

Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con implementación de PrimeFaces Mobile

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

Autor: Digna Isabel Quezada Pauta

Director: Ing. Marcos Orellana

Cuenca, Ecuador

2015

DEDICATORIA

Este trabajo les dedico en primero lugar a mis padres, por el apoyo incondicional que me han brindado, por sus consejos, su paciencia y por todo el esfuerzo que para ellos ha implicado el apoyarme en este proceso académico. A mis hermanos que siempre han están pendientes de mí y me han apoyado cuando más lo he necesitado. De igual forma a mis amigos, que me han brindado su amistad y que han aportado de manera positiva en este proceso.

AGRADECIMIENTO

A los docentes de la Universidad, por compartir con nosotros sus conocimientos a lo largo de todos estos años de estudio.

Al Ing. Marcos Orellana, por brindarme su respaldo en la realización de este proyecto, ya que sin su ayuda no hubiera alcanzado los objetivos propuestos.

A mis compañeros de clases, por brindarme su amistad, compartir sus conocimientos y por las innumerables experiencias que hemos compartido, por las alegrías, las tristezas, por los momentos de victoria y por los fracasos. Ustedes han contribuido a que sea mejores personas.

Contenido

Introducción	viii
Resumen	ix
Abstract	x
Capítulo 1. : Conceptos Básicos	1
1.1 Java	1
1.2 Tecnología JSF	1
1.3 Componentes Primefaces	2
1.3.1 Primefaces Mobile	2
1.4 Gestor de Base de Datos Postgres	4
1.5 Servidor de aplicaciones Jboss as	5
Capítulo 2. : Diseño	6
2.1 Diagramas de clases	6
2.2 Diagrama de Casos de Uso	10
2.3 Modelo Entidad – Relación	12
2.3.1 Diccionario de Datos	13
Capítulo 3. : Desarrollo	17
3.1 Mapeo de Entidades	17
Configuración del servidor de Aplicaciones JBoss	17
Mapeo de las Entidades	20
3.2 Creación de Managed Bean	26
3.2.1 Package	26
3.2.2 Managed Bean	27
3.2.3 EJB	28
3.3 Implementación con componentes Primefaces Mobile	31
3.3.1 Estructura de una página con soporte para dispositivos móviles	32
Capítulo 4. : Validación y Pruebas	33
4.1 Ejecutar el proyecto	33
4.2 Agregar Estilos	35
4.3 Validación y Pruebas	

Conclusiones y Recomendaciones	
Glosario	41
Bibliografía	42

Indice de Figuras

Figura 1 Clase Depósito	6
Figura 2 Clase Facultades	7
Figura 3 Clase Carrera	7
Figura 4 Clase Aspirante	8
Figura 5 Diagrama de Clases	9
Figura 6 Actualización de Información del Aspirante	10
Figura 7 Modelo Entidad Relación	12
Figura 8 Ilustración de Persistencia	17
Figura 9 Configuración Jboss, habilitar la vista de Servidor	18
Figura 10 Configuración Jboss, agregar Nuevo Server	18
Figura 11 Configuración Jboss, definir un nuevo servidor	19
Figura 12 Configuración Jboss, seleccionar ubicación del servidor	19
Figura 13 Configuración Jboss, vista del nuevo servidor agregado	20
Figura 14 Creación del proyecto JPA	20
Figura 15 Nombrar el Proyecto JPA	21
Figura 16 Creación de carpeta para alojar las entidades	21
Figura 17 Configuración del Proyecto JPA	22
Figura 18 Selección de la base de datos	22
Figura 19 Test de la conexión a la base de datos	23
Figura 20 Seleccionar una conexión a la base de datos	23
Figura 21 Mapear entidades desde las tablas	24
Figura 22 Selección de tablas para mapearlas	24
Figura 23 Relaciones entre tablas	25
Figura 24 Creación de entidades y sus atributos	25
Figura 25 Creación de Proyecto JSF	26
Figura 26 Package	27
Figura 27 Registro de un Managed Bean en la clase	27
Figura 28 Registro de un Managed Bean en el archivo faces-config.xml	27
Figura 29 Interacción de Session Bean con la base de datos	29
Figura 30 Consumir EJBs en un Managed Bean	30
Figura 31 Agregar librería Primefaces al proyecto JSF	31
Figura 32 Agregar soporte para dispositivos móviles	31
Figura 33 Estructura de página Primefaces Mobile	32
Figura 34 Configuración de ant/build.properties	33
Figura 35 Configuración de ant/build.xml	34
Figura 36 Persistence.xml	34
Figura 37 Vista de inicio de sesión del aplicativo	35
Figura 38 Crear una hoja de estilo para aplicación móvil	36
Figura 39 Agregar estilo a la página	37
Figura 40 Vista de Inicio de sesión del aplicativo con diseño	37
Figura 41 Pantalla de registro	38

Figura 42 Validaciones del sistema	
Figura 43 Guardar el Registro de Inscripción	

Indice de Tablas

Tabla 1 Gama de componentes de Primefaces Mobile	4
Tabla 2 Especificación de caso de uso: Actualización de información del Aspirante	11
Tabla 3 Diccionario de Datos Aspirante	13
Tabla 4 Diccionario de Datos Calificación	14
Tabla 5 Diccionario de Datos Carrera	14
Tabla 6 Diccionario de Datos Colegio	14
Tabla 7 Diccionario de Datos Depósito	15
Tabla 8 Diccionario de Datos Facultad	15
Tabla 9 Diccionario de Datos Teléfono	15
Tabla 10 Diccionario de Datos Teléfono Entidad	16
Tabla 11 Diccionario de Datos Tipo Teléfono	16

Introducción

Actualmente el tráfico de internet móvil está cada vez en aumento, y la creación de una navegación móvil sin problemas es uno de las mayores preocupaciones de cada Web Mater, pues independientemente de la conexión que los usuarios tengan, esperan una experiencia óptima en sus dispositivos móviles.

PrimeFaces Mobile ayuda en esta complicada tarea y permite crear rápidamente aplicaciones web para estos dispositivos móviles que se visualice bien en teléfonos independientemente de la plataforma que estos soporten.

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 en nueva y necesita tener soporte para dispositivos móviles, es por ello que con el presente trabajo se pretende explicar de forma sencilla la implementación de un proyecto web que sea compatible con dispositivos móviles, mediante el ejemplo de un prototipo aplicado a la inscripción para los nuevos alumnos de la Universidad del Azuay. En este trabajo se describe las configuraciones y los aspectos técnicos que se debe tener en cuenta para la creación e implementación de aplicaciones web móviles con componentes Primefaces Mobile.

Al finalizar se obtendrá el prototipo de inscripciones y una clara explicación sobre la funcionalidad de PrimeFaces Mobile con sus respectivos componentes, que servirán como guía para futuros proyectos basados en la tecnología JSF.

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.

ViagleeRadas UNIVERSIDAD DEL AZUAY Dpto. Idiomas

Lic. Lourdes Crespo

Capítulo 1. : Conceptos Básicos

En este capítulo se plantea los conceptos básicos sobre las herramientas y tecnologías que son necesarias para la realización de este proyecto.

1.1 Java

Java es un lenguaje de programación y una plataforma informática presentada por primera vez en el año de 1995 por SUN Microsystems. A más de ser un lenguaje de programación orientado a objetos, es altamente portable lo que permite crear aplicaciones que puedan ser ejecutadas en varias plataformas, además contiene un conjunto de tecnologías que hacen que el desarrollo de software sea rápido y como resultado aplicaciones más robustas y seguras. (Salazar, 2012)

Durante años, Java ha sido la tecnología de elección debido a los beneficios que ofrece:

- Independencia de la plataforma
- Facilidad de uso
- Bibliotecas completas que aceleran el desarrollo de aplicaciones
- Seguridad: tiene varios niveles de seguridad, desde el ámbito programador hasta la ejecución en la máquina virtual.
- Escalabilidad
- Amplio apoyo de la industria

1.2 Tecnología JSF

Java Server Faces es un *framework* estándar perteneciente a la industria JEE, creado para simplificar el desarrollo de aplicaciones web, el mismo que ofrece una amplia separación entre el comportamiento y la presentación de una aplicación. De esta forma, se logra dividir el trabajo de los desarrolladores, pudiendo un grupo encargarse de la lógica del negocio mientras otro grupo podría ocuparse de la parte de presentación de la aplicación, sin necesidad de saber a fondo como es el funcionamiento interno de la aplicación. Además, esta tecnología proporciona una amplia arquitectura para la gestión de componentes, datos de procesamiento, validación de datos de entrada y gestión de eventos. (Oracle, JavaServer Faces Technology Overview)

JSF se ejecuta como cualquier otra aplicación java web en un contenedor de servlets, además consta de las siguientes características:

- Conversión y validación automática de los datos de entrada proporcionadas por el cliente.
- Internacionalización.
- Oyentes de Eventos.

- Clases de ayuda por parte del servidor, como Beans de acceso a la base de datos.
- Biblioteca de etiquetas personalizada.
- Backing Beans, que definen las propiedades y funciones de los componentes de interfaz de usuario.

1.3 Componentes Primefaces

Primefaces es un conjunto de componentes que extienden del *framework* JSF de código abierto que cuenta con varias extensiones, el cual proporciona múltiples componentes que tienen capacidad Ajax sin necesidad de utilizar JavaScript. Primefaces tiene un solo archivo .jar llamado primefaces-{version}.jar que se puede descargar de forma fácil y rápida desde la página web de PrimeFaces. Para que este *framework* pueda ser ejecutado, lo único que necesita es una versión igual o superior de JAVA 5 y JSF 2.X

Este *framework*, trabaja con etiquetas propias, los mismos que envían peticiones generadas por el usuario al contenedor Ajax (Orellana, 2013). Como estas capacidades Ajax ya se encuentran inmersas en cada componente, el desarrollador puede concentrarse en aspectos de funcionamiento y visuales la poque de programar que atributo tendrá cada componente que necesite utilizar.

Ventajas:

- Contiene un conjunto de componentes (HTMLEditor, Dialog, Autocompletar, Gráficas, Calendarios, exportador de datos Excel, PDF, Word y muchos más).
- Soporte para Ajax, basándose en el estándar JSF 2.0 Ajax API.
- Es ligero, contiene un solo archivo .jar (3.1Mb en su versión primefaces-5.2.jar).
- No necesita configuraciones, cero dependencias.
- Soporte para interfaz de usuario móvil, posee un kit para crear aplicaciones web para móviles.
- Posee más de 35 temas de apariencia listos para usar.
- Cuenta con una amplia documentación.
- Tiene una gran comunidad de usuarios activos.

Desventajas:

- No se puede utilizar el soporte Ajax de JSF con los componentes de Primefaces
- Las nuevas versiones liberadas, ninguna hasta el momento ha tenido compatibilidad al 100% con las versiones anteriores, de esta forma se hace necesaria la reingeniería en el proyecto cada vez que se desea actualizar la versión de Primefaces

1.3.1 Primefaces Mobile

Primefaces Mobile (PFM) proporciona soluciones sofisticadas para la mayoría de los requisitos de las aplicaciones móviles. Construido sobre *jQuery Mobile*, que ofrece una gran cantidad de

componentes móviles listos que hacen un uso racional de la pantalla más pequeña, el mismo que proporciona una serie de eventos de toque que pueden ser utilizados para optimizar la experiencia móvil.

