



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces.

Monografía previa a la obtención del título de Ingeniera en Sistemas y Telemática

Autor:

Jenny Susana Lazo Cajilima

Director: Ing. Marcos Orellana

Cuenca, Ecuador

2015

DEDICATORIA

Esta monografía está dedicada con todo mi cariño y aprecio a mis padres que gracias a su apoyo y confianza incondicional pude cumplir mis objetivos como persona, estudiante y ahora como profesional.

A mis seres queridos que de una u otra manera me incentivaron para culminar este trabajo y no decaer.

AGRADECIMIENTOS

A mis padres, que gracias a su esfuerzo y trabajo me ayudaron a ser mejor y a superar los obstáculos de cada día.

A la Universidad del Azuay, a todos los docentes de la escuela de Ingeniería en Sistemas y Telemática por haber impartido conocimientos y haber formado en mí una persona profesional.

A ustedes por acompañarme en este trayecto.

Contenido

Introducción	9
Resumen	10
Abstract	11
1. Conceptos básicos	12
1.1 ¿Qué es Java?	12
1.2 Hibernate	12
1.3 Tecnología JSF	13
1.3.1 Java Server Faces (JSF)	13
1.3.2 Beneficios de la Tecnología Java Server Faces	14
1.3.3 ¿Qué es una Aplicación <i>Java Server Faces</i> ?	15
1.4 Comparativa de componentes para implementación	16
1.5 Componentes PrimeFaces	17
1.6 Gestor de base de datos PostgreSQL	17
Características	18
1.7 Servidor de aplicaciones JBoss	18
1.7.1 ¿Qué es JBoss?	19
1.7.2 ¿Para qué sirve?	19
1.7.3 Características	19
2. Diseño y Modelado del sistema de inscripción	20
2.1 Diagramas de Clase	20
2.2 Diagrama de Casos de Uso	24
2.2.1 Descripción de casos de uso:	25
2.3 Modelo Entidad-Relación	26
2.4 Diccionario de datos	28
3. Desarrollo del aplicativo	31
3.1 Creación de un proyecto JSF	31
3.2 Mapeo de Entidades	32
3.3 Creación de ManagedBean	37
3.4 Implementación de componentes PrimeFaces	37
4. Validación y Pruebas	42

4.1	Configuración del servidor de aplicaciones	42
4.1.1	Configuración del archivo de conexión de base de datos.....	43
4.2	Pruebas del aplicativo	43
	CONCLUSIONES Y RECOMENDACIONES	47
	APENDICES.....	48
	Manual de Usuario.....	48
	Datos Personales	49
	Datos del Colegio.....	51
	Calificaciones.....	51
	Carrera a estudiar.....	52
	Glosario	54
	BIBLIOGRAFIA	55

Índice de Tablas

Tabla 1: Ingresar depósito	25
Tabla 2: Registrar información	25
Tabla 3: Guardar información.....	25
Tabla 4: Modificar información	26
Tabla 5: Salir	26
Tabla 6: Diccionario alumno	28
Tabla 7: Diccionario tipo de identificación	29
Tabla 8: Diccionario estado civil	29
Tabla 9: Diccionario genero	29
Tabla 10: Diccionario parroquia	29
Tabla 11: Diccionario colegio.....	29
Tabla 12: Diccionario carrera.....	30
Tabla 13: Diccionario facultad	30
Tabla 14: Diccionario teléfono.....	30

Índice de Imágenes

Figura 1- modelo en capas.....	13
Figura 2- empresas jsf.....	16
Figura 3– Comparación tecnologías	16
Figura 4 -NuevoAlumno.....	20
Figura 5 – Tipoidentificacion	21
Figura 6 - Genero	21
Figura 7 - EstadoCivil	21
Figura 8 - Parroquia	21
Figura 9 - Colegio	21
Figura 10 - Teléfono.....	21
Figura 11 - Carrera	22
Figura 12 - Facultad	22
Figura 13 - Depósito.....	22
Figura 14: Diagrama de clases	23
Figura 15 – Diagrama de Casos de uso	24
Figura 16 – Modelo Entidad-Relación	27
Figura 17 – Nuevo proyecto	31
Figura 18 – Proyecto.....	32
Figura 19 - Conexión	34
Figura 20 - Tablas.....	34
Figura 21 - Relaciones.....	35
Figura 22 - Entidad.....	35
Figura 23 – Llaves foráneas	36
Figura 24 - Persistencia.....	36
Figura 25 – ManageBean	37
Figura 26 - PáginaPrime.....	38
Figura 27 – Librería Prime.....	38
Figura 28 – Build	39
Figura 29 – Página xhtml	39
Figura 30 – Componente Prime.....	40
Figura 31 – PáginaPrime Temas	40
Figura 32 – PáginaPrime Temas 2.....	41
Figura 33 – Tema Prime.....	41
Figura 34 - comandos	42
Figura 35 – Jboss.....	42
Figura 36 – Jboss configuración.....	43
Figura 37 – Servidor JBoss	44
Figura 38 – Jboss nuevo.....	44

Figura 39 – Configuración servidor.....	45
Figura 40 – página principal.....	45
Figura 41 – Aplicación.....	46

Introducción

Los lenguajes de programación usados para aplicaciones web son muy variados y extensos, lo que ha generado una gran cantidad de tecnologías de desarrollo que facilita la creación de un proyecto web, destacando como principales java server faces (JSF), PHP y .NET

Al ser JSF una herramienta que recién está surgiendo en el mercado, no existe la información necesaria sobre la implementación de un proyecto empleando esta tecnología, y menos enfocándose específicamente al uso de los componentes PrimeFaces

La información que se encuentra es superficial y no orientado a un proyecto integral además los tutoriales comunes no se encuentran orientados a problemas del mundo real esto causa que muchos desarrolladores opten por no utilizar esta herramienta, cuando en realidad es de fácil implementación y puede adaptarse a su proyecto sin importar su tamaño y alcance.

Resumen

El diseño y la implementación de un proyecto web se torna complicado, si no existe la información suficiente para realizarlo y más aún si la herramienta que se utiliza es nueva, es por ello, que con el presente trabajo se pretende explicar de forma sencilla y rápida la implementación de un proyecto web, detallando cada uno de los pasos a seguir, mediante el ejemplo aplicado a un sistema de inscripción para nuevos alumnos de la Universidad del Azuay, por medio de esto se demostrará la forma de llevar a cabo la implementación del proyecto, utilizando la tecnología JSF en combinación con componentes Prime Faces.

Finalmente se obtendrá un prototipo para la inscripción de nuevos alumnos de la Universidad del Azuay y una demostración sobre la funcionalidad de Prime Faces y cada uno de sus componentes, que servirá de guía para implementaciones en proyectos futuros.

Abstract

The design and implementation of a web project becomes complicated if there is not enough information on how to do it, and even more if the tool used is new. Hence, this paper explains in a simple and quick manner the implementation of a web project, detailing each step through an example applied to the new students' registration system at *Universidad del Azuay*. In this way, we will demonstrate how to carry out the implementation of a project using JSF technology in combination with Prime Faces components.

Finally, a prototype for the registration of new students at *Universidad del Azuay* will be obtained, and the demonstration of the functionality of Prime Faces and each of its components, which will guide the implementation of future projects will be presented.



Translated by,
Lic. Lourdes Crespo

1. Conceptos básicos

1.1 ¿Qué es Java?

Java es un lenguaje de programación orientado a objetos, fue comercializado por primera vez por *Sun Microsystems* en 1995. Hoy en día, abarca una gran cantidad de mercado, ya que existen varios sitios web que requieren java para su funcionamiento. “Java está en todas partes, lo utilizan desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet” (oracle, Oracle corporation us, 2014)

Entre las grandes ventajas de este lenguaje están:

- Un mismo programa puede ser implementado en diversos sistemas operativos, multiplataforma.
- Posibilidad de conectarse a cualquier base de datos empleando *applets* insertados en páginas web o mediante páginas jsp con código HTML dinámico.
- Es un lenguaje sencillo y cada vez van surgiendo nuevas características del lenguaje que las podemos utilizar por medio de APIs.
- Y la principal es que es un lenguaje libre, no requiere licenciamiento.

Hoy en día Java ha demostrado ser una plataforma capaz de soportar sistemas críticos, flexibles, confiables y robustos, es uno de los principales retos al momento de desarrollar software.

Java como lenguaje de programación tiene innumerables características, pero en este caso la investigación se centrará en la parte orientada a desarrollo web, puesto que Java es un pilar del desarrollo Web y últimamente cuenta con una gran cantidad de herramientas que facilitan y optimizan la creación de proyectos web.

1.2 Hibernate

Hibernate es una herramienta muy utilizada en Java para mapear un objeto relacional, permite generar clases con persistencia partiendo de clases comunes, también incluye herencia, asociación composición, etc.

El lenguaje que emplea Hibernate para sus consultas es HQL (*Hibernate Query Language*), se basa en SQL y, a pesar de tener su propio lenguaje, también permite realizar consultas nativas utilizando SQL y su integración con aplicaciones J2EE es simple.

1.3 Tecnología JSF

Desde la aparición del Internet, fueron surgiendo diferentes requerimientos por parte de los usuarios y a partir de esto varias tecnologías fueron apareciendo y desarrollándose para cumplir con todas estas demandas. Estas tecnologías fueron incorporando varias características entre ellas la interacción con el usuario y la utilización de bases de datos.

Entre las más conocidas está *Java Server Faces* (JSF) que actualmente está tomado ventaja ante las demás.

1.3.1 Java Server Faces (JSF)

JSF (*Java Server Faces*) es una tecnología para el desarrollo de aplicaciones web basadas en java, JSF surge después de *Struts* que es el que se basa en el modelo vista controlador (MVC), por lo tanto hereda todas sus características y las mejora, con esto se quiere estandarizar el desarrollo de aplicaciones web, uno de los grandes beneficios y la mayor ventaja de emplear la tecnología Java Server Faces es que en las aplicaciones web se mantiene el comportamiento y la parte de la presentación claramente separados. (Parkway, 2013)

Al utilizar un modelo de desarrollo compacto se presentan diversos problemas para integrar sistemas, como: escalabilidad, seguridad, disponibilidad, etc. A raíz de estos inconvenientes, surge un nuevo modelo de desarrollo de aplicaciones, que es, en capas; la primera capa sirve para el manejo de base de datos, la segunda para centralizar la lógica del sistema y finalmente la tercera capa es la que interactúa con el usuario puesto que es la parte gráfica.



Figura 1- Modelo en Capas (Global Mentoring, 2012)

La tecnología JSF utiliza el Modelo Vista Controlador, esta arquitectura emplea la tecnología en capas separando claramente la aplicación en tres partes:

- El modelo, es toda la parte lógica de la aplicación
- La vista, comprende la parte gráfica que es la que el usuario puede visualizar.
- El controlador, que recibe e interpreta la interacción del usuario, el controlador actúa sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

Esta arquitectura ha demostrado ser muy apropiada y se adapta correctamente para las aplicaciones web.

Para el desarrollo de las páginas se utiliza JSP (*Java Server Pages*), que son etiquetas personalizadas, para el manejo de eventos, validación de datos de entrada, accesibilidad, navegación, etc., cuenta con un conjunto de APIs, además, el usuario se ve obligado a separar su código java de su página xhtml puesto que en la página no se puede agregar sentencias java (Fain, 2011)

Java Server Pages (JSP) permite a los desarrolladores mantener las páginas Web independientes para que se pueda cambiar el diseño sin cambiar el contenido. Las etiquetas son similares a XML que permiten encapsular la lógica de la página.

1.3.2 Beneficios de la Tecnología Java Server Faces

- Separa la lógica de la presentación, se puede establecer grupos de desarrollo permitiendo a cada uno centrarse en su proceso. Esto la convierte en una plataforma flexible y de fácil uso que no limita al desarrollador ya que se puede encapsular la funcionalidad de los componentes y ser presentados de diversas formas.
- Los APIs de la tecnología *Java Server Faces* se han creado directamente sobre el API *Java Servlet*. Esto permite hacer algunas tareas: utilizar diversas tecnologías de presentación, desarrollar componentes personalizados y obtener diversas salidas a los dispositivos.
- La tecnología *Java Server Faces* cuenta con una gran arquitectura que controla eventos y estados de los componentes, validar información y procesar datos.
- Otra ventaja de *Java Server Faces* es que los componentes UI de la página están representados en el servidor como objetos con estado, esto permite que el estado del componente se pueda modificar y a su vez conectar el código con los eventos que se generen por el usuario.

- Finalmente, con la tecnología *Java Server Faces* se puede validar cualquier error directamente sobre el componente sin que sea necesario el envío de los datos al servidor.

1.3.3 ¿Qué es una Aplicación *Java Server Faces*?

Una aplicación *Java Server Faces* es muy similar a otra aplicación Web por lo tanto entre sus componentes principales están:

- La parte de funcionalidad y datos van a estar en los componentes *JavaBeans*.
- Las Páginas van a ser almacenadas como páginas JSP.
- No puede faltar la parte del procesador de eventos que son los “Oyentes de Eventos”.
- Si necesitamos trabajar con bases de datos se requiere la creación de *Beans*.
- Los componentes UI pueden ser dibujados en la página por medio de una librería de etiquetas personalizadas.
- Se requiere el uso de una librería de etiquetas personalizadas para capturar las acciones, eventos, navegación y diferentes validaciones de la página.

JSF es una especificación, por ello surge la necesidad de componentes especializados, existen varias empresas que han construido su set de componentes en base a esta especificación.

Cuando se inicia un proyecto real, no se debe quedar con la especificación de JSF sino necesariamente deberá elegirse una implementación, con una implementación se cuenta con un set de componentes más amplio, versátil y además un mejor manejo del entorno visual del aplicativo.

1.4 Comparativa de componentes para implementación



Figura 2 – Empresas JSF

Las empresas que proporcionan los componentes para implementar en un proyecto JSF son “PrimeFaces”, “IceFaces”, “RichFaces” y estos son los principales aspectos para comparación.

- Componentes: RichFaces cuenta con 39 componentes repartidos en tablas, paneles, menús, etc., IceFaces tiene alrededor de 70 componentes mientras que PrimeFaces tiene un conjunto de 117 componentes.
- Inicio: RichFaces no tiene un tutorial de inicio, IceFaces cuenta con una guía de inicio rápido mientras que *PrimeFaces* tampoco tiene una página de introducción pero lo único que se necesita es descargarse el .jar, agregarle al proyecto y listo
- Documentación: La guía de usuario de RichFaces es muy básica y no existe suficiente información para implementar estos componentes en la web; en cuenta a la documentación de IceFaces es muy variada y extensa con ejemplos y tutoriales; *PrimeFaces* tiene una guía de usuario completa a más de ofrecer foros de ayuda.
- Rendimiento: Se ha realizado mediciones de rendimiento, quedando como mejor *PrimeFaces*, luego IceFaces y por último RichFaces.

	Richfaces	Icefaces	Primefaces
Longitud de la documentación	47442 bytes	76072 bytes	53904 bytes
Solicitudes por segundo	28.59 [# /sec]	11.12 [# /sec]	33.46 [# /sec]
Tiempo por petición	174.869 [ms]	899.281 [ms]	149.422 [ms]
Velocidad de transferencia	1329.59 [Kbytes/sec]	829.41 [Kbytes/sec]	1767.06 [Kbytes/sec]
Tiempos de conexión (ms)	766	12969	1094

Fuente: <http://www.javahispano.org>

Figura 3 – Comparación tecnologías

Con los resultados obtenidos se considera la utilización de *PrimeFaces* debido a su mejor desempeño frente a otros.

