



## **DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**TRABAJO DE TITULACIÓN PREVIO LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: DESARROLLO DEL SISTEMA AQUINO APLICANDO  
MINERÍA DE DATOS PARA EL TEST VOCACIONAL  
DIRIGIDO A LOS POSTULANTES A BECAS ACADÉMICAS  
DEL SENESCYT**

**AUTOR: ANDRADE TANDAZO, RAMIRO ERNESTO**

**DIRECTOR: ING. CAMPAÑA, MAURICIO**

**SANGOLQUÍ, NOVIEMBRE 2015**



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, "**DESARROLLO DEL SISTEMA AQUINO APlicando MINERÍA DE DATOS PARA EL TEST VOCACIONAL DIRIGIDO A LOS POSTULANTES A BECAS ACADÉMICAS DEL SENESCYT**" realizado por el señor **ANDRADE TANDAZO, RAMIRO ERNESTO**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **ANDRADE TANDAZO, RAMIRO ERNESTO** para que lo sustente públicamente.

Sangolquí, 15 de noviembre del 2015

ING. MAURICIO EDUARDO CAMPAÑA ORTEGA

DIRECTOR



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**AUTORÍA DE RESPONSABILIDAD**

Yo, **ANDRADE TANDAZO, RAMIRO ERNESTO**, con cédula de identidad N° 1718515081, declaro que este trabajo de titulación "**DESARROLLO DEL SISTEMA AQUINO APLICANDO MINERÍA DE DATOS PARA EL TEST VOCACIONAL DIRIGIDO A LOS POSTULANTES A BECAS ACADÉMICAS DEL SENESCYT**" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas. Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

**Sangolquí, 15 de noviembre del 2015**

**RAMIRO ERNESTO ANDRADE TANDAZO  
C.C 1718515081**



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**AUTORIZACIÓN**

Yo, **ANDRADE TANDAZO, RAMIRO ERNESTO**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "**DESARROLLO DEL SISTEMA AQUINO APlicando MINERÍA DE DATOS PARA EL TEST VOCACIONAL DIRIGIDO A LOS POSTULANTES A BECAS ACADÉMICAS DEL SENESCYT**" cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

**Sanglquí, 15 de noviembre del 2015**

RAMIRO ERNESTO ANDRADE TANDAZO  
C.C 1718515081

## **DEDICATORIA**

Dedico este trabajo a toda mi familia por su incondicional apoyo durante todo mi proceso formativo, pero en especial a mi abuela Rosa Cando quien ha sido para mí más que una madre y que siempre ha estado presente en toda mi vida brindándome su amor y sus consejos. De igual forma se lo dedico a mi difunto abuelo Darío Tandazo, que al igual que mi abuela fue un padre para mí, quien con sus enseñanzas y conocimiento supo guiarme durante sus años de vida por el camino del bien, brindándome su conocimiento y su amor.

A mis padres que, aunque no han estado presentes en la mayor parte de mi vida, han sabido demostrar cuán importante soy para ellos y también por haberme dado la vida.

## **AGRADECIMIENTO**

Agradezco principalmente a mi novia Kassandra García por estar siempre presente apoyándome y brindándome todo su amor y cariño para lograr culminar con esta meta, la cual solo es la primera entre varias. De igual forma a todos los profesores y amigos que de una u otra forma han estado presentes durante toda mi formación académica y pero en especial a los que han colaborado en el proceso de desarrollo de esta tesis.

## AGRADECIMIENTO ESPECIAL

*“La disciplina es el puente que une las metas con los logros”*

Agradecemos al PhD. Filipe Mota Prometeo del Departamento de Ciencias de la Computación de la Universidad de las Fuerzas Armadas – ESPE y Asesor Técnico de la SENESCYT por la confianza brindada a nosotros y a su vez por la colaboración siendo el vínculo con La Secretaría de Educación Superior, Ciencia, Tecnología e Innovación quien nos otorgó el auspicio para el desarrollo de este proyecto de investigación.

Al Msc. Mauricio Campaña por aceptar este reto siendo nuestro Director de Proyecto y habernos guiado durante el proceso del desarrollo escrito del mismo; además de acompañarnos durante nuestro proceso formativo en la Carrera de Ingeniería de Sistemas e Informática

Al Ing. Henry Coral por su constante apoyo y guía para el desarrollo práctico del proyecto, compartiendo su experiencia y conocimiento no solo profesionales sino también lecciones de vida y forjando una gran amistad.

Ramiro Andrade

Natalia Cerón

Mireya Chillán

# ÍNDICE

<b>DEDICATORIA.....</b>	<b>v</b>
<b>AGRADECIMIENTO .....</b>	<b>vi</b>
<b>AGRADECIMIENTO ESPECIAL .....</b>	<b>vii</b>
<b>RESUMEN.....</b>	<b>xiv</b>
<b>ABSTRACT .....</b>	<b>xv</b>
<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1    ANTECEDENTES .....	1
1.2    PROBLEMÁTICA.....	1
1.3    JUSTIFICACIÓN.....	2
1.4    OBJETIVOS.....	2
1.4.1    OBJETIVO GENERAL .....	2
1.4.2    OBJETIVOS ESPECÍFICOS.....	2
1.5    ALCANCE .....	3
<b>CAPÍTULO 2.....</b>	<b>5</b>
<b>MARCO TEÓRICO .....</b>	<b>5</b>
2    APLICACIONES WEB .....	5
2.1.1    CARACTERÍSTICAS DE LAS APLICACIONES WEB.....	5
2.1.2    TIPOS DE APLICACIONES WEB .....	7
2.1.3    MODELADO DE APLICACIONES WEB.....	9
2.2    METODOLOGÍA UWE (UML-BASED WEB ENGINEERING) .....	11
2.2.1    INTEGRACION DE UWE .....	12
2.2.2    FASES PARA LA IMPLEMENTACIÓN DE UWE .....	13
2.3    JAVASERVER FACES (JSF) .....	14

2.3.1	VERSIÓN ACTUAL .....	14
2.3.2	IMPLEMENTACIONES .....	15
2.3.3	CARACTERÍSTICAS / FUNCIONALIDADES .....	16
2.3.4	BACKING BEANS .....	16
2.3.5	SCOPE .....	17
2.3.6	LENGUAJE EL .....	18
2.3.7	CICLO DE VIDA .....	19
2.3.8	FACELETS .....	21
2.3.9	PLANTILLAS .....	22
2.3.10	COMPONENTES PRINCIPALES.....	23
2.3.11	VALIDADORES .....	24
2.3.12	CONVERSORES.....	25
2.4	MODELO VISTA CONTROLADOR (MVC) .....	25
2.4.1	CAPAS .....	25
2.5	BASES DE DATOS .....	27
2.5.1	MODELAMIENTO DE BASES DE DATOS.....	28
2.5.2	NORMALIZACIÓN .....	29
2.5.3	MINERÍA DE DATOS .....	30
2.6	TEST VOCACIONAL .....	32
<b>CAPÍTULO 3.....</b>		<b>33</b>
<b>ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....</b>		<b>33</b>
3.1	INTRODUCCIÓN.....	33
3.2	PROPÓSITO .....	33
3.3	ALCANCE .....	33
3.4	DESCRIPCIÓN .....	34
3.4.1	REQUERIMIENTOS FUNCIONALES.....	34

3.4.2	RESTRICCIONES GENERALES .....	38
3.4.3	INTERFACES DE SOFTWARE.....	39
3.4.4	INTERFACES DE COMUNICACIÓN.....	39
3.5	METODOLOGÍA UWE .....	39
3.5.1	DEFINICIÓN DE ACTORES .....	39
3.5.2	DEFINICIÓN DE LOS CASOS DE USO.....	40
3.5.3	MODELO LÓGICO .....	41
3.5.4	MODELO FÍSICO .....	42
3.5.5	SCRIPT DE LA BASE DE DATOS.....	45
3.5.6	MODELO DE PRESENTACIÓN .....	45
3.5.7	DIAGRAMA NAVEGACIONAL.....	46
3.5.8	DIAGRAMA DE CLASES .....	47
3.5.9	DIAGRAMA DE SECUENCIA.....	49
3.6	ARQUITECTURA.....	53
3.6.1	EAR.....	53
3.6.2	EJB .....	53
3.6.3	WEB.....	54
3.7	PRUEBAS .....	54
3.7.1	ESTÁNDARES DE IMPLEMENTACIÓN .....	54
3.7.2	PRUEBAS DE CAJA NEGRA.....	56
3.7.3	PRUEBAS DE ESTRÉS .....	62
<b>CAPÍTULO 4.....</b>		<b>65</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>65</b>
4.1	CONCLUSIONES.....	65
4.2	RECOMENDACIONES .....	65
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>1</b>