Además de la integración perfecta con jQuery Mobile, PFM cuenta con un RenderKit móvil para componentes populares PrimeFaces, extensiones marco ajax, eventos de comportamiento ajax móviles, modelo integrado de navegación, widgets de respuesta y más. (Primefaces)

1.3.1.1 Componentes

PrimeFaces Mobile presenta una gama de componentes que pueden ser utilizados para la creación de aplicaciones web móviles. Entre ellos se encuentran:

Componente	Descripción
Accordion	Las características notables son la selección múltiple, la carga dinámica, devoluciones de llamada de evento, el apoyo modelo de datos y fichas de movilidad reducida.
AutoComplete	Es un componente de entrada con la función de sugerencia
Button	Se utiliza para acciones Ajax, No Ajax y Navegaciones
Calendar	Componente de entrada para ingresar fechas
DataList	Permite visualizar una colección de datos en forma de lista, admite paginación, imágenes, encabezado y pie de pagina
DataGrid	Muestra los datos en un diseño de cuadricula
DataTable	
Dialog	Componente contenedor que puede superponer otros elementos de la pagina
Growl	Trae el widget de Mac acoplado en JSF con la capacidad de mostrar FacesMessages. Growl simplemente reemplaza al componente de mensajes
Link	Link es una extensión para el componente de enlace estándar con varias páginas de extensión navegación y navegaciones regulares.
Menu	Es un componente de navegación que consiste en submenús y menuitems.
OverlayPanel	Es un componente contenedor para mostrar el contenido, permite agregar efectos personalizados,

	posicionamiento(izquierda, derecha)				
Panel	Componente que permite visualizar contenido				
PanelGrid	Es una extensión de la panelGrid estándar con la integración temática y apoyo colspan-rowspan.				
TabMenu	Componente de navegación utilizado principalmente en secciones de encabeza y pie de pagina				
TabView	Permite cargar información por pestañas, entre sus características notables resaltan: carga dinámica, devolución de llamadas a un evento, fichas de movilidad reducida.				
Toolbar	Permite agrupar contenido en forma horizontal				
FileUpload	Permite cargar archivos desde el cliente navegador hacia el servidor.				

Tabla 1 Gama de componentes de Primefaces Mobile

1.4 Gestor de Base de Datos Postgres

Postgresql es un sistema de gestión de bases de datos objeto-relacional, su código fuente está disponible libremente, es distribuido bajo licencia BSD. Es la base de datos más potente en el mercado en cuanto a código abierto se refiere, está en constante evolución y en las últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales, ya que funciona muy bien con grandes cantidades de datos y soporta alta concurrencia de los usuarios accediendo a la vez al sistema. (Rafaelma, 2010)

Utiliza el modelo cliente-servidor y usa múltiples procesos en lugar de multihilos, de esta forma se garantiza que si un proceso falla no afectará a los demás procesos y el sistema seguirá funcionando, manteniendo así al sistema estable.

Entre sus características principales tenemos:

- Es una base de datos 100% ACID (Atomicity, Consistency, Isolation and Durability)
- Permite Integridad referencial
- Contiene *Tablespaces*
- Soporta distintos tipos de datos
- Soporta el almacenamiento de objetos binarios grandes
- Funciones/procedimientos almacenados

- Permite realizar copias de seguridad en caliente
- Completa Documentación
- Licencia BSD
- Está disponible para Windows, Linux, Unix, Mac OS X

1.5 Servidor de aplicaciones Jboss as

JBoss es un servidor de código abierto desarrollado netamente en Java puro, lo que permite ser utilizado en cualquier plataforma de sistema operativo que soporte java. Los desarrolladores principales trabajan para la empresa de servicios, JBoss Inc., que fue adquirida por RedHat en abril del 2006. Esta empresa fue fundada por Marc Fleury que es el creador de la primera versión de JBoss. Actualmente se encuentra apoyado por una red a nivel mundial de colaboradores. (Alférez, 2010)

Características:

- Producto de licencia de código abierto
- Cumple los estándares
- Confiable a nivel de empresa
- Incrustable, orientado a arquitectura de servicios
- Flexibilidad consistente
- Servicio de Middleware para cualquier objeto Java
- Soporte completo para JMX

Capítulo 2. : Diseño

El paso fundamental para el desarrollo de este prototipo es la especificación de requisitos del sistema, es por ello, que este capítulo estará enfocado a la recolección de información acerca del funcionamiento del software por parte del departamento de Sistemas de la Universidad, al diseño de diagramas de clases, diagramas de uso, modelo entidad-relación y el desarrollo del diccionario de datos.

2.1 Diagramas de clases

Estos diagramas muestran un conjunto de clases con sus atributos, métodos, además de la relación que tiene contras clases, proporcionado una perspectiva estática del sistema que se pretende desarrollar.

En este diagrama se podrá observar las clases que intervendrán en el sistema de inscripción de nuevos alumnos en la Universidad del Azuay. Después de la especificación de requisitos del sistema, se considera que intervendrán en el sistema son las siguientes clases: Alumno, Depósito, Facultad, Carreras. Se detalla a continuación las estructuras de las clases involucradas y sus respectivas relaciones con otras.

• Depósito: Es la entidad que contiene el comprobante de depósito bancario que haya realizado el aspirante, la fecha y un identificador interno.

Deposito
- dep_id - dep_codigo - dep_fecha
+ obtenerDepId + obtenerDepCodigo + obtenerDepFecha



• Facultades: Contiene un identificador interno, un código y el nombre de las facultades con las que cuenta la Universidad.



Figura 2 Clase Facultades

• Carreras: Contiene un identificador interno, un código y el nombre de las carreras que oferta la Universidad.



Figura 3 Clase Carrera

 Aspirante: Contiene los datos personales de la persona, como: número de Identificación, tipo de identificación, primer apellido, segundo apellido, primer nombre, segundo nombre, género, estado civil, fecha de nacimiento, calle principal, calle secundaria, numeración, referencia domiciliaria, correo electrónico, teléfono, carrera, depósito.

Aspirante			
- numeroldentificacion			
- tipoldentificacion			
- primerApellido			
- segundoApellido			
- primerNombre			
- segundoNombre			
- genero			
- estadoCivil			
- fechaNacimiento			
- callePrincipal			
- calleSecundaria			
- parroquia			
- referenciaDomicialiaria			
- correoElectronico			
- telefono			
- deposito			
- carrera			
+ obtenervumerolaentricacion()			
+ obtener (polentificacion)			
+ obtenerr/mer/peilido()			
+ obtenersegundoApeilloo()			
+ optener/menivomore()			
+ obtenersegundoivombre()			
+ obtenerGenero()			
+ obtener:Stadocovii() + obtener:EndeNaciminate()			
+ obtener/=chalvadmiento()			
+ obtendraler-micipal()			
+ obtenerGenergenergenergenergenergenergenerg			
+ objected a provide a pro			
• obtender/orgenElectronico()			
+ obtener Telefono()			
 operation et al. insertation et al. 			
 Insertai-indimensioneminatation, uponeminatation, primerizzana oscillaria, corresElectronica telefono) FechaNacimiento callePrincipal calleSecundaria parcoulia referenciaDomicializia, corresElectronica telefono) 			
recisionamiento, care mogra, careveduntaria, panoquia, recentratornationa, coneccessionno, reciono, reciono, a de associativa de astronomia			
- montaintainto ella Principal calle Sacindaria paraguita referencia Domini alla sacindaria della della della sacindaria paraguita referencia Domini alla Sacindaria paraguita referencia pa			
realization and a second and a period of a second and a second and a second a			

+ obtenerEstudiantePorCedula(cedula)

Figura 4 Clase Aspirante

En la Figura 5, se observa la relación que tiene la clase aspirante con carreras y depósito, y a su vez como la clase carreras se relaciona con facultad.



Figura 5 Diagrama de Clases

2.2 Diagrama de Casos de Uso

Este diagrama especifica como es el comportamiento del sistema, "un caso de uso es una secuencia entre un sistema y alguien o algo que usa alguno de sus servicios" (Ceria)



Figura 6 Actualización de Información del Aspirante

En este caso de uso, el aspirante puede ingresar su información o modificarla en el caso de que haya ingresado anteriormente, para ello debe pasar primero por un proceso de validación del código del depósito que debió haber realizado con anterioridad en una entidad Bancaria. Al grabar la información también debe pasar por un proceso de validación de los datos proporcionados.

L'specificación del d	aso de oso: Actualización de información del Aspirante			
Código	001			
Nombre	Actualización de información del aspirante			
Descripción	En este caso el aspirante ingresa la información solicitada por el sistema, para que			
	el sistema valide y guarde los cambios.			
Actores	Cliente			
Precondiciones	El aspirante debe proporcionar al sistema el código de depósito que efectuó en la			
	entidad bancaria.			
Pos condiciones	La información es validada y actualizada			
Flujo Normal	1. El aspirante ingresa al sistema			
	2. El aspirante ingresa/modifica la información			

Especificación del caso de Uso: Actualización de información del Aspirante

	3. El aspirante guarda la información
Flujo Alternativo	1. El aspirante ingresa al sistema
	2. El sistema verifica el código de depósito
	3. El sistema informa la no existencia del código de depósito
Excepciones	Código de depósito no existe
	abla 2 Especificación de caso de uso: Actualización de información del Aspirante

2.3 Modelo Entidad - Relación

El modelo entidad-relación es un método del que se dispone para diseñar los esquemas que posteriormente se va a implementar en una base de datos, de este modelo se obtiene un diagrama que ayuda a comprender los datos que van a ser utilizados y su respectiva relación entre ellos.



Figura 7 Modelo Entidad Relación

2.3.1 Diccionario de Datos

El diccionario de datos contiene "datos acerca de los datos" es decir, definiciones de otros objetos del sistema, en lugar de simples "datos en bruto" (DATE, 2001). A continuación se detalla cada uno de las entidades que se va a utilizar en el sistema de inscripciones.

Aspirante

Campo	Тіро	Nulo	Predeterminado	Comentarios
asp_id	Serial	No		Código interno, identificador del aspirante
asp_tipo_identificacion	Varchar(1)	No		Tipo de identificación (cédula=1, pasaporte = 3)
asp_identificacion	Varchar(20)	No		Número de identificación
asp_primer_nombre	Varchar(100)	No		Primer nombre del aspirante
asp_segundo_nombre	Varchar(100)			Segundo nombre del aspirante
asp_primer_apellido	Varchar(100)	No		Primer apellido del aspirante
asp_segundo_apellido	Varchar(100)			Segundo apellido del aspirante
asp_genero	Varchar(1)	No		Género del aspirante (femenino=1, masculino = 2)
asp_estado_civil	Varchar(1)	No		Estado civil(soltero=1, casado=2, divorciado=3, viudo=4, unión
				libre=5)
asp_fecha_nacimiento	Date	No		Fecha de nacimiento del aspirante
asp_calle_principal	Varchar(100)	No		Domicilio del aspirante
asp_calle_secundaria	Varchar(100)	No		Domicilio del aspirante
asp_calle_numero	Varchar(20)			Domicilio del aspirante
asp_calle_referencia	Varchar(500)			Domicilio del aspirante
asp_correo_electronico	Varchar(100)	No		Dirección electrónica del aspirante
dep_id	Integer	No		Identificación del depósito
col_id	Integer	No		Identificación del colegio
par_id	Integer	No		Identificación de la parroquia
car_id	Integer	No		Identificación de la carrera

Tabla 3 Diccionario de Datos Aspirante

Calificación

Campo	Тіро	Nulo	Predeterminado	Comentarios
cal_id	Serial	No		Código interno, identificador de las calificaciones del aspirante
cal_notaprimero_bach	Decimal(15,4)	No		Nota del primero de bachillerato
cal_baseprimero_bach	Decimal(15,4)	No		Base del primero de bachillerato
cal_notasegundo_bach	Decimal(15,4)	No		Nota del segundo de bachillerato
cal_basesegundo_bach	Decimal(15,4)	No		Base del segundo de bachillerato
cal_notatercero_bach	Decimal(15,4)	No		Nota del tercero de bachillerato
cal_basetercero_bach	Decimal(15,4)	No		Base del tercero de bachillerato
asp_id	Integer	No		Aspirante al que pertenece las notas

Tabla 4 Diccionario de Datos Calificación

Carrera

Campo	Тіро	Nulo	Predeterminado	Comentarios
car_id	Serial	No		Código interno, identificador de la carrera
car_codigo	Varchar(15)	No		Código de la carrera
car_fecha	Varchar(200)	No		Nombre de la carrera

Tabla 5 Diccionario de Datos Carrera

Colegio

Campo	Тіро	Nulo	Predeterminado	Comentarios
col_id	Serial	No		Código interno, identificador del colegio
col_codigo	Varchar(15)	No		Código del colegio
col_fecha	Varchar(500)	No		Nombre del colegio

Tabla 6 Diccionario de Datos Colegio

Depósito

Campo	Тіро	Nulo	Predeterminado	Comentarios	
dep_id	Serial	No		Código interno, identificador del depósito	
dep_codigo	Varchar(15)	No		Código del depósito	
dep_fecha	Date	No		Fecha en que se realiza el depósito	
Tabla 7 Diccionario de Datos Depósito					

Facultad

Campo	Тіро	Nulo	Predeterminado	Comentarios
fac_id	Serial	No		Código interno, identificador de la facultad
fac_codigo	Varchar(15)	No		Código de la facultad
fac_nombre	Varchar(200)	No		Nombre de la facultad

Tabla 8 Diccionario de Datos Facultad

Teléfono

Campo	Тіро	Nulo	Predeterminado	Comentarios
tel_id	Serial	No		Código Interno, identificador del teléfono
tel_numero	Varchar(15)	No		Número de teléfono
tte_id	Integer	No		Tipo de teléfono(convencional, móvil, etc.)

