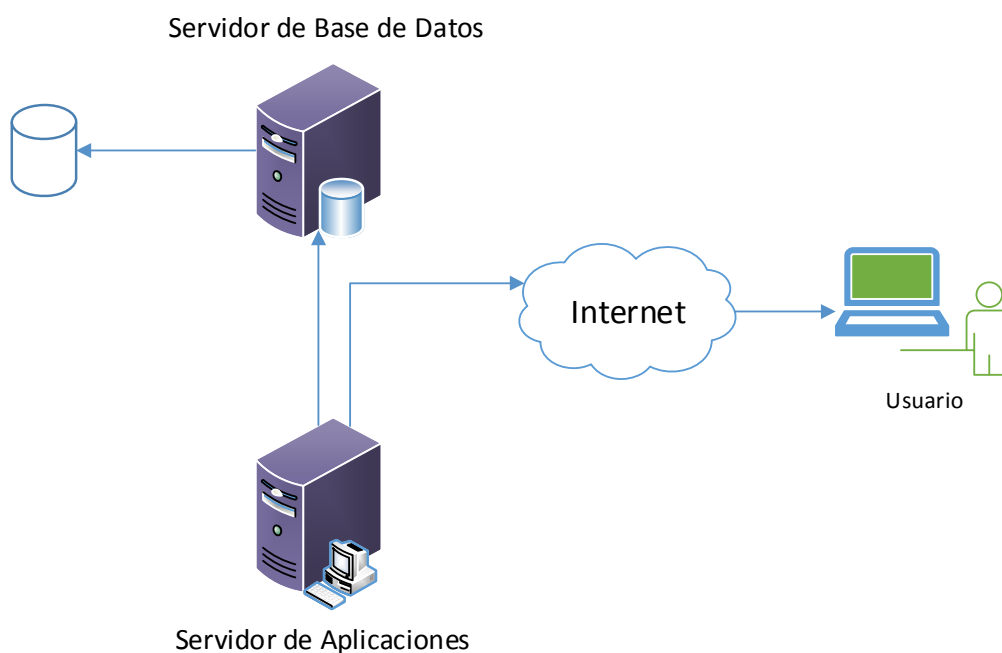


Figura 17: Diagrama de Implementación

2.4.1.3 Símbolos o Elementos de modelo:

Los conceptos utilizados en los diagramas son los elementos de modelo que representan conceptos comunes orientados a objetos, tales como clases, objetos y mensajes, y las relaciones entre estos conceptos incluyendo la asociación, dependencia y generalización. Un elemento de modelo es utilizado en varios diagramas diferentes, pero siempre tiene el mismo significado y simbología.

2.4.1.4 Reglas o Mecanismos generales:

Proveen comentarios extras, información o semántica acerca del elemento de modelo; además proveen mecanismos de extensión para adaptar o extender UML a un método o proceso específico, organización o usuario.

2.4.2 Fases del Desarrollo de un Sistema

Las fases del desarrollo de sistemas que soporta UML son: *Análisis, Diseño, Programación y Pruebas*.

2.4.2.1 Análisis

En la fase de análisis se da prioridad al conocimiento de los requerimientos, los conceptos y las operaciones relacionadas con el sistema, abarca las abstracciones primarias (clases y objetos) y mecanismos que están presentes en problema. Las clases que se modelan y sus relaciones son descritas en un diagrama de clases a través de modelos dinámicos UML, los artefactos que sirven para capturar los resultados de una investigación en esta primera fase son:

- Casos de Uso
- Modelo Conceptual
- Diagramas de Secuencia de un sistema

No se considera clases para interfaces de usuario, bases de datos, comunicaciones, concurrencia, etc.

2.4.2.2 Diseño

La fase de diseño, es el resultado del análisis de requerimientos. Aquí se agrega infraestructura técnica: interfaces de usuario, manejo de bases de datos para almacenar objetos en una base de datos, comunicaciones con otros sistemas, etc. Las clases de dominio del problema del análisis son agregadas en esta fase. El diseño resulta en especificaciones detalladas para la fase de programación.

2.4.2.3 Programación

En esta fase las clases del diseño son convertidas a código mediante un lenguaje de programación orientado a objetos. Cuando se crean los modelos de análisis y diseño en UML, lo más aconsejable es trasladar mentalmente esos modelos a código.

2.4.2.4 Pruebas

Las pruebas de aceptación conducidas por el cliente verifican que el sistema satisface los requerimientos y son similares a las pruebas de sistema.

2.5 Mapa Mental

2.5.1 Definición

El mapa mental fue desarrollado por el matemático y psicólogo británico Tony Buzan, para trabajar de forma asociativa con las ideas. Es un instrumento técnico y metodológico de diagrama o bosquejo que se desarrolla con la intención de manifestar conceptos o actividades vinculadas a una idea principal. La finalidad de estos mapas mentales es clasificar las ideas y facilitar su observación en un documento convirtiéndose en una herramienta útil para organizar datos y para estudiar un cierto tema facilitando su estudio, tal y como se muestra en la figura 18.

El uso de los mapas mentales cuenta con cinco elementos básicos.

1. “Idea principal que es simbolizada en una imagen central
2. Los temas principales que son los que emanan de la anterior a través de una serie de bifurcaciones,
3. Las imágenes o palabras clave que acompañan a las citadas bifurcaciones
4. Los temas menos importantes que ejercen como ramas,
5. Las bifurcaciones forman un entramado conectado.” (M. Davila, 2013)

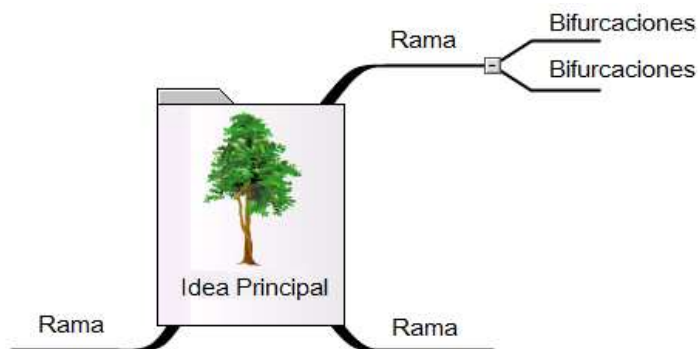


Figura 18: Mapa Mental

Los mapas mentales se utilizan a nivel empresarial para compartir ideas, mejorar la comunicación y organización aumentando la productividad además del ahorro de tiempo.

Los mapas mentales se utilizan a nivel educativo para facilitar el aprendizaje, permitiendo establecer visualmente el orden de la información, facilitando la comprensión y asimilación de las conexiones de las diversas ideas, tal y como se muestra en la figura 19.



Figura 19: Ideas de un Mapa Mental

2.5.2 Características de los Mapas Mentales

Para la construcción de mapas mentales es necesario:

- Utilizar ambos hemisferios del cerebro (el izquierdo y el derecho).
- Izquierdo: suministra material de información.
- Derecho: se usa para la representación gráfica del Mapa Mental.
- La combinación de los 2 hemisferios permite la estructuración de pensamientos, ya que une la información con aquello que se imagina.
- La creatividad.
- Recordar con mayor facilidad aspectos gráficos señalados en el mapa.
- Se pueden aplicar en todas las áreas del conocimiento.

2.5.3 Ventajas

- Las ideas son claras y precisas.
- Permite recordar con más efectividad las ideas sin mezclar las principales con las secundarias.
- Cada mapa mental es diferente, por cuanto en ellos se expresan ideas particulares con visiones diferentes.
- Es importante dejar siempre espacios abiertos dentro del mapa a fin de concatenar nuevas ideas.

2.6 J2EE

2.6.1 Definición

La plataforma J2EE (Java 2 Enterprise Edition) es una plataforma de programación diseñada por SUN, implementa aplicaciones corporativas multicapa basadas en la Web. Se apoya en el lenguaje Java beneficiándose de sus características ya que es un lenguaje sencillo orientado a objetos, seguro, portable y multitarea.

J2EE proporciona:

- Soporta tecnologías para internet, es un modelo de desarrollo de componentes Web (Servlet, JSP) y de componentes activos (EJB) bajo la forma de APIs de java.
- Un conjunto de servicios (JDBC, JTA, JNDI, JMS, RMI/IIop, Java mail, XML), herramientas para los componentes, bajo la forma de APIs de java.
- Un modelo de creación de módulos Web, EJB y corporativos, asociados a descriptores de despliegue en formato XML, herramientas para el desarrollo de aplicaciones de empresas.
- Contenedores web y EJB para la realización de los componentes.
- J2EE es una norma que permite a terceras personas desarrollar su propio servidor de aplicaciones.

2.6.2 Arquitectura de Aplicación Web J2EE

Se basa en los conceptos de *capas*, *containers*, *componentes*, *servicios*. Las aplicaciones J2EE se divide en 4 capas, la capa cliente, la capa web, la capa negocio y la capa datos, tal y como se muestra en la figura 20.

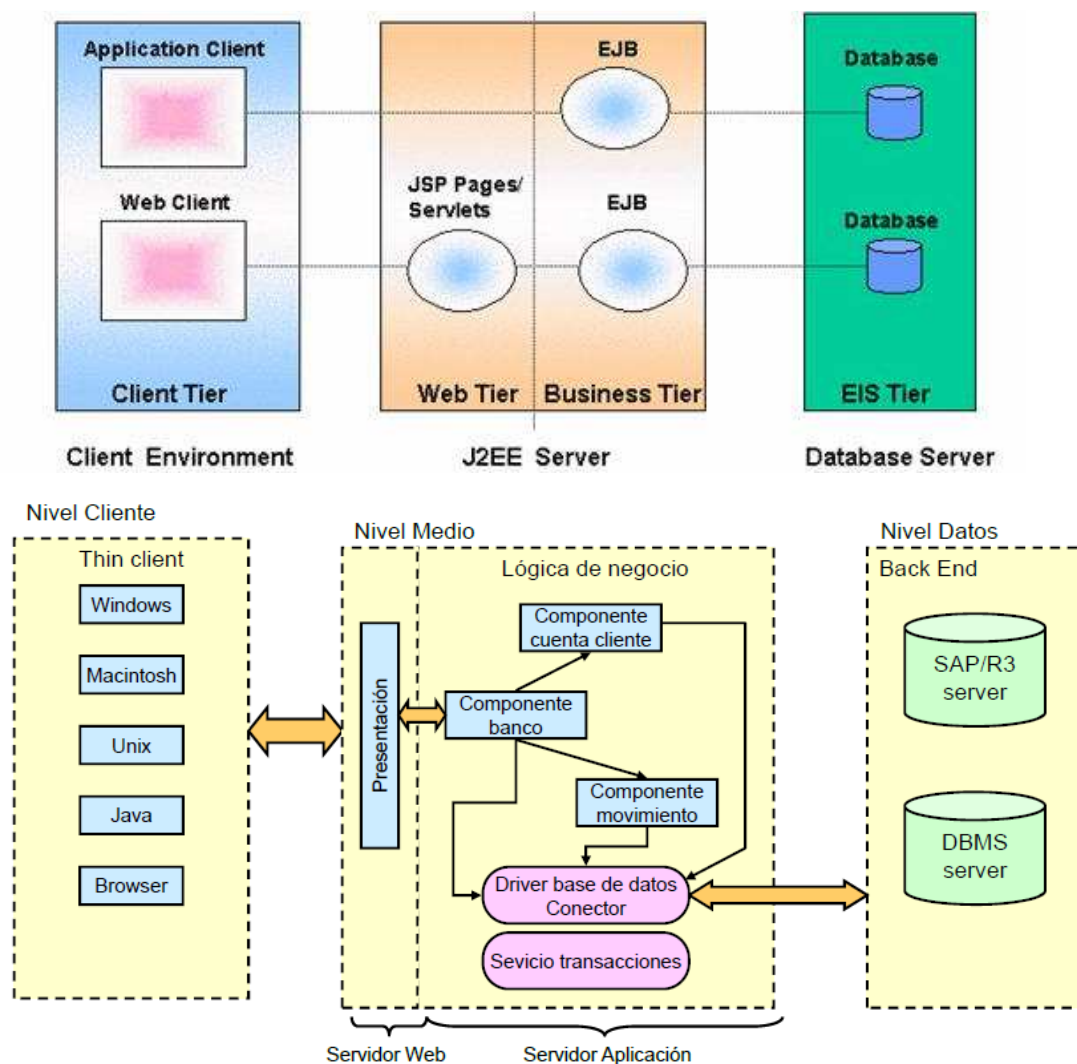


Figura 20. Representación de J2EE

Autor: (Galvis, 2008)

Capa Cliente.- Puede ser un programa, un applet dentro de un browser, el mismo que constata un serverlet, jsp o un web service. Es la interfaz gráfica del sistema y es aquella que interactúa con el usuario.

Capa Web.- Recibe los datos del usuario desde la capa cliente y basado en éstos genera una respuesta apropiada a la solicitud. Contiene las páginas html, los serverlets, JSP y los servicios

Capa de Aplicaciones.- Contiene y administra los servicios de los componentes EJB.

Contendor de Datos.- Se encarga de los sistemas de información de la empresa, incluye bases de datos, sistema de procesamiento datos, sistemas legados y sistemas de planificación de recursos.

2.7 Herramientas de Desarrollo

2.7.1 Net Beans

NetBeans es un IDE (Entorno de Desarrollo Integrado) libre y gratuito sin restricciones de uso para facilitar al máximo la programación, la prueba y la depuración de las aplicaciones que se desarrollan.

NetBeans utiliza como lenguaje de programación, Java, actualmente se tiene la versión 6.7, con esta versión se pueden crear aplicaciones de escritorio, páginas web y demás utilidades.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos que son un archivo Java, este contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo.

Las aplicaciones basadas en la plataforma NetBeans pueden ser usadas por otros desarrolladores de software debido a la independencia del desarrollo de los módulos.

El IDE NetBeans es un entorno de desarrollo integrado, es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas.

El NetBeans IDE está escrito en Java usa plataforma NetBeans y soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).

2.7.2 Maven

Maven es una herramienta de software para la gestión y construcción de proyectos Java y Java EE, se basa en un fichero central, pom.xml, donde se define todo lo que necesita el proyecto además, Maven maneja dependencias de proyectos, compila, empaqueta y ejecuta los test en un classpath caracterizándose porque está listo para usarse en red, tiene un modelo de configuración de construcción más simple, basado en un formato XML. Maven utiliza un Project ObjectModel (POM) que describe el proyecto de software a construir, dependen de otros módulos y componentes externos, y el orden de construcción de los elementos. El motor incluido en su núcleo puede dinámicamente descargar plugins de un repositorio, el mismo que provee acceso a muchas versiones de diferentes proyectos Open Source en Java, de Apache y otras organizaciones y desarrolladores.

El repositorio Maven 2, pugnan por ser el mecanismo de facto de distribución de aplicaciones en Java, pero su adaptación ha sido muy lenta.

Maven provee archivos de repositorio, también es utilizado para subir artefactos al repositorio final de la construcción de la aplicación, dejándolos al acceso de todos los usuarios. Una caché local de artefactos actúa como la primera fuente para sincronizar la salida de los proyectos a un sistema local.

Maven usa arquitectura basada en plugins que permite que sea utilizado por cualquier aplicación controlable a través de la entrada estándar. Actualmente existe un plugin para .Net Framework y es mantenido, y un plugin nativo para C/C++ fue alguna vez mantenido por Maven 1.

Integración con IDE.- Maven sustituye el entorno integrado de desarrollo IDE, por tanto la integración con diferentes IDEs es muy importante. Existen plugins de Maven para crear archivos de configuración del IDE a partir de los POMs. Actualmente se soportan:

- Eclipse
- Netbeans
- IntelliJ
- JDeveloper 11G (11.1.1.3)

Maven es una herramienta que compila proyectos Java, ejecutar pruebas unitarias, generar paquetes (jars, wars, ears o distribuciones en zip) y generar una serie de informes.

Existen plugins creados por maven y plugins de terceros, como los disponibles en freehep. Algunos de ellos son:

- Jar
- War
- Ear
- JUnit
- Distribution: permite ingresar en un fichero zip todos los archivos del proyecto
- PDF: genera documentación del proyecto en PDF.
- Checkstyle: control de convenciones de codificación Java
- PMD: verifica reglas y las codifica

2.7.2.1 Ciclo de vida

Las partes del ciclo de vida de Maven son:

1. Compile: Genera los ficheros .class compilando los fuentes .java
2. Test: Ejecuta ltest automáticos de JUnit, abortando el proceso si alguno de ellos falla.
3. Package: Genera el fichero .jar con los .class compilados
4. Install: Copia el fichero .jar a un directorio del ordenador donde maven deja todos los .jar. De esta forma los .jar pueden utilizarse en otros proyectos maven en el mismo ordenador.
5. Deploy: Copia el fichero .jar a un servidor remoto, poniéndolo disponible para cualquier proyecto maven con acceso a ese servidor remoto.

2.7.3 Postgres

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional de código abierto, distribuido bajo licencia BSD, utiliza un modelo cliente/servidor además es multiproceso y con su código fuente libre garantiza la estabilidad y escalabilidad de un sistema. Un fallo en uno de los procesos no afectará al resto y el sistema continuará funcionando, tal y como se muestra en la figura 21.

Componentes de un sistema PostgreSQL

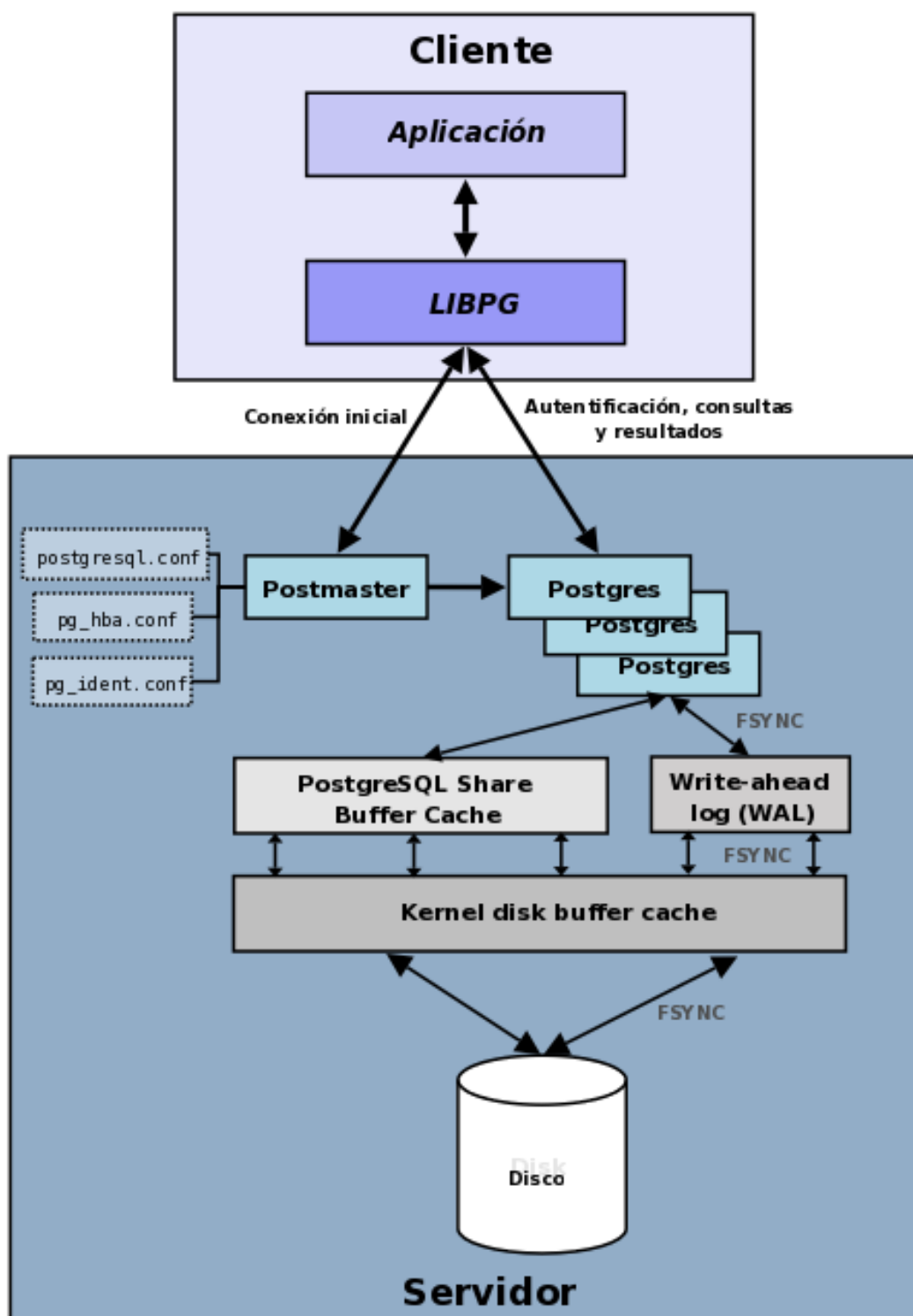


Figura 21. Postgress

Autor: (R. Martinez, 2009-2013)

- **Aplicación cliente:** utiliza PostgreSQL como administrador de bases de datos. La conexión puede ser mediante TCP/IP ó sockets locales.
- **Demonio postmaster:** Proceso inicial de PostgreSQL. Es el encargado de atender conexiones entrantes de clientes. También es el encargado de crear los procesos hijos.
- **Ficheros de configuración:** Los 3 ficheros principales de PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf
- **Procesos hijos postgres:** Se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO)
- **Kernel disk buffer cache:** Caché de disco del sistema operativo
- **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione

PostgreSQL es dirigido por una comunidad de desarrolladores que trabajan de forma libre, apoyados por organizaciones comerciales como PGDG (PostgreSQL Global DevelopmentGroup).

Se ejecuta en los principales sistemas operativos que existen en la actualidad como:

- Linux
- UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64)
- Windows

Además es compatible con ACID, soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C / C + +, Java, Punto Net, Perl, Python, Ruby, Tcl, ODBC, entre otros.

Postgress contiene una biblioteca de funciones integradas, van desde operaciones matemáticas básicas, operaciones con strings para criptografía y compatibilidad con Oracle. Del mismo modo, PostgreSQL incluye un framework que permite a los desarrolladores definir y crear sus propios tipos de datos personalizados.

2.7.3.1 Características Generales

1. “Es una base de datos 100% ACID.
2. Soporta distintos tipos de datos y permite la creación de tipos propios.
3. Incluye herencia entre tablas.
4. Copias de seguridad en caliente (Online/hotbackups).
5. Unicode.
6. Multi-VersionConcurrency Control (MVCC).
7. Múltiples métodos de autenticación.
8. Acceso encriptado via SSL.
9. SE-postgres.
10. Documentación Completa y extensa.
11. Licencia BSD.
12. Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.” (R. Martinez, 2009-2013)

2.7.3.2 Limitaciones

- Puntos de recuperación dentro de transacciones si se encuentra un fallo durante su ejecución.
- No soporta tablespaces.
- El soporte a orientación a objetos es una simple extensión que ofrece prestaciones como la herencia, no un soporte completo.

2.7.3.3 Ventajas

- Ideal para Tecnología Web
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Footprint bajo de memoria, bastante poderoso con una configuración adecuada.
- Multiplataforma.
- Capacidad de replicación de datos.
- Soporte disponible.

2.7.4 Power Designer

Es un conjunto de herramientas de Modelamiento que combina técnicas estándar de Modelamiento de aplicación a través de UML, técnicas de Modelamiento de Procesos Empresariales y técnicas tradicionales de Modelamiento de base de datos, tal y como se muestra en las figuras 22, 23, 24, 25, 26.

