

ESCUELA POLITÉCNICA DEL EJÉRCITO

FACULTAD DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**“DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN
WEB PARA LA GESTIÓN Y PLANIFICACIÓN DE CURSOS
DE CAPACITACIÓN USANDO HERRAMIENTAS OPEN-
SOURCE”**

**CASO PRÁCTICO: *“INSTITUTO PARA EL DESARROLLO
SOCIAL Y DE LAS INVESTIGACIONES CIENTÍFICAS –
INDESIC”***

Previa la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR: MANUEL ELÍAS ASUBADÍN ASUBADÍN

SANGOLQUÍ, JUNIO 2006

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. Manuel Elías Asubadín Asubadín, como requerimiento parcial a la obtención del título de Ingeniero de Sistemas e Informática.

11 de diciembre de 2010

.....
Ing. Fernando Galárraga

DEDICATORIA

Este trabajo dedico de manera muy especial a DIOS, porque sin ÉL, nada fuese posible en esta vida.

A mi familia: a mis hijos Samay y Pablo, por quienes he dado el mejor de mis esfuerzos; a mi esposa Ligia, quien me ha acompañado durante todos mis años de estudio; y a mis padres, quienes me han apoyado de todas las formas posibles para verme realizado profesionalmente.

Manuel Elías Asubadín Asubadín

AGRADECIMIENTO

Mi agradecimiento especial para:

La Escuela Politécnica del Ejército, a la Facultad de Sistemas e Informática, en donde me han preparado profesionalmente.

Al Soc. Armin Schlegl, representante de la Fundación Hanns Seidel por su incondicional apoyo en todo el trayecto de mis estudios.

Al Instituto para el Desarrollo Social y de las Investigaciones Científicas - INDESIC, por su ayuda económica brindada durante mis años de estudio.

A mis amigos, de los misioneros IDENTES, quienes me han dado su apoyo moral y espiritual durante mi carrera. Y a todos mis amigos quienes me han apoyado de todas las formas posibles en la realización de la tesis.

A mi director y codirector de tesis, Ing. Fernando Galárraga e Ing. Wilson Jácome, que sin la guía de ellos, no hubiera podido lograr culminar este proyecto.

Manuel Elías Asubadín Asubadín

ÍNDICE DE CONTENIDOS

RESUMEN	1
---------------	---

CAPÍTULO I

INTRODUCCIÓN

1.1 Antecedentes.....	3
1.2 Situación Actual	4
1.3 Justificación	6
1.4 Objetivos.....	7
1.4.1 General	7
1.4.2 Específicos.....	7
1.5 Alcance	8

CAPÍTULO II

MARCO TEÓRICO

2.1 Entorno Organizacional	9
2.1.1 Descripción del Hosting	11
2.2 La Educación a Distancia	12
2.3 E-learning	13
2.3.1 Beneficios del e-learning	13
2.4 Software de Código Abierto (Open Source) y Software Libre	14
2.4.1 Software de Código Abierto (Open Source)	16
2.4.2 Software Libre	16
2.4.3 Software Libre vs Software Propietario	17
2.4.4 Linux.....	18
2.4.5 JSP.....	19
2.4.5.1 Ventajas de JSP	20
2.4.6 MySQL	21
2.4.7 Apache.....	22

2.4.8 TOMCAT	23
2.5 <i>Seguridad en la Web</i>	24
2.6 <i>Acceso personalizado</i>	27
2.7 <i>Entorno Pedagógico</i>	27
2.7.1 Definición de Pedagogía	28
2.7.2 La Tecnología de la Información y Comunicación (TIC) en los momentos de aprendizaje	28

CAPÍTULO III

ANÁLISIS

3.1 <i>METODOLOGÍA</i>	30
3.1.1. Justificación de la Metodología	31
3.2 <i>ANÁLISIS DE REQUERIMIENTOS</i>	33
3.2.1 Encuestas.....	33
3.2.1.1 Análisis de la Encuesta	34
3.2.2 Especificación de Requerimientos del Software.....	41

CAPÍTULO IV

DISEÑO

4.1 <i>Diagrama De Casos De Uso</i>	42
4.2 <i>Diagramas De Secuencia</i>	43
4.3 <i>Diagramas De Actividad</i>	45
4.4 <i>Diagrama De Estado</i>	46
4.5 <i>Diagrama De Clases</i>	47
4.6 <i>Estándares De Diseño</i>	48
4.6.1 Estándar de Tablas	48
4.6.2 Estándar para Campos	48
4.6.3 Estándar de Clases	49
4.6.4 Estándar de Ventanas.....	49
4.6.5 Estándar de Archivos	51

CAPÍTULO V

IMPLEMENTACIÓN

5.1 Configuración de la Plataforma de Desarrollo.....	52
5.2 Construcción de la Aplicación.....	52
5.2.1 Arquitectura de la aplicación.....	52
5.2.2 Esquema de JDBC.....	53
5.2.3 Esquema de Componentes utilizados en la aplicación.....	53
5.3 Pruebas.....	56
5.3.1 Tipos de Pruebas.....	57
5.3.2 Métodos de Pruebas.....	58
5.3.3 Plan de Pruebas.....	59
5.3.4 Ejecución de Pruebas.....	60
5.4 Configuración de SSL con soporte para Jakarta Tomcat.....	69

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones.....	72
6.2 Recomendaciones.....	73

BIBLIOGRAFÍA.....	75
-------------------	----

ÍNDICE DE TABLAS

Tabla 2.1: Software Libre vs Software Propietario.....	18
Tabla 2.2: Comparación JSP vs PHP vs ASP vs ASP.NET.....	21
Tabla 2.3: Comparativa MySQL vs SQL Server vs Oracle.....	22
Tabla 2.4: Comparación Apache Web Server Vs IIS.....	23
Tabla 2.5: Comparación de Servidores de Aplicación.....	24
Tabla 2.6: Comparación entre SSL, TLS, SHTTP, SSH.....	27
Tabla 3.7 : Resultados de la Encuesta para la Presentación del Sitio Web.....	34

ÍNDICE DE FIGURAS

Figura 2.1 : Cableado del INDESIC - Planta Alta	10
Figura 2.2: Cableado del INDESIC - Planta Baja	11
Figura 3.3: Modelo en Espiral Típico	32
Figura 3.4 : Gráfico tipo Barra de la Presentación del Sitio Web.....	34
Figura 3.5: Gráfico en Porcentajes de la Presentación del Sitio Web	35
Figura 3.6: Gráfico de los servicios para los Becarios.....	36
Figura 3.7: Gráfico de Servicios Adicionales para la aplicación	37
Figura 3.8: Gráfico tipo cono de las seguridades del sitio	38
Figura 3.9: Gráfico tipo anillo de las seguridades para sitio	38
Figura 3.10: Gráfico de los Cursos Requeridos.....	39
Figura 3.11: Gráfico de la Razón para Tomar el Curso	40
Figura 3.12: Gráfico de la Actividad Beneficiada con el Curso	41
Figura 4.13: Un caso de uso para el Administrador	43
Figura 4.14: Diagrama de Secuencia - Inscripción.....	44
Figura 4.15: Diagrama de Actividad - Consultar Curso	45
Figura 4.16: Diagrama de Estado - Curso.....	46
Figura 4.17: Diagrama de Clase - Becario	47
Figura 4.18: Ejemplo 1 de estándar de tablas	48
Figura 4.19: Ejemplo 2 de estándar de tablas	48
Figura 4.20: Estándar de Ventanas.....	50
Figura 5.21: Esquema de Conexión a MySql	53
Figura 5.22: Esquema de Aplicación usando Servlet y JSP	54
Figura 5.23 : Estructura del Directorio de la Aplicación	56
Figura 5.24: Método de Caja Blanca	59
Figura 5.25: Método de Caja Negra	59
Figura 5.26: Prueba para Crear Usuarios.....	61
Figura 5.27: Prueba de inscripción.....	66
Figura 5.28: Configuración SSL para Tomcat	71

ANEXOS

ANEXO A: ENCUESTAS.....	77
ANEXO B: ESPECIFICACIÓN DE REQUERIMIENTOS DEL PROYECTO.....	81
ANEXO C: DIAGRAMAS DE DISEÑO	107
ANEXO D: MODELO FÍSICO DE DATOS.....	154
ANEXO E: SCRIPT DE LAS TABLAS.....	156
ANEXO F: CONFIGURACIÓN DE LA PLATAFORMA DE DESARROLLO	164
ANEXO G: PLAN DE PRUEBAS	179

RESUMEN

El presente proyecto se ha desarrollado con el propósito de facilitar al INDESIC en el trabajo al que se dedica, que es la de capacitar a jóvenes indígenas y presentarles una nueva alternativa de capacitar mediante la tecnología y medios disponibles en la actualidad.

Este trabajo se ha desarrollado usando software de código abierto (open source) y software libre, como son: sistema operativo Linux, base de datos MySQL, tecnología de desarrollo Java (JSP, Servlet), Apache Web Server y servidor de aplicaciones Tomcat.

El ciclo de vida usado para el desarrollo de la aplicación es el Espiral, que se basa en dar vueltas en círculo varias veces hasta obtener un producto terminado. Además, para la obtención de los requerimientos del usuario se ha procedido según el estándar IEEE 830, con la que se ha realizado la Especificación de Requerimientos de Software.

Como parte importante de la ingeniería del software se ha realizado diagramas de diseño del software basados en la metodología orientada a objetos. También se ha realizado el diagrama de contexto y los DFD con el fin de comprender los procesos de la aplicación.

En la implementación se ha utilizado los Servlets, Beans y JSP, todos estos componentes se basan en la tecnología Java, que es una de las

herramientas para realizar aplicaciones para Internet. Los JSP se usa para dar el formato a las ventanas de los usuario, los Servlets se han usado para implementar el acceso a datos y los Beans representan a las clases.

Finalmente, como parte importante en la ingeniería del software, se han realizado las pruebas a la aplicación, esto permite detectar fallas que pudieran no haberse controlado en el desarrollo de la aplicación. Estas pruebas se han realizado siguiendo el estándar IEEE Std. 829.

CAPÍTULO I

INTRODUCCIÓN

1.1 Antecedentes

El Instituto para el Desarrollo Social y de las Investigaciones Científicas – INDESIC, al cual se ha tomado como caso práctico de estudio, es una institución sin fines de lucro que trabaja capacitando a indígenas de nuestro país, con cursos y seminarios dirigidos a las comunidades indígenas, así como también a nivel académico superior.

Entre las necesidades que tiene el INDESIC, está la falta de una buena comunicación con los beneficiarios de la ayuda de la Institución, los que generalmente se encuentran en casi todas las provincias del país, siendo difícil lograr esta comunicación con ellos.

Actualmente el INDESIC para planificar un curso o seminario, debe primeramente enviar un comunicado indicando los detalles del evento, las personas interesadas contestan el comunicado confirmando o no su asistencia. Este comunicado se suele hacer por carta o teléfono lo que causa una pérdida de tiempo y dinero. Además, en caso de que por algún motivo se cancelara el evento, hay que volver a enviar un comunicado indicando el motivo de la cancelación.

Para la ejecución del curso o seminario, las personas deben viajar desde las provincias hasta la ciudad de Quito para poder participar, lo que es molesto y complicado. Otra de las posibilidades es que el instructor tenga que viajar desde Quito hasta las provincias para poder capacitar.

Ante estos problemas, es necesario que el INDESIC cuente con un sistema que gestione y tramite los cursos y seminarios vía internet.

1.2 Situación Actual

Una de las necesidades actuales de nuestra sociedad en general es, combinar las actividades educativas con las actividades laborales, ya que debido a la crisis económica que enfrentamos y a la falta de tiempo, muchas personas deben trabajar y no pueden movilizarse hasta la institución educativa para educarse. Por ello es necesario que las instituciones traten en lo posible, de facilitar formas diferentes a la tradicional para que los estudiantes puedan capacitarse.

Una de esas formas es mediante cursos on line, que es una manera de estudiar por Internet que permite al alumno aprender sin tener que movilizarse hasta la institución educativa.

Tomando al INDESIC como caso de estudio, actualmente se presentan los siguientes inconvenientes que se quiere mejorar:

- Comunicar el/los seminarios a los que están invitados: Tema, lugar, fecha, etc. (que pueden cambiar).
- Movilizar a los participantes desde las diferentes provincias.
- Que los instructores tengan que viajar hasta el lugar del evento, que puede ser en cualquier provincia.
- Los interesados en los seminarios, deben confirmar la asistencia.
- Entregar publicaciones nuevas (revista, libros, material de capacitación).
- Actualizar los datos personales de los contactos.

Ante esta situación, el personal que labora en esta la Institución, tiene que hacer lo imposible por contactarse con las personas relacionadas y además se gasta muchos recursos humanos y materiales para lograr cumplir con su objetivo.

El INDESIC actualmente no dispone de un medio que permita simplificar el trabajo y se pueda aprovechar los avances tecnológicos para hacer más productivo sus actividades.

Los cursos que imparte el INDESIC son de diferentes temáticas, por ejemplo:

- Política.
- Economía.
- Sociología.
- Computación e Informática (Word, Excel, Power Point, Access, Internet).

Los cursos que son más difíciles de ejecutar son los de computación, debido a que no siempre se puede conseguir un lugar donde se pueda dar el

curso. La forma en que se imparten los cursos de computación es por módulo, es decir, es necesario culminar un tema para avanzar con otro. Por esta razón sería de gran ayuda contar con un sistema que permita tomar el curso desde cualquier lugar.

1.3 Justificación

De la problemática que tienen las instituciones para poder educar, se puede desarrollar una aplicación web que permita gestionar los cursos y seminarios de capacitación en línea, adaptable para cualquier institución que requiera solventar esta necesidad y también para el INDESIC.

Con estos antecedentes que tiene el INDESIC, y conociendo que tiene un problema real, queda claro que se debe realizar una aplicación para solucionar sus necesidades.

Entonces, se propone diseñar una aplicación para la web, que realice lo siguiente:

- Publicación de los cursos o seminarios.
- Acceder al sitio con perfil de usuario.
- Inscribirse en uno o varios cursos.
- Cancelar la inscripción.
- Descargar materiales de capacitación.
- Acceder al curso en línea.
- Rendir pruebas de evaluación.

Toda esta información se guardará en una base de datos para un mejor control, obtener informes y consultas requeridas.

Otra de las problemáticas que enfrenta nuestra sociedad es la falta de recursos económicos, razón por la cual, nuestro esfuerzo debe estar orientada a sacar el mayor provecho del software open-source (código-abierto) y del software libre, como es el caso de Linux, por lo tanto se ha escogido este sistema operativo como plataforma para este proyecto.

Además, las herramientas open-source, nos permite adecuar el software a nuestras necesidades sin tener que depender completamente del propietario del sistema, como es el caso de Windows.

1.4 Objetivos

1.4.1 General

- Analizar, Diseñar e Implementar una aplicación web para gestionar y planificar cursos de capacitación en línea utilizando herramientas open-source.

1.4.2 Específicos

- Realizar el levantamiento de la información.
- Involucrar al usuario en la obtención de los requerimientos.
- Investigar los cursos realmente necesarios para su publicación en el sitio web.
- Obtener la especificación de requerimientos de software.
- Realizar el diagrama de casos de uso y los modelos necesarios para el desarrollo de la aplicación.
- Desarrollar la aplicación según la especificación obtenida.

- Investigar y configurar seguridades que permita a los usuarios navegar con confianza por la aplicación.

1.5 Alcance

En el proyecto de tesis está previsto trabajar en las siguientes etapas:

- Análisis de requerimientos.
- Diseño.
- Desarrollo de la aplicación.
- Implementación.

Para el desarrollo de esta aplicación se pretende usar las siguientes herramientas:

- Sistema Operativo Linux RedHat 9.0.
- Lenguaje de Programación JAVA (J2SDK 1.4.0).
- Base de Datos MySQL 4.1.
- Servidor Web Apache 1.3.34.
- Servidor de Aplicaciones TOMCAT 4.1.
- Diseño para los cursos: Macromedia.

Se realizarán los siguientes módulos:

- Mantenimiento de Usuarios.
- Mantenimiento de Becarios.
- Inscripciones.
- Mantenimiento de Cursos.
- Mantenimiento de Manuales.
- Evaluación en línea.
- Prototipo de dos cursos.

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se desarrolla la base teórica que sustente el trabajo a realizar, partiendo desde los aspectos institucionales que permiten descubrir la situación en que se encuentra, para poder centrarse en la problemática. Así también se hace referencia a los aspectos técnicos y herramientas que se utiliza para solucionar las necesidades de la institución, para quien se esta desarrollando la aplicación.

2.1 Entorno Organizacional

El Instituto Para el Desarrollo Social y de las Investigaciones Científicas – INDESIC, cuenta actualmente con una infraestructura tecnológica adecuada que permite desarrollar aplicaciones para la web, y así poder explotar al máximo los recursos disponibles.

Para acceder al internet se lo hace mediante un enlace de banda ancha de 128K contratado a AccessRam, y mediante un servidor proxy se distribuye internet para el resto de la institución. Para lograr esto se ha implementado una red en toda la organización.

El servicio de internet que más se usa en el INDESIC es el correo electrónico, medio por el cual se hace las invitaciones, convocatorias y todo lo necesario para el desarrollo de algún curso.

No existe ninguna aplicación que logre mejorar las actividades antes indicadas y sería de mucha utilidad para los becarios de esta institución, que se brindara servicios de cursos y capacitación por internet, de tal forma que los interesados puedan acceder al curso en línea desde cualquier parte del país y/o el mundo inclusive.

Cabe mencionar también que la institución ya dispone de un dominio de internet que es, www.indesic.org, por lo que ya no hace falta gestionar la adquisición de uno.

En la figura 2.1 se muestra el cableado estructurado de la planta Alta del INDESIC, en el que se puede observar el lugar donde están ubicados las estaciones de trabajo y las líneas de colores indican la trayectoria del cableado.

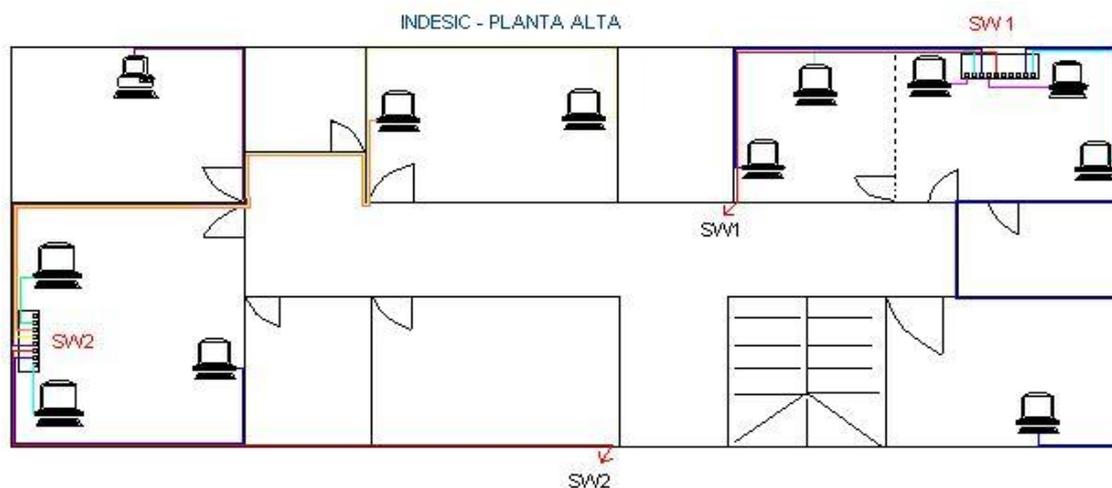


Figura 2.1 : Cableado del INDESIC - Planta Alta

En la figura 2.2 se muestra el cableado estructurado de la planta Baja del INDESIC, de igual manera que en la Planta Alta, se ilustra la ubicación de las estaciones de trabajo y la trayectoria del cableado.

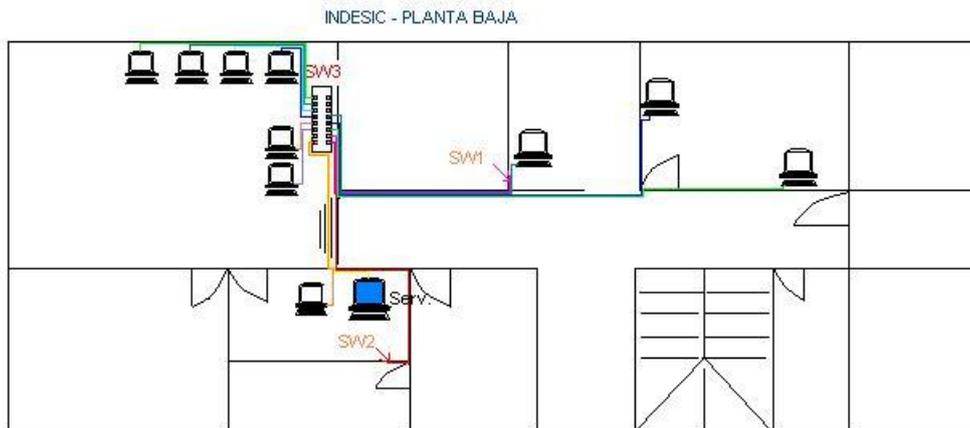


Figura 2.2: Cableado del INDESIC - Planta Baja

2.1.1 Descripción del Hosting

Por el momento, el servicio de hosting para la página que dispone el INDESIC, está siendo administrado por la empresa TU3W.

Para el alojamiento de la aplicación, se ha decidido contratar este servicio a TU3W por las siguientes razones:

- El INDESIC no dispone del personal calificado para administrar la plataforma necesaria para el funcionamiento de la aplicación.
- El INDESIC no dispone de suficiente presupuesto para contratar una persona con experiencia en manejo de las herramientas requeridas.
- El sitio web actual ya esta siendo administrado por TU3W y al contratar a ellos se reducirá los trámites y costos de redireccionamiento del DNS.
- Se reduce los costos de adquisición e instalación del servidor para la aplicación
- La empresa indicada dispone de las herramientas necesarias para el funcionamiento de la aplicación

Las razones indicadas, hacen que se haya optado por contratar los servicios a TU3W, quien se encargará de administrar el servicio.

2.2 La Educación a Distancia

Según una publicación hecha en el Diario El Heraldó, por Gustavo Montesinos, manifiesta lo siguiente:

“La educación o enseñanza a distancia es un sistema de educación, basado en la acción sistemática y conjunto de recursos didácticos y el apoyo de una organización o tutoría, que, separado físicamente de los estudiantes, participan en estos un aprendizaje independiente”¹

Por lo tanto, la educación a distancia es una manera de hacer actividad educativa, siendo ésta una forma flexible de aprendizaje, en la que el estudiante tiene la oportunidad de determinar sus metas y se hace responsable de aprender ya que es su objetivo planteado.

La educación a distancia no es algo nuevo, antiguamente ya se practicaba esto, por ejemplo, los primeros cristianos realizaban su apostolado mediante cartas y escritos, que es un ejemplo claro de la educación a distancia.

Hoy en día, se sigue dando esta modalidad de educación, claro que se puede adoptar medios más efectivos y convenientes para lograr un buen aprendizaje, tal es el caso de la educación virtual (on line) que se lo realiza mediante las redes de comunicación y del Internet. A esta forma de educación se la llama también e-learning, que se aclara un poco más en el siguiente numeral.

¹ Montesinos Gustavo, Educar a distancia, El Heraldó, Jueves 5 de mayo del 2005, Ambato

2.3 E-learning

E-learning es un nombre que se le ha dado a los sistemas de educación en línea. Es una forma de que el estudiante pueda acceder al curso mediante el Internet, a la hora que le sea posible, desde el lugar en el que se encuentre y en una forma personalizada.

Con esta forma de educación, el estudiante ya no tiene que acercarse hasta la institución educativa para dar los exámenes, para obtener ayuda o para entregar los trabajos como sucede con la educación tradicional. El estudiante encuentra los trabajos que tiene que realizar directamente en el sitio del curso, las preguntas los puede enviar por e-mail y los exámenes los puede realizar en línea cuando haya terminado el curso.

2.3.1 Beneficios del e-learning

José Antonio Ramírez, columnista de tecnología para el sitio web Entrepreneur, indica los siguientes beneficios del e-learning:

“Reducción de Costos: las empresas pueden reducir y hasta eliminar gastos de viajes para entrenamiento, además de reducir el tiempo que los empleados pasan lejos de su oficina.

Los participantes por su parte también disminuirían gastos de viajes, alimentación, etc.

Acceso: los usuarios pueden acceder al contenido desde cualquier conexión a Internet. Ello significa que el contenido del e-learning le puede llegar a una audiencia casi ilimitada.

Aprendizaje en colaboración: el e-learning permite compartir el conocimiento y el apoyo.

Objetividad: las evaluaciones, pruebas, revisiones y certificaciones están automatizadas para que el proceso sea totalmente transparente

Interactividad: es menos intimidatorio que un instructor típico o auditorio, por lo que aumenta la atención e interés de cada alumno “²

Como se ha podido notar, los beneficios del e-learning son concretos y considerando las dificultades que los estudiantes y/o profesores puedan tener para estar físicamente en una institución educativa, el e-learning ofrece una nueva forma de estudiar y autoprepararse.

Por lo mencionado, esta es una buena alternativa que el INDESIC pueda optar para poder llegar con los cursos de capacitación a todos los becarios que se encuentran en distintas ciudades del país.

2.4 Software de Código Abierto (Open Source) y Software Libre

Es necesario aclarar estos dos términos, que a menudo lleva a la confusión y se ha armado mucha polémica en la definición de los términos.

Una definición hecha por Nathan Meyers, escritor del libro Programación Java en Linux, nos dice lo siguiente:

“Cuando hablamos de software de uso libre, nos estamos refiriendo a la libertad, no al precio. Las licencias pública generales están pensadas para asegurarle la libertad de distribuir copias de software de uso libre (y cobrar por este servicio si lo desea), de recibir código fuente o de conseguirlo si lo

² Ramírez José Antonio, E-learning, <http://www.soyentrepreneur.com/pagina.hts?N=13771&Ad=S>

desea, de cambiar el software o utilizar fragmentos de él en programas nuevos de uso libre”³

La página oficial de GNU, nos dice lo siguiente del software de código abierto:

”El término software de `fuente abierta” es usado por algunas personas para dar a entender más o menos lo mismo que software libre”⁴

Según las definiciones encontradas, software de código abierto y software libre tratan de llegar a la misma definición.

Son cuatro las libertades que se tiene al usar el software libre, que a continuación se detalla:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar como funciona el programa, y adaptarlo a las necesidades del usuario. El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que se puede ayudar a otros.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Un ejemplo de software libre es el sistema operativo Linux, que actualmente es uno de los más populares y se usa principalmente en servidores web por su estabilidad y desempeño.

³ Meyers Nathan, Programación Java en Linux, Edición Especial, Prentice Hall, Madrid, 2000, Pág.276.

⁴ Categorías de Software Libre y No Libre,
<http://www.gnu.org/philosophy/categories.es.html#OpenSource>

De acuerdo a las definiciones encontradas, se describe a continuación las variantes que puede haber entre software de código abierto y software libre.

2.4.1 Software de Código Abierto (Open Source)

Este tipo de software se refiere a que se puede obtener el código fuente, pero eso no significa que sea gratis. En algunos casos su adquisición tendrá un costo mientras que en otros casos se podrá obtener gratis.

Un ejemplo de código abierto es Linux, sin embargo no todas las versiones de Linux se los puede obtener sin ningún costo, existe versiones comerciales que hay que pagar para poderlas usar, por ejemplo Linux RedHat Enterprise.

2.4.2 Software Libre

Se refiere al software que se puede adquirir sin ningún costo, sin embargo este tipo de software, no necesariamente tendrá el código fuente del programa.

Un ejemplo de software libre es, Winzip, que es una aplicación que se lo puede obtener sin ningún costo en Internet, sin embargo, este tipo de software no dispone de código fuente del programa, por lo tanto no es open source.

De acuerdo a las definiciones encontradas, para el presente proyecto se usará tanto software de código abierto (open source), como también software libre que no tiene costo.

2.4.3 Software Libre vs Software Propietario

La gran mayoría del software open-source que podemos encontrar son gratis, además de que se puede acceder al código fuente y se puede realizar los ajustes necesarios para un uso personalizado. Esta es una de las ventajas que tienen estos productos, que al contrario del software propietario, se debe pagar una licencia para poder usar el producto, esto por cada máquina en la que se instale y además no se puede acceder al código fuente ya que no ofrece la posibilidad de hacer cambios o personalizaciones.

Sin embargo, una de las ventajas de los productos pagados es que, muchos de los componentes ya vienen desarrollados, con los que se puede crear nuevos productos rápidamente.

Con respecto a las herramientas open-source, no se han desarrollado todos los componentes que disponemos en el software propietario, pero en cambio la gran mayoría es gratis y para muchas de las pequeñas empresas es conveniente usar estas herramientas ya que no tendrían que preocuparse por el costo de las licencias y una vez configurados y realizado la aplicación necesaria, se aprovecharía estos recursos para brindar un buen servicio.

En la Tabla 2.1 se puede ver un resumen de las características más importantes del software libre y del software propietario. Es claro que el software libre tiene ventajas sobre el software propietario, por lo menos en lo que a servidores se refiere.

Para este trabajo se usará las siguientes herramientas, tanto libres como de código abierto: Linux, Java, MySQL, Apache Web Server y Tomcat. A continuación se hace una breve descripción de cada uno de estos productos y el por qué se usará en este trabajo.

Tabla 2.1: Software Libre Vs Software Propietario

Propiedades	Software Libre (LINUX)	Software Propietario (WINDOWS)
Uso de programa con cualquier propósito.	Si	No
Posibilidad de estudiar el funcionamiento del programa.	SI	NO
Distribuir copias libremente.	SI	NO
Conocimiento general del software.	No	Si
Libertad de mejorar el programa.	Si	No
Facilidad de encontrar soporte técnico.	No	Si
Estabilidad.	Si	No
Posibilidad de usar un PC como servidor.	Si	No

2.4.4 Linux

LINUX hace su primera aparición en el año 1991, como resultado de las prácticas de un joven estudiante finlandés de nombre Linus Torvalds, quien ante la imposibilidad de adquirir un sistema *UNIX*⁵ (sistema operativo comercial) para realizar sus prácticas, implementó el ahora muy conocido sistema operativo LINUX, basándose en MINIX (Desarrollado por Andrew Tanenbaum) que a su vez se basaba en UNIX.

Por lo tanto, LINUX es un sistema operativo que está basado en UNIX, entonces, es un sistema operativo muy robusto y estable. Además de estas

⁵ González José, Red Hat Linux 8, Cap. 1, Pág. 34, 2003

características, es también multiusuario, lo que significa que se lo puede usar como servidor y funciona perfectamente en PC, es decir, no hace falta un hardware especial para que funcione, es open source y es libre.

Este sistema operativo se usará en el desarrollo del presente proyecto de tesis, porque sirve perfectamente como servidor Web e incluye algunas de las herramientas necesarias para el desarrollo de la aplicación.

2.4.5 JSP

Según una publicación hecha por Wikipedia, un sitio de enciclopedia virtual libre, menciona lo siguiente:

“Java Server Pages (JSP) es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java”⁶

Ampliando esta definición, se puede decir que JSP es una tecnología basada en Java y que genera páginas HTML dinámicamente, luego de que haya sido compilado y ejecutado en el servidor.

El código JSP se empotra en el código HTML, en el lugar donde se requiera obtener contenido dinámico, este código se escribe dentro de las marcas de etiqueta <% y %>.

⁶ Java Server Pages, http://es.wikipedia.org/wiki/Java_Server_Pages

Por lo tanto, el servidor web, antes de mostrar la página definitiva, deberá procesar el código que se encuentre entre las etiquetas `<%` y `%>` para de esa forma obtener los datos actualizados y finalmente mostrará al usuario el resultado de esa operación, apareciendo una página HTML común.

2.4.5.1 Ventajas de JSP

La enciclopedia virtual Wikipedia, señala la siguiente ventaja:

“La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp”.⁷

Según esta definición, mediante la tecnología Java se puede generar aplicaciones Web distribuidas, ya que se puede separar por niveles a las aplicaciones y estas aplicaciones pueden estar en clases de Java con la que se integra sin inconvenientes.

En la tabla 2.2, se hace una comparación entre los lenguajes de programación existentes para el desarrollo de aplicaciones web.

De la tabla 2.2, se observa que JSP (Java) y ASP.NET son dos competidores fuertes como lenguajes de desarrollo de aplicaciones Web. Sin

⁷ Java Server Pages, http://es.wikipedia.org/wiki/Java_Server_Pages

embargo, como herramienta open source, JSP tiene ventajas respecto a sus competidores, por lo tanto se usará JSP para el desarrollo de la aplicación.

Tabla 2.2: Comparación JSP vs PHP vs ASP vs ASP.NET

DESCRIPCIÓN	JSP	PHP	ASP	ASP.NET
Fácil de Aprender.	No	Si	Si	Si
Multiplataforma.	Si	Si	No	No
Facilidad de Conexión a Base de Datos.	Si	No	No	Si
Velocidad de Acceso.	Si	Si	No	Si
Distribuido.	Si	No	No	Si
Licencia Open Source.	Si	Si	Si (S.O. no)	Si (S.O. no)
Constante Desarrollo.	Si	No	No	Si
Soporta Clases.	Si	No	No	Si

2.4.6 MySQL

MySQL es un sistema para la administración de base de datos relacionales, el cual es una poderosa herramienta multi-usuario, es decir que soporta el acceso de varios usuarios simultáneamente y además, la estructura es compatible con cualquier base de datos SQL (Lenguaje de consulta estructurada).

En entorno Linux, es una de las herramientas más usadas para base de datos, es confiable, rápido, fácil de usar y por lo tanto se lo ha considerado para el presente proyecto.

Según la tabla 2.3, MySQL es un administrador de base de datos que puede competir con SQL Server, casi en iguales condiciones, tiene pros y contras al igual que SQL Server, pero como una base de datos libre y para aplicaciones

web es la más conveniente. Por lo tanto, se usará MySQL como base de datos para la aplicación.

Tabla 2.3: Comparativa MySQL vs SQL Server vs Oracle

DESCRIPCIÓN	MySQL	SQL Server	Oracle
Es un verdadero gestor de base de datos.	Si	Si	Si
Rendimiento.	Alto	Alto	Muy alto
Capacidad.	Alto	Alto	Muy alto
Eficiente manejo de concurrencia.	Si	Si	Si
Licencia libre.	Si	No	No
Multiplataforma.	Si	No	Si
Maneja buena seguridad.	Si	Si	Si
Soporta procedimientos almacenados.	No	Si	Si

2.4.7 Apache

Apache es un robusto servidor de páginas Web, que se la usa generalmente bajo el entorno de Linux.

Una definición de Apache, hecha por María Lorena García, de la Universidad Nacional de Salta – Argentina, nos dice:

“Hoy en día es el servidor Web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Por supuesto, se distribuye prácticamente con todas las implementaciones de Linux”⁸.

⁸ García María Lorena, Concepto de Apache, <http://g.unsa.edu.ar/instekeko/>

La razón por la que se ha escogido esta herramienta es clara, no hay más explicaciones que dar, ya que es la herramienta oficial como servidor Web en entorno de Linux.

Según la tabla 2.4, Apache Web Server tiene dos ventajas sobre Internet Information Server, es multiplataforma y software libre, que es precisamente la razón por la que se ha optado por este servidor.

Tabla 2.4: Comparación Apache Web Server Vs IIS

DESCRIPCIÓN	APACHE	IIS
Multiplataforma.	Si	No
Confiable.	Si	Si
Extensible.	Si	Si
Libre.	Si	No
Facilidad de uso.	No	Si

2.4.8 TOMCAT

TOMCAT es una herramienta que sirve como servidor de aplicaciones, es decir es un contenedor en el que se almacenan las clases. De este modo se logra desarrollar aplicaciones distribuidas, que es lo que hoy en día ha surgido con mucha fuerza.

El servidor Jakarta – Tomcat, es otro proyecto desarrollado por la organización Apache, como una implementación oficial para los servlets y para las Java Server Pages (JSP). Tomcat es un contenedor de aplicaciones, que en la página prodigyweb menciona lo siguiente:

“El contenedor es el que contiene o alberga una aplicación web y no es mas que la estructura de directorios, en donde están colocados todos los archivos necesarios para la ejecución de la aplicación web. Es lógico entonces que el primer paso en el desarrollo de cualquier aplicación web, consiste en crear la estructura de directorios en donde se colocaran los componentes”⁹.

En la siguiente tabla 2.5, se hace una comparación de 4 servidores de aplicación que soporten Java.

Tabla 2.5: Comparación de Servidores de Aplicación

DESCRIPCIÓN	TOMCAT	IPLANET	BEA WEBLOGIC	JBOSS
Multiplataforma.	Si	No	Si	Si
Confiabilidad.	Si	Si	No	Si
Compatibilidad con Java.	Si	Si	Si	Si
Libre.	Si	Si	Si	No
Facilidad de uso.	Si	No	Si	No
Compatibilidad con Apache Web Server.	Si	Si	No	Si

Según la tabla 2.5, se observa que Tomcat es un servidor de aplicaciones popular y soporta aplicaciones con Java, por lo que se ha escogido para este proyecto.

2.5 Seguridad en la Web

El Internet es considerado como la red de redes, por este mismo hecho, existe una gran cantidad de riesgos de ser accedido por hackers que quieran dañar la información, por lo que es necesario tomar medidas al respecto.

Según la página tejedoresdelweb, la forma de brindar seguridad en Internet es:

⁹ Apache TOMCAT, http://www.prodigyweb.net.mx/pupis/Cursos/jsp/ch02/pg01_text.htm

“La seguridad de este tipo se basa en el hecho de poder encriptar los mensajes que se envían por la red entre un servidor y un cliente y que solo ellos puedan descifrar los contenidos a partir de una clave común conocida solo por los dos”¹⁰.

En lo que se refiere a servicios web, existen varios protocolos que permiten realizar transacciones seguras, las más conocidas se describen a continuación.

SSL (Secure Socket Layer).- Fue desarrollado por Netscape Communications Corporation.

“SSL es una capa por debajo de HTTP y tal como lo indica su nombre esta a nivel de socket por lo que permite ser usado no tan solo para proteger documentos de hipertexto sino también servicios como FTP, SMTP, TELNET entre otros”¹¹.

TLS (Transport Layer Secure).- Se basa en SSL y son similares en el modo de operar.

Es importante señalar que ambos protocolos se ejecutan sobre una capa de transporte definida, pero no determinada. Esto indica que pueden ser utilizados para cualquier tipo de comunicaciones. La capa de transporte más usada es TCP sobre la cual pueden implementar seguridad en HTTP. Como punto de diferencia se puede mencionar que existen protocolos implementados sobre la capa de red, por ejemplo sobre IP. Tal es el caso de IPSec.

¹⁰ Seguridad en la transmisión, www.tejedoresdelweb.com

¹¹ Seguridad SSL, www.tejedoresdelweb.com

SHTTP (Secure Hyper Text Transfer Protocol).- Fue desarrollado por Enterprise Integration Technologies, este protocolo se encuentra en la capa de aplicación del TCP/IP. Para poder usar este protocolo los documentos deben tener la extensión .shttp.

SSH (Secure Shell).- Este protocolo fue diseñado para dar seguridad de acceso a computadores remotamente. Cumple casi la misma función que telnet o rlogin, solo que usando criptografía logra seguridad con los datos.

A diferencia de telnet u otro servicio similar, SSH utiliza el puerto 22 para la comunicación y la forma de efectuar su trabajo, es muy similar al efectuado por SSL.

Para su uso se requiere que por parte del servidor exista un demonio, que mantenga continuamente en el puerto 22 el servicio de comunicación segura.

Finalmente cabe mencionar que los tipos de protocolos vistos, usan cifrado para poder encriptar las claves y la información se intercambia entre el usuario y el servidor quien autoriza o niega el acceso dependiendo de la clave.

De acuerdo al resumen presentado en la tabla 2.6, el protocolo más adecuado para realizar transacciones seguras por Internet es el SSL o TLS. Se puede usar para realizar una simple comunicación segura entre el servidor y el cliente, o también para transacciones de alta seguridad como son los pagos con tarjeta de crédito.

Tabla 2.6: Comparación entre SSL, TLS, SHTTP, SSH

Descripción	SSL	TLS	SHTTP	SSH
Brinda seguridad de transmisión de datos por Internet.	Si	Si	Si	Si
Alta seguridad.	Si	Si	No	No
Usan cifrado para la transmisión.	Si	Si	Si	Si
Se debe pedir certificado de seguridad a una empresa certificadora.	Si	Si	No	No
Se puede usar para pagos con tarjetas de crédito.	Si	Si	No	No
Facilidad de uso.	No	No	SI	SI

En el presente proyecto se usará el protocolo SSL, que se configurará en el servidor de aplicaciones, de modo que brinde seguridad a los usuarios.

2.6 Acceso personalizado

En las aplicaciones de Internet, el acceso personalizado a la información es una parte importante que se debe manejar y tomar en cuenta en el diseño y desarrollo, ya que es necesario que cada usuario pueda tener opciones diferentes dependiendo del perfil que disponga.

En el presente trabajo, se tomará en cuenta las opciones que deben tener cada tipo de usuario. Los tipos de usuarios que se requiere para el INDESIC se describen en el Anexo C.1 Diagramas de Casos de Uso y en la Especificación de Requerimientos de Software en el Anexo B.