# ÍNDICE DE TABLAS

Tabla 1 Categorización de las aplicaciones web .....	7
Tabla 2 Versiones de JSF .....	15
Tabla 3 Implementaciones de JSF .....	16
Tabla 4 Tipos de Scope .....	17
Tabla 5 Ciclo de vida JSF .....	20
Tabla 6 Etiquetas UI .....	22
Tabla 7 Etiquetas Html .....	23
Tabla 8 Etapas de la minería de datos .....	31
Tabla 9 Requerimiento funcional 1 Registrar usuarios .....	34
Tabla 10 Requerimiento funcional 2 Iniciar sesión .....	34
Tabla 11 Requerimiento funcional 3 Cambiar clave .....	35
Tabla 12 Requerimiento funcional 4 Registrar titulo .....	36
Tabla 13 Requerimiento funcional 5 Desarrollar el test de intereses vocacionales CIPSA .....	36
Tabla 14 Requerimiento funcional 6 Buscar programas de maestrías .....	37
Tabla 15 Requerimiento funcional 7 Agregar comentarios .....	37
Tabla 16 Requerimiento funcional 8 Seleccionar .....	38
Tabla 17 CU01 Registrar usuarios .....	41
Tabla 18 Descripción de las entidades de la base de datos .....	41
Tabla 19 Prueba de Caja Negra 1 Registrar usuario .....	57
Tabla 20 Prueba de Caja Negra 2 Iniciar sesión .....	57
Tabla 21 Prueba de Caja Negra 3 Cambiar clave .....	58
Tabla 22 Prueba de Caja Negra 4 Registrar Título .....	59
Tabla 23 Prueba de Caja Negra 5 Desarrollar el test de intereses vocacionales .....	59
Tabla 24 Prueba de Caja Negra 6 Buscar programas de maestrías .....	60
Tabla 25 Prueba de Caja Negra 7 Agregar Comentarios .....	61
Tabla 26 Prueba de Caja Negra 8 Seleccionar .....	61

# ÍNDICE DE FIGURAS

Figura 1. Estructura General del Sistema AQUINO.....	4
Figura 2. Descripción general de los modelos (UWE, 2015) .....	12
Figura 3 Ciclo de vida JSF .....	19
Figura 4 MVC Estructura de comunicación entre capas (Wikipedia, 2015) .....	26
Figura 5 Ciclo de vida MVC (Israel, 2013) .....	27
Figura 6 Diagrama global de casos de uso del sistema AQUINO .....	40
Figura 7 Modelo lógico de la base de datos.....	43
Figura 8 Diagrama Físico de la base de datos.....	44
Figura 9 Modelo de presentación.....	46
Figura 10 Diagrama navegacional, usuario no registrado.....	46
Figura 11 Diagrama navegacional, usuario registrado.....	47
Figura 12 Diagrama de clases del sistema AQUINO.....	48
Figura 13 Diagrama de secuencia para registrar usuario .....	49
Figura 14 Diagrama de secuencia para iniciar sesión .....	50
Figura 15 Diagrama de secuencia para cambiar la clave .....	50
Figura 16 Diagrama de secuencia para Registrar título .....	51
Figura 17 Diagrama de secuencia para desarrollar el test CIPSA .....	51
Figura 18 Diagrama de secuencia para buscar programas .....	52
Figura 19 Diagrama de secuencia para agregar comentarios.....	52
Figura 20 Diagrama de secuencia para seleccionar el programa escogido.....	53
Figura 21 Estructura de los Backing Bean.....	55
Figura 22 Declaración de Variables.....	56
Figura 23 Configuración de cantidad de peticiones en Jmeter para prueba de estrés.....	63
Figura 24 Configuración del host en Jmeter para la prueba de estrés.....	63
Figura 25 Resultado de la prueba de estrés en Jmeter .....	64

## ÍNDICE DE ANEXOS

**Anexo A:** Script de la base de datos

**Anexo B:** Manual de usuario

**Anexo C:** Manual del Cuestionario de Intereses Profesionales

## RESUMEN

El sistema de adjudicación de becas de la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) está basado en procedimientos netamente administrativos, los cuales no permiten o ayudan a los postulantes en la selección de un programa. Una de las principales limitaciones identificadas, es la imposibilidad del SENESCYT en agregar información de todas las universidades (más de 2000) y de todos sus respectivos programas, es por este motivo que el postulante o posible becario, debe ingresar a los sitios web de las universidades, con el fin de obtener información acerca del programa de estudios en el que está interesado. Dado este problema la SENESCYT se ve en la necesidad de buscar una aplicación, la cual ayude a los futuros postulantes a orientar o a su vez filtrar los gustos y aptitudes del postulante para ofertarle maestrías o carreras según un perfil adecuado para el postulante, por esta razón se desarrolló el sistema “AQUINO”, el cual tiene como objetivo principal la simplificación del proceso de búsqueda y postulación para los nuevos becarios de la SENESCYT. El sistema “AQUINO” ha sido desarrollado siguiendo los estándares de la metodología UWE (UML-BASED WEB ENGINEERING), dicho sistema al llegar a su fase final logró demostrar una simplificación del proceso de búsqueda y postulación de becas, esto se debe a que el usuario después de haber creado una cuenta y de registrar todos sus datos personales, cuenta con el acceso a un motor de búsqueda que según una frase o palabra despliega información de los programas de becas que le pueden interesar al becario

### PALABRAS CLAVE:

DESARROLLO DE UN SISTEMA

AQUINO

SENESCYT

UWE

UML

## ABSTRACT

The system of awarding scholarships SENESCYT is based on purely administrative procedures, which don't allow or assist applicants in selecting a program. One of the main limitations identified, is the impossibility of SENESCYT to add information of universities (more than 2000) and all their programs, is for this reason that the applicant or prospective sizar must enter the web sites universities, for the purpose to obtain information about the curriculum in which you are interested. Given this problem SENESCYT seen in the need to find an application, which will help guide future applicants or turn filtering tastes and abilities of the applicant to bid masters or careers according to a profile suitable for the applicant, therefore the development of "AQUINO" system, which has as main objective to simplify the search and nomination process for new sizar SENESCYT. The "AQUINO" system has been developed following the standards of the UWE methodology (UML-based Web ENGINEERING), this system at reach its final phase, showed a simplification of the process of search and postulation of scholarships, this is because the user, after creating an account and register all their personal data, it has access to search engine as a phrase or word displays information about the scholarship programs that may be of interest to the grantee

### **KEYWORDS:**

DEVELOPMENT

AQUINO

SENESCYT

UWE

UML

# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1 ANTECEDENTES

Actualmente el SENESCYT tiene más de siete mil becados en todo el mundo y ha dado más de diez mil becas desde 2010.

El sistema de adjudicación de becas del SENESCYT está basado en procedimientos administrativos los cuales no cuentan con un sistema informático que permita o ayude los postulantes en la selección de un programa u otro. Una de las principales limitaciones identificadas, es la imposibilidad del SENESCYT en agregar información de todas las universidades (más de 2000) y de todos sus respectivos programas (varios por universidad)

Sin embargo tales números estadísticos, no son representativos cuando, según el SENESCYT, existen muchos becarios que habrían gustado de participar en otra selección de programa o universidad. Quizá sea este el principal motivo para alguna desmotivación y un bajo nivel de productividad científica.

### 1.2 PROBLEMÁTICA

SENECYT como institución gestora de fondos para la adjudicación de becas ha publicado un listado de Universidades a los que los ecuatorianos podrían aplicar (universidades reconocidas) con el fin de acceder a una beca, el mismo que solo contiene la dirección web de cada una de estas universidades como información adicional.

El postulante o posible becario, debe así, ingresar a los sitios web de las universidades con el fin de obtener información acerca del programa de estudios en el que está interesado, sin contar con una plataforma donde sea posible acceder a dicha información de forma centralizada, que le permitiese, por ejemplo acceder al listado de aquellas universidades que tienen programas en una determinada área o

ubicadas en una determinada región, etc.; es decir que le permitiese filtrar las universidades, según sus requerimientos.

En este contexto el presente proyecto consiste en proveer a los postulantes Ecuatorianos a becas de una plataforma que por medio de un test se pueda crear un perfil según las aptitudes y gustos del postulante, y por medio de esa información poder ofertarle al postulante opciones de carreras que sean más acordes a su afinidad.