Tabla 9 Diccionario de Datos Teléfono

Teléfono_Entidad

Campo	Тіро	Nulo Predeterminad	o Comentarios		
ten_id	Serial	No	Código interno, identificador del teléfono entidad		
tel_id	Integer	No	Identificador del teléfono		
asp_id	Integer	No	Identificador del aspirante		
Tabla 10 Diccionario de Datos Teléfono Entidad					

Tipo Teléfono

Campo	Тіро	Nulo	Predeterminado	Comentarios
tte_id	Serial	No		Código interno, identificador del tipo de teléfono
tte_nombre	Varchar(200)	No		Nombre del tipo de teléfono (convencional, móvil, trabajo)

Tabla 11 Diccionario de Datos Tipo Teléfono

Capítulo 3. : Desarrollo

En este capítulo se desarrollará el sistema de inscripciones, utilizando la herramienta de desarrollo Eclipse, la base de datos Postgres y el servidor de aplicaciones JBoss, los mismos que previamente ya han sido instalados en la terminal de trabajo.

3.1 Mapeo de Entidades

Para el mapeo de entidades, primero se abordará el término persistencia, que consiste en pasar la información ya sea desde los objetos a la base de datos y desde la base de datos hacia los objetos. Para determinar en qué lugar se debe guardar la información de los objetos o en qué objetos se debe recuperar la información desde la base, es necesario establecer un mapeo objeto relacional.

El mapeo consiste en establecer una relación entre el mundo de los objetos y el mundo de las bases de datos, como se muestra en el siguiente gráfico:



Figura 8 Ilustración de Persistencia

Para mapear las entidades se utilizará la herramienta Eclipse y el servidor de aplicaciones JBoss, a continuación se detalla los pasos a seguir para alcanzar este objetivo:

Configuración del servidor de Aplicaciones JBoss

Lo primero que se debe configurar es el servidor JBoss dentro de Eclipse, para ello:

• Escoger del menú la opción "Window", "Show View", "Server". Si no aparece dentro de las opciones presentadas, seleccionar "Other" y en las nuevas opciones presentadas buscar "Server" y aceptar.

			and the second second	-	-	
Window	/ <u>H</u> elp					
Ne Ne	w Window w Editor	4	• 🕂 🗮 💩 🖉 /	🖋 🕶 📔 👘 🖄	• 🖗 • 🖗 🔶	
Ор	en Perspective	E			_	Z Sauch Barr Community
Sh	ow View 🕨	*	Ant			Search boss community
Cu Sav Cic Cic Na Pre	stomize Perspective ve Perspective As set Perspective see Perspective see All Perspectives vigation therences		Console Declaration Navigator Outline Package Explorer Palette Problems Progress	Alt+Shift+Q, C Alt+Shift+Q, D Alt+Shift+Q, O Alt+Shift+Q, P Alt+Shift+Q, X	- 1 - % 8 &	News News News Discourse Control Contro Control Control Control
	Documentation New and Noteworthy Reference EAQ Screencasts Settings Show on Startun		Properties Seam Components Search SVN Resource History Synchronize Tasks Type Hierarchy Web Projects	Alt+Shift+Q, S Alt+Shift+Q, Y Alt+Shift+Q, T		Blogs Blogs Blogs Boss Tools 41.2 Final and Developer Studio 7.1.1.GA released 1 year ago by Fred Britonia On behalf of the JBoss Tools and Developer Studio taam, I am extremely proud to announce the general availability of the JBoss Tools 41.2 Final Boss Tools going to the Moon 2 years ago by Max Rydahl Andersen Its time to start the milestones counting up to the next Eclipse release train this summer. Here is the finit steed to work Alpha release trajecting
	•		Other	Alt+Shift+Q, Q		

Figura 9 Configuración Jboss, habilitar la vista de Servidor

• Una vez habilitada la Vista de Servidor, aparecerá en la parte inferior una nueva pestaña, la cual indica que la opción se encuentra habilitada pero que aún no se ha configurado ningún servidor en el área de trabajo.

Help 8 🞯 • 🛛 🖏 • 🔶 🦚 C JBoss Central 🕸) 🛱 🙋 🇀 🖋 🕶 目 🗐 🖉 🗸 🖓 🗸 🗇	• ⇔ • C 🗄 🚱 Web Develop
울 ⓒ ▼ 월 ▼ ▶ 兆 @ 종 Ç JBoss Central &) 🛱 🍅 😂 🖋 🔹 📓 🗐 🕈 🖗 수	🔹 🗘 🖈 🧯 🚱 Web Develop
🕻 JBoss Central 🛿		
🕻 JBoss Central		🎽 Search JBoss Community 💿 🌟 🖇
▼ Create Projects	: •	🔻 News 🔊 💟 🖑
Image: Second Strain	 3 OpenShift Application 12 Java EE Project 12 RichFaces Project 3 GWT Web Project 	This week in JBoss (27nd August 2015): Find your docking point and get started ! 2 days ago by Romain Pelicse In a few days, most kids will be back to school, if not already, and holidays season will come to an end (for now) It could be a depressing This week in JBoss (22nd August 2015): Time for a Double Dose 6 days ago by Kevin
 Project Examples JBoss Quickstarts 	1 19 1 1	Conner Welcome to another edition of the JBoss Weekly Editorial, this week bringing you a double dose of news as we try to catch-up on a selection of
▼ Documentation		🔻 Blogs 🦷 🦃
New and Noteworthy User For Reference Develop FAQ Wiki Screencasts Issue Tra	rum per Forum racker	Boss Tools 4.1.2.Final and Developer Studio 7.1.1.GA released 1 year ago by Fred Bricon On behalf of the JBoss Tools and Developer Studio team, I am extremely proud to announce the general availability of the JBoss Tools 4.1.2.Final
✓ Settings Ø Show on Statun		<u>Boss Tools going to the Moon!</u> 2 years ago by Max Rydahl Andersen Its time to start the milestones counting up to the next Eclipse release train this summer. Here is the first tested to work Alpha release trageting
<	m	
🧕 Getting Started 💲 Software/Upd. 🕺 Problems 🕢 Tasks 🌸 Ant 🖳 (No servers available. Define a new serv	Jate Console 🖼 Progress 📇 Synchronize 🔗 Search 🏭 SVN ver from the <u>fraw server witzerd</u>	N Resource History 🕷 Servers 🕸 📄 🌣 🔘 🖉 🗮 🕮 🌣 🗖

Figura 10 Configuración Jboss, agregar Nuevo Server

 Hacer click sobre "New Server Wizard", el cual presenta los servidores disponibles de acuerdo a la versión de Eclipse que se tenga instalada y las respectivas actualizaciones que se haya realizado al mismo, para el ejemplo, se escogerá la opción JBoss AS 7.1, en el cual se debe ingresar el nombre del servidor y en la opción de "Server runtime environment" escoger la opción "Add"

	New Server	×
Help	Define a New Server Choose the type of server to create	E 🞯 Web Develop »
Create Projec Create Projec Dynamic W Dynamic W Dava EE We HTML5 Pro Spring MYC Project Exam	Download additional server addaters type filter text # Boos A5 10 # Boos A5 50 # Boos A5 70 # Boos A5 70 # Boos A5 70 # Boos A5 70	▼ Search IBors Community ③ ★ ★ Siss (27nd August 2015): Find your docking point and get started [2 2 pr Palsise at bids will be back to school (if not already, and holidays season will or nonw) Ht cold be a degressing 0 or sol2nd August 2015): Time for a Double Dase 6 days ago by Kevin her edition of the JBors Weekly Editorial, this week bringing you a 1
 JBoss Quic 	JBoss Application Server 7.1	ews as we try to catch-up on a selection of
Documentat New and Notes Reference EAQ Screencasts Sotting	Server's host name: ⁰ locafhost Server name: JBoss 7.1 Runtime Server Server runtime environment: JBoss 7.1 Runtime <u>v</u> <u>Add.</u> <u>Configure runtime environments.</u>	2.Final and Developer Studio 7.1.1.GA refeased 1 year ago by Fred Boss Tools and Developer Studio 7.1.1.GA refeased 1 year ago by Fred Bear Javaibility for the Ross Tool 4.12.Final.
Settings Schw on Sta Getting Starte Getting Starte No servers availab	(Back Next > Finish Cancel	e miletones counting up to the not Eclipte relaset train this he first tested to work Alpha release trageting

Figura 11 Configuración Jboss, definir un nuevo servidor

• En la nueva pantalla, en la opción "Home Directory" seleccionar la ubicación de la carpeta del servidor Jboss y Finalizar.

New Server Runtin	me Environment	• x
JBoss Runtime JBoss Application S	erver 7.1	BOSS [®] by Red Hat
A JBoss Server runtir It can be used to set as well as by a "serve	me references a JBoss installation directory. up classpaths for projects which depend on this runtime, r° which will be able to start and stop instances of JBoss.	
Name		
JBoss 7.1 Runtime	1	
Home Directory		
C:\jboss-as-7.1.1.F	inal	Browse
JRE		
Default JRE for Java	aSE-1.6	▼ JRE
Configuration file:	standalone.xml	Browse
?	< Back Next > Finish C	ancel
?	< Back Next > Finish C	ancel

Figura 12 Configuración Jboss, seleccionar ubicación del servidor

 Se regresa nuevamente a la configuración inicial y a partir de esta pantalla se avanza presionando el botón "Next" en las pantallas posteriores hasta llegar al botón Finalizar. Terminado este proceso, se podrá observar en la vista "Servers" que ya consta el servidor que se ha configurado.

<u>User Forum</u>
ware/Update
Ant 🗐 Console 🗝 Progress 🖆 Synchronize 🔗 Sear
erver [Stopped]

Figura 13 Configuración Jboss, vista del nuevo servidor agregado

Mapeo de las Entidades

• En el entorno de desarrollo Eclipse, crear un nuevo proyecto JPA y presionar en el botón "Next"

New New	
Select a wizard Create a JPA project	
<u>Wi</u> zards: JPA	R.
PA G. JPA G. JPA Entities from Tables G. JPA Entity JPA Project JPA ORM Mapping File JPA Project	
(<u>Back</u>) <u>Next ></u> (<u>Finish</u>)	Cancel

Figura 14 Creación del proyecto JPA

 A continuación, se presenta una nueva pantalla en la que se debe ingresar el nombre del proyecto, en la opción "Target runtime" seleccionar el servidor que se configuró con anterioridad y en la opción de "JPA version" seleccionar la opción 2.0 y presionar en "Next".