2.7 Entorno Pedagógico

El e-learning debe cumplir también este punto, ya que es un sitio de aprendizaje y debe ser comprensible, dinámico, amigable y fácil de usar por parte

del usuario. Existen varias formas de hacer que el sistema sea pedagógico, como por ejemplo que se usen animaciones, videos, etc. para motivar al usuario.

2.7.1 Definición de Pedagogía

“El hombre al nacer, lleva en sí una gran potencialidad. Su esfuerzo consistirá en actualizar estas potencias, en madurar lo que su naturaleza tiene por vocación. A los seres inferiores les basta dejarse llevar por sus instintos, y así alcanzan la plenitud de su ser. No así el hombre. El “modo” de conseguir su actualización difiere radicalmente de los seres irracionales. El Creador no le impone físicamente el ser pleno que debe ser; sólo se lo ofrece. No pudiendo alcanzar determinísticamente, será necesario buscar un método, un sistema que haga del “aspirante a hombre” un hombre perfecto en su vida personal y social. A este método es lo que llamamos PEDAGOGÍA”¹².

De este concepto podemos comprender que como seres humanos, necesitamos de un método para poder alcanzar el conocimiento, es potestad de cada persona elegir el método que le convenga.

2.7.2 La Tecnología de la Información y Comunicación (TIC) en los momentos de aprendizaje

Con las TIC, el profesor ya no puede interactuar directamente con el alumno, sin embargo, se puede desarrollar otras formas de lograr la interacción utilizando los recursos tecnológicos, como por ejemplo:

- Foros en línea.
- Chat.

¹² Perelló Julio, Historia de la Educación, 1967 Editorial Don Bosco Quito – Ecuador

- Grupos de trabajo.
- Correo electrónico.

Como se puede notar, existen formas de hacer que la educación por Internet sea interactiva, mediante las formas descritas. Usando la tecnología se puede superar las limitaciones en la educación.

Cabe indicar que, tampoco se quiere erradicar por completo la educación tradicional, siempre existe la necesidad de que los seres humanos nos relacionemos, por lo tanto será necesario en algún momento una interacción Profesor-Alumno, para que exista la retroalimentación de conocimientos y sobre todo para que se cultive el respeto entre las personas. Por tal motivo, los profesores encargados de la planificación deberán analizar que materias pueden ser tomados por Internet y que otras materias deben ser tomados presencialmente y en que horario.

CAPÍTULO III

ANÁLISIS

La etapa de Análisis es muy importante en la ingeniería del software, permitiendo crear un producto estable y confiable, en base a una forma correcta para la obtención de requerimientos.

Mediante la etapa de Análisis se puede identificar las necesidades reales que tiene una institución o persona, por esta razón en este capítulo se presenta un análisis de requerimientos para el INDESIC.

3.1 METODOLOGÍA

Como parte de la ingeniería del software, corresponde escoger una metodología de diseño del software, por esta razón en este apartado se describe la metodología escogida.

Una metodología es una estrategia de desarrollo de software que se basa en principios y reglas para la obtención de un software de calidad, una metodología esta compuesta de métodos.

Para este proyecto de tesis, se ha escogido la Metodología Orientada a Objetos OOM, esta metodología pone mucho énfasis en los objetos que interviene en la aplicación, por lo tanto se realiza varios diagramas orientados a objetos, los

más importantes son: diagrama de casos de uso, diagramas de secuencia, de actividad, de estado y diagrama de clases.

3.1.1. Justificación de la Metodología

Para este proyecto de tesis se ha escogido la metodología orientada a objetos, ya que es una de las metodologías que permite modelar el software mediante representaciones de objetos que es lo más cercano a la realidad haciendo que se produzca un análisis de requerimientos más efectiva para el desarrollo de software.

Ciclo de Vida de Desarrollo de Software.

Roger Pressman, en su publicación, “Ingeniería del Software; Un Enfoque Práctico”, manifiesta que: *“En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas de ingeniería del sistema”*¹³.

Como modelo o ciclo de vida de desarrollo de software se ha escogido el Espiral, ya que es el paradigma que mejor se ajusta a la realidad del INDESIC, debido a que en la Institución, no hay un organigrama funcional que defina claramente lo que se necesita, los cursos que se realizan se los hace sin parámetros establecidos.

¹³ Pressman Roger, Ingeniería del Software: un enfoque práctico, Cuarta Edición.

El paradigma en espiral nos sugiere trabajar en seis etapas (cuadrantes) que son: Comunicación con el cliente, planificación, Análisis de Riesgos, Ingeniería, Construcción y Adaptación, y finalmente la Evaluación del Cliente.

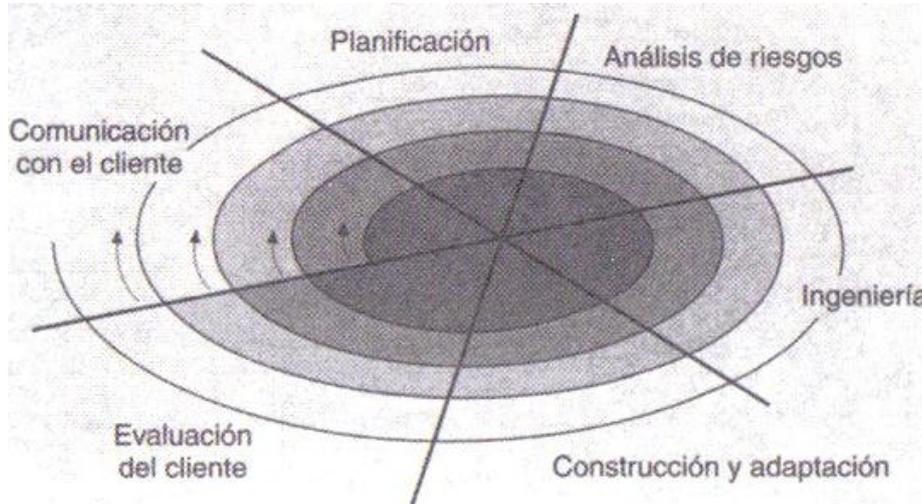


Figura 3.3: Modelo en Espiral Típico

Cada una de las áreas por las que se pasa en el modelo espiral se encarga de ciertas actividades específicas, que a continuación se detalla:

“Comunicación con el Cliente - las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.

Planificación – las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto.

Análisis de Riesgos – las tareas requeridas para evaluar riesgos técnicos y de gestión.

Ingeniería – las tareas requeridas para construir una o más representaciones de la aplicación.

Construcción y Adaptación – las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (p, ej.: documentación y práctica).

Evaluación del Cliente – las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación”¹⁴.

3.2 ANÁLISIS DE REQUERIMIENTOS

El análisis de los requerimientos se lo ha realizado en base a dos herramientas. La primera mediante encuestas realizada a los becarios, exbecarios y al personal que labora en el INDESIC. La segunda herramienta utilizada es la norma IEEE 830 de la Especificación de Requerimientos del Software, en cuyo documento se describe las características más comunes para obtener un software de calidad.

3.2.1 Encuestas

Mediante estas encuestas se ha logrado definir las necesidades más comunes que tienen los becarios y el personal que labora en el INDESIC, consiguiendo una guía para este trabajo.

Para la encuesta se ha tomado una muestra de 50 personas, de una población total de 150, 8 personas representan al personal que laboran en el INDESIC y el resto corresponde a los Becarios.

El formulario de la encuesta se podrá observar en el *ANEXO A: Formulario de Especificación de Requerimientos*.

¹⁴ Pressman Roger, Ingeniería del Software; un enfoque práctico, Cuarta Edición

3.2.1.1 Análisis de la Encuesta

Primera Pregunta: Presentación del Sitio Web

A continuación, en la Tabla 3.7, se presentan las respuestas de la primera pregunta, que hace referencia a la presentación del sitio Web en general vs la rapidez de acceso.

En esta primera pregunta, solo podía escoger una opción de las cuatro, y los resultados se muestran en la Tabla 3.7.

Tabla 3.7 : Resultados de la encuesta para la presentación del sitio web

Opciones	Respuestas
Simple	4
Sencillo	23
Sofisticado	17
Muy Sofisticado	6

La Figura 3.4, muestra gráficamente, el resultado del tipo de diseño que debe disponer el sitio con la aplicación.

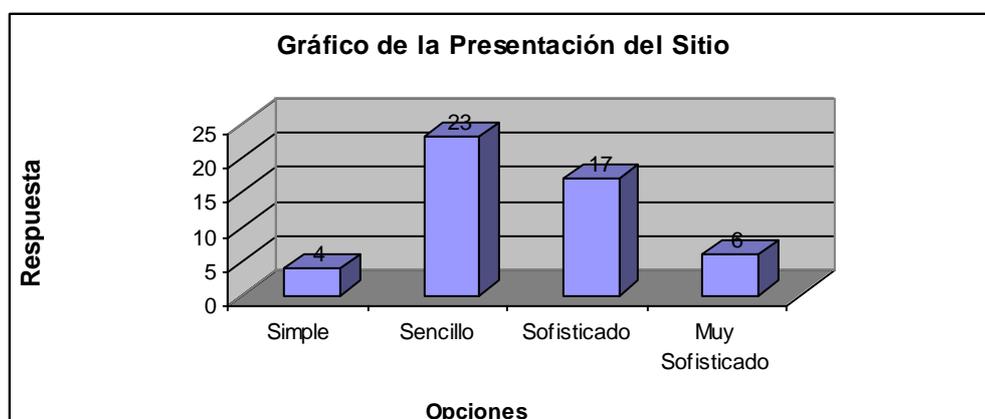


Figura 3.4 : Gráfico tipo Barra de la Presentación del Sitio Web

La Figura 3.5 muestra el mismo resultado del diseño, pero con un gráfico circular.

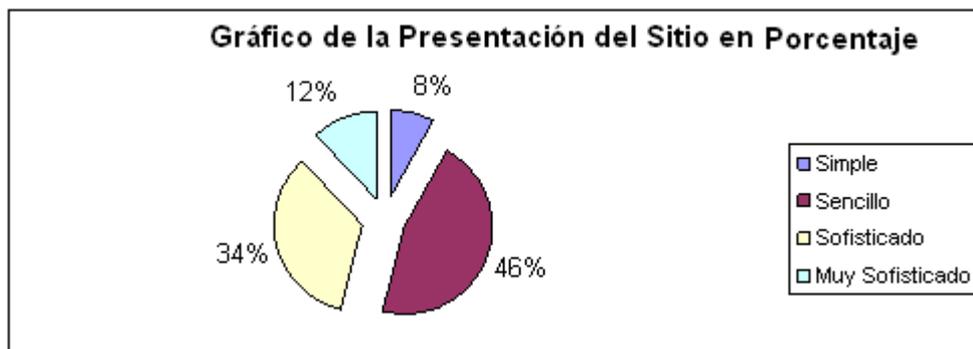


Figura 3.5: Gráfico en Porcentajes de la Presentación del Sitio Web

Según la tabla 3.7 y las figuras 3.4 y 3.5, 23 personas de las 50 desean una página sencilla, con ilustraciones que les ayude a comprender, pero que se puedan acceder rápidamente al sitio. Esto representa un 46% del total de los encuestados.

Otro buen grupo de 17 personas de las 50 (34%), en cambio se han decidido por una presentación Sofisticada, con muchas ilustraciones, pero que en realidad requiere un ancho de banda bastante amplia.

Ante estas dos principales inclinaciones, se optará por desarrollar páginas sencillas, con ilustraciones que guíen a los becarios hacia un rápido aprendizaje y sobre todo, que el acceso al sitio sea rápido.

La velocidad de acceso es un aspecto que se debe tomar muy en cuenta para que el visitante no se canse y pueda culminar el curso con éxito.

Segunda Pregunta: Servicios para los Becarios

En la pregunta dos, se muestra los servicios más comunes que se puede brindar a los becarios.

En la encuesta se pide escoger varias opciones y se obtiene el siguiente resultado que muestra la Figura 3.6.

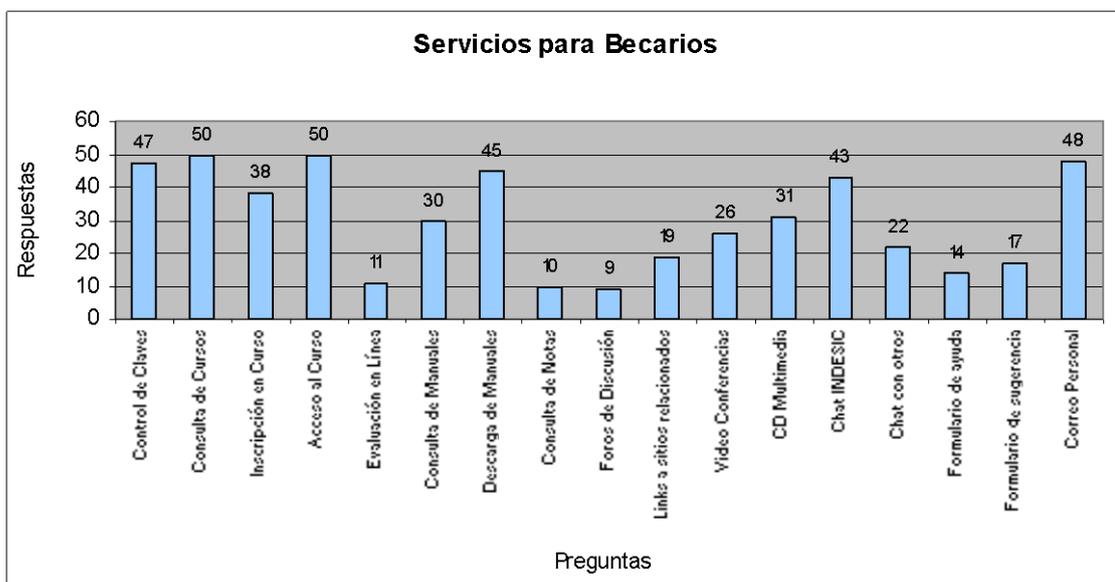


Figura 3.6: Gráfico de los servicios para los Becarios

Como se puede observar en la Figura 3.6, los servicios más requeridos son consultar cursos disponibles y la posibilidad de acceder al curso, lo cual coincide con un valor de 50 que es el total de los encuestados. Es decir, todos los encuestados prefieren disponer de esos servicios.

Tercera Pregunta: Otros Servicios

En lo que respecta a otros servicios adicionales, se tiene como principal, el mantenimiento de usuarios y el mantenimiento de cursos, con un valor de 50,

es decir de todos los encuestados. Contrario a la pregunta anterior, la opción de modificar notas y concatenar cursos, los encuestados prefieren que no se disponga de esos servicios. Por lo tanto se tomará mucho énfasis los puntos más relevantes.

La Figura 3.7, muestra el resultado sobre los servicios adicionales.

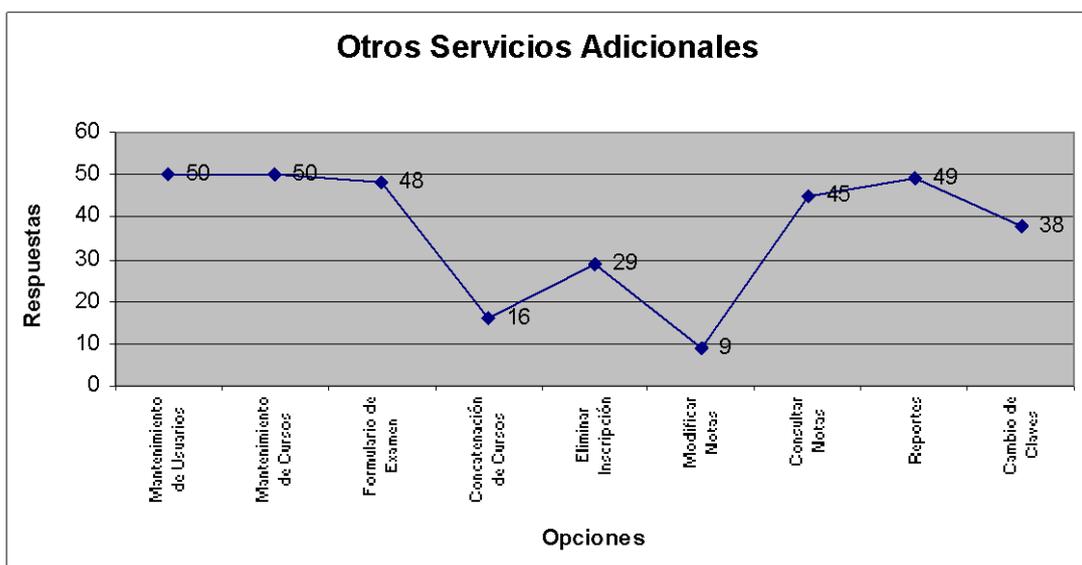


Figura 3.7: Gráfico de Servicios Adicionales para la aplicación

Cuarta Pregunta: Seguridad

Con respecto a las seguridades que debe poseer la aplicación, los encuestados quieren un sitio seguro, que sus notas estén seguras y que no los puedan modificar extraños.

Lo que menos quieren en las seguridades es que las notas sean abiertas para poder ser modificadas y que los usuarios deban ser creados por el administrador, es decir que ellos quieren que se deje el acceso libre para

todos los visitantes del sitio, sean estos becarios o no del INDESIC. En este punto, es necesario recalcar que el INDESIC como Institución requiere más bien hacer un control para que no todos puedan entrar libremente a los servicios Web del INDESIC, ya que este servicio es exclusivamente para la Institución.

La Figura 3.8, muestra que 50 encuestados prefieren un sitio seguro.

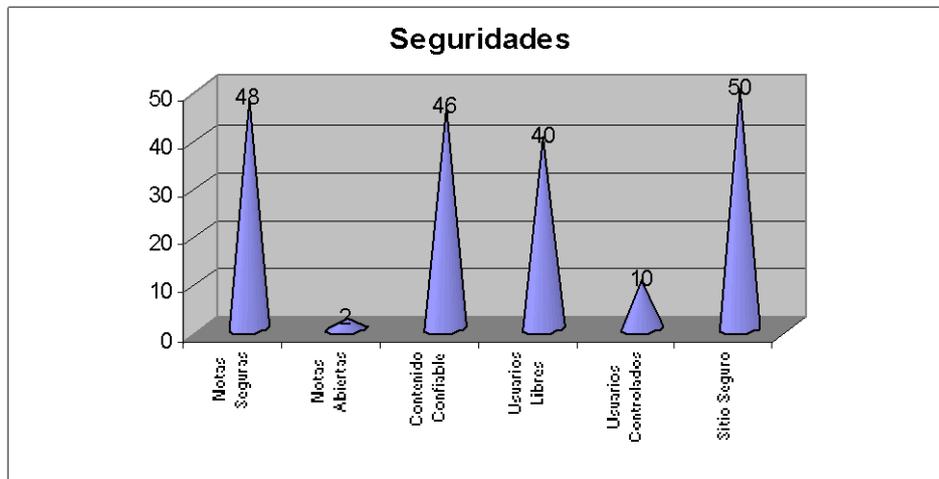


Figura 3.8: Gráfico tipo cono de las seguridades del sitio

La Figura 3.9, muestra el mismo resultado de las seguridades pero en gráfico circular, esto corresponde al 27% de los resultados totales.

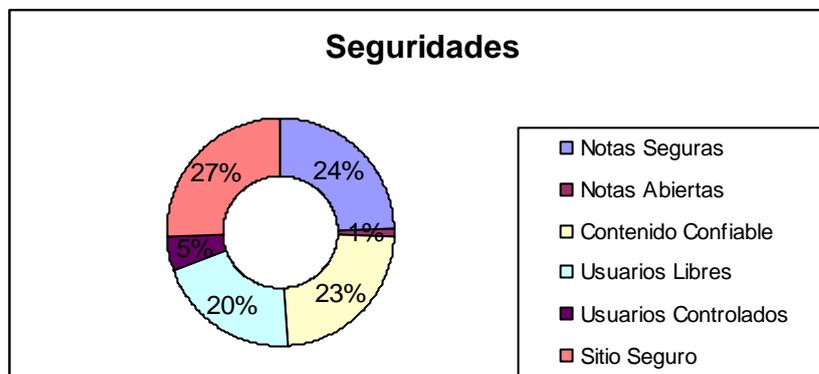


Figura 3.9: Gráfico tipo anillo de las seguridades del sitio

Curso a Publicar

En este proyecto de tesis, también se ha propuesto publicar dos cursos de capacitación, como parte de la implementación, razón por la que también se han realizado encuestas para poder determinar cuáles cursos son los más requeridos por los becarios del INDESIC.

Primera Pregunta: Cursos Requeridos

El resultado de la primera pregunta se muestra en la Figura 3.10, donde se verifica cuales cursos son los más necesarios para los becarios del INDESIC.

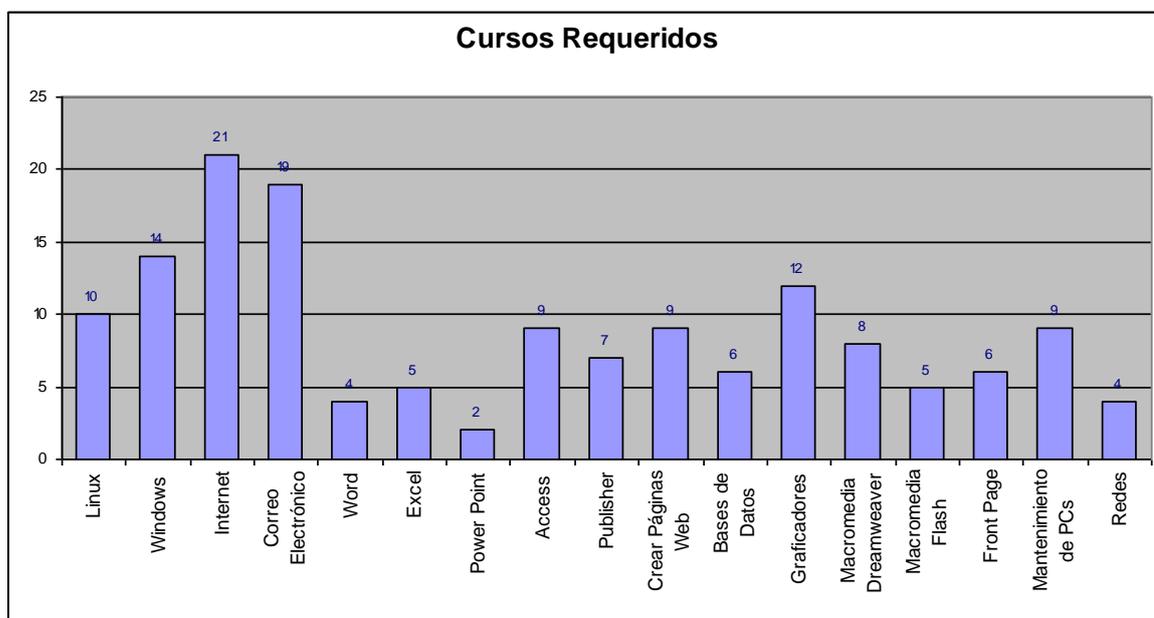


Figura 3.10: Gráfico de los Cursos Requeridos

De acuerdo con el gráfico de la Figura 3.10, los becarios del INDESIC necesitan que se les facilite la posibilidad de tomar cursos de Internet y correo electrónico. Estos cursos son requeridos por los becarios, debido a que en la actualidad, se usa bastante el Internet y el correo electrónico en el campo educativo, lo que implica que ellos precisen usar estas nuevas tecnologías.

Segunda Pregunta: Razones para Tomar el Curso

En la segunda pregunta de la encuesta sobre el curso necesario, hace referencia a la razón por la que necesita el curso seleccionado, en la Figura 3.11, se puede apreciar el resultado.

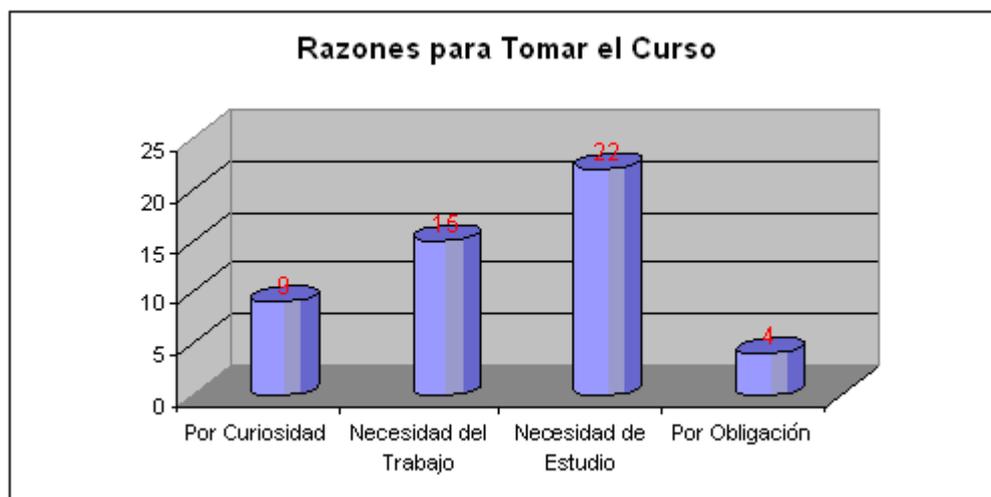


Figura 3.11: Gráfico de la razón para tomar el curso

La razón por la que requieren el curso, la mayoría opta por cuestiones de estudio, ya que los becarios del INDESIC se dedican a estudiar y otra gran parte a trabajar y estudiar, siendo el estudio la razón principal de querer aprender a manejar el Internet, que hoy en día se ha convertido en la principal herramienta de investigación.

Tercera Pregunta: Beneficios al tomar el Curso

Esta tercera pregunta es un refuerzo de la pregunta anterior, que analiza la actividad que se verá beneficiada con el aprendizaje del curso. Según la Figura 3.12, la actividad beneficiada es justamente la educativa, con lo que los becarios podrán acceder a Internet y usar la herramienta de la manera más adecuada en su aprendizaje.

En la Figura 3.12, se presenta los resultados en porcentajes, sobre las actividades que se mejorarán al tomar los cursos.

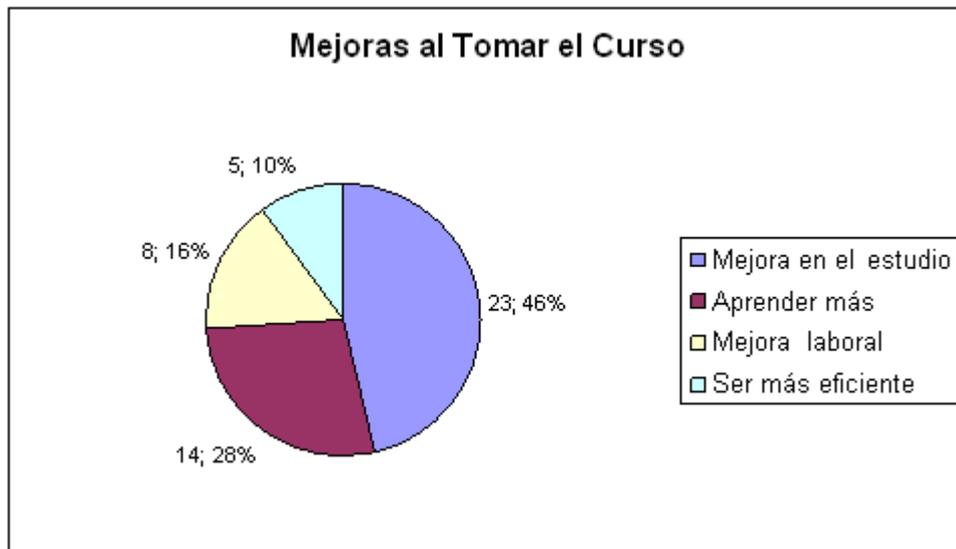


Figura 3.12: Gráfico de la Actividad Beneficiada con el Curso

3.1.2 Especificación de Requerimientos del Software

La especificación de requerimientos del software nos permitirá obtener los requerimientos más detallados y nos permitirá comprender las funcionalidades del sistema.

La especificación de requerimientos se ha realizado siguiendo la norma IEEE 830, en cuyo documento se describe las características más comunes para que el software sea eficiente y garantizado.

En el Anexo B: Especificación de Requerimientos de Software, se podrá ver la descripción de requerimientos para la aplicación.

CAPÍTULO IV

DISEÑO

La etapa de Diseño, es también una de las partes importantes de la ingeniería del software, ya que mediante el diseño se puede modelar el sistema, con lo que se obtiene el sistema mismo en una forma arquitectónica. Al igual que una casa que se la puede representar en un plano, así también el software se lo puede representar mediante diseños que permiten obtener una idea preliminar de lo que será el producto final, pudiendo en esta etapa, cambiar y depurar algunos detalles, para luego de esto pasar definitivamente a la implementación.

La aplicación web para cursos de capacitación del INDESIC, se ha planteado de forma que tengamos diagramas orientado a objetos, que es una de las tendencias actuales en el desarrollo de software, por eso a continuación se representa los diagramas más importantes en el Diseño Orientado a Objetos.

4.1 DIAGRAMA DE CASOS DE USO

Los diagramas de casos de uso permiten conocer la funcionalidad del sistema de un modo preciso.

Los casos de uso se representan mediante óvalos y las figuras de forma humana simbolizan a los actores, que interactúan con los casos de uso y la relación entre ellos se dibuja mediante flechas.

A continuación en la Figura 4.13, podrá observar un ejemplo de caso de uso que realiza el Administrador de la aplicación que se realizará.



Figura 4.13: Un caso de uso para el Administrador

Cada actor puede tener varios casos de uso como sea necesario, los actores y los casos de uso se los define al momento de obtener las especificaciones de requerimiento.

En el *Anexo C: Diagramas de Diseño (C.1: Diagrama de Casos de Uso)*, se muestra los casos de uso y los actores que intervienen.

La descripción de los casos de uso y los actores se podrá encontrar en el *Anexo C: Diagramas de Diseño - Descripción de Casos de Uso y Actores*.

4.2 DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia permiten identificar la interacción entre las clases, se puede observar la secuencia de actividades necesarias y flujos de información que se intercambian entre las clases para lograr un objetivo.

Los diagramas de secuencia se representan con rectángulos, en cuyo interior se escribe el nombre del objeto que interviene y bajo el rectángulo se lanza una línea vertical punteada, que representa un período de tiempo.

Para que exista interacción entre los objetos, es necesario que un objeto pase información a otro, esto se representa mediante flechas que se trazan horizontalmente entre las líneas verticales punteadas e indican de que objeto sale el flujo y hacia que objeto se dirige.

La figura 4.14, muestra un ejemplo del diagrama de secuencia, en el que se representa la interacción entre los objetos para realizar la inscripción en un curso.

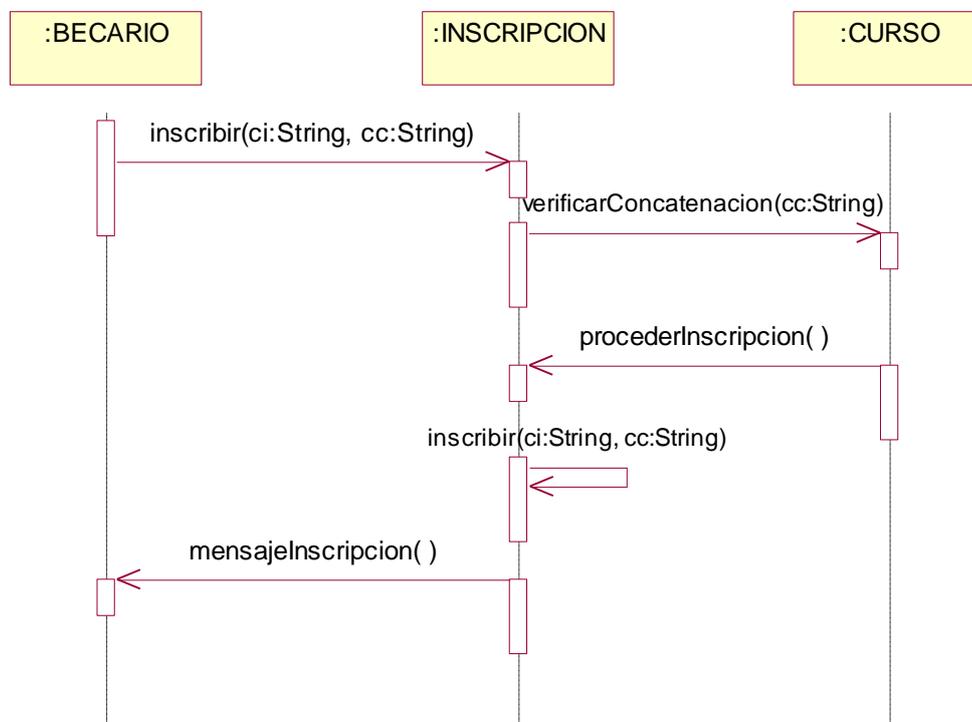


Figura 4.14: Diagrama de Secuencia - Inscripción

Debido a que los diagramas de secuencia muestran los objetos y métodos que intervienen en la aplicación, estos diagramas no presentarán las normas ortográficas en el texto que aparezca en ella.

En el *Anexo C: Diagramas de Diseño (C.2: Diagramas de Secuencia)*, podrá encontrar los demás diagramas de secuencia de la aplicación.

4.3 DIAGRAMAS DE ACTIVIDAD

Los diagramas de actividad son aquellos que permiten identificar las actividades que se deben seguir para lograr cumplir con un proceso. Los diagramas de actividad se representan mediante rectángulos con las esquinas redondas, en cuyo interior se describe la actividad que se va a realizar, cada actividad conduce hacia otras actividades, y el flujo de las actividades se representa mediante flechas.

Los diagramas de actividad deben tener siempre un inicio y un fin de las actividades. El inicio y el fin se representan mediante círculos con fondo de color negro, el final a más del círculo, tiene una línea que rodea el círculo y de esa forma se diferencia el inicio del final. La figura 4.15, muestra un ejemplo del diagrama de actividad.

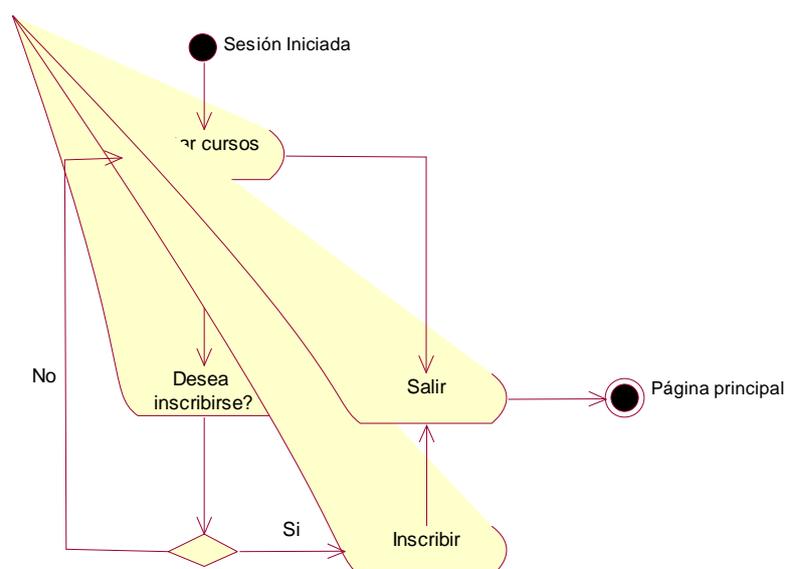


Figura 4.15 : Diagrama de Actividad - Consultar Curso

Para ver el resto de los diagramas de actividad diseñados para la aplicación, refiérase al *Anexo C: Diagramas de Diseño (C.3: Diagramas de Actividad)*.

4.4 DIAGRAMA DE ESTADO

Los diagramas de estado indican los estados por los que puede pasar un objeto. Estos diagramas de estado ayudan a visualizar los estados que deben tener los objetos, de modo que se considere para la construcción de las clases.

Los diagramas de estado se representan con rectángulos cuyos filos son redondos, y en el interior de los rectángulos se escribe el estado en el que se encuentra el objeto.

Un Objeto puede pasar de un estado a otro, y esto se lo representa mediante flechas que indican el cambio de estado. La Figura 4.16 muestra un ejemplo del diagrama de estado, para el objeto curso.

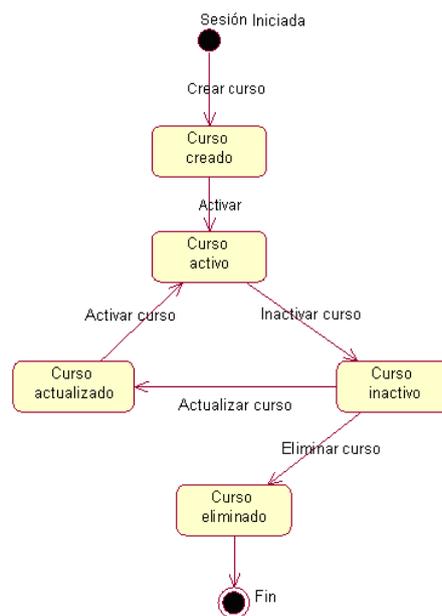


Figura 4.16 : Diagrama de Estado - Curso

En el *Anexo C: Diagramas de Diseño (C.4: Diagramas de Estado)*, se podrá encontrar los demás diagramas de estado para la aplicación a desarrollar.

4.5 DIAGRAMA DE CLASES

El diagrama de clases presenta la estructura del sistema mediante los objetos que se han identificado. Los objetos están compuestos por atributos, propiedades, operaciones y asociaciones.

Un diagrama de Clases representa principalmente la parte fundamental del análisis de datos, ya que de acuerdo a este diagrama se diseñarán las clases necesarias para crear el software.

El diagrama de clases se representa mediante un rectángulo que está compuesto por tres filas, en la fila superior se pone el nombre de la clase, en la fila del medio se describe los atributos de la clase y en la fila inferior se describe las operaciones que realizará la clase.

A continuación, en la Figura 4.17, se muestra un ejemplo de la clase becario, con sus atributos y operaciones.

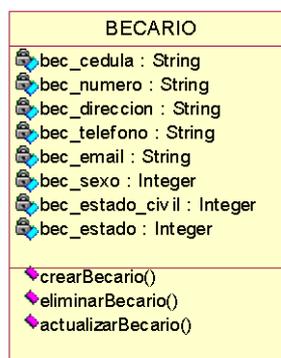


Figura 4.17: Diagrama de Clase - Becario

En el *Anexo C: Diagramas de Diseño (C.5: Diagramas de Clases)*, se muestra los diagramas completos para la aplicación a realizar.

4.6 ESTÁNDARES DE DISEÑO

4.6.1 Estándar de Tablas

Para la especificación de tablas se ha definido la siguiente estructura:

Nombre de la tabla, en mayúsculas.

Los campos de la tabla, también en mayúsculas.

Ejemplo: USUARIO

USUARIO	
USU_LOGIN	
USU_PASSWORD	
USU_NOMBRES	
USU_APELLIDOS	
USU_ESTADO	

Figura 4.18: Ejemplo 1 de estándar de tablas

El nombre de la tabla deberá estar compuesta por una palabra completa, en caso de que se componga de varias palabras, cada palabra está separada con un guión bajo. Ejemplo: PERFIL_OPCION

PERFIL_OPCION	
OPC_CODIGO	INTEGER
PER_CODIGO	INTEGER

Figura 4.19: Ejemplo 2 de estándar de tablas

4.6.2 Estándar para Campos

Se ha establecido el siguiente estándar para la descripción de los campos de las tablas.

TAB_NOMBRE_CAMPO

Donde:

TAB: representa al nombre de la tabla.

NOMBRE_CAMPO: identifica al nombre del campo para la tabla TAB. En caso de que el nombre del campo se componga de dos palabras, ésta deberá estar separada con guiones bajos.

Se especifica el nombre completo del campo, ya que a la hora de la codificación ayudará a recordar a que se refiere cada campo.

Manejo de Claves Primarias

Las claves primarias de las tablas se registrarán automáticamente, por lo tanto deberán ser de tipo Integer y Autoincrementable; excepto las tablas BECARIO Y USUARIO, en las que la clave primaria es el login, el mismo que es de tipo caracter.

4.6.3 Estándar de Clases

Para el manejo de las clases se han adoptado las características estándar del manejo de clases, que es: nombreDeLaClase(parámetros).

Este estándar consiste en iniciar el nombre de la clase en minúsculas, y cada palabra siguiente debe ir sin espacio y la primera letra de la palabra en mayúsculas.

4.6.4 Estándar de Ventanas

Las ventanas deberán ser diseñadas para ser mostradas usando cualquier explorador de Internet.

Se ha definido trabajar en base a marcos. El marco base debe contener a tres elementos: un marco izquierdo donde esta el menú de opciones, el marco superior, donde se coloca la información de la institución y un marco central, que es donde se mostrará la información y datos necesarios para interactuar con los usuarios del sistema.

En el marco izquierdo, a más de contener el menú para los diferentes usuarios, tendrá el logo del INDESIC, que está en la parte superior del marco.