### **1.3 JUSTIFICACIÓN**

Dentro del proceso de postulación y asignación de becas de la SENESCYT existe un grave problema, el cual es que algunos postulantes que sin tener una idea clara sobre en qué especializarse, aplican a una beca. La SENESCYT confiada de que el postulante está seguro sobre la aplicación de la beca asigna el presupuesto necesario para poder financiar la beca, la cual en dicho caso es abandonada por el postulante ya que no cumple con todas las expectativas necesarias o a su vez no es del agrado del postulante.

Dado este problema la SENESCYT se ve en la necesidad de buscar una aplicación, la cual ayude a los futuros postulantes a orientar o a su vez filtrar los gustos y aptitudes del postulante para ofertarle maestrías o carreras según un perfil adecuado para el postulante.

### **1.4 OBJETIVOS**

#### **1.4.1 OBJETIVO GENERAL**

Desarrollar una aplicación web aplicando minería de datos para realizar un test vocacional dirigido a los postulantes a becas del SENESCYT

#### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Desarrollar un test vocacional para la web enfocada a la obtención de preferencias, deseos o tendencias de los posibles postulantes del sistema de Becas del SENESCYT.

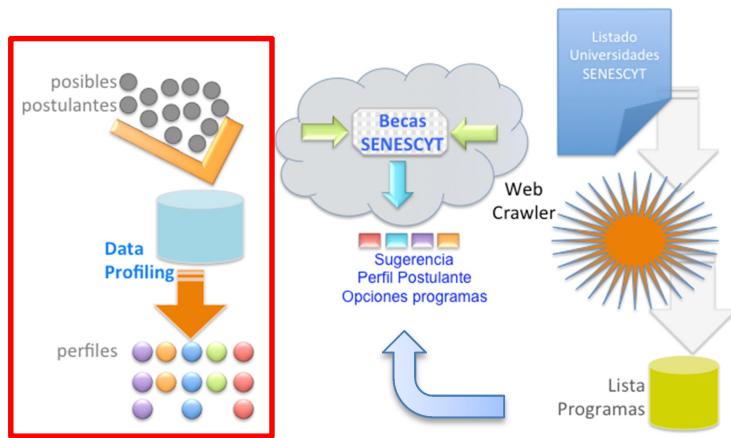
- Determinar las preferencias obtenidas del test vocacional para obtener perfiles estudiantiles de los posibles postulantes al sistemas de becas del SENESCYT.
- Implementar un buscador de programas de postgrados dentro de las universidades que tienen convenio con la SENESCYT
- Desarrollar una página web en la cual el postulante al Sistema de Becas de la SENESCYT pueda acceder a las becas recomendadas según su perfil profesional y actitudinal

## 1.5 ALCANCE

Se realizará el sistema acorde al diseño del test vocacional, el cual permitirá ofertarlo a todas y cada una de las personas que necesiten postularse para el sistema de becas de la SENESCYT enfocado al área de Ciencias de la Computación.

Las tareas a desarrollarse se relacionan con el registro de los postulantes para el sistema de becas de la SENESCYT, la obtención de perfiles de los postulantes y registro en bases de datos para posteriormente desarrollar actividades de minería de datos que permitan encontrar perfiles de líneas de investigación en el listado de programas de maestría y doctorado del SENESCYT.

Dicho sistema será desarrollado como un prototipo dentro de la comunidad de estudiantes de la Facultad de Ciencias de la Computación de la Universidad de las Fuerzas Armadas “ESPE” y el alcance inicial contempla búsquedas de datos solamente con respecto a programas de Ciencias de la Computación en el listado del SENESCYT. Como se muestra en la figura 1 se presenta la estructura general del Sistema AQUINO



**Figura 1. Estructura General del Sistema AQUINO**

## CAPÍTULO 2

### MARCO TEÓRICO

#### 2 APLICACIONES WEB

En la actualidad donde el desarrollo de aplicaciones y sistemas están migrando cada vez más a la Web, dichos sistemas están diseñados para ofrecer una amplia gama de servicios a sus usuarios, los cuales son más exigentes a la hora de verificar la calidad, revisar el desempeño y comprobar la confiabilidad del mismo.

Las aplicaciones Web son sistemas diseñados para funcionar dentro de la World Wide Web (WWW), dichos sistemas pueden ser accedidos desde un navegador Web, lo que significa que el único requisito para poder acceder a una de ellas es tener internet, simplificando el proceso ya que no es necesario una distribución, instalación o actualización del software para cada usuario.

Con el paso del tiempo la cantidad de sitios Web en Internet sigue aumentando considerablemente, esto es dado por la nueva tendencia a existir dentro de la web y también por la existencia de herramientas que ayudan a crear de forma fácil y sencilla dichos sitios, a la par empresas que como agencias de viajes, bancos, hoteles, instituciones educativas, hospitales, entre otros se han visto en la necesidad de subir sus servicios en la nube para mejorar y extender sus operaciones.

##### 2.1.1 CARACTERÍSTICAS DE LAS APLICACIONES WEB

Las aplicaciones Web al igual que otras aplicaciones poseen características las cuales las hacen diferentes de otro tipo de aplicaciones, las cuales son:

- Las aplicaciones web se mantienen en constante evolución; especialmente cuando ya son puestas en producción, tanto a nivel de fondo (contenido e información) como de forma (requerimientos y funcionalidad), por estas razones el tratamiento de los cambios se convierte en un reto en el cual involucra tanto la

parte técnica como la administrativa y a su vez la parte organizacional que las empresas deben tener en cuenta

- Se maneja componentes de tipo multimediales, dicho contenido puede ser texto, gráficos, imágenes o videos; este tipo de contenido se lo debe de tener muy en cuenta ya que de esto dependerá el tiempo de respuesta del sistema
- Están construidas para todo tipo de usuarios, los cuales tienen diferentes requerimientos, conocimientos y a su vez pueden manejar de forma diferente la aplicación, por esta razón las aplicaciones deberán ser construidas y orientadas a presentarse de forma simplificada y sin mucha complicación, de esta forma cada uno de los usuarios podrá utilizarlas sin inconvenientes
- Despliegue de información de bases de datos, ya que la aplicación dependerá de una base de datos debe de ser apta para ejecutar las transacciones necesarias, ya sea las de creación de contenido o las de despliegue de información ya existente.
- Como en todo software los errores o fallos pueden significar una gran pérdida para toda empresa
- El cambio es indispensable para estas aplicaciones, es por esta razón que tanto el desarrollo y a su vez el mantenimiento deben de ser aptos para el cambio ya sea por las tecnologías o por estándares; de esta forma podrán evolucionar para poder brindar un mejor servicio
- Con la constante evolución de la informática las aplicaciones web deben ser adaptables para diferentes tipos de dispositivos, como por ejemplo celulares, computadoras, tabletas, entre otros; y de la misma manera considerar las diferentes velocidades, para mejorar el tiempo de respuesta

- Sus requerimientos de seguridad y privacidad de información deben ser mucho más elevados que los del software no orientado a la Web (Fuseau García & Silva Enríquez, 2015).

### 2.1.2 TIPOS DE APLICACIONES WEB

Basadas en su funcionalidad, las aplicaciones web pueden categorizarse como se mostrará en la tabla 1 (Athula & San, 2001):

**Tabla 1**  
**Categorización de las aplicaciones web**

Características	Descripción	Ejemplo
<b>Informativas</b>	Son páginas en las que se muestra información del acontecer diario como libros en línea, boletines de noticias, catálogos de productos, clasificados.	Monografias.com, CNN
<b>Interactivas</b>	Son la mayoría de páginas que conforman la actualidad del internet, también catalogadas como páginas web 2.0, las cuales poseen formularios juegos en línea, en donde la interacción con el usuario es de importancia.	A10
<b>Transaccionales</b>	En este tipo de páginas web predominan las transacciones donde se puede realizar procesos como matriculación, compras online, banca electrónica, reservación de	MercadoLibre, Banco del Pichincha

	billetes aéreos, mercados electrónicos, subastas.
<b>Orientadas a los flujos de trabajo</b>	Son aplicaciones de tipo Bonitasoft empresarial en las cuales se pueden realizar la administración de inventarios, planeación de agendas en línea y manejo de suministros.
<b>Comunidades en línea</b>	En este grupo se encuentran las páginas web en las que se promueven los grupos de discusión, comunidades, blogs.
<b>Estáticas</b>	Proporcionan únicamente Yahoo información sin la interacción del usuario
<b>De descarga</b>	Este tipo de páginas MediaFire, Mega permiten la descarga de información o archivos
<b>Personalizables</b>	Este tipo de páginas dan al Wix usuario la opción de personalizar el contenido de la misma según sus necesidades.
<b>Con entradas de usuarios:</b>	El usuario puede ingresar eVisaForms información a través de formularios.
<b>Orientadas a transacciones</b>	a Este tipo de páginas dan la E-Banking opción de usar servicios como por ejemplo

<b>Orientadas a servicios</b>	Este tipo de páginas se enfocan a ofrecer servicios para los usuarios.
<b>Portales</b>	Este tipo de páginas son A.O.L de tipo intermediario ya que el usuario podrá navegar en la misma y ésta, por medio de links le facilitara el acceso a servicios de otras páginas o empresas.
<b>Con acceso a bases de datos</b>	Este tipo de páginas son File Database las que muestran información que principalmente se encuentran dentro de una base de datos.
<b>Almacenes de datos</b>	Son muy parecidas a la Series Danko anterior categoría, a diferencia que el usuario podrá realizar consultas de una gran colección de datos.