1.5 Componentes PrimeFaces

PrimeFaces es una implementación de la tecnología JSF, se puede obtener mediante la descarga de una librería que proporciona una gran variedad de componentes para su uso en proyectos web. La gran ventaja con la que cuenta PrimeFaces, es que no limita el uso exclusivo de estos componentes, sino que interactúan normalmente con otros, “PrimeFaces es un marco de código abierto muy popular para JavaServer Faces con más de 100 componentes”. (Take, PrimeFaces showcase, 2015)

Los componentes de esta librería son muy amigables y tienen un diseño que es agradable para los usuarios finales, además permite la implementación de hojas de estilo adaptándose a los requerimientos de programador.

Una desventaja es que las versiones de las nuevas librerías no son compatibles con las anteriores, es por ello que si el proyecto se inicia con una versión se recomienda terminarlo con la misma versión y no migrar de librería porque implicaría varios cambios para la integración.

1.6 Gestor de base de datos PostgreSQL

PostgreSQL es un potente gestor de bases de datos relacionales con una arquitectura orientada a objetos, trabaja con código abierto es decir cualquier persona puede acceder a su código fuente, fue publicado bajo la licencia BSD y es dirigido por una comunidad de desarrolladores denominada PGDG (*PostgreSQL Global Development Group*).

Por ahora se encuentra en producción la versión 9.4.2 cuyas características la convierten en una base de datos robusta capaz de competir con el resto de base de datos comerciales, puesto que soporta diversos tipos de datos, maneja herencia, permite realizar copias de seguridad en caliente, entre otras.

PostgreSQL está orientado a trabajar con gran cantidad de información y con una concurrencia elevada de usuarios accediendo al mismo tiempo, ya que se ha enfocado en su estabilidad, robustez, potencia y facilidad de administración e implementación de estándares, su desde que inició su desarrollo, 16 años atrás. (rafaelma, 2010)

Una falla en alguno de los procesos que se encuentre ejecutándose en PostgreSQL no afecta al resto y seguirán funcionando correctamente gracias a su estabilidad por medio del uso de multiprocesos.

Características

- Alta concurrencia, no bloquea la tabla si un proceso está ejecutándose en la misma, lo que permite que otros usuarios puedan acceder al mismo tiempo.
- Integridad referencial.
- Amplia variedad de tipos nativos de datos: números, texto, figuras geométricas, arrays y también se incluye datos GIS creados por el proyecto PostGIS.
- Tablespace.
- Llaves foráneas.
- Triggers: es una acción específica que se realiza cuando ocurre un evento dentro de la base de datos.
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Replicación asincrónica/sincrónica.
- Copias de seguridad en caliente.
- Regionalización por columna.
- Acceso encriptado vía SSL.
- Completa documentación.
- Disponible para Linux y UNIX y Windows 32/64bit.

1.7 Servidor de aplicaciones JBoss

Un servidor web tiene una estructura de dos capas a diferencia de un servidor de aplicaciones que cuenta con tres capas permitiendo una estructura eficiente en todo sistema especialmente en aplicaciones complejas.

Un servidor de aplicaciones soporta todos los componentes de la arquitectura JEE6 a más de la parte web, es el corazón de un sistema distribuido, puesto que para un sistema distribuido se requiere un servidor de aplicaciones que soporte la ejecución y disponibilidad de la aplicación así como la relación con la base de datos entre otros

Existen varios servidores web tanto *Open Source* como de propietarios, entre los cuales podemos mencionar: “JBoss”, “Apache Geronimo 2.0”, “JOnAS”, “GlassFish”, “Oracle WebLogic Application server”, “IBM WebSphere Application Server”.

1.7.1 ¿Qué es JBoss?

Es un servidor de aplicaciones de rendimiento alto, basado en *J2EE Open Source* desarrollado por *Red Hat* 2006, implementado 100% en Java, gracias a esto se lo puede implementar en cualquier plataforma que tenga instalado un JDK, se lo puede descargar de manera gratuita brindando la posibilidad de modificar y distribuir sin restricción alguna, es muy flexible, potente y consistente ya que cumple con todos los estándares y cuenta con una red de colaboradores a nivel mundial.

1.7.2 ¿Para qué sirve?

JBoss Enterprise es un servidor de aplicaciones que permite el desarrollo, la implementación y toda la gestión de las aplicaciones de una manera sencilla. Éste es un servidor de código abierto con avanzada tecnología para alojar las aplicaciones; utiliza una arquitectura que se orienta a los servicios. (Red Hat, 2015)

Facilita el desarrollo de Aplicaciones web con base de datos y se adapta a aplicaciones web que van desde las básicas, intermedias e incluso complejas.

1.7.3 Características

- Sin costo, código abierto.
- Confiable ya que cumple con los estándares.
- Orientado a arquitectura de servicios.
- Flexible
- Servicios del middleware para cualquier objeto de Java
- Ofrece persistencia objeto/relación y consultas para Java
- Mantiene en caché los objetos Java que son accedidos con mayor frecuencia
- Cuenta con una plataforma para los portales web

2. Diseño y Modelado del sistema de inscripción

A través del diseño se pretende especificar todos los detalles de la aplicación, lo que permitirá interpretarla, quedando reflejados todos los requerimientos del sistema, su funcionalidad y comportamiento durante la implementación, definiendo componentes, datos y módulos

2.1 Diagramas de Clase

El diagrama de clases permite visualizar las diferentes clases que conforman el sistema describiendo sus atributos y métodos y sus respectivas relaciones. Al momento de utilizar la arquitectura Modelo Vista Controlador el diagrama de clases servirá como base.

En el sistema para la inscripción de nuevos alumnos, se tiene las siguientes clases: "Nuevo alumno", "Tipo de identificación", "Género", "Estado Civil", "Parroquia", "Colegio", "Teléfono", "Carrera", "Facultad", "Depósito".

Nuevo Alumno

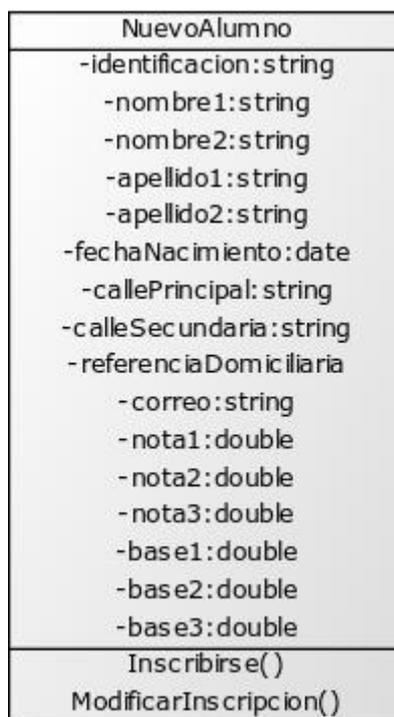


Figura 4 -NuevoAlumno

Tipo de Identificación

TipoIdentificacion
-codigo: integer
-descripcion: string

Figura 5 – TipoIdentificacion

Género

Genero
-codigo: integer
-descripcion: string

Figura 6 - Genero

Estado Civil

EstadoCivil
-codigo: integer
-descripcion: string

Figura 7 - EstadoCivil

Parroquia

Parroquia
-codigo: integer
-nombre: string

Figura 8 - Parroquia

Colegio

Colegio
-codigo: integer
-nombre: string

Figura 9 - Colegio

Teléfono

Telefono
-numero: string
-tipo: string

Figura 10 - Teléfono

Carrera

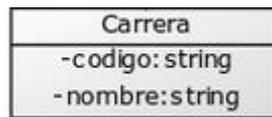


Figura 11 - Carrera

Facultad

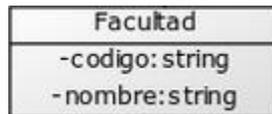


Figura 12 - Facultad

Depósito



Figura 13 - Depósito

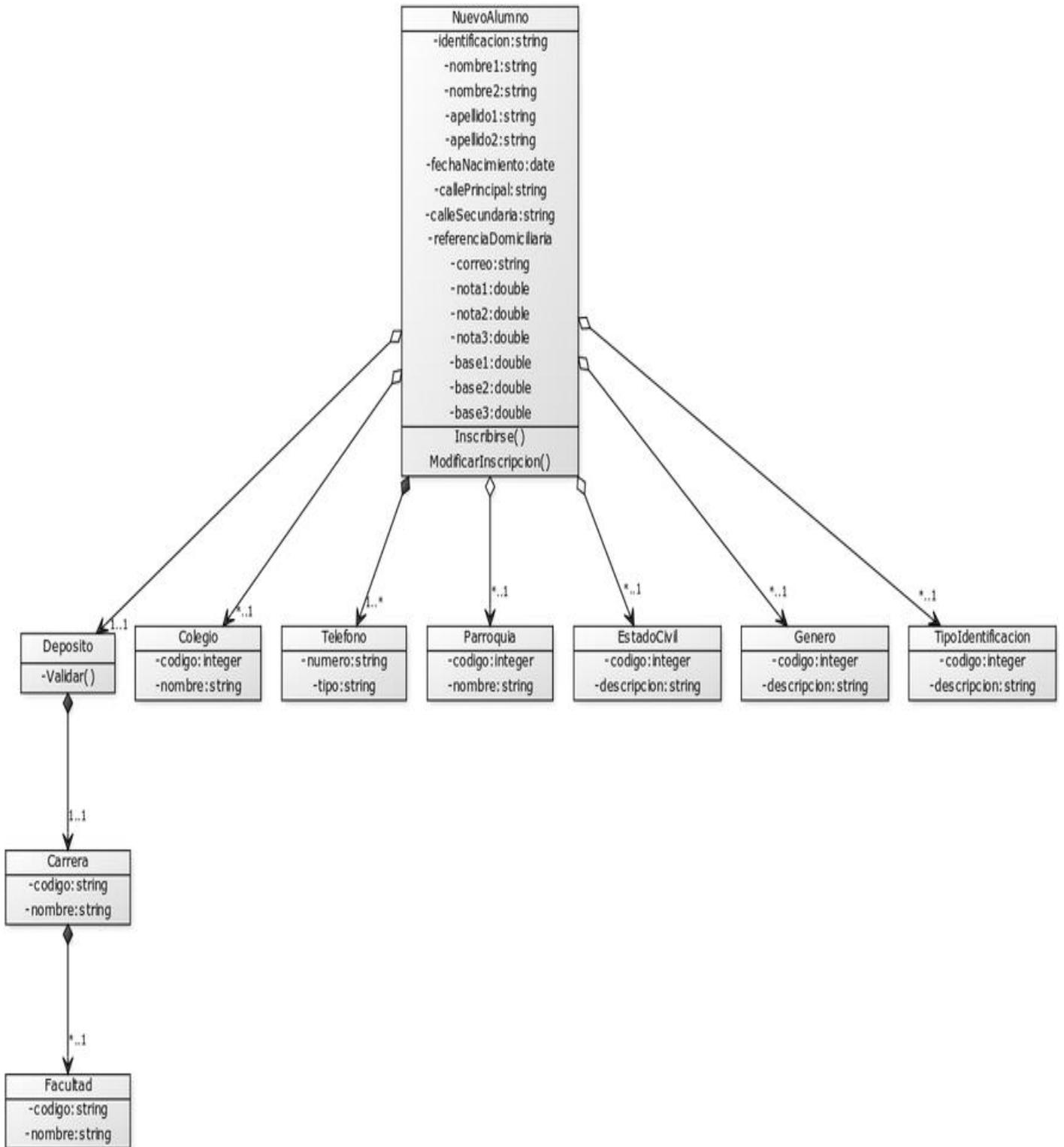


Figura 14: Diagrama de clases

2.2 Diagrama de Casos de Uso

Este diagrama permite representar la manera en la que un actor opera con el sistema y cómo interactúan los elementos entre sí. El siguiente gráfico indica el actor y cada uno de los comportamientos definidos en el sistema:

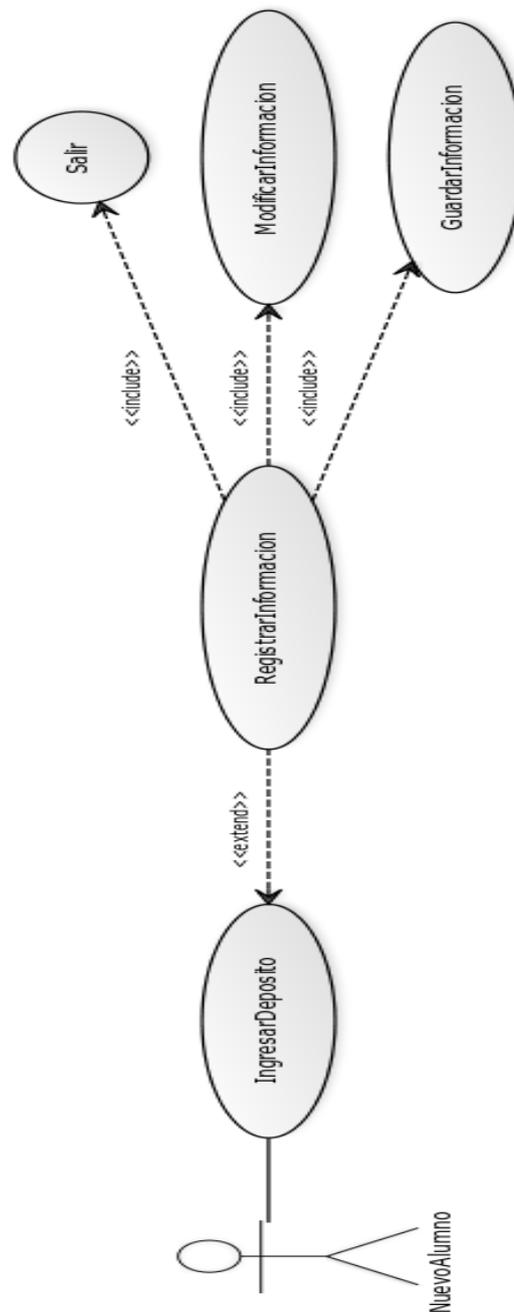


Figura 15 – Diagrama de Casos de uso

2.2.1 Descripción de casos de uso:

CASO 1	Ingresar Deposito
actor:	nuevo alumno
descripción:	el alumno accede a la inscripción registrando su depósito previamente en el banco
prioridad:	Obligatorio
Requisitos Asociados: R1.1 Se podrá acceder al sistema únicamente con el código del deposito R1.2 El sistema validará que el código sea correcto R1.3 El sistema indicará si el código fue aceptado caso contrario mostrará un mensaje de error	

Tabla 1: Ingresar depósito

CASO 2	Registrar Información
actor:	nuevo alumno
descripción:	el alumno ingresa todos los campos que le solicita el formulario
prioridad:	Obligatorio
Requisitos Asociados: R2.1 Únicamente pueden ingresar a este caso el alumno cuyo código de depósito haya sido aceptado R2.2 El sistema indicará si algún campo no se ingresa correctamente con un mensaje de error	

Tabla 2: Registrar información

CASO 3	Guardar Información
actor:	nuevo alumno
descripción:	el alumno guarda la información ingresada
prioridad:	Obligatorio si no existe un alumno registrado
Requisitos Asociados: R3.1 Únicamente pueden ingresar a este caso el alumno cuyo código de depósito haya sido aceptado R3.2 El sistema validará que los campos requeridos estén correctamente ingresados R3.3 El sistema indicará si no se pudieron guardar los datos con un mensaje de error R3.4 Solo se activará este caso si es la primera vez que el alumno registra su información	

Tabla 3: Guardar información

CASO 4	Modificar Información
actor:	nuevo alumno
descripción:	el alumno modifica la información ingresada
prioridad:	Obligatorio si existen datos de un alumno registrado
Requisitos Asociados: R4.1 Únicamente pueden ingresar a este caso el alumno cuyo código de depósito haya sido aceptado R4.2 El sistema validará que los campos requeridos estén correctamente ingresados R4.3 El sistema indicará si no se pudieron guardar los datos con un mensaje de error R4.4 Solo se activará este caso de existir información previamente almacenada del alumno <p style="text-align: center;">Tabla 4: Modificar información</p>	
CASO 5	Salir
actor:	nuevo alumno
descripción:	Cierra la sesión
prioridad:	Obligatorio
Requisitos Asociados: R5.1 El destruye la sesión del alumno	

Tabla 5: Salir

2.3 Modelo Entidad-Relación

Este modelo indica los diferentes tipos de datos que van a ser almacenados por el gestor de base de datos, así como la iteración entre ellos mediante sus atributos y relaciones.