A continuación, en la Figura 4.20, se muestra un ejemplo base de la ventana

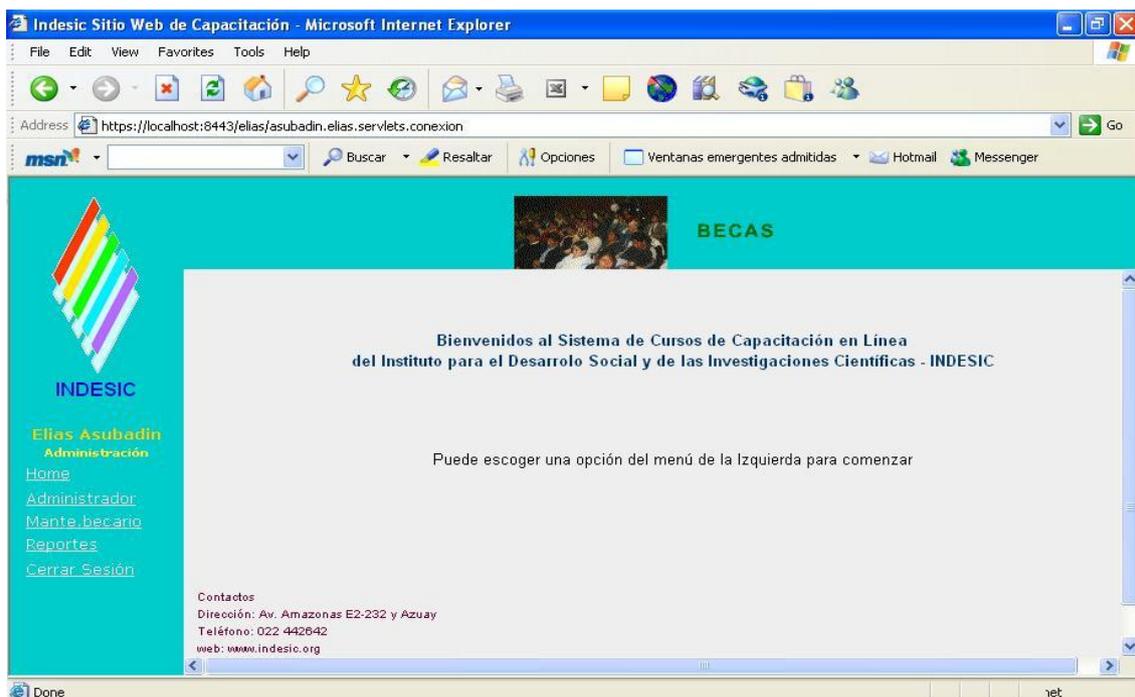


Figura 4.20: Estándar de Ventanas

El color que se muestra en esta ventana es el celeste, y es el color que identifica a la Institución, por tal motivo se usará este color como estándar. Para los elementos del formulario se usarán colores que más se ajusten al celeste especificado.

El color del marco principal será el blanco, se ha decidido este color para facilitar la lectura y presentación de los elementos que se muestran.

4.6.5 Estándar de Archivos

El nombre de los archivos, serán de la forma:

nombreArchivo.ext

Ejemplo: mantenimientoUsuario.jsp

Los archivos de los cursos publicados se almacenarán en la carpeta fichero, los archivos de imágenes necesarios para el sitio estarán ubicados en la carpeta image y en la carpeta manual se almacenarán los archivos de los manuales.

Los archivos jpg y las demás carpetas y paquetes de la aplicación, se almacenarán en una carpeta principal, que para este caso se lo ha denominado elias.

Las carpetas upfile y WEB-INF, que aparecen dentro de la carpeta elias, son paquetes que contienen clases que se utiliza en la aplicación.

CAPÍTULO V

IMPLEMENTACIÓN

En este capítulo se describe el proceso mismo de la implementación de la aplicación Web, partiendo desde la configuración de las herramientas que se ha utilizado, así también se describen las clases creadas y la base de datos.

5.1 Configuración de la Plataforma de Desarrollo

La configuración de las herramientas necesarias para el funcionamiento de la aplicación se describe detalladamente en el *Anexo F: Configuración de la Plataforma de Desarrollo*.

5.2 Construcción de la Aplicación

Para la construcción de la aplicación se ha utilizado la tecnología Java, específicamente se han usado los siguientes componentes:

5.2.1 Arquitectura de la aplicación

La aplicación Web se ha desarrollado utilizando las tecnologías que corran en el lado del servidor, es decir, que toda las políticas del negocio sea procesado en el servidor y el resultado se devuelva al cliente a través de un navegador de Internet, cualquiera que éste sea.

Por un lado está el esquema de la conexión a la base de Datos mediante JDBC, y por otro lado está el esquema de componentes que se utiliza en la construcción de la aplicación, en la que se utiliza la tecnología Java.

5.2.2 Esquema de JDBC

Para la conexión desde la aplicación web a MySQL, se usa un JDBC, que hace de mediador entre las dos partes. La Figura 5.21, muestra el esquema de lo descrito.

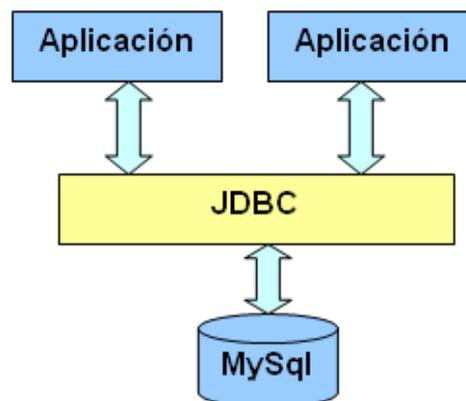


Figura 5.21: Esquema de Conexión a MySQL

Un JDBC es un componente desarrollado con Java, que facilita al desarrollador la conexión a la base de datos. Permitiendo de esta forma, que el desarrollador se dedique a su aplicación.

5.2.3 Esquema de Componentes utilizados en la aplicación

Para el desarrollo de la aplicación se ha usado Servlet, Beans y JSP. Y para el funcionamiento de la misma, está involucrado el motor de Servlet (servidor de aplicaciones), que en este caso es el Tomcat; el Servidor Web, que se encargará de entregar las páginas finales al usuario, para este caso se ha seleccionado a Apache y finalmente está el cliente, quien realizará las peticiones al servidor web.

En la Figura 5.22 se puede observar la forma en el que se encuentran organizados los componentes mencionados en el párrafo anterior.

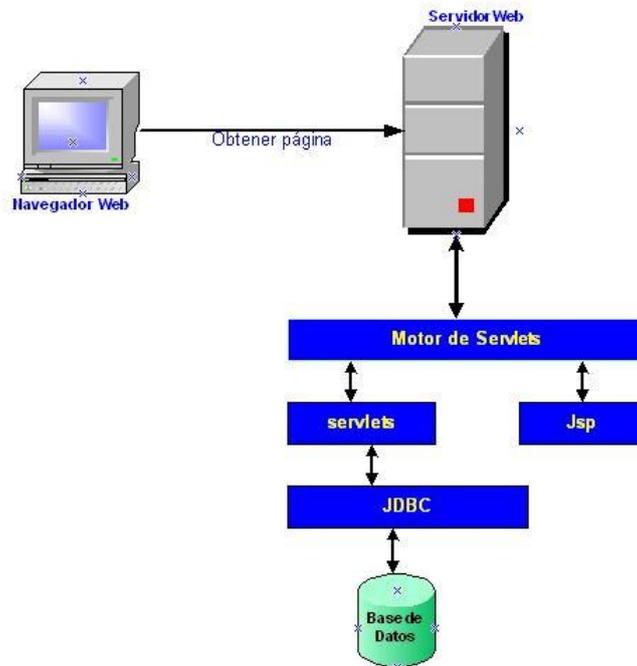


Figura 5.22: Esquema de Aplicación usando Servlet y JSP

5.2.3.1 Servlets

Según Javier García De Jalón, de la escuela de Ingeniería de la Universidad de Navarra, define a los Servlets así:

“Un programa escrito en Java que se ejecuta en el marco de un servicio de red (un servidor HTTP, por ejemplo), y que recibe y responde a las peticiones de uno o más clientes.”¹⁵

Para la aplicación, se han creado varios servlets, y mediante estos se hace la conexión a la base de datos, según la necesidad que se haya tenido.

¹⁵ García De Jalón Javier, Servlets, www.navarra.edu.es

5.2.3.2 Beans

Los Beans representan a las clases diseñadas para la aplicación, en ella se guardan los atributos y las operaciones asociadas con el ingreso y obtención de los datos al momento en que se instancia la clase.

Los beans se encuentran en el paquete Beans, dentro de la estructura del directorio creado para la aplicación, como se muestra en la figura 5.23.

5.2.3.3 JSP

Los JSP, son las páginas que contienen código java embebido en el código HTML. Los JSP se utilizan para dar formato a las pantallas que aparecerán a los usuarios. El usuario podrá observar el código HTML y no JSP, logrando con esto, una mayor seguridad.

Los JSP, se los ubica en la carpeta principal creado para la aplicación, que en este caso es elias. Desde ahí, llamarán a los servlets cuando requieran conectarse a la base de datos.

5.2.3.4 Estructura de Directorios

Para la aplicación, se ha creado una estructura de directorios, con el fin de encajar con el esquema de la tecnología Java mostrada en la Figura 5.23.

Esta estructura de directorios está compuesta por una carpeta principal, donde se guardan todos los componentes de la aplicación, esta carpeta tiene por nombre elias. Dentro de éste se encuentran otras carpetas, por ejemplo fichero,

image, manuales y upfile. Además, dentro de elias se encuentran los JSP y el Módulo Web (WEB-INF), donde están los Beans y los Servlets. Vea la Figura 5.23.

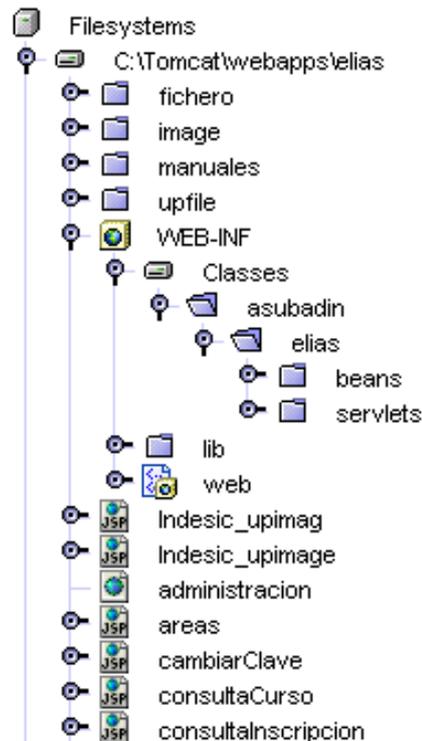


Figura 5.23 : Estructura del Directorio de la Aplicación

5.3 Pruebas

Una definición hecha en el sitio web de alarcos dice lo siguiente:

“Una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan y registran y se realiza una evaluación de algún aspecto”¹⁶

Las pruebas son una de las etapas que se debe considerar en el desarrollo del Software, ya que mediante ellas se puede identificar las fallas que una aplicación pueda tener.

¹⁶ Pruebas del Software, <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema09.pdf>

Para asegurar que las pruebas sean realizadas correctamente, es necesario basarse en un estándar. Para este caso se ha decidido seguir las normas IEEE Std. 829, que describe la forma de realizar las pruebas de software.

5.3.1 Tipos de Pruebas

5.3.1.1 Prueba Modular

Consiste en realizar pruebas a cada módulo, de modo que se pruebe a cada elemento posible.

5.3.1.2 Prueba de Integración

Este tipo de prueba se realiza a medida que los diferentes módulos del sistema se integran. El objetivo fundamental de esta prueba es comprobar que las interfaces entre los distintos módulos son correctas.

Según Jaime Ramírez, desarrollador de software de la Universidad de Madrid, comenta que existen tres posibles estrategias de integración, que a continuación se describe:

“Estrategias de integración

- *De arriba a abajo (top-down): Consiste en empezar la integración y la prueba por los módulos que están en los niveles superiores de abstracción, e integrar incrementalmente los niveles inferiores.*
- *De abajo a arriba (bottom-up): Consiste en empezar la integración y la prueba por los módulos que están en los niveles inferiores de abstracción, e integrar incrementalmente los niveles superiores.*

- *De big-bang: Consiste en integrar y probar todo al mismo tiempo.*¹⁷

5.3.1.3 Prueba del Sistema

Se realiza cuando se han integrado todos los módulos, y su objetivo es comprobar que el sistema satisface los requisitos del usuario, tanto los funcionales como los no funcionales.

5.3.1.4 Prueba de Aceptación

Se realiza una vez que el sistema se ha implantado en su entorno real de funcionamiento, y su objetivo es demostrar al usuario que el sistema satisface sus necesidades.

5.3.2 Métodos de Pruebas

Los métodos de prueba se refieren al enfoque que se aplica para encontrar casos de pruebas, y de esa forma se puede determinar su grado de cobertura.

Existen tres métodos de pruebas que son, de Caja Blanca, de Caja Negra y el aleatorio.

5.3.2.1 Método de Caja Blanca

Con este método se analiza la estructura interna del programa, se analizan los pasos de ejecución del programa.

¹⁷ Jaime Ramírez: Tipos de Pruebas, <http://lml.ls.fi.upm.es/~jramirez/ed2/pruebas.ppt>

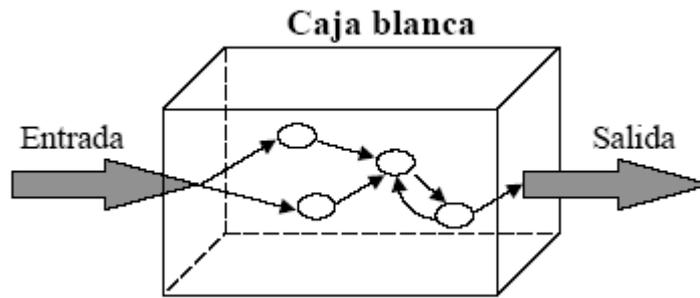


Figura 5.24: Método de Caja Blanca

5.3.2.2 Método de Caja Negra

Este método se centra en las funciones del programa, es decir se enfoca en la entrada y salida de datos.

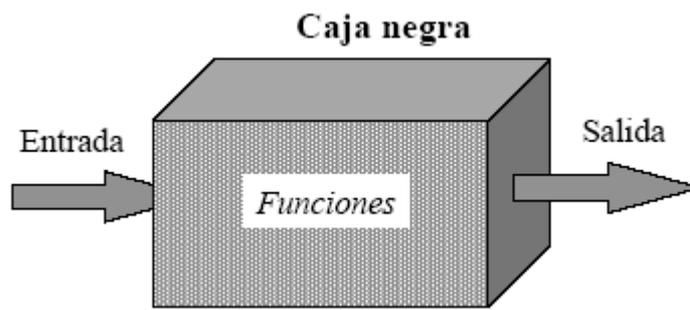


Figura 5.25: Método de Caja Negra

5.3.2.3 Método Aleatorio

Este método utiliza modelos estadísticos que representan las entradas de datos y a partir de ellos se crea los casos de prueba.

5.3.3 Plan de Pruebas

El plan de pruebas, es un documento en el que se describe las características que se probarán en el software.

En el plan hay que identificar también el tipo de prueba que se realizará, el método que se usará y los casos de prueba que se probarán.

La descripción del plan de pruebas para el presente proyecto, según el estándar IEEE Std. 829, los podrá encontrar en el Anexo G: *Plan de Pruebas*.

5.3.4 Ejecución de Pruebas

La ejecución de las pruebas se realizará a cada uno de los casos de uso que para las pruebas se denominan casos de prueba, ya que éstos representan a los módulos de la aplicación y según la descripción hecha en la planificación de las pruebas, se ha escogido hacer pruebas modulares (a cada módulo), por ende, se realizará para los casos de uso diseñado.

A continuación se presenta la ejecución de las pruebas:

Caso de Prueba: 0	<i>Inicio de Sesión</i>	
	Entradas	Salidas
	1. Datos al azar (datos incorrectos). 2. Datos de un usuario registrado.	1. Regresa a la pantalla de inicio de sesión. 2. Ingresa a la aplicación con las opciones según el perfil de usuario.
Resultado de la Prueba:	La validación de usuarios esta correcta para el inicio de sesión.	
Observación:	Cuando los datos ingresados no sean los correctos, debe aparecer un mensaje que indique que el usuario ingresado no existe.	

Caso de Prueba: 1	<i>Crear usuarios</i>	
	Entradas	Salidas
	1. Se ingresa datos de inicio de sesión como becario. 2. Se ingresa datos de inicio de sesión como Administrador.	1. Aparece la pantalla de la aplicación con opciones para el becario. 2. Aparece la pantalla con opciones para el Administrador.

<p>3. Se escoge la opción Administrador en el link de la derecha.</p> <p>4. Se ingresa los datos de un usuario con perfil de Becario.</p>	<p>3. Aparece el menú del administrador, con un fallo.</p> <p>4. Vuelve a aparecer la misma ventana de crear usuarios.</p>
<p>Resultado de la Prueba:</p>	<ul style="list-style-type: none"> • Se verifica que los usuarios que no tiene el perfil de Administrador, no pueden crear usuarios. • Se puede verificar que el usuario administrador es quien dispone del permiso para crear usuarios. • Se produce el fallo que se indica en la Figura 5.26. Este error es de comentario hecho en el código. • Los datos del usuario se registraron correctamente, sin embargo, no hay un mensaje que indique si se registró o no el usuario.
<p>Observación:</p>	<p>Es necesario que se arregle el error que ha producido el comentario puesto en el código.</p> <p>Se debe manejar los mensajes que confirmen el resultado de la operación.</p>

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*" errorPage="" %>
```

Módulo Administrador

[Mantenimiento de Usuarios](#)

[Establecer Nota](#)

Figura 5.26: Prueba para Crear Usuarios

Caso de Prueba: 2	<i>Registrar Becarios</i>	
Entradas	Salidas	
<p>1. Se ingresa los datos de un usuario ya registrado.</p>	<p>1. Aparece un mensaje en rojo, en la parte superior del formulario que dice que el usuario ya está registrado.</p>	

2. Se ingresa los datos de un usuario nuevo.	2. Aparece un mensaje en la parte superior del formulario que indica que se ha registrado correctamente el becario.
3. Se ingresa el login de un becario y se envía a Buscar.	3. Los datos del becario buscado se recuperan correctamente y se muestran en el formulario.
Resultado de la Prueba:	El caso de uso de registrar becarios se encuentra correctamente desarrollado.
Observación:	

Caso de Prueba: 3	<i>Reporte de Becarios Inscritos y el Estado en que se encuentran</i>	
	Entradas	Salidas
	1. Se ingresa con perfil de Planificador. 2. Se ingresa con perfil de Tutor. 3. Se ingresa con perfil de Administrador.	1. Este perfil si dispone de la opción de reportes. 2. Tutor también dispone de la opción de reportes. 3. El administrador no dispone de la opción de reportes.
Resultado de la Prueba:	Le hace falta la opción de reportes al administrador, ya que según los casos de uso, el administrador también debe tener la posibilidad de obtener el reporte. Las pruebas se ha realizado sin datos de inscripciones, y la pantalla de reportes aparece sin ningún dato, lo cual es correcto.	
Observación:	Es necesario dar permiso al administrador para que obtenga el reporte.	

Caso de Prueba: 4	<i>Publicar Cursos</i>	
	Entradas	Salidas
	1. Se ingresa con perfil de Becario, Administrador y Tutor.	1. Estos usuarios no disponen del permiso para publicar cursos.

2. Se ingresa con perfil de Planificador.	2. Este usuario si dispone de la opción para publicar cursos.
3. Se selecciona un curso desde una unidad de disco (d:) y se ingresan los demás datos requeridos.	3. Al hacer clic en guardar, aparece nuevamente la ventana de publicar cursos.
Resultado de la Prueba:	La opción de publicar los cursos, dispone el Planificador y los demás usuarios no dispondrán de esta opción. Se guarda los datos del curso correctamente y el curso mismo se publica con éxito.
Observación:	Es necesario que aparezca un mensaje que indique si la operación fue exitosa o no.

Caso de Prueba: 5	<i>Subir Manuales</i>	
	Entradas	Salidas
	1. Se ingresa con perfil de Planificador. 2. Elegimos el link de Subir Manual. 3. Seleccionamos el manual e ingresamos los demás datos del formulario.	1. Muestra la ventana con las opciones del Planificador, en ella se encuentra Subir Manual. 2. Aparece la ventana de subir manuales 3. Al hacer clic en Guardar, regresa al formulario de manuales otra vez.
Resultado de la Prueba:	El proceso de subir el manual lo realiza correctamente, sin embargo, debería haber un mensaje que indique que se ha subido correctamente el curso o no.	
Observación:	Controlar el despliegue de mensajes para las operaciones que se realice.	

Caso de Prueba: 6	<i>Crear Preguntas</i>	
	Entradas	Salidas
	1. Se ingresa con perfil de Tutor. 2. Seleccionamos el curso del combo y escribimos la pregunta que queremos realizar para el curso seleccionado.	1. Aparece la ventana con las opciones para el Tutor. 2. Al guardar la pregunta nos aparece la ventana de Preguntas.

Resultado de la Prueba:	<p>El módulo de crear preguntas está ligado con los cursos y con las respuestas, ya que la pregunta debe pertenecer a un curso en particular y cada pregunta debe tener 4 posibles respuestas, de las cuales una debe ser la verdadera y las otras no.</p> <p>Según como se ha planteado, al ingresar una pregunta nos aparece la ventana de respuestas.</p> <p>Según esta característica, este módulo de crear preguntas se encuentra bien.</p>
Observación:	

Caso de Prueba: 7	Crear Respuestas	
	Entradas	Salidas
Resultado de la Prueba:	<ol style="list-style-type: none"> 1. Datos de la pregunta (que viene desde la ventana de preguntas). 2. Se ingresa la respuesta para la pregunta y se escoge si es verdadero o falso. 3. Se sigue ingresando las respuestas para la pregunta, hasta un total de 5 respuestas. 4. Con otra pregunta se ingresa hasta 3 respuestas. 	<ol style="list-style-type: none"> 1. Aparece la ventana de respuestas, en el primer cuadro de texto, podemos observar la pregunta planteada, la misma que no se puede modificar. 2. Al dar clic en guardar, se muestra la misma ventana de respuestas, con el mensaje de que se guardó la respuesta. 3. Al dar clic en guardar, se sigue mostrando la misma ventana de respuestas 4. Tampoco me da error por ingresar menos respuestas que las necesarias.
Observación:	<p>En este módulo no se controla que se ingrese hasta 4 preguntas, sin embargo, el hecho de que se ingrese más respuestas no afecta para el examen.</p> <p>Tampoco tenemos errores por ingresar menos preguntas que las definidas.</p> <p>Sería de gran ayuda mostrar un indicador del número de preguntas que se ha ingresado.</p>	

Caso de Prueba: 8		Consultar Notas	
Entradas		Salidas	
1. Se ingresa con datos de perfil de becario. 2. Se escoge la opción de Consultar nota, que el becario dispone.		1. Se muestra la ventana de becarios y las opciones para este perfil. 2. Muestra una ventana en la que aparece el nombre del curso, nota, observación y el número de inscripción para ese curso.	
Resultado de la Prueba:		Las notas que aparecen en esta consulta depende de cada becario, es decir cada becario podrá observar sus notas y no las de los demás. La consulta se realiza correctamente, en caso de que el becario no tenga ningún examen registrado, el reporte no presenta nada.	
Observación:			

Caso de Prueba: 9		Consultar Cursos Disponibles	
Entradas		Salidas	
1. Ingresar a la aplicación con perfil de usuario. 2. En el menú de la izquierda, se escoge la opción Menú Becario. 3. Se escoge la opción Consultar Curso.		1. Aparece la ventana con las opciones para el becario. 2. Aparece un menú de opciones que pueden escoger los becarios. 3. Muestra la lista de cursos disponibles en la aplicación, así como el pre-requisito en caso de que existiera.	
Resultado de la Prueba:		La consulta de los cursos disponibles se los ha obtenido satisfactoriamente.	
Observación:			

Caso de Prueba: 10		Inscribirse en Curso	
Entradas		Salidas	
1. Se ingresa con perfil de becario.		1. Aparece la ventana con las opciones para los becarios.	

<p>2. Escogemos la opción Menú Becario.</p> <p>3. Escogemos la opción de Inscripción.</p> <p>4. Elegimos el curso para la inscripción y hacemos clic en Guardar.</p>	<p>2. Aparece un submenú de opciones que pueden escoger los becarios.</p> <p>3. Muestra otra ventana con la lista de los cursos en los que el becario puede inscribirse.</p> <p>4. Nos muestra un mensaje que dice que la inscripción se ha realizado correctamente.</p>
<p>Resultado de la Prueba:</p>	<p>Los controles necesarios para hacer esta validación se encuentra bien, por lo que la prueba se realiza con éxito.</p> <p>Esta validación es una de las más importantes de la aplicación, ya que aquí se controla que el becario pueda tomar los cursos permitidos, mientras que los no permitidos no se los muestra, y esto se lo realiza por cada uno de los becarios.</p> <p>En este módulo se verifica también si el becario ha aprobado el curso o no, o si aún esta pendiente, de acuerdo a estos resultados se presenta el curso en el que se puede inscribir.</p>
<p>Observación:</p>	

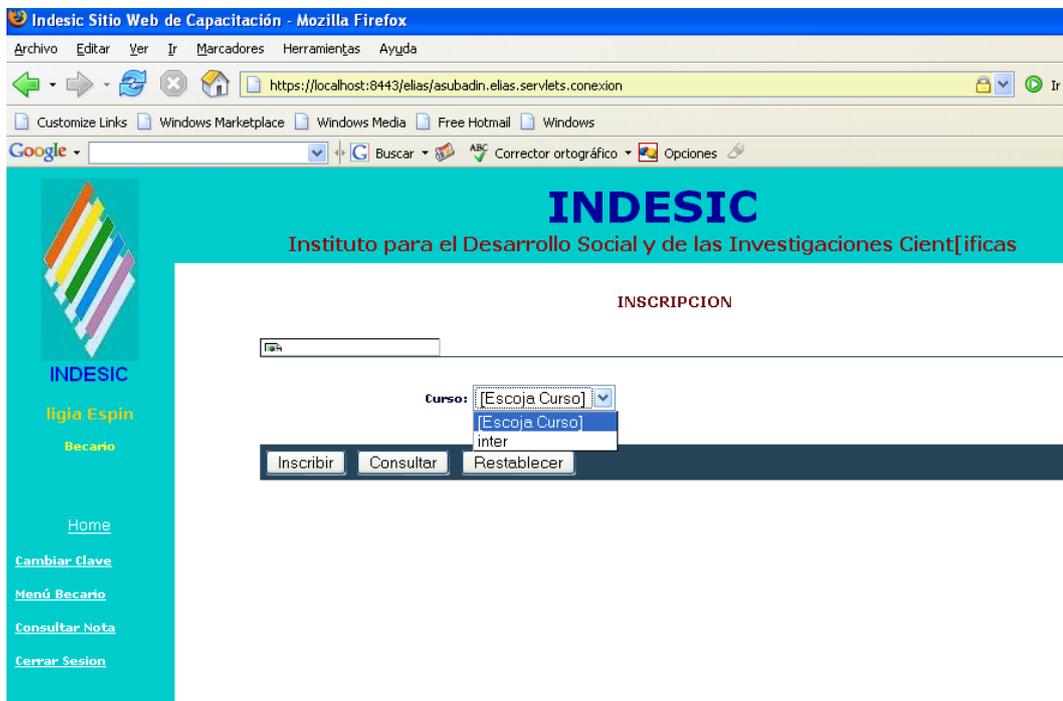


Figura 5.27: Prueba de inscripción

Caso de Prueba: 11		Tomar Curso Inscrito	
Entradas		Salidas	
1. Se ingresa a la aplicación con perfil de becario. 2. En el submenú, escogemos la opción Iniciar Curso. 3. Se escoge un curso de la lista.		1. Aparece la ventana con las opciones para los becarios, en la que hacemos clic en Menú Becario. 2. Nos muestra la lista de cursos que puede tomar ese becario. 3. Aparece el curso que se haya elegido	
Resultado de la Prueba:		El becario antes de poder ingresar al curso debe primero inscribirse en un curso. Esto se ha controlado correctamente, ya que si no está inscrito en ningún curso, no podrá ingresar a ninguno.	
Observación:		El formato para los cursos que se vayan a publicar es html, no puede ser diseñado en otro formato, ya que se requiere que el becario pueda acceder en línea.	

Caso de Prueba: 12		Rendir Examen	
Entradas		Salidas	
1. El ingreso a la aplicación se hace con perfil de becario. 2. En el submenú, escogemos la opción Rendir Examen. 3. Escogemos el curso del que queremos rendir el examen. 4. Para cada pregunta, seleccionamos una respuesta y Aceptamos. 5. Escogemos otro curso para rendir el examen, pero este curso no tiene las preguntas completas ni las respuestas completas.		1. Aparece la ventana con las opciones para los becarios. En el menú de la izquierda hacemos clic en Menú Becario. 2. Nos muestra la lista de cursos inscritos y que podemos dar el examen. 3. Aparece el formulario de Exámenes, con las preguntas y respuestas creadas por el tutor. 4. se muestra un mensaje que nos indica la nota obtenida y una observación (aprobado o reprobado) 5. Aparece el formulario de Exámenes, con las preguntas y respuestas repetidas.	

6. Para cada pregunta, seleccionamos una respuesta y Aceptamos.	6. La operación de calificar y asignar la nota se realiza correctamente, pero no aparece el mensaje del estado de la operación.
Resultado de la Prueba:	Para el caso de los cursos que no se han ingresado las preguntas y respuestas completas para el examen, se produce un error, ya que no se controla que se haya ingresado las preguntas y respuestas completamente antes de mostrar el formulario.
Observación:	Debería verificar si las preguntas y respuestas están ingresadas completamente antes de mostrar el formulario.

Caso de Prueba: 13	Consultar Manuales
Entradas	Salidas
<ol style="list-style-type: none"> 1. Se ingresa con perfil de Becario. 2. Se escoge la opción Consultar Manuales. 	<ol style="list-style-type: none"> 1. Aparece la ventana de la aplicación con las opciones para el becario. 2. Se despliega la ventana con los manuales disponibles, y cada manual dispone de un link que permite descargar el manual.
Resultado de la Prueba:	La Consulta de manuales esta disponible para los becarios, lo cual esta correcto de acuerdo a los casos de uso, por lo tanto la prueba fue exitosa.
Observación:	

Caso de Prueba: 14	Bajar Manuales
Entradas	Salidas
<ol style="list-style-type: none"> 1. De la lista de cursos que nos aparece en la consulta de cursos, hacemos un clic en el link del curso que queremos descargar. 	<ol style="list-style-type: none"> 1. Inmediatamente, aparece el cuadro de diálogo que permite descargar el archivo.

2. Se escoge la opción de guardar el curso y se especifica el lugar donde se guardará.	2. El curso es guardado en el disco donde se haya elegido.
Resultado de la Prueba:	El proceso cumple el requerimiento del caso de uso de permitir bajar el curso a los becarios, por lo tanto se ha cumplido exitosamente la prueba.
Observación:	

5.4 Seguridad del Sistema

Para la aplicación que se ha desarrollado, se ha decidido manejar seguridades mediante certificados emitidos por el servidor, de esta forma el usuario que quiera acceder a la aplicación Web, primero verificará el certificado, y si el certificado está correcto, entonces realiza la conexión.

Este certificado puede ser emitido por una entidad certificadora, como por ejemplo VerySing, que es un emisor de certificados para operaciones por Internet.

Para este caso, no se realizará la adquisición de un certificado de otro servidor, si no que se configurará el servidor de aplicaciones Tomcat con soporte para SSL (Secure Socket Layer). Esto se lo realizará usando una herramienta que se incluye en el JDK de java (J2sdk), que es el Keytool.

Se configurará el SSL a 128 bits, esto significa que la encriptación del certificado se realiza usando 128 bits de datos.

Los pasos para configurar el SSL con keytool del J2sdk son los siguientes:

1. Buscar en el directorio de instalación de Tomcat, el fichero server.xml (para este caso, se encuentra en C:\Tomcat\conf).
2. Dentro del fichero Server.xml buscar hasta donde encontremos el siguiente elemento: <!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 --> luego debemos quitar los comentarios de la etiqueta <Connector
3. Salir del fichero guardando los cambios realizados.
4. Ingresamos al símbolo del sistema e ingresamos hasta la carpeta bin del J2sdk.
5. Entonces digitamos lo siguiente: keytool -genkey -alias tomcat -keyalg RSA.
6. Posteriormente debemos ingresar una clave para el almacén del certificado, que para este caso se ha usado la clave que tiene Tomcat por defecto que es, changeit.
7. Luego, nos pedirá los datos de nombre del servidor, unidad organizativa, organización, provincia, ciudad y país. Estos datos son importantes para generar el SSL.

Confirmamos estos datos y se generará el certificado SSL.

La Figura 5.28, indica un ejemplo de la configuración del certificado SSL para Tomcat.