---

Fuente: (Fuseau García & Silva Enríquez, 2015)

### 2.1.3 MODELADO DE APLICACIONES WEB

Con el pasar de los años las aplicaciones web han evolucionado notablemente, es por esta razón que el desarrollo y mantenimiento de las mismas se han convertido en un proceso más extenso y a veces difícil, ya que al existir factores externos, estos representan un reto de mayor complejidad que otro tipo de aplicaciones.

La esencia de la ingeniería Web es administrar satisfactoriamente la diversidad y complejidad del desarrollo de aplicaciones Web, y por tanto evitar fallos potenciales que podrían tener implicaciones serias (Brandon, 2007).

El proceso de modelado como en todo tipo de aplicaciones permite realizar una descripción detallada sobre todo el proceso necesario para el desarrollo de aplicaciones, dicho proceso se basa en la estructuración de elementos y reglas para la construcción de todos los diferentes tipos de modelos existentes.

Como en todo desarrollo de software si existe un correcto proceso de modelado y a su vez realizado de forma adecuada, dicho proceso significará una reducción notable de los esfuerzos ya sea a nivel de costos y a su vez de tiempos necesarios para el desarrollo de la aplicación web. Esto es factible ya que al tener un proceso más elaborado y estructurado permitirá obtener un software útil y a la vez coherente para el usuario final. Para esto, los diferentes modelos que se realizan deberán comunicar de forma clara todos los requerimientos tanto funcionales como no funcionales de la aplicación, de la misma manera los modelos deben seguir un estándar para que cualquier persona con conocimientos de desarrollo puedan comprenderlos.

El tener un correcto análisis de requerimientos se deriva en un proceso de modelado adecuado, lo cual ayuda a evitar problemas de navegación, dificultades en las búsquedas, de usabilidad, brechas de seguridad, los cuales son errores muy comunes durante el proceso de desarrollo ya que no existe información adecuada para evitarlos.

El proceso de desarrollo de una aplicación Web no implica únicamente el diseño visual y de interfaces, sino también un proceso de planeación, documentación, diseño, pruebas, aseguramiento de la calidad, evaluación del desempeño y mantenimiento que determinarían una aplicación con atributos de calidad como: usabilidad, navegabilidad, accesibilidad, escalabilidad, mantenibilidad, compatibilidad, interoperabilidad, seguridad y confiabilidad (Fuseau García & Silva Enríquez, 2015).

## 2.2 METODOLOGÍA UWE (UML-BASED WEB ENGINEERING)

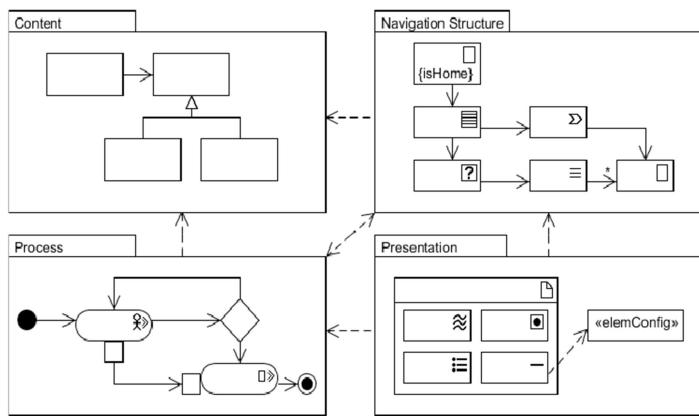
Desde el año de 1960 el desarrollo de sistemas comenzó a ser normado, especificando un conjunto de técnicas de modelado para sistemas que permitan el correcto desarrollo de un software de calidad, en el cual se deberá incluir heurísticas para la construcción utilizando correctos criterios de comparación de modelos de sistemas. Estas técnicas con el paso del tiempo y la aparición de nuevas formas de desarrollo de sistemas se comenzaron a adoptar como metodologías, las cuales permiten el desarrollo de sistemas que sean de calidad.

En la actualidad una metodología es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información (Wikipedia, 2015), con el pasar de los años y según las necesidades se han creado diferentes tipos de metodologías ya sea para aplicaciones de escritorio, móvil, web o a su vez de tipo ágil o normal.

Para el desarrollo de este sistema se ha escogido la metodología ágil UWE, la cual es la Ingeniería web basada en UML fue presentada por Nora Koch en el año 2000. Esta metodología utiliza un paradigma orientado a objetos, y está orientada al usuario. Está basada en los estándares UML y UP (Proceso Unificado), cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas (Quiroga, 2015).

El enfoque UWE proporciona una notación de dominio específico, un proceso de desarrollo dirigido por modelos, y soporte de herramientas para la ingeniería de aplicaciones Web. Las principales razones para el uso de los mecanismos de extensión del UML es debido a que es una de las técnicas de modelado más conocida internacionalmente, además que la creación de un lenguaje de modelado propietario no era necesario, debido a que se acepta como si fuera un estándar a UML en el desarrollo de sistemas de software, además de facilitar su flexibilidad a la hora de definir un lenguaje de modelado específico de dominio Web, sin mencionar su amplio apoyo de modelado visual por herramientas CASE UML existentes. UWE utiliza "puro" notación UML y tipos de diagramas UML siempre que sea posible para el análisis y diseño de aplicaciones Web, es decir, sin las extensiones de cualquier tipo. Por las características específicas de la Web, como nodos y enlaces de

la estructura de hipertexto, el perfil UWE incluye estereotipos, valores etiquetados y restricciones definidas por los elementos de modelado. La extensión UWE cubre la navegación, la presentación, los procesos de negocio y los aspectos de adaptación. La notación UWE se define como una extensión "ligero" de la UML (UWE, 2015). La figura 2 muestra un ejemplo de modelado de una aplicación web.



**Figura 2. Descripción general de los modelos (UWE, 2015)**

### 2.2.1 INTEGRACION DE UWE

Para la integración de UWE se utiliza principalmente UML el cual es aceptado para todo proceso de desarrollo de software, además de tener la flexibilidad para la definición de un lenguaje de modelamiento específico en el dominio WEB, y un gran soporte con las herramientas existentes de UML CASE.

Adicionalmente UWE utiliza las siguientes características:

- XMI como modelo de intercambio de formatos
- MOF para los metamodelos
- Los principios de la aproximación MDA (dirigida por el modelo)
- El modelo de transformación del lenguaje QVT y5.

Para las características de aplicaciones WEB específicas, como nodos y vínculos de la estructura de hyper-texto, el perfil UWE incluye:

- Estereotipos
- Valores marcados
- Limitaciones definidas para los elementos de modelamiento.

Para la parte del modelamiento en la metodología UWE de las características adaptativas se las realizan de manera no invasiva, usando técnicas de AOM (modelamiento orientadas por aspectos por sus siglas en inglés), siguiendo el principio separación de preocupaciones UWE propone construir un modelo adaptativo para sistemas personalizados o dependientes del contexto y después entrelazar los modelos.

## 2.2.2 FASES PARA LA IMPLEMENTACIÓN DE UWE

Una de las características de la metodología UWE es que cubre todo el ciclo de vida para las aplicaciones web enfocándose en dar facilidad a aplicaciones personalizadas o adaptativas. Es por esta razón que la metodología UWE está dividida en 6 diferentes etapas o fases, las cuales son:

- 1) **Captura, análisis y especificación de requisitos:** En esta etapa, se recolecta y se especifican todas las características funcionales y no funcionales que tendrá la aplicación web a desarrollarse.