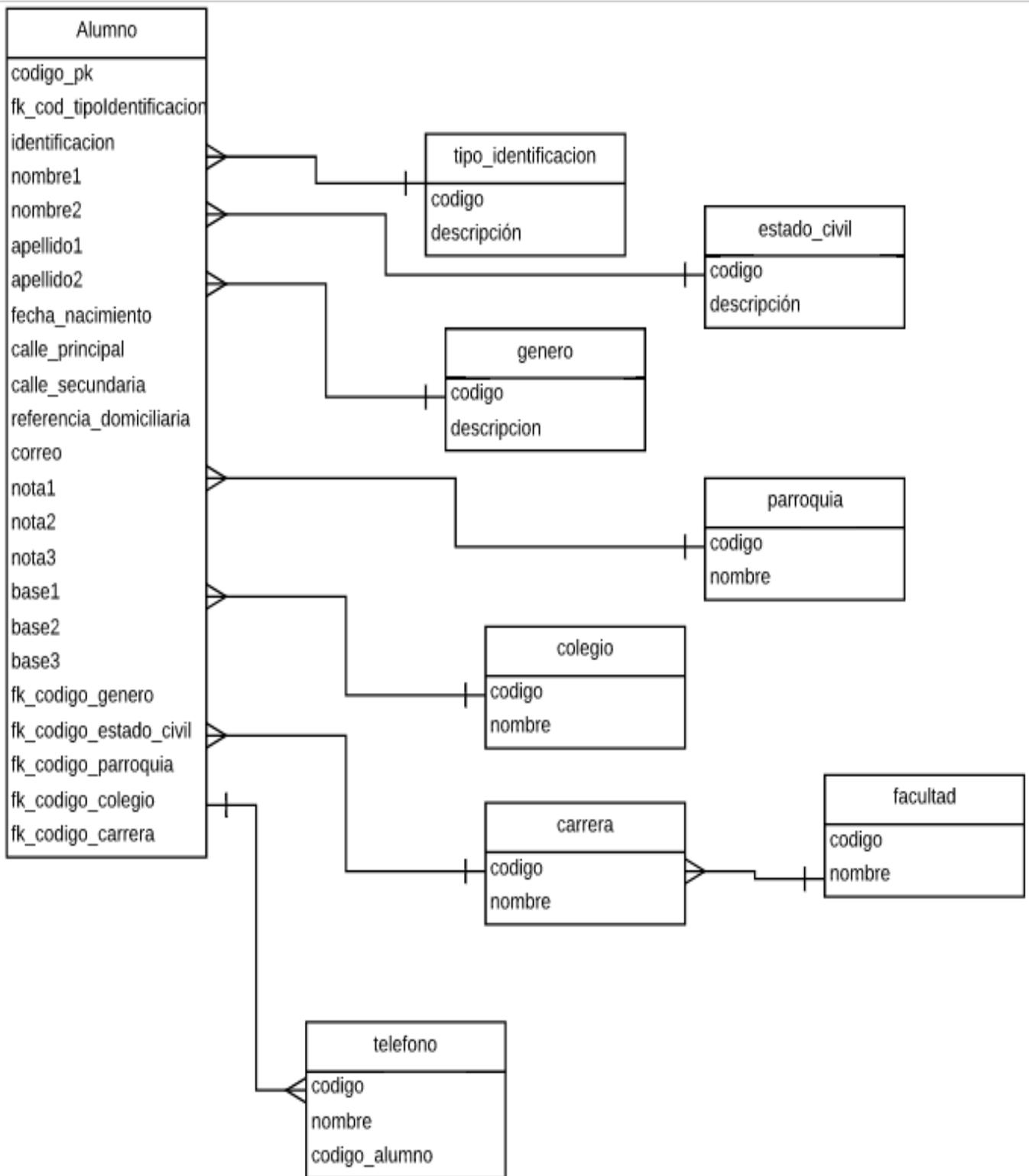


Figura 16 – Modelo Entidad-Relación

2.4 Diccionario de datos

Por medio del diccionario de datos se especifica características de los datos que van a ser almacenados en el sistema, se detalla el nombre de la tabla a la que pertenecen, el nombre del campo, el tipo, el tamaño y una breve descripción.

Entidad:	Alumno		
Descripción:	contiene la información de cada alumno		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
identificacion	String	20	identificacion del alumno
nombre1	String	200	primer nombre del alumno
nombre2	String	200	segundo nombre del alumno
apellido1	String	200	primer apellido del alumno
apellido2	String	200	segundo apellido del alumno
fecha_nacimiento	Date	-	fecha de nacimiento del alumno
calle_principal	String	500	calle principal de la direccion del alumno
calle_secundaria	String	500	calle secundaria de la direccion del alumno
referencia_domiciliaria	String	500	referencia extra de la ubicacion del domicilio del alumno
correo	String	200	correo electrónico del alumno
nota1	Double	-	nota registrada en el primer año de bachillerato del alumno
nota2	Double	-	nota registrada en el segundo año de bachillerato del alumno
nota3	Double	-	nota registrada en el tercer año de bachillerato del alumno
base1	Double	-	base con la que se registró la nota del primer año de bachillerato del alumno
base2	Double	-	base con la que se registró la nota del segundo año de bachillerato del alumno
base	Double	-	base con la que se registró la nota del tercer año de bachillerato del alumno
LLAVE PRIMARIA		RELACIONES	
codigo_pk	fk_cod_tipoidentificacion con la tabla tipo_identificacion		
	fk_codigo_genero con la tabla genero		
	fk_codigo_estadocivil con la tabla estado_civil		
	fk_codigo_parroquia con la tabla parroquia		
	fk_codigo_colegio con la tabla colegio		
	fk_codigo_carrera con la tabla carrera		

Tabla 6: Diccionario Alumno

Entidad:	Tipo_identificacion		
Descripción:	contiene la información de los tipos de identificación		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	código del tipo de identificación
descripcion	String	200	descripción del tipo de identificación

Tabla 7: Diccionario Tipo de Identificación

Entidad:	estado_civil		
Descripción:	contiene la información de los tipos de estados civil que existe		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	código del tipo de estado civil
descripcion	String	200	descripción del estado civil

Tabla 8: Diccionario Estado Civil

Entidad:	genero		
Descripción:	contiene la información de los tipos de género		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	código del género
descripcion	String	200	descripción del género

Tabla 9: Diccionario Género

Entidad:	parroquia		
Descripción:	contiene la información de las parroquias que existen		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	código de la parroquia
nombre	String	200	nombre de la parroquia

Tabla 10: Diccionario Parroquia

Entidad:	colegio		
Descripción:	contiene la información de los colegios		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	código del colegio
nombre	String	200	nombre del colegio

Tabla 11: Diccionario Colegio

Entidad:	carrera		
Descripción:	contiene la información de las carreras de la Universidad del Azuay		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	código de la carrera
nombre	String	200	nombre de la carrera

Tabla 12: Diccionario Carrera

Entidad:	facultad		
Descripción:	contiene la información de las facultades de la Universidad del Azuay		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	código de la facultad
nombre	String	200	nombre de la facultad

Tabla 13: Diccionario Facultad

Entidad:	telefono		
Descripción:	contiene la información de los números telefónicos de los alumnos		
Fecha de creacion:	05/10/2015		
CAMPO	TIPO	TAMAÑO	DESCRIPCION
codigo	String	100	Tipo de teléfono
numero	String	50	Número telefónico

Tabla 14: Diccionario Teléfono

3. Desarrollo del aplicativo

3.1 Creación de un proyecto JSF

Para crear un proyecto JSF, se va a utilizar la herramienta “Eclipse Indigo”, en la página principal se ubica en el menú: “File -> new -> JSF Project”, se ingresa el nombre, el lugar donde se va almacenar, y uno de los campos importantes que se debe tener en cuenta es el ambiente de JSF, se debe seleccionar 2.0 puesto que en versiones anteriores no funcionan los componentes PrimeFaces.

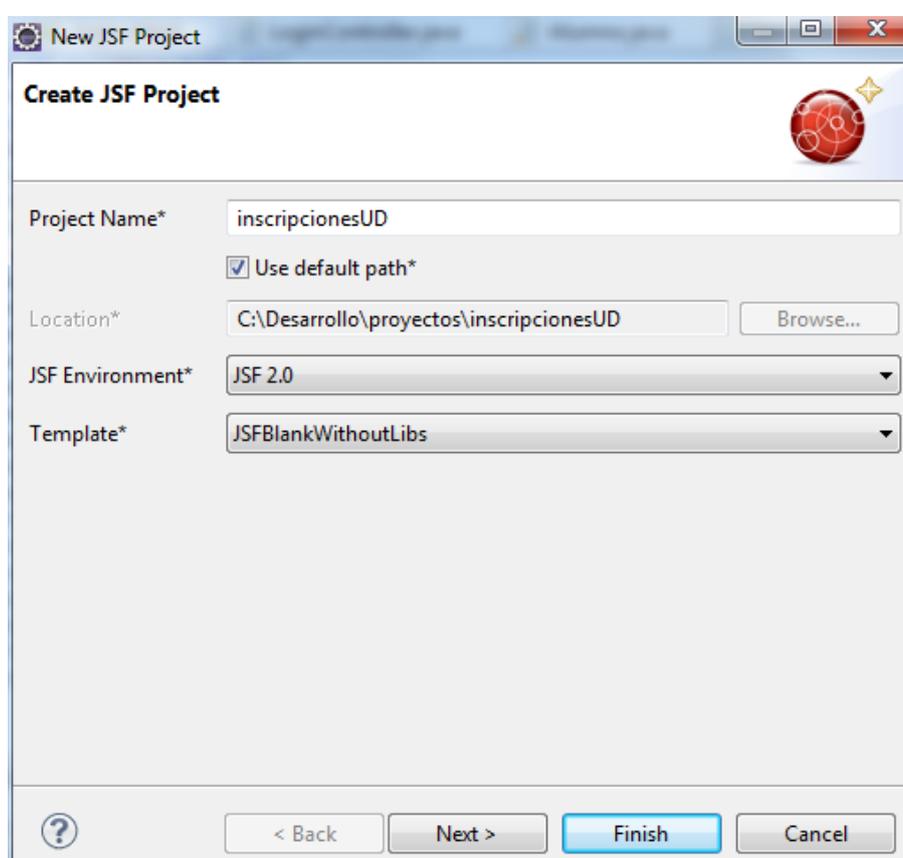


Figura 17 – Nuevo proyecto

En el proyecto se generan automáticamente varias carpetas

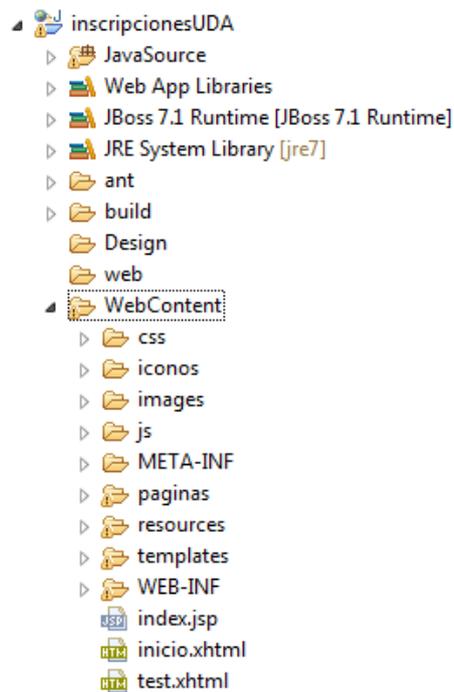


Figura 18 – Proyecto

Es importante entender la utilidad de cada una de las carpetas presentes en el proyecto, a continuación se detalla las importantes:

- “JavaSource”: en esta ubicación se guardarán todas las clases java que generemos, entidades, servicios, controladores, etc.
- “ant”: aquí se encuentran las configuraciones para ejecutar el proyecto, ruta del navegador, ruta del servidor, puerto, etc.
- “WebContent”: en esta carpeta se almacena toda la parte web, las páginas xhtml, hojas de estilo, imágenes, iconos, etc.

3.2 Mapeo de Entidades

Lo que se busca en esta aplicación es configurar la base de datos mediante anotaciones *JPA* y ensamblar correctamente la capa de persistencia *Hibernate*, logrando con ello una aplicación independiente del gestor de base de datos que se adaptará automáticamente a Oracle, SQL Server, Postgres, etc.

Para ello, se genera una entidad por cada tabla que se encuentra creada en la base, una entidad es una clase JAVA pero con la diferencia de que esta clase va a persistir en la base de datos.

Hibernate, en una herramienta muy útil, permite mapear (generar) directamente nuestras entidades a partir de las tablas de un base de datos armada, para ello cuenta con una serie de anotaciones del estándar de JPA que se encuentran en el paquete `javax.persistence`, también, utiliza anotaciones propias que se encuentran en el paquete `org.hibernate.annotations`.

Algunos aspectos importantes que se deben considerar son:

- Todas las clases deben implementar la interfaz serializable.
- Utiliza @Anotaciones para especificar elementos de persistencia.
- Cada atributo con los que cuenta una clase representa a una columna de la tabla.
- La anotación @Entity representa una tabla de la base de datos.
- @Id indica que este atributo es la llave primaria de la tabla.
- @GeneratedValue Indica que el Id es autogenerado y de tipo secuencial, lo que significa que la base de datos asigna un valor incremental en cada nuevo registro por lo tanto no se debe asignar este valor al momento de construir el objeto.
- @NotNull Indica que este campo no puede estar vacío en la tabla.
- @column indica el nombre del campo con la que se relaciona.
- @ManyToOne: Es usado para las relaciones cabecera-detalle, y a través de la anotación JoinColumn especificamos la columna en la tabla detalle que tiene la referencia de la tabla cabecera.
- @OneToMany: Mediante esta anotacion en la tabla cabecera hacemos referencia a los detalles indicando con que atributo de la tabla detalle es mapeado (mappedBy).