```

C:\j2sdk1.4.0\bin>keytool -genkey -alias prueba -keyalg RSA
Escriba la contraseña del almacén de claves: changeit
¿Cuáles son su nombre y su apellido?
[Unknown]: Elias Asubadin
¿Cuál es el nombre de su unidad de organización?
[Unknown]: Sistemas
¿Cuál es el nombre de su organización?
[Unknown]: Tesis
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Ambato
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Tungurahua
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: Ecuador
¿Es correcto CN=Elias Asubadin, OU=Sistemas, O=Tesis, L=Ambato, ST=Tungurahua, C
=Ecuador?
[no]:

```

Figura 5.28: Configuración SSL para Tomcat

Con esto se ha logrado, configurar el SSL para Tomcat. Cuando ingresemos a la aplicación nos preguntará si el servidor al que queremos conectarnos es el correcto, aceptamos y luego de esto, en la barra de dirección se mostrará con https://, esto nos indica que estamos navegando en un sitio seguro.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- El software de código abierto y software libre, con respecto al software propietario, brindan una importante fuente de desarrollo e innovación de nuevos productos y aplicaciones, ya que se encuentran al alcance de los desarrolladores.
- La aplicación desarrollada puede ser usada en múltiples plataformas, como puede ser en Linux, Windows, Unix, Solaris. Esto permite que la institución que vaya a usar el producto, pueda sentirse libre de optar por una plataforma y luego pueda cambiarse a otra y no quedarse atado con una que no le convenga.
- Una buena práctica en el desarrollo del software, es basarse en estándares y normas que permitan obtener el producto adecuado y apropiado para las necesidades del cliente. Las normas, IEEE 830 para la especificación de requerimientos de software y la IEEE 829 para la ejecución de pruebas del software, han permitido en este proyecto, obtener una aplicación acorde a las necesidades de la institución que los requiere.

- Para que un software sea desarrollado con éxito, es importante que se defina una metodología a seguir, ya que esto permitirá optimizar los recursos de la mejor manera.
- Aprender a desarrollar aplicaciones para Internet utilizando la tecnología Java, al principio resulta bastante difícil ya que hay que construir todos los componentes y clases que se necesita. Sin embargo, luego de conocer el lenguaje se puede avanzar más rápidamente con el desarrollo ya que se puede reutilizar componentes, además Java proporciona seguridad y eficiencia de manejo de datos para redes.

6.2 RECOMENDACIONES

- Antes de desarrollar cualquier aplicación de cualquier índole, es recomendable realizar primero el levantamiento de la información, y con ello se debe proceder a realizar el diagrama de Casos de Uso, que es una forma efectiva de modelar el software.
- Cuando se diseñe los casos de uso del software, es necesario documentar los casos de uso y actores diseñados, con esto se dispondrá por escrito lo que se ha diagramado.
- Con respecto a los productos de software que se han usado para el desarrollo de la aplicación, algunos de ellos son de código abierto, pero todos ellos se los puede conseguir libremente sin costo y funcionan adecuadamente con la aplicación. Sin embargo, para aplicaciones

comerciales cuyas transacciones sean de gran impacto, se debería cambiar a herramientas más sofisticadas y que tengan garantía legal.

- Con respecto al software desarrollado, se puede seguir innovando la aplicación, ya que se puede incrementar otros módulos adicionales, de modo que la aplicación que actualmente es satisfactorio para el INDESIC, en el futuro sea un software comercial que la puedan adquirir instituciones que se dediquen a la capacitación y requieran de un sistema con estas características.
- Se recomienda el uso de la tecnología Java, que es una plataforma muy estable y confiable para desarrollo de aplicaciones que tengan que usar las redes de datos, además Java es la tecnología sobre el que se basa otras herramientas importantes lo que le hace aún más confiable.

BIBLIOGRAFÍA

Libros

- Pressman Roger, Ingeniería del Software: un enfoque práctico, Cuarta Edición, McGraw-Hill, Madrid, 1998.
- Meyers Nathan, Programación Java en Linux, Edición Especial, Prentice Hall, Madrid, 2000.
- González José, Red Hat Linux 8, MacGraw-Hill, Madrid, 2000.
- Phil Hanna, JSP: Manual de Referencia, MacGraw-Hill, Madrid, 2002.
- James R. Groff, Paul N. Weinberg, Guía Lan Times de SQL, Tercera Edición, McGraw-Hill, Madrid, 1999
- Perelló Julio, Historia de la Educación, Editorial Don Bosco, Quito, 1967.

Internet

- Ramírez José Antonio, E-learning,
<http://www.soyentrepreneur.com/pagina.hts?N=13771&Ad=S>
- Categorías de Software Libre y No Libre,
<http://www.gnu.org/philosophy/categories.es.html#OpenSource>
- Java Server Pages, http://es.wikipedia.org/wiki/Java_Server_Pages
- García María Lorena, Concepto de Apache, <http://g.unsa.edu.ar/instekeko/>
- Apache TOMCAT,
http://www.prodigyweb.net.mx/pupis/Cursos/jsp/ch02/pg01_text.htm
- Seguridad en la transmisión, www.tejedoresdelweb.com
- Estándares para el desarrollo de software, <http://www.ieee.org/portal/site>,
<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema09.pdf>

Tesis

- Benjamín Wuilfrido Suqui Cepeda, Análisis, Diseño e Implementación de una Intranet en Codenpe, Escuela Politécnica del Ejército, Facultad de Ingeniería en Sistemas e Informática, Julio del 2004.

Normas

- Norma IEEE Std. 830, Estándar recomendado para la Especificación de Requerimientos de Software, <http://www.ieee.org/portal/site>
- Norma IEEE Std. 829, Estándar para la elaboración y documentación de pruebas de software, <http://www.ieee.org/portal/site>

Otros

- Montesinos Gustavo, Educar a distancia, El Heraldo, Jueves 5 de mayo del 2005, Ambato

ANEXO A: ENCUESTAS

El Propósito de esta encuesta es, determinar las necesidades y requerimientos específicos que tiene el INDESIC, con respecto al sitio Web de cursos de capacitación.

1. ¿Cómo le gustaría que fuese la presentación de la página Web de Capacitación del INDESIC? Seleccione solo una opción.

- Simple, sin ilustraciones y de acceso veloz.
- Sencillo, con algunas ilustraciones, entendible y de rápido acceso.
- Sofisticado, con muchas ilustraciones, de fácil captación pero que requiere un ancho de banda considerable.
- Muy Sofisticado, con ilustraciones, animaciones y Videos, de muy fácil captación, pero que requiere gran ancho de banda

2. ¿Qué servicios cree que debería contemplar el becario? Puede seleccionar varias opciones.

- Control de claves
- Consulta de cursos disponibles
- Inscripción en un curso
- Acceso al curso
- Evaluación en línea
- Consulta de Manuales
- Descarga de Manuales
- Consulta de Notas
- Foros de Discusión
- Links hacia otros sitios relacionados
- Video Conferencias
- Acceso a CDs Multimedia
- Chat con miembros del INDESIC
- Chat con otras personas del Internet
- Formulario para solicitar asistencia
- Formulario para enviar sugerencias
- Correo personal con el dominio del INDESIC

3. ¿Qué Otros Servicios cree que debería contemplar la Aplicación? Puede seleccionar varias opciones.

- Mantenimiento de Usuarios
- Mantenimiento de Cursos
- Mantenimiento de Formularios de examen
- Control de concatenación de cursos
- Eliminar Inscripción
- Modificar Notas
- Consulta de Notas
- Reportes
- Cambio de claves

4. Señale los aspectos de seguridad que se debería manejar (puede seleccionar varias).

- Notas seguras, que no se pueda modificar desde el sitio por el Usuario.
- Notas abiertas, que se pueda modificar desde el sitio por el usuario.
- Contenido confiable, que sea publicado solo por el planificador.
- Usuarios libres, que cualquier usuario pueda inscribirse y tomar el Curso.
- Usuarios controlados, que hayan sido creados previamente por el Planificador.
- Sitio seguro, que el sitio Web esté protegido por un protocolo que garantice la seguridad.

ENCUESTA PARA SELECCIONAR EL CURSO A PUBLICAR

El propósito de esta encuesta es con la finalidad de conocer las necesidades específicas en cuanto a los cursos de computación que los becarios del INDESIC necesitan.

1. De la lista que a continuación se presenta, señale con una x, máximo tres cursos que le interesaría tomar.

- Linux
- Windows
- Internet
- Correo Electrónico
- Word
- Excel
- Power Point
- Access
- Publisher
- Crear Páginas Web
- Bases de Datos
- Graficadotes
- Macromedia Dreamweaver
- Macromedia Flash
- Front Page
- Mantenimiento de PCs
- Redes

2. ¿Por qué considera que los cursos seleccionados arriba, los requiere?

- Por curiosidad
- Necesidad de trabajo
- Necesidad de estudio
- Por Obligación

3. ¿Cómo le ayudará, si aprende el curso seleccionado?

- Mejorar en el estudio
- Aprender más
- Mejorar Laboralmente
- Ser más eficiente

ANEXO B: ESPECIFICACIÓN DE REQUERIMIENTOS DEL PROYECTO

1. Introducción

El presente documento hace referencia a la especificación de requerimientos de software correspondiente a la Aplicación Web para la Gestión y Planificación de Cursos de Capacitación del INDESIC. Este documento se ha desarrollado siguiendo las normas IEEE 830 con el afán de garantizar la calidad y desempeño del producto que se va a crear.

En adelante, para no entrar en redundancia y con la finalidad de mejorar la redacción, se ha decidido poner un nombre corto a la *Aplicación Web para la Gestión y Planificación de Cursos de Capacitación del INDESIC*, mismo que se denominará:

e-curso-os

1.1 Propósito

El propósito fundamental del presente documento es, detallar las especificaciones de requerimientos para e-curso-os y de esta manera poder obtener las necesidades puntuales del cliente.

1.2 Alcance

El sistema e-curso-os está planificado alcanzar la etapa de implementación. No se realizará la implantación, pruebas ni corrección, sin embargo se puede realizar todo aquello, luego de coordinar con el cliente, y posteriormente a la culminación de este proyecto de tesis.

El ciclo de vida en espiral esta compuesto de 6 etapas que son: Comunicación con el cliente, Planificación, Análisis de Riesgos, Ingeniería, Construcción y adaptación y evaluación del cliente.

Por lo tanto para desarrollar el e-CURSO-OS, se tomará en cuenta las siguientes etapas:

- **Etapa 1:** Comunicación con el cliente, en esta etapa se busca la información necesaria del cliente, mediante encuestas, entrevistas, reuniones, etc.
- **Etapa 2:** En esta etapa de la Planificación, es donde justamente se planifica las fechas de reunión con el cliente, las fechas para presentar prototipos. También se planifican costos, personal y todo lo que involucre en el desarrollo de software.
- **Etapa 3:** La etapa de la Ingeniería, es donde se determina las tareas para la construcción del software. Como resultado de esta etapa están, la Especificación de Requerimientos de Software, que se encuentra en el Anexo B, así también se obtiene los diagramas, que son la representación arquitectónica del software a desarrollar.
- **Etapa 4:** La Etapa de Construcción y Adaptación, es donde se pone en marcha el desarrollo del software. Para la tesis, en esta etapa se ha realizado el desarrollo del software, cuyos detalles se presenta en el Capítulo V.
- **Etapa 5:** Evaluación del Cliente, En esta etapa es donde se visita nuevamente al cliente y se hace una presentación del trabajo realizado,

con lo que se obtiene más especificaciones para luego mejorar o añadir nuevas funciones al producto.

El principio de la metodología en espiral es, partir de algo simple hasta obtener el producto final. Cada vuelta al espiral se debe pasar por las etapas indicadas con el fin de ir mejorando hasta obtener el sistema adecuado.

Para el caso de e-CURSO-OS, se pasará por todas las etapas del modelo en espiral, exceptuando la etapa de análisis de riesgos que impidan culminar la aplicación, que para este caso no es importante analizar los riesgos de no terminar el proyecto.

1.3 Definiciones, abreviaciones, y siglas

1.3.1 Definiciones

Web.- Se le denomina así a un sitio o página que se encuentra alojado en algún lugar del Internet y que provee información electrónica para usuarios conectados al Internet.

Aplicación Web.- software desarrollado para cumplir con una necesidad específica, cuyo servicio es proporcionado a través del Internet.

JAVA.- La enciclopedia Encarta, define a Java de la siguiente forma:

“Lenguaje de programación orientado a objetos desarrollado por la empresa Sun Microsystems en 1995 y que se ha extendido ampliamente en World Wide Web.

Es un lenguaje de alto nivel y propósito general similar a C++, con marcadas características de seguridad y transportabilidad. Este lenguaje define una máquina virtual independiente de la plataforma donde se ejecuta, que procesa programas, llamados Applets, descargados desde el servidor Web. Además, debido al modo de ejecución de los Applets, este lenguaje es muy seguro frente a la presencia y ataque de virus informáticos.”¹⁸