Una de las características de esta metodología es que se tiene mucho cuidado con la información, la navegación, la adaptación y la interfaz de usuario, así como algunos requisitos adicionales. Además como principalmente se basa en UML también se dedica al estudio de los casos de uso, la generación de los glosarios y el desarrollo de prototipos para la interfaz de usuario.

- 2) **Diseño del sistema:** En esta fase se realiza la especificación de requisitos, la cual se produce por el análisis de los requerimientos que se la realiza en la primera fase, adicionalmente también se define como los requisitos se cumplirán, y también la estructura que deberá tener la aplicación web.

- 3) **Codificación del software:** Esta fase es netamente la de desarrollo de la aplicación; que consiste, en llevar a código fuente, todo lo diseñado en la fase anterior.

- 4) **Pruebas:** Como dice su nombre esta fase se dedica principalmente a realizar las pruebas, las cuales aseguran el correcto funcionamiento del sistema.
- 5) **La Instalación o Fase de Implementación:** Esta fase comprende el proceso por el cual el programa desarrollado es transferido apropiadamente y a su vez es inicializado y configurado adecuadamente para que el usuario final pueda darle su respectivo uso.
- 6) **El Mantenimiento:** Una vez instalado el sistema, se procede a realizar el control, mejora y optimización del mismo, que también incluye depuración de errores y defectos que puedan haber pasado desapercibidos en la fase de pruebas de control.

## 2.3 JAVASERVER FACES (JSF)

JavaServer Faces (JSF) es el framework para aplicaciones Web en Java de Sun Microsystems, liberado en Marzo del 2004. JSF usa JSP como la tecnología para el despliegue de páginas aunque también puede usar XUL (XMLbased Userinterface Language). JSF ofrece un marco de trabajo que facilita el desarrollo de aplicaciones, separando las diferentes capas de una arquitectura: presentación, reglas y entidades de negocio (Buenas Tareas, 2012)

### 2.3.1 VERSIÓN ACTUAL

Desde su primera aparición en el 2004 JSF ha ido evolucionando, añadiendo más características y a su vez eliminando errores, por esa razón la versión actual de JSF es la 2.2 siendo actualmente su sexto reléase. Como se muestra en la tabla 2 se lista cada uno de los reléase de JSF, junto a las características de cada uno.

**Tabla 2**  
**Versiones de JSF**

Versión	Fecha	Características
<b>JSF 1.0</b>	11/03/2004	Lanzamiento inicial de las especificaciones de JSF.
<b>JSF 1.1</b>	27/05/2004	Lanzamiento que solucionaba errores. Sin cambios en las especificaciones ni en el renderkit de HTML
<b>JSF 1.2</b>	11/05/2006	Lanzamiento con mejoras y corrección de errores
<b>JSF 2.0</b>	12/08/2009	Lanzamiento con mejoras de funcionalidad y performance y facilidad de uso
<b>JSF 2.1</b>	22/10/2010	Lanzamiento de mantenimiento, con mínimos cambios
<b>JSF 2.2</b>	16/04/2013	Lanzamiento que introduce soporte a HTML 5, Faces Flow, Stateless views y Resource library contracts

Fuente: (Wikipedia, 2014)

### 2.3.2 IMPLEMENTACIONES

Una de las mayores características de JSF es la gran apertura a librerías y a su forma de uso ya que se puede utilizar más de una al mismo tiempo según los deseos o necesidades del desarrollador. En la tabla 3 se muestran las diferentes implementaciones que tiene JSF, las cuales son:

**Tabla 3**  
**Implementaciones de JSF**

Nombre	Desarrollador
<b>MyFaces Tomahawk</b>	Apache
<b>MyFaces Sandbox</b>	Apache
<b>ICEfaces</b>	ICEsoft
<b>RichFaces</b>	RedHat (Framework de código abierto)
<b>AJAX blueprints components</b>	java.net
<b>JSF Extensions</b>	java.net
<b>Ajax4jsf</b>	Desarrollado por java.net y patrocinado por Exadel
<b>RC Faces (Rich Client Faces)</b>	Vedana
<b>PrimeFaces</b>	PrimeTek

Fuente: (Un aporte a Java, 2009)

### 2.3.3 CARACTERÍSTICAS / FUNCIONALIDADES

JSF es un framework orientado a la interfaz gráfica de usuario (GUI), facilitando el desarrollo de éstas, y que sin embargo, realiza una separación entre comportamiento y presentación, además de proporcionar su propio servlet como controlador, implementando así los principios del patrón de diseño Modelo Vista Controlador (MVC).

A pesar de que en JSF el HTML es su lenguaje de marcado por default, no está limitado a éste ni a ningún otro, pues tiene la capacidad de utilizar diferentes renders para los componentes GUI y obtener así diferentes salidas para mandar al cliente.

### 2.3.4 BACKING BEANS

Un Backing Bean es un Bean común de java que sirve de soporte para un objeto manejado dentro de la aplicación. En los Backing Beans se codifican los

comportamientos específicos asociados a cada control del Managed Bean representado por un servlet.

La ventaja de los Backing Beans es que pueden ser compartidos por un mismo Managed Bean, de manera que para diferentes páginas se pueden agrupar comportamientos comunes en un mismo Bean que se comparte con ambos. (nDeveloper)

### 2.3.5 SCOPE

El Scope en aplicaciones JSF permite determinar el ciclo de vida de cada uno de los Beans que tiene la aplicación, permitiendo un mejor manejo de recursos y de accesibilidad a datos. La tabla 4 muestra los tipos de Scope que se puede usar para los Beans:

**Tabla 4**  
**Tipos de Scope**

Scope	Propósito
<b>View</b>	Dura desde que se muestra una página JSF al usuario hasta que el usuario navega hacia otra página. Es muy útil para páginas que usan AJAX
<b>Application</b>	Se guarda la información durante toda la vida de la aplicación web, independientemente de todas las peticiones y sesiones que se realicen.
<b>Request</b>	Comienza cuando se envía una petición al servidor y termina cuando se devuelve la respuesta al usuario. Esto hace que se cree una instancia del bean asociado para cada petición
<b>Session</b>	Como su mismo nombre indica, este scope guarda la información desde que el usuario comienza una sesión hasta que ésta termina, ya sea por petición del usuario o por

---

	expiración de la sesión
<b>None</b>	Los beans se instancian cuando son necesitados por otros beans, y se eliminan cuando esta necesidad desaparece

Fuente: (El blog de Ana Cortés, 2011)

### 2.3.6 LENGUAJE EL

“El Unified Expression Language (EL) de Java es un lenguaje de programación de propósito especial utilizado principalmente en aplicaciones web en Java para incrustar expresiones en páginas web.

El lenguaje de expresiones fue llamado originalmente SPEL (Simplest Possible Expression Language), y después simplemente Expression Language (EL). Era un lenguaje de script que permitía el acceso a los JavaBeans a través de JSP. Desde JSP 2.0, se ha utilizado dentro de las etiquetas JSP para separar el código Java de JSP, y para permitir un acceso más fácil a los componentes de Java (que en código Java).

Con los años, el lenguaje de expresiones ha evolucionado para incluir una funcionalidad más avanzada y fue incluido en la especificación JSP 2.0. El scripting se hizo más fácil para los diseñadores de contenido web que tienen poco o prácticamente ningún conocimiento del núcleo del lenguaje Java.

Durante el desarrollo de JSP 2.0, la tecnología JavaServer Faces fue publicada, la cual también necesitaba un lenguaje de expresiones, pero el lenguaje de expresiones definido en la especificación JSP 2.0 no satisfacía todas las necesidades para el desarrollo con la tecnología JSF. La limitación más obvia es que sus expresiones son evaluadas de inmediato. Y también, los componentes de JSF necesitan una manera de invocar métodos de objetos del lado del servidor. Un lenguaje más poderoso fue creado con las siguientes nuevas características:

- Expresiones diferidas, que no se evalúan de inmediato
- Expresiones que pueden establecer (set), así como recuperar datos (get)
- Expresiones de métodos, que pueden invocar métodos” (Wikipedia, 2013)

### 2.3.7 CICLO DE VIDA

Java Server Faces posee un ciclo de vida muy similar al de una página JSP: el cliente hace una petición HTTP y el servidor responde con la página en HTML. Sin embargo, debido a las características que ofrece JSF, el ciclo de vida incluye algunos pasos más. El proceso de una petición estándar incluye seis fases, las cuales se muestran en la figura 3.