New JPA Project	_ D X
JPA Project Configure JPA project settings.	
Project name: Inscripciones_Mapeo	
Project location	
Use <u>d</u> efault location	
Location: C:\Users\Owner\Documents\eclipseWorkspace\Inscripcion	Browse
Target r <u>u</u> ntime	
JBoss 7.1 Runtime 1	New <u>R</u> untime
JPA version	
2.0	•
Configuration	
Default Configuration for JBoss 7.1 Runtime 1	Mod <u>i</u> fy
A good starting point for working with JBoss 7.1 Runtime 1 runtime. Add can later be installed to add new functionality to the project.	itional facets
EAR membership	
Add project to an EAR	
EAR project name:	New <u>P</u> roject
Working sets	
Add project to working sets	
Working sets:	S <u>e</u> lect
(?) < Back Next > Finish	Cancel

Figura 15 Nombrar el Proyecto JPA

• En la siguiente opción, se deja todas las configuraciones que presenta el sistema por defecto y se presiona continuar.

New JPA Project	_ D X
Java Configure project for building a Java application.	
Source folders on build path:	
C Sec	<u>Add Folder</u> <u>É</u> dit <u>R</u> emove
Default output folder:	
buildclasses	
? < <u>Back</u> <u>Next</u> <u>Finish</u>	Cancel

Figura 16 Creación de carpeta para alojar las entidades

• En la siguiente pantalla escoger la Plataforma: "Generic", "JPA implementation Type: Library Provide by Target Runtime" y en "Connection" como no se tiene registrada ninguna configuración a la base de datos seleccionar "AddConnection".

New JPA Project			_ D X
PA Facet			
Configure JPA settings.			
Platform			
Generic 2.0			•
JPA implementation			
Type: Library Provided by Target Runti	me		•
The targeted runtime is able to provide option will configure the project to use	the library require that library.	d by this facet. !	Selecting this
Connection			
<none></none>			•
		A	Connection
Add driver library to build path			Connect
Driven			~
Override default catalog from conne	ction		
Catalog:			Ţ
Override default schema from conne	ection		
Schema:			
Persistent class management			
Oiscover annotated classes automati	cally		
Annotated classes must be listed in p	ersistence.xml		
Create <u>m</u> apping file (orm.xml)			
? <u>B</u> ack	Next >	Einish	Cancel

Figura 17 Configuración del Proyecto JPA

• Para configurar una nueva conexión a la base de datos, se tiene que buscar el tipo de conexión, en este caso se utilizará PostgreSQL, asignar el nombre de la conexión y presionar en "Next".

Connection Profile Create a PostgreSQL connection profile. Connection Profile Types: type filter teat Google Cloud SQL Hoformix Bingres DBC Database Connection for Query Builder MaxBB MySQL Goracle PostgreSQL SQL Server SQL Server SQL Server SQL Server SQL Server Sybase ASA Sybase ASA S	New Connection Profile	
Create a PostgreSQL connection profile.	Connection Profile	
Connection Profile Types: Type filter text © Google Cloud SQL © Hoformix © Informix © DBC Database Connection for Query Builder © MaxDB © MaxDB © MaxDB © MaxDB © Social Call © Social Call © Social Call © Sybase ASE © Sybase ASE © ConecionPostgres Description (optional):	Create a PostgreSQL connection profile.	
type filter teat	Connection Profile Types:	
Google Cloud SQL Hindrmix Bindrmix Bindrmix Bindrmix Bindrmix Bindrmix Boc Database Connection for Query Builder MaxBit Moscal PostgreSQL SQLiserver Solybase ASA Sybase ASE	type filter text	
Informix Informix Ingres DBC Database Connection for Query Builder MaxDB MaxDB MaxDB SQL PortgreSQL SQL Server SQL Server Sybase ASA Sybase ASE	Google Cloud SQL	
Informix Ingres IDBC Database Connection for Query Builder IMA:08 IMA:09 IMA:08 IMA:08 <	Image: Barbara Barbar Barbara Barbara Barb	
Images Images <td< td=""><td>💕 Informix</td><td></td></td<>	💕 Informix	
Image: Connection for Query Builder Image: Concle Image: Concle </td <td>Ingres</td> <td></td>	Ingres	
Image Image Image ConecionPostgres Description (optional):	JDBC Database Connection for Query Builder	
Impose Impose <td< th=""><th>MaxDB</th><th></th></td<>	MaxDB	
Image: Conceleration Image: Conceleration	MySQL	
PostgreSQL Image: ConcessionPostgres SqLite Sybase ASE Sybase ASE Image: ConcessionPostgres Recription (optional): Image: ConcessionPostgres	U Oracle	
© SQL Server Sybase ASA Sybase ASE ConexionPostgres Description (optional):	PostgreSQL	
Sybase ASA Sybase ASA Sybase ASE Name: conexionPostgres Description (optional):	SQL Server	
Sybase ASA Sybase ASE Name: conexionPostgres Description (optional): ? < Rack	SQLite	
Sybase Ase Name ConexionPostgres Description (optional):	Sybase ASA	
Name ConexionPostgres Description (optional):	Sybase ASE	-
conexionPostgres Description (optional): () < Rack Next> Einish Cancel	Na <u>m</u> e:	
@escription (optional): ? < Rack	conexionPostgres	
? < <u>Rack</u> <u>Next></u> Einish Cancel	Description (optional):	
? < Back Next> Einish Cancel		
? < Back Net> Einish Cancel		
? < gack Next> Einish Cancel		
(<u>Back</u> <u>Next></u> <u>Finish</u> Cancel		
Cancel		
	(?) < <u>Back</u> <u>Next</u> <u>Fin</u>	ish Cancel

Figura 18 Selección de la base de datos

• Se deberá ingresar la información correspondiente a la conexión con la base de datos que se va a utilizar y probar la conexión presionando "Test Connection", si todos los datos proporcionados están correctos, este test devolverá un mensaje indicando que la conexión

es exitosa. Presionar "Next" y en el recuadro presentado a continuación presenta un resumen de la conexión que va a crearse. Si todo es correcto presionar en "*Finish*".

S New DEC Convertion Profile	New JDIC Connection Pre	22	
Specify a Driver and Connection Details Select a driver from the drop-down and provide login details for the connection.	Summary Information pathered from	previous pages.	
Divers PotgedQLOBCDiver Properties General Database inscripciones Upt potgenty-explicitional- Upt name potgen Patagord Sign paramet Potged Prog successed Prog successed Prog successed Inst Completes Inst Complete Inst C	Property Home Decolption Auto connect on finish Detailsaus URL Sent Password	Value considerPortgres false true inscripciones jotecportgrespl//tocahost5032/inscripci. true	
Connect every time the workbanch is glated		gek. jer - Enit	Cancel

Figura 19 Test de la conexión a la base de datos

 Nuevamente se regresa a la pantalla que se presentó al inicio y como se observa en la Figura 20, en la opción "Connection" ya está seleccionada la conexión que seacaba de configurar. Para terminar presionar en el botón "Finish". De esta forma se crea un proyecto JPA que servirá para alojar las entidades en base a las tablas que se necesitarán para el desarrollo del proyecto y que han sido creadas con anterioridad en la base de datos.

New JPA Project	t.			
JPA Facet				_
Configure JPA set	ttings.			
<u>P</u> latform				
Generic 2.0				•
JPA implementat	tion			
Type: Library Pr	ovided by Target Ru	ntime		•
The targeted run option will config	time is able to provi gure the project to u	de the library req se that library.	uired by this facet. S	Selecting this
<u>C</u> onnection				
conexionPostgre	es			•
			A	dd connection
				Connected
Add driver lib	rary to build path			
Driver: Postgr	eSQL JDBC Driver			_
Override defa	ult catalog from con	inection		
Catalog: inscrip	iciones			Ψ
Override defa	ult schema from cor	nnection		
Schema: public				
Persistent class n	management			
Discover anno	otated classes a <u>u</u> tom	atically		
O Annotated cla	asses must be listed i	n persistence.xm	l i i i i i i i i i i i i i i i i i i i	
Create <u>m</u> appir	ng file (orm.xml)			
?	< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel

Figura 20 Seleccionar una conexión a la base de datos

• Una vez creado el proyecto JPA, el siguiente paso es mapear las entidades, es decir, convertir las tablas de la base de datos en objetos, para esto: escoger el proyecto JPA, hacer click derecho, escoger "New" y la opción JPA "Entities from Tables".



Figura 21 Mapear entidades desde las tablas

A continuación presenta un nuevo cuadro de configuración, que como se ve en la Figura 22, ya se encuentra seleccionada la conexión a la base de datos que se realizó en los pasos anteriores, por lo tanto, también se puede ver las todas las tablas que esta contiene. Seleccionar las tablas con las que se va a trabajar y presionar "Next".

🐑 Generate C	ustom Entities	x
Select Table Select tables	s to generate entities from.	
Connection:	conexionPostgres	•
Schema:	public	1
Isples	I sejonte III calificacion III calificacion III calificacion III parcquia III teófono III teófono III teófono III teófono	
🕑 Update cl	ss list in persistence.xml Restore Defaults]
?	< <u>Rack</u> Next > Einish Cano	el

Figura 22 Selección de tablas para mapearlas

• Seleccionadas las tablas con las que se pretende trabajar, se muestra la relación que tienen entre las mismas, en esta opción se puede agregar más asociaciones en caso de que fueran necesarias.

Generate Custom Entities	- • • ×
Table Associations Edit a table association by selecting it and modifying the controls in the editing pa	nel.
Table associations Table associations Leach acolegio has many aspirante. Leach acolegio has many aspirante. Leach acrolegio has many aspirante. Leach aproquia has many aspirante. Leach aproquia has many aspirante. Leach aproquia has many calificacion. Leach aspirante has many telefono. Leach apprante has many telefono. Leach telefono has many telefono. Leach telefon	
(?) < Back Next > Einish	Cancel

Figura 23 Relaciones entre tablas

 Por último, se detalla cada entidad que se va a generar con sus respectivos atributos, por defecto el sistema presenta un nombre sugerido tanto para las entidades como para sus atributos, los mismos que se podría modificar el nombre y el tipo de dato en el caso de que se considere necesarios.

Generate Custom Entities	X
Customize Individual Entities	
Iables and columns	
Garapiante G	•
Generate this property Column mapping	
Property name: aspld	
Mapping type: Integer	-
Mapping kind:	•
 ✓ Column is updatable ✓ Column is insertable 	
Domain Java Class	
Getter scope: public protected private	
Setter scope: ve public ve protected private	
(?)	Cancel

Figura 24 Creación de entidades y sus atributos

3.2 Creación de Managed Bean

Una vez mapeadas las entidades, se procede a crear un proyecto Web, en el cual se desarrollará todo el módulo de inscripciones para los nuevos alumnos de la Universidad, con versión para dispositivos móviles. Para esto, escoger desde el menú la opción de "New", "JSF Proyect" y asignar un nombre al proyecto, seleccionar de la opción "JSF Environment" la versión "JSF 2.0", indicarle el servidor que se utilizará para ejecutar el proyecto, que en este caso es el JBoss 7.1 que anteriormente se configuró.

New JSF Project	N Appendix 10 percent	_ D X
Create JSF Projec	t	\bigcirc^\diamond
Project Name*	Inscripcion	
	✓ Use default path*	
Location*	s\Owner\Documents\eclipseWorkspace\Inscripcion	Browse
JSF Environment*	JSF 2.0	•
Template*	JSFBlankWithoutLibs	•
?	< Back Next > Finish	Cancel

Figura 25 Creación de Proyecto JSF

3.2.1 Package

Un *package* sirve para organizar clases afines, este es opcional, pero es una buena práctica de programación mantener las clases organizadas entre sus similares. Puede almacenar varias clases, de igual manera, un proyecto puede contener el número de package que considere necesarios. Una clase debe tener un nombre único dentro de cada package y puede utilizar otras clases que se encuentren dentro del mismo package o en otros.

Para el desarrollo de este proyecto, de necesitaran 6 package que organizarán las clases entre controladores, convertidores, entidades, servicios, utilidades y validadores, como se muestra a continuación:

Inscripciones
 JavaSource
 inscripciones.controladores
 inscripciones.conventidores
 inscripciones.entidades
 inscripciones.servicios
 inscripciones.utils
 inscripciones.validadores

Figura 26 Package

En el paquete *inscripciones.entidades* se copiará las entidades que se mapeó anteriormente en el proyecto JPA.