2.7.4.1 Técnicas de Modelamiento

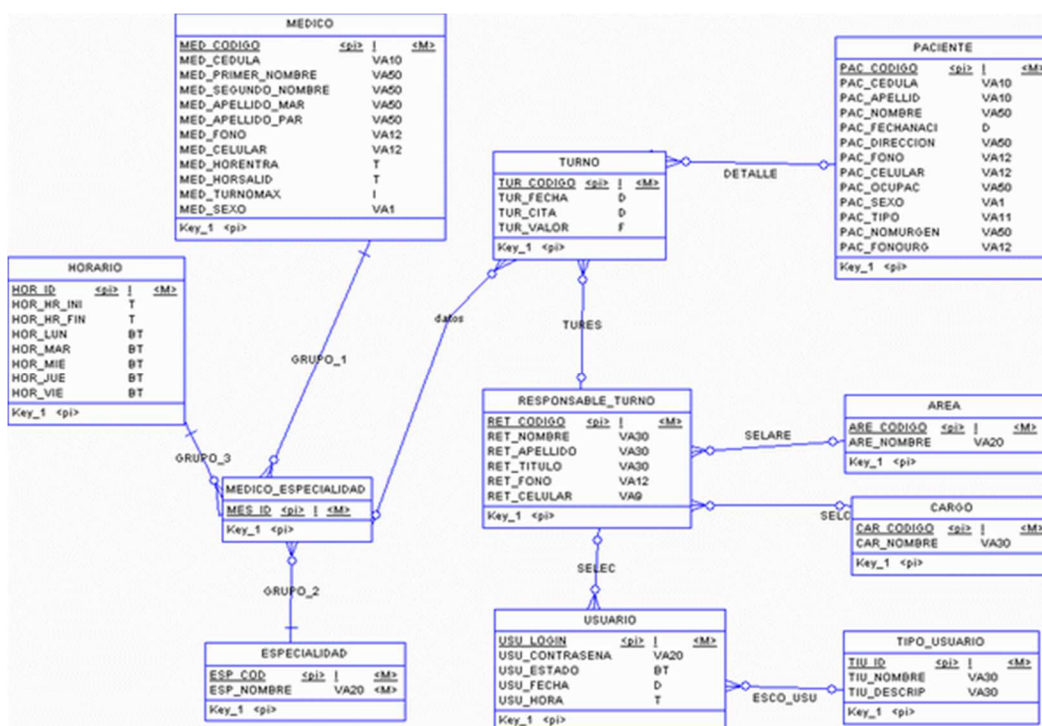


Figura 22. Modelo Conceptual

Autor: (R. Gonzalez, 2002)

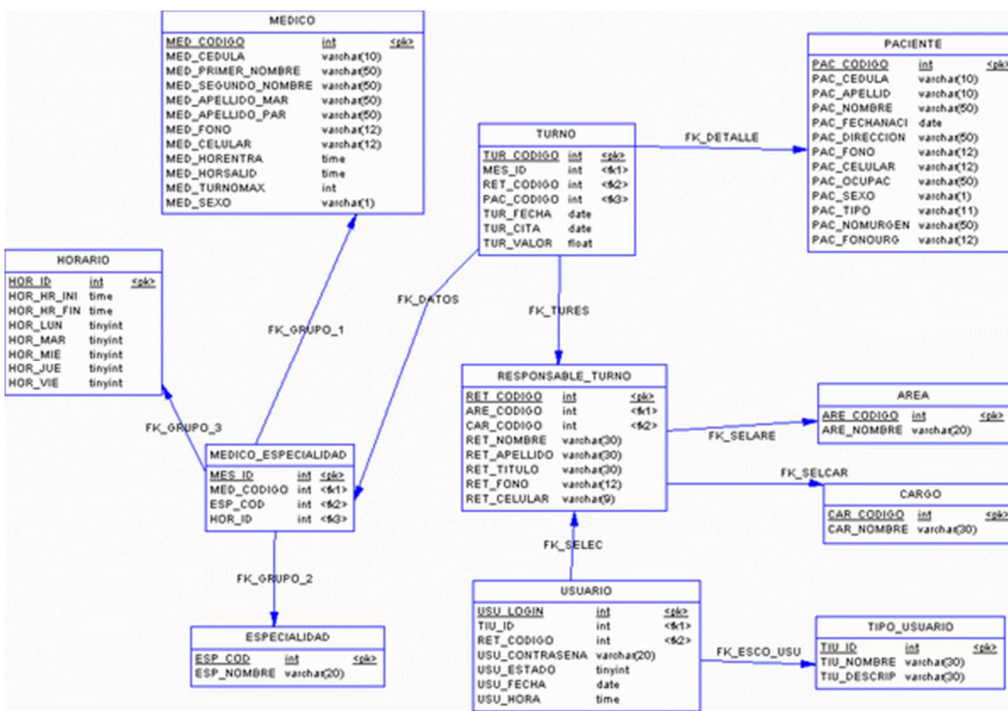


Figura 23. Modelo Físico

Autor: (R. Gonzalez, 2002)

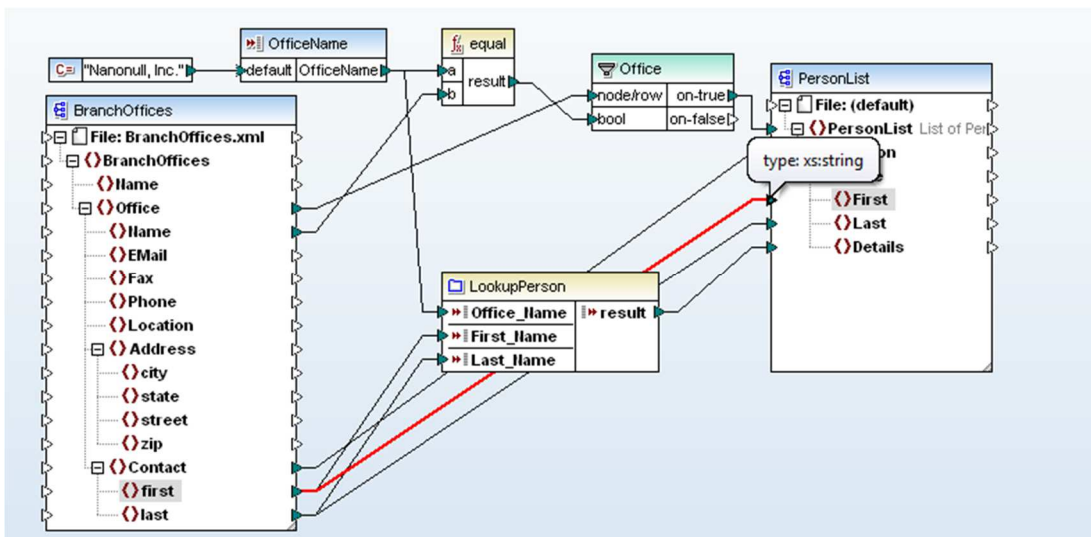


Figura 24. Data Mapping Editor

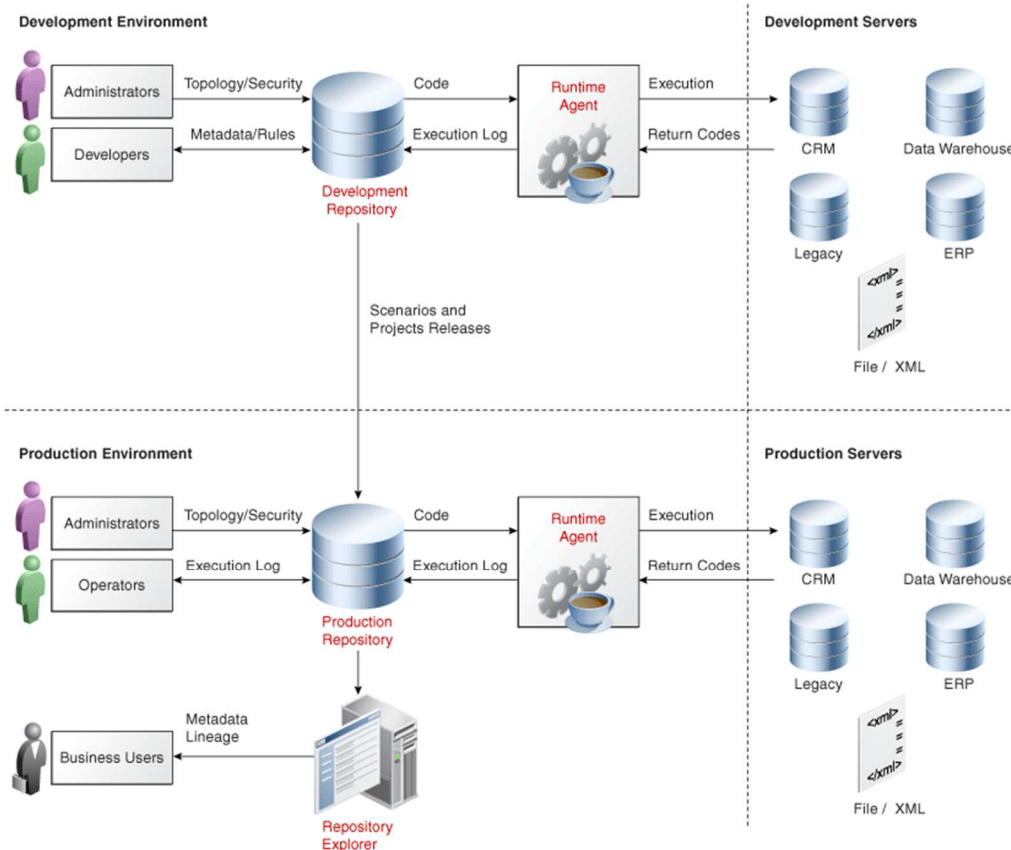


Figura 25. Data Movement Modeling

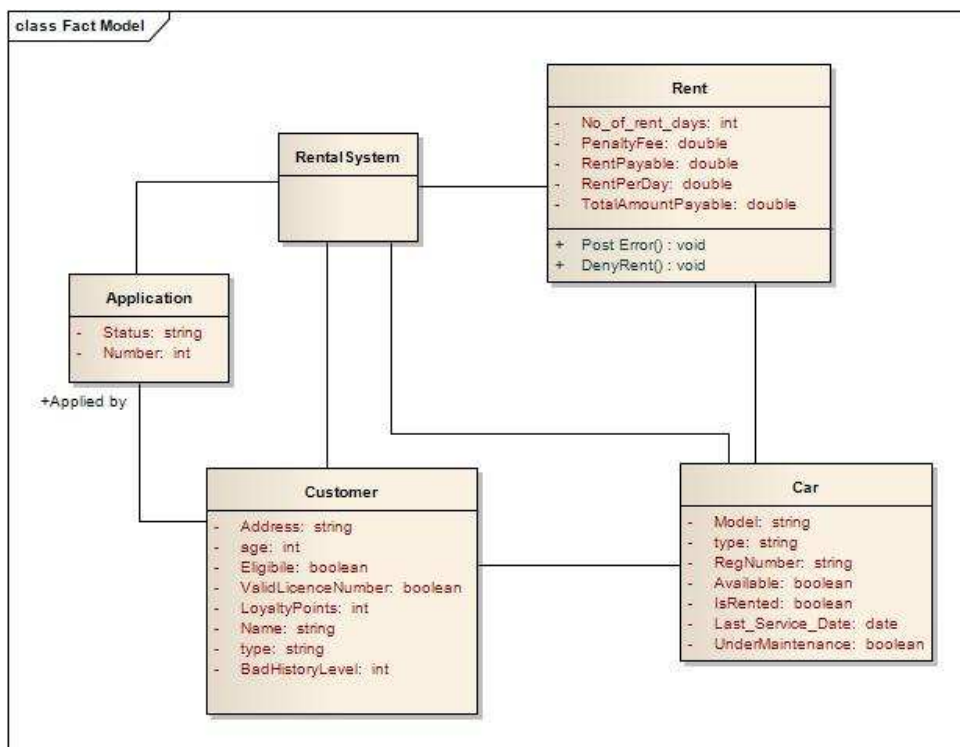


Figura 26. XML Modeling

Permite a las empresas una solución de modelado y diseño empresarial, brinda técnicas de análisis, diseño, visualización, y manipulación de metadatos durante todo el ciclo de vida del desarrollo del proyecto, logrando una efectiva arquitectura empresarial de información, además tiene funciones de análisis de impacto y repositorio que busca mejorar la colaboración y la comunicación entre todas las partes interesadas dentro de una empresa, consiguiendo facilitar una respuesta rápida a cualquier cambio en el proceso, y una mayor agilidad empresarial.

PowerDesigner combina varias técnicas de modelado con herramientas de desarrollo, como .NET, SybaseWorkSpace, Sybase Powerbuilder, Java y Eclipse, ofreciendo un servicio de solución completa de análisis de negocio, y de diseño formal de base de datos.

La última versión de PowerDesigner 12.0 ofrece innovaciones en Modelamiento de Procesos Empresariales, incluyendo soporte de simulación y procesamiento ejecutable de procesos de negocio, mayor integración con los ciclos de desarrollo de lenguaje como C# y VB.Net y técnicas más poderosas de generación inter-modelos, encadenamiento y sincronización.

2.7.4.2 Beneficios

- Alinea el negocio y la tecnología de información para mejorar la productividad.
- Brinda soporte abierto a ambientes heterogéneos de todas clases.
- Es altamente personalizable, permitiendo acogerse a los estándares y regulaciones.
- Brinda Facilidad de uso gráfica.
- Facilita la arquitectura empresarial, documentando los sistemas existentes.
- Aumenta la agilidad del negocio con “Link & Sync” y análisis de impacto.
- Reduce el impacto del cambio.

2.7.4.3 Ambientes en PowerDesigner:

2.7.4.3.1 Arquitectura de Datos

PowerDesigner integra múltiples capas de modelado de datos (lógicos y físicos) y las perspectivas únicas de almacenamiento (movimiento tridimensional y datos).

Permite:

- Modelos totalmente integrados.
- Técnicas de modelado para usuarios específicos.
- El repositorio empresarial es un sistema totalmente integrado, organizado por la elección de base de datos relacional.
- Apoyo abierto para todas las plataformas principales de desarrollo.

2.7.4.3.2 *Arquitectura Empresarial*

PowerDesigner permite a los usuarios de todos los grupos visualizar con claridad y aplicar cambios de forma rápida, fiable y predecible. Su enfoque de la AE es hacer frente a la toma de decisiones y necesidades de cada área de negocio distinta de una organización, proporcionándoles la información que necesitan, en el "lenguaje" que lo necesitan.

PowerDesigner permite a los usuarios eliminar información, aumentar la alineación y mejorar la agilidad del negocio para responder a los cambios en la economía, tecnología, competencia y reglamentos. Los enlaces y la tecnología de sincronización con el análisis del impacto en el Repositorio garantizan que los usuarios reúnen todas las conexiones en toda la empresa para proporcionar soporte de misión crítica de decisión.

PowerDesigner se integra para ofrecer el mismo producto. De hecho, cuando cada usuario está utilizando el mismo marco la información simplemente encaja mejor.

- Permite:
- Modelos totalmente integrados.
- Gestionar el impacto del cambio.
- El Repositorio Empresarial es una herramienta completamente integrada, organizado por la elección de base de datos relacional.
- Documentar toda la empresa.

2.7.4.3.3 *Arquitectura de la Información*

PowerDesigner ofrece el modelado completo de arquitectura de la información en un repositorio de cohesión, de metadatos integrados. Esta integración de los negocios y los metadatos gestionados de TI en una sola herramienta agiliza la gestión de las organizaciones de los datos, inteligencia empresarial, la integración y los esfuerzos de consolidación.

PowerDesigner soporta los cuatro componentes de la inteligencia de negocio (calidad, fiabilidad, integridad y agilidad) mediante la documentación, la integración, la arquitectura y la articulación del estado de los negocios hoy en día, y representa la visión de futuro de la empresa.

PowerDesigner reduce el tiempo de análisis de impacto, riesgo y costo asociado a cambios en toda la arquitectura de la información.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

3.1 Identificación de Requerimientos Funcionales

La identificación de los requerimientos funcionales permite definir las verdaderas necesidades informáticas del Centro de Transferencias de Tecnologías, permitiendo de esta manera una automatización de los procesos más importantes para impartir los cursos en los diferentes centros de capacitación a nivel nacional.

3.2 Descripción del Problema

El Centro de Transferencias de Tecnologías imparte cursos de capacitación a nivel nacional, para lo cual se planifican los cronogramas de clases, cursos y temarios, profesores y horarios de clases de forma manual.

Cuando un curso se encuentra planificado las inscripciones de los alumnos se la realiza de forma presencial en cada centro de capacitación, esto provoca que los alumnos se acerquen al centro de capacitación y registre su información con las secretarías, al finalizar la inscripción se procede con la generación del impreso con la información básico de estudiante y el curso.

El registro de los pagos se la realiza de forma manual y el control del mismo es llevado en archivos de Excel.

Cuando inician clases los profesores llevan el registro de asistencias y notas en archivos Excel, los cuales son enviados al Centro de Transferencias de Tecnologías matriz ubicado en la provincia de Pichincha, para que se determine si el alumno aprobó o reprobó el curso. Si el alumno aprobó el curso se procede con la generación del certificado.

El certificado de aprobación es generado y luego enviado al centro de capacitación que impartió el curso.

Todos estos trámites manuales provocan que el manejo de los datos de los alumnos, calificaciones y generación de certificados sea ineficiente, por lo que es necesario contar con un sistema informático que permite la automatización de estos procesos.

3.3 Propuesta de Solución

Debido a que el Centro de Transferencias de Tecnologías de Información planifica los cursos que se impartirán en los centros de capacitación a nivel nacional, es

necesario contar con una herramienta informática que permita centralizar la información que se genera en cada uno de estos centros al momento de impartir las capacitaciones.

Este sistema informático debe aprovechar el Internet para desplegar la información de los cursos disponibles para cada uno de los centros de capacitación a nivel nacional, y permitir que los alumnos puedan visualizar toda la información de cada uno de los cursos que se imparte, de igual manera el registro de inscripciones de los alumnos puede ser Online y generada por el mismo alumno, evitando de esta manera que las persona se acerquen al centro de capacitación y agilitando el proceso de inscripción.

Se llevar un registro de pagos para desplegar un estado financiero de los cursos dictados.

3.3.1 Casos de Uso

Se procede con la descripción de los casos de uso para el sistema informático del CTT.

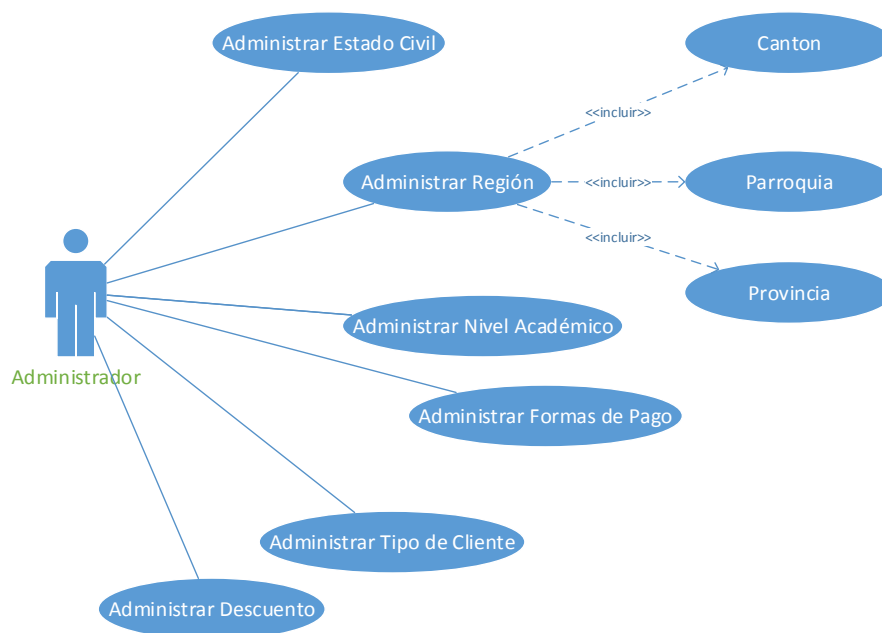


Figura 27. Diagrama de Casos de Uso de Parámetros del Sistema

Tabla 1**Administrador de Estado Civil**

Caso de Uso	
Nombre	Administrar Estado Civil
Código	CU-001
Actores	Administrador del Sistema
Descripción	
Se registra todos los tipos de estado civil.	
Precondiciones	Administrador debe estar logeado en el sistema.
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Crear el estado civil	Almacena en la base de datos el estado civil.
Editar el estado civil	Actualiza la información del estado civil en la base de datos.
Eliminar el estado civil	Elimina la información del estado civil en la base de datos.
Fallo Posible	Solución

Tabla 2

Administrar Región

Caso de Uso	
Nombre	Administrar Región
Código	CU-002
Actores	Administrador del Sistema
Descripción	
Se registra todas las regiones es decir las provincias con cada una de sus cantones y parroquias.	
Precondiciones	Administrador debe estar logeado en el sistema. Para crear los cantones deben estar ingresadas provincias. Para crear parroquias deben estar ingresadas provincias y cantones.
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Crear provincia	Almacena en la base de datos el nombre de la provincia.
Editar provincia	Actualiza la información de la provincia en la base de datos.
Eliminar provincia	Elimina la información de la provincia en la base de datos.
Crear cantón	Almacena en la base de datos la información del cantón.
Editar cantón	Actualiza la información del cantón en la base de datos.
Eliminar cantón	Elimina la información del cantón en la base de datos.
Crear parroquia	Almacena en la base de datos la información de la parroquia.
Editar parroquia	Actualiza la información de la parroquia en la base de datos.
Eliminar parroquia	Elimina la información de la parroquia en la base de datos.
Fallo Posible	Solución
No se puede eliminar la provincia	Borrar las parroquias y cantones asociados a la provincia que se desea eliminar.
No se puede eliminar el cantón	Borrar las parroquias asociados al cantón que se desea eliminar.

Tabla 3

Administrar Nivel Académico

Caso de Uso	
Nombre	Administrar Nivel Académico
Código	CU-003
Actores	Administrador del Sistema
Descripción	
Se registra todos los tipos de niveles académicos	
Precondiciones	Administrador debe estar logeado en el sistema.
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Crear nivel académico	Almacena en la base de datos el nivel académico.
Editar nivel académico	Actualiza la información del nivel académico en la base de datos.
Eliminar nivel académico	Elimina la información del nivel académico en la base de datos.
Fallo Posible	Solución

Tabla 4

Administrar Forma de Pago

Caso de Uso	
Nombre	Administrar Formas de Pago
Código	CU-004
Actores	Administrador del Sistema
Descripción	
Se registra todas las formas de pago existentes.	
Precondiciones	Administrador debe estar logeado en el sistema.
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Crear forma de pago	Almacena en la base de datos la forma de pago.
Editar forma de pago	Actualiza la información de la forma de pago en la base de datos.
Eliminar forma de pago	Elimina la información de la forma de pago en la base de datos.
Fallo Posible	Solución

Tabla 5**Administrar Tipo de Cliente**

Caso de Uso	
Nombre	Administrar Tipo de Cliente
Código	CU-005
Actores	Administrador del Sistema
Descripción	
Se registra todos los tipos de cliente existentes.	
Precondiciones	Administrador debe estar logeado en el sistema.
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Crear tipo de cliente	Almacena en la base de datos la información del tipo de cliente.
Editar tipo de cliente	Actualiza la información del tipo de cliente en la base de datos.
Eliminar tipo de cliente	Elimina la información del tipo de cliente en la base de datos.
Fallo Posible	Solución
No se lista el tipo de cliente	Al crear el tipo de cliente se debe activarlo en el estado.

Tabla 6

Administrar Documento

Caso de Uso	
Nombre	Administrar Descuento
Código	CU-006
Actores	Administrador del Sistema
Descripción	
Se registra todos los tipos de descuentos existentes.	
Precondiciones	CU-013, CU-005, CU-009
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Seleccionar el Centro de Capacitación al que desee agregar los descuentos.	Muestra el formulario de descuentos.
Ingresar el valor del descuento, las fechas de validez del mismo y escoger el tipo de cliente para el que es válido el descuento y por último agrega el descuento.	Almacena en la base de datos la información de los descuentos para el Centro de Capacitación y el tipo de cliente escogido.
Eliminar descuento	Elimina la información del descuento de la base de datos.
Fallo Posible	Solución
No se puede eliminar la información del descuento.	Eliminar previamente la información que esté relacionada con el descuento.