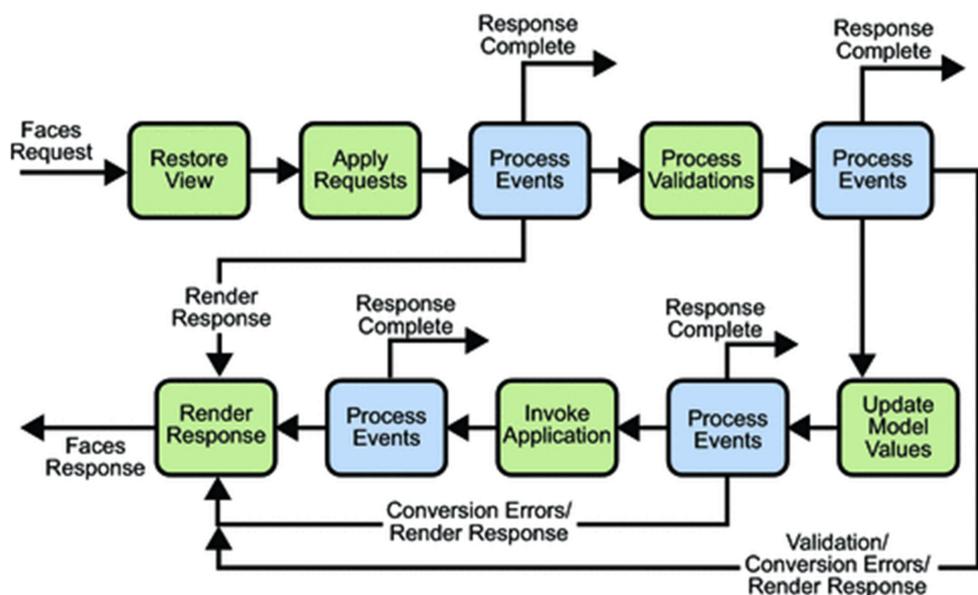


Figura 3 Ciclo de vida JSF

Dentro del ciclo de vida de JSF se mencionan 6 procesos principales y 4 subprocesos llamados “Processs Events”, los cuales representan la ejecución de cualquier evento producido durante el ciclo de vida. La tabla 5 muestra la descripción de la funcionalidad de cada uno de los procesos principales.

**Tabla 5**  
**Ciclo de vida JSF**

Proceso	Función
<b>Restore View / Component Tree</b>	Es la primera etapa que se lleva a cabo, e inicia cuando se hace una petición. Su objetivo es la creación de un árbol con todos los componentes de la página en cuestión.
<b>Apply Request Values</b>	Cada uno de los componentes del árbol creado en la fase anterior obtiene el valor que le corresponde de la petición realizada y lo almacena
<b>Process Validations</b>	Después de almacenar los valores de cada componente, estos son validados según las reglas que se hayan declarado
<b>Update Model Values</b>	Durante esta fase los valores locales de los componentes son utilizados para actualizar los Beans que están ligados a dichos componentes. Esta fase se alcanzará únicamente si todas las validaciones de la etapa anterior fueron exitosas
<b>Invoke Application</b>	Se ejecuta la acción u operación correspondiente al evento inicial que dio comienzo a todo el proceso
<b>Render Response</b>	La respuesta es renderizada y se regresa al cliente. Dependiendo del éxito o fracaso de las tareas en cada una de las fases del ciclo de vida, el flujo normal descrito puede cambiar hacia caminos alternos según sea el caso

Fuente: ( Colección de Tesis Digitales)

### 2.3.8 FACELETS

Está centrado en la creación de árboles de componentes y el entrelazamiento de contenido con el complejo ciclo de vida de JSF. Facelets remplaza a JSP con una muy simple API que es el reflejo de principios simples y esto incorpora numerosas características que facilitan el desarrollo. Entre las diferencias existentes por las que usar Facelets antes que JSP para el desarrollo de JSF se encuentran:

- No depende del contenedor, Facelets puede trabajar con cualquier implementación y versión de JSF.
- La interacción entre JSP y JSF puede traer problemas.
- Facelets provee templates, que pueden ser reutilizados
- Su simpleza maximiza la reutilización, es extensible y escalable.
- Facelets soporta EL (Unified Expression Language)
- Facelets no requiere ninguna librería especial para el renderizado.
- Usa XHTML para la creación de páginas web
- Soporta la librería de etiquetas de facelets, además de Jsf y JSTL
- Soporta expresiones EL
- Plantillas para páginas y componentes
- Reutilización de código, por el uso de plantillas y la composición de componentes
- Muy rápido de compilar y renderizar, con validaciones en tiempo de compilación (No todo código, 2013).

### 2.3.9 PLANTILLAS

Una de las características de los facelets es la reutilización de código pre-programado que puede ser usado como plantillas o temas, dichas plantillas se pueden usar desde JSF 1.2, y se basa en el uso de algunas etiquetas, las cuales se encuentran definidas en la tabla 6:

**Tabla 6**  
**Etiquetas UI**

Etiqueta	Definición
<b>ui:composition</b>	Envuelve un conjunto de componentes para ser reutilizados en otra página. Se utiliza para hacer referencia a una plantilla. Todo lo que quede fuera de la etiqueta ui:composition no será renderizado
<b>ui:define</b>	Define un contenido nombrado para ser insertado en una plantilla, su contenido es un conjunto de componentes y se identifica con un nombre. Ese conjunto de componentes será insertado en una etiqueta ui:insert con el mismo nombre
<b>ui:insert</b>	Declara un contenido nombrado que debe ser definido en otra página
<b>ui:decorate</b>	Esta etiqueta sirve para hacer referencia a una plantilla, como la etiqueta ui:composition, solo que con ui:decorate lo definido antes y después de la etiqueta sí será renderizado
<b>ui:param</b>	Ayuda en la declaración de parámetros y su valor en el uso de plantillas y componentes por composición
<b>ui:include</b>	Es una etiqueta que sirve para incluir en

---

una página el contenido de otra, como si el contenido de esta última formase parte de la primera.

---

### 2.3.10 COMPONENTES PRINCIPALES

De igual forma que las plantillas, JSF cuenta con código pre-programado, el cual, dependiendo del tipo de implementación que se esté usando puede variar ligeramente su etiqueta, La tabla 7 muestra las principales etiquetas de tipo h (html) más comunes dentro de la programación en JSF.

**Tabla 7**  
**Etiquetas Html**

Etiqueta	Descripción
h:commandButton	Un botón al que podemos asociar una acción
h:commandLink	Un enlace hipertexto al que podemos asociar una acción
h:dataTable	Crea una tabla de datos dinámica con los elementos de una propiedad de tipo Array o Map del bean
h:form	Define el formulario JSF en la página JSP
h:graphicImage	Muestra una imagen jpg o similar
h:inputHidden	Incluye un campo oculto del formulario
h:inputSecret	Incluye un campo editable de tipo contraseña (no muestra lo que se escribe)
h:inputText	Incluye un campo de texto normal
h:inputTextarea	Incluye un campo de texto multilínea
h:message	Imprime un mensaje de error en la página (si se ha producido alguno)
h:messages	Imprime varios mensajes de error en la

	página, si se han producido
h:outputFormat	Muestra texto parametrizado. Utiliza la clase java.text.MessageFormat de formateo
h:outputLabel	Muestra un label
h:outputLink	Crea un enlace hipertexto
h:outputText	Muestra un texto básico
h:panelGrid	Crea una tabla con los componentes incluidos en el panelGrid
h:panelGroup	Agrupa varios componentes para que cierto componente los trate como un único componente (por ejemplo para meter varios componentes en una celda de un panelGrid)
h:selectBooleanCheckbox	Crea una casilla con dos estados: activado y desactivado
h:selectManyCheckbox	Crea un conjunto de casillas activables
h:selectManyListbox	Crea una lista que permite seleccionar múltiples elementos
h:selectManyMenu	Crea una lista desplegable de selección múltiple
h:selectOneListbox	Crea una lista en la que se puede seleccionar un único elemento
h:selectOneMenu	Crea na lista desplegable de selección
h:selectOneRadio	Crea una lista de botones, redondos normalmente, excluyentes

Fuente: (PrimeFaces, 2015)

### 2.3.11 VALIDADORES

Los validadores están definidos por la interfaz javax.faces.validator.Validator y sus múltiples implementaciones. El proceso de validación se asegura que el dato introducido en el correspondiente componente es correcto según la lógica de la aplicación. El proceso de validación ocurre antes de que el FrameWork asigne los

valores introducidos en el formulario a las propiedades del bean y justo después de que se hayan aplicado las conversiones, en el caso de que haya. Las validaciones aseguran que un dato introducido en un formulario JSF tenga un valor correcto. (Adictos al trabajo, 2006)