Para mapear una base de datos, lo primero que se debe hacer es crear una conexión; esto depende del gestor de base de datos que se utiliza (en este caso Postgres):

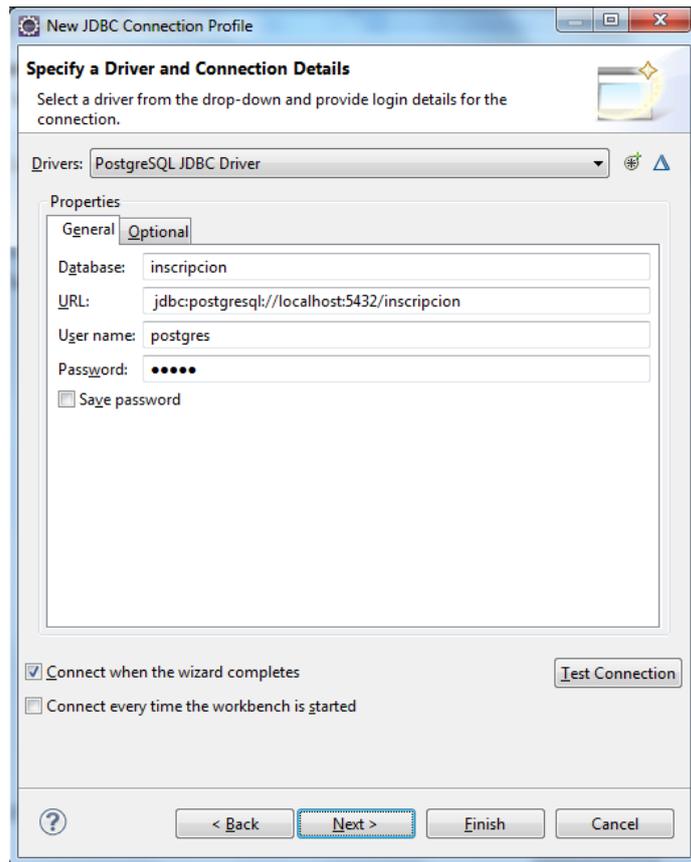


Figura 19 - Conexión

Una vez creada la conexión se selecciona las tablas que a mapear

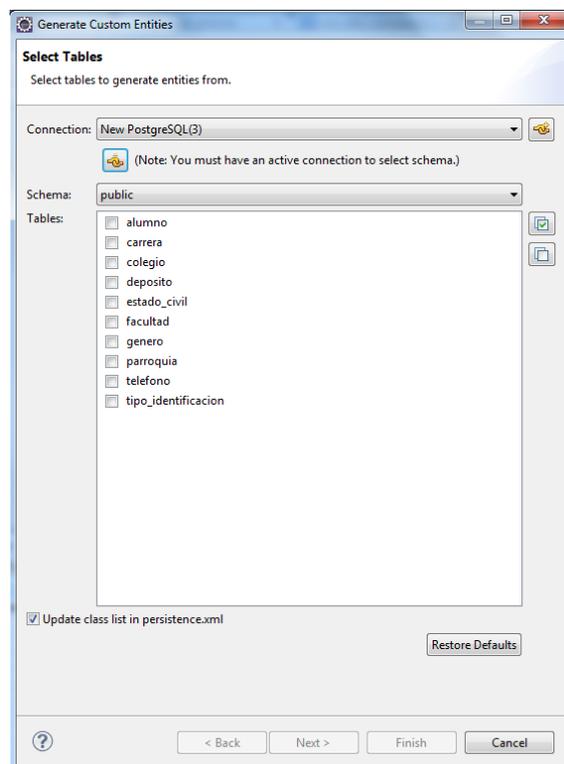


Figura 20 - Tablas

Se tiene la relación entre tablas que se va a generar:

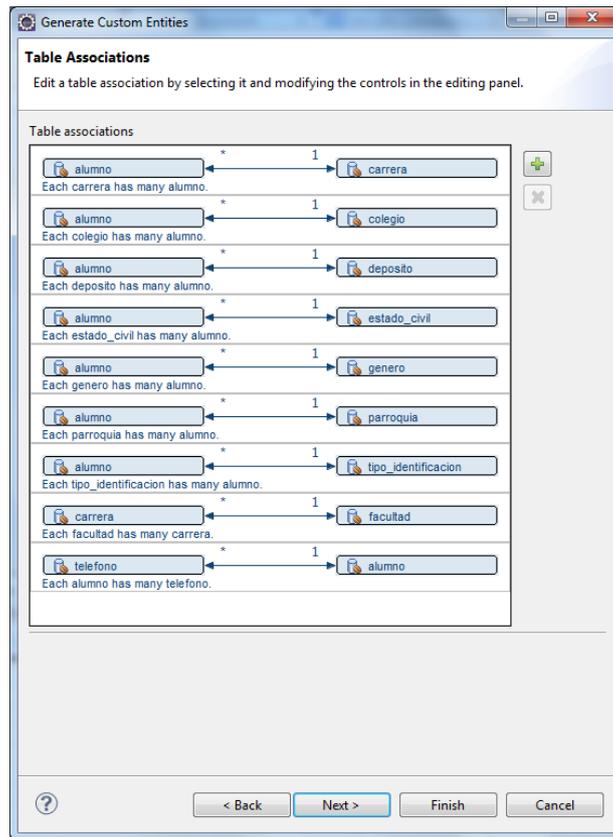


Figura 21 - Relaciones

Una vez especificado todas estas relaciones, se genera automáticamente las entidades:

```
@Entity
public class Alumno extends EntityOC implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="alm_id")
    private Integer id;

    @Column(name="alm_apellido1")
    private String almApellido1;

    @Column(name="alm_apellido2")
    private String almApellido2;

    @Column(name="alm_base1")
    private Double almBase1;

    @Column(name="alm_base2")
    private Double almBase2;

    @Column(name="alm_base3")
    private Double almBase3;

    @Column(name="alm_calleprincipal")
    private String almCalleprincipal;

    @Column(name="alm_callesecundaria")
    private String almCallesecundaria;
}
```

Figura 22 - Entidad

Con sus respectivas relaciones a otras tablas si tienen llaves foráneas:

```
//bi-directional many-to-one association to Carrera
@ManyToOne
@JoinColumn(name="car_id")
private Carrera carrera;

//bi-directional many-to-one association to Colegio
@ManyToOne
@JoinColumn(name="col_id")
private Colegio colegio;

//bi-directional many-to-one association to Deposito
@ManyToOne
@JoinColumn(name="dep_id")
private Deposito deposito;

//bi-directional many-to-one association to EstadoCivil
@ManyToOne
@JoinColumn(name="esc_id")
private EstadoCivil estadoCivil;
```

Figura 23 – Llaves foráneas

Cuando se hayan generado correctamente las entidades se debe configurar el archivo de persistencia, este se encuentra ubicado en de la carpeta META-INF (persistence.xml), se debe incluir todas las entidades junto con su ubicación dentro de este archivo.

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0"
  xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
  <persistence-unit name="inscripcion">
    <jta-data-source>java:/inscripcion</jta-data-source>
    <class>com.ins.entities.Alumno</class>
    <class>com.ins.entities.Carrera</class>
    <class>com.ins.entities.Colegio</class>
    <class>com.ins.entities.EstadoCivil</class>
    <class>com.ins.entities.Facultad</class>
    <class>com.ins.entities.Genero</class>
    <class>com.ins.entities.Parroquia</class>
    <class>com.ins.entities.Telefono</class>
    <class>com.ins.entities.TipoIdentificacion</class>
    <class>com.ins.entities.Deposito</class>
    <class>com.ins.entities.Parametro</class>
    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQLDialect" />
      <property name="hibernate.show_sql" value="false" />
      <property name="hibernate.format_sql" value="true" />
      <property name="hibernate.jdbc.use_streams_for_binary" value="true"/>
    </properties>
  </persistence-unit>
</persistence>
```

Figura 24 - Persistencia

3.3 Creación de ManagedBean

Un ManagedBean es un objeto manejado por un contenedor, esto quiere decir que el programador no se encarga de instanciar el objeto, sino que lo hace el contenedor.

El objetivo de convertir a una simple clase java en un ManagedBean es que se pueda utilizar instancias de esta clase desde la página, el ManagedBean permite pasar información de los objetos de la clase a la página y viceversa.

Para convertir una clase java plana en un ManagedBean, se coloca la etiqueta @ManagedBean, importando el paquete javax.faces.bean. De esta forma ya se puede usar instancias de esta clase desde la página.

```
package com.ins.control.controller;

import javax.annotation.PostConstruct;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.ViewScoped;

import com.ins.commons.controllers.WizardMaintenanceController;
import com.ins.entities.Alumno;

@ManagedBean(name = "trnInscripcionController")
@ViewScoped
public class InscripcionMangeBean extends
    WizardMaintenanceController<Alumno> {

    public InscripcionMangeBean() {
        super(InscripcionMangeBean.class, Alumno.class);
    }

    @PostConstruct
    public void init() {

    }

}
```

Figura 25 – ManageBean

3.4 Implementación de componentes PrimeFaces

Para utilizar los componentes PrimeFaces en un proyecto lo que se debe hacer es descargarse la librería que se encuentra en la página www.primefaces.org opción download.

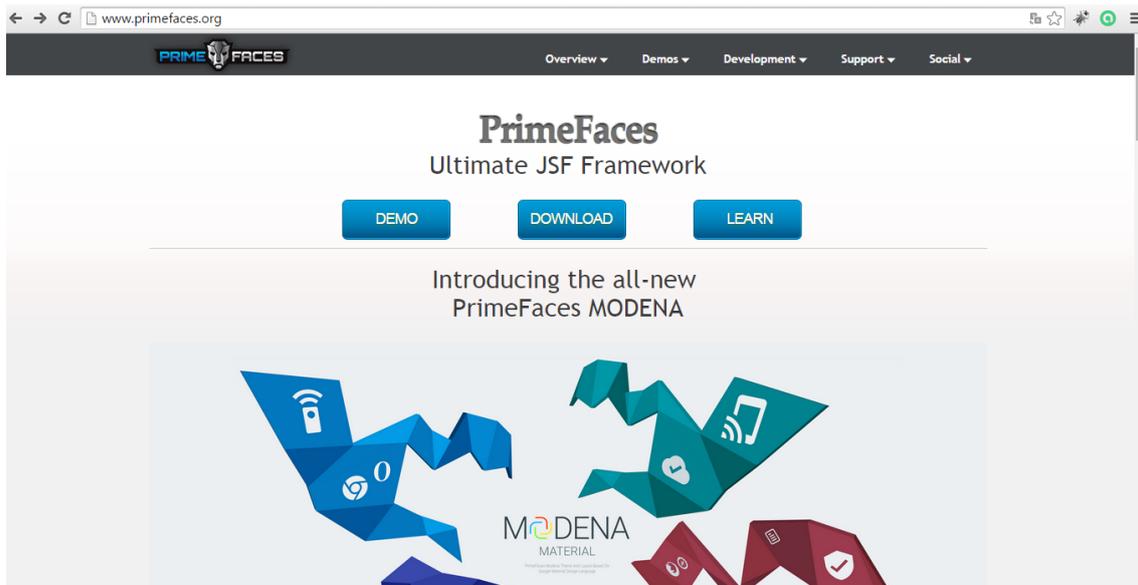


Figura 26 - PáginaPrime (Prime Tek, 2009)

Existen librerías pagadas que se incluyen en la versión ELITE, pero también está disponible la versión COMUNIDAD que es libre y es la que se adoptará para este proyecto. Se puede encontrar varias versiones de librerías, para ello, se debe seleccionar siempre la más actual, es así que se descarga el .jar y se lo agrega en el paquete de lib del proyecto JSF.

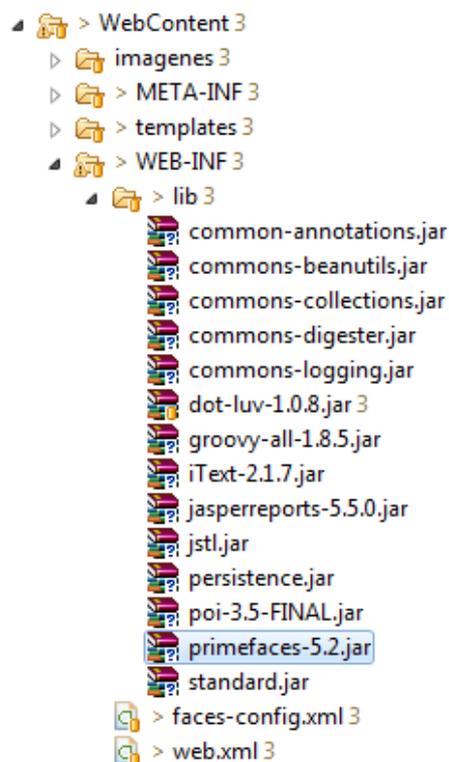


Figura 27 – Librería Prime

Es necesario verificar que se encuentre agregada la ubicación de la librería prime en el archivo de configuración build.xml, ya que solo así se podrá utilizar los componentes.

```
build.xml
<?xml version="1.0" encoding="UTF-8" ?>
<project name="JSF_Navegacion" basedir=".." default="deploy-show">

  <!-- Project settings -->
  <property name="project.distname" value="JSF_Navegacion" />

  <!-- Local system paths -->
  <property file="${basedir}/ant/build.properties" />
  <property name="webroot.dir" value="${basedir}/WebContent" />
  <property name="webinf.dir" value="${webroot.dir}/WEB-INF" />
  <property name="build.dir" value="build" />

  <path id="compile.classpath">
    <!-- classpath for JSF 2.0 -->
    <pathelement path="${webinf.dir}/lib/commons-annotations.jar" />
    <pathelement path="${webinf.dir}/lib/commons-beanutils.jar" />
    <pathelement path="${webinf.dir}/lib/commons-collections.jar" />
    <pathelement path="${webinf.dir}/lib/commons-digester.jar" />
    <pathelement path="${webinf.dir}/lib/commons-logging.jar" />
    <pathelement path="${webinf.dir}/lib/jsf-facelets.jar" />
    <pathelement path="${webinf.dir}/lib/jsf-tlds.jar" />
    <pathelement path="${webinf.dir}/lib/jstl.jar" />
    <pathelement path="${webinf.dir}/lib/standard.jar" />
    <pathelement path="${webinf.dir}/lib/primefaces-3.4.2.jar" />
  </path>
</project>
```

Figura 28 – Build

Al momento de crear las páginas xhtml, se debe agregar el namespace correspondiente a PrimeFaces para utilizar sus componentes, esto debe ser agregado en cada página que se crea.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:oc="http://java.sun.com/jsf/composite/components"
      xmlns:ocm="http://java.sun.com/jsf/composite/maintenance">

  <h:head>
    <title>#{msgs.pt_login}</title>
    <link rel="stylesheet" type="text/css"
          href="#{request.contextPath}/css/base.css" />

    <script type="text/javascript"
            src="#{request.contextPath}/js/openCEO.js" />
  </h:head>
  <h:body styleClass="LoginNuevo">[]
</html>
```

Figura 29 – Página xhtml

Es importante que la página contenga la etiquetas `<h:head>` y un `<h:body>` y dentro del `body` se puede utilizar todos los componentes PrimeFaces por medio de la etiqueta `<p: ...>`

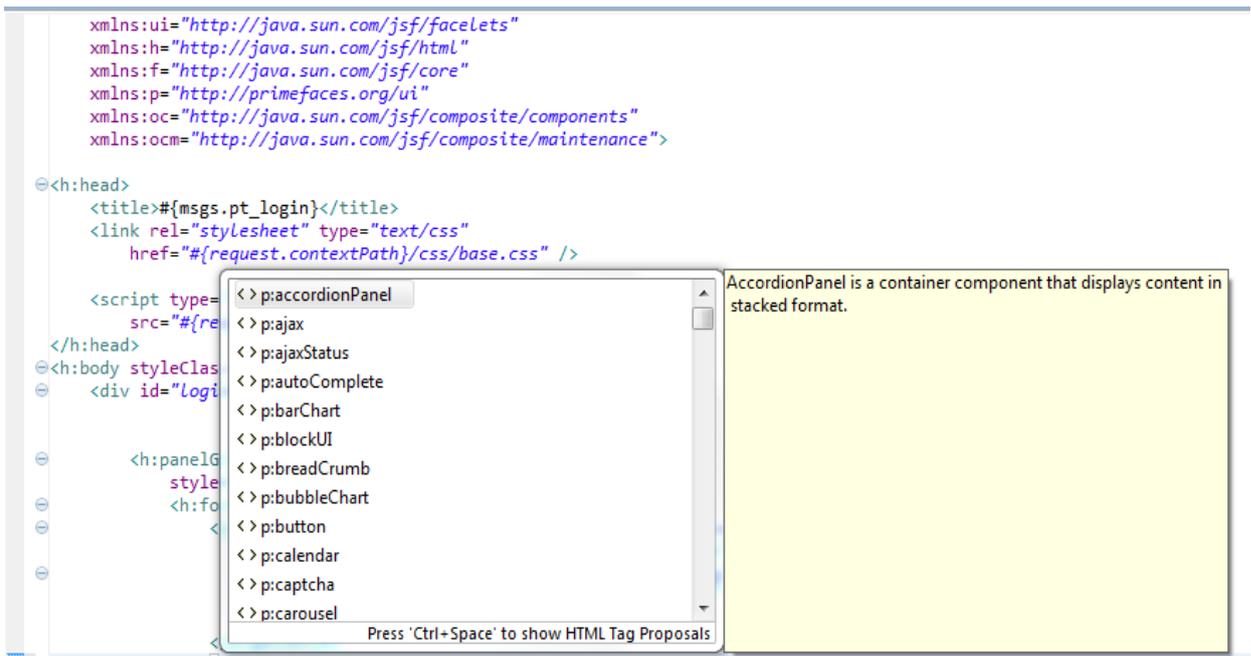


Figura 30 – Componente Prime

Existe una gran variedad de componentes PrimeFaces que se puede utilizar, como: cajas de texto, botones, paneles, calendarios, tablas, etc. Estos componentes vienen prediseñados y lo único que se necesita es invocar la etiqueta y colocar los valores que se requiere.