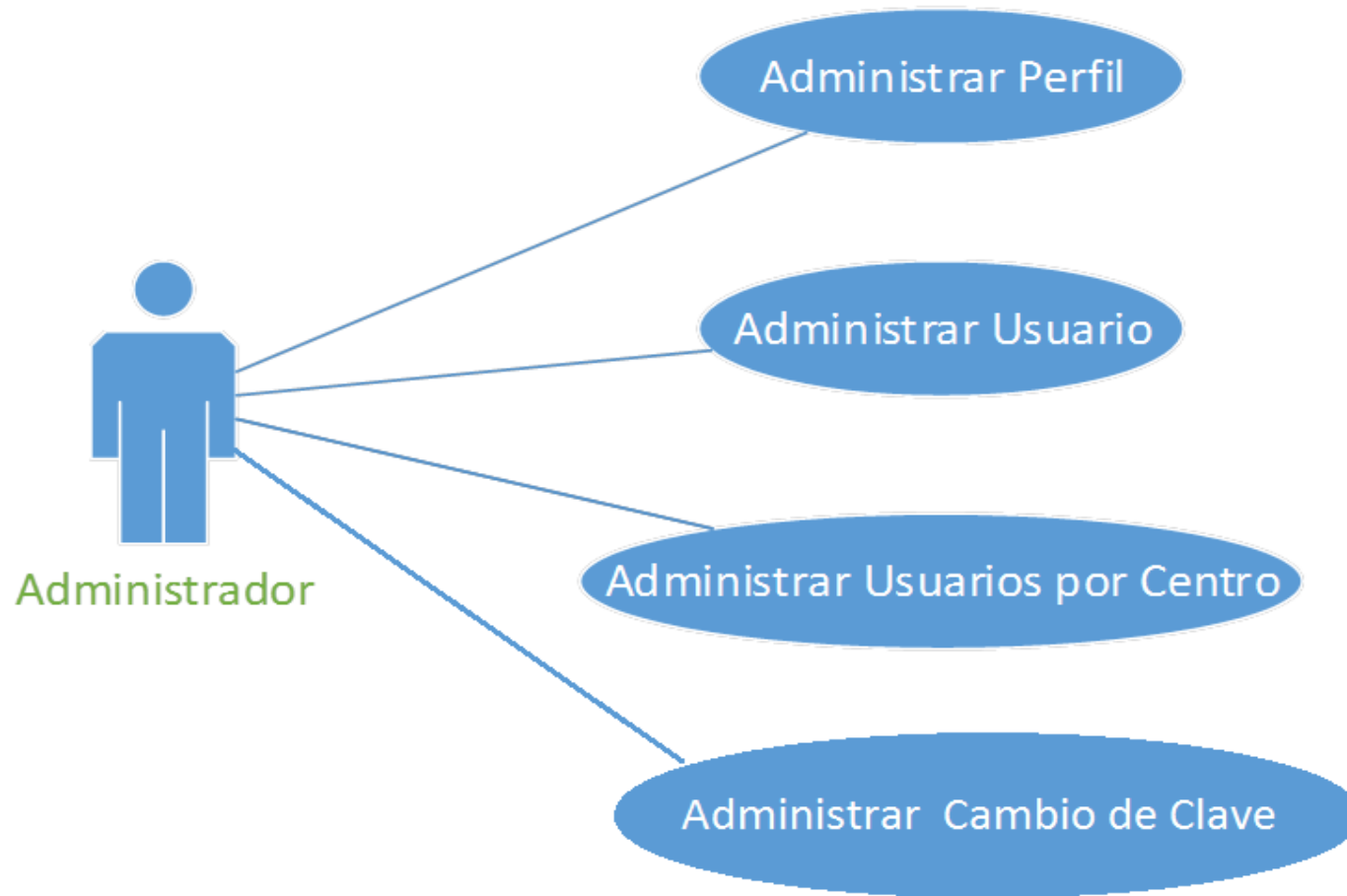


Figura 28. Diagrama de Casos de Uso de Seguridad del Sistema


Tabla 7

Administrar Perfil

Caso de Uso	
Nombre	Administrar Perfil
Código	CU-007
Actores	Administrador del Sistema
Descripción	
Se registra la información de los perfiles que se manejarán en el sistema.	
Precondiciones	N/A
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Crear perfil	Almacena en la base de datos la información del perfil.
Editar perfil	Actualiza la información del perfil.
Eliminar perfil	Elimina la información del perfil.
Fallo Posible	Solución
No se puede eliminar la información del perfil.	Eliminar previamente la información que esté relacionada con el perfil.

Tabla 8

Administrar Usuario

Caso de Uso	
Nombre	Administrar usuario
Código	CU-008
Actores	Administrador del Sistema
Descripción	
Se registra la información de los usuarios que tendrán acceso al sistema.	
Precondiciones	CU-007
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Crear usuario	Almacena en la base de datos la información del usuario. 

	Continúa
Editar usuario	Actualiza la información del usuario.
Eliminar usuario	Elimina la información del usuario.
Fallo Posible	Solución

Tabla 9

Administrar Usuario por Centro

Caso de Uso	
Nombre	Administrar usuarios por centro
Código	CU-009
Actores	Administrador del Sistema
Descripción	
Se asigna al usuario los Centros de Capacitación a los cuales tendrá acceso.	
Precondiciones	CU-013
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Seleccionar un Centro de Capacitación	Muestra el formulario de los usuarios que están disponibles para asignar y los usuarios asignados.
Asigna a un usuario disponible.	Almacena en la base de datos el centro de capacitación con su usuario y lo muestra en la lista de usuarios registrados.
Cambiar estado de la asignación	Actualiza la información de la asignación en la base de datos.
Eliminar asignación	Elimina de la base de datos la información de la asignación.
Cambiar centro	Muestra el listado de Centros de Capacitación para que el Administrador escoja el nuevo centro y se repite el proceso.
Fallo Posible	Solución

Tabla 10**Administrar Cambio de Clave**

Caso de Uso	
Nombre	Administrar cambio de clave
Código	CU-010
Actores	Administrador del Sistema
Descripción	
Se ingresa el número de cédula de usuario o profesor	
Precondiciones	CU-008, CU-012, CU21
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Ingresar Número de Cédula de Usuario	Muestra los datos del usuario, nombre, cédula, perfil, tipo de alumno y estado.
Selecciono el Usuario	Muestra los datos del usuario, correo, clave y estado.
Actualización de la Clave de Usuario	Actualiza la información de clave en la base de datos.
Ingresar Número de Cédula de Profesor	Muestra los datos del usuario, nombre, cédula, perfil, tipo de alumno y estado.
Selecciono el Profesor	Muestra los datos del usuario Nombre, cédula, perfil, tipo de alumno y estado.
Actualización de la Clave de Profesor	Actualiza la información de clave en la base de datos.
Ingresar Número de Cédula de Alumno	Muestra los datos del usuario, nombre, cédula, perfil, tipo de alumno y estado.
Selecciono el Alumno	Muestra los datos del usuario, correo, clave y estado.
Actualización de la Clave de Alumno	Actualiza la información de clave en la base de datos.
Fallo Posible	Solución



Figura 29. Diagrama de Casos de Uso de Estudios del Sistema

*Tabla 11**Administrar Curso*

Caso de Uso	
Nombre	Administrar curso
Código	CU-011
Actores	Administrador del Sistema
Descripción	
Registra la información de los cursos.	
Precondiciones	N/A
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Agregar Módulo	Muestra el formulario de agregar módulo.
Ingresa la información del módulo y agrega.	Almacena en la base de datos la información del módulo y muestra un listado de los módulos existentes.
Crear curso	Almacena en la base de datos la información del curso.
Editar curso	Actualiza la información del curso en la base de datos.
Eliminar curso	Elimina de la base de datos la información del curso.
Fallo Posible	Solución
No se puede eliminar el curso	Eliminar previamente la información que está asociada al curso.

*Tabla 12**Administrar Profesor*

Caso de Uso	
Nombre	Administrar profesor
Código	CU-012
Actores	Administrador del Sistema
Descripción	
Registra la información de los profesores.	
Precondiciones	CU-001, CU-003, CU-007
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Ingresar información del profesor	
Agregar estudios al profesor	Almacena en la base de datos la información de los estudios realizados por el profesor.
Crear profesor	Almacena en la base de datos la información del profesor.
Editar profesor	Actualiza la información del profesor en la base de datos.
Eliminar profesor	Elimina de la base de datos la información del profesor.
Fallo Posible	Solución
No se puede eliminar el profesor	Eliminar previamente la información que está asociada al profesor.

Tabla 13

Administrar Centro de Capacitación

Caso de Uso	
Nombre	Administrar centro de capacitación
Código	CU-013
Actores	Administrador del Sistema
Descripción	
Registra la información de los centros de capacitación.	
Precondiciones	CU-002
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Ingresar información del centro	
Agregar aulas	Muestra el formulario de las aulas, luego almacena la información del aula en la base de datos y muestra en el listado de aulas asignadas.
Eliminar aulas	Elimina información del aula asignada al centro.
Crear centro de capacitación	Almacena en la base de datos la información del centro.
Editar centro de capacitación	Actualiza la información del centro en la base de datos.
Eliminar centro de capacitación	Elimina de la base de datos la información del centro.
Fallo Posible	Solución
No se puede eliminar el centro de capacitación.	Eliminar previamente la información que está asociada al centro.

Tabla 14**Registrar Planificación**

Caso de Uso	
Nombre	Registrar Planificación
Código	CU-014
Actores	Administrador del Sistema
Descripción	
Registra la información de los cursos a ser impartidos en los diferentes centros de capacitación.	
Precondiciones	CU-011, CU-012, CU-013
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Seleccionar centro de capacitación a la cual se le asignará una de planificación de capacitación.	Se muestra la lista de los centros de capacitación asignado al usuario. Selecciona el centro de capacitación y se despliega el formulario para el ingreso de la información de la planificación de capacitación.
Seleccionar curso	Se muestra el listado de los cursos activos.
Seleccionar profesor	Se muestra el listado de profesores activos.
Seleccionar aula	Se muestra el lista de las aulas por centro de capacitación, se registra el aula y las fechas y horas en las que se impartirán clases.
Actualiza planificación	Se actualiza la información ingresada en registro de planificación.
Lista planificaciones	Se muestra el listado de las planificaciones registradas.
Fallo Posible	Solución

Tabla 15

Actualiza Planificación

Caso de Uso	
Nombre	Actualizar Planificación
Código	CU-015
Actores	Administrador del Sistema
Descripción	
Se actualiza la información de los cursos a ser impartidos en los diferentes centros de capacitación. Se registra las fechas de inicio y fin del curso, el profesor asignado, las aulas, las aulas planificadas y sus horarios d clases y el curso que se va impartir.	
Precondiciones	CU-014
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Seleccionar centro de capacitación a la cual se le asignará una planificación de capacitación.	Se muestra la lista de los centros de capacitación asignado al usuario. Selecciona el centro de capacitación y se despliega el formulario para el ingreso de la información de la planificación de capacitación.
Seleccionar curso	Se muestra el listado de los cursos activos.
Seleccionar profesor	Se muestra el listado de profesores activos.
Seleccionar aula	Se muestra el lista de las aulas por centro de capacitación, se registra el aula y las fechas y horas en las que se impartirán clases.
Actualiza planificación	Se actualiza la información ingresada en registro de planificación.
Lista planificaciones	Se muestra el listado de las planificaciones registradas.
Fallo Posible	Solución

Tabla 16**Activar Inscripción**

Caso de Uso	
Nombre	Activar Inscripción
Código	CU-016
Actores	Administrador
Descripción	
Las planificaciones se encuentran activas pueden activar las fechas de inscripción.	
Precondiciones	CU-015, CU-016
Postcondiciones	CU-019
Acción de los actores	Respuesta del Sistema
Listar planificaciones	Se listan las planificaciones y se visualizan las fechas de inicio y fin, el nombre del centro de capacitación, nombre del curso.
Activar inscripción	Se registran las fechas de inicio y finalización de las fechas de inscripción.
Visualizar inscripción	Las planificaciones con fechas de inscripción deben ser visualizadas en la pantalla de inicio para que los usuarios puedan inscribirse.
Fallo Posible	Solución

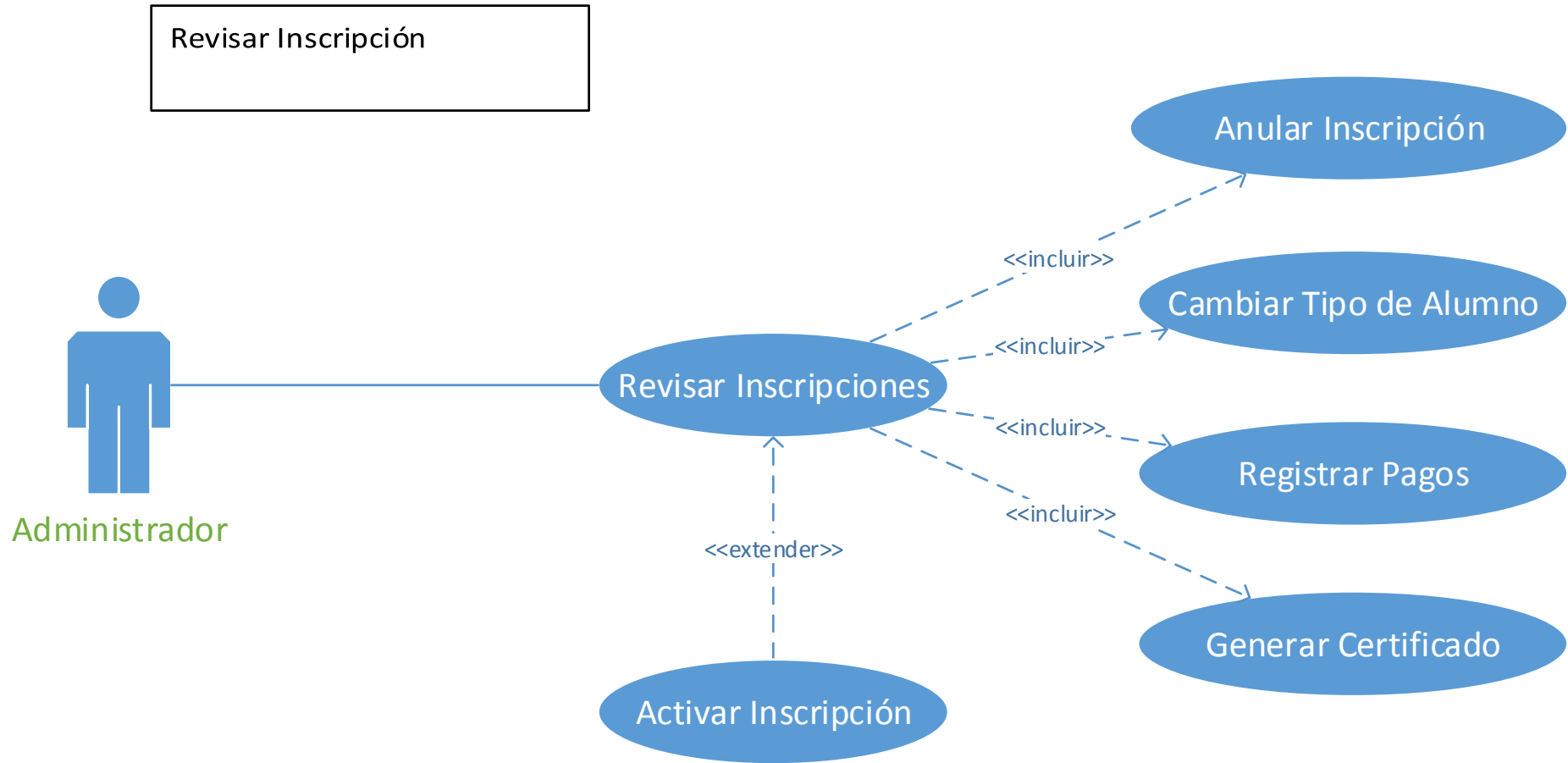


Figura 30. Diagrama de Casos de Uso de Revisar Inscripción

Tabla 17**Revisar Inscripción**

Caso de Uso	
Nombre	Revisar Inscripciones
Código	CU-017
Actores	Administrador
Descripción	
Permite visualizar las personas que se inscribieron en el curso, se puede anular la inscripción, cambiar el tipo de alumno, registrar los pagos, generar los certificados para las personas que aprobaron el curso.	
Precondiciones	CU-021
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Anular inscripción	Se cambia el estado de la inscripción ha anulado.
Cambiar tipo de alumno	Se selecciona la persona y se cambia el tipo de alumno, esta acción recalcula los valores a cancelar por parte del alumno.
Registra pagos	Se registran los pagos que realiza la persona, al registrar un pago se actualiza el valor por pagar.
Generar certificado	Se realiza la impresión de los certificados de aprobación de las personas que aprobaron el curso.
Fallo Posible	Solución

Tabla 18**Actualizar datos de alumno**

Caso de Uso	
Nombre	Actualizar Datos de Alumno
Código	CU-018
Actores	Administrador
Descripción	
Permite actualizar la información de alumnos inscritos.	
Precondiciones	CU-021
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Ingresar número de cédula Para verificarlo	Despliega la información del alumno nombre, cédula, correo, dirección, teléfono, celular, correo y género.
Actualiza información	Actualiza los datos del alumno nombre, cédula, correo, dirección, teléfono, celular, correo y género.
Guarda información	Guarda en la base de datos la información del alumno nombre, cédula, correo, dirección, teléfono, celular, correo y género.
Fallo Posible	Solución

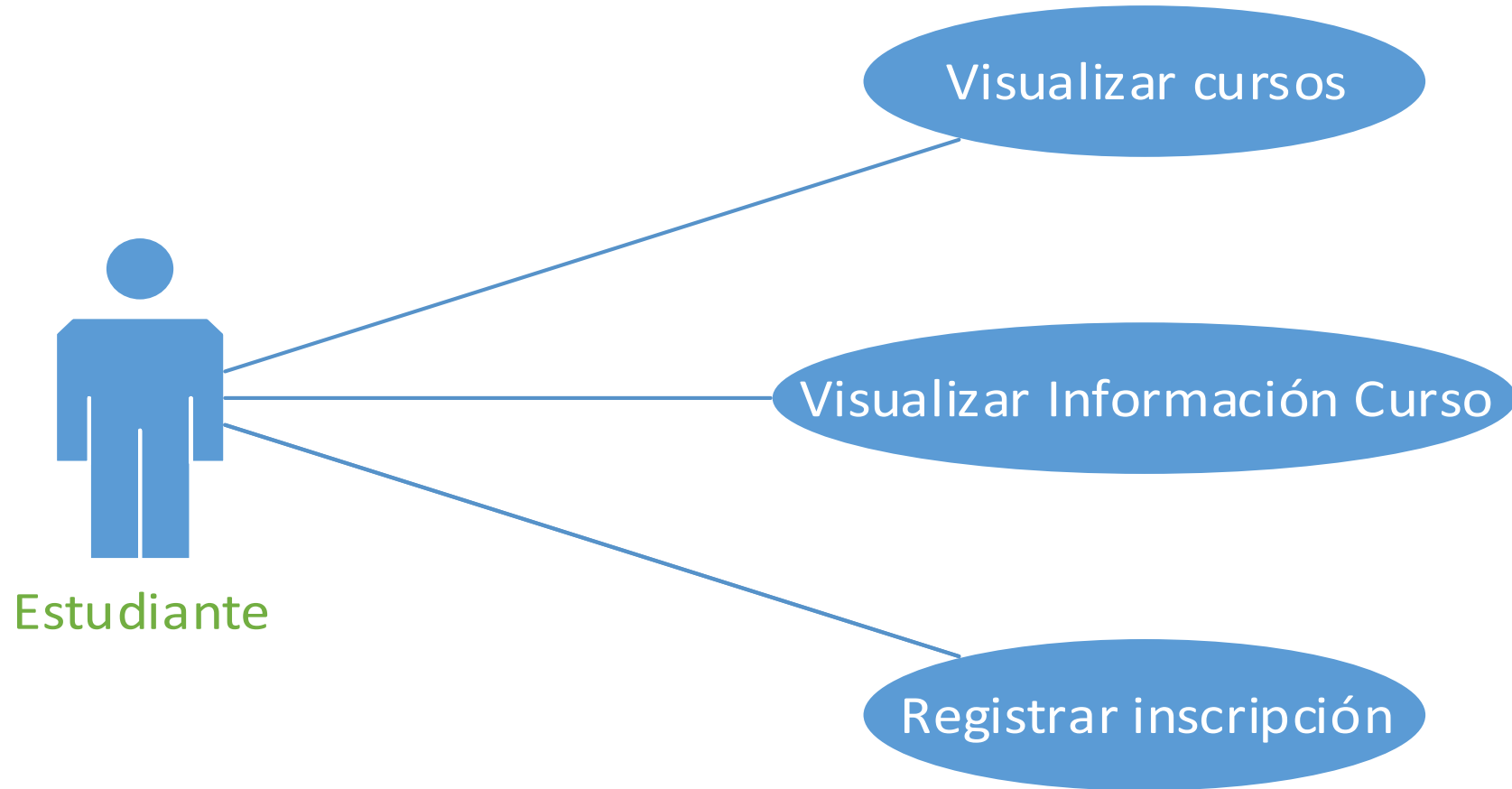


Figura 31. Diagrama de Casos de Uso de Visualizar Información

Tabla 19**Visualizar Cursos**

Caso de Uso	
Nombre	Visualizar cursos
Código	CU-019
Actores	Estudiante
Descripción	
Se permite la visualización de los cursos planificados y que su fecha de inscripción se encuentre activa.	
Precondiciones	CU-016
Postcondiciones	CU-018
Acción de los actores	Respuesta del Sistema
Listar cursos	Se lista la información: Nombre del curso, centro de capacitación, provincia, fecha de inicio del curso
Fallo Posible	Solución

Tabla 20**Visualizar Información**

Caso de Uso	
Nombre	Visualizar Información Curso
Código	CU-020
Actores	Estudiante
Descripción	
Se puede visualizar la información del curso selecciona de la lista de cursos con fechas de inscripción activas.	
Precondiciones	CU-017
Postcondiciones	CU-019
Acción de los actores	Respuesta del Sistema
Seleccionar curso	De la lista de cursos con fechas de inscripción activas se puede seleccionar el curso y visualizar información detallada del temario, fechas de inicio y finalización, costos y centros de capacitación.
Cancelar visualización	Se dispone de la opción cancelar visualización, que permite cerrar el formulario y regresar a la pantalla de inicio.
Fallo Posible	Solución

Tabla 21**Visualizar Información**

Caso de Uso	
Nombre	Visualizar Información Curso
Código	CU-020
Actores	Estudiante
Descripción	
Se puede visualizar la información del curso selecciona de la lista de cursos con fechas de inscripción activas.	
Precondiciones	CU-017
Postcondiciones	CU-019
Acción de los actores	Respuesta del Sistema
Seleccionar curso	De la lista de cursos con fechas de inscripción activas se puede seleccionar el curso y visualizar información detallada del temario, fechas de inicio y finalización, costos y centros de capacitación.
Cancelar visualización	Se dispone de la opción cancelar visualización, que permite cerrar el formulario y regresar a la pantalla de inicio.
Fallo Posible	Solución

Tabla 22

Registrar Inscripción

Caso de Uso	
Nombre	Registrar Inscripción
Código	CU-021
Actores	Estudiante
Descripción	
El estudiante registra su inscripción en el curso seleccionado.	
Precondiciones	CU-017, CU-018
Postcondiciones	N/A
Acción de los actores	Respuesta del Sistema
Visualizar información del curso	Se muestra la información del centro de capacitación, lugar de ubicación y temario del curso.
Seleccionar horario	Se lista las aulas y sus horarios de clases de los cuales el estudiante puede escoger un solo horario.
Ingresar datos del alumno	Se registra la información de: número de cédula, nombres y apellidos, dirección, número telefónicos, correo electrónico y género.
Ingresar nivel académico	Se registra el nivel académico del estudiante.
Registrar tipo de alumno	Se lista los tipos de alumnos y sus descuentos, el estudiante puede seleccionar solo un tipo de alumno. De acuerdo al tipo de alumno seleccionado se genera el monto a cancelar por el curso.