Servlet.- Nathan Meyers manifiesta lo siguiente:

“Los servlets son programas Java que se ejecutan en servidores web. Son los equivalentes en Java a los guiones CGI, que se ejecutan en respuesta a las solicitudes del navegador en URL particulares”¹⁹

Encriptación.- Es una técnica usada para proteger los datos que viajan por una red, para que no pueda ser accedido por extraños.

Becario.- Se refiere a la persona que es beneficiaria de una beca otorgada por el INDESIC para que estudie una carrera universitaria.

Planificador.- Se refiere a la persona que se encargará de ingresar los datos del curso, temas del curso, preguntas y los datos del becarío.

Java bean.- un Java Bean se refiere a un componente que se utiliza en Java y que permite agrupar funcionalidades para formar parte de una aplicación.

1.3.2 Siglas y Abreviaturas

¹⁸ JAVA: Enciclopedia Encarta 2003

¹⁹ Meyers, Nathan: Edición Especial Programación Java en Linux; Prentice Hall; Madrid; 2000

e-curso-os: sigla usada para referirse a Aplicación Web para la Gestión y Planificación de Cursos de Capacitación, Usando Herramientas Open Source.

INDESIC: Instituto para el Desarrollo Social y de las Investigaciones Científicas.

SSL: Secure Socket Layer, que traducido al español significa capa de seguridad de socket.

JDK: Java Development Kit (Kit de Desarrollo Java), es la herramienta de desarrollo de software de referencia de Sun Microsystems.

SDK: Software Development Kit, conjunto de aplicaciones para desarrollar programas para un determinado entorno.

JRE: Java Runtime Environment, es el entorno donde se ejecutan las aplicaciones Java.

JVM: Java Virtual Machina, es la Maquina Virtual de Java, que representa el núcleo del entorno runtime, permite interpretar las clases de java.

J2SDK: Java 2 Software Development Kit, es el kit de desarrollo de software Java 2, edición estándar, que es el nuevo nombre para el SDK asociado con la plataforma 2 de java.

1.4 Referencias

Las referencias las podrá encontrar en la bibliografía correspondiente al presente proyecto.

1.5 Vista general / Resumen del Documento

Este documento contiene información general del producto, en el que se describe la funcionalidad, el ambiente en el que operará, los usuarios, las interfaces que tendrá y las restricciones a las que está sujeto el producto.

2. Descripción general

El e-CURSO-OS tiene su propósito en brindar soporte para cursos de capacitación a los becarios del INDESIC, que muchos de ellos viven en provincias y tienen que viajar para poder asistir a los cursos.

Para ello habrá que publicar los cursos constantemente y actualizarlos. Además los interesados deberán poder inscribirse en el curso.

El sistema deberá permitir acceder a los cursos inscritos y al finalizar el curso deberá realizar una evaluación en línea y asentar la nota automáticamente.

Adicionalmente, se controlará si el curso tiene pre-requisito o no, aunque esto es muy difícil que se dé, para el caso del INDESIC.

Finalmente, la aplicación debe permitir bajar archivos de manuales que el INDESIC quiera subir al Internet.

Todo esto debe estar disponible para los becarios del INDESIC, y debe ser accedido mediante el nombre de usuario y una clave que el administrador asigne a cada becario.

2.1 Perspectiva del producto

El e-CURSO-OS está diseñado para que cumpla el esquema cliente servidor, donde el cliente desde una computadora con sistema operativo Windows o Linux, pueda acceder a los servicios Web que necesita, entonces la aplicación que se encuentra en un servidor de Linux, debe procesar y proporcionar la solicitud respectiva.

2.1.1 Interfases de sistema

No es aplicable porque no se desarrollarán interfaces externas al sistema.

2.1.2 Interfaces de usuario

Para desarrollar las interfaces de usuario, se ha tomado en cuenta la variedad de usuarios que pueden acceder al servicio. Las computadoras que se conecten al Internet pueden ser de las más modernas, como también existe la posibilidad de que sean obsoletas, por lo tanto se deberá tomar en cuenta estas características.

- La resolución de la pantalla se ha definido en 800 X 600 píxeles.
- Las pantallas deben ser compatibles con Internet Explorer, Netscape y los navegadores más comunes.
- Las pantallas de usuario deberá tener el siguiente diseño:
 - En la parte superior se presentará el nombre de la institución con efectos animados

- En la parte superior izquierda debe aparecer el logotipo de la institución, misma que no tendrá animación.
 - En la parte izquierda bajo el logotipo, estará los enlaces de los diferentes servicios que brinda el INDESIC para los becarios.
 - Finalmente, en la parte derecha deberá aparecer toda la información que queremos presentar.
-
- El color que prevalecerá será el celeste, mismo que se aplicará al marco izquierdo y superior de la pantalla, mientras que en el resto de la pantalla prevalecerá el color blanco.

Cabe recalcar que a más de los becarios que acceden a los cursos, tenemos las pantallas del Administrado, Planificador y Tutor, quienes tendrán otras funciones adicionales, las cuales deben estar restringidas para los usuarios comunes. Sin embargo el modelo de las pantallas será la misma que se ha descrito.

2.1.3 Interfaces de hardware

Las interfaces para el hardware no son aplicables para este proyecto.

2.1.4 Interfaces de software

No se desarrollará interfaces para interactuar con otro software.

2.1.5 Interfaces de comunicación

El servicio de cursos de capacitación, está pensado para que pueda ser accedida mediante Internet, por lo tanto, para poder mostrar la página al

usuario, se debe usar los navegadores de Internet más comunes, lo que implica que funcionará bajo el protocolo TCP/IP.

2.1.6 Memoria

El servidor donde se encuentre alojada la aplicación, debe disponer de por lo menos 512 MB de memoria Ram.

Las computadoras de los usuarios deberán disponer de por lo menos 64 MB de memoria Ram, con el fin de que el servicio funcione adecuadamente.

2.1.7 Espacio en Disco

El sistema estará alojado en un servidor (PC) con Sistema Operativo Linux 9.0 y debe disponer de por lo menos 10 GB de espacio en Disco. Se ha considerado que los usuarios pueden hacer uso del servicio e-CURSO-OS con computadoras que tengan instalado Windows 98 o Linux 7.0 como mínimo.

2.1.8 Operaciones

Las características operacionales del sistema son las siguientes:

- a) Gran parte de las operaciones serán iniciadas por los usuarios.
- b) Las operaciones de crear usuario y crear becarios serán iniciados por el Planificador de los cursos.
- c) Las operaciones de mantenimiento de cursos y mantenimiento de manuales serán iniciados por el Planificador.

- d) El Tutor será quien inicie las operaciones de crear preguntas y respuestas para los respectivos cursos.
- e) La aplicación realizará la evaluación automáticamente luego de que el becario haya culminado el curso y de haber llenado un formulario de preguntas.

2.2. Funciones del producto

Las funciones que desempeñará el producto son:

2.2.1. Publicación de los cursos

Mediante esta función se pondrá un curso en el sitio del INDESIC, para que pueda ser tomado por los becarios.

2.2.2. Crear Usuarios

Esta función permite crear a los usuarios del sistema, realizar actualizaciones, cambios, y eliminación.

2.2.3. Inscripción en un curso de capacitación

Un becario puede inscribirse en los cursos que requiera.

2.2.4. Descargar materiales de capacitación

Esta función es importante, ya que mediante ésta se controlará la descarga de materiales desde el sitio del INDESIC.

2.2.5. Examen

Esta función genera un formulario del examen y verifica las respuestas para luego asentar la nota respectiva

2.2.6. Generar reporte por becario y los cursos en los que se ha inscrito

Mediante ésta se generará un reporte general de los becarios y sus cursos tomados.

2.2.7. Generar una consulta por cada becario, de las notas obtenidas en los cursos evaluados.

2.3 Características del usuario

Como se ha dicho, la aplicación web está dirigida para becarios y exbecarios del INDESIC, quienes son estudiantes universitarios o profesionales, por lo que se supone que tienen un conocimiento básico de computación, como para poder tomar un curso por Internet.

A continuación se hace una descripción de las características de los usuarios.

2.3.1 Departamento de Capacitación

El departamento de capacitación del INDESIC es quien actualmente se encarga de controlar los cursos, por lo tanto deberá asignar una persona para que publique los cursos previamente creados. Para ello deberá tener las siguientes características:

- a) Amplios conocimientos de Computación
- b) Saber planificar cursos
- c) Habilidad para realizar análisis de resultados

2.3.2 Becario

El becario es el usuario que ingresará al sitio para acceder a un curso por lo que no se le exige demasiado, solamente lo que a continuación se detalla:

- a) El usuario deberá tener conocimientos de Internet y saber como navegar por Internet.
- b) Decisión por auto aprendizaje

2.4 Restricciones para el desarrollador

La aplicación debe ser desarrollada para alojarse en Sistema Operativo Linux, pero los usuarios pueden acceder desde un Sistema Operativo Windows usando explorador u otro browser de Internet.

2.5 Suposiciones y dependencias

La aplicación que se desarrollará deberá ser alojada en un sitio web, por lo que el INDESIC debe mantener el Dominio activo.

Se requiere de un enlace de alta velocidad para que los usuarios puedan acceder al servicio, por lo que deberá mantener el enlace que ahora dispone.

La aplicación requiere de un servidor, por lo que se supone que el INDESIC proveerá uno o contratará un servicio para alojar la aplicación.

2.6 Requerimientos que no serán satisfechos en esta versión.

Esta versión de la aplicación web, no se integrará con ningún sistema exterior.

En esta versión no se ha contemplado la posibilidad de video conferencias, Chat ni correos electrónicos personales, sino únicamente la posibilidad de acceder a cursos vía Internet.

3. Requerimientos específicos

3.1. Requerimientos de interfases externos

3.1.1. Especificación de Casos de Uso

La Especificación de los Casos de Uso los puede encontrar en el *Anexo C: Diagramas de Diseño - Descripción de Casos de Uso y Actores.*

La Descripción de Los Casos de Uso se los a puesto en el Anexo C, con el fin de facilitar la comprensión, ya que está junto al diagrama de casos de uso.

3.1.2 Interfaces de hardware

No existen interfaces de hardware en este proyecto.

3.1.3 Interfases de Software

Las interfaces de software, no son aplicables a este proyecto.

3.2 Descripción de los Diagramas de Secuencia

3.2.1 Validar datos del usuario

3.2.1.1 Introducción

Función para becarios que quieren acceder a los servicios del INDESIC a través del Internet.

3.2.1.2 Entradas

Nombre de usuario y la clave.

3.2.1.3 Proceso

Leer Nombre de Usuario y Clave

Validar Nombre de Usuario y Clave

Si la identificación es correcta entonces

Mostrar la pantalla de servicios

Caso Contrario

Pedir que ingrese nuevamente los datos

Fin de validación

3.2.1.4 Salidas

Luego de validar al usuario, aparecerán las pantallas de servicio y el mensaje que pide ingrese nuevamente los datos del usuario, en caso de que los haya ingresado mal.

3.2.2 Inscribir en curso

3.2.2.1 Introducción

Los becarios pueden inscribirse en un curso de capacitación.

3.2.2.2 Entradas

Código del curso a tomar (para ello debe escoger un curso de la lista de cursos)

Código del Usuario.

3.2.2.3 Procesos

Recibir los datos del curso y del usuario

Verificar el número de cursos que está tomando

Si número de cursos que está tomando es menor que 2 entonces

Registrar datos de inscripción

Caso contrario

Enviar un mensaje, “primero debe terminar los otros cursos que está tomando”

Finalizar control

3.2.2.4 Salidas

Mensaje de la Inscripción: Satisfactorio si se ha inscrito correctamente y Error, si no se pudo registrar los datos.

3.2.3 Procesar Examen

3.2.3.1 Introducción

Si el usuario ha terminado un curso, puede rendir un examen que nos indique su nivel de aprendizaje

3.2.3.2 Entradas

Código de usuario

Código del curso

Código de Inscripción

3.2.3.3 Procesos

Verificar si el curso que ha tomado el usuario, lo concluyó o no

Si concluyó el curso entonces

Mostrar las preguntas correspondientes a ese curso

Leer la respuesta

Registrar las respuestas en la base de datos

Calificar cada respuesta del examen

Caso Contrario

Indicar que no puede rendir el examen todavía

Fin de control

Sumar las respuestas de las preguntas

Presentar en pantalla la nota obtenida