3.2.2 Managed Bean

Un Managed Bean es una clase java que no extiende de ninguna otra clase, sino que posee una anotación la cual hace que se construya y se inyecte en el marco de JSF. Un Managed Bean se registra con la anotación @ManagedBean sobre el nombre de la clase o se puede registrar en el archivo "faces-config.xml".

```
@ManagedBean(name = "mntAspiranteControlador")
@ViewScoped
public class MntAspiranteControlador {
}
```

Figura 27 Registro de un Managed Bean en la clase

```
<managed-bean>
        <managed-bean-name>mntAspiranteControlador</managed-bean-name>
        <managed-bean-class>inscripciones.controladores.MntAspiranteControlador</managed-bean-class>
        <managed-bean-scope>view</managed-bean-scope>
        </managed-bean>
```

Figura 28 Registro de un Managed Bean en el archivo faces-config.xml

3.2.2.1 Alcance de un Managed Bean

El alcance de un *Managed Bean* describe el periodo de tiempo en que una instancia de un *Managed Bean* está disponible para otras partes del programa. Incluye los siguientes tipos de alcances:

@NoneScoped

Bean no gestionado, se crea una instancia según la demanda de un Bean gestionado y se conservará mientras el Bean gestionado exista.

- @RequestScope
 Se instancia bajo cualquier petición HTTP y existirá hasta que exista una petición HTTP diferente a la que se creó.
- @ViewScoped
 Este Bean se crea y estará disponible mientras el usuario permanezca en la misma vista.
- @SessionScoped
 Este Bean se almacena en una sesión HTTP, es decir el Bean existirá más allá de una petición HTTP.
- @ApplicationScoped
 Este Bean se mantiene a lo largo de la duración de la aplicación y está disponible para todos los usuarios.
- @CustomScoped
 Define una implementación propia de java.util.Map, generalmente incluidas en marcos de trabajos que extiende JSF.

3.2.3 EJB

Un servidor JEE como JBoss, es un servidor de aplicaciones que tienen contenedor web y contenedor de EJBs. La ventaja de utilizar EJBs es utilizar todo lo que el servidor ya tiene programado, en lugar de partir desde cero. Por ejemplo si se usara una clase java plana para conectar al sistema con la base de datos, se debe crear código para abrir la conexión, cerrar la conexión, si tiene alta concurrencia manejar un pool de conexiones, etc. Este manejo de conexiones ya lo hace el servidor de aplicaciones y el desarrollador lo puede utilizar.

Para este proyecto se utilizará el tipo de EJB llamado Session Bean que a su vez este tiene 3 tipos que se distinguen entre sí por el ciclo de vida que manejan:

- *Stateless*: no mantienen estado entre invocaciones.
- Stateful: guardan estado entre invocaciones.
- *Singleton*: crea una sola instancia que es invocada por muchos clientes.

Para crear un Session Bean, se debe crear una clase java y agregarle la anotación @Stateless, @Stateful o @Singleton dependiendo el caso.

3.2.3.1 EntityManager

La clase Java que contiene los métodos para interactuar con la base de datos es la clase *EntityManager*, que tiene los métodos para insertar, actualizar, eliminar registros en la base y también para realizar consultas. El *EntityManager*, al igual que un *ManagedBean* o un *Session Bean* no se instancia nunca, se crea utilizando inyección de código con la anotación @PersistenceContext.

```
цΤ
12 @Stateless
13 public class ServicioAspirante {
14⊝
       @PersistenceContext
15
       private EntityManager em;
16
17⊝
       public Aspirante insertar(Aspirante aspirante) {
18
            em.persist(aspirante);
19
            em.refresh(aspirante);
20
            return aspirante;
21
       }
22
       public Aspirante actualizar(Aspirante aspirante) {
23<del>0</del>
24
            em.merge(aspirante);
25
            return aspirante;
       }
26
27
       public void eliminar(Aspirante id) {
28⊝
29
            em.remove(em.merge(id));// hace el select y despues elimina
30
        3
31
320
       public Aspirante obtnerAspirante(int id) {
33
            try {
34
                Query q = em
                        .createQuery("Select c from Aspirante c where c.id = :parametro");
35
                q.setParameter("parametro", id);
36
37
                return (Aspirante) q.getSingleResult();
38
            } catch (NoResultException ex) {
39
                return new Aspirante();
40
            3
41
       }
42
43⊕
       public Aspirante obtnerAspirantePorDeposito(Deposito dep) {[]
53
54 }
```

Figura 29 Interacción de Session Bean con la base de datos

3.2.3.2 Consumir un Session Bean desde un Managed Bean

Para consumir un *Session Bean* desde un *Managed Bean* se debe declarar un atributo del tipo del Session Bean que se va a utilizar, se lo anota con @EJB, esto hará que a través de la inyección de código se pueda utilizar una instancia de *Session Bean*. Una vez declarada la variable y agregada la anotación, se puede utilizar los métodos que cada *Session Bean* declarado contenga.

```
@ManagedBean(name = "mntAspiranteControlador")
@ViewScoped
public class MntAspiranteControlador {
    @EJB
    private ServicioCalificacion calificacionServ;
@EJB
    private ServicioAspirante aspiranteServ;
    @EJB
    private ServicioCarrera carreraServ;
    @EJB
    private ServicioColegio colegioServ;
    @EJB
    private ServicioParroquia parroquiaServ;
    @ManagedProperty("#{iniciarSessionControlador}")
    private IniciarSessionControlador sessionController;
    private Aspirante aspirante;
    private Calificacion calificacion;
    private List<Carrera> listaCarreras;
    private Carrera carreraSeleccionada;
    private Colegio colegio;
    private Parroquia parroquia;
    private Integer codigoCarrera;
public static String colegioSeleccionadoString;
    public static String parroquiaSeleccionadoString;
    private Boolean banderaIdentificacion;
    public MntAspiranteControlador() {
    @PostConstruct
    public void init() {
        FacesContext.getCurrentInstance().getExternalContext().getSession(true);
        aspirante = new Aspirante();
        calificacion = new Calificacion();
        colegio = new Colegio()
        parroquia = new Parroquia();
        listaCarreras = new ArrayList<Carrera>();
        listaCarreras = carreraServ.obtenerCarreras();
        carreraSeleccionada = new Carrera();
        codigoCarrera = 0;
        aspirante = aspiranteServ.obtnerAspirantePorDeposito(sessionController
                 .getDeposito());
        cargarDatosAspirante();
    З
```

Figura 30 Consumir EJBs en un Managed Bean

3.3 Implementación con componentes Primefaces Mobile

Para utilizar los componentes de Primefaces lo primero que hace, es agregar la librería al proyecto, para ello, se puede descargar desde la página web oficial en la siguiente dirección: <u>http://primefaces.org/downloads</u> que presenta descargas tanto en la versión *Community* que es gratuita o la *Elite* que necesita registrarse como usuario y pagar por la membresía, esta consta de mejoras significativas en cuanto a la versión libre y cuenta con soporte de parte del equipo de Primefaces.

Descargada la librería *primefaces-version.jar*, agregar al proyecto dentro de la carpeta *WebContent/WEB-INF/lib*, de esta forma el proyecto está listo para desarrollar una aplicación JSF con implementación de Primefaces, no se necesitan bibliotecas adicionales, pues dentro de la biblioteca central de Primefaces contiene el soporte para móviles.



Figura 31 Agregar librería Primefaces al proyecto JSF

Para definir una página con soporte para dispositivos móviles, se crea un archivo XHTML normal y se agrega el siguiente namespace correspondiente a Primefaces y Primefaces Mobile:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
20@html xmlns="http://www.w3.org/1999/xhtml"
3 xmlns:h="http://java.sun.com/jsf/html"
4 xmlns:f="http://java.sun.com/jsf/core"
5 xmlns:ui="http://java.sun.com/jsf/facelets"
6 xmlns:p="http://primefaces.org/ui"
7 xmlns:pm="http://primefaces.org/mobile">
```

Figura 32 Agregar soporte para dispositivos móviles

3.3.1 Estructura de una página con soporte para dispositivos móviles

Los componentes como: textos, combos, autocomplete, mensajes, áreas de texto, radio button, entre otros, son utilizados de los que proporciona Primefaces y para dotar de un ambiente móvil a la aplicación, se debe agregar las etiquetas que se muestran en la siguiente.



Figura 33 Estructura de página Primefaces Mobile

Como se observa en la **¡Error! No se encuentra el origen de la referencia.**, lo primero que se efine es la página, dentro de un archivo XHTML se puede definir cuantas *<pm:page>* sean necesarias, las mismas que necesitarán un identificador para ser invocadas. A continuación, se da una breve explicación de cada uno de los componentes:

- *Page*: Es el componente principal que define las paginas internas dentro de un xhtml.
- *Header*: Es un componente contenedor para la parte superior de una página.
- Content: Es un componente contenedor para el área de contenido de una página.
- *Field*: Es un componente de diseño de respuesta para los pares de etiquetas de entrada.
- Footer: Es un componente de contenedor para la zona inferior de la página.

Capítulo 4. : Validación y Pruebas

En este capítulo se hará las respectivas validaciones y pruebas del sistema de inscripciones para los nuevos alumnos, en el cual existen patrones de validación como: cédula, nombre, género, estado civil, correo electrónico, número de teléfono, calificaciones, fecha de nacimiento y algunos campos más que tienen que cumplir con un formato y que son un requisito obligatorio para el registro.

4.1 Ejecutar el proyecto

En el capítulo anterior se creó el *Session Bean*, el *Managed Bean* y la página principal para iniciar sesión, ahora se procederá a probar la aplicación y revisar las respectivas validaciones, para ello lo primero se debe hacer es configurar el archivo de ejecución del proyecto. Dentro del proyecto, en la carpeta *ant/build.properties* se agregan los parámetros con los que el proyecto tendrá que ejecutarse, como se describe en la Figura 34.