Registrar forma de pago	Continúa Se listan las formas de pago y el estudiante selecciona una opción.
Registrar datos de facturación	Se registra los datos de la persona a la que se genera la factura. Continúa
Guardar Registro	Se almacena la información del estudiante y se registra su inscripción en el curso seleccionado.
Fallo Posible	Solución

Tabla 23

Visualizar Nota

Caso de Uso	
Nombre	Visualizar Nota
Código	CU-023
Actores	Estudiante
Descripción	
Se puede visualizar la información del curso selecciona de la lista de cursos con fechas de inscripción activas.	
Precondiciones	CU-017
Postcondiciones	CU-019
Acción de los actores	Respuesta del Sistema
Seleccionar Nota	Visualiza la información de curso, centro de capacitación, fecha inicio, fecha fin, fecha de inscripción y estado. Visualiza Fecha de registro Nota y Motivo de la nota
Cancelar visualización	Sale dela pantalla de alumno.
Fallo Posible	Solución

Tabla 24**Visualizar Asistencia**

Caso de Uso	
Nombre	Visualizar Asistencia
Código	CU-024
Actores	Estudiante
Descripción	
Se puede visualizar la información del curso seleccionado y la asistencia del alumno.	
Precondiciones	CU-017
Postcondiciones	CU-019
Acción de los actores	Respuesta del Sistema
Seleccionar Asistncia	Visualiza la información de curso, centro de capacitación, fecha inicio, fecha fin, fecha de inscripción y estado. Visualiza Fecha de estado, Fecha de Registro y Estado.
Salir	Sale dela pantalla de alumno.
Fallo Posible	Solución

3.3.2 Diagramas de Secuencia

A continuación se muestran los diagramas de secuencia.

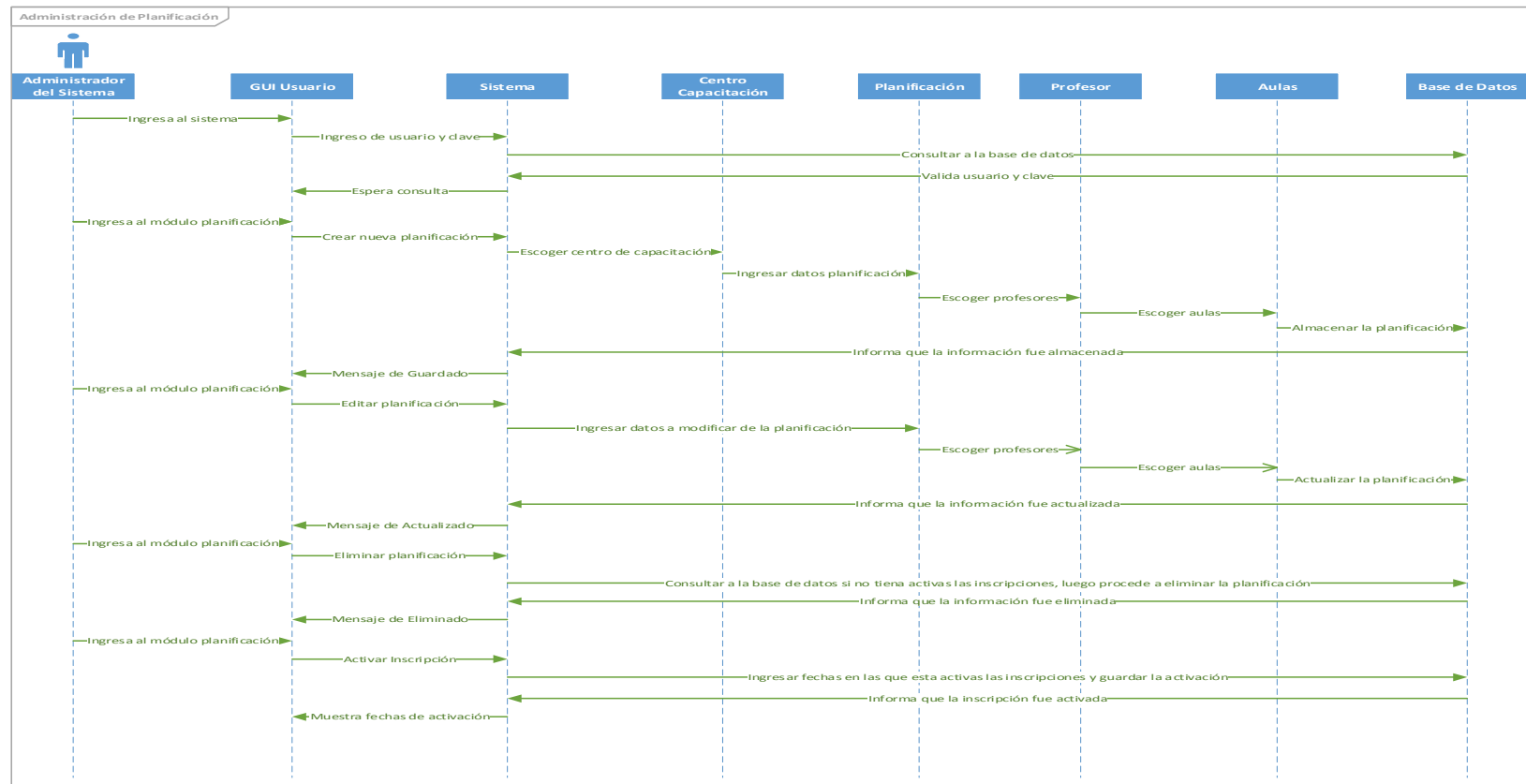


Figura 32. Diagrama de Secuencia de Planificación

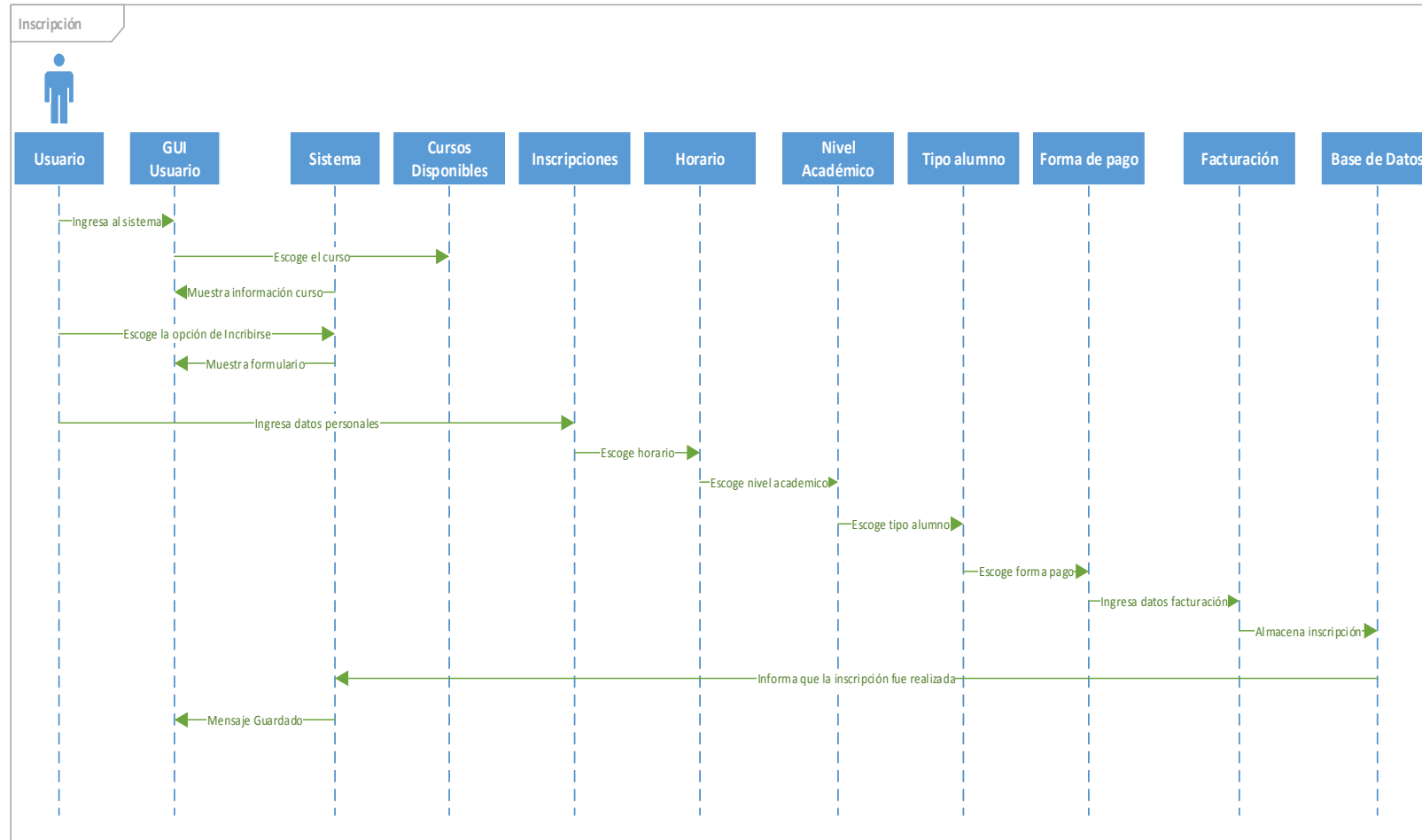


Figura 33. Diagrama de Secuencia de Inscripciones

3.3.3 Diagrama de Base de Datos

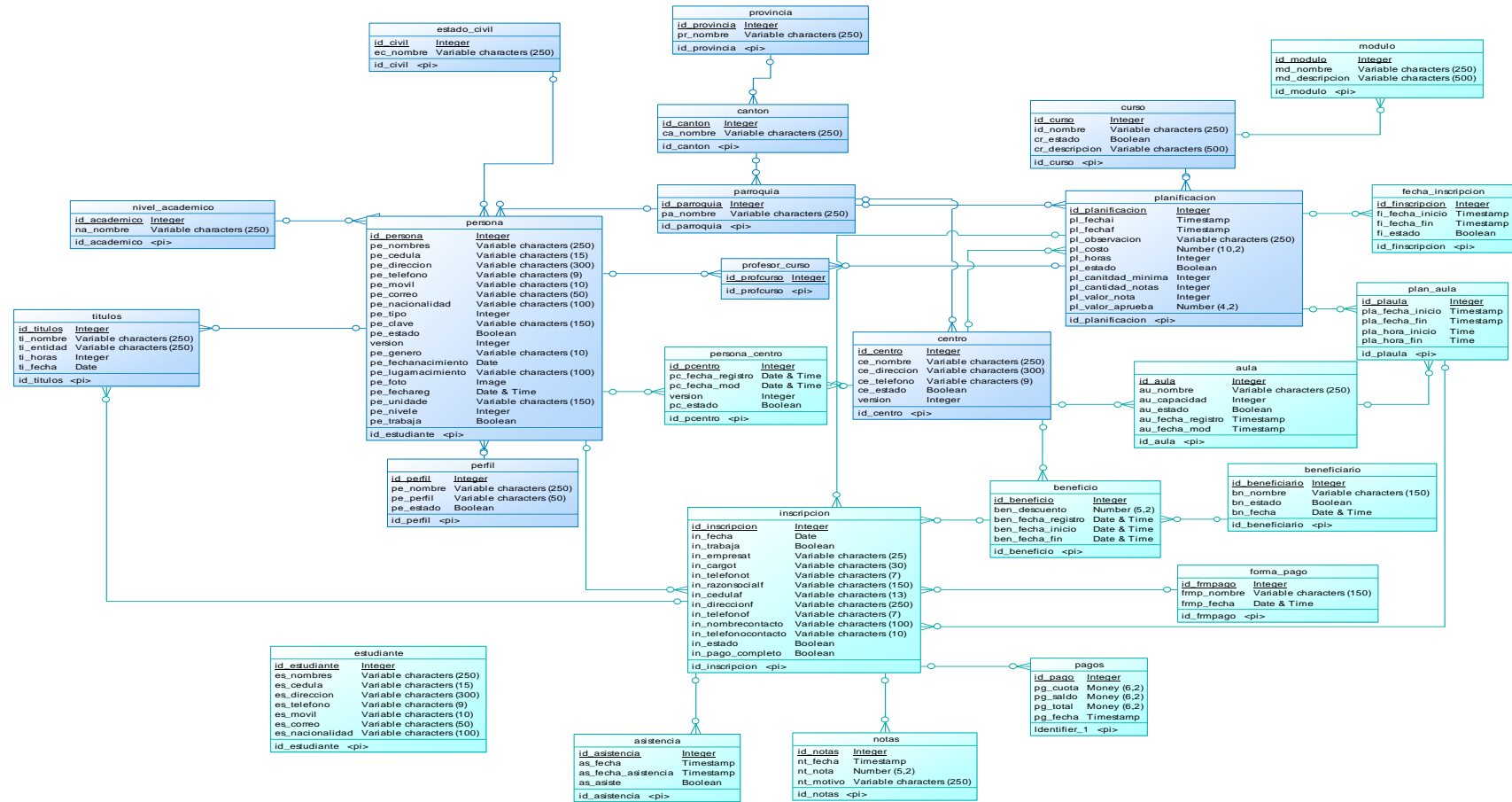


Figura 34. Diagrama de Modelo Lógico

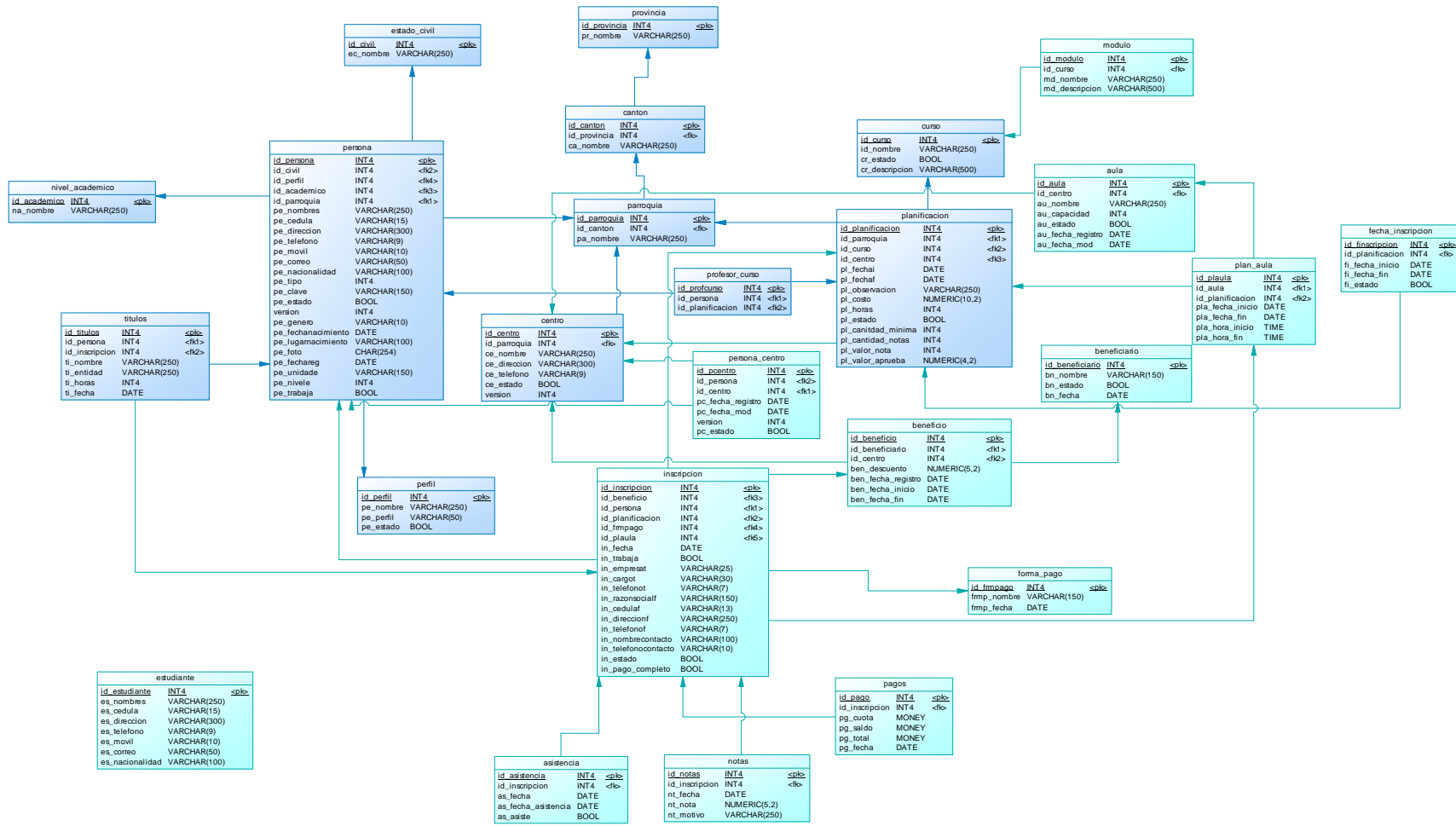


Figura 35. Diagrama de Modelo Físico

3.3.4 Diccionario de Datos

3.3.4.1 Listado de Tablas

Tabla 25

Listado de Tablas

<i>Nombre</i>	<i>Código</i>	<i>Descripción</i>
Asistencia	ASISTENCIA	
Aula	AULA	
beneficiario	BENEFICIARIO	
Beneficio	BENEFICIO	
Canton	CANTON	
Centro	CENTRO	
Curso	CURSO	
estado_civil	ESTADO_CIVIL	
fecha_inscripcion	FECHA_INSCRIPCION	
forma_pago	FORMA_PAGO	
inscripcion	INSCRIPCION	
modulo	MODULO	
nivel_academico	NIVEL_ACADEMICO	
notas	NOTAS	
pagos	PAGOS	
parroquia	PARROQUIA	
perfil	PERFIL	
persona	PERSONA	
persona_centro	PERSONA_CENTRO	
plan_aula	PLAN_AULA	
planificacion	PLANIFICACION	
profesor_curso	PROFESOR_CURSO	
provincia	PROVINCIA	
titulos	TITULOS	

*Tabla 26**Asistencia*

Tabla Asistencia	
Nombre	Código
id_asistencia	ID_ASISTENCIA
id_inscripcion	ID_INSCRIPCION
as_fecha	AS_FECHA
as_fecha_asistencia	AS_FECHA_ASISTENCIA
as_asiste	AS_ASISTE

*Tabla 27**Aula*

Tabla Aula	
Nombre	Código
id_aula	ID_AULA
id_centro	ID_CENTRO
au_nombre	AU_NOMBRE
au_capacidad	AU_CAPACIDAD
au_estado	AU_ESTADO
au_fecha_registro	AU_FECHA_REGISTRO
au_fecha_mod	AU_FECHA_MOD

*Tabla 28**Beneficiario*

Tabla Beneficiario	
Nombre	Código
id_beneficiario	ID_BENEFICIARIO
bn_nombre	BN_NOMBRE
bn_estado	BN_ESTADO
bn_fecha	BN_FECHA

*Tabla 29**Beneficiario*

Tabla Beneficio	
Nombre	Código
id_beneficio	ID_BENEFICIO
id_beneficiario	ID_BENEFICIARIO
id_centro	ID_CENTRO
ben_descuento	BEN_DESCUENTO
ben_fecha_registro	BEN_FECHA_REGISTRO
ben_fecha_inicio	BEN_FECHA_INICIO
ben_fecha_fin	BEN_FECHA_FIN

*Tabla 30**Cantón*

Tabla Canton	
Nombre	Código
id_canton	ID_CANTON
id_provincia	ID_PROVINCIA
ca_nombre	CA_NOMBRE

*Tabla 31**Centro*

Tabla Centro	
Nombre	Código
id_centro	ID_CENTRO
id_parroquia	ID_PARROQUIA
ce_nombre	CE_NOMBRE
ce_direccion	CE_DIRECCION
ce_telefono	CE_TELEFONO
ce_estado	CE_ESTADO
Versión	VERSION

Tabla 32**Curso**

Tabla Curso	
Nombre	Código
id_curso	ID_CURSO
id_nombre	ID_NOMBRE
cr_estado	CR_ESTADO
cr_descripcion	CR_DESCRIPCION

Tabla 33**Civil**

Tabla Estado_Civil	
Nombre	Código
id_civil	ID_CIVIL
ec_nombre	EC_NOMBRE

Tabla 34**Fecha descripción**

Tabla Fecha Inscripción	
Nombre	Código
id_finscripcion	ID_FINSCRIPCION
id_planificacion	ID_PLANIFICACION
fi_fecha_inicio	FI_FECHA_INICIO
fi_fecha_fin	FI_FECHA_FIN
fi_estado	FI_ESTADO

Tabla 35**Forma Pago**

Tabla Forma_Pago	
Nombre	Código
id_frm_pago	ID_FRMPAGO
frmp_nombre	FRMP_NOMBRE
frmp_fecha	FRMP_FECHA

*Tabla 36**Inscripción*

Tabla Inscripcion	
Nombre	Código
id_inscripcion	ID_INSCRIPCION
id_beneficio	ID_BENEFICIO
id_persona	ID_PERSONA
id_planificacion	ID_PLANIFICACION
id_frm_pago	ID_FRMPAGO
id_plaula	ID_PLAULA
in_fecha	IN_FECHA
in_trabaja	IN_TRABAJA
in_empresat	IN_EMPRESAT
in_cargot	IN_CARGOT
in_telefonot	IN_TELEFONOT
in_razonsocialf	IN_RAZONSOCIALF
in_cedulaf	IN_CEDULAF
in_direccionf	IN_DIRECCIONF
in_telefonof	IN_TELEFONOF
in_nombrecontacto	IN_NOMBRECONTACTO
in_telefonocontacto	IN_TELEFONOCONTACTO
in_estado	IN_ESTADO
in_pago_completo	IN_PAGO_COMPLETO
in_descuento	IN_DESCUENTO
in_costo_curso	IN_COSTO_CURSO
in_total_pagar	IN_TOTAL_PAGAR
in_por_pagar	IN_POR_PAGAR
in_aprobado	IN_APROBADO

*Tabla 37**Módulo*

Tabla Modulo	
Nombre	Código
id_modulo	ID_MODULO
id_curso	ID_CURSO
md_nombre	MD_NOMBRE
md_descripcion	MD_DESCRIPCION

*Tabla 38**Nivel Académico*

Tabla Nivel_Academico	
Nombre	Código
id_academico	ID_ACADEMICO
na_nombre	NA_NOMBRE

*Tabla 39**Notas*

Tabla Notas	
Nombre	Código
id_notas	ID_NOTAS
id_inscripcion	ID_INSCRIPCION
nt_fecha	NT_FECHA
nt_nota	NT_NOTA
nt_motivo	NT_MOTIVO

*Tabla 40**Pagos*

Tabla Pagos	
Nombre	Código
id_pago	ID_PAGO
id_inscripcion	ID_INSCRIPCION
pg_cuota	PG_CUOTA
pg_saldo	PG_SALDO
pg_total	PG_TOTAL
pg_fecha	PG_FECHA

*Tabla 41**Parroquia*

Tabla Parroquia	
Nombre	Código
id_parroquia	ID_PARROQUIA
id_canton	ID_CANTON
pa_nombre	PA_NOMBRE