3.2.3.4 Salidas

Mensaje de haber rendido el examen y la calificación obtenida

3.2.4 Consultar Nota

3.2.4.1 Introducción

Obtiene la nota de uno o varios cursos que haya tomado un becario.

3.2.4.2 Entradas

Código de usuario

Código del curso

3.2.4.3 Procesos

Consultar los cursos que el becario ha tomado

Buscar las notas de los cursos

Si encontró las notas de los cursos entonces

Sacar los totales de cada examen

Mostrar en pantalla las notas y los nombres de los respectivos cursos

Caso Contrario

Enviar un mensaje de error

Fin de control Si.

3.2.4.4 Salidas

Pantalla de las notas, o mensaje de error cuando ésta se produzca.

3.2.5 Crear Becario

3.2.5.1 Introducción

Esta función permite crear un becario y asignarle sus datos para que pueda acceder a los servicios desde el Internet.

3.2.5.2 Entradas

Datos del becario

3.2.5.3 Procesos

Leer datos

Asignar un perfil de usuario

Verificar la existencia del becario

Si no encontró al becario entonces

Registrar los datos leídos en la tabla de becarios

Mostrar mensaje, “El becario ha sido registrado”

Caso Contrario

Mostrar mensaje, “El becario ya existe”

Fin de control Si

Salir de la pantalla de crear becario

3.2.5.4 Salidas

Mensajes de registro o de error según el caso.

3.2.6 Publicar curso

3.2.6.1 Introducción

El Planificador debe ingresar cursos antes de que el becario las pueda tomar.

3.2.6.2 Entradas

Datos del curso

3.2.6.3 Procesos

Leer los datos del curso

Verificar si no existe el curso

Si el curso no existe entonces

Registrar los datos en la tabla de cursos

Enviar mensaje “El curso ha sido registrado”

Cargar el curso físicamente al sitio web

Caso Contrario

Enviar un mensaje de error “el curso ya está registrado”

Fin de control

Salir de la pantalla de publicación de cursos

3.2.6.4 Salidas

Mensaje de curso registrado, o de error según el caso.

3.2.7 Generar Reporte

3.2.7.1 Introducción

Este reporte es requerido exclusivamente por el Planificador, en el que se presenta el nombre del becario, los cursos que ha tomado con sus respectivas notas y la fecha en que ha iniciado y finalizado el curso.

Se debe especificar la fecha de inicio y fin de la búsqueda

3.2.7.2 Entradas

Fecha de inicio y fin para la búsqueda

3.2.7.3 Procesos

Leer la fecha de inicio y fin de la búsqueda

Consultar en la tabla de inscripción en el rango de fechas, los códigos del becario y curso

Si existe inscripción en esas fechas entonces

Verificar si el becario ha terminado el curso

Si ha terminado el curso entonces

Consultar el nombre del becario

Consultar los títulos de los cursos tomados

Consultar las notas obtenidas

Presentar el resultado en pantalla

Caso Contrario

Mostrar mensaje “no ha terminado el curso”

Caso Contrario

Mostrar mensaje “no existe cursos en el rango de fechas indicadas”

Fin de control

3.2.7.4 Salidas

Pantalla de resultados o mensajes según el caso.

3.3 Requerimientos de rendimiento

La aplicación Web estará disponible en línea, lo que significa que deberá permitir el acceso concurrente de los usuarios. Entonces se calcula que podrá soportar por lo menos 10 accesos simultáneos a un curso desde diferentes terminales, y con ello lograr un funcionamiento eficiente.

Se requiere un enlace dedicado simétrico de por lo menos 64 K en las instalaciones donde se instalará el servidor de aplicaciones. Actualmente ya se dispone de un enlace de 128 K.

El tiempo de respuesta en abrir una ventana para los usuarios, no debe superar los 30 segundos, y se recomienda que los usuarios tengan también un enlace dedicado para que la navegación no sea muy lenta.

3.4 Requerimientos de la base de datos lógica

El requerimiento más importante, es que para actualizar, registrar cualquier información, se debe primeramente verificar los datos del usuario. Es decir, que se manejará restricciones de seguridad.

Los datos de los cursos, becarios, exámenes, notas, manuales, no podrán ser accedidos ni por los becarios del INDESIC, si no solamente por el Administrador o Planificador, quienes son los que tienen autorización para hacer operaciones de modificación sobre esos datos.

La base de datos será accedida por un JDBC, E es decir que no se conectará directamente a la base de datos desde una ventana del usuario. Cuando el usuario haga alguna operación desde el broker de Internet, se llamará al JDBC, que desde el servidor de aplicaciones TOMCAT se conectara a la base de datos.

3.5 Atributos del software

El software debe ser desarrollado de modo que funcione en los browser más comunes de Internet, lo que significa que podrá ser accedido sin ajustarse a un sistema operativo en especial.

Las pantallas están diseñadas en base a frames o marcos, lo que permite un rápido acceso hacia un servicio, por lo que el navegador que se utilice debe soportar frames.

Toda la información que se requiera guardar será enviada a la base de datos del servidor.

3.5.1 Fiabilidad

La aplicación web, permitirá acceder a la totalidad de los servicios en línea, y los datos que se envíen serán procesados con el 100% de seguridad, a menos que exista alguna falla en el enlace de usuario remoto o local que impidan realizar la transacción respectiva.

3.5.2 Disponibilidad

Los cursos de capacitación estarán disponibles el 99% del tiempo en el Internet, solamente en casos extremos se cortará la comunicación, como por ejemplo en un terremoto, incendio, etc.

3.5.3 Seguridad

La aplicación web estará expuesta en el Internet para los becarios del INDESIC y ellos podrán acceder usando la debida identificación, sin embargo, existe la posibilidad de que los datos que el usuario envíe puedan ser interceptados por espías de la red.

Para garantizar la seguridad de esa información, se usará encriptación de los datos, esto se realizará usando el protocolo SSL, que es uno de los que brindan seguridad en las transacciones comerciales de Internet.

3.5.4 *Mantenibilidad*

La aplicación se ha desarrollado siguiendo la norma IEEE 830 y con una metodología que posibilita que se pueda realizar el mantenimiento que sea necesario.

3.5.5 *Portabilidad*

El sistema es completamente portable, ya que se ha diseñado usando herramientas open source y con herramientas para la web lo que garantizan que funcionará en varias plataformas.

4. Información de Soporte

4.1 *Elección de Ciclo de Vida*

El ciclo de vida que se ha elegido para este sistema es el espiral, que para el tiempo en se ha planteado este proyecto de tesis, es el mas adecuado para crear aplicaciones para Internet, ya que parte desde algo básico y se va construyendo el software de acuerdo a las necesidades del cliente.

4.2 *Diagrama de Contexto*

El diagrama de contexto permite identificar las entidades y los flujos de datos que intervienen en el sistema.

En primera instancia se ha diseñado el diagrama de contexto de la situación actual, en el que se muestra los procesos actuales para desarrollar los cursos de capacitación del INDESIC. Ver el Anexo C: Diagramas de Diseño (C.6: Diagrama de Contexto – Situación Actual).

En segundo lugar se ha diseñado el diagrama de contexto adecuado para la aplicación web de cursos de capacitación del INDESIC, este diagrama lo podrá observar el *Anexo C: Diagramas de Diseño (C.7: Diagramas de Contexto Propuesto)*.

4.3 Diagrama de Flujo de Datos

Los diagramas de flujo de datos son el resultado de la descomposición del diagrama de contexto, y permiten obtener un mayor detalle del proceso que se realiza.

Los Diagramas de Flujo de Datos se encuentran adjuntos en el *Anexo C: Diagramas de Diseño (C.8: Diagramas de Flujo de Datos)*.

4.4 Modelo Lógico de Datos

El modelo Lógico de Datos se lo representa mediante el Modelo Entidad Relación que se realiza para representar lógicamente las tablas, los campos y las relaciones que existe entre cada una de las tablas. El Modelo se encuentra adjunto en el *Anexo C: Diagramas de Diseño (C.9: Diagrama Lógico de Datos)*.

4.5 Modelo Físico de Datos

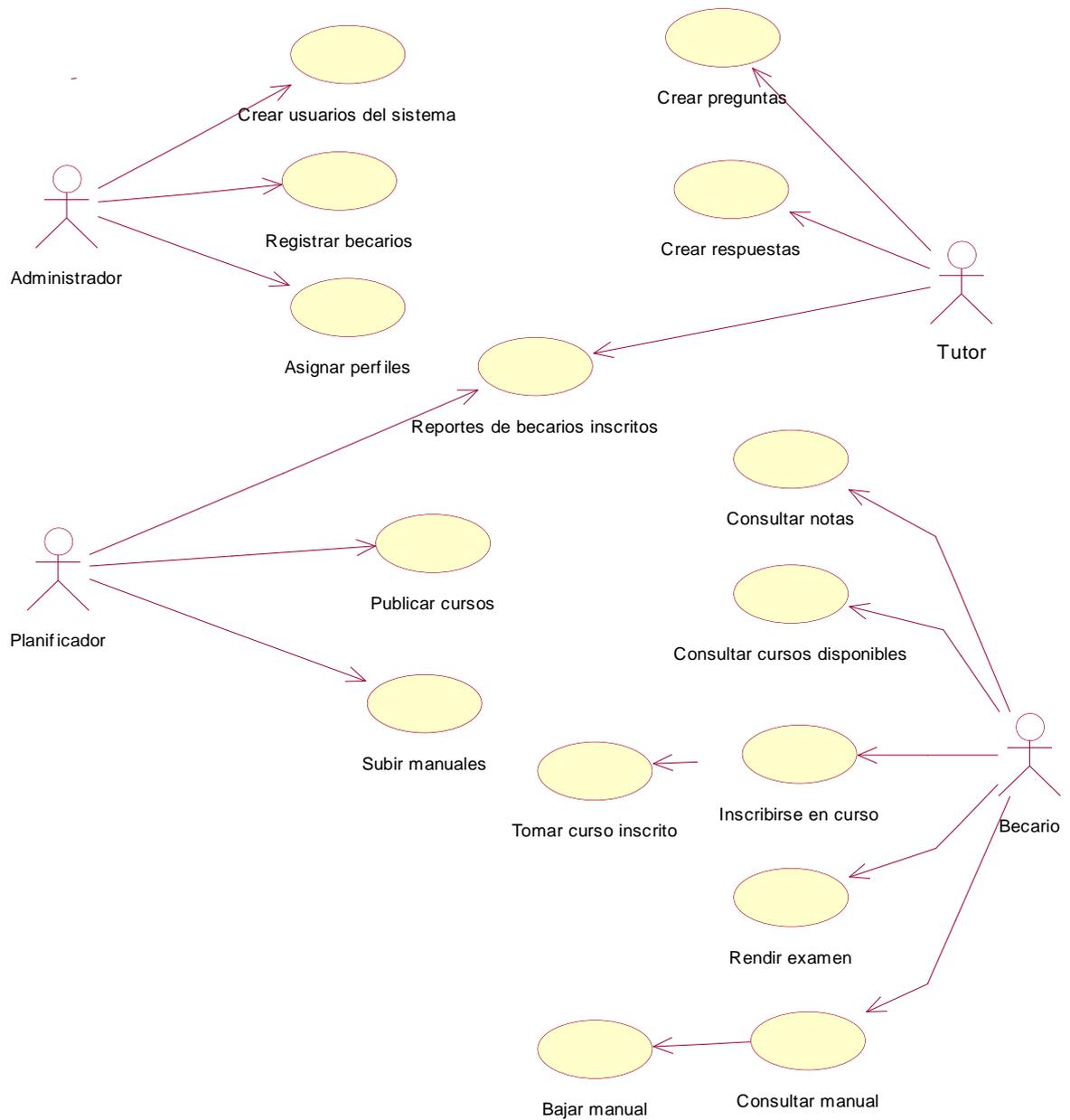
El modelo Físico de Datos se obtiene del Modelo Lógico, luego de que se haya verificado que este último no tenga errores.

El modelo Físico representa ya la base de datos, ya que a partir de este se genera un script de las tablas o se genera directamente la base de datos desde la herramienta case en caso de existir esta opción, como es el caso de Power Designer por ejemplo.

El diseño del modelo físico de datos se encuentra en el *Anexo D: Modelo Físico de Datos*.

ANEXO C: DIAGRAMAS DE DISEÑO

C.1 DIAGRAMA DE CASOS DE USO



DESCRIPCIÓN DE CASOS DE USO Y ACTORES

Descripción de Actores



Administrador: Encargado del proceso administrativo del sistema, en el que realizará dos cosas específicas que son, crear usuarios y crear becarios.



Planificador: Usuario con permisos para publicar cursos, subir manuales y obtener reportes de los becarios inscritos y el estado en que se encuentran.

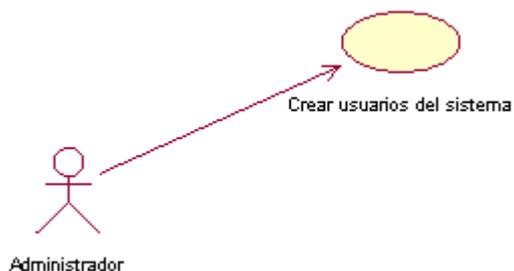


Becario: Son usuarios con permisos para hacer uso de los cursos en línea. Estos usuarios pueden realizar los siguientes procesos: inscribirse en un curso, consultar cursos disponibles, acceder a un curso, rendir examen, bajar manuales, y consultar notas.



Tutor: Es un usuario que tiene permisos para crear preguntas por cada curso, crear respuestas por cada pregunta y también puede obtener reporte de los becarios inscritos.

Descripción de los Casos de Uso



Caso de Uso: 1	Crear usuarios del sistema
Actor:	Administrador
Descripción:	El Administrador creará un usuario y le asignará un perfil. El perfil puede ser de Administrador, Planificador, Becario o Tutor.

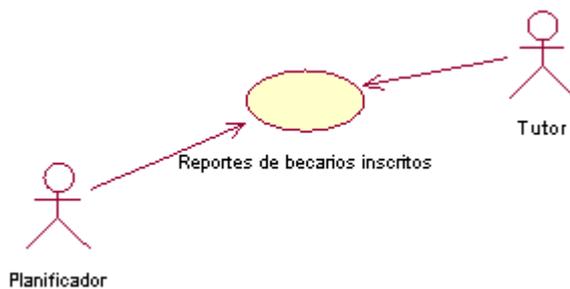
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R1.1: La interfaz del usuario debe estar compuesta por cuadros de texto para el ingreso de los datos del usuario, como son: login, password, apellidos, nombres y email.</p> <p>R1.2: Para el perfil y el estado, tenemos dos cuadros de selección, de los cuales se debe escoger el valor que se asignará al usuario.</p> <p>R1.3: Los datos de los cuadros de texto y de los combos se guardarán en la base de datos que corresponda a los usuarios.</p> <p>R1.4: El administrador tendrá la posibilidad de modificar los datos del usuario, incluyendo la clave en caso de que sea necesario.</p> <p>R1.5: El login y el password serán necesarios para ingresar al sistema y de acuerdo al perfil se mostrarán las opciones que pueda realizar.</p>	



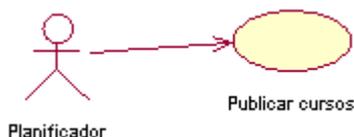
Caso de Uso: 2	Registrar becarios
Actor:	Administrador
Descripción:	El Administrador creará (registrará) un becario y el perfil se asignará automáticamente como becario.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R2.1: La interfaz para registrar becarios, debe contener cuadros de texto para el ingreso de los datos y también debe disponer de tres combos; para seleccionar el sexo, estado civil y estado del becario.</p> <p>R2.2: Los datos ingresados se guardarán en la tabla de becarios y el perfil del usuario será asignado automáticamente como becario.</p>	

R2.3: El planificador puede actualizar los datos del becario en caso de ser necesario.

R2.4: Cuando el becario ingrese con su identificación, se le mostrará solamente los módulos del becario.

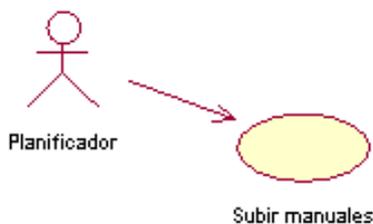


Caso de Uso: 3	Reporte de becarios inscritos
Actor:	Planificador y Tutor
Descripción:	Los actores indicados tendrán la posibilidad de obtener un reporte de los becarios inscritos y del estado en que se encuentran, con el fin de tomar decisiones en la publicación de los cursos.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R3.1: Este reporte se obtendrá de la inscripción. En caso de que no haya becarios inscritos, simplemente no mostrará ninguna información.</p> <p>R3.2: Cada campo recuperado se mostrará en columnas y cada registro existente se mostrará en filas.</p> <p>R3.3: El reporte dispondrá de un link que permita imprimir el reporte obtenido.</p>	



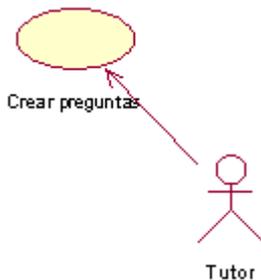
Caso de Uso: 4	Publicar cursos
Actor:	Planificador
Descripción:	El Planificador deberá crear los cursos que se impartirán por Internet y también deberá escoger el curso que debe estar en formato html.
Prioridad:	Necesario

Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R4.1: Antes de publicar un curso, el curso debe ser diseñado con anterioridad en formato html.</p> <p>R4.2: El curso puede ser diseñado con las herramientas que sea necesario, como puede ser Macromedia o Frontpage, lo importante es que sea compatible para cualquier navegador web.</p> <p>R4.3: Para publicar un curso, debe escoger el curso desde una unidad de disco o carpeta del ordenador, luego ingresar los datos restantes como el nombre del curso, el pre-requisito, el estado y si es necesario, alguna observación.</p> <p>R4.4: El Pre-requisito y el estado deben estar en un combo, de modo que se pueda seleccionar una opción.</p> <p>R4.5: El Pre-requisito debe ser obtenido de la misma tabla de cursos, de modo que cada curso que se haya creado aparecerá luego como una opción de pre-requisito para otro curso que se esté creando.</p>	