```
1 #direccion del navegador web
2 browser=():\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe
3 #host donde se levanta el servidor jboss
4 jboss.host=192.168.0.101
5 #puerto donde se levanta el servidor web de jboss
6 jboss.web.port=8090
7 #directorio de instalacion de jboss (JBOSS_HOME)
8 jboss.home=():\\jboss_as-7.1.1.Final\\
9 #configuracion del jboss que se está utilizando
10 jboss.server=default
```

Figura 34 Configuración de ant/build.properties

Además, en el archivo *ant/build.xml* see registrará el nombre con el que se empaqueta el proyecto al momento de ejecutarse, las librerías que utilizará, la ubicación del archivo de persistencia, entre otros.

login.x	html 🕕 ServicioAspirante.ja 🕕 Aspirante.java 🛞 build.xml 🛛 🔳 build.properties	»
1⊖ <r< th=""><th>project name="inscripciones" basedir="/" default="deploy-show"></th><th></th></r<>	project name="inscripciones" basedir="/" default="deploy-show">	
2	······································	
3	Project settings	
4	<property name="project.distname" value="inscripciones"></property>	
5		
6	Local system paths	
7	<property file="\${basedir}/ant/build.properties"></property>	
8	<property name="webroot.dir" value="\${basedir}/WebContent"></property>	
9	<property name="webinf.dir" value="\${webroot.dir}/WEB-INF"></property>	
10	<property name="build.dir" value="build"></property>	
11	variables para crear el jar de persistencia	
12	<property name="persistence.tmp" value="persistenceTmp"></property>	
13	<property name="persistence.metaInf" value="\${persistence.tmp}/META-INF"></property>	
14	<property name="persistence.jarFile" value="persistence.jar"></property>	
15		
16	_	
17⊕	<path id="compile.classpath"></path>	
46		
47	define your folder for deployment	
48	<property name="deploy.dir" value="\${jboss.home}/standalone/deployments"></property>	
49		
50	Check timestamp on files	
510	<target name="prepare"></target>	
52	<tstamp></tstamp>	
53		
54		
55	<pre><!-- Copy any resource or configuration files--></pre>	
560	<target name="resources"></target>	
67	al Marriel build of confidential a	
60	<pre><: NORMAL DUILD OT APPLICATION> </pre>	
69®	<target depends="prepare,resources" name='compile"'>[]</target>	

Figura 35 Configuración de ant/build.xml

En la carpeta *WebContent/META-INF* agregar el archivo de persistence.xml que se creó en el proyecto de mapeo JPA, este contiene el nombre de las entidades, en que paquete están alojadas y el gestor de base de datos que se va a utilizar.

Web Development - Inscripciones/V	WebContent/META-INF/persistence.xml - Eclipse
File Edit Source Refactor Navig	jate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp
📑 🕶 🔛 🖻 🔮 🔹 🔅	▼ Ø ▼ Q₂ ▼ 🕮 Ø ▼ 💐 ▼ ♦ 🦄 ⊜ ⊗ 🕮 🧶 😂 🖋 ▼ 💷 🖄 ▼ 🖗 ▼ ⇔ ▼ ↔ ▼ 🗧
🗎 P 🛛 🔚 W 🕑 S 🖢 🗖	Be persistence.xml 🖾
Burdipciones Burdipciones Burdipciones Burdipciones Burdipciones.contro Binscripciones.contro Binscripciones.contro Binscripciones.contro Binscripciones.entida Binscripciones.utida Binscripciones.utida	<pre>1 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>
 ▷ imagenes ▷ js △ META-INF ○ MAIFEST.MF ○ paginas □ paginas □ motAsinets.aht 	<pre>22</pre>
> 😥 WEB-INF	4 III
index.jsp	Source

Figura 36 Persistence.xml

Se procede a probar la aplicación, para lo cual, se debe agregar el archivo de *ant/build.xml* en la perspectiva "*Ant*", y se la ejecuta haciendo doble click sobre la misma. Ya ejecutado el proyecto, se puede revisar la aplicación desde el navegador tanto de un computador como de un dispositivo móvil, siendo esta la vista que presenta:

•	9:42
192.168.0.103 :8090	:
Bienvenido	
Ingrese el No. Depósito:	
Aceptar	
Universidad del Azuay	



Figura 37 Vista de inicio de sesión del aplicativo

4.2 Agregar Estilos

Como se observa en la Figura 37, el diseño de la página es básico. Primefaces permite modificar este diseño por medio de hojas de estilo, que pueden hacerse manualmente o también con algunas aplicaciones web que brindan estas facilidades para crear temas personalizados para cada aplicación, en este caso se utilizará la aplicación *web ThemeRoller* que la misma página web de Primefaces sugiere, pues ofrece al usuario una interfaz intuitiva para el diseño y que permite utilizar estos estilos para cualquier aplicación web.

Para crear un diseño, se ingresará a la página web <u>https://themeroller.jquerymobile.com/</u>, esta presenta por defecto tres pantallas de dispositivos móviles que tienen las respectivas configuraciones para cada una, en donde permite combinar los colores de cada componente de la aplicación, además del estilo de la fuente, fondos de pantalla, entre otros, de manera que puede ir visualizando en ese mismo instante la apariencia que tomaría la aplicación.

← → C	eroller.jquerymobile.com			
	Version 1.4.5	r off	ort Share Help	
Global A B C +	colored by kinsav Johnny Cash	Theme 3 ×		
Swatch C Delete Duplicate	Adobe* kuler	PAGE Recent Colors	jQuery	
» Page	Newest V	Phaedra 1/4		
# Header/Footer Bar	Show alternative icons in preview See	icon demos for usage.		
+ Body				
TEXT COLOR #45ated TEXT SHADOW 0 1px 0 #131313		О в 🕕	0 c 🕕	
BACKGROUND #910818	Body	Body	Body	Add swatch
» Link	Sample text and links	Sample text and links	Sample fast and links	
= Button: Normal			Compto tox and mixed.	
TEXT COLOR #333333 TEXT SHADOW 0 1px 0 #131313	List Header	List Header	List Header	
BACKGROUND #bee5ae	Read-only list item	Read-only list item	Read-only list item	
BORDER #dddddd	Linked list item	Linked list item	Linked list item	
* Button: Hover				
TEXT COLOR #333333 TEXT SHADOW 0 1px 0 #131313	O Radio	C Radio	• Radio	
BACKGROUND	Checkbox	Checkbox	Checkbox	
BORDER #dddddd				
+ Button: Pressed	On Off	On Off	On Off	
TEXT COLOR #353553				
BACKGROUND #13e715	Option 1 📀	Option 1	Option 1 🔍	
BORDER #dddddd				
» Active State	Text Input	Text Input	Text Input	
	48	39	50	
	Button	© Button	Button	

Figura 38 Crear una hoja de estilo para aplicación móvil

Terminado el diseño para la aplicación, se descarga, descomprime y copia los estilos dentro del proyecto JSF dentro de la dirección /*WebContent/css/*. Finalmente, dentro de la página en la etiqueta <*h:head>* agregar una nueva etiqueta <*h: link>* con la dirección de las hojas de estilos que se va a utilizar. Como en este caso, se ha creado 3 estilos(a, b, c), dentro de la etiqueta <*pm:page>* se agregar la propiedad *swatch* y asignarle el nombre del diseño que se va a utilizar, es este caso el primero(a).



Figura 39 Agregar estilo a la página

Finalmente, volver a ejecutar el proyecto, en donde se podrá observar los cambios en diseño en relación a la página inicial que se presentó en la Figura 37.

192.168.0.103:8090		:
Bienvenido		
Ingrese el No. Depósito:		
001		\odot
Ace	ptar	
Universida	d del Azuay	
< <	\mathbf{D}	

Figura 40 Vista de Inicio de sesión del aplicativo con diseño

4.3 Validación y Pruebas

Lo primero que realiza el usuario, es registrar el comprobante de depósito; el sistema verifica que sea correcto y le permite ingresar sus datos. La pantalla de registro se muestra a continuación:

		Cancelar	🕗 Guardar
Datos P	Dirección	Califica	Carrera
Tipo Identifica	ción		
	Selecc	ione	Ø
Nro. Identifica	ción:*		
Primer Apellide	D:*		
Segundo Apell	ido:		
Primer Nombre	e:*		
Segundo Nomi	bre:		

Figura 41 Pantalla de registro

Todas las etiquetas que contienen un asterisco (*) son obligatorios, además, deben cumplir con un formato específico de acuerdo al tipo de información que se solicita, en caso de no cumplir con estos requisitos, el sistema mostrará un mensaje indicando que los datos proporcionados son obligatorios y/o incorrectos.

11:55		120
1	192.168	3.0.103:8090
alifice. Carrera	Formulario	de Inscripción
	Los Campo	us con * son Obligatorios de llenar
0	8	Requiere la nota del primero de
		Our days to a start of a second of the
8	8	bachillerato
correcto	8	Requiere la nota del tercero de bachillerato
	8	Requiere la nota base del primero de bachillerato
	8	Requiere la nota base del segundo de bachillerato
	8	Requiere la nota base del tercero de bachillerato
	altica. Carrers	i 192.166 alitica. Carrera Sorrecto

Figura 42 Validaciones del sistema

Si todos los datos son correctos, estos se guardarán y se enviará un comprobante de inscripción al correo electrónico que el usuario haya indicado en la aplicación.

10.10.22.17	5 :8090		:
		🕖 Salir	🕑 Guardar
Datos P	Dirección	Califica	Carrera
Tipo Identificad	ción		
	Cedu	la	O
	atos Actuali Correctamen	zados te.	
	Datos Actuali Correctamen a confirmac erá enviada Hectrónico.	izados te. ión de su R a su correo	egistro
Image: Segundo Apelli	Datos Actuali correctamen a confirmac erá enviada dectrónico.	izados te. ión de su R a su correo	egistro
Image: Segundo Apelli PAUTA	Datos Actuali Correctamen .a confirmac .erá enviada .lectrónico. do:	izados te. ión de su R a su correo	egistro
Segundo Apelli PAUTA Primer Nombre	atos Actuali correctamen .a confirmac erá enviada lectrónico. do:	zados te. ión de su R a su correo	egistro
Image: Segundo Apelli PAUTA Primer Nombre SSSS	atos Actuali correctamen a confirmac erá enviada electrónico. do:	zados te. ión de su R a su correo	egistro

Figura 43 Guardar el Registro de Inscripción

Conclusiones y Recomendaciones

En el presente trabajo se describió los conceptos básicos necesarios para la elaboración de una aplicación JSF, se hizo el levantamiento de los requerimientos para las inscripciones de los nuevos alumnos, se realizó los diagramas de clases, diagramas de uso, modelo entidad-relación, se indicó como mapear las entidades, creación de *Session Bean* y *Managed Bean*, configuración e implementación de componentes Primefaces, Primefaces Mobile, y se desarrolló la aplicación prototipo para la inscripción de nuevos alumnos en la Universidad del Azuay.

Como puntos débiles de Primefaces, se puede indicar que al ser un *framework* que está en desarrollo, sometido a constantes actualizaciones de versiones, es susceptible a problemas como la migración entre versiones, pues hasta el momento no ha existido una actualización que sea 100% compatible con versiones anteriores. Por lo tanto, los aplicativos que se desarrollen utilizando Primefaces deben estar en permanentes actualizaciones, si se desea aprovechar de cada una de las mejoras que va presentando.

La visión central de Primefaces Mobile, es proporcionar al usuario una aplicación que permita interactuar desde un terminal móvil sin que esta pierda su funcionalidad, y que permita ser utilizada por dispositivos móviles independientemente de la plataforma con la que estos trabajen.

Conocer las herramientas con las que se puede desarrollar una aplicación JSF utilizando el framework Primefaces, permitió que se desarrolle con éxito la aplicación prototipo para la inscripción de nuevos alumnos de la Universidad del Azuay, en donde configuró y demostró la utilización de este framework.

Este prototipo tuvo como objetivo ingresar la información de los nuevos alumnos de la universidad, previo al ingreso de la información por parte del nuevo alumno, es necesario que haya pagado en una de las instituciones financieras el derecho de inscripción, pues este comprobante es un dato obligatorio al momento de ingresar al sistema. Una vez validado el comprobante de depósito, el nuevo alumno puede ingresar la información solicitada y podrá modificarla las veces que considere necesarias, siempre y cuando se encuentre dentro de las fechas establecidas para la inscripción. En el proceso de validación del comprobante de depósito se pudo notar que es muy sencillo, por lo tanto podría ser vulnerable a cualquier ataque de personas o software malintencionado, se recomienda aumentar la seguridad, utilizando métodos de validación como: *captcha, google authenticator*, entre otros.

Con la implementación de este *framework* se pudo notar que cada componente que este presenta, consta con un sinnúmero de propiedades, eventos, Ajax, que facilitan el desarrollo de aplicaciones utilizando lo que ya está programado y permitiéndonos centrarnos en detalles más específicos.

Finalmente, uno de las restricciones es que al ser un *framework* nuevo y que se encuentra en constante actualización de versiones, no existe cuantiosa información para los desarrolladores como lo hay de otros *frameworks*. Para cualquier consulta sobre el manejo de los componentes de Primefaces, se recomienda recurrir en primera instancia a la documentación que Primefaces ofrece en su sitio oficial, pues con cada actualización de versión que realizan, presentan también su respectiva documentación, el cual detalla cada componente y sus respectivos atributos y propiedades.