*Tabla 42**Perfil*

Tabla Perfil	
Nombre	Código
id_perfil	ID_PERFIL
pe_nombre	PE_NOMBRE
pe_perfil	PE_PERFIL
pe_estado	PE_ESTADO

Tabla 43**Persona**

Tabla Persona	
Nombre	Código
id_persona	ID_PERSONA
id_civil	ID_CIVIL
id_perfil	ID_PERFIL
id_academico	ID_ACADEMICO
id_parroquia	ID_PARROQUIA
pe_nombres	PE_NOMBRES
pe_cedula	PE_CEDULA
pe_direccion	PE_DIRECCION
pe_telefono	PE_TELEFONO
pe_movil	PE_MOVIL
pe_correo	PE_CORREO
pe_nacionalidad	PE_NACIONALIDAD
pe_tipo	PE_TIPO
pe_clave	PE_CLAVE
pe_estado	PE_ESTADO
Versión	VERSION
pe_genero	PE_GENERO
pe_fechanacimiento	PE_FECHANACIMIENTO
pe_lugarnacimiento	PE_LUGARNACIMIENTO
pe_foto	PE_FOTO
pe_fechaereg	PE_FECHAREG
pe_unidade	PE_UNIDADE
pe_nivele	PE_NIVELE
pe_trabaja	PE_TRABAJA

Tabla 44**Persona Centro**

Tabla Persona_Centro	
Nombre	Código
id_pcentro	ID_PCENTRO
id_persona	ID_PERSONA
id_centro	ID_CENTRO
pc_fecha_registro	PC_FECHA_REGISTRO
pc_fecha_mod	PC_FECHA_MOD
Versión	VERSION
pc_estado	PC_ESTADO

Tabla 45**Plan aula**

Tabla Plan_Aula	
Nombre	Código
id_plaula	ID_PLAULA
id_aula	ID_AULA
id_planificacion	ID_PLANIFICACION
pla_fecha_inicio	PLA_FECHA_INICIO
pla_fecha_fin	PLA_FECHA_FIN
pla_hora_inicio	PLA_HORA_INICIO
pla_hora_fin	PLA_HORA_FIN

*Tabla 46**Planificación*

Tabla Planificacion	
Nombre	Código
id_planificacion	ID_PLANIFICACION
id_parroquia	ID_PARROQUIA
id_curso	ID_CURSO
id_centro	ID_CENTRO
pl_fecha_i	PL_FECHAI
pl_fecha_f	PL_FECHAF
pl_observacion	PL_OBSERVACION
pl_costo	PL_COSTO
pl_horas	PL_HORAS
pl_estado	PL_ESTADO
pl_cantidad_minima	PL_CANTIDAD_MINIMA
pl_cantidad_notas	PL_CANTIDAD_NOTAS
pl_valor_nota	PL_VALOR_NOTA
pl_valor_aprueba	PL_VALOR_APRUEBA

*Tabla 47**Profesor Curso*

Tabla Profesor_Curso	
Nombre	Código
id_profcurso	ID_PROFCURSO
id_persona	ID_PERSONA
id_planificacion	ID_PLANIFICACION

Tabla 48: Provincia

Tabla Provincia	
Nombre	Código
id_provincia	ID_PROVINCIA
pr_nombre	PR_NOMBRE

*Tabla 49**Títulos*

Tabla Titulos	
Nombre	Código
id_titulos	ID_TITULOS
id_persona	ID_PERSONA
id_inscripcion	ID_INSCRIPCION
ti_nombre	TI_NOMBRE
ti_entidad	TI_ENTIDAD
ti_horas	TI_HORAS
ti_fecha	TI_FECHA

*Tabla 50**Asistencia*

Tabla Titulos	
Nombre	Código
id_asistencia	ID_ASISTENCIA
as_fecha	AS_FECHA
as_fecha_asistencia	AS_FECHA_ASISTENTE
as_asiste	AS_ASISTE

*Tabla 51**Notas*

Tabla Titulos	
Nombre	Código
id_notas	ID_NOTAS
nt_fecha	NT_FECHA
nt_nota	NT_NOTA
nt_motivo	NT_MOTIVO

3.3.5 Arquitectura del Sistema

3.3.5.1 Arquitectura de Despliegue

Para el desarrollo y despliegue de la aplicación se emplea una arquitectura de tres capas basados en el modelo–vista–controlador (MVC), que es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

La capa de vista que incluye no sólo el navegador, sino también el servidor web que es el responsable de presentar los datos un formato adecuado.

La capa de control, contiene la lógica de programación.

La capa de modelo, proporciona el almacenaje de los datos en al motor de base de datos.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Las diferentes capas de la arquitectura se presentan a continuación:

Vista: Páginas web desplegadas en navegadores web empleando Java Server Faces

Controlador: Servidor de Aplicaciones Glassfish versión 4

Modelo: Servidor de Datos Postgress

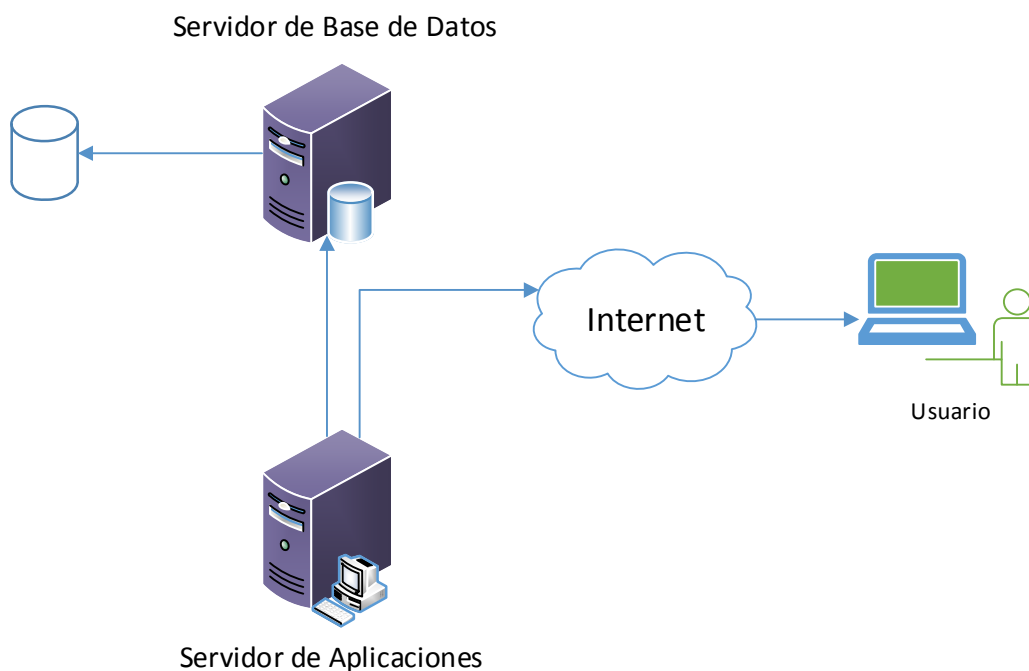


Figura 36. Servidor de Base de Datos

3.3.5.2 Arquitectura de Desarrollo

Para el desarrollo del sistema informático se emplea el estándar de J2EE, mediante la implementación de Enterprise JavaBeans (EJB). El objetivo de los EJB es dotar de un modelo de desarrollo que permite abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.), para la creación del sistema informático se emplean:

EntityBeans, su objetivo es encapsular los objetos del lado del servidor que almacena los datos. Los EJB de entidad presentan la característica fundamental de la persistencia de objetos.

SessionBeans, su objetivo es proveer servicios que encapsulan la lógica de negocio y son consumidos por la capa de presentación.

ManagedBean, su objetivo es consumir los servicios provistos por los SessionBean y procesar la información ingresada en las páginas web.

HTML, su objetivo es permitir la comunicación entre el usuario y el sistema informático mediante la visualización de formularios para el ingreso y despliegue de datos.

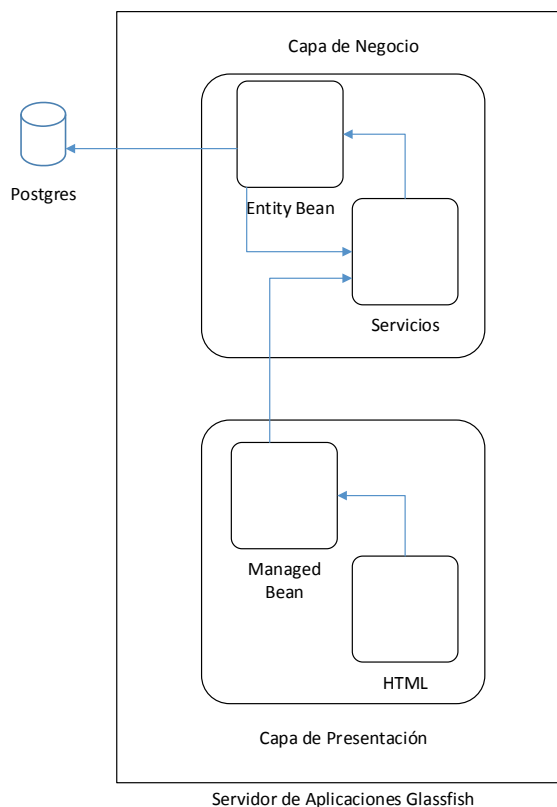


Figura 37. Servidor de Aplicaciones Glassfish

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

4.1 Implementación

Para la implantación del sistema se procede con la instalación y configuración del entorno de producción.

El equipo sobre el que se realiza la instalación tiene las siguientes características:

Tabla 52

Características del Servidor

HARDWARE	
Servidor	HP Proliant DL380p Gen8 virtualizado con tiene 6 cores de 2.294 Ghz de 2.3 Ghz disco de disco duro de 600 GB en arreglo raid 1 (con disco espejo, Memoria RAM de 20 GB)

Sistema Operativo Windows 8 Virtualizado sobre el servidor descrito anteriormente.

4.1.1 Instalación y configuración del Sistema

La instalación se realiza sobre un Sistema Operativo Windows 8, por lo que todas las versiones de las herramientas necesarias para desplegar la aplicación deben ser descargadas para soportar este sistema operativo.

4.1.1.1 Instalación del JDK

Se trabaja con la versión 7 del JDK update 45, la cual puede ser descargado de la siguiente url:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

La instalación del JDK se describe a continuación:

1. Se descarga el instalador y lo guarda en el disco duro.
2. Se ejecuta el instalador y especifica la ubicación en la que se almacenará, tal como se muestra en la figura 38.



Figura 38. Ejecución del Instalador

3. Luego de especificar la ubicación de la instalación presionando Next.
4. Al finalizar la instalación dar clic en Close, tal como se muestra en la figura 39.



Figura 39. Finalización de la Instalación

4.1.1.2 ***Instalación de Postgres***

Se trabaja con la versión 9.3.4 del Motor de Base Datos PostgreSQL, la cual puede ser descargado de la siguiente url:

<http://www.enterprisedb.com/products-services-training/pgdownload#windows>

La instalación de Postgres se describe a continuación,:

1. Se descarga el instalador y lo guarda en el disco duro.
2. Se ejecuta el instalador y especifica la ubicación en la que se almacenará, tal y como se muestra en la figura 40.

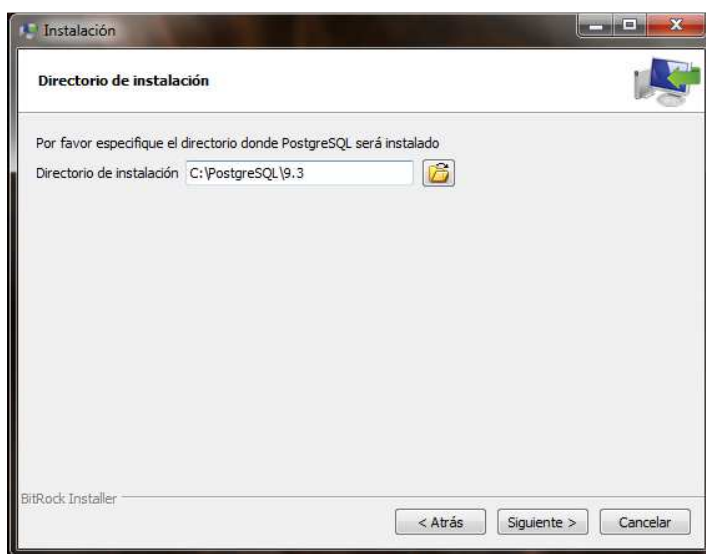


Figura 40. Instalador de Postgres

3. Especifique el nombre de usuario y la clave con los siguientes datos:
 - a. Usuario.- postgres
 - b. Clave.- postgres

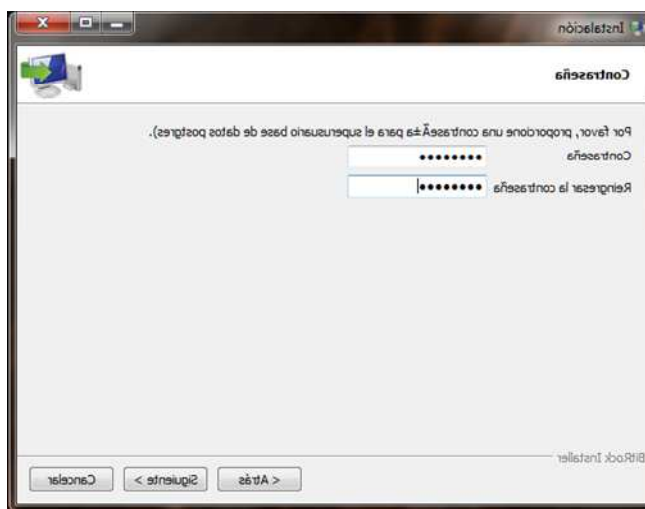


Figura 41. Especificación de Usuario y Clave

4. Especifique el puerto 5432, tal y como se muestra en la figura 42.

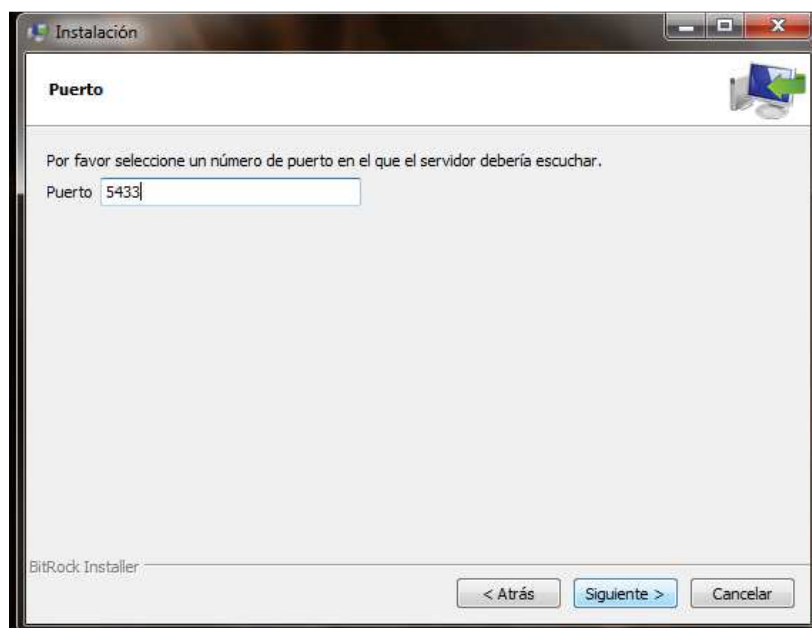


Figura 42. Especificación del Puerto

5. Seleccione la configuración regional y el lenguaje para la base de datos, tal y como se muestra en la figura 43.

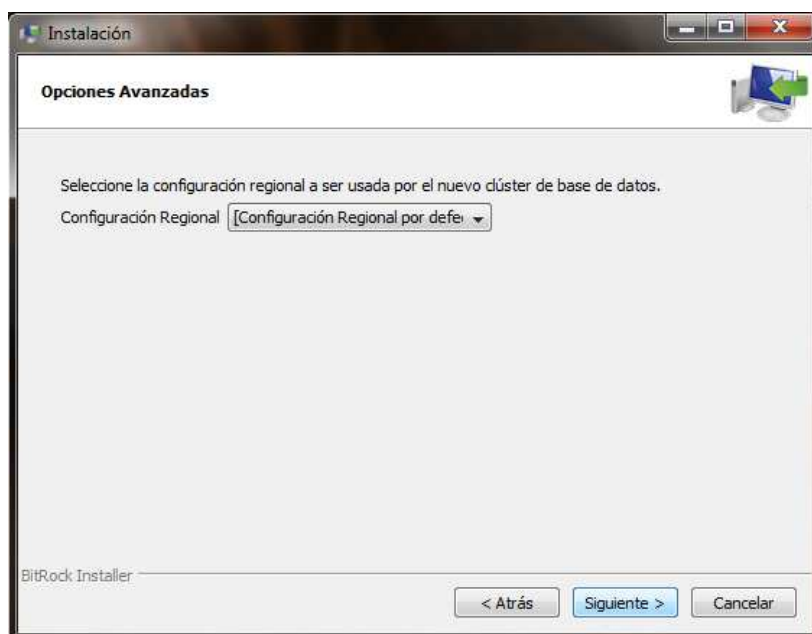


Figura 43. Configuración Regional y Lenguaje de la Base de Datos

6. Presione Next para iniciar la instalación tal como se muestra en la figura 44.

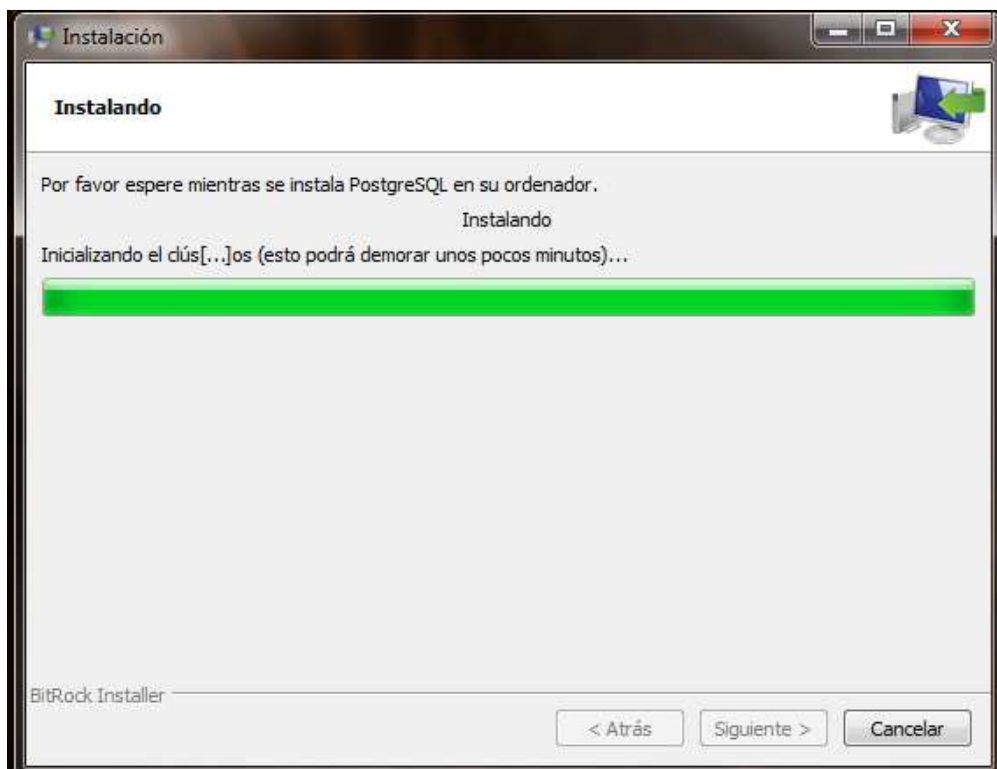


Figura 44. Pantalla de Inicio de Instalación

7. Al finalizar la instalación podrá acceder a la ventana de administración del Postgres, ingresa el usuario y clave, tal y como se muestra en la figura 45.

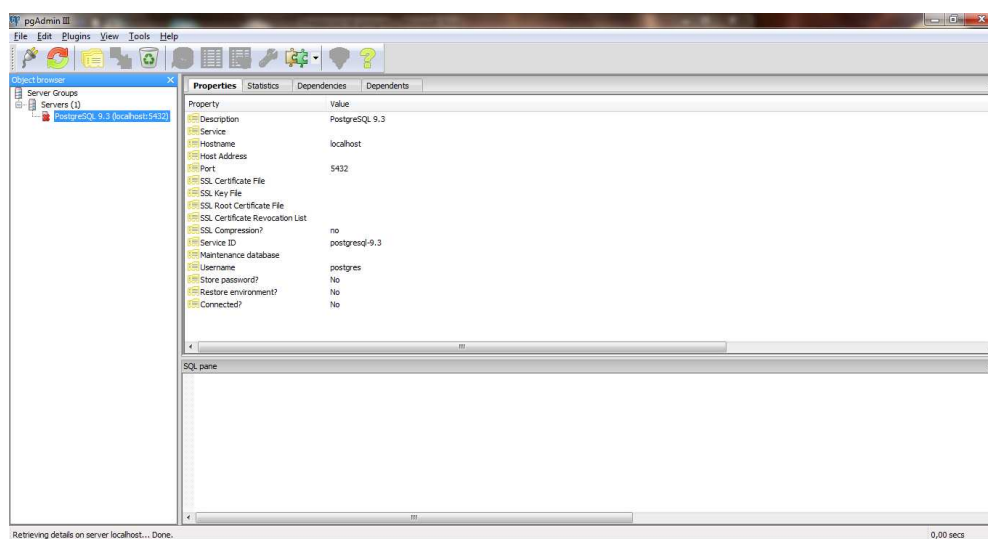


Figura 45. Pantalla de Administrador de Postgres

- Para crear la base de datos de clic derecho sobre Databases y seleccione la opción New Database, tal y como se muestra en la figura 46.

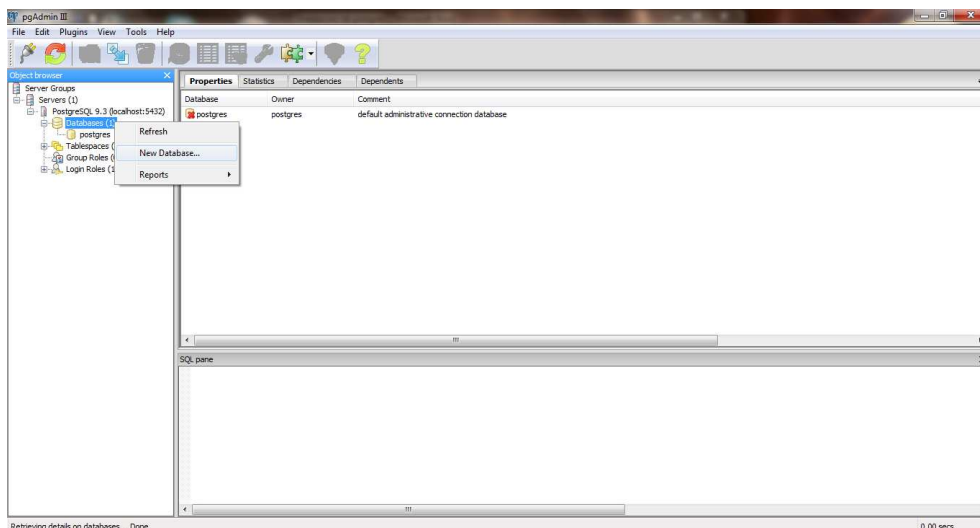


Figura 46. Pantalla de Creación de Base de Datos

- Ingresa el nombre de la base de nombre educación, tal y como se muestra en la figura 47.