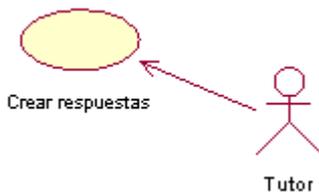


Caso de Uso: 5	Subir manuales
Actor:	Planificador
Descripción:	El Planificador deberá subir los manuales al sitio, de modo que los becarios que lo requieran lo puedan bajar. Estos manuales pueden estar en cualquier formato.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R5.1: Antes de subir los manuales al sitio, debe ser creado y desarrollado usando cualquier herramienta, como puede ser Word, Excel, Power Point, etc.</p> <p>R5.2: En caso de que un manual conste de varios archivos, y se quiera subir todos los archivos como uno solo, se debe comprimir el archivo.</p>	

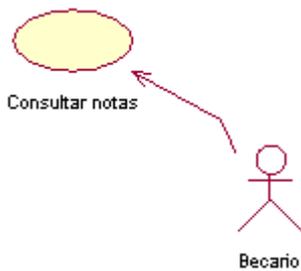
R5.3: Cuando el usuario descargue un manual, se registrará automáticamente los datos del usuario y del manual descargado.



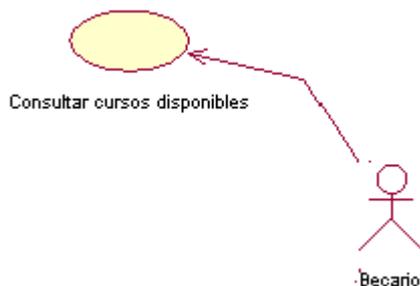
Caso de Uso: 6	Crear preguntas
Actor:	Tutor
Descripción:	El Tutor deberá crear las preguntas para cada curso existente, de modo que cuando el becario quiera rendir el examen de un curso, se pueda generar el formulario del examen.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R6.1: Todas las preguntas estarán relacionadas con un curso.</p> <p>R6.2: La interfaz del usuario tendrá dos elementos, un combo para seleccionar el curso al que queremos asignarle una pregunta y un cuadro de texto para ingresar la pregunta para el curso seleccionado.</p> <p>R6.3: Las claves de las preguntas se crearán automáticamente al guardar los datos.</p> <p>R6.4: Cada curso deberá disponer de al menos 4 preguntas como mínimo.</p> <p>R6.5: Cada vez que se crea una pregunta, deberá lanzar a la interfaz de respuestas ya que cada pregunta, también deberá disponer de varias respuestas.</p>	



Caso de Uso: 7	Crear respuestas
Actor:	Tutor
Descripción:	El tutor deberá crear las respuestas para cada pregunta que haya ingresado. Cada pregunta deberá tener al menos 3 opciones de respuesta, de las cuales una debe ser la correcta y las demás serán incorrectas para que el becario pueda elegir.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R7.1: Para crear una respuesta, primero es necesario crear una pregunta, ya que cada pregunta estará relacionada con varias respuestas. .</p> <p>R7.2: Para cada pregunta, solamente puede haber una respuesta correcta, pero si puede haber varias respuestas que sean incorrectas.</p> <p>R7.3: La interfaz del usuario en este caso de uso será un campo que indique la pregunta a la que se está asignando la respuesta, un campo de texto en el que ingresaremos la respuesta para la pregunta y un combo que permitirá seleccionar si la respuesta es verdadera o falsa.</p> <p>R7.4: El código para la respuesta se creará automáticamente, cada vez que se envíe a guardar los datos de la pregunta.</p> <p>R7.5: Para cada pregunta se deberá ingresar al menos tres respuestas, de las cuales una debe ser registrada como verdadera.</p> <p>R7.6: Luego de guardar una respuesta, debe volver al formulario para crear respuestas, con la descripción de la pregunta, de modo que se pueda ingresar las tres respuestas para cada pregunta.</p> <p>R7.7: Cuando el usuario haya terminado de crear las respuestas para la pregunta, debe hacer clic en Salir y esto hará que regrese al formulario para crear preguntas.</p>	



Caso de Uso: 8	Consultar notas
Actor:	Becario
Descripción:	El Becario puede consultar la nota obtenida en un examen de un curso.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R8.1: El becario tendrá la posibilidad de consultar las notas obtenidas en los exámenes que haya rendido.</p> <p>R8.2: Desplegará un listado de los cursos y la respectiva nota obtenida.</p> <p>R8.3: Si no ha rendido ningún examen, no aparecerá ninguna información al respecto.</p> <p>R8.4: Se podrá consultar únicamente las notas del usuario con el que ingresó al sistema, no se podrá consultar las notas de otros usuarios.</p>	

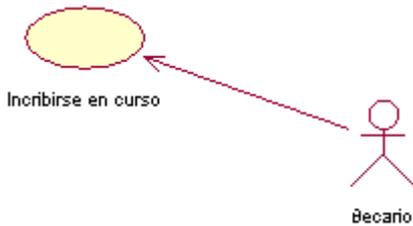


Caso de Uso: 9	Consultar cursos disponibles
Actor:	Becario
Descripción:	El Becario tendrá la posibilidad de consultar los cursos disponibles y sus respectivos pre-requisitos.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
R9.1: Todos los becarios sin excepción pueden verificar los cursos disponibles	

con sus respectivos pre-requisitos, si es que los tiene.

R9.2: Para mostrar la consulta, cada campo recuperado de la tabla, se pondrá en columnas y los registros se imprimirán en filas.

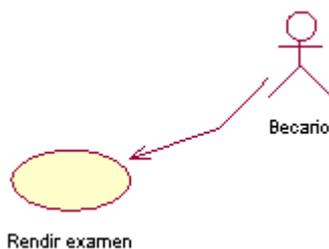
R9.3: Esta consulta se mostrará con la finalidad de que los becarios conozcan los cursos disponibles y no podrán hacer nada más.



Caso de Uso: 10	Inscribirse en curso
Actor:	Becario
Descripción:	El Becario deberá inscribirse en un curso antes de poder acceder a uno de ellos.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R10.1: La interfaz para la inscripción esta compuesta de un combo, que permite seleccionar el curso en el que se quiere inscribir.</p> <p>R10.2: Antes de mostrar los cursos en los que el usuario se pueda inscribir, se hace una validación de los pre-requisitos de los cursos aprobados, reprobados, y pendientes.</p> <p>R10.3: Para validar pre-requisitos, cada curso tiene un campo que indica el código del curso pre-requisito, si no existe pre-requisito ese campo tendrá un valor nulo.</p> <p>R10.4: Para verificar si el curso fue aprobado, reprobado o está pendiente, se lo hace en la tabla de inscripciones, donde se dispone de un campo que tendrá: 0 para pendiente (que no ha dado ningún examen), 1 para aprobado y 2 para reprobado.</p> <p>Hacemos un filtrado con las observaciones indicadas y esto por cada usuario y finalmente ese resultado mostramos en el combo.</p>	



Caso de Uso: 11	Tomar curso inscrito
Actor:	Becario
Descripción:	El becario puede acceder al curso (tutorial) en línea según la inscripción.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R11.1: Esta interfaz mostrará una lista de los cursos en los que un becario se ha inscrito con el nombre del curso y un link para poder ingresar al curso.</p> <p>R11.2: En caso de que no se haya inscrito en ningún curso, simplemente no podrá ingresar a ninguno. Por lo tanto, primero hay que inscribirse en un curso</p> <p>R11.3: El curso debe aparecer en el navegador de internet, ya que debe estar diseñado en formato html, lo cual es compatible para que se abra en el mismo browser.</p> <p>R11.4: Podrá ingresar al curso las veces que el becario lo requiera, hasta que se sienta listo para rendir el examen.</p>	



Caso de Uso: 12	Rendir examen
Actor:	Becario
Descripción:	El Becario deberá rendir el examen cuando haya terminado el tutorial y esté listo para ello.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R12.1: El formulario para el examen estará diseñado para un máximo de 4 preguntas y cada pregunta deberá disponer de tres opciones de respuesta.</p>	

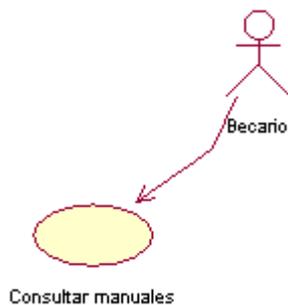
R12.2: De las opciones de respuestas presentadas, una deberá ser la verdadera y las otras dos serán falsas.

R12.3: En este formulario no se ha contemplado la posibilidad de presentar las preguntas y respuestas aleatoriamente, mas bien se ha definido en donde aparecerá la respuesta verdadera y en donde aparecerán las respuestas falsas.

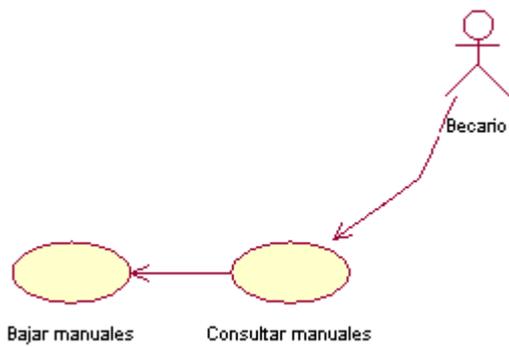
R12.4: La validación del examen se realizará internamente y de acuerdo a las respuestas seleccionadas se asentará la nota.

R12.5: Para este caso se ha creado el formulario utilizando cuadros de selección, sin embargo, el becario debe estar consciente que no debe seleccionar más de una respuesta por cada pregunta.

R12.6: La nota obtenida se guardará en la tabla BECARIO_NOTA, en el que debe registrarse también el nombre del usuario y el código del examen.



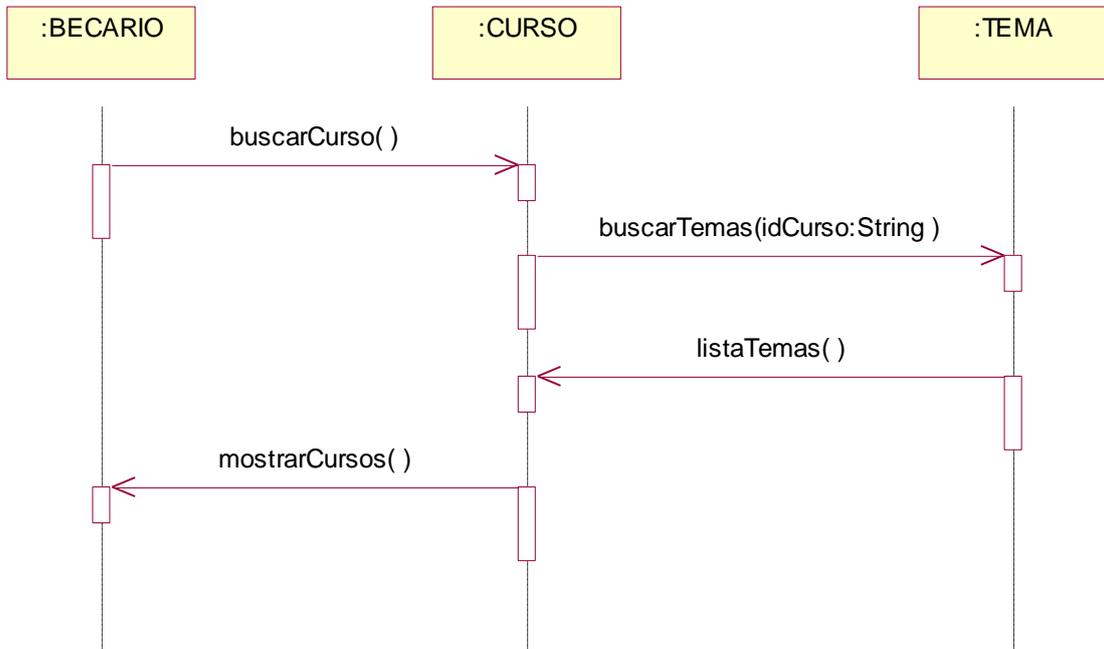
Caso de Uso: 13	Consultar manuales
Actor:	Becario
Descripción:	El Becario tendrá la posibilidad de consultar los manuales existentes.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
R13.1: Se mostrará una lista de los manuales existentes.	
R13.2: La lista de los manuales se mostrará a todos los becarios sin ninguna restricción.	
R13.3: Los becarios podrán descargar los manuales que requieran, no hace falta ningún requisito para esto, lo único que necesita el usuario es que sea becario.	
R13.4: El planificador deberá analizar los manuales que crea necesario publicar.	



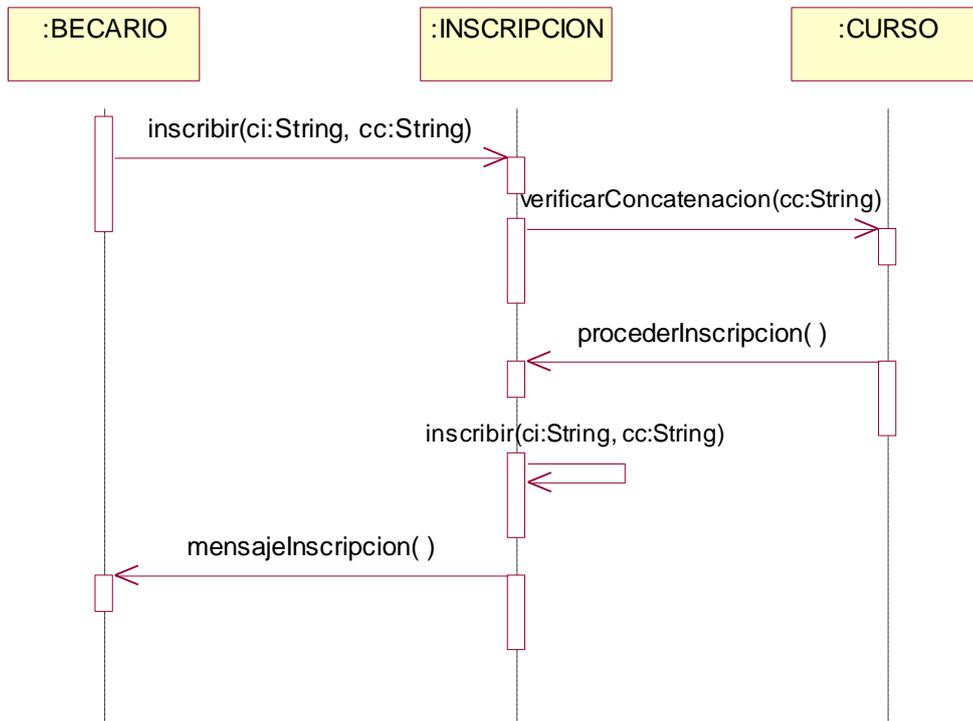
Caso de Uso: 14	Bajar manuales
Actor:	Becario
Descripción:	El usuario con perfil de becario podrá descargar los manuales que requiera.
Prioridad:	Necesario
Tiempo aprox. De desarrollo:	
REQUISITOS ASOCIADOS	
<p>R14.1: De la interfaz de consultas, el becario dispondrá de un link que le permitirá descargar el archivo que lo requiera.</p> <p>R14.2: Los manuales podrán ser descargados únicamente por los becarios y no por otros usuarios.</p>	

C.2 DIAGRAMAS DE SECUENCIA

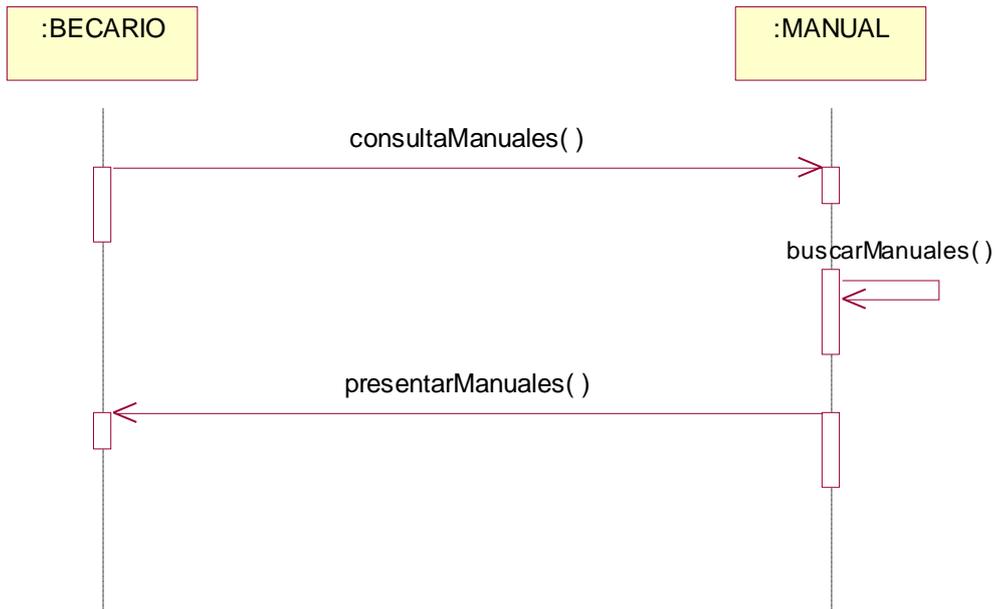
Consultar Cursos



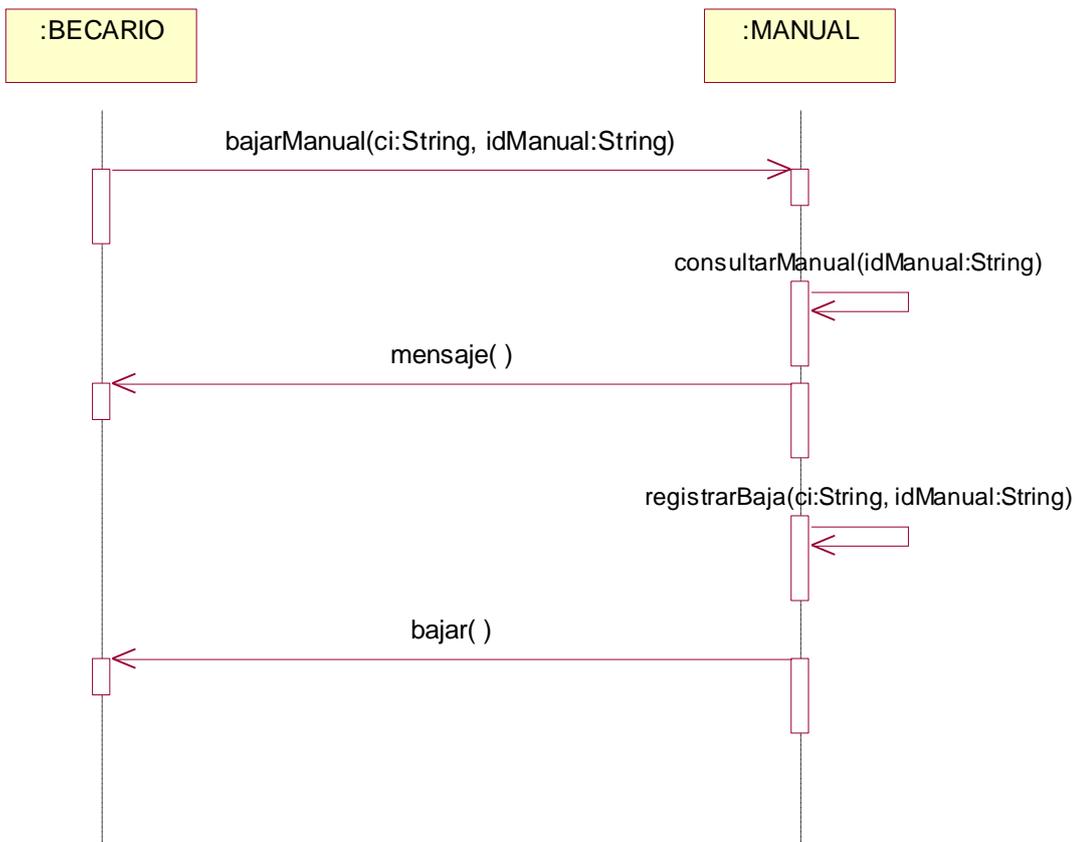
Inscribirse en Curso



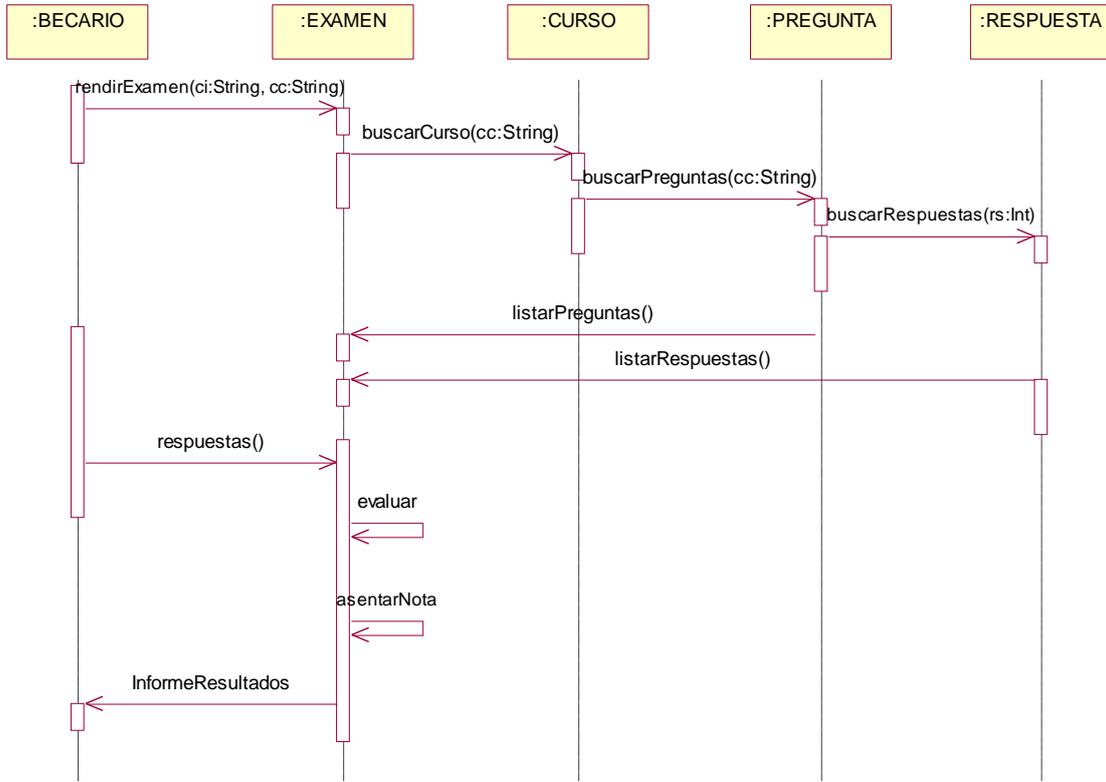
Consultar Manuales



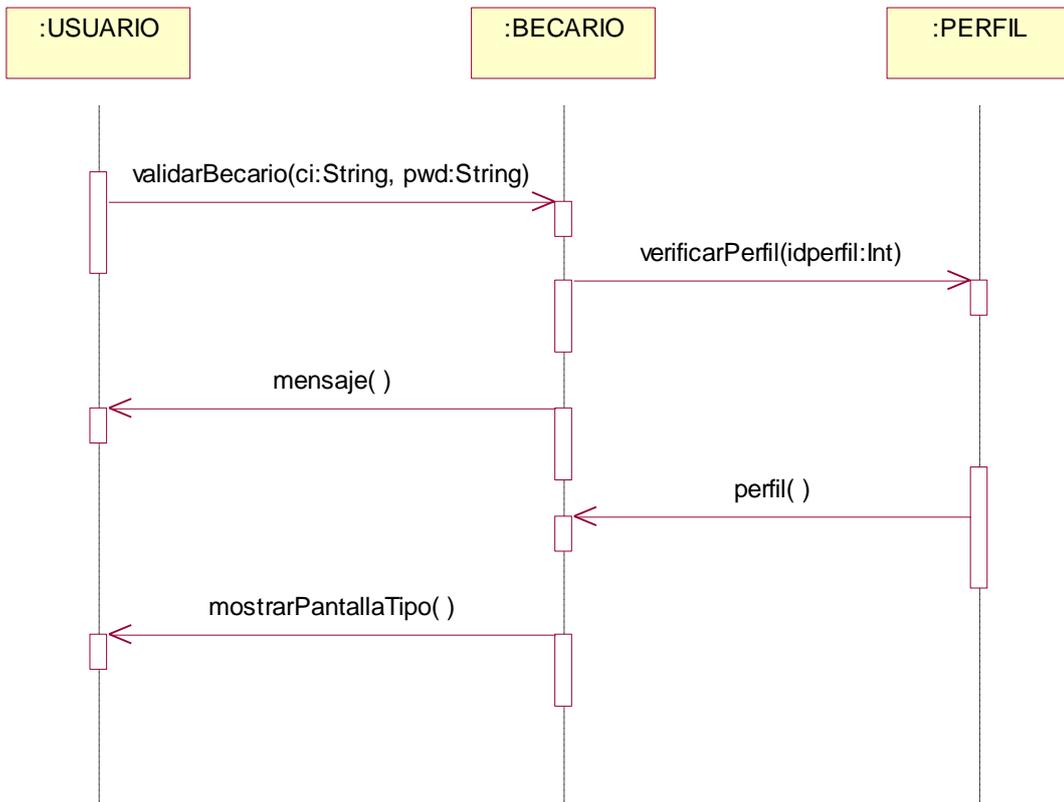
Bajar Manual



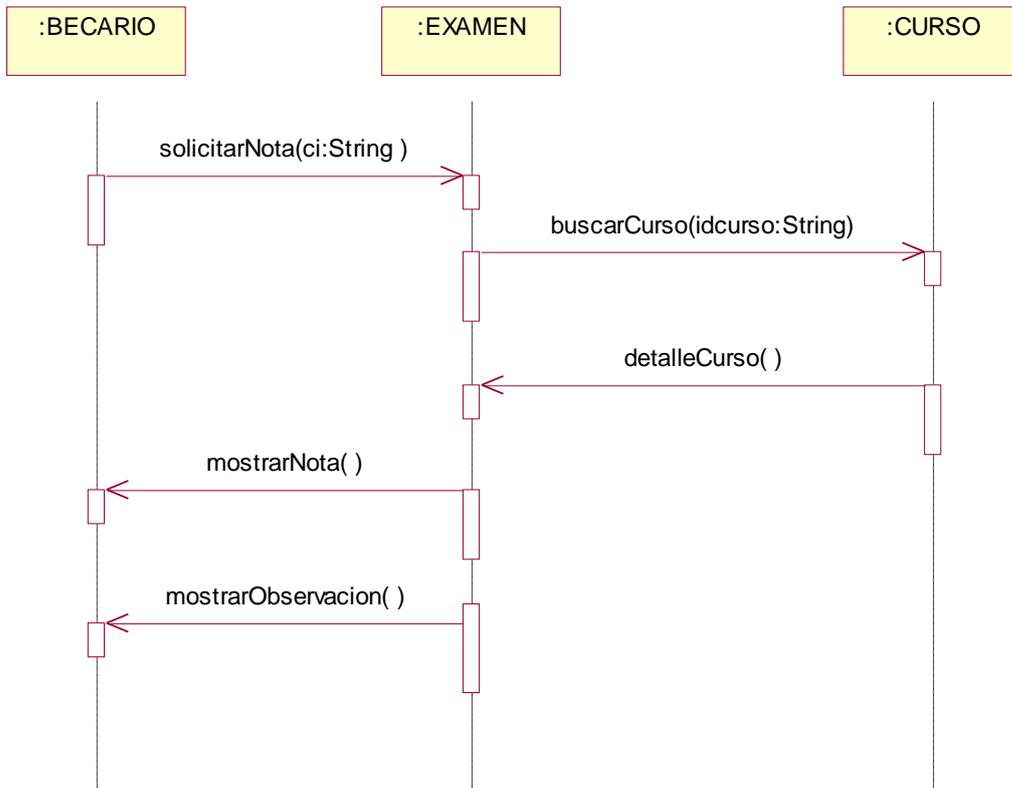
Rendir Examen y Registrar Nota



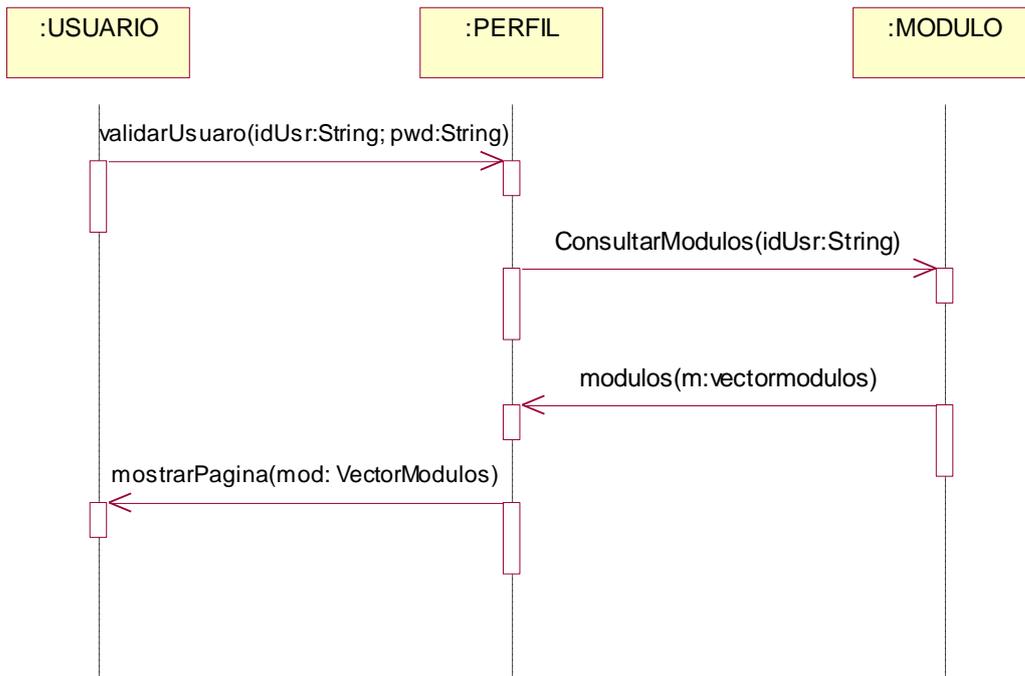
Validar Usuario y Clave



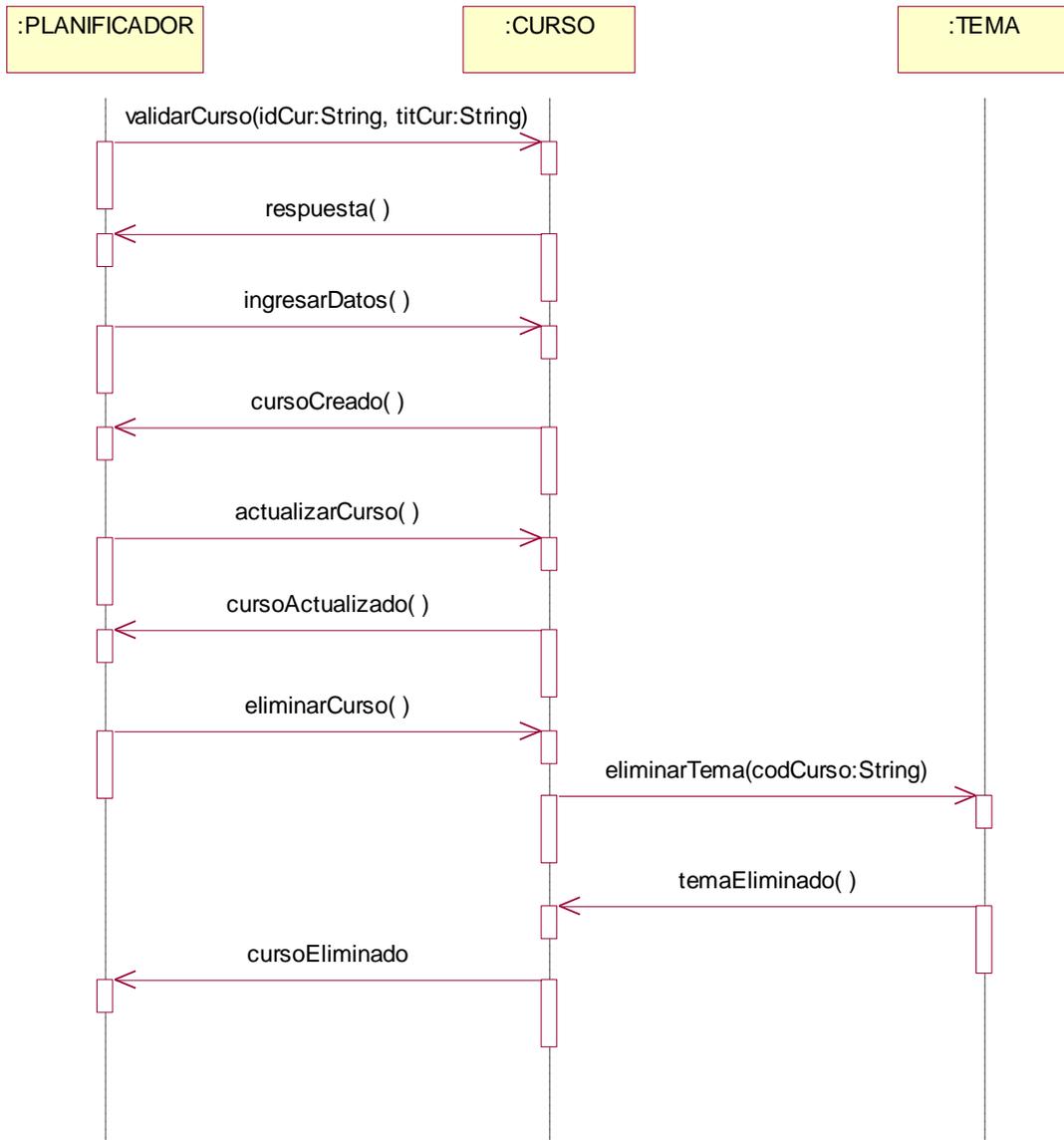
Consultar Resultados



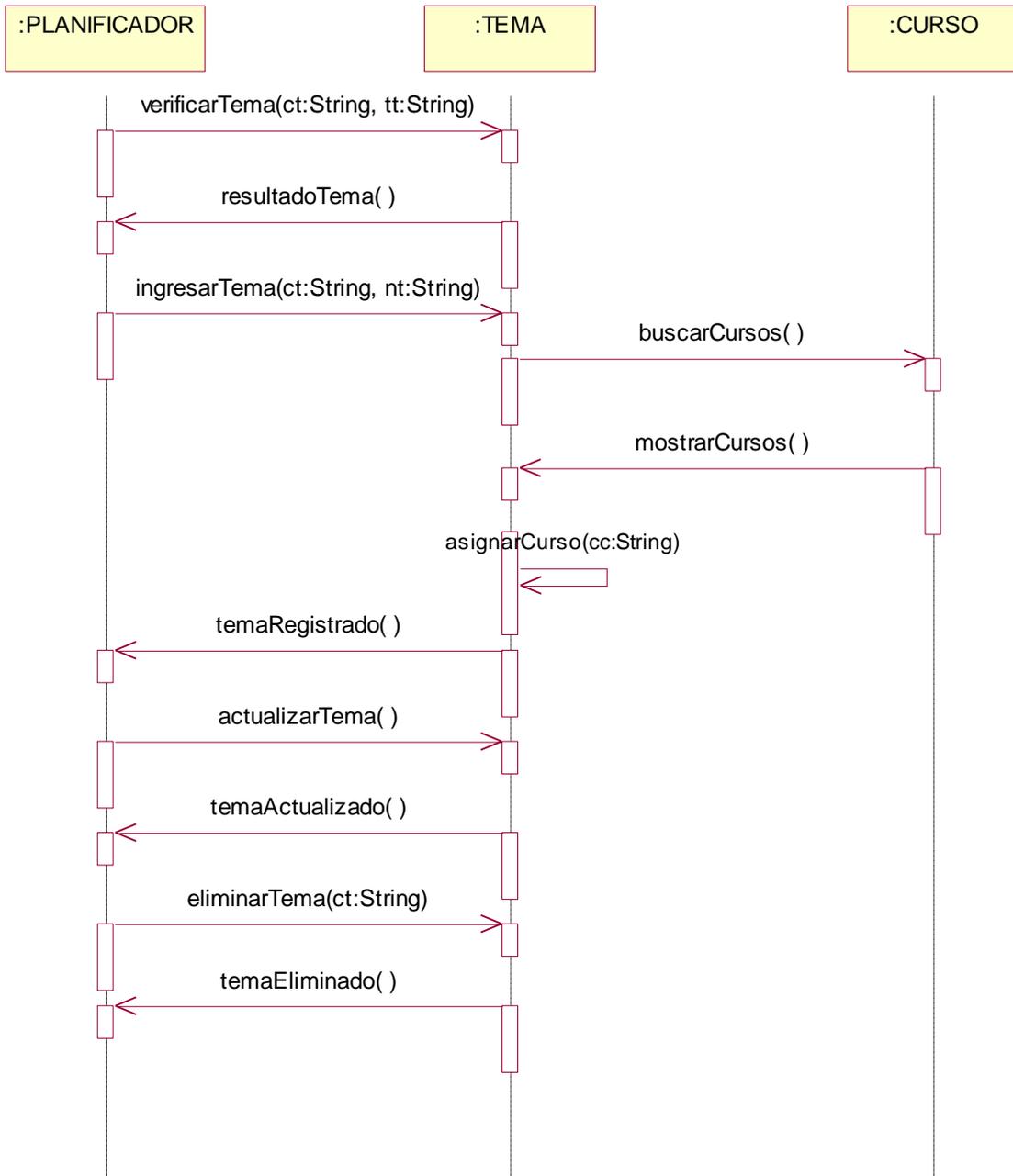
Mostrar Página Según Perfil



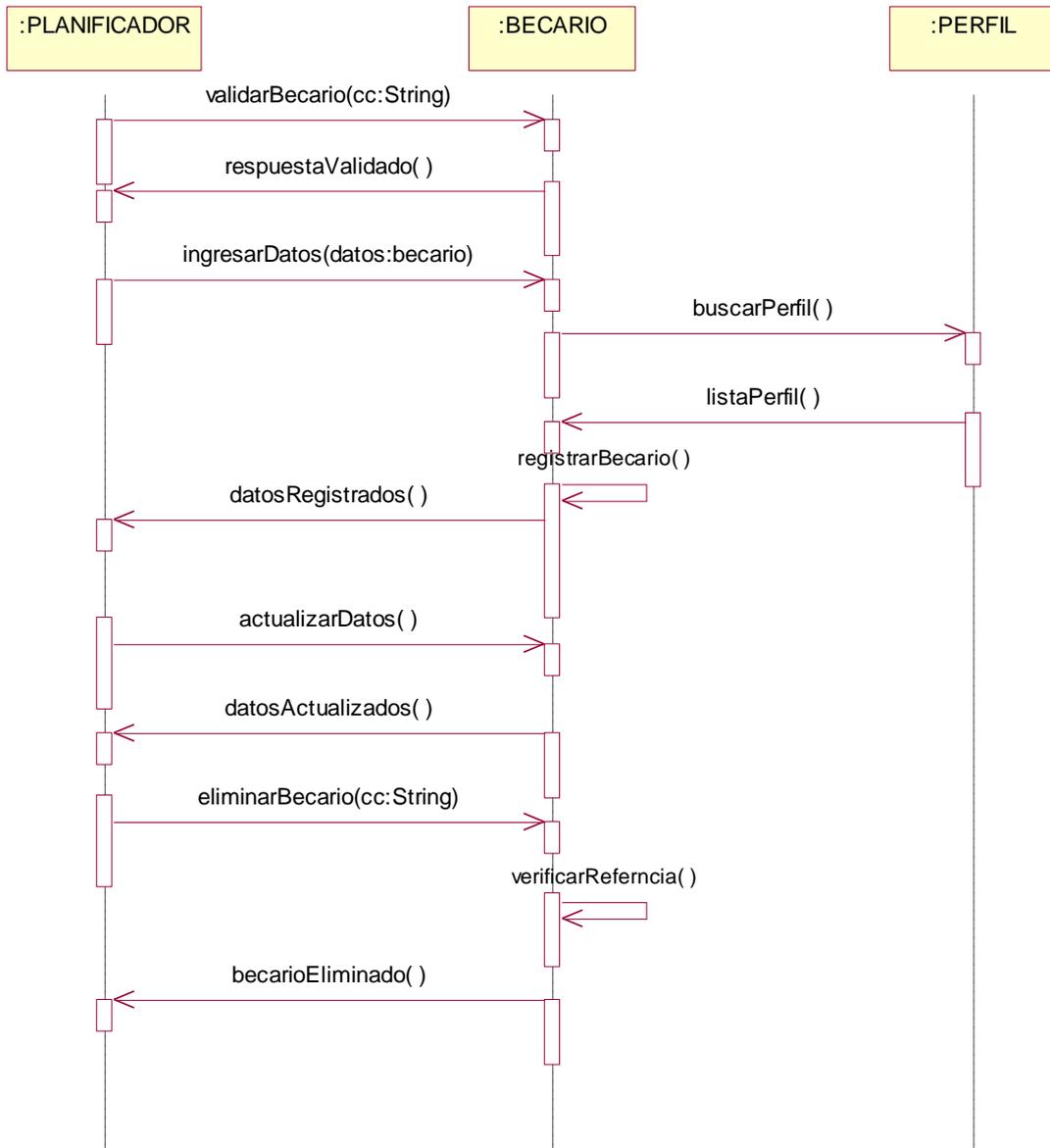
Mantenimiento de Cursos



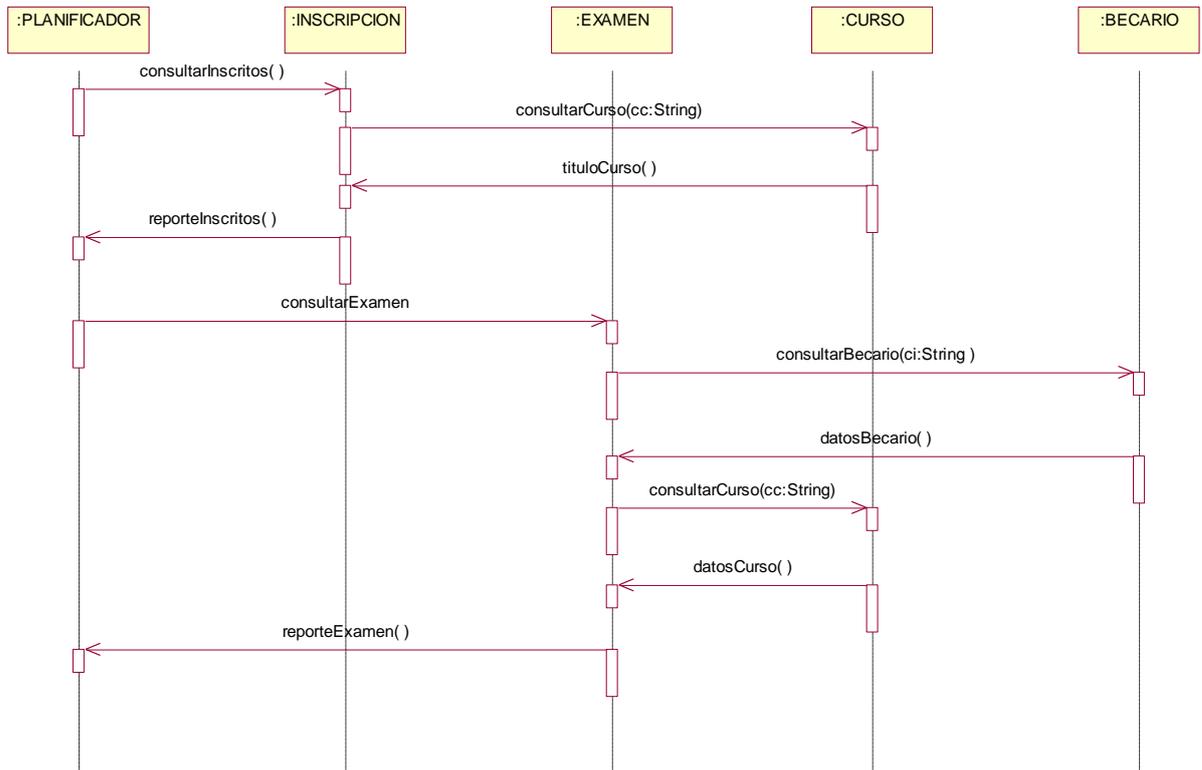
Mantenimiento de Preguntas



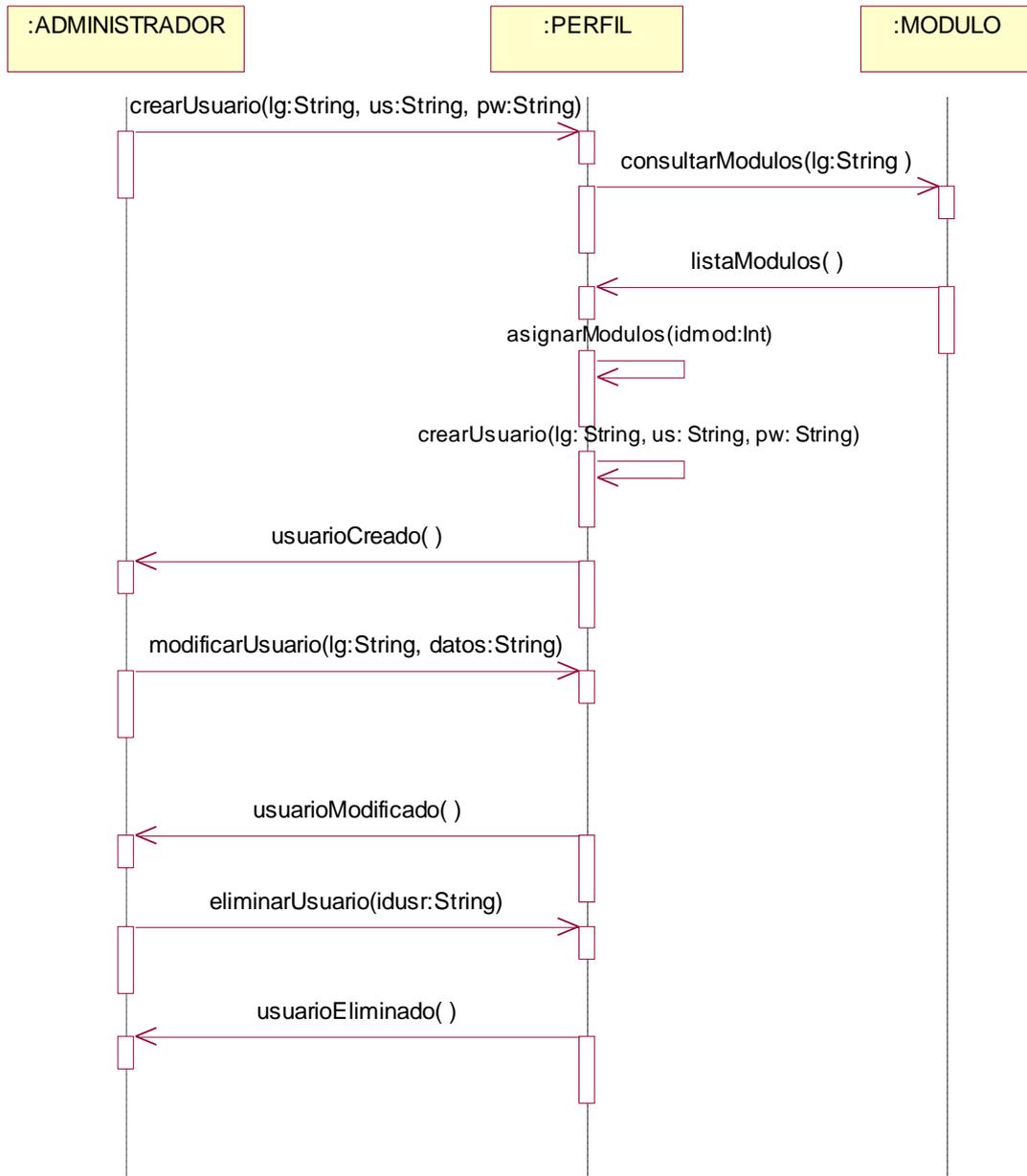
Mantenimiento de Becarios



Reportes

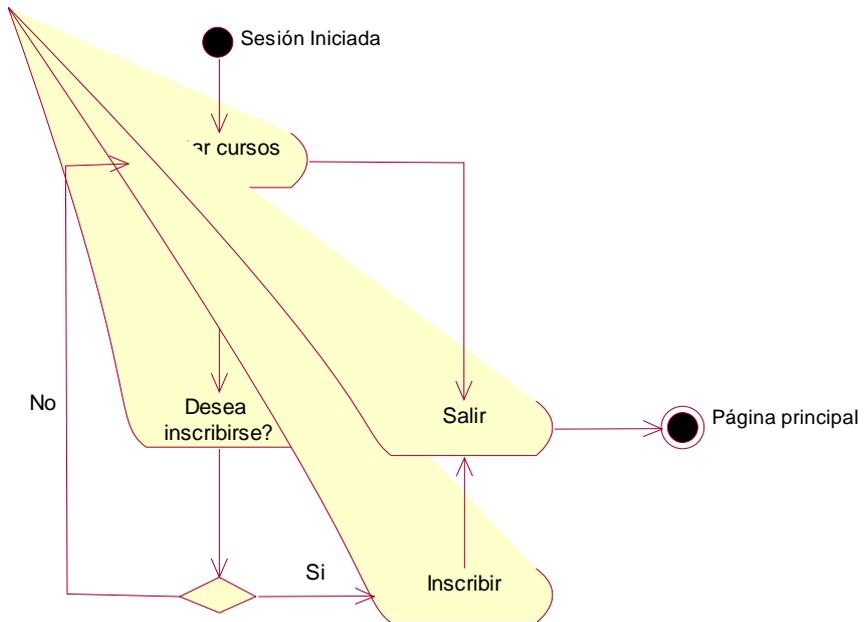


Mantenimiento de Usuarios

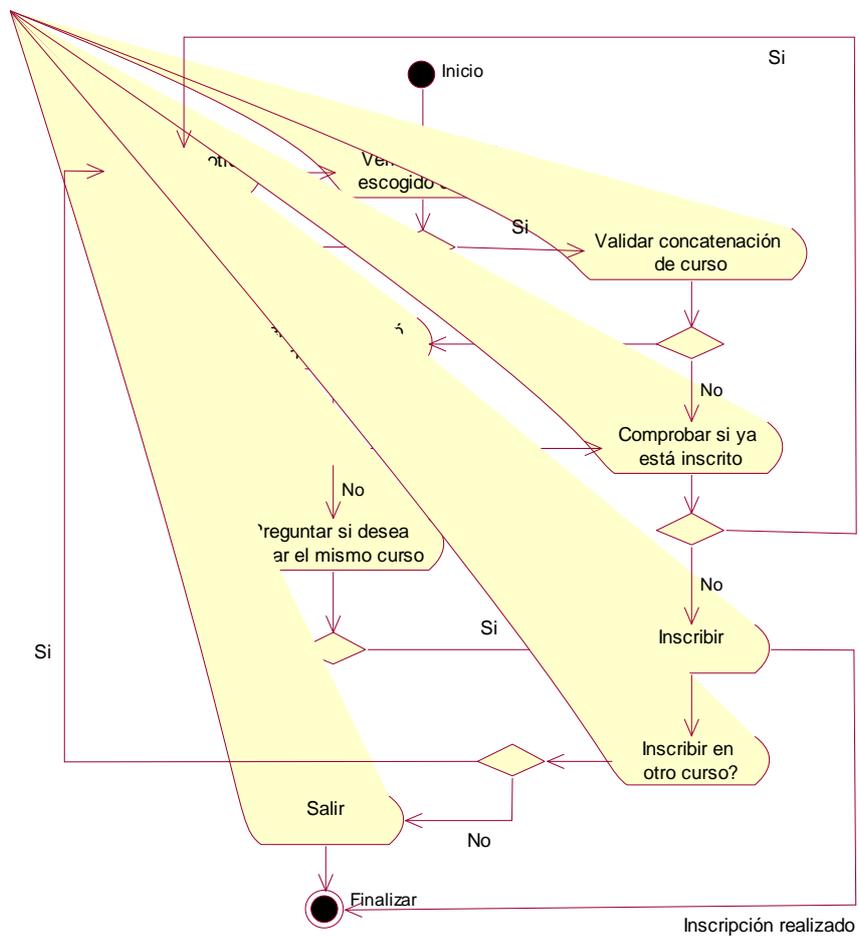


C.3 DIAGRAMAS DE ACTIVIDAD

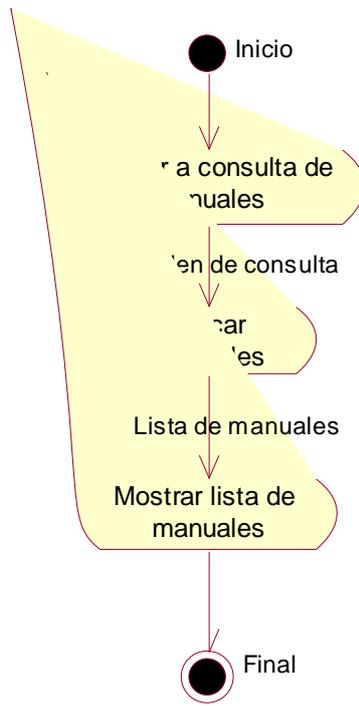
Consultar Curso



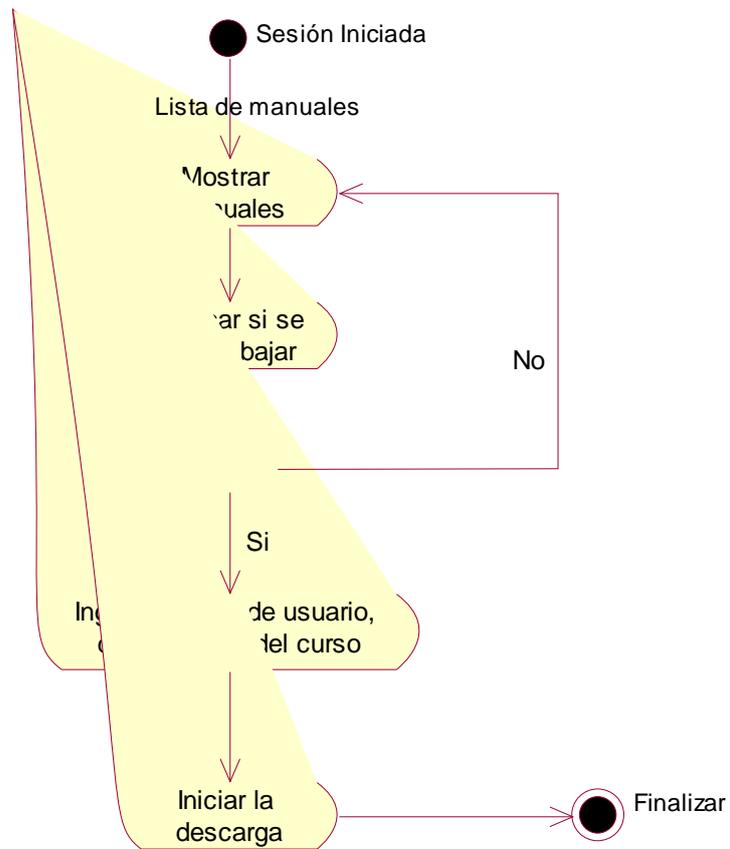
Inscribirse en Curso



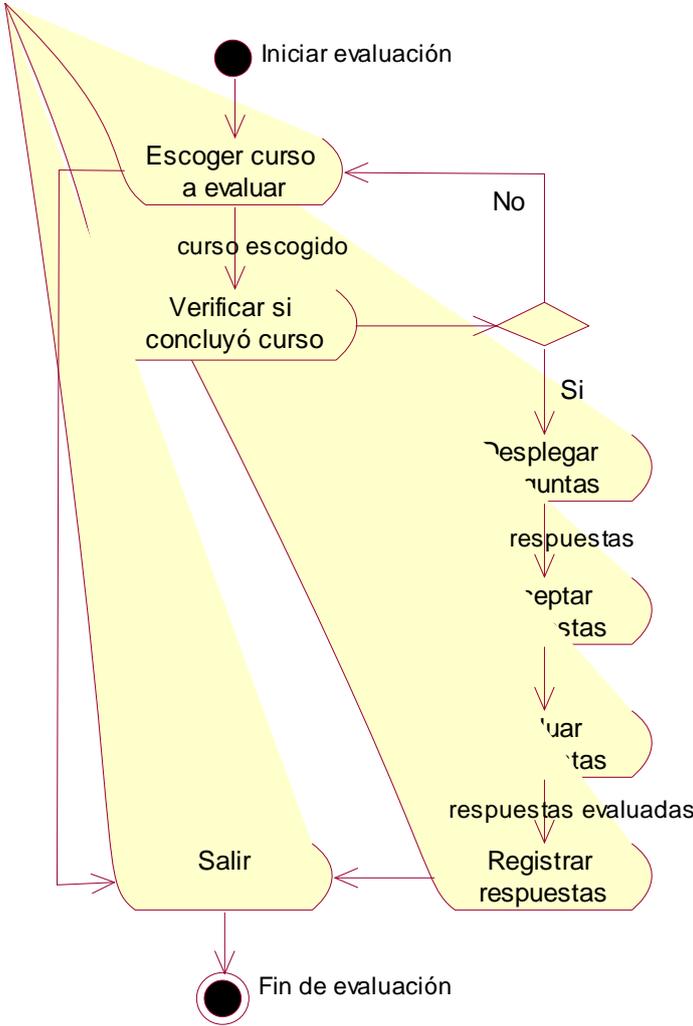
Consultar Manuales



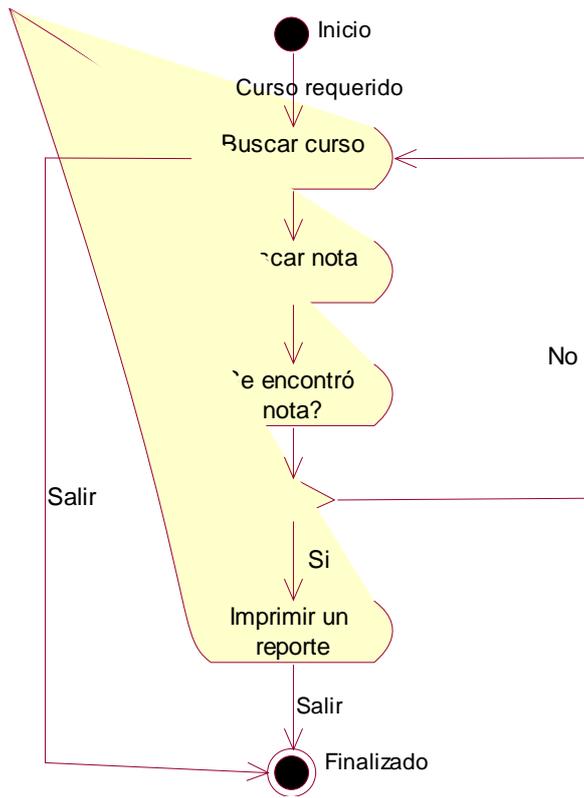
Descargar Manual



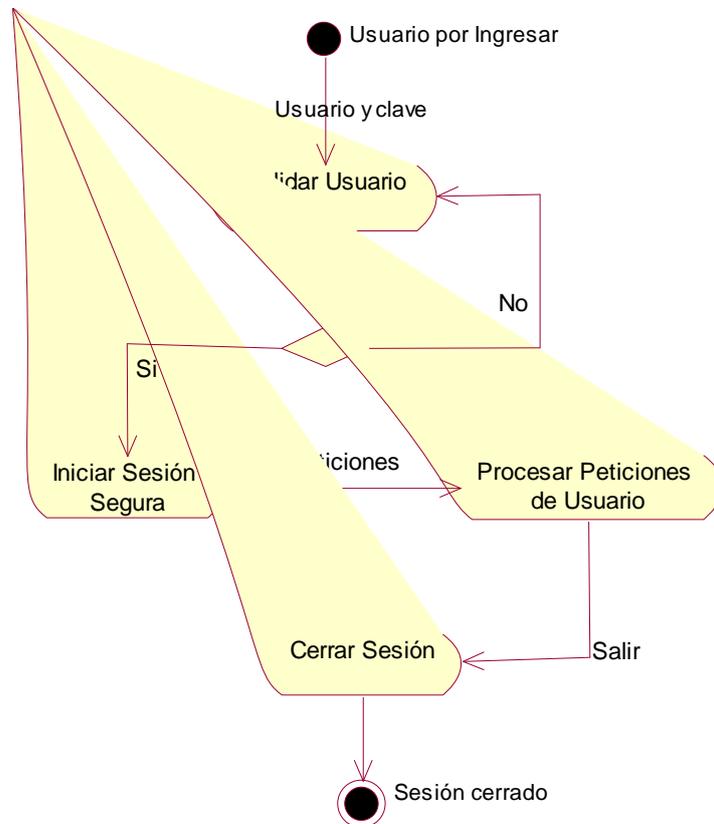
Rendir Examen y Registrar Notas



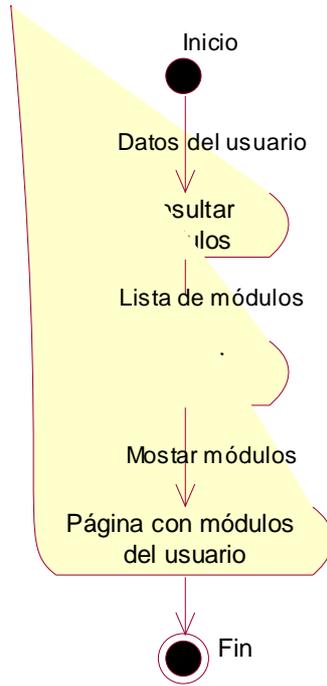
Consultar Resultados



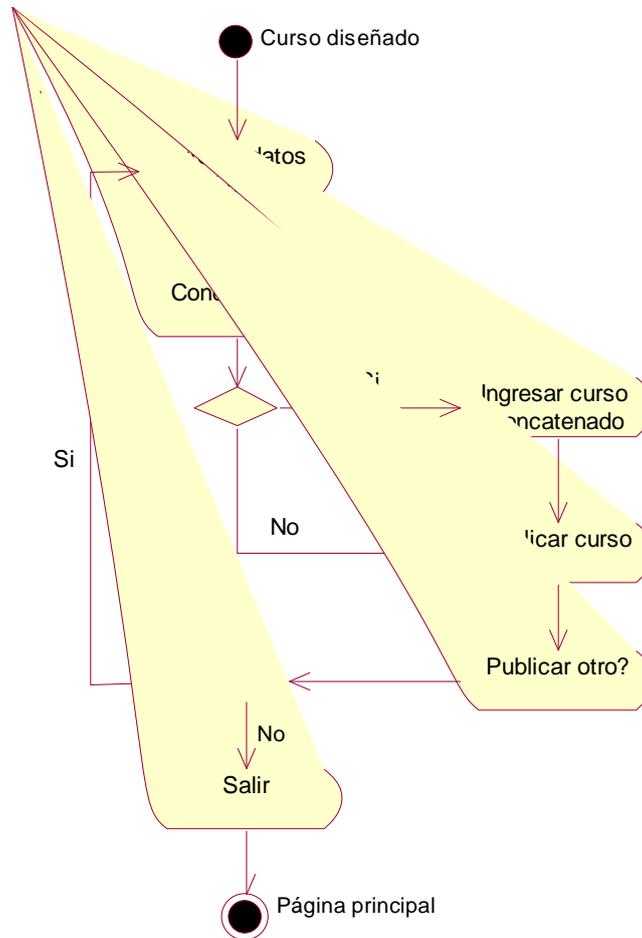
Validar Usuario



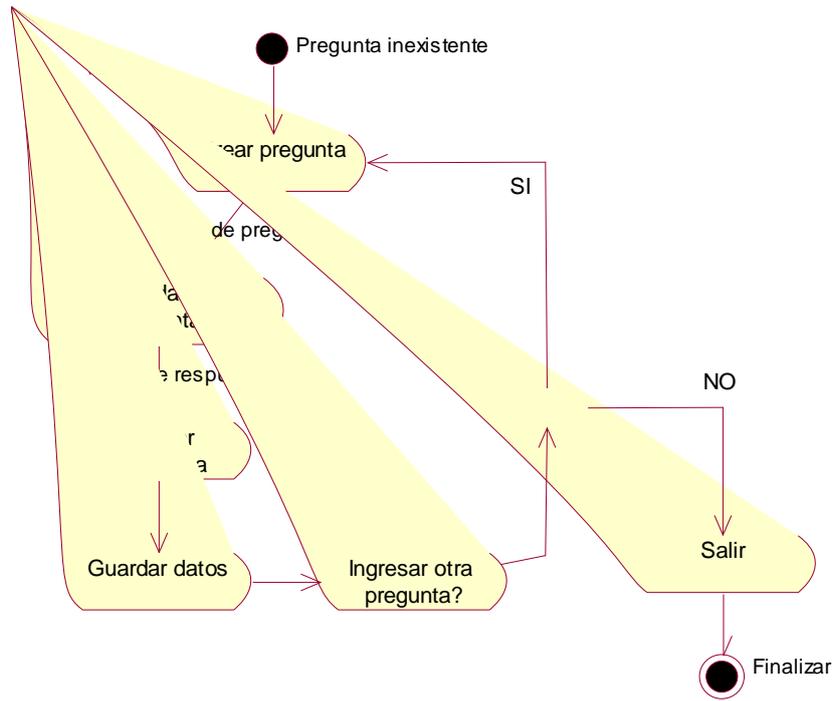
Mostrar Página Según Perfil del Usuario



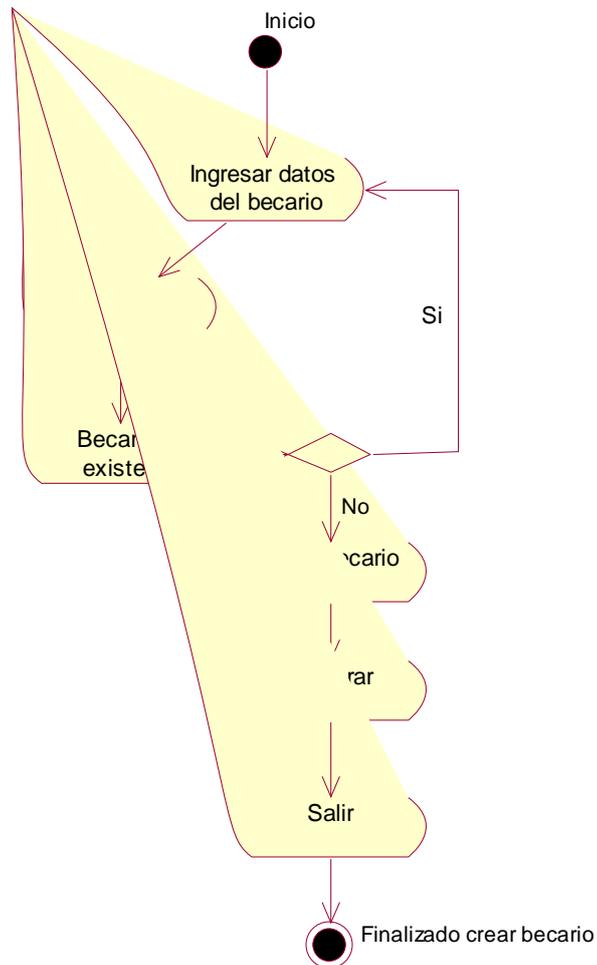
Mantenimiento de Cursos



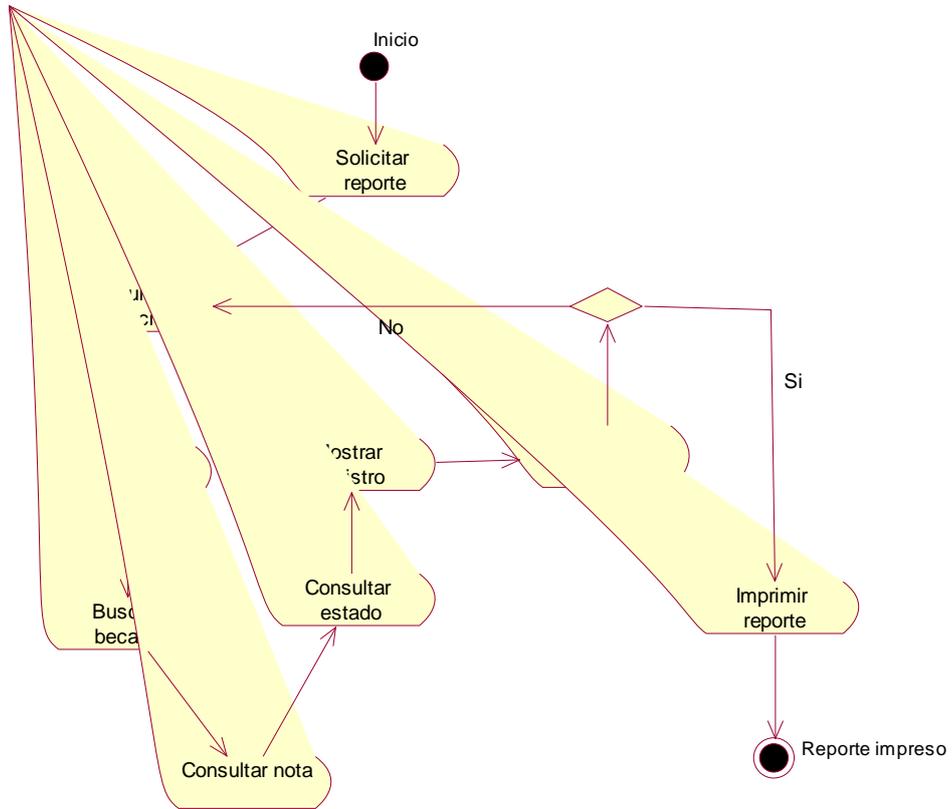
Mantenimiento de Preguntas



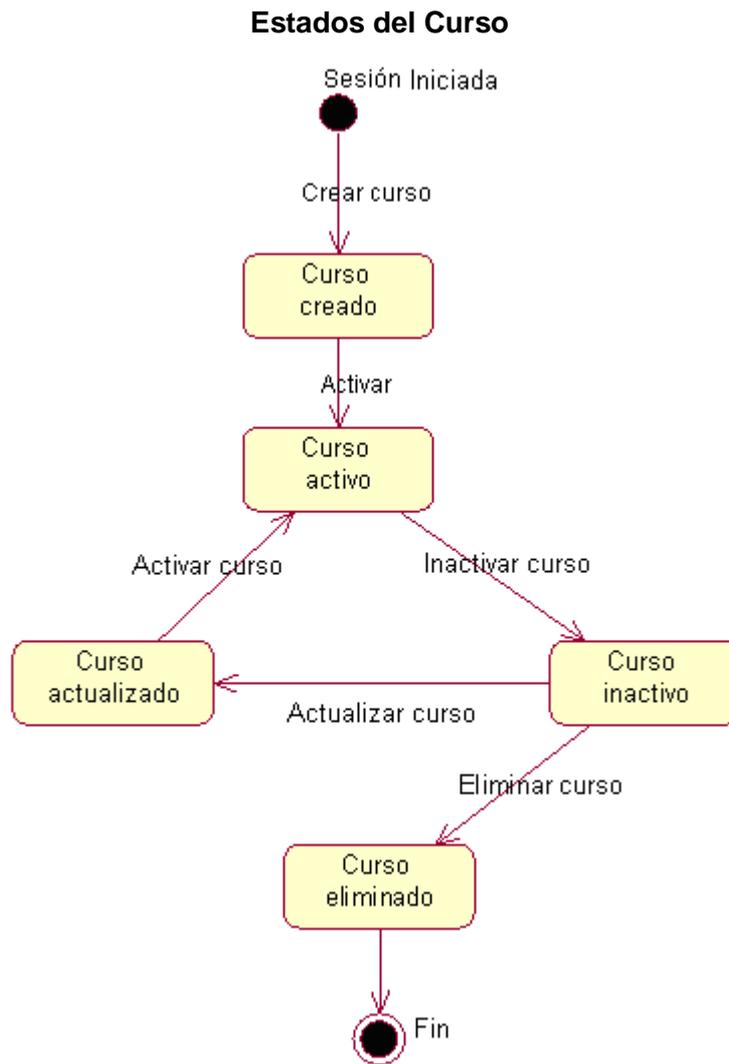
Mantenimiento de Becarios



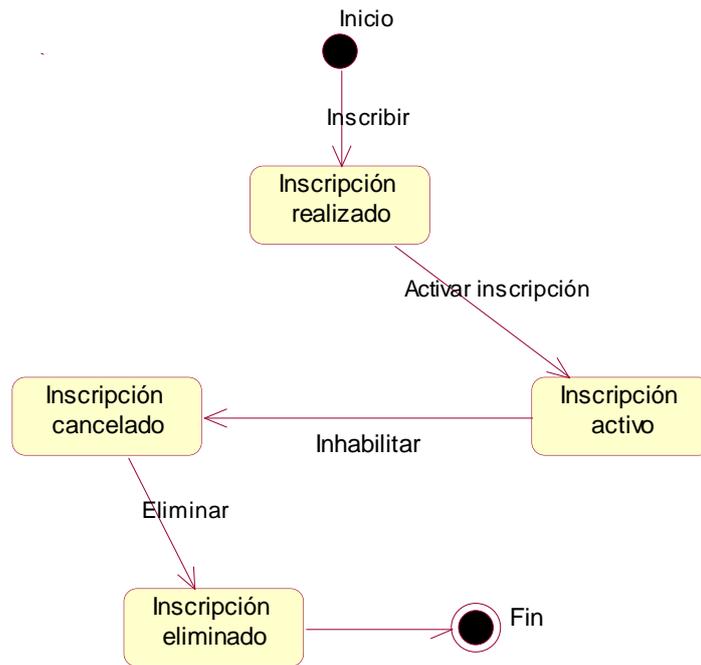
Reporte de Becarios Inscritos



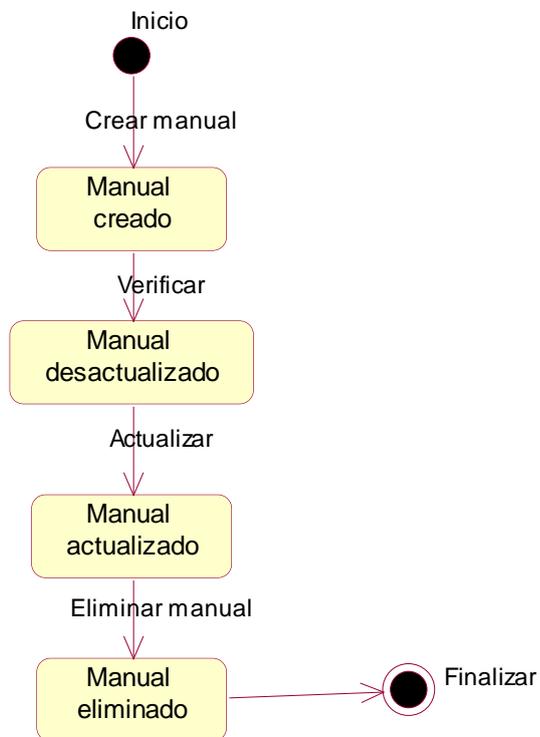
C.4 DIAGRAMAS DE ESTADO



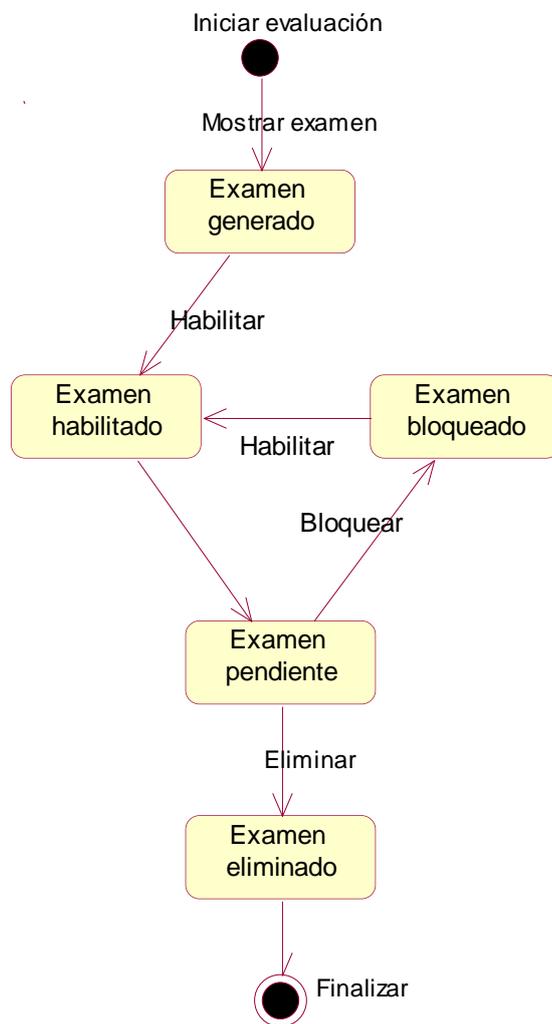
Estados de Inscripción



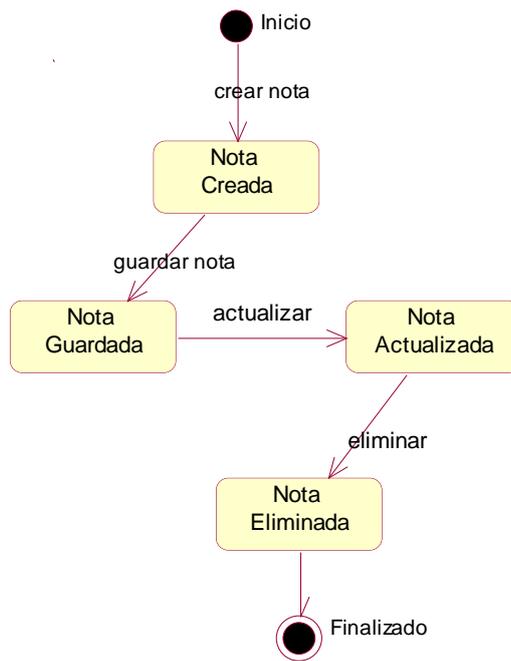
Estados de Manual



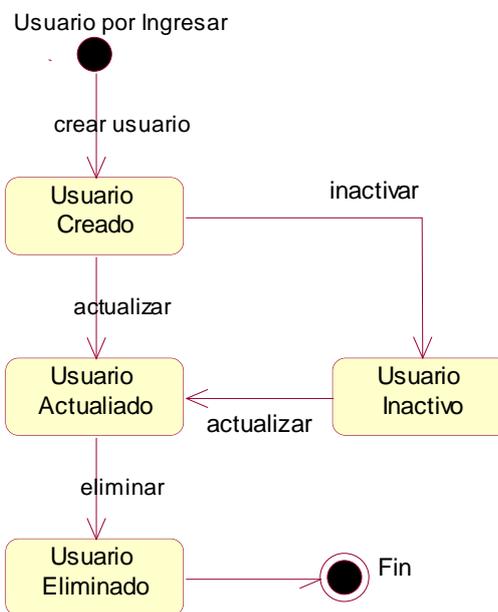
Estados de Examen



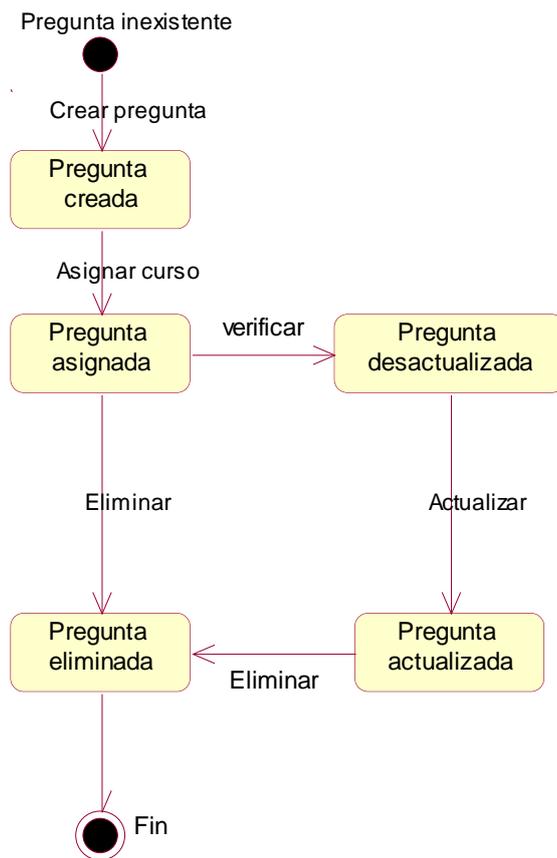
Estados de Nota



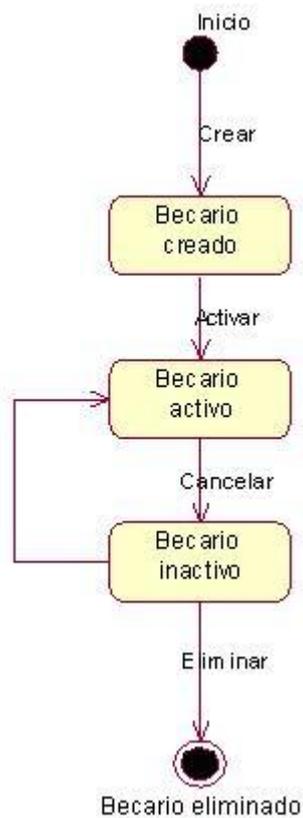
Estados de Usuario



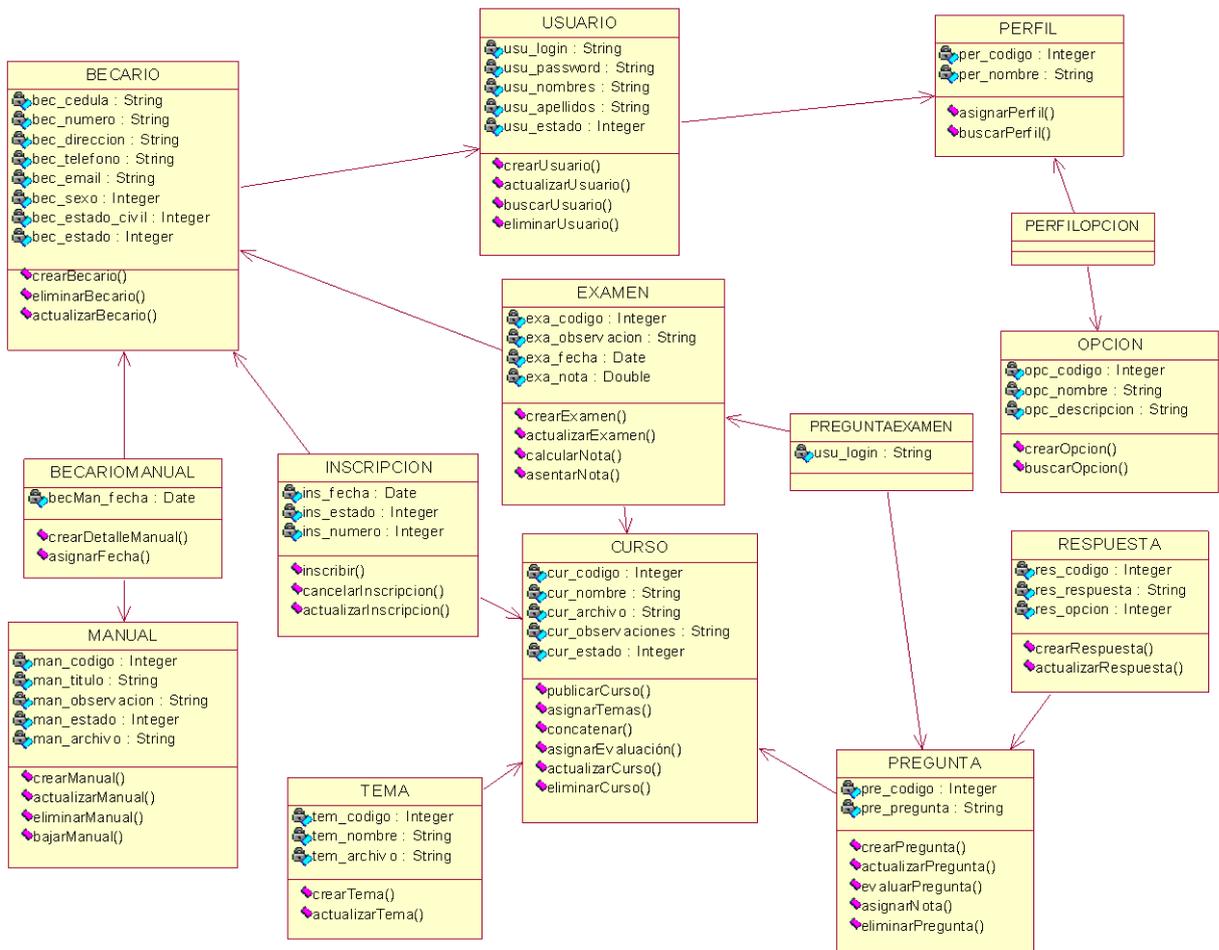
Estados de Pregunta



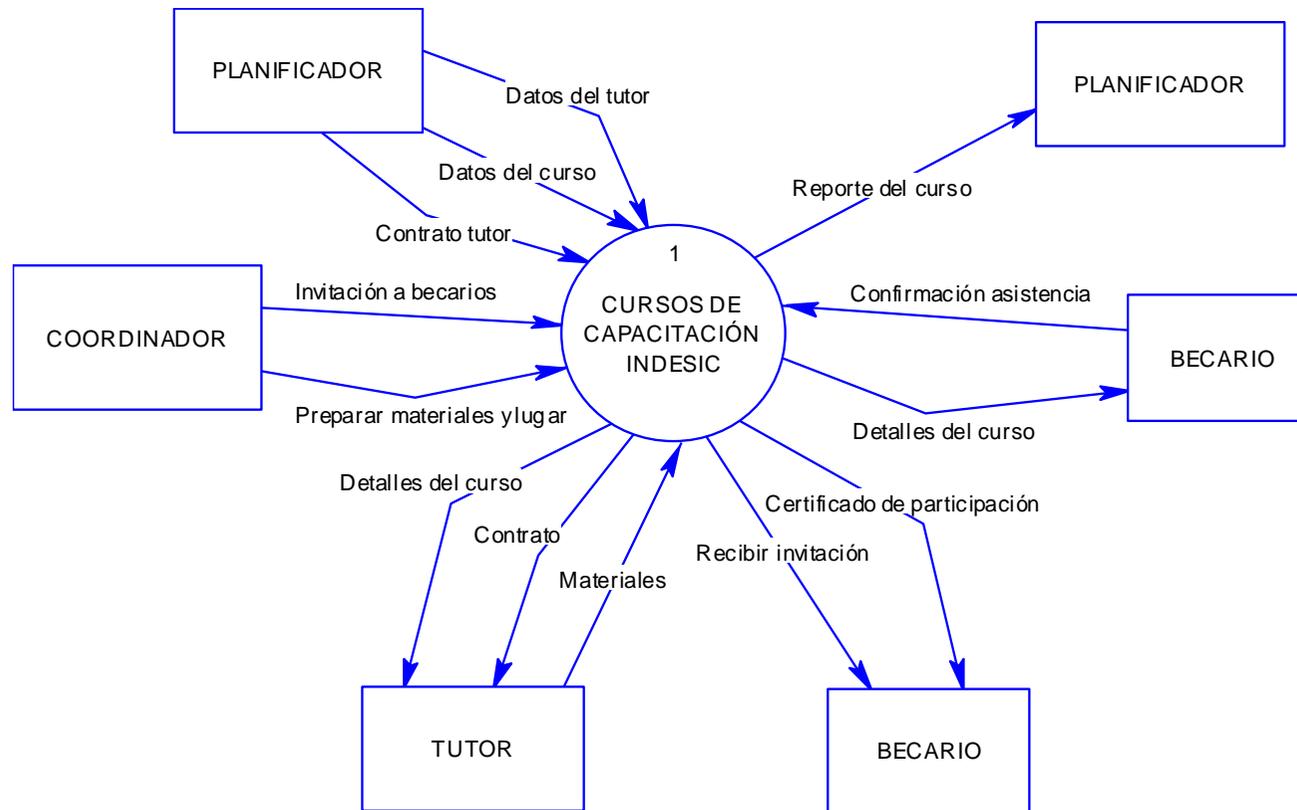
Estado de Becario



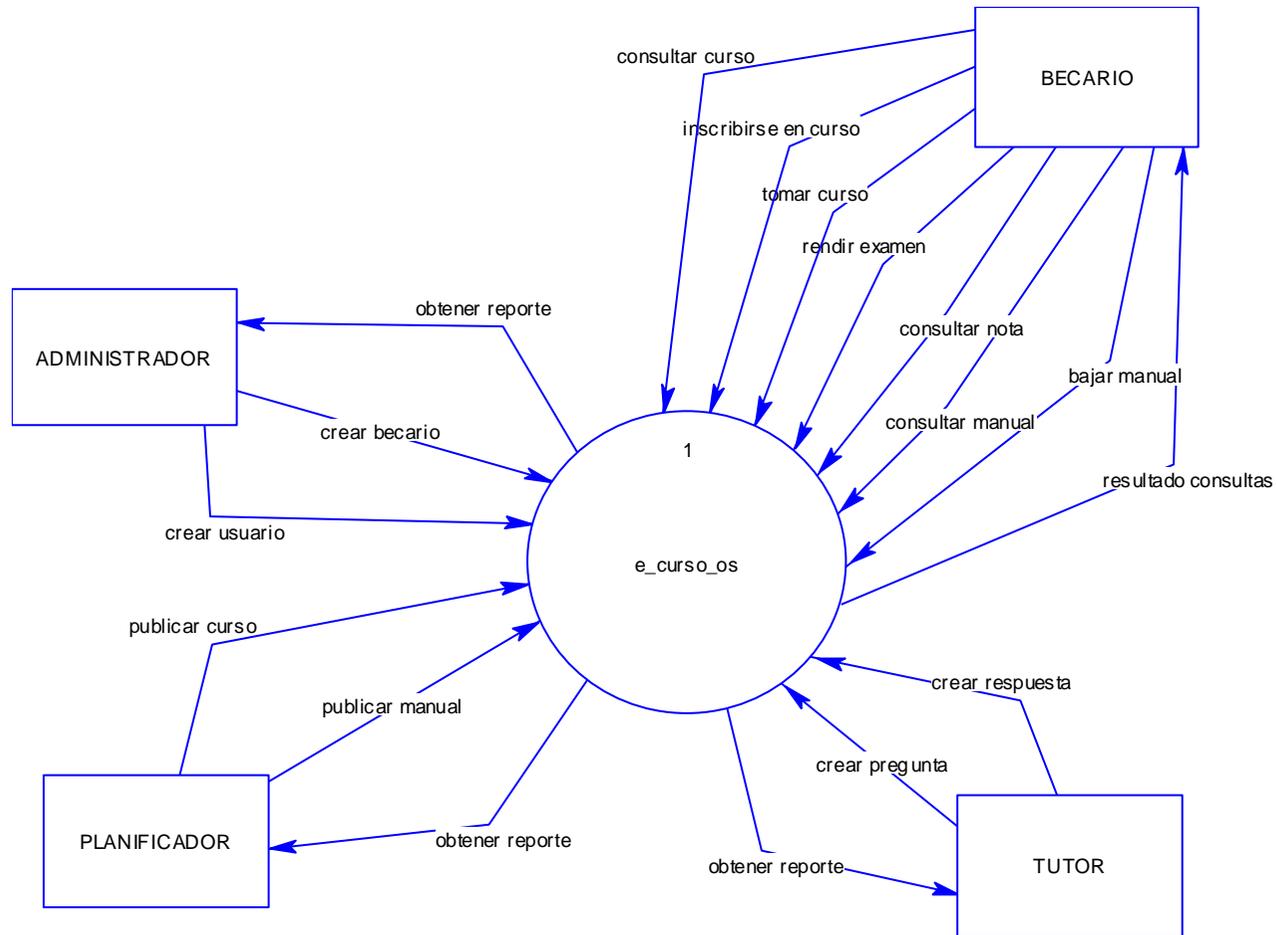
C.5 DIAGRAMAS DE CLASES



C.6 DIAGRAMA DE CONTEXTO - SITUACIÓN ACTUAL

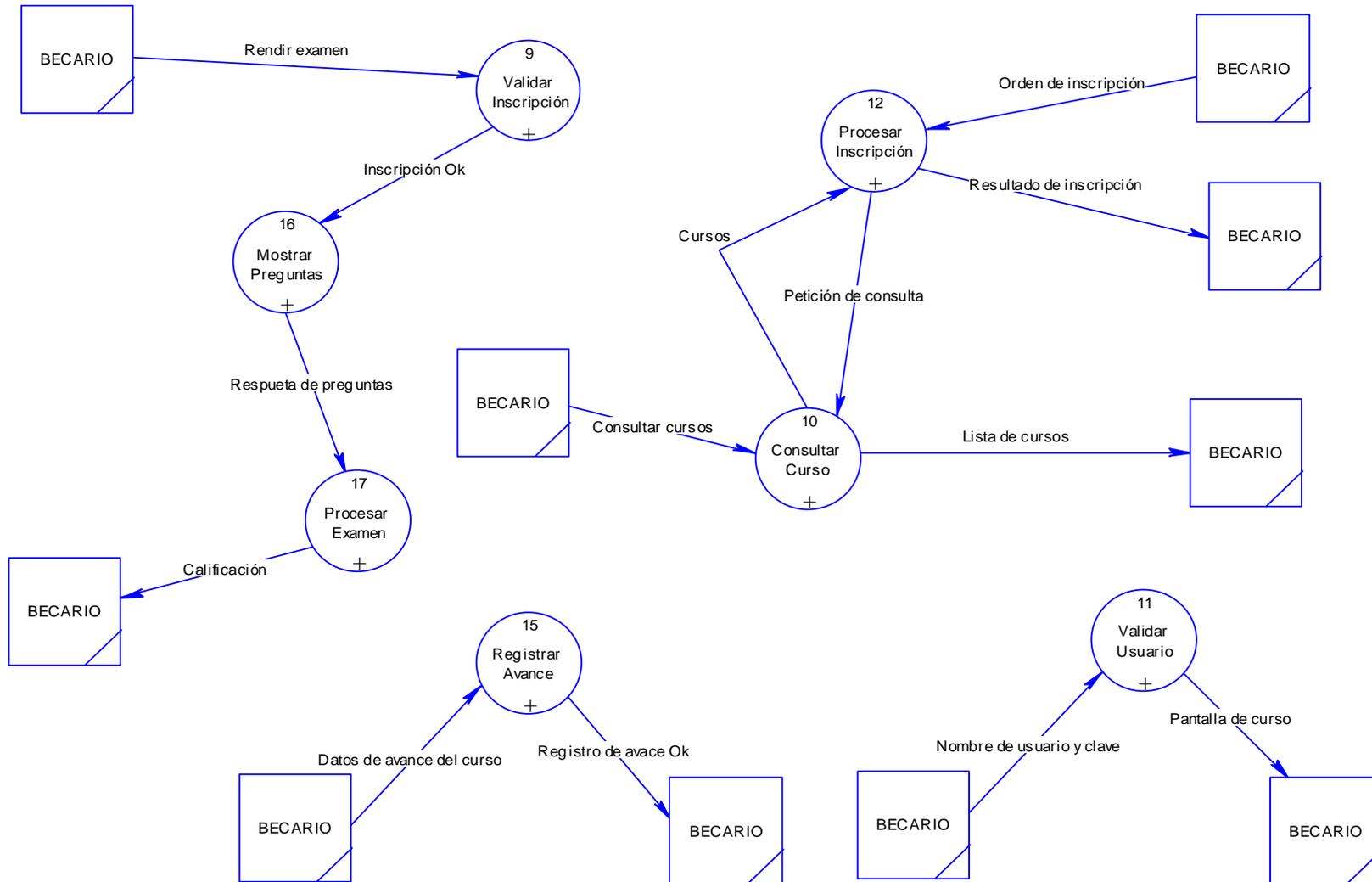


C.7 DIAGRAMA DE CONTEXTO PROPUESTO

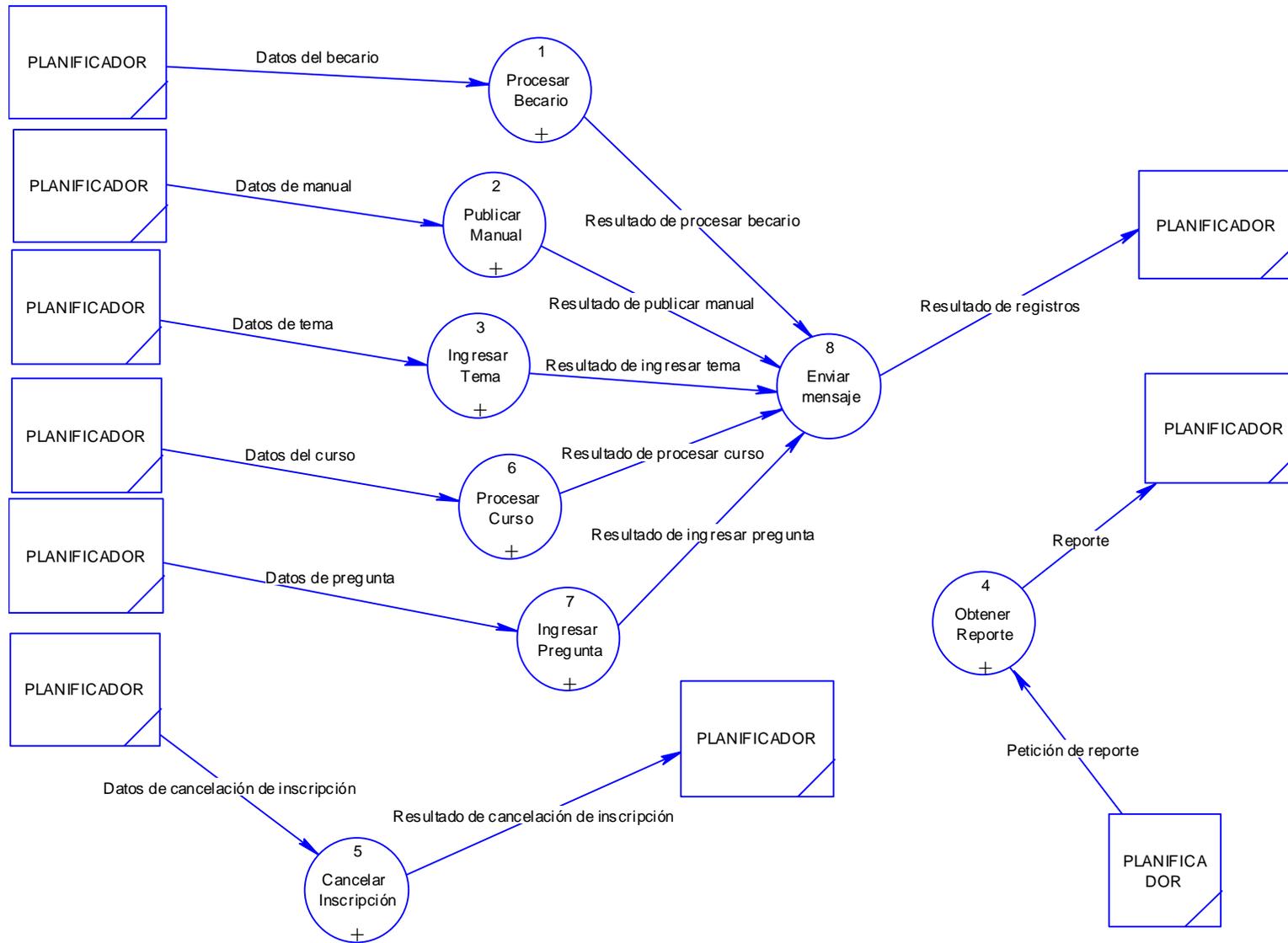


C.8 DIAGRAMAS DE FLUJO DE DATOS

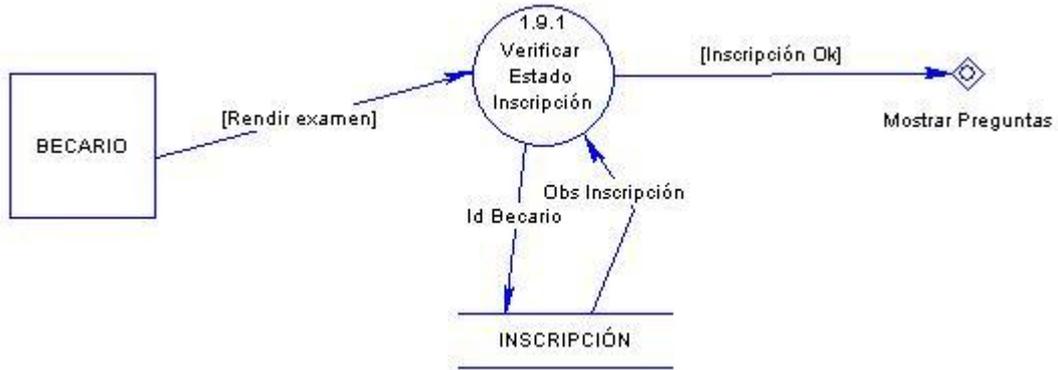
DFDs - Becario



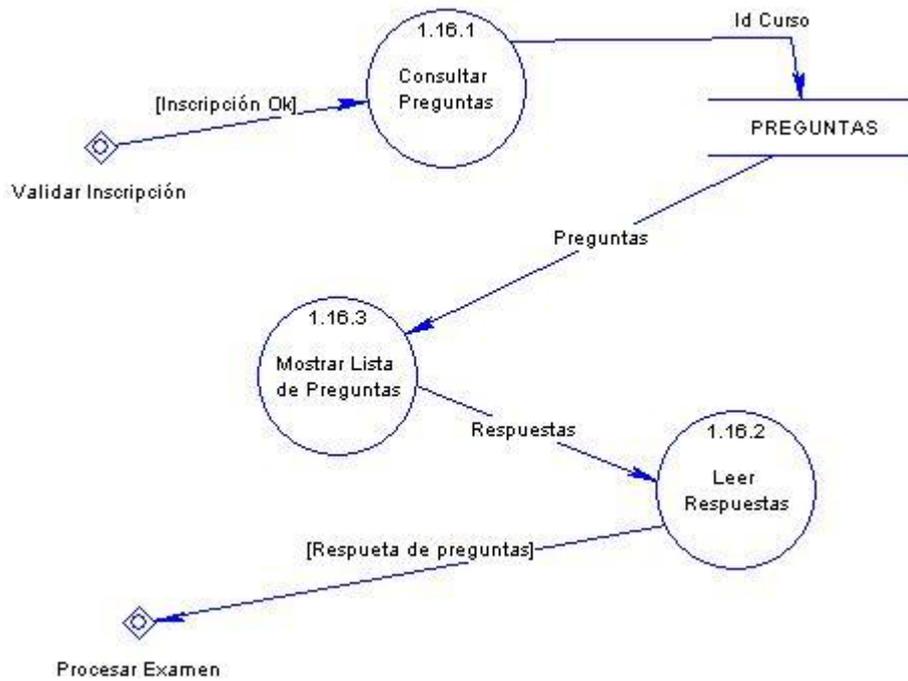
DFDs - Planificador



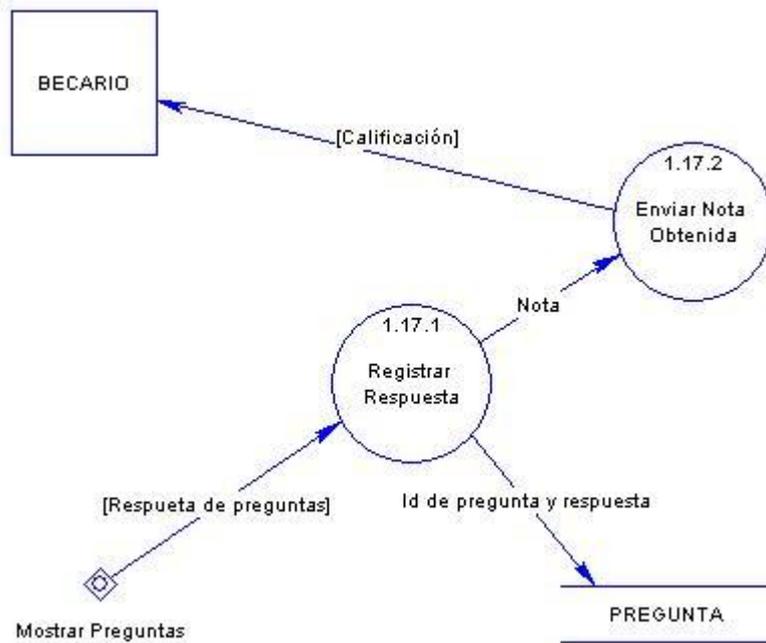
DFD - Validar Inscripción



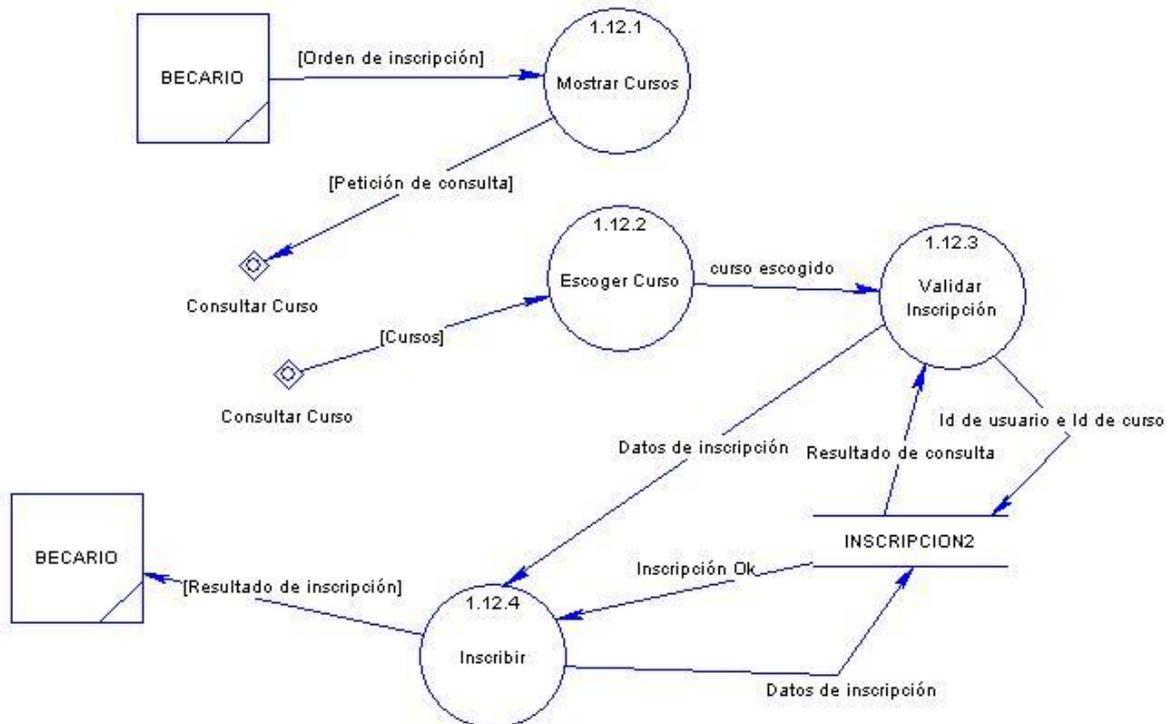
Mostrar Preguntas



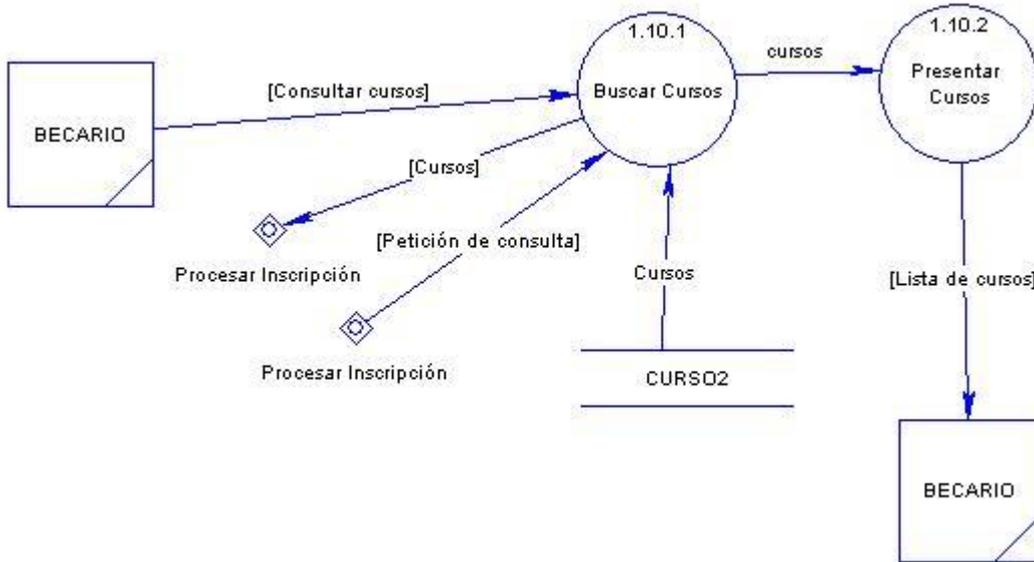
Procesar Examen



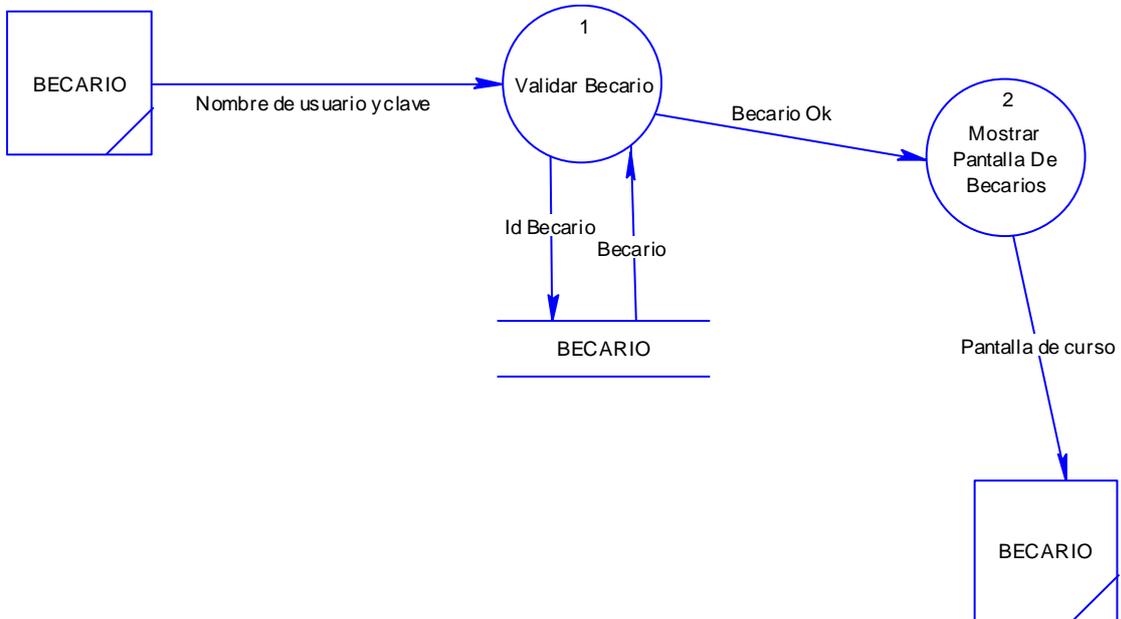
Procesar Inscripción



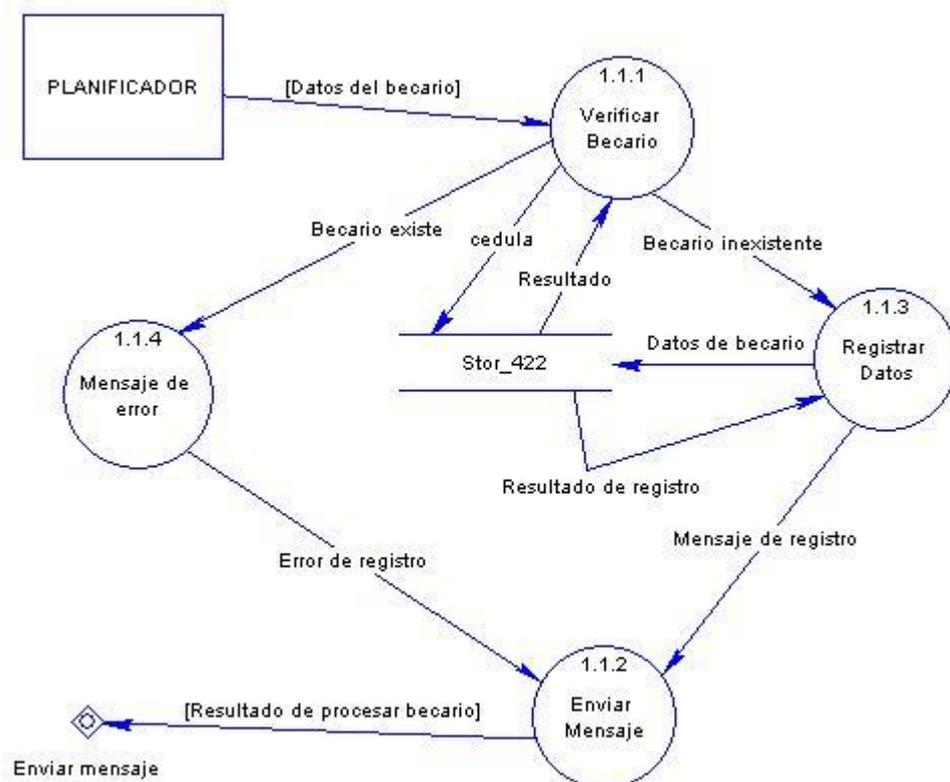
Consultar Curso



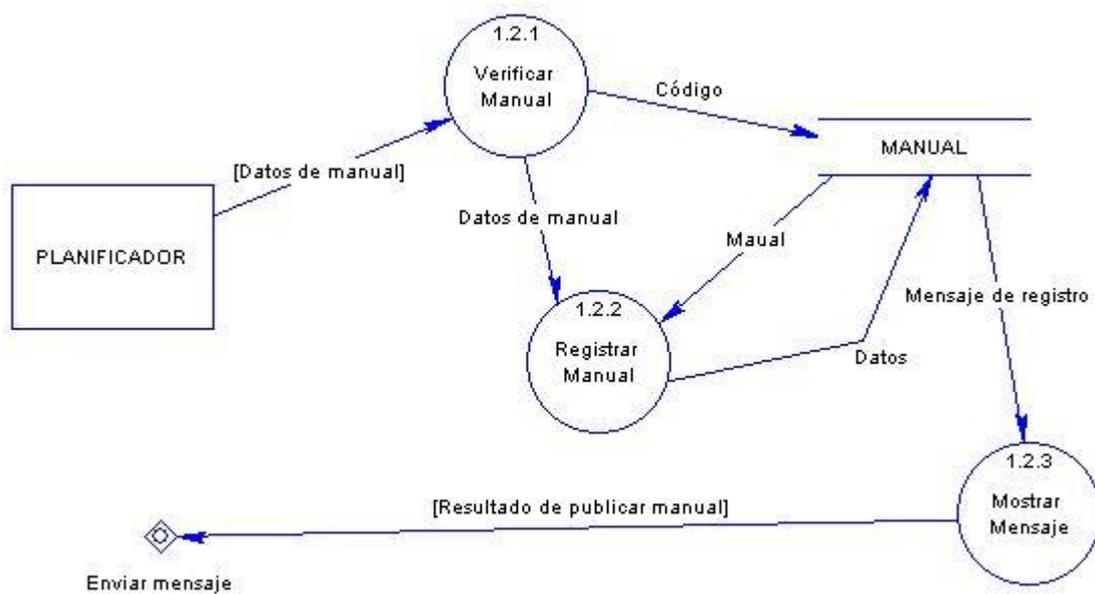
Validar Becario



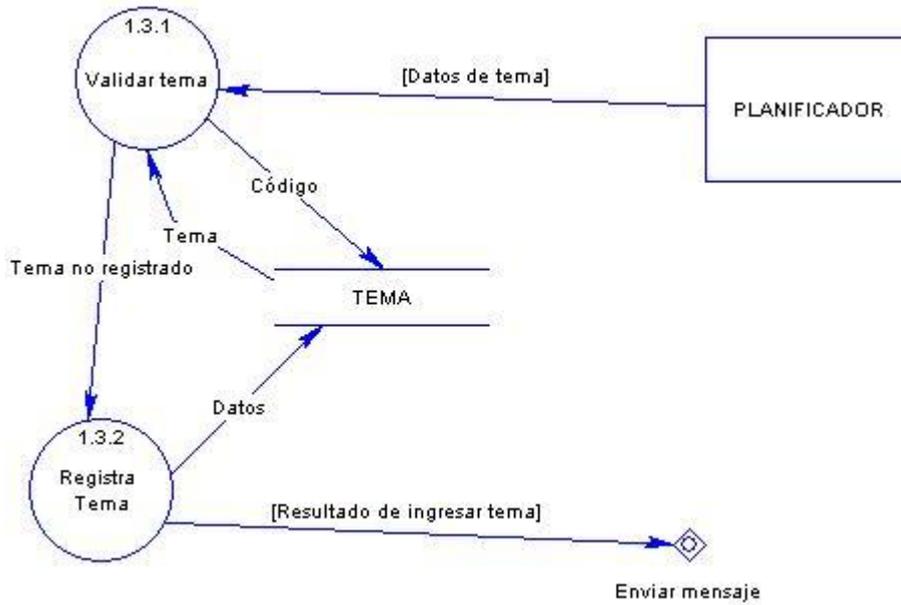
Procesar Becario



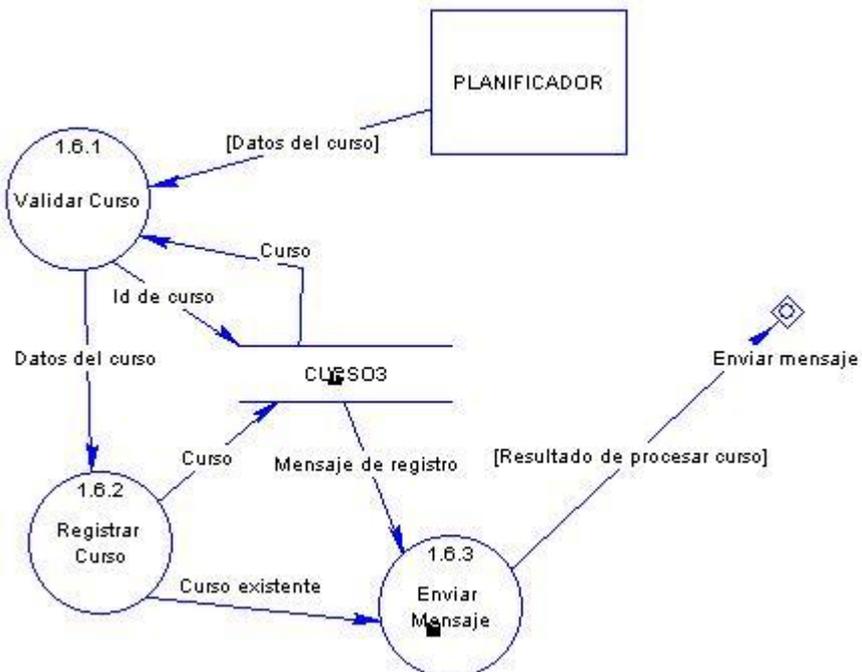
Publicar Manual



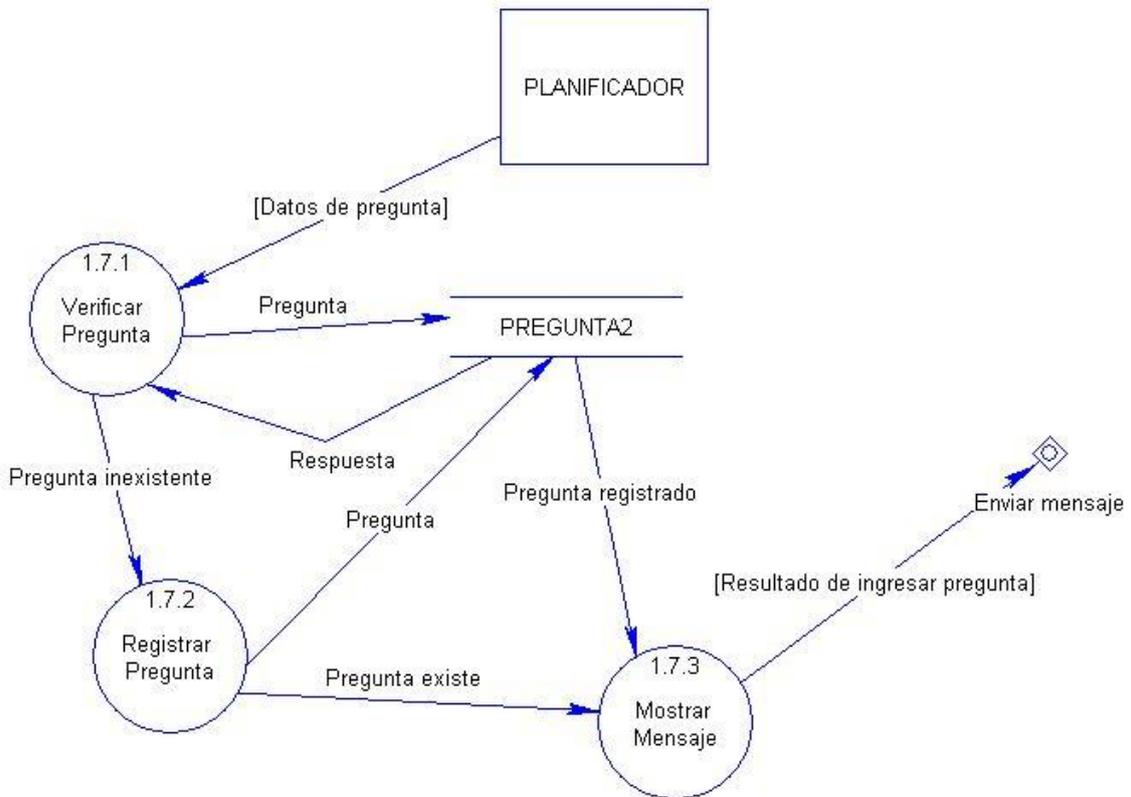
Ingresar Tema



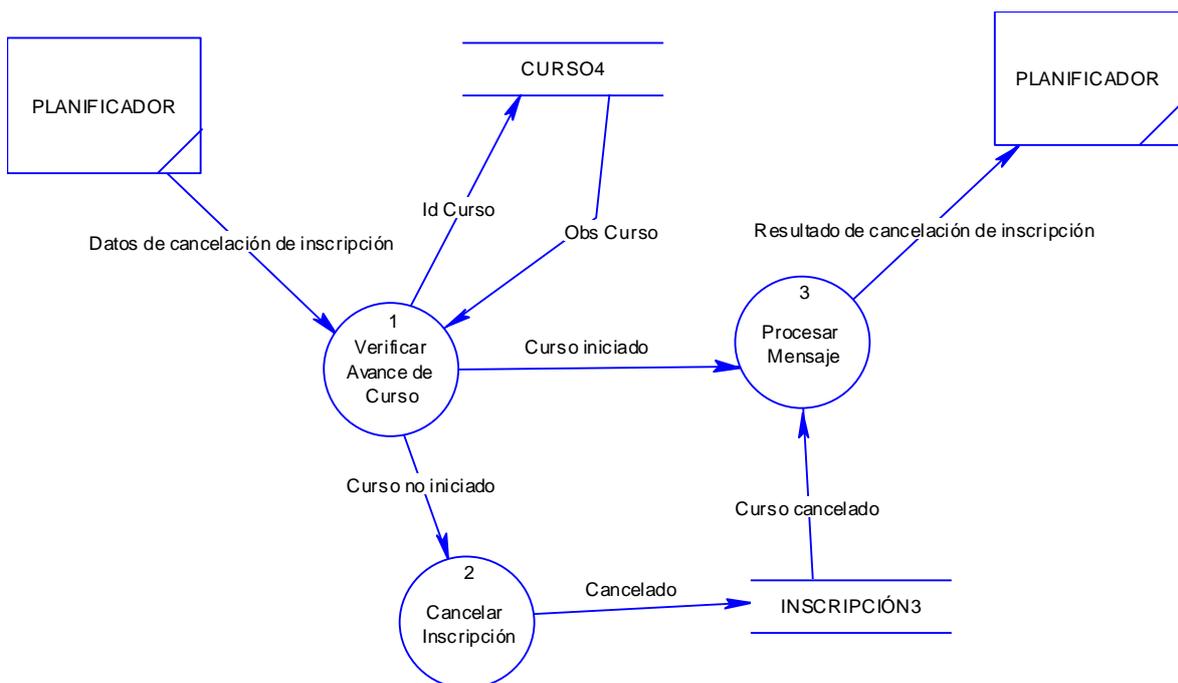
Procesar Curso



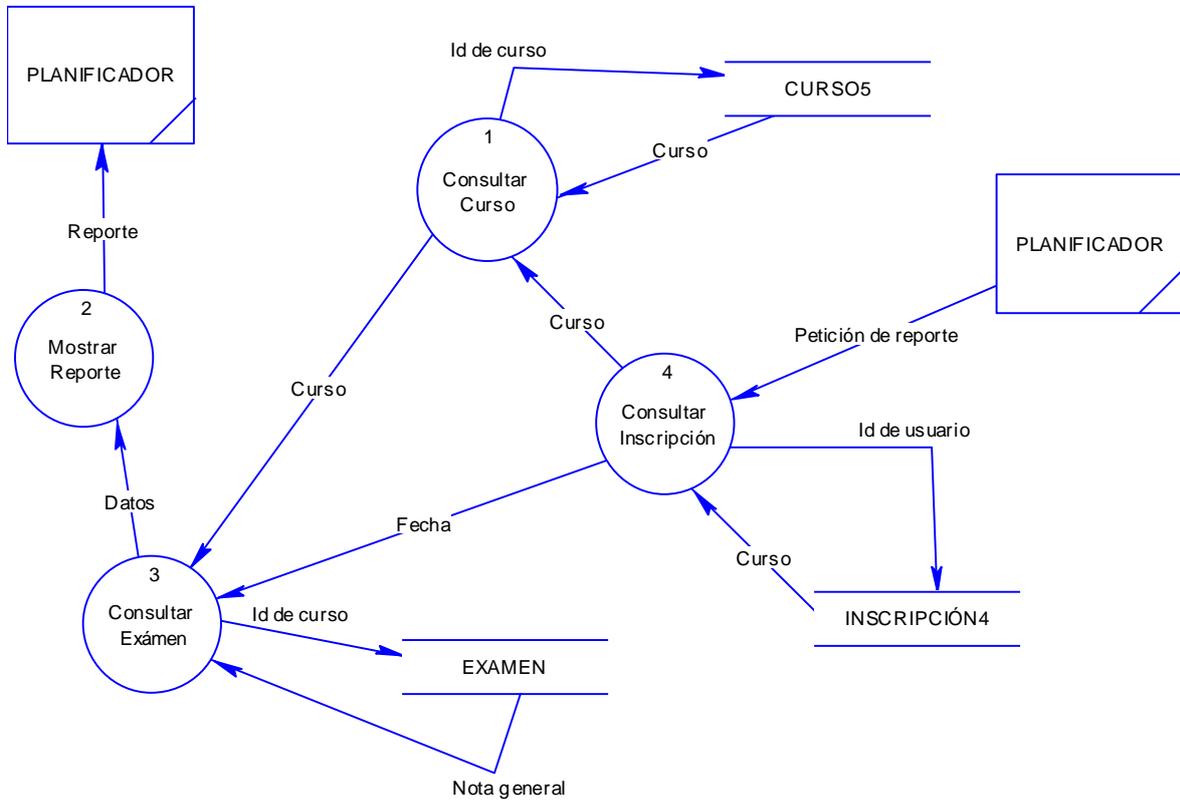
Ingresar Pregunta



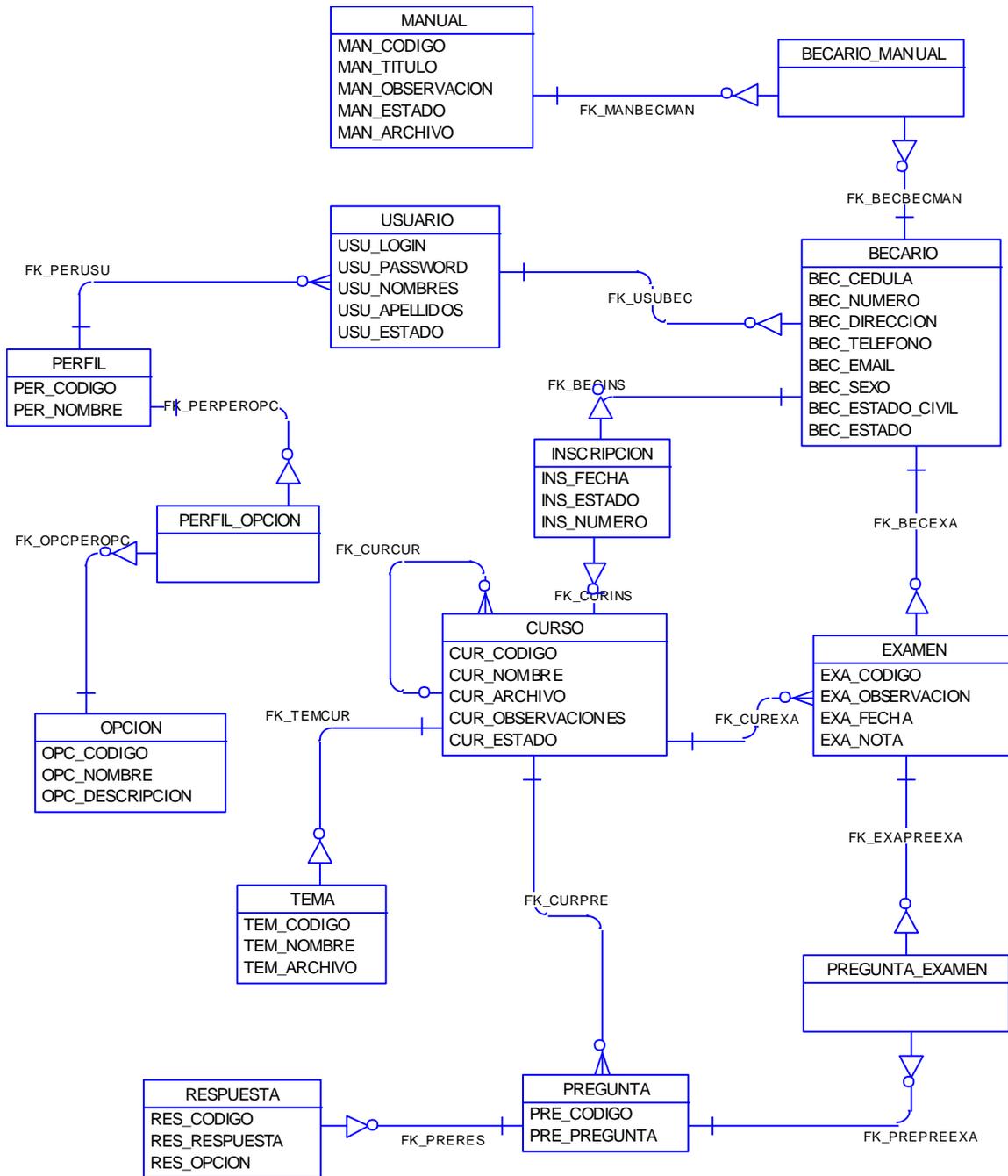
Cancelar Inscripción



Reportes

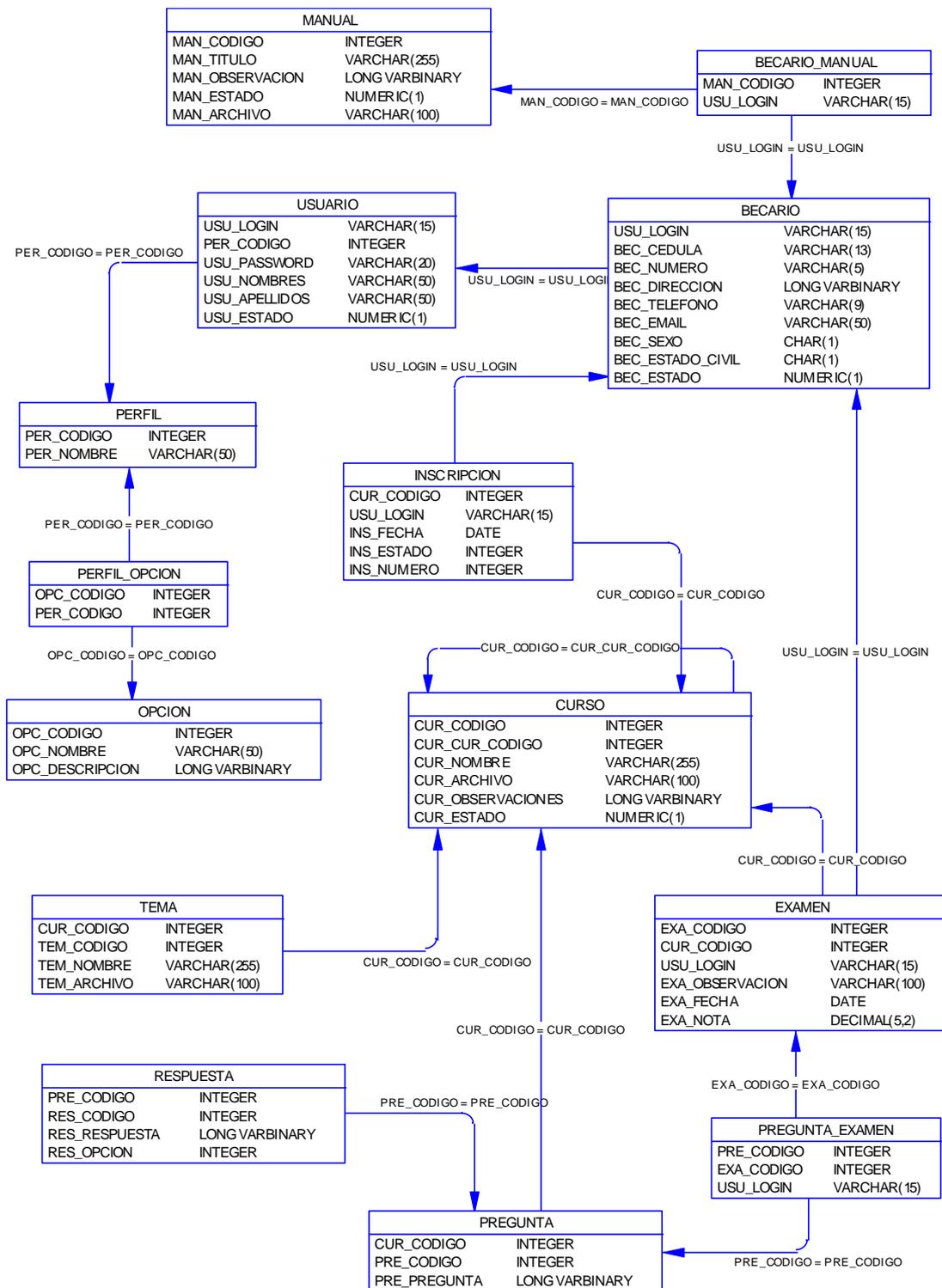


C.9 MODELO LÓGICO DE DATOS



ANEXO D: MODELO FÍSICO DE DATOS

Modelo Físico



ANEXO E: SCRIPT DE LAS TABLAS

create table PERFIL