Glosario

Eclipse: Es una comunidad de personas y organizaciones quienes colaborar en el negocio de la comercialización de software de código abierto. Fue creado por IBM en noviembre de 2001 con el apoyo de un consorcio de proveedores de software. La fundación Eclipse es una organización sin fines de lucro que fue creado en enero de 2004 para que actúe como administrador dela comunidad de eclipse. En general, la fundación Eclipse ofrece cuatro servicios a la comunidad Eclipse: 1) Infraestructura de TI, 2) Gestión IP, 3) Proceso de Desarrollo, 4) Desarrollo del Ecosistema. (Foundation)

Bean: Un Bean es una simple clase que tiene un constructor sin argumentos y sus atributos tienen sus correspondientes métodos get/set. Si en una clase no se indica ningún constructor, el compilador crea este constructor por defecto (Cortés, 2011).

Servlet: Los Servlets son módulos en Java utilizados en un servidor, que puede o no ser servidor web, que sirven para extender sus capacidades de respuesta a los clientes al utilizar las potencialidades de Java. Los Servlets son para los servidores lo que los applets para los navegadores, pero no tienen una interfaz gráfica. Pueden ser incluidos en servidores que soporten la API de Servlet. (Barrios, 2001)

JPA: Java Persistence Api, proporciona un modelo de persistencia para el mapeo objeto-relacional. Fue desarrollado por el grupo de expertos de software 3.0 EJB como parte de JSR 220, pero su uso no se limita a los componentes de software EJB. También puede ser utilizada directamente por las aplicaciones web y clientes de aplicaciones e incluso fuera de la plataforma Java EE. (Oracle, Java Persistence API)

EJB: Enterprise Java Beans, es un componente de negocio Java Enterprise, que para su ejecución necesita de un contenedor EJB/J2EE. El hecho de utilizar EJB's permite el acceso a los servicios del contenedor EJB, es decir, manejo de transacciones, seguridad, persistencia, entre otros (Pesquera, 2014).

Bibliografía

- Alférez, J. A. (25 de Octubre de 2010). *Instalación, Configuración y Administración del servidor de aplicaciones JBoss.* Recuperado el 15 de Agosto de 2015
- Barrios, J. M. (30 de 11 de 2001). *Java Servlets*. Recuperado el 28 de 09 de 2015, de http://users.dcc.uchile.cl/~jbarrios/servlets/general.html
- Ceria, S. (s.f.). *Casos de Uso, Un método práctico para explorar requerimientos*. Recuperado el 01 de 10 de 2015, de http://www-2.dc.uba.ar/materias/isoft1/2001_2/apuntes/CasosDeUso.pdf
- Çivici, Ç. (2015). Primefaces User Guide 5.2.
- Cortés, A. (21 de 10 de 2011). *El Bog de Ana Cortés*. Recuperado el 28 de 09 de 2015, de https://anadreamy.wordpress.com/2011/10/21/jsf-2-0-managed-beans/
- DATE, C. (2001). Introducción a los sistemas de bases de datos. México: Pearson Educación.
- Foundation, E. (s.f.). *Eclipse*. (About the Eclipse Foundation) Recuperado el 1 de 10 de 2015, de https://eclipse.org/org/
- Inc., O. C. (s.f.). *Oracle corporation us*. Recuperado el 01 de 12 de 2014, de Oracle corporation us: https://www.java.com/es/
- Kurniawan, B. (2014). Java 7: A Comprehensive Tutorial. Secaucus: Brainy Software Team.
- Oracle. (Octubre de 2014). PrimeFaces in the Enterprise, Part 2: Mobile Solutions for the Enterprise. Recuperado el 17 de Febrero de 2015, de http://www.oracle.com/technetwork/articles/java/java-primefaces-pt2-2340750.html
- Oracle. (s.f.). Java Persistence API. Recuperado el 28 de 09 de 2015, de http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html
- Pesquera, C. (19 de 3 de 2014). *Que es un POJO, EJB y un Bean?* Recuperado el 28 de 09 de 2015, de http://carlospesquera.com/que-es-un-pojo-ejb-y-un-bean/
- Primefaces. (s.f.). *Primefaces Mobile*. Recuperado el 02 de Marzo de 2015, de http://www.primefaces.org/showcase/mobile/index.xhtml
- Rafaelma. (02 de Octubre de 2010). *Sobre PostgreSQL*. Recuperado el 28 de Septiembre de 2015, de sitio web de PostgreSQL-es: http://www.postgresql.org.es/sobre_postgresql

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 DIGNA ISABEL QUEZADA PAUTA con código 47721, que denuncia su trabajo de titulación denominado: "DESARROLLO DE UN PROTOTIPO PARA DISPOSITIVOS MOVILES PARA LA INSCRIPCION DE NUEVOS ALUMNOS EMPLEADO LA TECNOLOGIA JAVA SERVER FACES CON IMPLEMENTACION DE PRIMEFACES MOBILE.", 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 Graduación 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

CNACED STAR LIG I FACIALIAD DE ADASIM IRACION SECLETARIA

() ()

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 para dispositivos móviles para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con implementación de PrimeFaces Mobile"", presentado por el estudiante Digna Isabel Quezada Pauta con código47721, previa a la obtención del grado de Ingeniero de Sistemas y Telemática, para el día <u>MIERCOLES 25 DE MARZO 2015 A LAS 08H30</u>.

Cuenca, 19 de marzo de 2015

Dra. Jenny Rios Coello Secretaria de la Facultad

Ing. Oswaldo Merchán Manzano

Ing. Marcos Orellana Cordero

() ()

Guía para Trabajos de Titulación 💙



1. Protocolo/Acta de sustentación



SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante: Digna Isabel Quezada Pauta Código 47721
- 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 para dispositivos móviles para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con implementación de PrimeFaces Mobile"
- 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

ROGE Sr. Digna Isabel Quezada Pauta

Ing. Oswaldo Merchán Manzano

Dra. Jenny tios Edello Secretario de Facultad

Fecha de sustentación: Miércoles 25 de marzo de 2015

Guía para Trabajos de Titulación



UNIVERSIDAD DEL AZUAY

RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

- 1.1 Nombre del estudiante: Digna Isabel Quezada Pauta
 - 1.1.1 Código 47721
- 1.2 Director sugerido: Ing. Marcos Orellana Cordero
- 1.3 Codirector (opcional):.

 \bigcirc

- **1.4 Título propuesto:** "Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con implementación de PrimeFaces Mobile"
- 1.5 Revisores (tribunal): Ing. Oswaldo Merchán Manzano
- 1.6 Recomendaciones generales de la revisión:

	Cumple	Cumple	No	Observaciones
	totalmente	parcialmente	cumple	(*)
Linea de investigación				
1. ¿El contenido se enmarca en la línea	1		1	·····
de investigación seleccionada?				
Título Propuesto	1/			
2. ¿Es informativo?	1,			
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?	1			
11.¿Es factible de verificación?				
Objetivo general				
12.¿Concuerda con el problema				
formulado?	1/			
13.¿Se encuentra redactado en tiempo				
verbal infinitivo?				

1

Guía para Trabajos de Titulación

1. Protocolo/Rúbrica

(*) Breve justificación, explicación o recomendación.

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

.....

.....

Ing. Marcos Orellana Cordero

Ing. Oswaldo Merchán Manzano



Facultad de Ciencias de la Administración Escuela de Ingeniería de Sistemas y Telemática

Oficio Nro. 032-2015-DIST-UDA

Cuenca, 05 de Marzo de 2015

Señor Ingeniero Xavier Ortega Vázquez DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMNISTRACIÓN 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 para dispositivos móviles para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con implementación de PrimeFaces Mobile", presentada por la estudiante Digna Quezada, 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: 09-03-2015

ESCUELA DE INGENIERIA DE SISTEMAS

المادة فالمحكمة والمحافظ والمحتوج والمحار المادات

Diseños de Monografía Escuela de Ingeniería de Sistemas

Estudiante: Digna Isabel Quezada Pauta con código 47721. Tema: "DESARROLLO DE UN PROTOTIPO PARA DISPOSITIVÓS MOVILES PARA LA INSCRIPCION DE NUEVOS ALUMNOS EMPLEANDO LA TECNOLOGIA JAVA SERVER FACES CON IMPLEMENTACION **DE PRIMEFACES MOBILE**" Para: La obtención del título de Ingeniera en Sistemas y Telemática Director: Ing. Marcos Orellana Cordero.

Tribunal: Ing. Oswaldo Merchán Manzano.

DIA: Mireeoles FECHA: 25 De Raezo hois HORA: 08/30



د النودية بقال هلوهو : وتانيا فإيها و شده .

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

CERTIFICA:

Que, la Señorita Digna Isabel Quezada Pauta registrada con código 47721, 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 TRADUSIDAD.DE 270 110 BE 加加工作工作的

್ಷ ಎಲ್. ಇಂಗ್ಲಿಂಗ್ ಗ್ರೈನ್ ಗ್ರೈನ್ ಕ್ರಿ ಇನ್ನ ಸ್ಮೇಖ್ಯ ಸಂಸ್ಥೆ ಕಾರ್ಯನಿಗಳು ಇನ್ನ ಸ್ಮೇಖ್ಯ ಸಂಸ್ಥೆ ಕಾರ್ಯನಿಗಳು

Derecho 71633 vcf.-

.....

.....

.....

Edición autorizada de 20.000 ejemplares Nº. Del 658.501 al 678.500 0668344

.....



.....

Cuenca, 05 de marzo de 2015

Ing. Xavier Ortega Vasquez

Decano de la Facultad de Ciencias de la Administración

Ciudad

De mis consideraciones:

YO, DIGNA ISABEL QUEZADA PAUTA con código 47721, 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 para dispositivos móviles para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces Mobile", 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

ES sucha

Digna Isabel Quezada Pauta

Ø105150445

autorizada de 30.000 ejemplares Nº

0688800

Cuenca, 05 de Marzo del 2015

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

 $\hat{N} \in \int_{-\infty}^{\infty} \frac{d}{dt} \, \hat{N}_{ijk}(t) \, dt \, t$

De mis consideraciones:

Luego de revisar el diseño del trabajo de graduación denominado: "Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alumnos empleando la tecnología Java Server Faces con componentes PrimeFaces Mobile"; presentado por la estudiante Digna Quezada, 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	
		,
**********	CONSERVATION AND A CONSERVATIONA	

	Universidad del Azuay	
	Facultad de Ciencias de la Administración	
******	Escuela de Ingeniería de Sistemas y Telemática	<u></u>
		······
*******		••••••
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mol	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mol Autor: Digna Quezada	mnos pile
······	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana	mnos pile
······	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana	mnos pile
······	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mol Autor: Digna Quezada Director: Ing. Marcos Orellana	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mol Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mol Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mot Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile
	Desarrollo de un prototipo para dispositivos móviles para la inscripción de nuevos alu empleando la tecnología Java Server Faces con implementación de PrimeFaces Mol Autor: Digna Quezada Director: Ing. Marcos Orellana Cuenca, Ecuador Marzo, 2015	mnos pile

 $\langle 0 \rangle$ $\langle \rangle$ $\langle \rangle$ $\langle \rangle$ $\langle\!\langle \rangle\!\rangle$ $\langle \cdot \rangle$

PROTOCOLO DE TRABAJO DE TITUL



1. DATOS GENERALES

1.1.	Nombre del estudiante: Quezada Pauta Digna Isabel
•••••••••	1.1.1.Código: 47721
	1.1.2.Contacto: (07)2275475, 0958727637, dignaquezada@outlook.com
1:2:	Director sugerido: Marcos Orellana Cordero
·····	1.2.1.Contacto: 0999955611, marore@uazuay.edu.ec
1.3.	Tribunal designado:
1.4.	Aprobación:
	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 para dispositivos móviles para la
	inscripción de nuevos alumnos empleando la tecnología Java Server Faces con
	implementación de PrimeFaces Mobile
1.8.	Subtítulo: Como implementar un proyecto para dispositivos móviles con PrimeFaces
	Mobile
····1:9:	Estado del proyecto: el trabajo es nuevo

2. CONTENIDO

2.1. Motivación de la investigación

Actualmente el tráfico de internet móvil está cada vez en aumento, y la creación de una navegación móvil sin problemas es uno de las mayores preocupaciones de cada Web Mater, pues independientemente de la conexión que los usuarios tengan, esperan una experiencia óptima en sus dispositivos móviles.

PrimeFaces Mobile ayuda en esta complicada tarea y permite crear rápidamente aplicaciones web para estos dispositivos móviles que se visualice bien en teléfonos independientemente de la plataforma que estos soporten.

2.2 Problemática

Mientras la venta de smartphones y tablets se incrementa cada día más, aumenta también la necesidad de desarrollar aplicaciones web que soporten el acceso desde estos dispositivos móviles. Existen lenguajes de programación que nos permiten desarrollar aplicaciones móviles, pero cada una de ellas enfocada a una plataforma específica como iPhone, Android, Blackberry, Windows Mobile y muchos otros más. PrimeFaces Mobile permite construir una interfaz de usuario que proporciona soporte táctil optimizado diferentes plataformas. Pero el inconveniente prima puesto que es una herramienta nueva y se conoce poco sobre su implementación y las múltiples ventajas que esta presta.