Cabe resaltar que PrimeFaces también cuenta con un conjunto de temas que nos permite dar colores y un mejor aspecto visual al sistema sin necesidad de crear hojas de estilo, para agregar un tema al proyecto se debe ingresar a la página de Prime www.primefaces.org y seleccionar *Theme Gallery*, existe una gran variedad de temas a elegir, éstos son gratuitos y otros un poco más detallados que requieren pago.

Overview	Demos	Development	Support	Social
Why PrimeFaces	Showcase	Getting Started	Support Options	Blog
Who Uses PrimeFaces	Mock OS X	Documentation	PrimeFaces PRO	Forum
Testimonials	Mobile	Downloads	PrimeFaces ELITE	Newsletter
License	PrimeUI	Source	Training	Mobile Apps
Multimedia		Theme Gallery	Forum	Twitter
FAQ		Extensions	Issue Tracker	Facebook
			Partners	LinkedIn
			Certification	

Copyright 2009-2014 PrimeTek. All rights reserved.

Figura 31 – PáginaPrime Temas (Take, 2015)

COMMUNITY

Community themes are open source and free to use under [Apache License](#). These themes include the ones available in ThemeRoller and custom themes like Twitter Bootstrap.



Figura 32 – PáginaPrime Temas 2 (Prime Tek, 2009)

PrimeFaces proporciona un conjunto de muestras de diseño en cada tema, se debe seleccionar el tema adecuado y descargar el .jar, teniendo en cuenta que también existen varias versiones y es mejor seleccionar la más actual. Luego se agrega el .jar en la carpeta lib del proyecto, y se configura el archivo web.xml para que se asigne el tema seleccionado.

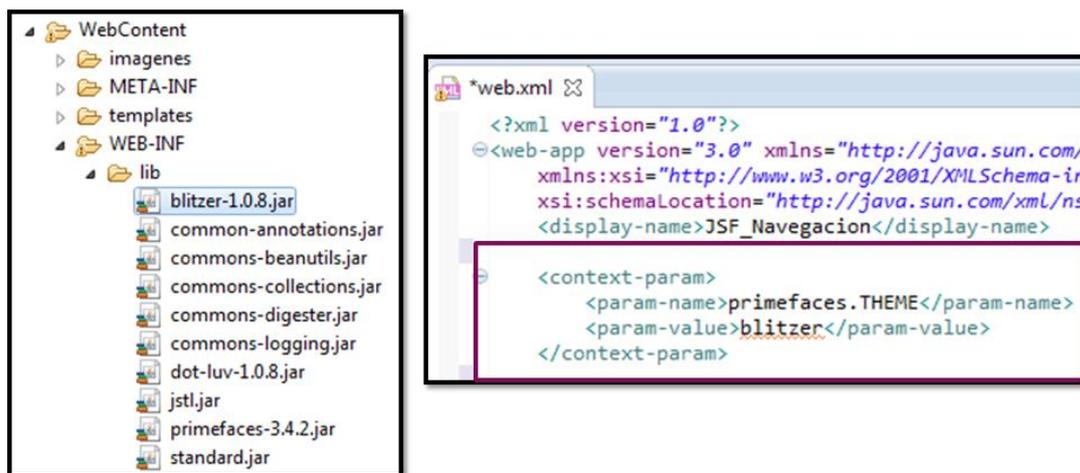


Figura 33 – Tema Prime

4. Validación y Pruebas

4.1 Configuración del servidor de aplicaciones

La aplicación que se va a desarrollar se ejecutará sobre un servidor de aplicaciones, en este caso la versión 7.1.1 final de JBoss, éste para su ejecución requiere la instalación de *Java Platform (JDK)* que puede ser descargado directamente de la página de Oracle (<http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>). Una vez instalado Java es necesario configurar las variables de entorno, indicando dónde se ubica Java dentro del Sistema Operativo, estas variables son “JAVA_HOME” y “PATH”

Se puede descargar JBoss de su sitio web oficial (www.jboss.org), no necesita instalación, basta con descomprimir la carpeta de archivos. Para probar el servidor se abre la ventana de comandos, se ubica en el directorio bin de JBoss y se ejecuta standalone.bat

```
C:\Users\Usuario>cd C:\Desarrollo\jboss-as-7.1.1.Final\bin
C:\Desarrollo\jboss-as-7.1.1.Final\bin>standalone.bat
```

Figura 34 - comandos

Si se despliega correctamente en el navegador digitamos 127.0.0.1:8080 y se muestra una pantalla como esta

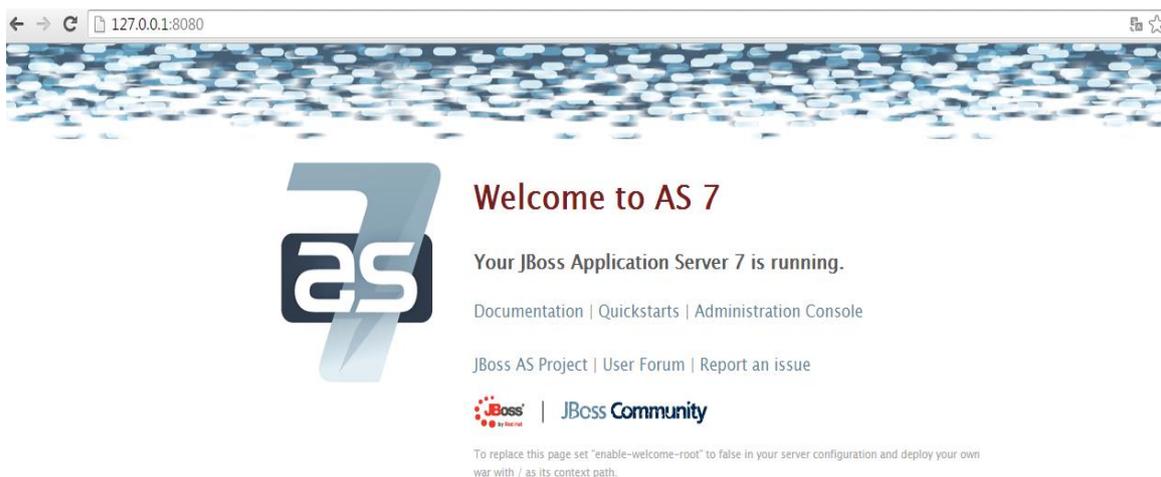


Figura 35 – Jboss

4.1.1 Configuración del archivo de conexión de base de datos

JBoss viene con una conexión por defecto a una base de datos *Hypersonic*, pero se puede configurar conexiones a distintos motores de base de datos como Oracle, MySQL, Postgres, etc.

La conexión se puede realizar en forma gráfica ingresando por medio de la url (127.0.0.1:8080) o directamente en el archivo de configuración (standalone.xml) que se encuentra en el directorio “standalone” de JBoss, para esta aplicación la base de datos se generó en Postgres por lo tanto la conexión debe:

```
<datasource jta="false" jndi-name="java:/inscripcion" pool-name="inscripcion" use-ccm="false">
  <connection-url>jdbc:postgresql://127.0.0.1:5432/inscripcion</connection-url>
  <driver-class>org.postgresql.Driver</driver-class>
  <driver>postgresql-8.4-702.jdbc4.jar</driver>
  <security>
    <user-name>postgres</user-name>
    <password>123</password>
  </security>
  <validation>
    <validate-on-match>>false</validate-on-match>
    <background-validation>>false</background-validation>
  </validation>
  <statement>
    <share-prepared-statements>>false</share-prepared-statements>
  </statement>
</datasource>
```

Figura 36 – Jboss configuración

Siendo “inscripcion” el nombre de la base de datos, el driver utilizado *postgresql-8.4-702.jdbc4.jar*, el usuario “postgres” y la contraseña 123.

Una vez realizada la conexión se reinicia el servidor JBoss y ya se encuentra listo para ejecutar la aplicación.

4.2 Pruebas del aplicativo

Para indicarle a la aplicación el servidor donde se ejecutará Eclipse cuenta con una ventana server donde se lo debe agregar, para ello se elige new server y aparece la siguiente ventana:

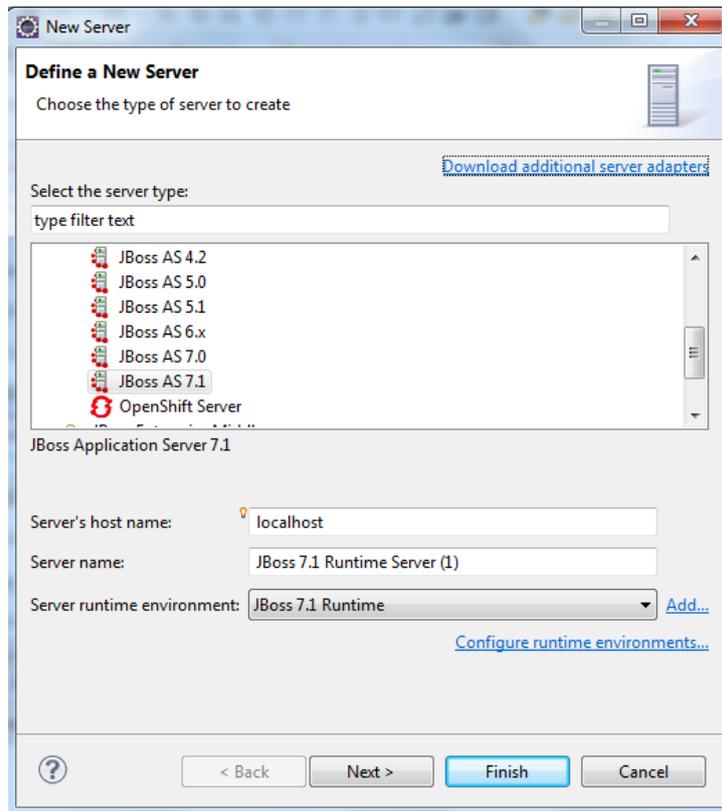


Figura 37 – Servidor JBoss

Se elige la versión de JBoss y el nombre del servidor, en el ambiente de servidor se presiona “Add” y se especifica el directorio de JBoss

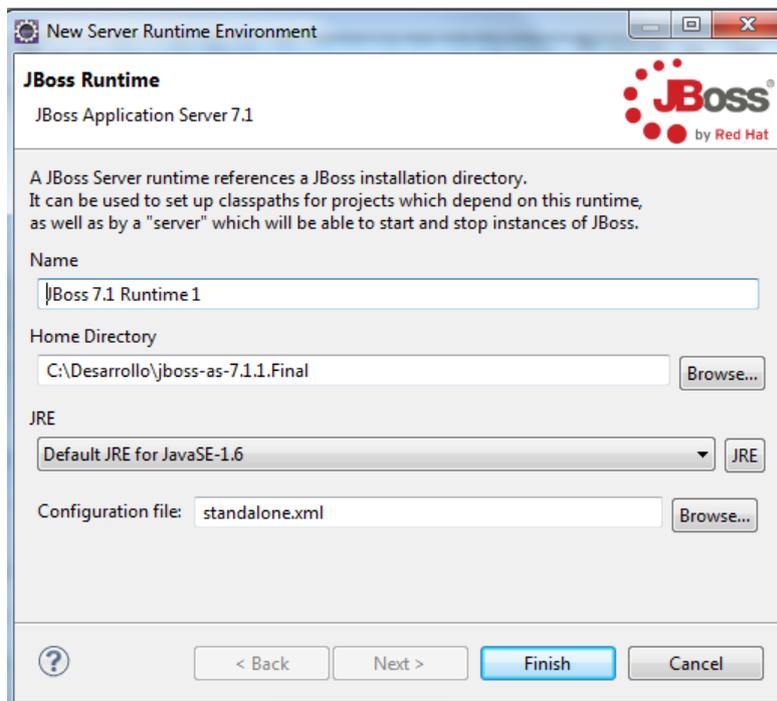


Figura 38 – Jboss nuevo

Se presiona “Finalizar” y “Siguiente”, le solicita los proyectos que se va a ejecutar en el servidor, se debe seleccionar “inscripción” y “Finalizar”.

En el directorio “ant” del proyecto se encuentra el archivo de configuración “build.properties” que indica la ruta del servidor donde se ejecuta, el puerto, la ip y el navegador.

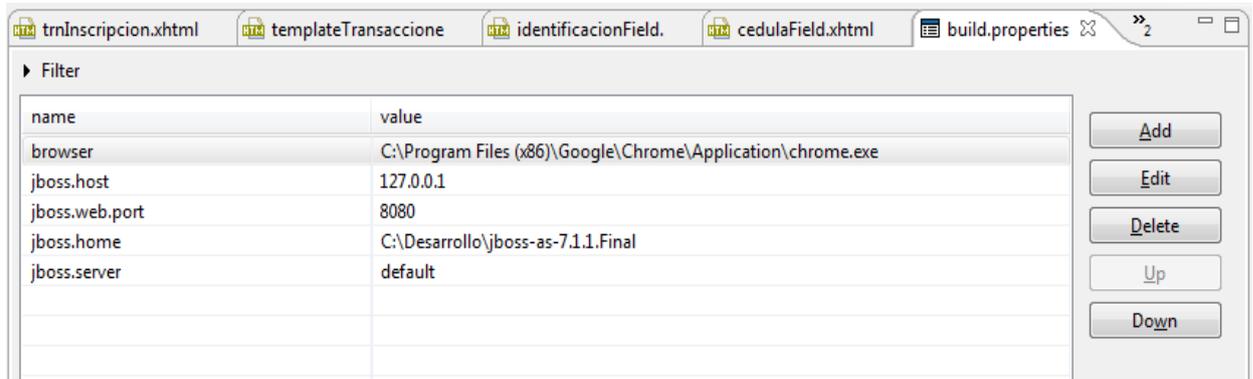


Figura 39 – Configuración servidor

Luego de ejecutar la aplicación, se despliega automáticamente en el navegador registrado y muestra la página principal.