```
(
  PER_CODIGO      INTEGER      not null,
  PER_NOMBRE      VARCHAR(50)   not null,
  primary key (PER_CODIGO)
);
```

create unique index PERFIL_PK on PERFIL (PER_CODIGO asc);

create table OPCION

```
(
  OPC_CODIGO      INTEGER      not null,
  OPC_NOMBRE      VARCHAR(50)   not null,
  OPC_DESCRIPCION LONG VARBINARY ,
  primary key (OPC_CODIGO)
);
```

create unique index OPCION_PK on OPCION (OPC_CODIGO asc);

create table CURSO

```
(
  CUR_CODIGO      INTEGER      not null,
  CUR_CUR_CODIGO  INTEGER      ,
  CUR_NOMBRE      VARCHAR(255)  not null,
  CUR_ARCHIVO     VARCHAR(100)  not null,
  CUR_OBSERVACIONES LONG VARBINARY ,
  CUR_ESTADO      NUMERIC(1)   not null,
  primary key (CUR_CODIGO),
  foreign key (CUR_CUR_CODIGO)
  references CURSO (CUR_CODIGO)
);
```

```

create unique index CURSO_PK on CURSO (CUR_CODIGO asc);
create index FK_CURCUR_FK on CURSO (CUR_CUR_CODIGO asc);
create table MANUAL
(
    MAN_CODIGO      INTEGER          not null,
    MAN_TITULO      VARCHAR(255)     not null,
    MAN_OBSERVACION LONG VARBINARY  ,
    MAN_ESTADO      NUMERIC(1)       not null,
    MAN_ARCHIVO     VARCHAR(100)     not null,
    primary key (MAN_CODIGO)
);
create unique index MANUAL_PK on MANUAL (MAN_CODIGO asc);
create table USUARIO
(
    USU_LOGIN       VARCHAR(15)      not null,
    PER_CODIGO      INTEGER          not null,
    USU_PASSWORD    VARCHAR(20)      not null,
    USU_NOMBRES     VARCHAR(50)      not null,
    USU_APELLIDOS   VARCHAR(50)      not null,
    USU_ESTADO      NUMERIC(1)       not null,
    primary key (USU_LOGIN),
    foreign key (PER_CODIGO)
        references PERFIL (PER_CODIGO)
);
create unique index USUARIO_PK on USUARIO (USU_LOGIN asc);
create index FK_PERUSU_FK on USUARIO (PER_CODIGO asc);
create table BECARIO
(

```

```

USU_LOGIN      VARCHAR(15)      not null,
BEC_CEDULA     VARCHAR(13)     not null,
BEC_NUMERO     VARCHAR(5)      not null,
BEC_DIRECCION  LONG VARBINARY  not null,
BEC_TELEFONO   VARCHAR(9)      ,
BEC_EMAIL      VARCHAR(50)     ,
BEC_SEXO       CHAR(1)      not null,
BEC_ESTADO_CIVIL CHAR(1)      not null,
BEC_ESTADO     NUMERIC(1)     not null,
primary key (USU_LOGIN),
foreign key (USU_LOGIN)
    references USUARIO (USU_LOGIN)
);

create unique index BECARIO_PK on BECARIO (USU_LOGIN asc);

create table EXAMEN
(
    EXA_CODIGO     INTEGER      not null,
    CUR_CODIGO     INTEGER      not null,
    USU_LOGIN      VARCHAR(15)  not null,
    EXA_OBSERVACION VARCHAR(100)  ,
    EXA_FECHA      DATE          ,
    EXA_NOTA       DECIMAL(5,2) ,
    primary key (EXA_CODIGO),
    foreign key (CUR_CODIGO)
        references CURSO (CUR_CODIGO),
    foreign key (USU_LOGIN)
        references BECARIO (USU_LOGIN)
);

```

```

create unique index EXAMEN_PK on EXAMEN (USU_LOGIN asc, EXA_CODIGO asc);
create index FK_CUREXA_FK on EXAMEN (CUR_CODIGO asc);
create index FK_BECEXA_FK on EXAMEN (USU_LOGIN asc);
create table PREGUNTA
(
    CUR_CODIGO    INTEGER        not null,
    PRE_CODIGO    INTEGER        not null,
    PRE_PREGUNTA LONG VARBINARY not null,
    primary key (PRE_CODIGO),
    foreign key (CUR_CODIGO)
        references CURSO (CUR_CODIGO)
);
create unique index PREGUNTA_PK on PREGUNTA (PRE_CODIGO asc);
create index RELATION_318_FK on PREGUNTA (CUR_CODIGO asc);
create table PERFIL_OPCION
(
    OPC_CODIGO    INTEGER        not null,
    PER_CODIGO    INTEGER        not null,
    primary key (OPC_CODIGO, PER_CODIGO),
    foreign key (PER_CODIGO)
        references PERFIL (PER_CODIGO),
    foreign key (OPC_CODIGO)
        references OPCION (OPC_CODIGO)
);
create unique index PERFIL_OPCION_PK on PERFIL_OPCION (OPC_CODIGO asc,
PER_CODIGO asc);
create index FK_PERPEROPC_FK on PERFIL_OPCION (PER_CODIGO asc);

```

```

create index FK_OPCEPEROPC_FK on PERFIL_OPCION (OPC_CODIGO asc);

create table PREGUNTA_EXAMEN
(
    PRE_CODIGO    INTEGER        not null,
    EXA_CODIGO    INTEGER        not null,
    USU_LOGIN     VARCHAR(15)    not null,
    primary key (PRE_CODIGO, EXA_CODIGO, USU_LOGIN),
    foreign key (EXA_CODIGO)
        references EXAMEN (EXA_CODIGO),
    foreign key (PRE_CODIGO)
        references PREGUNTA (PRE_CODIGO)
);

create unique index PREGUNTA_EXAMEN_PK on PREGUNTA_EXAMEN
(PRE_CODIGO asc, EXA_CODIGO asc, USU_LOGIN asc);

create index FK_PREPREEXA_FK on PREGUNTA_EXAMEN (PRE_CODIGO asc);

create table RESPUESTA
(
    PRE_CODIGO    INTEGER        not null,
    RES_CODIGO    INTEGER        not null,
    RES_RESPUESTA LONG VARBINARY not null,
    RES_OPCION    INTEGER        ,
    primary key (PRE_CODIGO, RES_CODIGO),
    foreign key (PRE_CODIGO)
        references PREGUNTA (PRE_CODIGO)
);

create unique index RESPUESTA_PK on RESPUESTA (PRE_CODIGO asc,
RES_CODIGO asc);

```

```

create index FK_PRERES_FK on RESPUESTA (PRE_CODIGO asc);

create table INSCRIPCION
(
    CUR_CODIGO    INTEGER        not null,
    USU_LOGIN     VARCHAR(15)    not null,
    INS_FECHA     DATE           not null,
    INS_ESTADO    INTEGER        ,
    INS_NUMERO    INTEGER        ,
    primary key (CUR_CODIGO, USU_LOGIN),
    foreign key (CUR_CODIGO)
        references CURSO (CUR_CODIGO),
    foreign key (USU_LOGIN)
        references BECARIO (USU_LOGIN)
);

create unique index INSCRIPCION_PK on INSCRIPCION (CUR_CODIGO asc,
USU_LOGIN asc);

create index FK_CURINS_FK on INSCRIPCION (CUR_CODIGO asc);

create index FK_BECINS_FK on INSCRIPCION (USU_LOGIN asc);

create table BECARIO_MANUAL
(
    MAN_CODIGO    INTEGER        not null,
    USU_LOGIN     VARCHAR(15)    not null,
    primary key (MAN_CODIGO, USU_LOGIN),
    foreign key (MAN_CODIGO)
        references MANUAL (MAN_CODIGO),
    foreign key (USU_LOGIN)
        references BECARIO (USU_LOGIN)
);

```

```

create unique index BECARIO_MANUAL_PK on BECARIO_MANUAL (MAN_CODIGO
asc, USU_LOGIN asc);
create index FK_MANBECMAN_FK on BECARIO_MANUAL (MAN_CODIGO asc);
create index FK_BECBECMAN_FK on BECARIO_MANUAL (USU_LOGIN asc);
create table TEMA
(
CUR_CODIGO    INTEGER        not null,
TEM_CODIGO    INTEGER        not null,
TEM_NOMBRE    VARCHAR(255)    not null,
TEM_ARCHIVO   VARCHAR(100)    not null,
primary key (CUR_CODIGO, TEM_CODIGO),
foreign key (CUR_CODIGO)
references CURSO (CUR_CODIGO)
);
create unique index TEMA_PK on TEMA (CUR_CODIGO asc, TEM_CODIGO asc);
create index FK_TEMCUR_FK on TEMA (CUR_CODIGO asc);

```

ANEXO F: CONFIGURACIÓN DE LA PLATAFORMA DE DESARROLLO

INSTALACIÓN DE LINUX

Para este proyecto se ha escogido Linux RedHat 9.0, ya que es una de las versiones de Linux del que más documentación se puede encontrar, además no tiene ningún costo comercial.

La instalación de Linux se ha hecho en una computadora con Procesador Intel P4 de 2.0 Ghz, con tarjeta madre BioStar, 256 MB de RAM y 60 GB de disco duro.

La instalación de Linux se ha realizado en una computadora con el sistema operativo Windows preinstalado, por lo que se ha tenido que particionar el disco. La primera partición de 40 Gb. se ha usado para Windows y la otra partición de 20 Gb. se ha usado par Linux.

Pasos Previos a la Instalación

Antes de la instalación de Linux es necesario que se tome en cuenta los siguientes aspectos:

- Revisar que la versión de Linux que se va a instalar, soporte el hardware disponible.
- Revisar el correcto funcionamiento de las tarjetas incorporada, como por ejemplo: red, módem, video, sonido.
- Respalidar la información que considere importante

Como se mencionó, se tenía preinstalado Windows en todo el disco, por lo que se ha respaldado la información y se procede a particionar usando el fdisk del DOS.

Se borra la partición única existente y se vuelve a crear una nueva partición primaria de 40 GB de capacidad, en el que se instala nuevamente el Windows y se deja libre los 20 GB restantes, este espacio es usado para las particiones necesarias de Linux.

La instalación de Linux requiere de al menos tres particiones que se los realiza al momento de la instalación, estas particiones son:

- /boot : es donde se guardan los parámetros para arrancar el sistema, requiere de al menos de 75 MB. Para este caso se ha asignado 1 GB.
- / : (raíz), es donde se coloca el sistema operativo, requiere de al menos 512 MB. Para este caso se le ha asignado 10 GB, ya que es donde quedara alojado el sistema Linux.
- swap: Es el espacio para la memoria virtual del sistema, se le debe asignar el doble del RAM físico disponible. Pero en este caso, se ha asignado 1 GB.

Como se puede apreciar, solamente se ha usado 12 GB para la instalación del Linux, quedando libre 8GB, este espacio se lo puede usar en otra necesidad que se tenga.

En el caso de que el sistema operativo ya vaya a entrar en producción, es muy importante que se analice detenidamente el espacio necesario para cada partición. Además, no es conveniente que convivan dos sistemas operativos en un servidor de producción, ya que no se puede usar uno y otro a la vez.

En este caso, se lo ha configurado de esta manera, porque es una etapa de construcción de la aplicación y soporta eficientemente los requerimientos.

Proceso de Instalación

La instalación de Linux se lo puede realizar de dos maneras, en modo gráfico o en modo texto, con las dos opciones se logra el mismo resultado, solo que en modo gráfico es más fácil e intuitivo, por lo que se seguirá el modo gráfico.

- Primeramente reiniciamos el sistema, con el disco 1 de Linux insertado en el CD ROM, el BIOS debe estar configurado para iniciar desde el CD.



Figura 29: Ventana Inicial de Instalación de Linux

- Luego nos aparece la ventana en donde debemos escoger si queremos instalar en modo texto o en modo gráfico, para modo texto digitamos text y para modo gráfico damos enter. Vea la Figura 29.
- Luego nos aparece la opción de verificar el estado del CD, a la que en este caso nos saltamos escogiendo skip, ya que estamos seguros del buen estado del CD.
- Entonces nos aparece la ventana del asistente de instalación, que nos muestra una breve explicación en el lado izquierdo y para avanzar simplemente damos clic en Next. A partir de ahora la instalación es en modo gráfico.
- Seguidamente debemos escoger el lenguaje para la instalación, se ha optado por instalar en español. Continuamos dando clic en Next. Vea la Figura 30.

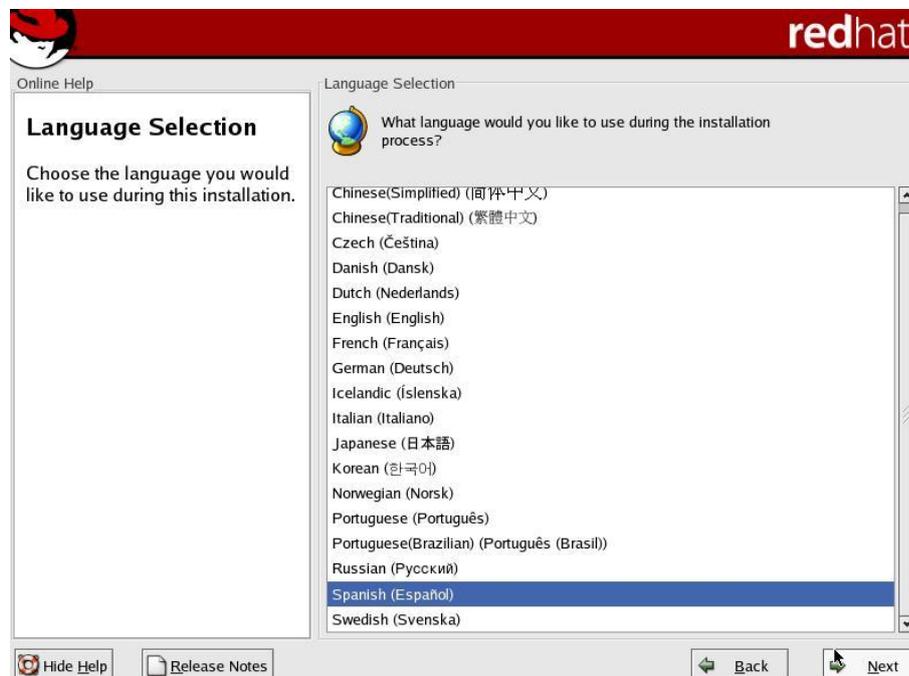


Figura 30: Ventana de Selección de Lenguaje

- En las siguientes dos ventanas, nos muestra la opción para configurar el teclado y Mouse, se lo debe escoger de acuerdo a las características disponibles, en este caso se ha escogido un teclado en inglés, y Mouse PS2 de rueda.
- Luego tenemos la ventana para escoger el tipo de instalación que necesitamos, mismo que pueden ser: Escritorio Personal, Estación de Trabajo, Servidor o Personalizada. Para el caso se requiere instalar como servidor. Hacemos clic en Siguiente y continuamos, como se muestra en la Figura 31.

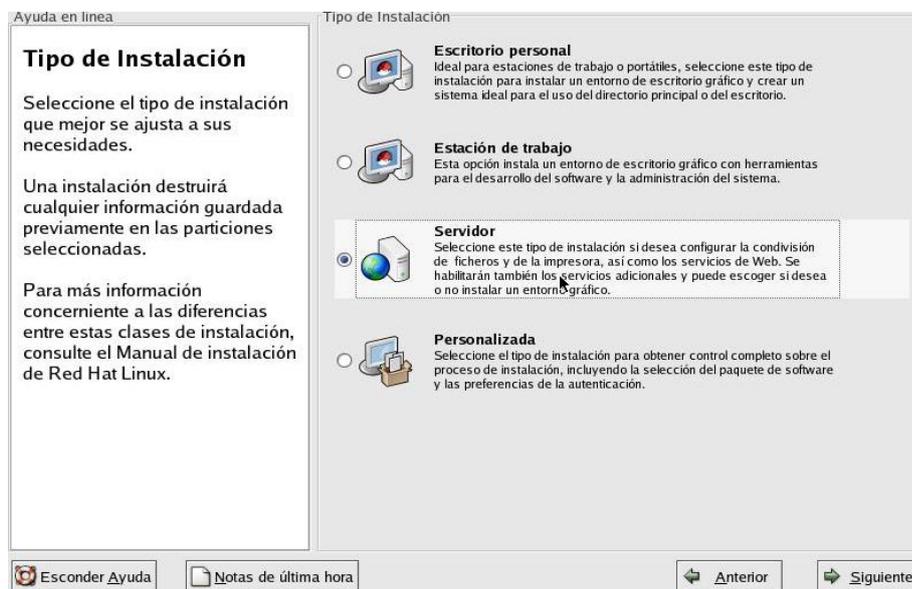


Figura 31: Ventana para Escoger el Tipo de Instalación

- A continuación debemos escoger el modo de crear las particiones, estos pueden ser automático o manual. Se ha escogido de forma manual para asignarles los espacios específicos. Damos clic en Siguiente para avanzar.
- Seguidamente nos aparece la ventana de partición donde debemos especificar las particiones correspondientes como se muestra a continuación, en la Figura 32.

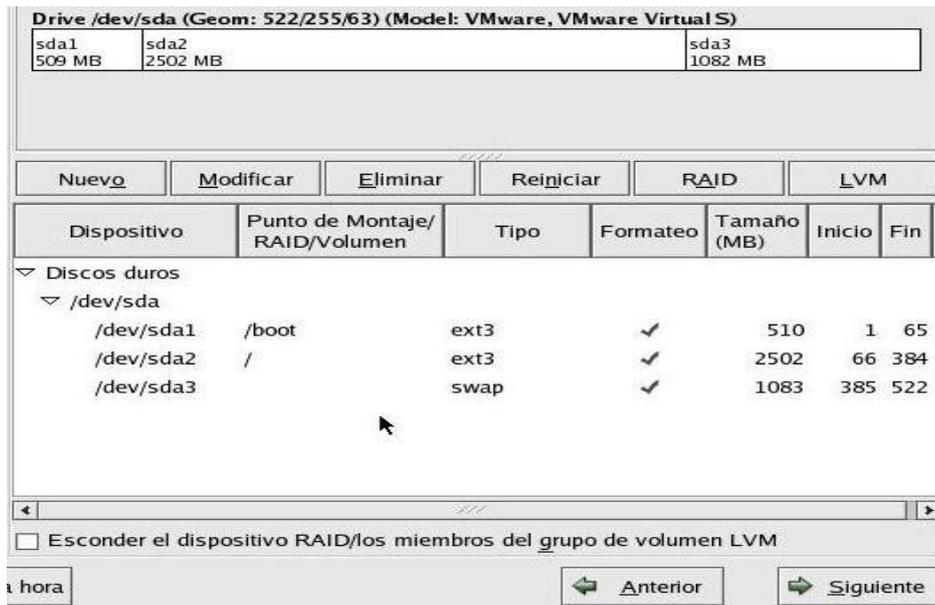


Figura 32: Ventana que indica las particiones

- Al dar clic en siguiente, tenemos una nueva ventana en la que debemos escoger con que sistema operativo se quiere iniciar, esto en caso de que tengamos dos sistemas operativos conviviendo en la misma maquina. Nuevamente damos clic en siguiente para continuar.
- Luego tenemos la ventana de configuración de la red. En este caso, se ha configurado como red tipo C, que es la que se usa en el INDESIC. También hay que asignar un nombre de host, que se lo ha puesto como curso.indesic.org, finalmente es necesario que se especifique la puerta de enlace y el DNS primario. Damos clic en siguiente y avanzamos. En la Figura 33, se muestra un ejemplo de la configuración de red.
- Seguidamente aparece la ventana del firewall, en el que se debe seleccionar con un visto los servicios permitidos. Damos clic en siguiente para continuar.



Figura 33: Ventana de Configuración de Red

- Después nos aparece la ventana para seleccionar el idioma de los paquetes adicionales que queremos instalar, en la que se ha escogido español, damos clic en siguiente para avanzar.
- Luego aparece la opción de configurara la zona horaria, que se puede hacerlo gráficamente, como se muestra en la Figura 34.



Figura 34: Configuración de la Zona Horaria

- Entonces nos aparece la opción de ingresar la contraseña de root, que es el usuario administrador de Linux, es importante ingresar una contraseña

segura pero también que no se olvide. Damos clic en siguiente para avanzar.

- Después debemos seleccionar los paquetes que deseamos instalar, empezando por las aplicaciones visuales si queremos administrar gráficamente, hasta las aplicaciones principales como Servidor Web, de Correo, DNS, y los que necesitemos, como se muestra en la Figura 35.

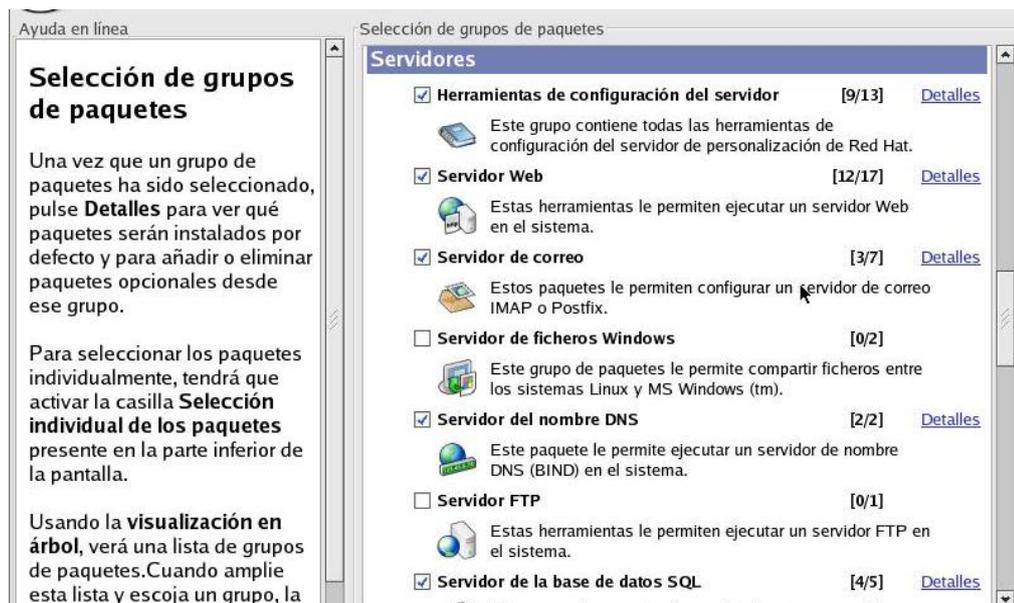


Figura 35: Selección de Paquetes a Instalar

- Finalmente Iniciamos la instalación, haciendo clic en el botón Instalar y hay que esperar hasta que el asistente pida el CD2 y CD3 respectivamente, luego de esto quedara instalado el Linux y listo para configurar según lo necesario.

Una vez terminado la instalación de Linux, se reinicia el sistema y nos pide ingresar el nombre de usuario y contraseña. Para poder configurar el servidor es necesario ingresar con la cuneta de *root* y la clave asignado al momento de la instalación.

El escritorio de Linux RedHat, es muy parecido al de Windows, lo que hace bastante agradable el uso. Además que Linux dispone de 4 escritorios, que hace la diferencia con Windows.

La Figura 36, muestra un ejemplo del escritorio de Linux.

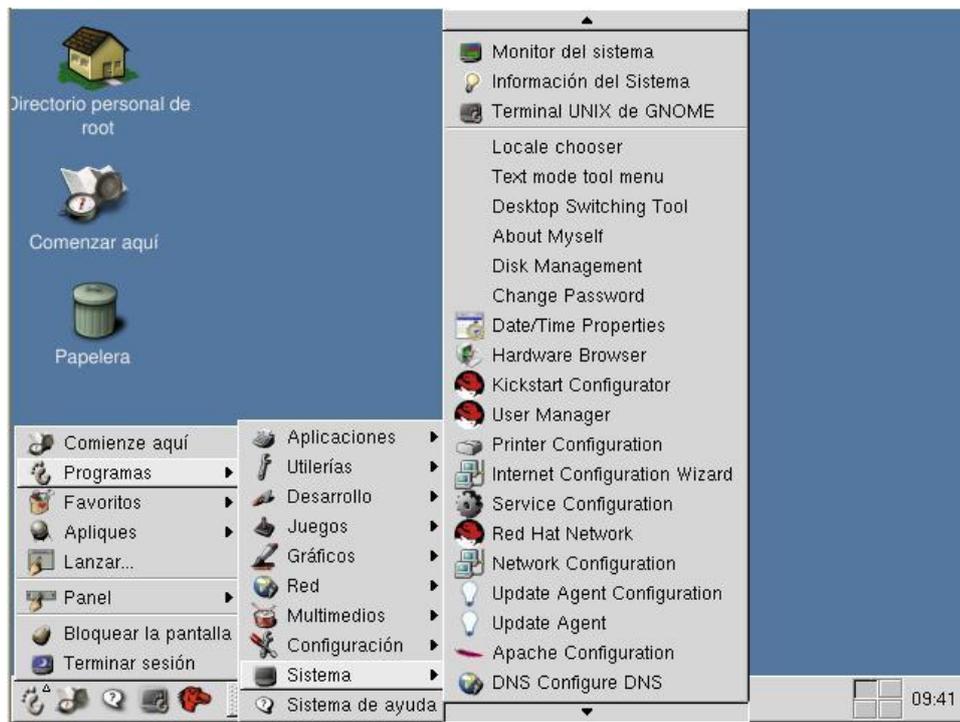


Figura 36: Escritorio de Linux RedHat

CONFIGURACIÓN DEL SERVIDOR DNS

El Sistema de Nombres de Dominio (DNS) permite transformar la dirección del sitio web ingresado desde un browser, en dirección IP, de modo que se pueda encontrar el host que contiene la página que se busca.

Por lo tanto es necesario que se configure adecuadamente para que encuentre el host. Esto se lo puede hacer en modo texto o, en modo gráfico que es más fácil e intuitivo.

La configuración del DNS se lo realizará gráficamente, en el que se puede realizar de una manera intuitiva y sin mayores dificultades.

Proceso de Configuración del DNS

- Hacemos un clic en el símbolo del pie de la ventana de Linux, apuntamos hasta Programas, luego vamos hasta Sistemas y Finalmente buscamos la opción Configure DNS.
- Entonces nos aparece una ventana en el que debemos crear una nueva zona maestra de envío, y en el nombre de dominio ponemos indesic.org, ya que al instalar habíamos puesto como nombre de host curso.indesic.org. Curso representa al nombre del host, mientras que indesic.org es el dominio. Vea el ejemplo en la Figura 37.

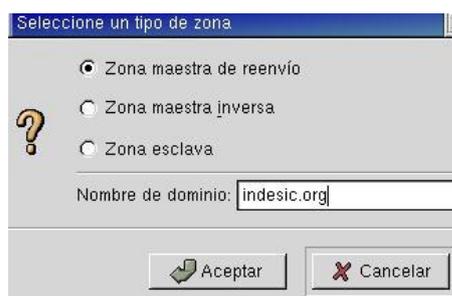


Figura 37: Ventana de Configuración de Zona Maestra

- Luego debemos especificar el nombre del host y la dirección IP asignado a ese host. Para esto hacemos clic en *Añadir* en la ventana de *configuración de traducciones del nombre a IP*, como se muestra en la Figura 38.



Figura 38: Configuración de Host e IP

- Seguidamente también, debemos registrar el alias para la zona configurada. El alias representa la palabra que antecede al dominio (indesic.org), es decir www; es un alias, mail; también puede ser un alias, y de esta forma configuramos los alias que queramos.



Figura 39: Configuración de Alias

Finalmente, la configuración del dominio aparecerá como en la Figura 40, que para este caso, se ha registrado el dominio *indesic.org*, con el host *curso* y con los alias: *www*, *mail*, *learning* e *intranet*.

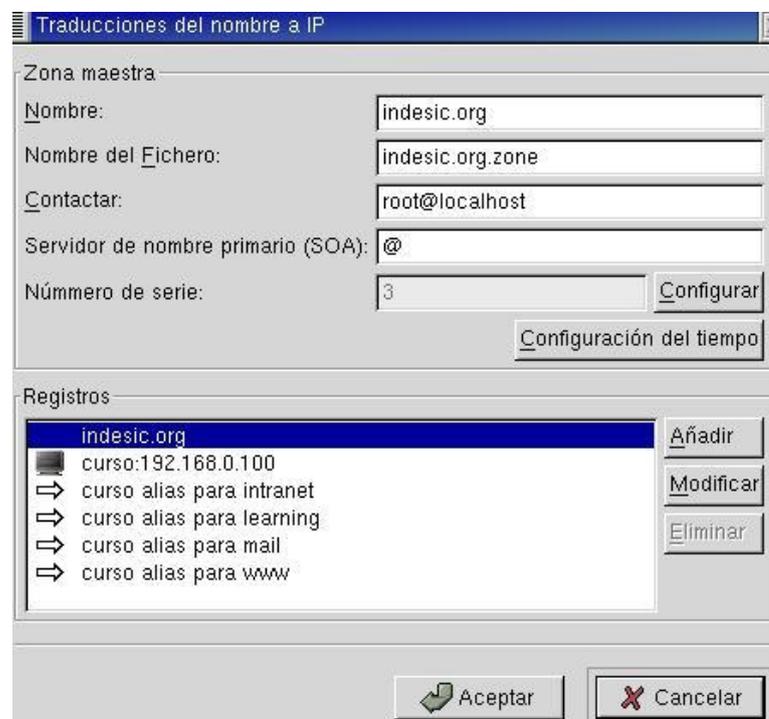


Figura 40: Ventana de Traducciones del Nombre al IP

Verificación de la configuración

Para verificar si el DNS está correctamente configurado, basta con hacer ping a la dirección del dominio desde la ventana de comandos, para este caso hacemos lo siguiente: ping www.indesic.org y debe darnos la respuesta.

Otra forma de verificar es mediante un explorador de Internet y escribir la dirección del dominio en la barra de dirección. Pero antes de esto se debe iniciar el servidor web, este proceso se explica en el siguiente numeral (Iniciar el Servidor Web Apache).

En este caso, para probar la configuración del DNS, se ha usado el explorador de Internet Mozilla, y el resultado se muestra en la Figura 41.



Figura 41: Prueba de configuración DNS

INICIAR EL SERVIDOR WEB APACHE

En modo gráfico, la forma de iniciar el servidor web apache es muy sencillo:

- Vamos hasta el *Menú Inicio*, apuntamos a *Programas*, luego a *Sistema* y finalmente hacemos clic en *Service Configuration*
- Entonces aparecerá la ventana de Configuración de Servicios, buscamos *httpd* y seleccionamos la caja de verificación, y hacemos clic en *Comenzar*, y finalmente hacemos clic en *Guardar*.

El Servidor web Apache viene con los instaladores del Linux, basta con seleccionar esta aplicación al momento de la instalación y luego levantar el servicio para poder usarlo.

La Figura 42, muestra un ejemplo de la forma de iniciar el servidor web Apache en el modo gráfico.

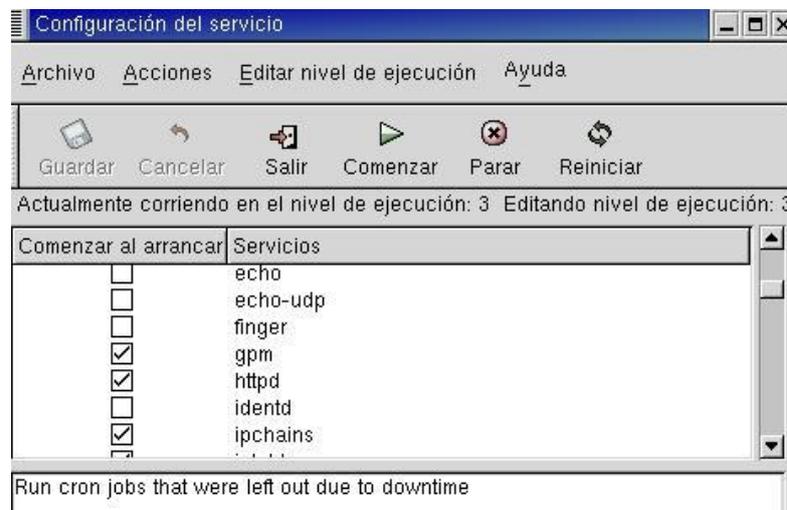


Figura 42: Inicio de Servidor Web Apache

ANEXO G: PLAN DE PRUEBAS

Introducción y resumen de elementos y características a probar

Este plan de prueba se ha desarrollado para probar la aplicación Web para la Gestión y Planificación de cursos de capacitación de cursos de capacitación en línea, desarrollado para el INDESIC.

Estas pruebas se realizarán para verificar si el producto cumple con los requisitos del cliente y según lo descrito en los casos de uso.

Elementos software a probar

La aplicación se ha realizado tomando en cuenta los casos de uso, por lo tanto las pruebas se realizarán por cada caso de uso que se haya identificado en el análisis y diseño. Los casos de uso aprobar son:

- Crear usuarios del sistema
- Registro de Becarios
- Publicar cursos
- Crear Preguntas
- Crear Respuestas
- Inscripción
- Tomar cursos
- Rendir Examen
- Consultar Nota

Características a probar

- Que los datos ingresados se guarden correctamente en la base de datos.
- Manejo de los mensajes de error, cuando éstos se produzcan

- Las consultas y reportes sean los correctos
- Control de claves primarias duplicadas
- La presentación de resultados en la ventana
- La aplicación será probada usando el navegador Mozilla FireFox.

Características que no se probarán

No se probará este sistema en la plataforma Linux, en este caso se probará en Windows. Cuando la aplicación este probada y lista, se subirá a dicha plataforma.

La aplicación no se probará con Internet Explorer, ya que las pruebas del desarrollo se han realizado con ese navegador y no ha presentado inconvenientes.

Enfoque general de la prueba

Se ha decidido probar la aplicación según el tipo de prueba modular, es decir que se probará módulo por módulo, ya que la aplicación se ha desarrollado en base a módulos y es mejor que se siga el mismo procedimiento de la construcción.

En lo que respecta al método para las pruebas, se ha decidido usar el método de caja negra, es decir, se probará con datos de ingreso para verificar los resultados en los datos de salida. Para este caso, se ha elegido esta metodología, ya que nos interesa verificar si la aplicación cumple con los requerimientos del usuario y por lo tanto, se debe ingresar datos de entrada y obtener datos de salida para la verificación.

Criterios de paso/fallo para cada elemento

Se debe definir los datos que e ingresarán para cada uno de los módulos y la respuesta esperada, en caso de cumplirse, se describirá como prueba exitosa, caso contrario, se especificará como errónea y se hará la sugerencia de lo que hay que arreglar.

Criterios de suspensión y requisitos de reanudación

Se suspenderá las pruebas en caso de que la aplicación presente algún error grave que impida seguir probando o que se produzca un resultado completamente fuera de lo requerido.

Para reanudar las pruebas será necesario que se corrija la falla que impidió continuar con las mismas.

Documentos a entregar

El detalle de las pruebas de cada modulo se insertará en la sección de Ejecución de Pruebas, en el capítulo VI de la Implementación. Ahí, se describe los resultados de cada módulo.

Actividades de preparación y ejecución de pruebas

Antes de la ejecución de las pruebas es necesario realizar las siguientes actividades:

- Crear un formato de pruebas, en el que se describa los datos de entrada, de salida y los resultados obtenidos.

- Instalar las herramientas necesarias para el funcionamiento y las pruebas de la aplicación.
- Dejar la base de datos en blanco, de modo que se pueda iniciar desde cero.
- Dar inicio a los servicios necesarios para correr la aplicación

En el proceso de la ejecución de las pruebas es necesario realizar las siguientes actividades:

- Describir en el formato de pruebas, las entradas que tendrá cada caso de uso.
- Correr la aplicación e ingresar datos según se haya descrito en el formato de pruebas.
- Describir las salidas para cada caso de uso probado.
- Describir los resultados y las sugerencias para cada caso de prueba.

Necesidades de entorno

Se requiere que las herramientas necesarias (como el Tomcat, MySql, J2SDK) estén configuradas correctamente antes de iniciar las pruebas

Las pruebas se realizarán en entorno de desarrollo (Windows), es decir antes de la implantación en entorno de producción

Responsabilidades en la organización y realización de las pruebas

No se aplica para este proyecto.

Necesidades de personal y formación

No se aplica para este proyecto.

Esquema de tiempos

Para probar cada módulo se ha considerado un tiempo aproximado de 20 minutos mínimo y un máximo de 25 minutos.

La aplicación esta compuesta de 14 módulos, por lo que se estima que la prueba llevará un tiempo máximo de 6 horas probando la aplicación.