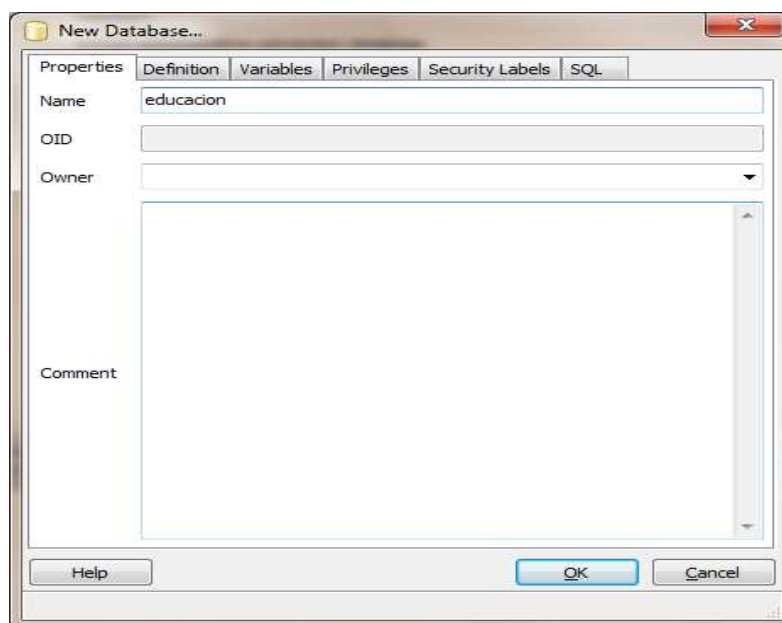


Figura 47. Pantalla de Ingreso del Nombre de la Base de Datos

10. Proceda con la creación de la estructura de la base de datos para lo cual des clic derecho sobre la base de datos creada, seleccione la opción SQL en la barra de menú, tal y como se muestra en la figura 48.



Figura 48. Creación de la Estructura de la Base de Datos

11. En la ventana de ejecución de comandados ejecute el siguiente script tal y como se muestra en la figura 49.

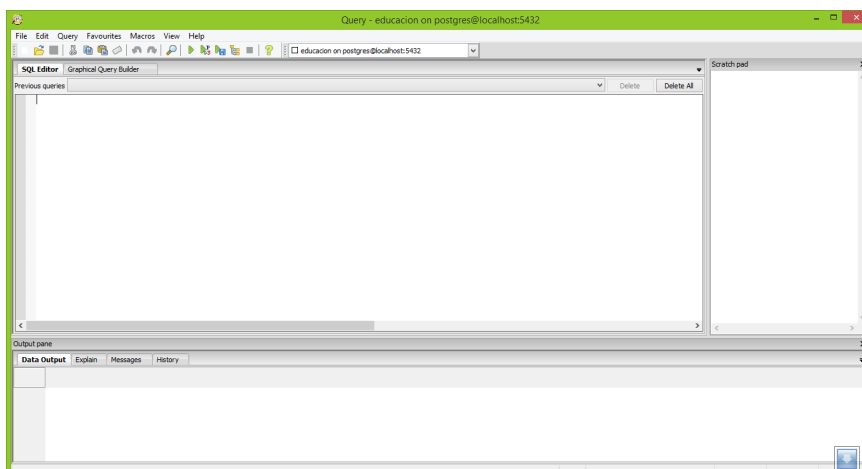


Figura 49. Ejecución del Script

Script

```

/*=====*/
/* Table: ASISTENCIA */
/*=====*/
create table ASISTENCIA (
ID_ASISTENCIA    INTEGER           not null,
ID_INSCRIPCION   INTEGER           null,
AS_FECHA         TIMESTAMP WITHOUT TIME ZONE    not null,
AS_FECHA_ASISTENCIA  TIMESTAMP WITHOUT TIME ZONE  not null,
AS_ASISTE        BOOL             not null,
constraint PK_ASISTENCIA primary key (ID_ASISTENCIA));
comment on table ASISTENCIA is 'Registra las asistencias de las personas
inscritas en el curso.';
comment on column ASISTENCIA.ID_ASISTENCIA is 'Clave primaria.';

```

comment on column ASISTENCIA.AS_FECHA is 'Fecha en la que se crea el registro.';

comment on column ASISTENCIA.AS_FECHA_ASISTENCIA is 'Fecha en la que se imparte la clase.';

```
/*=====*/
```

```
/* Index: ASISTENCIA_PK */
```

```
/*=====*/
```

```
create unique index ASISTENCIA_PK on ASISTENCIA (ID_ASISTENCIA);
```

```
/*=====*/
```

```
/* Index: RELATIONSHIP_30_FK */
```

```
/*=====*/
```

```
create index RELATIONSHIP_30_FK on ASISTENCIA (ID_INSCRIPCION);
```

```
/*=====*/
```

```
/* Table: AULA */
```

```
/*=====*/
```

```
create table AULA (
```

```
  ID_AULA      INTEGER      not null,
```

```
  ID_CENTRO    INTEGER      null,
```

```
  AU_NOMBRE    VARCHAR(250)  not null,
```

```
  AU_CAPACIDAD INTEGER      not null,
```

```
  AU_ESTADO    BOOL        not null,
```

```
  AU_FECHA_REGISTRO  TIMESTAMP WITHOUT TIME ZONE not null,
```

```
  AU_FECHA_MOD      TIMESTAMP WITHOUT TIME ZONE null,
```

```
  constraint PK_AULA primary key (ID_AULA));
```

comment on table AULA is 'Almacena la información de las aulas de los centros de capacitación.';

comment on column AULA.ID_AULA is 'Clave primaria';

comment on column AULA.AU_NOMBRE is 'Nombre del aula.';

comment on column AULA.AU_CAPACIDAD is 'Capacidad de alumnos del aula.';

comment on column AULA.AU_ESTADO is 'True.- Activo. False.- Inactivo';

comment on column AULA.AU_FECHA_REGISTRO is 'Fecha de creación del registro.';

comment on column AULA.AU_FECHA_MOD is 'Fecha de modificación del registro.';

```
/*=====*/
```

```
/* Index: AULA_PK */
```

```
/*=====*/
```

```
create unique index AULA_PK on AULA (ID_AULA);
```

```
/*=====*/
```

```
/* Index: RELATIONSHIP_23_FK */
```

```
/*=====*/
```

```
create index RELATIONSHIP_23_FK on AULA (ID_CENTRO);
```

```
/*=====*/
```

```
/* Table: BENEFICIARIO */
```

```
/*=====*/
```

```
create table BENEFICIARIO (
```

```
    ID_BENEFICIARIO    INTEGER        not null,
```

```
    BN_NOMBRE          VARCHAR(150)   not null,
```

```
    BN_ESTADO          BOOL            not null,
```

```
    BN_FECHA           TIMESTAMP WITHOUT TIME ZONE not null,
```

```
    constraint PK_BENEFICIARIO primary key (ID_BENEFICIARIO));
```

comment on table BENEFICIARIO is 'Almacena la información de los tipos de beneficiarios o tipos de alumnos de los centros de capacitación.';

comment on column BENEFICIARIO.ID_BENEFICIARIO is 'Clave primaria';

comment on column BENEFICIARIO.BN_NOMBRE is 'Nombre del tipo de beneficiario.';

comment on column BENEFICIARIO.BN_ESTADO is 'True.- Activo. False.- Inactivo';

comment on column BENEFICIARIO.BN_FECHA is 'Fecha de creación de registro.';

```
/*=====
```

```
==*/
```

```

/* Index: BENEFICIARIO_PK                                     */
/*=====*/
create unique index BENEFICIARIO_PK on BENEFICIARIO
(ID_BENEFICIARIO
);
/*=====*/
/* Table: BENEFICIO                                         */
/*=====*/
create table BENEFICIO (
  ID_BENEFICIO    INTEGER          not null,
  ID_BENEFICIARIO INTEGER          null,
  ID_CENTRO       INTEGER          null,
  BEN_DESCUENTO   NUMERIC(5,2)     not null,
  BEN_FECHA_REGISTRO  TIMESTAMP WITHOUT TIME ZONE not null,
  BEN_FECHA_INICIO   TIMESTAMP WITHOUT TIME ZONE not null,
  BEN_FECHA_FIN      TIMESTAMP WITHOUT TIME ZONE not null,
  constraint PK_BENEFICIO primary key (ID_BENEFICIO));
comment on table BENEFICIO is'Almacena los descuentos aplicados por centros
de capacitación a los diferentes tipos de eneficiarios.';
comment on column BENEFICIO.ID_BENEFICIARIO is'Clave primaria';
comment on column BENEFICIO.BEN_DESCUENTO is'Porcentaje de
descuento para el beneficiario.';
comment on column BENEFICIO.BEN_FECHA_REGISTRO is'Fecha de
creación del registro.';
/*=====*/
/* Index: BENEFICIO_PK                                     */
/*=====*/
create unique index BENEFICIO_PK on BENEFICIO (ID_BENEFICIO);
/*=====*/
/* Index: RELATIONSHIP_17_FK                               */
/*=====*/
create index RELATIONSHIP_17_FK on BENEFICIO (ID_BENEFICIARIO);

```

```

/*=====*/
/* Index: RELATIONSHIP_18_FK */
/*=====*/
create index RELATIONSHIP_18_FK on BENEFICIO (ID_CENTRO );
/*=====*/
/* Table: CANTON */
/*=====*/
create table CANTON (
    ID_CANTON      INTEGER      not null,
    ID_PROVINCIA   INTEGER      null,
    CA_NOMBRE      VARCHAR(250)  not null,
    constraint PK_CANTON primary key (ID_CANTON));
comment on table CANTON is 'Almacena la información de los cantones.';
/*=====
*/
/* Index: CANTON_PK */
/*=====
*/
create unique index CANTON_PK on CANTON (ID_CANTON);
/*=====*/
/* Index: PROVINCIA_CANTON_FK */
/*=====*/
create index PROVINCIA_CANTON_FK on CANTON (ID_PROVINCIA);
/*=====*/
/* Table: CENTRO */
/*=====*/
create table CENTRO (
    ID_CENTRO      INTEGER      not null,
    ID_PARROQUIA   INTEGER      null,
    CE_NOMBRE      VARCHAR(250)  null,
    CE_DIRECCION   VARCHAR(300)  null,
    CE_TELEFONO    VARCHAR(9)    null,

```



```

CE_ESTADO      BOOL      null,
VERSION        INTEGER    null,
constraint PK_CENTRO primary key (ID_CENTRO));
/*=====*/
/* Index: CENTRO_PK                                */
/*=====*/
create unique index CENTRO_PK on CENTRO (ID_CENTRO);
/*=====*/
/* Index: PARROQUIA_CENTRO_FK                       */
/*=====*/
create index PARROQUIA_CENTRO_FK on CENTRO (ID_PARROQUIA);
/*=====*/
/* Table: CURSO                                    */
/*=====*/
create table CURSO (
  ID_CURSO      INTEGER      not null,
  ID_NOMBRE     VARCHAR(250)  not null,
  CR_ESTADO     BOOL         null,
  CR_DESCRIPCION VARCHAR(500) null,
  constraint PK_CURSO primary key (ID_CURSO));
comment on table CURSO is Almacena el nombre de los cursos que imparte el
ctt.';
comment on column CURSO.ID_NOMBRE is 'Nombre del curso';
comment on column CURSO.CR_ESTADO is 'True.-Activo, False.- Inactivo';
comment on column CURSO.CR_DESCRIPCION is 'Descripción del curso.';
/*=====*/
/* Index: CURSO_PK                                */
/*=====*/
create unique index CURSO_PK on CURSO (ID_CURSO );
/*=====*/
/* Table: ESTADO_CIVIL                            */
/*=====*/

```

```

create table ESTADO_CIVIL (
    ID_CIVIL          INTEGER          not null,
    EC_NOMBRE        VARCHAR(250)     not null,
    constraint PK_ESTADO_CIVIL primary key (ID_CIVIL));
comment on table ESTADO_CIVIL is 'Almacena la información del estado civil.';
/*=====*/
/* Index: ESTADO_CIVIL_PK                               */
/*=====*/
create unique index ESTADO_CIVIL_PK on ESTADO_CIVIL (
ID_CIVIL);
/*=====*/
/* Table: FECHA_INSCRIPCION                             */
/*=====*/
create table FECHA_INSCRIPCION (
    ID_FINSCRIPCION  INTEGER          not null,
    ID_PLANIFICACION INTEGER          null,
    FI_FECHA_INICIO  TIMESTAMP WITHOUT TIME ZONE      not
null,
    FI_FECHA_FIN     TIMESTAMP WITHOUT TIME ZONE      not null,
    FI_ETADO         BOOL             not null,
    constraint PK_FECHA_INSCRIPCION primary key (ID_FINSCRIPCION));
comment on table FECHA_INSCRIPCION is'Fechas las fechas de inscripción
para la planificación del curso.';
comment on column FECHA_INSCRIPCION.ID_FINSCRIPCION is 'Clave
primaria.';
comment on column FECHA_INSCRIPCION.FI_FECHA_INICIO is'Fecha de
inicio de la inscripción.';
comment on column FECHA_INSCRIPCION.FI_FECHA_FIN is 'Fecha de
finalización de la inscripción.';
comment on column FECHA_INSCRIPCION.FI_ESTADO is'True.- Activo.
False.- Inactivo';
/*=====*/

```

```

/* Index: FECHA_INSCRIPCION_PK                                */
/*=====*/
create unique index FECHA_INSCRIPCION_PK on FECHA_INSCRIPCION (
ID_FINSCRIPCION);
/*=====*/
/* Index: RELATIONSHIP_26_FK                                  */
/*=====*/
create index RELATIONSHIP_26_FK on FECHA_INSCRIPCION (
ID_PLANIFICACION );
/*=====*/
/* Table: FORMA_PAGO                                         */
/*=====*/
create table FORMA_PAGO ( ID_FRMPAGO      INTEGER      not
null, FRMP_NOMBRE      VARCHAR(150)      not null, FRMP_FECHA
TIMESTAMP WITHOUT TIME ZONE not null, constraint PK_FORMA_PAGO
primary key (ID_FRMPAGO));
comment on table FORMA_PAGO is 'Almacenas las formas de pago para las
inscripciones, estos datos son comunes para todos los centros de capacitación.';
comment on column FORMA_PAGO.ID_FRMPAGO is 'Clave primaris';
comment on column FORMA_PAGO.FRMP_FECHA is 'Fecha de creación del
registro.';
/*=====*/
/* Index: FORMA_PAGO_PK                                      */
/*=====*/
create unique index FORMA_PAGO_PK on FORMA_PAGO (ID_FRMPAGO);
/*=====*/
/* Table: INSCRIPCION                                         */
/*=====*/
create table INSCRIPCION (
ID_INSCRIPCION  INTEGER      not null,
ID_BENEFICIO    INTEGER      null,
ID_PERSONA      INTEGER      null,

```

```

ID_PLANIFICACION  INTEGER          null,
ID_FRMPAGO        INTEGER          null,
ID_PLAULA         INTEGER          null,
IN_FECHA          TIMESTAMP WITHOUT TIME ZONE  not null,
IN_TRABAJA       BOOL            null,
IN_EMPRESAT      VARCHAR(25)      null,
IN_CARGOT        VARCHAR(30)      null,
IN_TELEFONOT     VARCHAR(7)       null,
IN_RAZONSOCIALF  VARCHAR(150)     null,
IN_CEDULAF       VARCHAR(13)      null,
IN_DIRECCIONF    VARCHAR(250)     null,
IN_TELEFONOF     VARCHAR(7)       null,
IN_NOMBRECONTACTO VARCHAR(100)    null,
IN_TELEFONOCONTACTO VARCHAR(10)    null,
IN_ESTADO        BOOL            not null,
IN_PAGO_COMPLETO  BOOL            not null,
IN_DESCUENTO     NUMERIC(10,2)    null,
IN_COSTO_CURSO   NUMERIC(10,2)    null,
IN_TOTAL_PAGAR   NUMERIC(10,2)    null,
IN_POR_PAGAR     NUMERIC(10,2)    null,
IN_APROBADO      BOOL            null,

```

constraint PK_INSCRIPCION primary key (ID_INSCRIPCION));

comment on table INSCRIPCION is 'Almacena la información de las inscripciones.';

comment on column INSCRIPCION.ID_FRMPAGO is 'Clave primaris';

comment on column INSCRIPCION.ID_PLAULA is 'Clave primaria.';

comment on column INSCRIPCION.IN_FECHA is 'Fecha de inscripción de la persona.';

comment on column INSCRIPCION.IN_TRABAJA is 'Define si la persona que se inscribe trabaja.';

comment on column INSCRIPCION.IN_EMPRESAT is 'Empresa en la que trabaja la persona.';

comment on column INSCRIPCION.IN_CARGOT is 'Cargo que emplea la persona en su trabajo.';

comment on column INSCRIPCION.IN_TELEFONOT is 'Telefono del trabajo.';

comment on column INSCRIPCION.IN_RAZONSOCIALF is 'Nombre de la persona o empresa a la que saldra la factura.';

comment on column INSCRIPCION.IN_CEDULAF is 'Cédula o ruc de la persona o empresa de facturación.';

comment on column INSCRIPCION.IN_DIRECCIONF is 'Dirección de emisión de la factura.';

comment on column INSCRIPCION.IN_TELEFONOF is 'Telefono de facturación';

comment on column INSCRIPCION.IN_NOMBRECONTACTO is 'Nombre de contacto.';

comment on column INSCRIPCION.IN_TELEFONOCONTACTO is 'Telefono de contacto.';

comment on column INSCRIPCION.IN_ESTADO is 'True.- Activo. False.- Inactivo';

comment on column INSCRIPCION.IN_PAGO_COMPLETO is 'True.- Realizó el pago total del curso.';

comment on column INSCRIPCION.IN_DESCUENTO is 'Descuento aplicado en la inscripción.';

comment on column INSCRIPCION.IN_COSTO_CURSO is 'Costo original del curso.';

comment on column INSCRIPCION.IN_TOTAL_PAGAR is 'Total a cancelar del curso.';

comment on column INSCRIPCION.IN_APROBADO is 'True. Si la persona aprobo el curso.';

/*=====*/

/* Index: INSCRIPCION_PK */

/*=====*/

create unique index INSCRIPCION_PK on INSCRIPCION (ID_INSCRIPCION);

/*=====*/

```

/* Index: PERSONA_INSCRIPCION_FK          */
/*=====*/
create      index  PERSONA_INSCRIPCION_FK  on  INSCRIPCION
(ID_PERSONA);
/*=====*/
/* Index: PLANIFICACION_INSCRIPCION_FK    */
/*=====*/
create index PLANIFICACION_INSCRIPCION_FK on INSCRIPCION (
ID_PLANIFICACION );
/*=====*/
/* Index: RELATIONSHIP_19_FK              */
/*=====*/
create index RELATIONSHIP_19_FK on INSCRIPCION (ID_BENEFICIO);
/*=====*/
/* Index: RELATIONSHIP_20_FK              */
/*=====*/
create index RELATIONSHIP_20_FK on INSCRIPCION ( ID_FRMPAGO);
/*=====*/
/* Index: RELATIONSHIP_28_FK              */
/*=====*/
create index RELATIONSHIP_28_FK on INSCRIPCION (
ID_PLAULA);
/*=====*/
/* Table: MODULO                          */
/*=====*/
create table MODULO (
    ID_MODULO      INTEGER          not null,
    ID_CURSO       INTEGER          null,
    MD_NOMBRE      VARCHAR(250)     not null,
    MD_DESCRIPCION VARCHAR(500)     null,
    constraint PK_MODULO primary key (ID_MODULO));

```

comment on table MODULO is 'Almacena la información de los módulos o temas del curso.';

comment on column MODULO.ID_MODULO is 'Clave primaria.';

comment on column MODULO.MD_NOMBRE is 'Nombre del módulo.';

comment on column MODULO.MD_DESCRIPCION is 'Descripción del módulo.';

```
/*=====*/
```

```
/* Index: MODULO_PK */
```

```
/*=====*/
```

```
create unique index MODULO_PK on MODULO ( ID_MODULO);
```

```
/*=====*/
```

```
/* Index: RELATIONSHIP_27_FK */
```

```
/*=====*/
```

```
create index RELATIONSHIP_27_FK on MODULO (D_CURSO);
```

```
/*=====*/
```

```
/* Table: NIVEL_ACADEMICO */
```

```
/*=====*/
```

```
create table NIVEL_ACADEMICO (
    ID_ACADEMICO    INTEGER        not null,
    NA_NOMBRE       VARCHAR(250)   not null,
    constraint PK_NIVEL_ACADEMICO primary key (ID_ACADEMICO));
```

comment on table NIVEL_ACADEMICO is

'Almacena la información del nivel académico.';

comment on column NIVEL_ACADEMICO.NA_NOMBRE is

'Almacena el nombre del nivel académico.';

```
/*=====*/
```

```
/* Index: NIVEL_ACADEMICO_PK */
```

```
/*=====*/
```

```
create unique index NIVEL_ACADEMICO_PK on NIVEL_ACADEMICO (
    ID_ACADEMICO
```

```
);
```

```
/*=====*/
```

```

/* Table: NOTAS                                     */
/*=====*/
create table NOTAS (
  ID_NOTAS      INTEGER      not null,
  ID_INSCRIPCION  INTEGER      null,
  NT_FECHA      TIMESTAMP WITHOUT TIME ZONE      not null,
  NT_NOTA       NUMERIC(5,2)  not null,
  NT_MOTIVO     VARCHAR(250)  not null,
  constraint PK_NOTAS primary key (ID_NOTAS));
comment on table NOTAS is
'Registra las notas de las personas que se inscribieron en el curso.';
comment on column NOTAS.ID_NOTAS is
'Clave primaria.';
comment on column NOTAS.NT_FECHA is
'Fecha de registro de la nota.';
comment on column NOTAS.NT_NOTA is
'Registra la nota del alumno.';
comment on column NOTAS.NT_MOTIVO is
'Registra el motivo de la nota.';
/*=====*/
/* Index: NOTAS_PK                                   */
/*=====*/
create unique index NOTAS_PK on NOTAS (
  ID_NOTAS);
/*=====*/
/* Index: RELATIONSHIP_31_FK                         */
/*=====*/
create index RELATIONSHIP_31_FK on NOTAS (
  ID_INSCRIPCION);
/*=====*/
/* Table: PAGOS                                     */
/*=====*/

```



```

create table PAGOS (
  ID_PAGO      INTEGER      not null,
  ID_INSCRIPCION  INTEGER      null,
  PG_CUOTA     MONEY        not null,
  PG_SALDO     MONEY        not null,
  PG_TOTAL     MONEY        not null,
  PG_FECHA     TIMESTAMP WITHOUT TIME ZONE      not null,
  constraint PK_PAGOS primary key (ID_PAGO));

comment on table PAGOS is
'Almacena la información de las fechas de pagos del curso.';
comment on column PAGOS.PG_CUOTA is
'Cantidad de dinero calcelado.';
comment on column PAGOS.PG_SALDO is
'Monto faltante de pago. Total del curso - cuota.';
comment on column PAGOS.PG_TOTAL is
'Costo total del curso para la persona.';
comment on column PAGOS.PG_FECHA is
'Fecha de pago';

/*=====*/
/* Index: PAGOS_PK                               */
/*=====*/

create unique index PAGOS_PK on PAGOS (
  ID_PAGO);

/*=====*/
/* Index: RELATIONSHIP_29_FK                       */
/*=====*/

create index RELATIONSHIP_29_FK on PAGOS (
  ID_INSCRIPCION);

/*=====*/
/* Table: PARROQUIA                                */
/*=====*/

create table PARROQUIA (