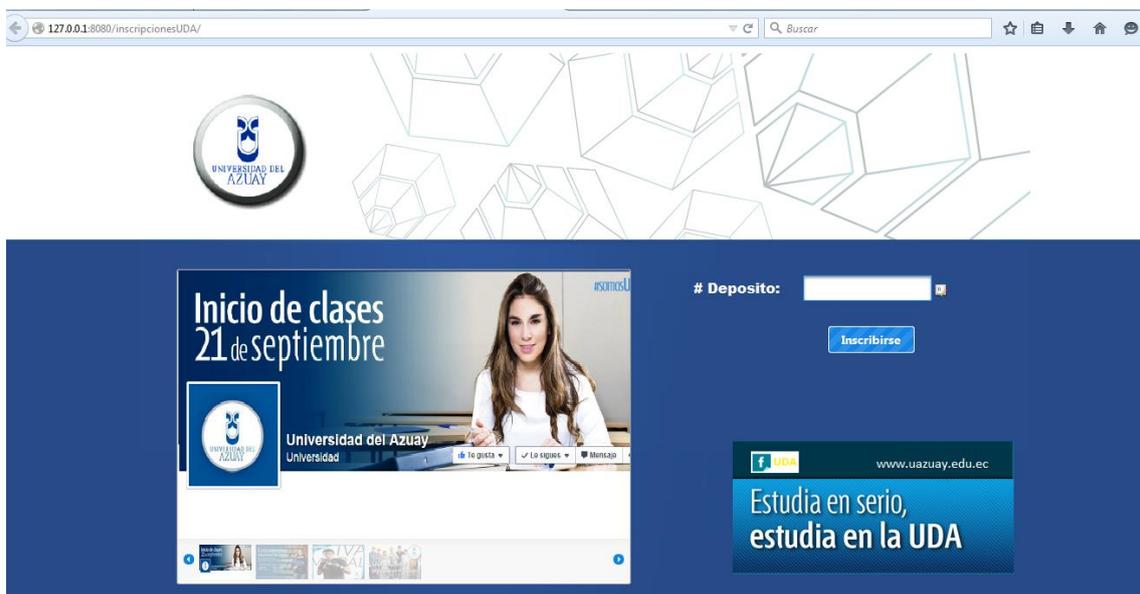


Figura 40 – página principal

Se procede a ingresar el número de depósito y se abre la pantalla de inscripción solicitando la información necesaria.

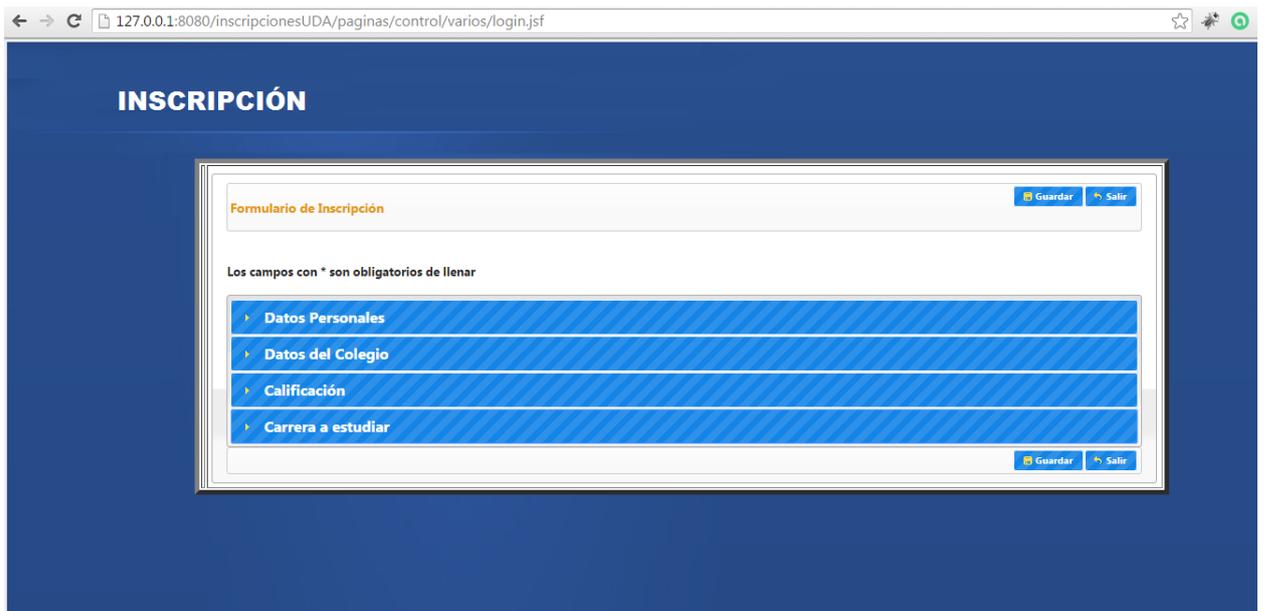


Figura 41 – Aplicación

La información se solicita por secciones: datos personales, datos de colegio, calificaciones y la carrera a estudiar, se incluye las validaciones respectivas en cada campo.

CONCLUSIONES Y RECOMENDACIONES

Es verdad que existen innumerables herramientas para desarrollar aplicaciones web pero no todas ofrecen las bondades necesarias para un correcto desarrollo, he allí la importancia de saber elegir la correcta antes de iniciar con el proyecto para evitar inconvenientes posteriores.

Con ayuda de investigación bibliográfica y proyectos desarrollados con esta tecnología se llega a la conclusión de utilizar JSF junto con los componentes PrimeFaces para crear un aplicación y demostrar todas las ventajas y virtudes que brinda esta fusión.

Además incorpora Java que es un lenguaje importante en el desarrollo, es por eso que su utilización brinda la oportunidad de crear un sistema robusto y consistente, la combinación con la tecnología JSF lo convierte en una herramienta eficaz para el desarrollo.

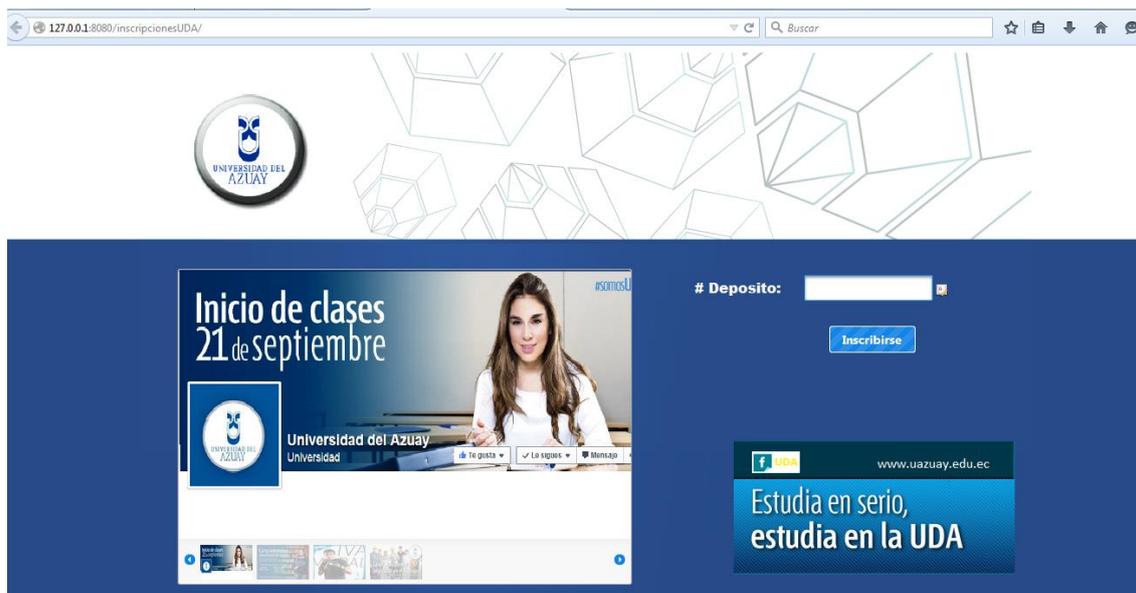
Entre las ventajas a destacar se encuentra su fácil implementación, sencilla integración a diversos gestores de base de datos debido a que su estructura se basa en 3 capas, reducción de código fuente por la reutilización de componentes, menor tiempo de implementación ya que cuenta con una gran variedad de componentes listos para usarse, mejora en el aspecto visual por su configuración de temas y un manejo adecuado de validaciones en cada campo.

Se recomienda que cada desarrollador analice todas las bondades que presenta esta tecnología y por medio de este documento verifique su fácil implementación para utilizar estas herramientas en aplicaciones futuras.

APENDICES

Manual de Usuario

Para acceder a la aplicación de inscripciones de la Universidad del Azuay digitamos en el navegador la siguiente dirección: 127.0.0.1:8080/inscripcionesUDA a continuación se despliega la siguiente pantalla:



En esta pantalla se puede apreciar una galería en la que incluyen las principales novedades de la institución, las imágenes están parametrizadas y se las puede cambiar en cualquier momento.

Para ingresar se requiere registrar el número de depósito que se realizó en el banco por el pago de la inscripción y se presiona el botón **inscribirse**

Si ya se realizó una inscripción con el número de depósito ingresado el sistema le reconoce automáticamente y aparecerá una pantalla con la información que se registró brindándole la opción de modificar dicha información cuantas veces sea necesario, en caso de no existir ninguna información almacenada se le mostrara la pantalla en blanco

Formulario de Inscripción Guardar Salir

Los campos con * son obligatorios de llenar

- ▶ Datos Personales
- ▶ Datos del Colegio
- ▶ Calificación
- ▶ Carrera a estudiar

Guardar Salir

La información que se solicita esta agrupada en 4 secciones: datos personales, datos del colegio, calificaciones y carrera a estudiar, se identifica con un asterisco (*) los campos que son obligatorios.

Datos Personales

Datos Personales

Identificación * cedula pasaporte ?

Primer Apellido * Segundo Apellido

Primer Nombre * Segundo Nombre

Genero * Estado Civil *

Fecha de Nacimiento *

Calle Principal * Número

Calle Secundaria * Parroquia *

Referencia Domiciliaria

E-mail * Confirmacion E-mail *

Número de Teléfono Tipo de Teléfono Agregar

Listado de Números Teléfonicos

Número	Tipo
No records found.	

** Debe ingresar por lo menos un número de teléfono

Identificación: se debe seleccionar primero el tipo que puede ser cedula o pasaporte y en el campo de la derecha ingresar el número, este es un campo requerido, la cedula debe constar de 10 números y cumplir con el dígito verificador.

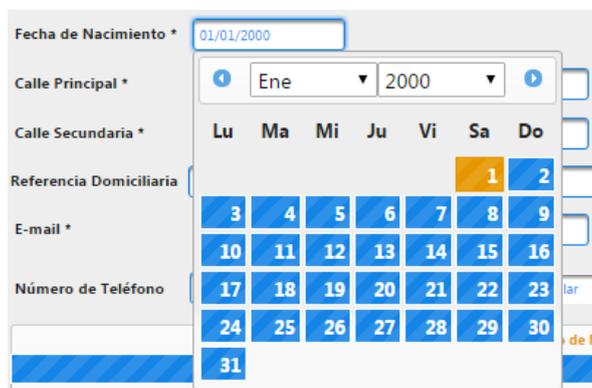
Identificación * cedula pasaporte ? La cedula debe tener 10 digitos

Apellidos: campos alfanuméricos que se deben registrar por separado siendo solo el primer apellido obligatorio.

Nombres: campos alfanuméricos que se deben registrar por separado siendo solo el primer nombre obligatorio.

Género y Estado Civil: Tiene cargado información por defecto, se debe escoger de una lista el valor correspondiente.

Fecha de nacimiento: este campo tiene un componente especial (calendario) del cual se debe seleccionar la fecha de su nacimiento, cuenta con listas desplegables que le permiten seleccionar el mes y año respectivamente.

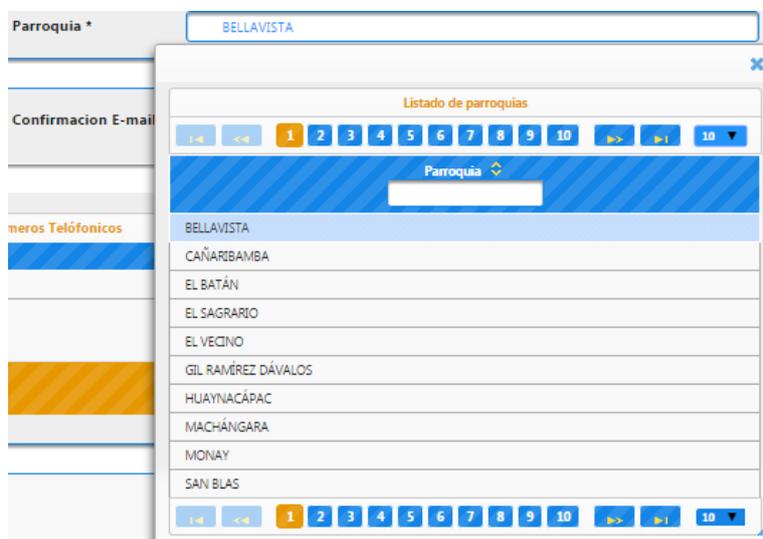


The image shows a form with a date selection calendar. The calendar is open, showing the month of January (Ene) for the year 2000. The date 01/01/2000 is entered in the input field above the calendar. The calendar grid shows days from Sunday to Saturday, with the 1st of January highlighted in orange. The form fields visible are: Fecha de Nacimiento *, Calle Principal *, Calle Secundaria *, Referencia Domiciliaria, E-mail *, and Número de Teléfono.

Calle Principal y calle secundaria: son campos alfanuméricos requeridos de su domicilio actual.

Número y referencia domiciliaria: son campos alfanuméricos opcionales.

Parroquia: contiene una tabla con los nombres de las distintas parroquias que existen, se puede filtrar por coincidencias de nombre y ordenar, cuenta con paginación.



The image shows a dropdown menu for selecting a parish. The dropdown is open, showing a list of parishes: BELLAVISTA, CAÑARIBAMBA, EL BATÁN, EL SAGRARIO, EL VECINO, GIL RAMÉREZ DÁVALOS, HUAYNACÁPAC, MACHÁNGARA, MONAY, and SAN BLAS. The dropdown is titled "Listado de parroquias" and has a search filter and pagination controls. The dropdown is open, showing a list of parishes: BELLAVISTA, CAÑARIBAMBA, EL BATÁN, EL SAGRARIO, EL VECINO, GIL RAMÉREZ DÁVALOS, HUAYNACÁPAC, MACHÁNGARA, MONAY, and SAN BLAS. The dropdown is titled "Listado de parroquias" and has a search filter and pagination controls.

E-mail y confirmación de E-mail: son campos alfanuméricos requeridos, deben coincidir y cumplir con el formato de un correo electrónico.