Edición autorizada de 30.000 ejemplares Nº

Del 678.501 al 708.500

0688762

2.3 Pregunta de Investigación

Como implementar una interfaz de usuario para crear una aplicación basada en JSF conel diseño de respuesta para dispositivos móviles utilizando PrimeFaces Mobile?

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 utilizamos en nueva, es por ello que con el presente trabajo se pretende explicar de forma sencilla la implementación de un proyecto web que soporte vistas de dispositivos móviles, mediante el ejemplo aplicado a la inscripción para los alumnos nuevos de la Universidad del Azuay, por medio de esto detallaremos la forma de llevar a cabo la implementación utilizando la tecnología JSF con componentes PrimeFaces Mobile.

Al finalizar obtendremos nuestro proyecto culminado y una clara explicación sobre la funcionalidad de PrimeFaces Mobile 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 nos centraremos en la parte orientada a desarrollo web (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, "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 -bases 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) AZUAY

Esto lo convierte en una plataforma flexible y 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 iniciamos un proyecto real, debemos necesariamente elegir una implementación, ya que tenemos 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 Mobile

El marco PrimeFaces móvil JSF proporciona soluciones sofisticadas para la mayoria de los requisitos de las aplicaciones móviles. Construido sobre jQuery Mobile, que ofrece una gran cantidad de componentes móviles listos que hacen un uso racional de la pantalla más pequeña, y el marco proporciona una serie de eventos de toque que pueden ser utilizados para optimizar la experiencia móvil. (Oracle, 2014)

PrimeFaces móvil (PFM) es un kit de interfaz de usuario para crear aplicaciones JSF optimizados para dispositivos móviles. PFM se construye en la parte superior de jQuery Mobile, un marco HTML5 interfaz de usuario que proporciona soporte táctil optimizada para diferentes plataformas.

Además de la integración perfecta con jQuery Mobile. PFM cuenta con un RenderKit móvil para componentes populares PrimeFaces, extensiones marco ajax, eventos de comportamiento ajax móviles, modelo integrado de navegación, carga lenta de las páginas, los widgets de respuesta y más. (Primefaces)

2.6 Objetivo General

Desarrollar un prototipo para la inscripción de nuevos alumnos en la Universidad del Azuay para dispositivos móviles a través del framework Primefaces Mobile.

2.7 Objetivos Específicos

 Analizar las características y opciones que ofrece el frameework PrimeFaces Mobile

 Desarrollar una aplicación prototipo para la inscripción de los nuevos alumnos a través de dispositivos móviles.

torizada de 30.000 ejemplares Nº

Del 678 501 al 708 500

0688763

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 investigara la tecnología JSF y la forma más sencilla para implementar un proyecto móvil con componentes PrimeFaces Mobile a través de revisión bibliográfica.

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

Finalmente se desarrollará un prototipo móvil 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 son soporte para dispositivos móviles, empleando la tecnología JSF con los componentes PrimeFaces Mobile 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 con profesionales que conozcan de 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 alterno para el desarrollo del aplicativo, instalado previamente 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
	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

- Carlos



2 Financiamiento UNIVERSIDAD DEL AZUAY 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.

	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 Mobile	
	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 Mobile	
	8. Capítulo 4: Validación y pruebas	
******	1. Pruebas del aplicativo	
	9. Conclusiones y recomendaciones	
	10. Apéndices, tablas, gráficos	
	11. Referencias	
,		

2.14 Cronograma

Buscar información sobre la tecnología JSF con componentes PrimeFaces Mobile Capacitación necesaria para implementar un por otorga el framework PrimeFaces Mobile: • Creación del proyecto ISF framework PrimeFaces proyecto web utilizando la seri tecnología JSF junto a los componentes PrimeFaces Mobile: • Creación del proyecto ISF persistencia proyecto web utilizando la componentes PrimeFaces Detallar el proceso a seguir para la insertipción de los muevos alumnos, las pantallas, validación de datos, etc prototipo web para la insertipción de alumnos la base de datos requerida para la aplicación nuevos a través de una interface web. Prototipo web para la insertipción de nuevos alumnos Trabajar sobre una arquitectura de 3 capas da base, de datos, capa de acceso a la base de datos, en et cual se distinguirá claramente lo que abarca cada una de las capas. prototipo web para la inscripción de nuevos	Objetivo Específico	Actividad	Resultado esperado	Tiempo (semana
con componentes PrimeFaces Mobile Analizar las earacterísticas y opciones que otorga el franework PrimeFaces Mobile: Desarrolla Detallar el proceso a seguir para la inscripción de los Manage Bean Detallar el proceso a seguir para la inscripción de los nuevos alumnos. las pantallas, validación de datos, etc. Desarrollar una para implementar Mobile Creación de los Manage Bean Detallar el proceso a seguir para la inscripción de los nuevos alumnos. las pantallas, validación de datos, etc. Desarrollar una pricación para la interface web. Trabajar sobre una arquitectura de 3 capas (capa de acceso a la base de datos), en el cual se distinguirá claramente lo que abarca cada una de las capas.		Buscar información sobre la tecnología JSF		
Analizar las Configuración del ambiente de características y Creación del proyecto JSF para implementar un proyecto veb utilizando la 3 ser framework PrimeFaces Configuración del archivo de Mobile. Configuración del archivo de persistencia componentes PrimeFaces Mobile. Mapeo de entidades omponentes PrimeFaces Detallar el proceso a seguir para la inscripción de los nuevos alumnos, las pantallas, validación de datos, etc. pantallas, validación de datos, etc. Desarrollar una aplicación para la inscripción de lasse de datos requerida para la aplicación de alumnos y relaciones que formarán parte de inscripción de alumnos retadas sobre una arquitectura de 3 capas Trabajar sobre una arquitectura de 3 capas alumnos alumnos capa de acceso a la base de datos), en el cual se distinguirá claramente lo que abarca cada una de las capas.		con componentes PrimeFaces Mobile		
características y características para implementar un opciones que otorga el • Creación del proyecto JSF provecto web utilizando la 3 ser framework PrimeFaces • Configuración del archivo de componentes PrimeFaces Mobile: • Mapeo de entidades • Creación de los Manage Bean Mobile - - Detallar el proceso a seguir para la inscripción de los nuevos alumnos, las - - aplícación para la inscripción de los requerida para la aplicación - - - nuevos a través de una inase de datos requerida para la aplicación - - - - interface web. Trabajar sobre una arquitectura de 3 capas alumnos - - - - una de las capas. una de las capas. -	Analizar las	Configuración del ambiente de desarrollo	Capacitación necesaria	
Trainework PrimeFaces Configuración del archivo de persistencia Mapeo de entidades Mapeo de entidades Creación de los Manage Bean Mageo de entidades Creación de los Manage Bean Detallar el proceso a seguir para la inscripción de los nuevos alumnos, las pantallas, validación de datos, etc. Desarrollar una Especificar las diferentes entidades, atributos y relaciones que formarán parte de lase de datos requerida para la aplicación de nuevos alumnos Iabase de datos requerida para la aplicación de nuevos alumnos (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. 	características y opciones que otorga el	Creación del proyecto JSF	para implementar un proyecto web utilizando la	3 seman
Mapeo de entidades	framework PrimeFaces Mobile:	 Configuración del archivo de persistencia 	tecnologia JSF junto a los componentes PrimeFaces Mobile	
Creación de los Manage Bean Detallar el proceso a seguir para la inscripción de los nuevos alumnos, las pantallas, validación de datos, etc. Desarrollar una Especificar las diferentes entidades, atributos y relaciones que formarán parte de la base de datos requerida para la aplicación inscripción de nuevos 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.		• Mapeo de entidades	Woone	
Detallar el proceso a seguir para la inscripción de los nuevos alumnos, las pantallas, validación de datos, etc. Desarrollar una aplicación para la aplicación de alumnos nuevos a través de una interface web. Especificar las diferentes entidades, atributos y relaciones que formarán parte de la base de datos requerida para la aplicación nuevos a través de una Prototipo web para la nuevos Trabajar sobre una arquitectura de 3 capas alumnos (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. alumnos		Creación de los Manage Bean		
inscripción de los nuevos alumnos, las pantallas, validación de datos, etc. Desarrollar una aplicación para la antibutos y relaciones que formarán parte de inscripción de alumnos nuevos a través de una interface web. 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.		Detallar el proceso a seguir para la		
Desarrollar una aplicación para la inscripción de alumnos nuevos a través de una interface web. Especificar las diferentes entidades, atributos y relaciones que formarán parte de la base de datos requerida para la aplicación nuevos a través de una interface web. Prototipo web para la inscripción de nuevos alumnos 9 ser 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 alumnos		pantallas, validación de datos, etc		
aplicación para la atributos y relaciones que formarán parte de inscripción de alumnos la base de datos requerida para la aplicación de nuevos a través de una interface web. 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.	Desarrollar una	Especificar las diferentes entidades.		÷
interface web. 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.	aplicación para la inscripción de alumnos	atributos y relaciones que formarán parte de la base de datos requerida para la aplicación	Prototipo web para la inscripción de nuevos	9 seman
(capa de presentación, capa de negociós y capa de acceso a la base de datos), en el cual se distinguirá claramente lo que abarca cada una de las capas.	interface web.	Trabajar sobre una arquitectura de 3 capas	alumnos	
capa de acceso a la base de datos), en el cual se distinguirá claramente lo que abarca cada una de las capas.		(capa de presentación, capa de negocios y		
se distinguirá claramente lo que abarca cada una de las capas.		capa de acceso a la base de datos), en el cual		
		se distinguirá claramente lo que abarca cada		
		una de las capas.		va na servit mente
	nuevos a través de una interface web.	Trabajar sobre una arquitectura d (capa de presentación, capa de n capa de acceso a la base de datos), se distinguirá claramente lo que al una de las capas.	e 3 capas egocios y en el cual parca cada	alumnos egocios y en el cual parca cada
			e en de la constante de la const	
	- - - -			

 $\langle \rangle$ \langle



 $\left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right)$

 $\langle \rangle$ (

	2.15	Referencias
	(s.f.). O	btenido de
		http://www.sicuma.uma.es/export/sites/sicuma/es/formacion/descargas/JSF.pdf
	Fain, Y.	(2011). Java Programming 24-Hour Trainer. John Wiley & Sons.
	oracle. (s.f.). Oracle corpotation us. Recuperado el 30 de 01 de 2015, de Oracle corporation us: https://www.java.com/es/
	Parkway	7, 5. O. (01 de 2013). <i>The Java EE 6 Tutorial.</i> 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 Mobile. Recuperado el 01 de 02 de 2015, de PrimeFaces showcase: http://www.primefaces.org/showcase/mobile/index.xhtml
	Take, P.	(2015). PrimeFaces Mobile: Recuperado el 01 de 02 de 2015, de PrimeFaces showcase: http://www.primefaces.org/showcase/mobile/index.xhtml
	2 16	Firmes de Desponschilided (estudioute)
		And Esolo
	2.17	Firma de Responsabilidad (director)
	2:18 05-M	Fecha de Entrega
	•••••••••••••••••••••••••••••••••••••••	
· · · · · · · · · · · · · · · · · · ·		

0688767

autorizada de 30.000 ejemplares **No**s 501 al 708:500