```

```

ID_PARROQUIA    INTEGER        not null,
ID_CANTON      INTEGER        null,
PA_NOMBRE      VARCHAR(250)    not null,
constraint PK_PARROQUIA primary key (ID_PARROQUIA));
comment on table PARROQUIA is
'Almacena la información de las parroquias.';
/*=====*/
/* Index: PARROQUIA_PK                                */
/*=====*/
create unique index PARROQUIA_PK on PARROQUIA (
ID_PARROQUIA);
/*=====*/
/* Index: CANTON_PARROQUIA_FK                        */
/*=====*/
create index CANTON_PARROQUIA_FK on PARROQUIA (
ID_CANTON);
/*=====*/
/* Table: PERFIL                                    */
/*=====*/

create table PERFIL (
ID_PERFIL      INTEGER        not null,
PE_NOMBRE      VARCHAR(250)    not null,
PE_PERFIL      VARCHAR(50)     not null,
PE_ESTADO      BOOL           not null,
constraint PK_PERFIL primary key (ID_PERFIL));
comment on table PERFIL is
'Almacena la información de los perfiles de usuarios.';
comment on column PERFIL.ID_PERFIL is
'Clave primaria.';
comment on column PERFIL.PE_NOMBRE is
'Nombre del perfil.';

```

```

comment on column PERFIL.PE_PERFIL is
'Lista de perfiles necesarios para la seguridad de Spring Security.';
comment on column PERFIL.PE_ESTADO is
'True.- Activo. False.- Inactivo';
/*=====*/
/* Index: PERFIL_PK */
/*=====*/
create unique index PERFIL_PK on PERFIL (
ID_PERFIL
);
/*=====*/
/* Table: PERSONA */
/*=====*/
create table PERSONA (
ID_PERSONA      INTEGER      not null,
ID_CIVIL        INTEGER      null,
ID_PERFIL       INTEGER      null,
ID_ACADEMICO   INTEGER      null,
ID_PARROQUIA   INTEGER      null,
PE_NOMBRES     VARCHAR(250)  not null,
PE_CEDULA      VARCHAR(15)   null,
PE_DIRECCION   VARCHAR(300)  null,
PE_TELEFONO    VARCHAR(9)    null,
PE_MOVIL       VARCHAR(10)   null,
PE_CORREO      VARCHAR(50)   null,
PE_NACIONALIDAD VARCHAR(100)  null,
PE_TIPO        INTEGER      not null,
PE_CLAVE       VARCHAR(150)  null,
PE_ESTADO      BOOL         null,
VERSION        INTEGER      null,
PE_GENERO      VARCHAR(10)   null,

```

```

    PE_FECHANACIMIENTO    TIMESTAMP WITHOUT TIME ZONE
null,
    PE_LUGARNACIMIENTO    VARCHAR(100)    null,
    PE_FOTO                CHAR(254)    null,
    PE_FECHAREG            TIMESTAMP WITHOUT TIME ZONE    null,
    PE_UNIDADE             VARCHAR(150)    null,
    PE_NIVELE              INTEGER    null,
    PE_TRABAJA             BOOL    null,
    constraint PK_PERSONA primary key (ID_PERSONA));
comment on table PERSONA is
'Almacena los datos de los estudiantes y docentes';
comment on column PERSONA.ID_PERFIL is
'Clave primaria.';
comment on column PERSONA.PE_NOMBRES is
'Almacena la información de los nombres de los estudiantes o docentes.';
comment on column PERSONA.PE_CEDULA is
'Almacena la información de la cédula o pasaporte.';
comment on column PERSONA.PE_DIRECCION is
'Almacena la infomración de la dirección del estudiante o docente.';
comment on column PERSONA.PE_TELEFONO is
'Almacena la información del teléfono convencional del estudiante o docente.';
comment on column PERSONA.PE_MOVIL is
'Almacena la información del número de celular.';
comment on column PERSONA.PE_CORREO is
'Almacena la información de la cuenta de correo del estudiante o docente.';
comment on column PERSONA.PE_NACIONALIDAD is
'Almacena la nacionalidad del estudiante o docente.';
comment on column PERSONA.PE_TIPO is
'Almacena la información del tipo de persona. 1.- Estudiante. 2.- Docente.';
comment on column PERSONA.PE_ESTADO is
'True.- Activo, False.- Inactivo';
comment on column PERSONA.PE_GENERO is

```

```

'M.- Masculino. F.- Femenino';
comment on column PERSONA.PE_FECHAREG is
'Fecha de creacion del registro.';
comment on column PERSONA.PE_UNIDADE is
'Unidad educativa del ultimo nivel academico.';
comment on column PERSONA.PE_NIVELE is
'Año de estudio del último nivel académico.';
comment on column PERSONA.PE_TRABAJA is
'True.- Trabaja. False.- No trabaja';

/*=====*/
/* Index: PERSONA_PK                                */
/*=====*/

create unique index PERSONA_PK on PERSONA (
ID_PERSONA
);

/*=====*/
/* Index: PARROQUIA_PERSONA_FK                      */
/*=====*/

create index PARROQUIA_PERSONA_FK on PERSONA (
ID_PARROQUIA);

/*=====*/
/* Index: ESTADOCIV_PERSONA_FK                      */
/*=====*/

create index ESTADOCIV_PERSONA_FK on PERSONA (
ID_CIVIL);

/*=====*/
/* Index: NIVELAC_PERSONA_FK                        */
/*=====*/

create index NIVELAC_PERSONA_FK on PERSONA (
ID_ACADEMICO);

```

```

/*=====*/
/* Index: PERFIL_PERSONA_FK */
/*=====*/
create index PERFIL_PERSONA_FK on PERSONA (
ID_PERFIL);
/*=====*/
/* Table: PERSONA_CENTRO */
/*=====*/
create table PERSONA_CENTRO (
    ID_PCENTRO    INTEGER    not null,
    ID_PERSONA    INTEGER    null,
    ID_CENTRO     INTEGER    null,
    PC_FECHA_REGISTRO  TIMESTAMP WITHOUT TIME ZONE    not
null,
    PC_FECHA_MOD   TIMESTAMP WITHOUT TIME ZONE    null,
    VERSION        INTEGER    null,
    PC_ESTADO      BOOL      not null,
    constraint PK_PERSONA_CENTRO primary key (ID_PCENTRO));
comment on table PERSONA_CENTRO is
'Almacena la informacion de las personas asignadas a los centros de capacitacion.';
comment on column PERSONA_CENTRO.PC_FECHA_REGISTRO is
'Almacena la fecha de registros.';
comment on column PERSONA_CENTRO.PC_FECHA_MOD is
'Almacena la fecha de modificacion.';
/*=====*/
/* Index: PERSONA_CENTRO_PK */
/*=====*/
create unique index PERSONA_CENTRO_PK on PERSONA_CENTRO (
ID_PCENTRO);
/*=====*/
/* Index: RELATIONSHIP_21_FK */
/*=====*/

```

```

create index RELATIONSHIP_21_FK on PERSONA_CENTRO (
ID_CENTRO);
/*=====*/
/* Index: RELATIONSHIP_22_
create index RELATIONSHIP_22_FK on PERSONA_CENTRO (
ID_PERSONA);
/*=====*/
/* Table: PLANIFICACION */
/*=====*/
create table PLANIFICACION (
    ID_PLANIFICACION    INTEGER           not null,
    ID_PARROQUIA        INTEGER           null,
    ID_CURSO             INTEGER           null,
    ID_CENTRO           INTEGER           null,
    PL_FECHAI           TIMESTAMP WITHOUT TIME ZONE    not null,
    PL_FECHAF           TIMESTAMP WITHOUT TIME ZONE    not null,
    PL_OBSERVACION      VARCHAR(250)       null,
    PL_COSTO            NUMERIC(10,2)      null,
    PL_HORAS            INTEGER           null,
    PL_ESTADO           BOOL              null,
    PL_CANITDAD_MINIMA  INTEGER           null,
    constraint PK_PLANIFICACION primary key (ID_PLANIFICACION));
comment on table PLANIFICACION is
'Almacena la información de la planificación de los cursos.';
comment on column PLANIFICACION.PL_FECHAI is
'Fecha de inicio del curso.';
comment on column PLANIFICACION.PL_FECHAF is
'Fecha de finalización del curso.';
comment on column PLANIFICACION.PL_COSTO is
'Almacena la información del costo del curso.';
comment on column PLANIFICACION.PL_HORAS is
'Almacena el número de horas del curso.';

```

```

comment on column PLANIFICACION.PL_ESTADO is
'True.- Activo. False.- Inactivo';
comment on column PLANIFICACION.PL_CANITDAD_MINIMA is
'Cantidad minima de personas que deben inscribirse.';
/*=====*/
/* Index: PLANIFICACION_PK */
/*=====*/
create unique index PLANIFICACION_PK on PLANIFICACION (
ID_PLANIFICACION);
/*=====*/
/* Index: PARROQUIA_PLANIFICACION_FK */
/*=====*/
create index PARROQUIA_PLANIFICACION_FK on PLANIFICACION (
ID_PARROQUIA);
/*=====*/
/* Index: CURSO_PLANIFICACION_FK */
/*=====*/
create index CURSO_PLANIFICACION_FK on PLANIFICACION (
ID_CURSO);
/*=====*/
/* Index: CENTRO_PLANIFICACION_FK */
/*=====*/
create index CENTRO_PLANIFICACION_FK on PLANIFICACION (
ID_CENTRO);
/*=====*/
/* Table: PLAN_AULA */
/*=====*/
create table PLAN_AULA (
    ID_PLAULA      INTEGER          not null,
    ID_AULA        INTEGER          null,
    ID_PLANIFICACION  INTEGER          null,
    PLA_FECHA_INICIO  TIMESTAMP WITHOUT TIME ZONE  null,

```



```

PLA_FECHA_FIN    TIMESTAMP WITHOUT TIME ZONE    null,
PLA_HORA_INICIO  TIME                not null,
PLA_HORA_FIN     TIME                not null,
constraint PK_PLAN_AULA primary key (ID_PLAULA));
comment on table PLAN_AULA is
'Almacena la informacion de las aulas asignadas a la planificacion.';
comment on column PLAN_AULA.ID_PLAULA is
'Clave primaria.';
comment on column PLAN_AULA.ID_AULA is
'Clave primaria';
comment on column PLAN_AULA.PLA_FECHA_INICIO is
'Fecha de inicio de la planificacion.';
comment on column PLAN_AULA.PLA_FECHA_FIN is
'Fecha de fin de la planificacion.';
comment on column PLAN_AULA.PLA_HORA_INICIO is
'Hora de inicio de clases.';
comment on column PLAN_AULA.PLA_HORA_FIN is
'Hora de fin de clases.';
/*=====*/
/* Index: PLAN_AULA_PK                */
/*=====*/
create unique index PLAN_AULA_PK on PLAN_AULA (
ID_PLAULA);
/*=====*/
/* Index: RELATIONSHIP_24_FK          */
/*=====*/
create index RELATIONSHIP_24_FK on PLAN_AULA (
ID_AULA);
/*=====*/
/* Index: RELATIONSHIP_25_FK          */
/*=====*/
create index RELATIONSHIP_25_FK on PLAN_AULA (

```

```

ID_PLANIFICACION);
/*=====*/
/* Table: PROFESOR_CURSO */
/*=====*/
create table PROFESOR_CURSO (
  ID_PROFCURSO    INTEGER    not null,
  ID_PERSONA      INTEGER    null,
  ID_PLANIFICACION  INTEGER    null,
  constraint PK_PROFESOR_CURSO primary key (ID_PROFCURSO));
comment on table PROFESOR_CURSO is
'Almacena la información de los profesores asignados a un curso.';
/*=====*/
/* Index: PROFESOR_CURSO_PK */
/*=====*/
create unique index PROFESOR_CURSO_PK on PROFESOR_CURSO (
  ID_PROFCURSO
/*=====*/
/* Index: PERSONA_PROFCURSO_FK */
/*=====*/
create index PERSONA_PROFCURSO_FK on PROFESOR_CURSO (
  ID_PERSONA);
/*=====*/
/* Index: PLANIFICACION_PROFCURSO_FK */
/*=====*/
create index PLANIFICACION_PROFCURSO_FK on PROFESOR_CURSO (
  ID_PLANIFICACION);
/*=====*/
/* Table: PROVINCIA */
/*=====*/
create table PROVINCIA (
  ID_PROVINCIA    INTEGER    not null,
  PR_NOMBRE       VARCHAR(250) not null,

```

```

constraint PK_PROVINCIA primary key (ID_PROVINCIA));
comment on table PROVINCIA is
'Almacena la información de las provincias.';
/*=====*/
/* Index: PROVINCIA_PK                */
/*=====*/
create unique index PROVINCIA_PK on PROVINCIA (
ID_PROVINCIA);
/*=====*/
/* Table: TITULOS                      */
/*=====*/
create table TITULOS (
  ID_TITULOS      INTEGER      not null,
  ID_PERSONA      INTEGER      null,
  ID_INSCRIPCION  INTEGER      null,
  TI_NOMBRE       VARCHAR(250) not null,
  TI_ENTIDAD      VARCHAR(250) null,
  TI_HORAS        INTEGER      null,
  TI_FECHA        TIMESTAMP WITHOUT TIME ZONE null,
  constraint PK_TITULOS primary key (ID_TITULOS));
comment on table TITULOS is
'Almacena la información de los títulos o certificados de estudios. Esta información
es solo para el tipo de persona docente.';
comment on column TITULOS.TI_NOMBRE is
'Nombre del título o certificado.';
comment on column TITULOS.TI_ENTIDAD is
'Entidad en la que obtuvo el título o certificado.';
comment on column TITULOS.TI_HORAS is
'Número de horas para obtener el certificado.';
comment on column TITULOS.TI_FECHA is
'Fecha en la que se obtiene el título.';
/*=====*/

```

```

/* Index: TITULOS_PK                                */
/*=====*/
create unique index TITULOS_PK on TITULOS (
ID_TITULOS);
/*=====*/
/* Index: PERSONA_TITULOS_FK                        */
/*=====*/
create index PERSONA_TITULOS_FK on TITULOS (
ID_PERSONA);
/*=====*/
/* Index: INSCRIPCION_TITULOS_FK                    */
/*=====*/
create index INSCRIPCION_TITULOS_FK on TITULOS (
ID_INSCRIPCION);
alter table ASISTENCIA
  add constraint FK_ASISTENC_RELATIONS_INSCRIPC foreign key
(ID_INSCRIPCION)
  references INSCRIPCION (ID_INSCRIPCION)
  on delete restrict on update restrict;
alter table AULA
  add constraint FK_AULA_RELATIONS_CENTRO foreign key
(ID_CENTRO)
  references CENTRO (ID_CENTRO)
  on delete restrict on update restrict;
alter table BENEFICIO
  add constraint FK_BENEFICI_RELATIONS_BENEFICI foreign key
(ID_BENEFICIARIO)
  references BENEFICIARIO (ID_BENEFICIARIO)
  on delete restrict on update restrict;
alter table BENEFICIO
  add constraint FK_BENEFICI_RELATIONS_CENTRO foreign key
(ID_CENTRO)

```

```
references CENTRO (ID_CENTRO)
on delete restrict on update restrict;
alter table CANTON
add constraint FK_CANTON_PROVINCIA_PROVINCI foreign key
(ID_PROVINCIA)
references PROVINCIA (ID_PROVINCIA)
on delete restrict on update restrict;
alter table CENTRO
add constraint FK_CENTRO_PARROQUIA_PARROQUI foreign key
(ID_PARROQUIA)
references PARROQUIA (ID_PARROQUIA)
on delete restrict on update restrict;
alter table FECHA_INSCRIPCION
add constraint FK_FECHA_IN_RELATIONS_PLANIFIC foreign key
(ID_PLANIFICACION)
references PLANIFICACION (ID_PLANIFICACION)
on delete restrict on update restrict;
alter table INSCRIPCION
add constraint FK_INSCRIPC_PERSONA_I_PERSONA foreign key
(ID_PERSONA)
references PERSONA (ID_PERSONA)
on delete restrict on update restrict;
alter table INSCRIPCION
add constraint FK_INSCRIPC_PLANIFICA_PLANIFIC foreign key
(ID_PLANIFICACION)
references PLANIFICACION (ID_PLANIFICACION)
on delete restrict on update restrict;
alter table INSCRIPCION
add constraint FK_INSCRIPC_RELATIONS_BENEFICI foreign key
(ID_BENEFICIO)
references BENEFICIO (ID_BENEFICIO)
on delete restrict on update restrict;
```

```
alter table INSCRIPCION
  add constraint FK_INSCRIPC_RELATIONS_FORMA_PA foreign key
(ID_FRMPAGO)
  references FORMA_PAGO (ID_FRMPAGO)
  on delete restrict on update restrict;
alter table INSCRIPCION
  add constraint FK_INSCRIPC_RELATIONS_PLAN_AUL foreign key
(ID_PLAULA)
  references PLAN_AULA (ID_PLAULA)
  on delete restrict on update restrict;
alter table MODULO
  add constraint FK_MODULO_RELATIONS_CURSO foreign key
(ID_CURSO)
  references CURSO (ID_CURSO)
  on delete restrict on update restrict;
alter table NOTAS
  add constraint FK_NOTAS_RELATIONS_INSCRIPC foreign key
(ID_INSCRIPCION)
  references INSCRIPCION (ID_INSCRIPCION)
  on delete restrict on update restrict;
alter table PAGOS
  add constraint FK_PAGOS_RELATIONS_INSCRIPC foreign key
(ID_INSCRIPCION)
  references INSCRIPCION (ID_INSCRIPCION)
  on delete restrict on update restrict;
alter table PARROQUIA
  add constraint FK_PARROQUI_CANTON_PA_CANTON foreign key
(ID_CANTON)
  references CANTON (ID_CANTON)
  on delete restrict on update restrict;
alter table PERSONA
```

```
add constraint FK_PERSONA_ESTADOCIV_ESTADO_C foreign key
(ID_CIVIL)
references ESTADO_CIVIL (ID_CIVIL)
on delete restrict on update restrict;
alter table PERSONA
add constraint FK_PERSONA_NIVELAC_P_NIVEL_AC foreign key
(ID_ACADEMICO)
references NIVEL_ACADEMICO (ID_ACADEMICO)
on delete restrict on update restrict;
alter table PERSONA
add constraint FK_PERSONA_PARROQUIA_PARROQUI foreign key
(ID_PARROQUIA)
references PARROQUIA (ID_PARROQUIA)
on delete restrict on update restrict;
alter table PERSONA
add constraint FK_PERSONA_PERFIL_PE_PERFIL foreign key
(ID_PERFIL)
references PERFIL (ID_PERFIL)
on delete restrict on update restrict;
alter table PERSONA_CENTRO
add constraint FK_PERSONA__RELATIONS_CENTRO foreign key
(ID_CENTRO)
references CENTRO (ID_CENTRO)
on delete restrict on update restrict;
alter table PERSONA_CENTRO
add constraint FK_PERSONA__RELATIONS_PERSONA foreign key
(ID_PERSONA)
references PERSONA (ID_PERSONA)
on delete restrict on update restrict;
alter table PLANIFICACION
add constraint FK_PLANIFIC_CENTRO_PL_CENTRO foreign key
(ID_CENTRO)
```

```
references CENTRO (ID_CENTRO)
on delete restrict on update restrict;
alter table PLANIFICACION
add constraint FK_PLANIFIC_CURSO_PLA_CURSO foreign key
(ID_CURSO)
references CURSO (ID_CURSO)
on delete restrict on update restrict;
alter table PLANIFICACION
add constraint FK_PLANIFIC_PARROQUIA_PARROQUI foreign key
(ID_PARROQUIA)
references PARROQUIA (ID_PARROQUIA)
on delete restrict on update restrict;
alter table PLAN_AULA
add constraint FK_PLAN_AUL_RELATIONS_AULA foreign key
(ID_AULA)
references AULA (ID_AULA)
on delete restrict on update restrict;
alter table PLAN_AULA
add constraint FK_PLAN_AUL_RELATIONS_PLANIFIC foreign key
(ID_PLANIFICACION)
references PLANIFICACION (ID_PLANIFICACION)
on delete restrict on update restrict;
alter table PROFESOR_CURSO
add constraint FK_PROFESOR_PERSONA_P_PERSONA foreign key
(ID_PERSONA)
references PERSONA (ID_PERSONA)
on delete restrict on update restrict;
alter table PROFESOR_CURSO
add constraint FK_PROFESOR_PLANIFICA_PLANIFIC foreign key
(ID_PLANIFICACION)
references PLANIFICACION (ID_PLANIFICACION)
on delete restrict on update restrict;
```



```
alter table TITULOS
  add constraint FK_TITULOS_INSCRIPCION foreign key
  (ID_INSCRIPCION) references INSCRIPCION (ID_INSCRIPCION)
  on delete restrict on update restrict; alter table TITULOS
  add constraint FK_TITULOS_PERSONA_T_PERSONA foreign key
  (ID_PERSONA) references PERSONA (ID_PERSONA) on delete restrict on update
  restrict;
```

12. En la ventana de ejecución de comandados ejecute el script con los datos iniciales del sistema.

4.1.1.3 Instalación de Glassfish

4.1.1.3.1 Instalación de Glassfish

Se trabaja con la versión 4 del Glassfish, la cual puede ser descargado de la siguiente url: <https://glassfish.java.net/download.html>

La instalación de Glassfish se describe en la figura 50:

1. Se descarga el archivo zip y guarde en el disco duro.
2. Se descomprime el archivo glassfish-4.0.zip en la raíz del disco duro.

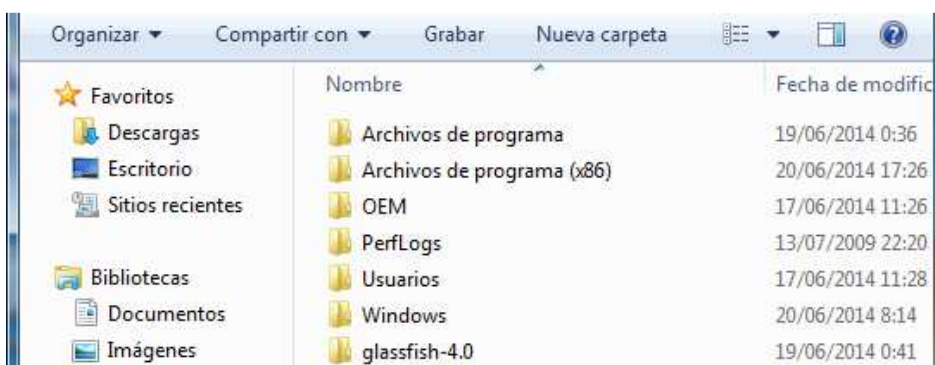


Figura 50. Pantalla de descompresión del Archivo Glassfish

3. Abra la consola de comandos y ubíquese en C:/glassfish-4.0/glassfish4/bin, tal y como se muestra en la figura 51.

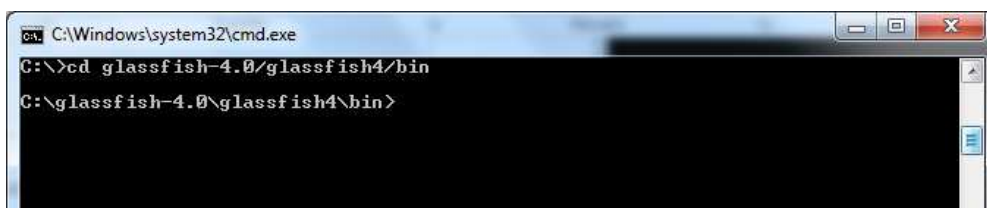
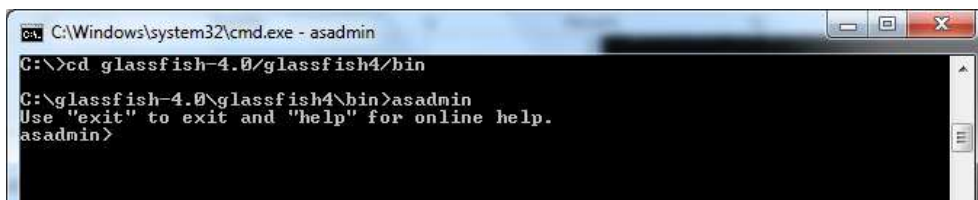


Figura 51. Pantalla de Consola de Comandos

4. Ejecuta comando asadmin para proceder con la configuración del dominio de producción, tal y como se muestra en la figura 52.