Teléfono: se requiere ingresar como mínimo un teléfono, indicando el número y el tipo, pulsamos agregar y se nos carga en la tabla de teléfonos, los registros que se

encuentren en la tabla se pueden eliminar presionando el botón  o se puede editar con el botón 

Número de Teléfono	<input type="text"/>	Tipo de Teléfono	Convencional		<input type="button" value="Agregar"/>
Listado de Números Teléfonicos					
Número	Tipo				
1234567890	Celular				
12341234	Convencional				

Datos del Colegio

Datos del Colegio

* Nombre del Colegio:

** De no existir su colegio por favor póngase en contacto con: inscripciones@uazuay.edu.ec

La única información que se solicita en esta sección es el nombre del colegio, se encuentran cargados en una tabla todos los nombres de los colegios que existen, se puede filtrar por coincidencias de nombre y ordenar, cuenta con paginación para mejorar en la búsqueda

Datos del Colegio

* Nombre del Colegio:

** De no existir su colegio por favor póngase en contacto con: inscripciones@uazuay.edu.ec

Calificación

* La nota que ingrese debe ser exactamente la que consta en el acta de grado emitida por la institución educativa (con decimales, sin redondeo)

Nota / Base

Listado de colegios

- CIUDAD DE CUENCA
- SUDAMERICANO
- BENIGNO MALO
- HERLINDA TORAL
- ASUNCION

Calificaciones

Calificación

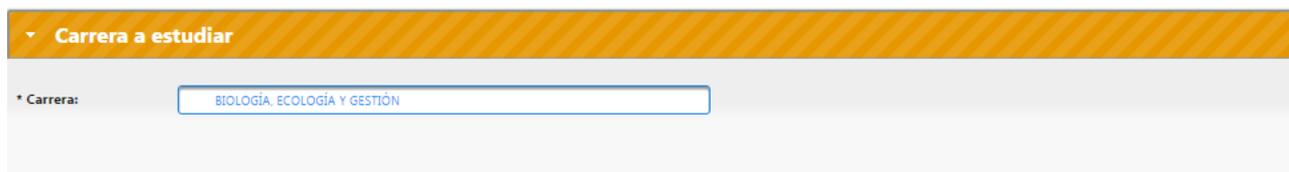
** La nota que ingrese debe ser exactamente la que consta en el acta de grado emitida por la institución educativa (con decimales, sin redondeo)

	Nota	/	Base
Ejemplo	18,63	/	20
Ejemplo	9,35	/	10
* Nota 1º Bachillerato	<input type="text" value="0.0"/>	/	<input type="text" value="0.0"/>
* Nota 2º Bachillerato	<input type="text" value="0.0"/>	/	<input type="text" value="0.0"/>
* Nota 3º Bachillerato	<input type="text" value="0.0"/>	/	<input type="text" value="0.0"/>

En caso de estar cursando nota del 1º Quimestre

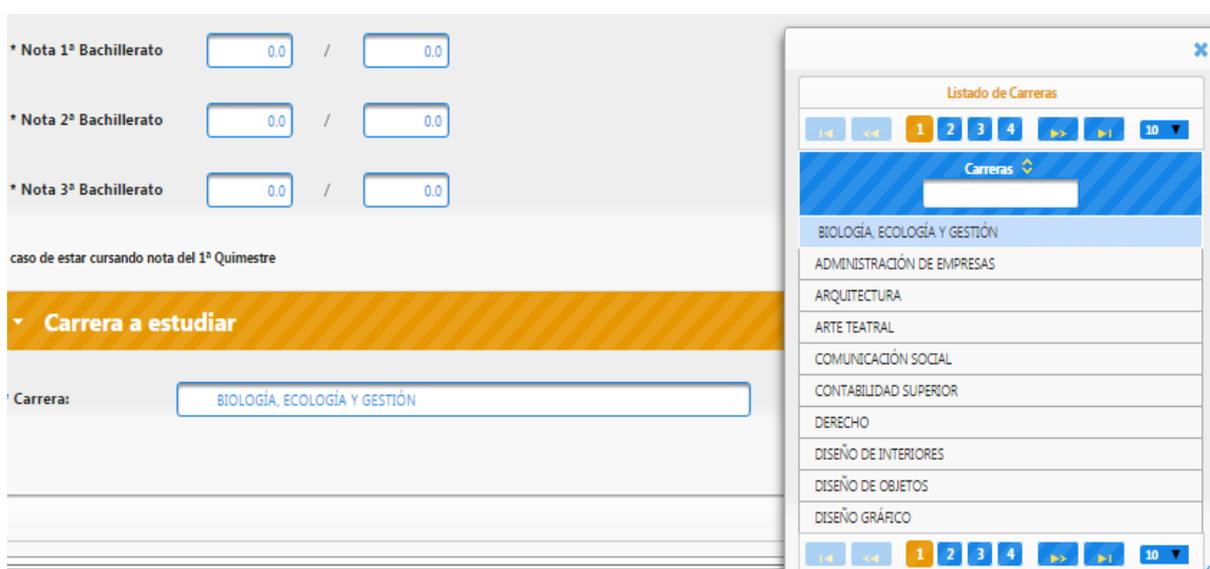
Se debe registrar las calificaciones obtenidas en el bachillerato, el primer campo numérico es para la nota y el segundo igual numérico es para la base de calificación, las notas no pueden ser mayores a sus bases.

Carrera a estudiar



Formulario de "Carrera a estudiar" con un campo de texto que contiene "BIOLOGÍA, ECOLOGÍA Y GESTIÓN".

Se encuentran listadas todas las carreras con las que cuenta la institución deberá elegirse la que se desea en la tabla, cuenta con búsqueda por coincidencia de nombre y se puede ordenar alfabéticamente, tiene paginación.



Formulario de "Carrera a estudiar" con campos de notas y un modal de "Listado de Carreras".

* Nota 1º Bachillerato: /

* Nota 2º Bachillerato: /

* Nota 3º Bachillerato: /

caso de estar cursando nota del 1º Quimestre

Carrera a estudiar

Carrera:

Listado de Carreras

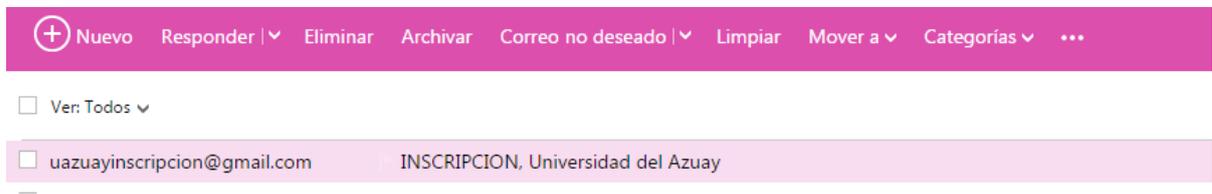
Carreras
BIOLOGÍA, ECOLOGÍA Y GESTIÓN
ADMINISTRACIÓN DE EMPRESAS
ARQUITECTURA
ARTE TEATRAL
COMUNICACIÓN SOCIAL
CONTABILIDAD SUPERIOR
DERECHO
DISEÑO DE INTERIORES
DISEÑO DE OBJETOS
DISEÑO GRÁFICO

Finalmente cuando se haya registrado toda la información en la parte derecha de la página tanto superior como inferior se encuentran dos botones; se debe pulsar el botón Guardar si desea completar con la inscripción o cancelar para salir a la página inicial.



Si pulsa guardar y la información no cumple con la validación se indica los campos con inconvenientes para que pueda arreglar, no se podrá guardar mientras existan errores.

Una vez almacenada la información se enviará un correo electrónico al estudiante indicándole que su inscripción se ha realizado con éxito.



Glosario

- **API:** (del inglés Application Programming Interface), interfaz de programación de aplicaciones, es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. (gris, 2015)
- **Applet:** Es un tipo especial de programa de Java que un navegador habilitado con tecnología Java, se puede descargar desde Internet y ejecutarlo. Frecuentemente es incluido dentro de una página web y se ejecuta en el contexto de un navegador. (oracle, 2014)
- **Arrays:** Es una matriz o vector que forma una zona de almacenamiento continuo y contiene una serie de elementos del mismo tipo, los elementos de la matriz. (gris, 2015)
- **JavaBeans:** Es una arquitectura de componentes para la plataforma *Java Standard Edition 2* (J2SE). Son programas de software reutilizables que se pueden desarrollar y ensamblar fácilmente para crear aplicaciones sofisticadas. (oracle, 2014)
- **JPA:** (del inglés Java Persistence API), proporciona un modelo de persistencia basado en POJO's para mapear bases de datos relacionales en Java. (oracle, 2014)
- **Multiplataforma:** Hace referencia a la capacidad de los programas o sistemas operativos de funcionar en diferentes plataformas. (ecured, 2014)
- **Open Source:** Indica que algo es de código abierto es decir que todas las personas tiene acceso a su diseño y pueden modificarlo.
- **Struts:** Esta herramienta soporta aplicaciones Web construidas bajo el modelo vista controlador.
- **XML:** (del inglés eXtensible Markup Language), lenguaje de marcas extensible, fue desarrollado por el World Wide Web Consortium (W3C). XML da soporte a bases de datos permitiendo que varias aplicaciones puedan comunicarse entre sí o integrar información. (gris, 2015)

BIBLIOGRAFIA

- ecured. (06 de 10 de 2014). *ecured*. Recuperado el 05 de 10 de 2015, de ecured: <http://www.ecured.cu/index.php/Multiplataforma>
- Fain, Y. (2011). *Java Programming 24-Hour Trainer*. John Wiley & Sons.
- Global Mentoring. (12 de julio de 2012). *global mentoring*. Recuperado el 01 de 09 de 2015, de global mentoring: <http://globalmentoring.com.mx/cursos-java/java-empresarial/arquitectura-multicapas/>
- gris, a. (26 de 01 de 2015). *usabilidad, promoción y analítica web*. Recuperado el 05 de 10 de 2015, de usabilidad, promoción y analítica web: <http://www.alexgris.com/>
- oracle. (01 de 12 de 2014). *Oracle corporation us*. Recuperado el 30 de 01 de 2015, de Oracle corporation us: <https://www.java.com/es/>
- Parkway, 5. O. (01 de 2013). *The Java EE 6 Tutorial*. Recuperado el 01 de 02 de 2015, de The Java EE 6 Tutorial: <http://docs.oracle.com/javasee/6/tutorial/doc/bnaph.html>
- Prime Tek. (2009). *PrimeFaces*. Recuperado el 04 de 09 de 2015, de PrimeFaces: <http://www.primefaces.org/>
- rafaelma. (02 de 10 de 2010). *Sobre PostgreSQL*. Recuperado el 30 de 05 de 2015, de Sobre PostgreSQL: http://www.postgresql.org.es/sobre_postgresql
- Red Hat. (01 de 01 de 2015). *www.redhat.es/jboss*. Obtenido de www.redhat.es/jboss: http://es.redhat.com/pdf/jboss/JBoss_Ent_app_platform_ES_web.pdf
- Red Hat. (01 de 01 de 2015). *www.redhat.es/jboss*. Recuperado el 01 de 06 de 2015, de www.redhat.es/jboss: http://es.redhat.com/pdf/jboss/JBoss_Ent_app_platform_ES_web.pdf
- Smart, J. F. (2007). *JSF Jumpstart*. Wellington, Nueva Zelanda: Wakaleo Consulting Limited.
- Take, P. (2015). *PrimeFaces showcase*. Recuperado el 01 de 02 de 2015, de PrimeFaces showcase: <http://www.primefaces.org/showcase/>

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay,

CERTIFICA:

Que, el H. Consejo de Facultad en sesión realizada el 18 de junio de 2015, conoció la petición de la estudiante **JENNY SUSANA LAZO CAJILIMA** con código 47107, que denuncia su trabajo de titulación denominado: **“DESARROLLO DE UN PROTOTIPO WEB PARA LA INSCRIPCION DE NUEVOS ALUMNOS EMPLEANDO LA TECNOLOGIA JAVA SERVER FACES CON COMPONENTES PRIMEFACES.”**, presentado como requisito previo a la obtención del título de Ingeniera en Sistemas y Telemática.- El Consejo de Facultad acoge el informe de la Junta Académica y aprueba la denuncia del trabajo de titulación - Designa como **Director al : Ing. Marcos Orellana Cordero** y como miembro del Tribunal Examinador al Ing. Oswaldo Merchán Manzano.- De conformidad con el Instructivo de Grados de la Facultad, la peticionaria para desarrollar y presentar su trabajo de titulación, tiene un plazo de **TRES MESES**, contados a partir de la fecha de aprobación del Consejo de Facultad, estos es **hasta el 18 de septiembre de 2015.**

Cuenca, junio 23 de 2015



Dra. Jenny Ríos Coello
Secretaria de la Facultad

CONVOCATORIA

Por disposición de la Junta Académica de Ingeniería de Sistemas y Telemática, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación : "Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces", presentado por la estudiante Jenny Susana Lazo Cajilima con código 47107, previa a la obtención del grado de Ingeniero de Sistemas y Telemática, para el día **MIÉRCOLES 25 DE MARZO 2015 A LAS 09H00.**

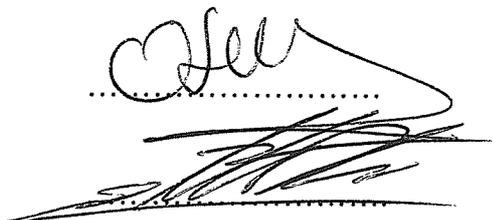
Cuenca, 19 de marzo de 2015



Dra. Jenny Ríos Coello
Secretaria de la Facultad

Ing. Oswaldo Merchán Manzano

Ing. Marcos Orellana Cordero



Manzano

Oficio Nro. 033-2015-DIST-UDA

Cuenca, 05 de Marzo de 2015

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 05 de marzo del 2015, recibió el proyecto de monografía titulado "Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces", presentada por la estudiante Jenny Lazo, estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Marcos Orellana, previo a la obtención del título de Ingeniero de Sistemas y Telemática.

La Junta solicita por su digno intermedio notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior a al Ing. Marcos Orellana y como miembro del Tribunal al Ing. Oswaldo Merchán.

Atentamente,



Ing. Esteban Crespo
Miembro de la Junta
Académica



Ing. Catalina Astudillo
Miembro de la Junta
Académica

Fecha: 13-03-2015

ESCUELA DE INGENIERIA DE SISTEMAS

Diseños de Monografía

Escuela de Sistemas

Estudiante: Jenny Lazo Cajilima con código 47107.

Tema: "DESARROLLO DE UN PROTOTIPO WEB PARA LA INSCRIPCION DE NUEVOS ALUMNOS EMPLEANDO LA TECNOLOGIA JAVA SERVER FACES CON COMPONENTES PRIMEFACES"

Para: La obtención del título de Ingeniera de Sistemas y Telemática.

Director: Ing. Marcos Orellana Cordero.

Tribunal: Ing. Oswaldo Merchán Manzano.