### **2.3.12 CONVERSORES**

Los conversores están definidos por el interfaz javax.faces.convert.Converter y sus múltiples implementaciones. Las conversiones aseguran que el tipo de un dato introducido en un formulario JSF sea el correcto, es decir, que el dato tipo cadena del formulario corresponde con el tipo JAVA esperado, y que está especificado en la propiedad correspondiente del bean. Los Conversores (implementaciones de la interfaz Converter) son los componentes que se encargan de hacer estas transformaciones (cadena Tipo JAVA y viceversa). JSF invoca a los Conversores antes de efectuar las validaciones y por lo tanto antes de aplicar los valores introducidos a las propiedades del bean. En el caso de que un dato tipo cadena no se corresponda con el tipo JAVA apropiado, el Conversor correspondiente lanzará un ConversionException y el componente se marcará como invalidado. (Adictos al trabajo, 2006)

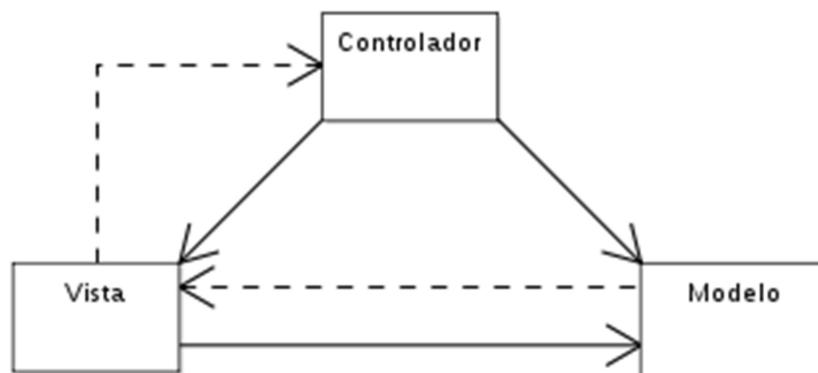
## **2.4 MODELO VISTA CONTROLADOR (MVC)**

El Modelo Vista Controlador (MVC) es un patrón de desarrollo de software que fue desarrollado en 1979 por Trygve Reenskaug. Este patrón separa en diferentes capas la programación dividiendo los datos y la lógica del negocio de la aplicación de la capa de la interfaz de usuario. Su primera implementación fue realizada en el software Smalltalk-76 en los laboratorios Xerox.

### **2.4.1 CAPAS**

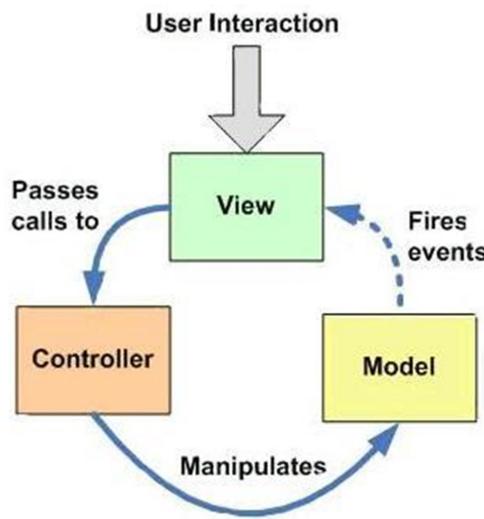
El patrón de software MVC está compuesto por 3 capas, las cuales tienen un propósito específico dentro del desarrollo de software, además de la separación de funciones y características para el desarrollo, estas capas son:

- **Modelo:** Es la representación específica del dominio de la información sobre la cual funciona la aplicación, por dicha razón esta capa controla todo tipo de acceso a la información y a su vez otorga privilegios de acceso (lógica del negocio) para poder realizar consultas o actualizaciones.
- **Vista:** La capa de la vista es la más externa, donde se presenta la interfaz de la aplicación donde el usuario podrá interactuar con la aplicación, la cual muestra la información dada por la capa de modelo.
- **Controlador:** La capa controlador se puede decir que es la más importante de la aplicación ya que esta capa es la que siempre está pendiente de lo que sucede entre la vista y el modelo, encargándose de responder las peticiones realizadas por el usuario para crear instancias de los modelos necesarios con el fin de devolver una respuesta a la vista y que esta pueda ser presentada al usuario.



**Figura 4 MVC Estructura de comunicación entre capas (Wikipedia, 2015)**

Como se muestra en la figura 4 El patrón Modelo Vista Controlador divide en 3 capas la programación de la aplicación, la comunicación entre las capas se encuentran representadas por flechas de tipo sólidas que representan una comunicación directa; mientras que las líneas entrecortadas representan una comunicación indirecta.



**Figura 5 Ciclo de vida MVC (Israel, 2013)**

## 2.5 BASES DE DATOS

Con el paso del tiempo y el avance del desarrollo de software, la necesidad de tener acceso a grandes cantidades de información de diversas temáticas y a su vez la necesidad de dejar de lado la antigua práctica del guardado de información en archivos, se desarrolló una nueva forma tanto para el acceso como para el guardado de la información.

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente (Silberschatz, Korth, & Sudarshan, 2002).

El objetivo del diseño de un sistema de bases de datos es para poder gestionar grandes cantidades de información de una forma fácil y sencilla. Para poder realizar dicha tarea se ve implicado en el proceso la definición de una estructura para la base de datos, que almacenará toda la información para la cual está siendo construida, además la estructura contendrá información sobre la provisión de mecanismos los cuales ayudarán en el proceso de la manipulación de la información.

De la misma forma que cualquier software los sistemas de bases de datos no están exentos a ataques informáticos, por esta razón deberán proporcionar tanto seguridad en la fiabilidad, confidencialidad y disponibilidad de la información almacenada. Es por esta razón que de la misma forma que en el desarrollo de software existen normas y reglas para la creación de los mismo, las cuales se aplican principalmente durante el proceso de modelado de la bases de datos, estas reglas ayudan en el desarrollo de la estructura de una base de datos, haciéndola más fiable y estable en el momento de realizar las transacciones necesarias.

### **2.5.1 MODELAMIENTO DE BASES DE DATOS**

El diseño de una base de datos es un proceso complejo, esto se debe a que durante esta tarea se deben revisar factores ya sea de acceso, eficiencia o a su vez de redundancia. Una base de datos puede ser tan compleja como sea necesario, esto dependerá del ámbito de la aplicación o a su vez de la experticia de la persona que la esté desarrollando.

Un modelo de datos es básicamente una “descripción” de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos (Guerrero & Vásquez, 2015).

Para el modelamiento de las bases de datos principalmente se realizan 3 diferentes modelos, con los cuales se puede representar de forma acorde la base de datos, estos modelos son:

- **Modelo conceptual (entidad relación):**

Para realizar el modelo conceptual se debe obtener la estructura básica de la información para la base de datos, este proceso de lo realiza de forma independiente del gestor de bases de datos a usar. El modelo conceptual es completamente independiente del tipo de base de datos, ya que esta puede ser relacional, no relacional, orientada a objetos, etc. Es por esta razón que el

modelo conceptual permite concentrarse únicamente en la estructura de la información.

- **Modelo Lógico**

El diseño del modelo lógico es la continuación del modelo conceptual, en esta etapa el modelo se adapta a la tecnología del gestor de bases de datos, el cual puede ser relacional, no relación, orientado a objetos, entre otras. Una de las tareas en esta fase es la de determinar las diferentes relaciones que tendrán las entidades con sus respectivos atributos y claves ya sean primarias o foráneas.

- **Modelo Físico**

El diseño del modelo físico es la transformación de la estructura del modelo lógico, este proceso se lo realiza para mejorar la eficiencia de la base de datos. Una de las principales tareas a realizarse dentro de esta etapa es la transformación de relaciones de muchos a muchos en relaciones de uno a muchos

## 2.5.2 NORMALIZACIÓN

La normalización es el proceso por el cual se corrige y se mejora el rendimiento de una base de datos, la cual para que sea optima se recomienda realizar este proceso hasta la tercera forma normal, a pesar de esta recomendación este proceso tiene cinco pasos o también llamadas cinco formas normales, las cuales dividen todo el proceso de la normalización en tareas sencillas, las cuales son:

- **Primera forma normal:** Esta forma normal tiene 3 tareas, en las cuales se solicita:
  - Eliminación de los campos repetido de cada una de las tablas
  - Creación de una única tabla por cada grupo de datos que mantengan una relación en común.
  - Creación de una clave primaria para cada tabla
- **Segunda forma normal:** Esta forma normal tiene 2 tareas, en las cuales se solicita:

- La creación de tablas para el conjunto de datos que sean necesarios en varios registros
- Relacionar las tablas creadas anteriormente según sea necesario por medio de claves foraneas
- **Tercera forma normal:** Esta forma normal consta de una única tarea la cual es la eliminación de campos que no tengan una relación de dependencia de la clave
- **Cuarta forma normal:** Esta forma normal dice que en las relaciones muchos a muchos en una misma tabla no pueden ser almacenados entidades independientes
- **Quinta forma normal:** Esta forma normal dice que la tabla original debe ser reconstruida desde las tablas resultantes en las cuales ha sido troceada (UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH, 2015).