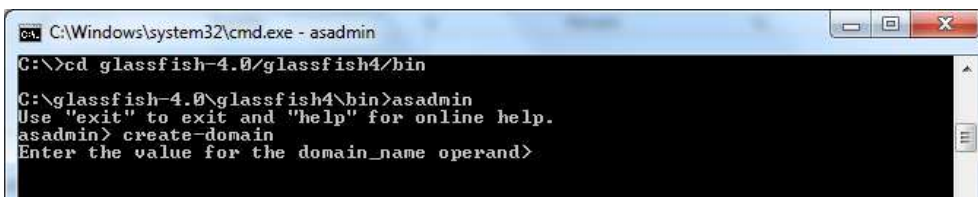
```

C:\Windows\system32\cmd.exe - asadmin
C:\>cd glassfish-4.0\glassfish4\bin
C:\glassfish-4.0\glassfish4\bin>asadmin
Use "exit" to exit and "help" for online help.
asadmin>

```

Figura 52. Pantalla de Configuración del Dominio

5. Se ejecuta el comando create-domain, tal y como se muestra en la figura 53.



```

C:\Windows\system32\cmd.exe - asadmin
C:\>cd glassfish-4.0\glassfish4\bin
C:\glassfish-4.0\glassfish4\bin>asadmin
Use "exit" to exit and "help" for online help.
asadmin> create-domain
Enter the value for the domain_name operand>

```

Figura 53. Pantalla del Comando Create Domain

Al crear el dominio ingrese los siguientes datos:

- Nombre del dominio.- CTT
- Nombre del administrador.- adminCTT
- Clave del administrado.- CTT2014

Los puertos creados para el dominio son los siguientes:

- Puerto de administración.- 4848
- Puerto de despliegue.- 8080
- Se levanta al servidor con el comando start-domain ctt

4.1.1.3.2 Configuración de Glassfish

Es necesario configurar el acceso a la base de datos desde el servidor Glassfish para los cual se descarga el driver de conexiónPostgres de la siguiente url:

<http://jdbc.postgresql.org/download.html>

La configuración de acceso a base de datos en Glassfish se describe a continuación:

1. Se descarga el archivo zip y guarde en el disco duro.
2. Se descomprime el archivo postgresql-9.2-1002.jdbc4.jar.zip en la ubicación C:\glassfish-4.0\glassfish4\glassfish\domains\ctt\lib, tal y como se muestra en la figura 54.

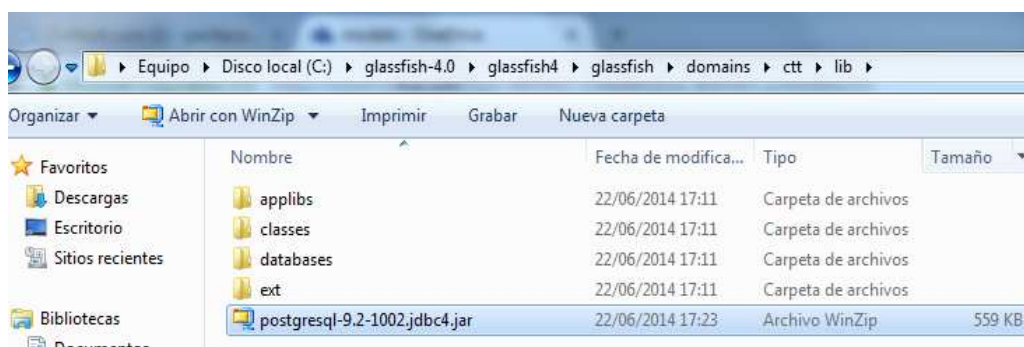


Figura 54. Pantalla de descompresión del archivo postgres

3. El siguiente paso es Levantar el dominio, para lo cual ejecute el comando start-domain ctt, se accede a la aplicación mediante el url. <http://localhost:4848/>, tal y como se muestra en la figura 55.



Figura 55. Pantalla de Acceso a la Aplicación

4. Se ingresa el usuario y clave creados anteriormente e ingresará al sistema de administración del Glassfish, tal y como se muestra en la figura 56.

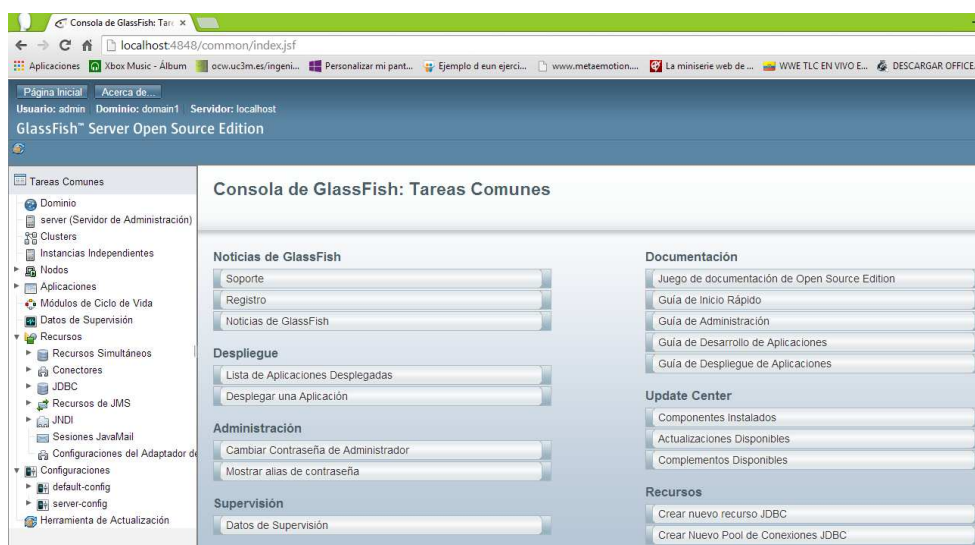


Figura 56. Pantalla de ingreso al Sistema

5. Acceda al Resources, JDBC, JDBC Conexión Pools, opción Nuevo, tal y como se muestra en la figura 57.

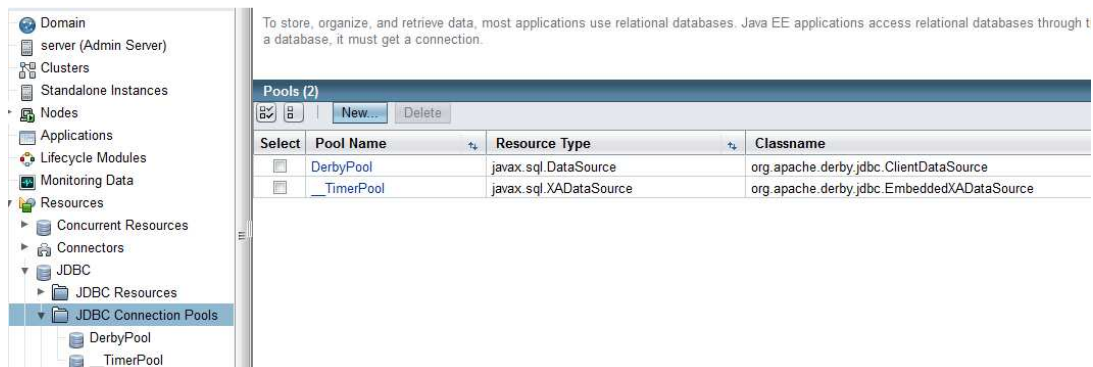


Figura 57. Pantalla de Acceso al Resources JDBC

Configure el pool de conexión con los siguientes datos, tal y como se muestra en la figura 58:

- Nombre del pool: educacionPool
- Tipo de conexión: DataSource
- Tipo de Base de datos: Postgres

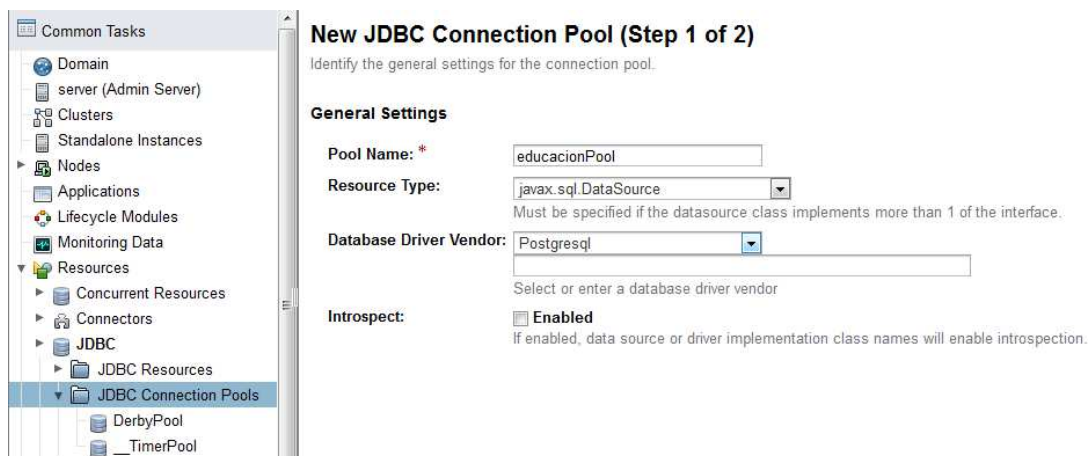


Figura 58. Pantalla del Pool de Conexión

Debe ingresar los siguientes datos, tal y como se muestra en la figura 59:

- portNumber: 5432
- databaseName: educación
- serverName: 127.0.0.1
- user: postgres
- password: postgres

Select	Name	Value	Description
<input type="checkbox"/>	portNumber	5432	
<input type="checkbox"/>	databaseName	educacion	
<input type="checkbox"/>	serverName	127.0.0.1	
<input type="checkbox"/>	user	postgres	
<input type="checkbox"/>	password	postgres	

Figura 59. Pantalla de Configuración del Pool de Conexión

Una vez creado el pool este podrá ser visualizado en la sección JDBC Connection Pool, tal y como se muestra en la figura 60.

Select	Pool Name	Resource Type	Classname	Description
<input type="checkbox"/>	DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource	
<input type="checkbox"/>	_TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource	
<input checked="" type="checkbox"/>	educacionPool	javax.sql.DataSource	org.postgresql.ds.PGSimpleDataSource	

Figura 60. Pantalla de Visualización del Pool de Conexión

Para verificar la conexión seleccione el pool creado y de clic en ping, tal y como se muestra en la figura 61.

General **Advanced** **Additional Properties**

✓ Ping Succeeded

Edit JDBC Connection Pool
Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

General Settings

Pool Name: educacionPool

Resource Type: javax.sql.DataSource

Datasource Classname: org.postgresql.ds.PGSimpleDataSource

Driver Classname:

Ping: Enabled

Figura 61. Pantalla de Verificación de la Conexión

6. Acceda al Resources, JDBC, JDBC Resource, opción Nuevo. Cree el data source educacionDS y apunte el Pool educaciónPool, tal y como se muestra en la figura 62.



Figura 62. Pantalla de Creación del Data Source

4.1.1.4 Despliegue del Sistema

Para desplegar la aplicación ubíquese en la opción Applications y seleccione deploy. Abra y seleccione el archivo educacionCTT-1.0.war que se encuentra en la carpeta Instaladores, tal y como se muestra en la figura 63.

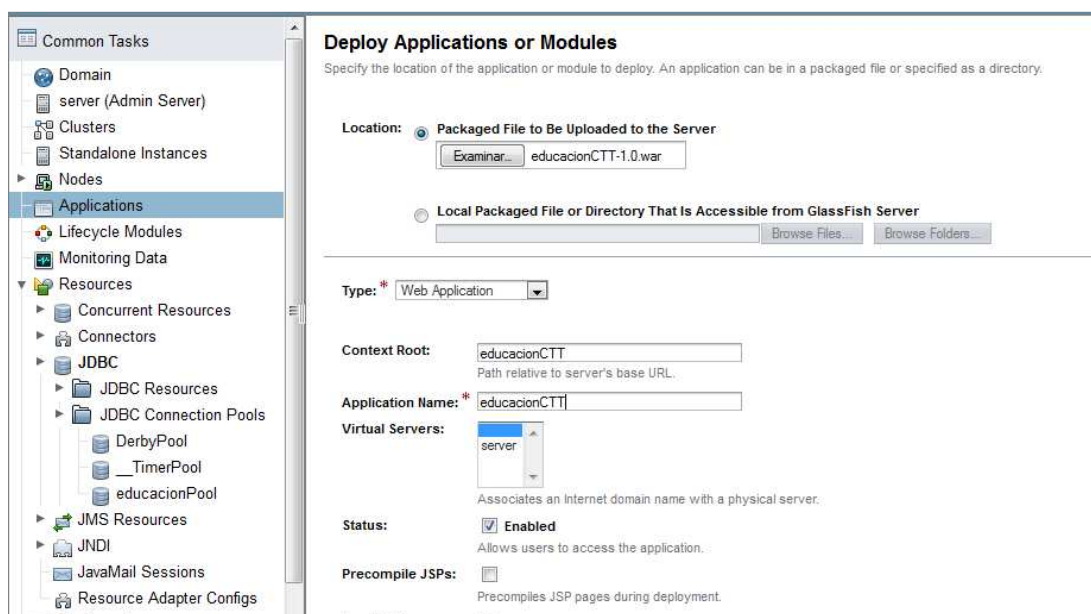


Figura 63. Pantalla de despliegue del Sistema

La aplicación se visualiza en la sección de aplicaciones y está lista para ser accedida por el cliente, tal y como se muestra en la figura 64.

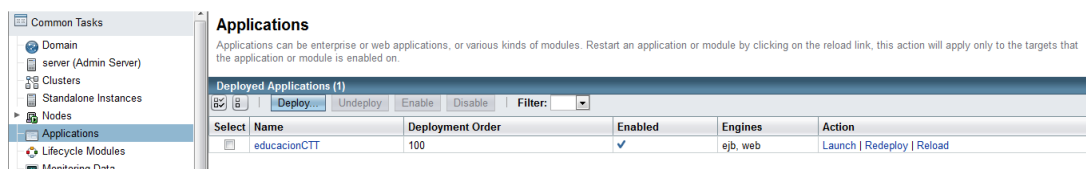


Figura 64. Pantalla de Visualización de Aplicaciones

La aplicación puede ser accedida mediante el url: <http://190.152.181.18:8080/educacionCTT/>, tal y como se muestra en la figura 65.

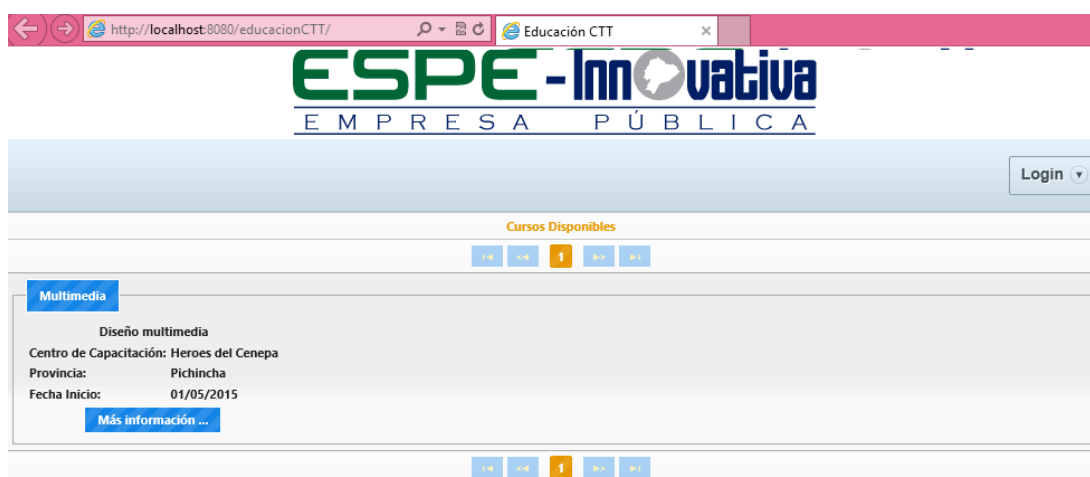


Figura 65. Acceso al Sistema de Gestión Académica

4.2 Seguimiento y ejecución del sistema de Gestión Académica

Luego de haber planificado las iteraciones se procede al seguimiento y ejecución de las tareas. Hay que tomar en cuenta, que la planificación inicial puede ser distinta a la real, debido a que las fechas de inicio y fin pueden variar durante la realización del proyecto.

En cada iteración por realizarse en el proyecto, se implementará las historias de usuario involucradas, controlando que la ejecución de estas se encuentre de acuerdo a los permisos de acceso al sistema establecido

En el transcurso de una iteración el equipo de desarrollo como primer punto debe ejecutar las tareas y concluir con el control de las mismas.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El levantamiento y análisis de requerimientos para el desarrollo de la aplicación se realizó utilizando los artefactos definidos en la metodología de desarrollo, específicamente el diseño de los casos de uso.
- Una vez identificados los procedimientos se pudo registrar la información necesaria para la generación de reportería y mejorar la toma de decisiones.
- Conforme a los requerimientos del usuario se diseñó el Sistema de gestión académica principalmente los módulos de inscripción de alumnos, registro de instructores, materias, horarios, notas y generación de certificados de aprobación del CECAI
- En la fase de implementación se utilizó el modelo XP, porque se caracteriza en la simplicidad, comunicación y retroalimentación lo que permitió la implementación del Sistema de Gestión Académica en los diferentes centros de capacitación a nivel nacional con la participación de los usuarios durante todo el desarrollo del proyecto.

5.2 RECOMENDACIONES

- Se recomienda que las especificaciones de los usuarios sean documentadas ya que es la única forma en la cual el sistema puede ser desarrollado conforme a las necesidades del usuario.
- Es necesario contar con reportes en línea en tiempo real para que las autoridades de las diferentes instituciones puedan tomar las mejores decisiones ya sean de la institución o del proceso en evaluación.
- Antes de tomar una decisión de las metodologías de desarrollo a utilizarse, se debe realizar un análisis de la magnitud alcance tiempo recursos de proyecto y con estos factores seleccionar la metodología más apropiada para involucrar de mejor manera al usuario con el desarrollador y al desarrollador con el equipo de desarrollo.

- Con la finalidad de mantener la seguridad en el software y tomando en cuenta que es de impacto nacional el modulo contiene una tabla de seguridad.
- Encaminar el uso de software libre por el ahorro en licenciamiento, por otro lado estas herramientas han evolucionado en los últimos tiempo a tal punto que existen librerías que realizan funcionalidades completa y ambientes de desarrollo muchos más ágiles.

BIBLIOGRAFÍA

LIBROS BIBLIOTECA ESPE

<http://www.extremeprogramming.org/>

<http://www.objectmentor.com/>

<http://programacion-extrema.wikispaces.com/5.+Ciclo+de+vida+y+fases>

Welles, 2002; Brewer, 2002; Jeffries, 2002; Kerievsky, 2002, y Wake, 2002. Beck, Kent (1999) *Extreme Programming Explained*. Brewer, John (2002) [en línea].

Buzan, Tony, *El Libro de los Mapas mentales*, Ediciones Urano, Barcelona (España), 1996. de Montes, Zoraida, *Mas Allá de la Educación*, Editorial Galac, Caracas (Venezuela), 1997.

<http://www.enterate.unam.mx/Articulos/2003/junio/j2ee.htm>

M. Davila. (Viernes de 8 de 2013). *Esquemas, Mapas mentales y Mapas conceptuales*.

Obtenido de <http://entornodeestrategias.blogspot.com/2013/11/esquemas-mapas-mentales-y-mapas.html>

R. Martinez. (2 de Octubre de 2009-2013). *PostgreSQL*. Obtenido de

http://www.postgresql.org.es/sobre_postgresql

Beck, K. . (2000). *Extreme Programming Explained: Embrace Change*. Obtenido de

<http://www.acit2k.org/ACIT/2013Proceedings/132.pdf>

Carmen Penadés. (Abril de 2006). *Métodologías ágiles para el desarrollo de software:*

eXtreme Programming (XP). Obtenido de <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Centro de Desarrollo Tecnológico ESPE CECAI. (24 de Junio de 2003). *Reseña*

Histórica ESPE CECAI. Obtenido de http://www.innovativa-espe.edu.ec/index.php?option=com_content&view=article&id=55&Itemid=63

3

Cockbun, A. W. (2000). *The Costs and Benefits of Pair Programming*. Obtenido de

<http://www.cyta.com.ar/ta0502/v5n2a1.htm>

D. Valverde. (06 de Septiembre de 2007). *Introducción a la Programación Extrema*.

Obtenido de <http://www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp/>

- G. Vanderburg, . (2005). *A Simple Model of Agile Software Processes Extreme Programming*. Obtenido de <http://www.acit2k.org/ACIT/2013Proceedings/132.pdf>
- Galvis, P. (17 de Septiembre de 2008). *SERVIDOR DE APLICACIONES*. Obtenido de <http://ffabiang.wordpress.com/2008/09/17/servidor-de-aplicaciones/>
- H. Cervantes. (1990). *Sistema de reconstrucción d imagns 3d a partir de cortes*. Obtenido de <http://www.humbertocervantes.net/homepage/itzamna/DOCUMENTACION/Indice.html>
- J. Joskowicz . (2008). *Reglas y Prácticas en Xtreme Programming*. Obtenido de <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- J. Montoya Guzmán. (17 de Enero de 2009). *Ejemplos de diagramas UML, interfaces gráficas de usuario, y usos del UML en la ingeniería inversa*. Obtenido de <http://www.monografias.com/trabajos67/diagramas-uml/diagramas-uml2.shtml>
- Kent Beck. (Julio de 2013). *Metodología XP*. Obtenido de <https://www.facebook.com/MetodologiaXp/posts/206740129481561>
- Londoño, T. (s.f.). *Base de Datos*. Obtenido de <http://www.monografias.com/trabajos90/base-datos-farmacia/base-datos-farmacia.shtml>
- Longman, A. W. (1999). *UML*. Obtenido de <http://jms32.eresmas.net/tacticos/UML/UML08/UML0801.html>
- Luis. (Miércoles de Noviembre de 2009). *Diagrama de Colaboración*. Obtenido de <http://tonouml.blogspot.com/2009/11/diagrama-de-colaboracion.html>
- R. Gonzalez. (2002). *Sistema informático para la administración de consulta externa de un hospital público*. Obtenido de <http://www.monografias.com/trabajos28/informatica-hospital/informatica-hospital.shtml>
- Roger S. Pressman, Joyanes L. (1997). *Ingeniería de Software: Enfoque Práctico*. Obtenido de <https://www.google.com.ec/search?q=cARACTRISTICAS+DE+LA+METODOLOGIA+XTREME+PROGRAMMING&ie=utf-8&oe=utf->

8&aq=t&rls=org.mozilla:es-MX:official&client=firefox-
 a&channel=np&source=hp&gfe_rd=cr&ei=QJRIVOLyN5PAqAXw9ICgDQ
 #rls=org.mozilla:es-MX:official&channel=n

- S. Chica Echeverr. (s.f.). *Vntajas de OpenSource*. Obtenido de <https://espaudunal20112.wikispaces.com/file/view/VENTAJAS+OPENSOURCE.pdf>
- W. Quishpe, J. Tipan. (Marzo de 2011). DESARROLLO E IMPLEMENTACION DEL SISTEMA DE LA DOCUMENTACIÓN DE LA ESCUELA POLITECNICA DEL EJERCITO EXTENSIÓN LATACUNGA. *ESCUELA POLITCNICA DE LA ESPE Repositorio Digital ESPE*. Latacunga, Cotopaxi, Ecuador. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/4462/1/M-ESPEL-0052.pdf>
- Y. Dzhurov, I. Krasteva, and S. Ilieva. (2008). *Personal Extreme Programming – An Agile Process for Autonomous Developers*. Obtenido de https://research.uni-sofia.bg/bitstream/10506/647/1/S3T2009_37_YDzhurov_IKrasteva_SIlieva.pdf