DIA: *MIERCOLES*

FECHA: *25 MARZO 2015*

HORA: *09h00*



ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante: Jenny Susana Lazo Cajilima
Código 47107
- 1.2 Director sugerido: Ing. Marcos Orellana Cordero
- 1.3 Codirector (opcional): _____
- 1.4 Tribunal: Ing. Oswaldo Merchán Manzano
- 1.5 Título propuesto "Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces"
- 1.6 Resolución:

1.6.1 Aceptado sin modificaciones

1.6.2 Aceptado con las siguientes modificaciones:

- Responsable de dar seguimiento a las modificaciones: Ing. Marcos Orellana Cordero

1.6.3 No aceptado
• Justificación:

Tribunal

Ing. Marcos Orellana Cordero

Ing. Oswaldo Merchán Manzano

Srta. Jenny Susana Lazo Cajilima

Dra. Jenny Ríos Coello
Secretario de Facultad

Fecha de sustentación: Miércoles 25 de marzo de 2015



RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante: Jenny Susana Lazo Cajilima
- 1.1.1 Código 47107
- 1.2 Director sugerido: Ing. Ing. Marcos Orellana Cordero
- 1.3 Codirector (opcional):.
- 1.4 Título propuesto: "Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces"
- 1.5 Revisores (tribunal): Ing. Oswaldo Merchán Manzano
- 1.6 Recomendaciones generales de la revisión:

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	/			
Título Propuesto				
2. ¿Es informativo?	/			
3. ¿Es conciso?	/			
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	/			
5. ¿Describe la teoría en la que se enmarca el trabajo	/			
6. ¿Describe los trabajos relacionados más relevantes?	/			
7. ¿Utiliza citas bibliográficas?	/			
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?	/			
9. ¿Tiene relevancia profesional y social?	/			
Hipótesis (opcional)				
10. ¿Se expresa de forma clara?	/			
11. ¿Es factible de verificación?	/			
Objetivo general				
12. ¿Concuerda con el problema formulado?	/			
13. ¿Se encuentra redactado en tiempo verbal infinitivo?	/			
Objetivos específicos				



14.¿Concuerdan con el objetivo general?	/			
15.¿Son comprobables cualitativa o cuantitativamente?	/			
Metodología				
16.¿Se encuentran disponibles los datos y materiales mencionados?	/			
17.¿Las actividades se presentan siguiendo una secuencia lógica?	/			
18.¿Las actividades permitirán la consecución de los objetivos específicos planteados?	/			
19.¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	/			
Resultados esperados				
20.¿Son relevantes para resolver o contribuir con el problema formulado?	/			
21.¿Concuerdan con los objetivos específicos?	/			
22.¿Se detalla la forma de presentación de los resultados?	/			
23.¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	/			
Supuestos y riesgos				
24.¿Se mencionan los supuestos y riesgos más relevantes?	/			
25.¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	/			
Presupuesto				
26.¿El presupuesto es razonable?	/			
27.¿Se consideran los rubros más relevantes?	/			
Cronograma				
28.¿Los plazos para las actividades son realistas?	/			
Referencias				
29.¿Se siguen las recomendaciones de normas internacionales para citar?	/			
Expresión escrita				
30.¿La redacción es clara y fácilmente comprensible?	/			
31.¿El texto se encuentra libre de faltas ortográficas?	/			

(*) Breve justificación, explicación o recomendación.



- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

.....

.....

.....

Ing. Marcos Orellana Cordero

Ing. Oswaldo Merchán Manzano

Cuenca, 05 de marzo de 2015.

Ing. Xavier Ortega Vasquez

Decano de la Facultad de Ciencias de la Administración

Ciudad

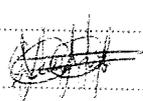
De mis consideraciones:

YO, JENNY SUSANA LAZO CAJILIMA con código 47107, estudiante de la Escuela de Sistemas y Telemática, solicito a usted de la manera más respetuosa y por su intermedio al Honorable Consejo de Facultad, se sirvan revisar mi diseño de tesis titulado: **“Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces”**, previo a la obtención del título de Ingeniero en Sistemas y Telemática.

Me permito sugerir el nombre del Ing. Marcos Orellana como director, quién me ha asesorado en la elaboración del presente esquema y ya cuento con su previa aceptación.

Por la favorable acogida que se sirva a la presente, suscribo a usted.

Atentamente


Jenny Susana Lazo Cajilima

0105891428

Cuenca, 05 de Marzo de 2015

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De mis consideraciones:

Luego de revisar el diseño del trabajo de graduación denominado "Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces"; presentado por la estudiante Jenny Lazo, estudiante de la Escuela de Ingeniería de Sistemas y Telemática, considero que el documento cumple con las normas legales y reglamentarias de la Universidad y de la Facultad de Ciencias de la Administración, por lo que recomiendo su aprobación por parte del Consejo de Facultad.

Atentamente,



Ing. Marcos Orellana Cordero
Director



UNIVERSIDAD DEL
AZUAY

DOCTORA JENNY RIOS COELLO, SECRETARIA DE LA FACULTAD DE
CIENCIAS DE LA ADMINISTRACIÓN DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, la Señorita **Jenny Susana Lazo Cajilima** registrada con código **47107**, perteneciente a la Escuela de Ingeniería de Sistemas luego de cumplir con todas las asignaturas de su Pensum de estudios, egresó de la Facultad el día 20 de Julio de 2013.

Cuenca, Enero 26 de 2015

Derecho 71633
vcf.-



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

PROTOCOLO DE TRABAJO DE TITULACION

Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología
Java Server Faces con componentes PrimeFaces.

Autor: Jenny Lazo

Director: Ing. Marcos Orellana

Cuenca, Ecuador

Marzo, 2015



1. DATOS CENTRALES

1.1. Nombre del estudiante: Lazo Cajilima Jenny Susana

1.1.1. Código: 47107

1.1.2. Contacto: (02)2250750, 0998856724, jennys202@hotmail.es

1.2. Director sugerido: Marcos Orellana Cordero

1.2.1. Contacto: 0999955611, marore@uazuay.edu.ec

1.3. Tribunal designado:

1.4. Aprobación:

1.5. Línea de Investigación de la carrera: 1203 Informática de computadores

1.5.1. Código UNESCO: 1203.17. Informática

1.5.2. Tipo de Trabajo: Estudio comparado basándose en una investigación formativa

1.6. Área de estudio: Lenguajes de programación

1.7. Título propuesto: Desarrollo de un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces

1.8. Subtítulo: Como implementar un proyecto con PrimeFaces

1.9. Estado del proyecto: el trabajo es nuevo

2. CONTENIDO

2.1 Motivación de la investigación:

Con el avance tecnológico hoy en día se puede encontrar una gran variedad de herramientas que nos permiten realizar aplicaciones web, entre las cuales se encuentra jsf (Java Server Faces), que actualmente ha tenido un gran auge debido a su facilidad de uso. Como desarrolladores es importante conocer esta herramienta, las bondades que esta ofrece y su respectiva implementación mediante un ejemplo práctico.

2.2 Problemática:

Los lenguajes de programación usados para aplicaciones web son muy variados y extensos, lo que ha generado una gran cantidad de tecnologías de desarrollo que nos pueda facilitar la creación de un proyecto web, destacando como principales java server faces (JSF), PHP y NET.

Al ser JSF una herramienta que recién está surgiendo en el mercado, no existe la información necesaria sobre la implementación de un proyecto empleando esta tecnología y menos enfocándose específicamente al uso de los componentes PrimeFaces

La información que se encuentra es superficial y por partes mas no orientado a un proyecto integral, esto causa que muchos desarrolladores opten por no utilizar esta herramienta, cuando en realidad es de fácil implementación y puede adaptarse a su proyecto.

2.3 Pregunta de investigación:

¿Se puede implementar un prototipo web para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces?

2.4 Resumen

El diseño y la implementación de un proyecto web se torna complicado si no existe la información suficiente para realizarlo y más aún si la herramienta que se utiliza es nueva, es por ello que con el presente trabajo se pretende explicar de forma sencilla la implementación de un proyecto web, mediante el ejemplo aplicado a la inscripción para los alumnos nuevos de la Universidad del Azuay, por medio de esto se detallará la forma de llevar a cabo la implementación utilizando la tecnología JSF con componentes Prime Faces.

Al finalizar se obtendrá un proyecto culminado y una clara explicación sobre la funcionalidad de Prime Faces para implementaciones en proyectos futuros.

2.5 Estado del Arte y Marco Teórico

Desde la aparición del Internet, fueron surgiendo diferentes requerimientos por parte de los usuarios y a partir de esto varias tecnologías fueron apareciendo y desarrollándose para cumplir con todas estas demandas. Estas tecnologías fueron incorporando varias características entre ellas la interacción con el usuario y la utilización de bases de datos.

Entre las más conocidas está Java Server Faces (JSF), que actualmente está tomado ventaja ante las demás.

Java Server Faces (JSF)

Java es un lenguaje de programación orientado a objetos y fue comercializado por primera vez por Sun Microsystems en 1995. Hoy en día abarca una gran cantidad de mercado ya que existen varios sitios web que requieren java para su funcionamiento. “Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes” (oracle)

Java como lenguaje de programación tiene innumerables características pero en este caso la investigación se centrará en la parte orientada a desarrollo web.

JSF (Java Server Faces) es una tecnología para el desarrollo de aplicaciones web basadas en java, JSF surge después de Struts que es el que se basa en el modelo vista controlador (MVC) por lo tanto hereda todas sus características y las mejora, con esto se quiere estandarizar el desarrollo de aplicaciones web, “una de las mayores ventajas de la tecnología JavaServer Faces es que ofrece una separación clara entre el comportamiento y la presentación de las aplicaciones web” (Parkway, 2013)

Para el desarrollo de las páginas usa JSP (Java Server Pages) que son etiquetas personalizadas, pero también se puede usar otras tecnologías como XUL (XML-based User-interface Language). Para el manejo de evento, validación de datos de entrada, accesibilidad, navegación, etc., cuenta con un conjunto de APIs, además “no puede añadir java a las páginas xhtml, por lo que se ve obligado a separar su código java de los componentes html” (Fain, 2011)



Esto lo convierte en una plataforma flexible de fácil uso que no limita al desarrollador ya sea en lenguaje o a la presentación al cliente ya que se puede encapsular la funcionalidad de los componentes y ser presentados de diversas formas, separando claramente la lógica de la presentación, por lo tanto se puede establecer grupos de desarrollo y cada uno centrarse en su proceso.

JSF es una especificación es por ello que surge la necesidad de componentes especializados, es por esto que hay varias empresas que han construido su set de componentes en base a esta especificación. Como PrimeFaces, IceFaces, RichFaces.

Cuando se inicia un proyecto real, necesariamente se debe elegir una implementación, ya que se puede contar un set de componentes más amplio, versátil y además un mejor manejo de la parte visual, en el mercado hay distintas empresas que nos brindan estos frameworks entre los cuales está PrimeFaces.

PRIMEFACES

PrimeFaces es una implementación de la tecnología JSF, se lo puede obtener mediante la descarga de una librería que nos proporciona una gran variedad de componentes para utilizarlos en nuestros proyectos web. La gran ventaja con la que cuenta PrimeFaces es que no nos limita al uso exclusivo de estos componentes sino que interactúan normalmente con otros; "PrimeFaces es un marco de código abierto muy popular para JavaServer Faces con más de 100 componentes". (Take, PrimeFaces showcase, 2015)

Los componentes de esta librería son muy amigables y tienen un diseño es muy agradable para los usuarios finales, a más de eso también permite la implementación de hojas de estilo adaptándose a los requerimientos de programador.

Una desventaja es que las versiones de las nuevas librerías no son compatibles con las anteriores, es por ello que si el proyecto se inicia con una versión se recomienda terminarlo con la misma versión y no migrar de librería porque esto implica varios cambios para la integración.

2.6 Objetivo General

Desarrollar una aplicación prototipo web para la inscripción de nuevos alumnos en la Universidad del Azuay a través del framework PrimeFaces con tecnología JSF.

2.7 Objetivos Específicos

- Analizar las características y opciones que ofrece el framework PrimeFaces.
- Desarrollar una aplicación prototipo para la inscripción de alumnos nuevos a través de una interface web.

2.8 Metodología

El proyecto es un trabajo investigativo por lo tanto se procederá a consultar la información que nos brinda las páginas web referentes a la temática.

Se investigará la tecnología JSF y la manera de implementar un proyecto con componentes PrimeFaces a través de revisiones bibliográficas.

Se realizará una entrevista al Ing. Juan Pablo Ezquivel para obtener los requisitos para la inscripción de nuevos alumnos.

Finalmente se desarrollará un prototipo web para la inscripción de nuevos alumnos.

2.9 Alcances y resultados esperados

Al concluir la presente investigación se pretende detallar paso a paso la implementación de un proyecto web empleando la tecnología JSF con componentes PrimeFaces por medio de la creación del aplicativo web que permitirá la inscripción de nuevos alumnos.

Con esto también se pretende que los desarrolladores y personas interesadas en aprender sobre este framework obtengan una guía práctica.

2.10 Supuestos y riesgos

- Falta de información o información irrelevante en la web: Para contrarrestar este riesgo se entrevistará a profesionales que conozcan esta herramienta.
- Retrasos en los tiempos establecidos para el desarrollo del software: Para esto se pretende realizar un cronograma que abarque tiempos de contingencia.
- Daño en equipos: se contará con un equipo alternativo para el desarrollo del aplicativo previamente instalado con todas las herramientas necesarias y un servidor de repositorio para mantener siempre respaldada la aplicación.

2.11 Presupuesto

Rubro – Denominación	Costo USD	Justificación
Computador	\$50 x 3 meses=\$150	Herramienta necesaria para el desarrollo de la monografía
Gastos de internet	\$25 x 3 meses=\$75	Para consultas directas en la web sobre la herramienta



UNIVERSIDAD DEL
AZUAY

Gastos de movilización	\$50	Consultas o reuniones con personas que utilizan esta herramienta
Materiales de oficina	\$50	Requeridas durante la elaboración del proyecto

2.12 Financiamiento

Todos los gastos que se presenten en la elaboración del proyecto serán cubiertos por el estudiante, que es la persona interesada en la elaboración de este proyecto.

2.13 Esquema tentativo

1. Página de título
2. Tabla de contenido
3. Introducción
4. Resumen
5. Capítulo 1: Conceptos básicos
 1. Que es Java
 2. Tecnología JSF
 3. Componentes Primefaces
 4. Gestor de base de datos Postgres
 5. Servidor de aplicaciones JBoss
6. Capítulo 2: Diseño
 1. Diagramas de clases
 2. Diagrama de casos de uso
 3. Modelo entidad-relación
7. Capítulo 3: Desarrollo
 1. Mapeo de entidades
 2. Creación de los Manage Bean
 3. Implementación de los componentes PrimeFaces
8. Capítulo 4: Validación y pruebas
 1. Pruebas del aplicativo
9. Conclusiones y recomendaciones
10. Apéndices, tablas, gráficos
11. Referencias

2.14 Cronograma

Objetivo Específico	Actividad	Resultado esperado	Tiempo (semanas)
	Buscar información sobre la tecnología JSF con componentes PrimeFaces		
Analizar las características y opciones que otorga el framework PrimeFaces.	<ul style="list-style-type: none"> • Configuración del ambiente de desarrollo • Creación del proyecto JSF • Configuración del archivo de persistencia • Mapeo de entidades • Creación de los Manage Bean 	Capacitación necesaria para implementar un proyecto web utilizando la tecnología JSF junto a los componentes PrimeFaces	3 semanas
	Detallar el proceso a seguir para la inscripción de los nuevos alumnos, las pantallas, validación de datos, etc.		
Desarrollar una aplicación para la inscripción de alumnos nuevos a través de una interface web.	<p>Especificar las diferentes entidades, atributos y relaciones que formarán parte de la base de datos requerida para la aplicación</p> <p>Trabajar sobre una arquitectura de 3 capas (capa de presentación, capa de negocios y capa de acceso a la base de datos), en el cual se distinguirá claramente lo que abarca cada una de las capas.</p>	Prototipo web para la inscripción de nuevos alumnos	9 semanas

2.15 Referencias

Fain, Y. (2011). *Java Programming 24-Hour Trainer*. John Wiley & Sons.

oracle. (s.f.). *Oracle corporation us*. Recuperado el 30 de 01 de 2015, de Oracle corporation us: <https://www.java.com/es/>

Parkway, S. O. (01 de 2013). *The Java EE 6 Tutorial*. Recuperado el 01 de 02 de 2015, de The Java EE 6 Tutorial: <http://docs.oracle.com/javaee/6/tutorial/doc/bnaph.html>

Smart, J. F. (2007). *JSF Jumpstart*. Wellington, Nueva Zelanda: Wakaleo Consulting Limited.

Take, P. (2015). *PrimeFaces showcase*. Recuperado el 01 de 02 de 2015, de PrimeFaces showcase: <http://www.primefaces.org/showcase/>



UNIVERSIDAD DEL
AZUAY

2.16 Firma de responsabilidad (estudiante)

Jenny Lazo

2.17 Firma de responsabilidad (director)

Ing. Marcos Orellana

2.18 Fecha de entrega

05/03/2015