### 2.5.3 MINERÍA DE DATOS

La minería de datos o exploración de datos es un campo de las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos. El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior. Además de la etapa de análisis en bruto, que involucra aspectos de bases de datos y de gestión de datos, de procesamiento de datos, del modelo y de las consideraciones de inferencia, de métricas de Intereses, de consideraciones de la Teoría de la complejidad computacional, de post-procesamiento de las estructuras descubiertas, de la visualización y de la actualización en línea (Wikipedia, 2015).

Una de las características de la minería de datos es que cada caso es totalmente distinto a sus predecesores, es por esta razón que se ha clasificado en 4 diferentes etapas principales, dichas etapas se encuentran detalladas en la tabla 8.

**Tabla 8**  
**Etapas de la minería de datos**

<b>ETAPA</b>	<b>DESCRIPCIÓN</b>
Determinación de los objetivos.	Trata de la delimitación de los objetivos que el cliente desea bajo la orientación del especialista en data mining
Pre procesamiento de los datos.	Se refiere a la selección, la limpieza, el enriquecimiento, la reducción y la transformación de las bases de datos. Esta etapa consume generalmente alrededor del setenta por ciento del tiempo total de un proyecto de data mining
Determinación del modelo.	Se comienza realizando los análisis estadísticos de los datos, y después se lleva a cabo una visualización gráfica de los mismos para tener una primera aproximación. Según los objetivos planteados y la tarea que debe llevarse a cabo, pueden utilizarse algoritmos desarrollados en diferentes áreas de la Inteligencia Artificial
Análisis de los resultados.	Verifica si los resultados obtenidos son coherentes y los coteja con los obtenidos por los análisis estadísticos y de visualización gráfica. El cliente determina si son novedosos y si le aportan un nuevo conocimiento que le permita considerar sus decisiones

---

Fuente: (Sinnexus, 2015)

## 2.6 TEST VOCACIONAL

La orientación vocacional es un conjunto de prácticas destinadas al esclarecimiento de la problemática vocacional. Se trata de un trabajo preventivo cuyo objetivo es proveer los elementos necesarios para posibilitar la mejor situación de elección para cada sujeto. La orientación vocacional puede concretarse de forma individual o grupal, ya que supone actividades ligadas tanto a la exploración personal como al análisis de la realidad a través de información sobre la oferta académica y las particularidades del mercado laboral (Definición de, 2015).

Entre la infinidad de test vocacionales o de profesionalismo se ha escogido el Test CIPSA (Cuestionario de Intereses Profesionales), el cual está basado en los test y cuestionarios de Strong (1941), Kuder (1948) y Thurstone (1947).

El Cuestionario de Intereses Profesionales se enfoca en conocer los distintos tipos de estructura y jerarquías de las preferencias profesionales y al mismo tiempo la evolución de estos valores cognitivos.

Este test ayuda a conocer las habilidades propias del individuo, también se suele emplear otros instrumentos para llevar a cabo la orientación educativa. Así, se hace uso de todo tipo de información sobre las opciones educativas existentes, cómo se encuentra el mercado laboral, las carreras universitarias con más salidas profesionales (Definición de, 2015).

## **CAPÍTULO 3**

### **ANÁLISIS Y DISEÑO DE LA APLICACIÓN**

#### **3.1 INTRODUCCIÓN**

En Este capítulo se presenta la especificación de requerimientos para el sistema desarrollado bajo el nombre “AQUINO”, dicho sistema se lo ha desarrollado con el propósito de satisfacer las necesidades actuales de la SENESCYT y a su vez se ha seguido los estándares de la metodología de desarrollo web UWE.

#### **3.2 PROPÓSITO**

El presente capítulo se lo ha desarrollado con el propósito de definir de manera clara cada uno de los requerimientos, restricciones y funcionalidades del sistema AQUINO. Este capítulo puede ser interpretado como el manual de desarrollo del sistema, obteniendo así después de su aprobación la línea base para posteriores modificaciones o implementación de nuevos módulos.

#### **3.3 ALCANCE**

Se realizará el sistema acorde al diseño del test de intereses vocacionales CIPSA, el cual permitirá ofertarlo a todas y cada una de las personas que necesiten postularse para el sistema de becas de la SENESCYT enfocado al área de Ciencias de la Computación.

Las tareas a desarrollarse se relacionan con el registro de los postulantes para el sistema de becas de la SENESCYT, la obtención de perfiles de los postulantes y registro en bases de datos para posteriormente desarrollar actividades de minería de datos que permitan encontrar perfiles de líneas de investigación en el listado de programas de maestría y doctorado del SENESCYT.

Dicho sistema será desarrollado como un prototipo dentro de la comunidad de estudiantes de la Facultad de Ciencias de la Computación de la Universidad de las

Fuerzas Armadas “ESPE” y el alcance inicial contempla búsquedas de datos relacionados únicamente a programas de Ciencias de la Computación en el listado del SENESCYT.

### 3.4 DESCRIPCIÓN

El sistema se lo ha desarrollado según los siguientes aspectos:

#### 3.4.1 REQUERIMIENTOS FUNCIONALES

El sistema cuenta con los siguientes requerimientos funcionales:

**Tabla 9**

**Requerimiento funcional 1 Registrar usuarios**

<b>Id. Requerimiento</b>	<b>RE01 Registrar usuarios</b>
<b>Descripción</b>	Permite registrar un nuevo usuario en el aplicativo WEB.
<b>Entradas</b>	Datos del usuario: cédula de identidad, nombre, apellido y correo electrónico.
<b>Salidas</b>	Confirmación de registro de cliente exitosamente con un mensaje.
<b>Proceso</b>	El cliente accederá a la pantalla de registro donde ingresará y seleccionará los parámetros descritos anteriormente. El usuario deberá llenar todos los campos remarcados con un asterisco (*), lo que indica que es obligatorio.
<b>Precondiciones</b>	Ninguna
<b>Postcondiciones</b>	Se actualizará la tabla de registro usuario.
<b>Efectos Colaterales</b>	Aumenta un registro de cliente de la base de datos.
<b>Prioridad</b>	Alta
<b>Rol que lo ejecuta</b>	Usuario

**Tabla 10**

**Requerimiento funcional 2 Iniciar sesión**

<b>Id. Requerimiento</b>	<b>RE02 Iniciar sesión</b>
<b>Descripción</b>	Permite al usuario iniciar sesión en el aplicativo WEB.

Continua →

<b>Entradas</b>	Usuario: cédula de identidad Clave: Contraseña enviada al correo.
<b>Salidas</b>	Mensaje de bienvenida con el nombre del usuario.
<b>Proceso</b>	El cliente accederá a la cualquiera de las páginas de la aplicación donde podrá ingresar su nombre de usuario y su contraseña.
<b>Precondiciones</b>	Haberse registrado en la aplicación
<b>Postcondiciones</b>	Se creará un registro nuevo donde conste la fecha del inicio de sesión del usuario en la base de datos.
<b>Efectos Colaterales</b>	Ninguno.
<b>Prioridad</b>	Alta
<b>Rol que lo ejecuta</b>	Usuario

**Tabla 11**  
**Requerimiento funcional 3 Cambiar clave**

<b>Id. Requerimiento</b>	<b>RE03 Cambiar clave</b>
<b>Descripción</b>	Permite al usuario cambiar su clave en el aplicativo WEB.
<b>Entradas</b>	Contraseña antigua que fue enviada al correo (si es la primera vez que inicia sesión), contraseña nueva y repetir la contraseña nueva.
<b>Salidas</b>	Mensaje de confirmación del cambio de clave.
<b>Proceso</b>	El usuario accederá a la cualquiera de las páginas de la aplicación donde podrá ingresar su nombre de usuario y su contraseña.
<b>Precondiciones</b>	Haber iniciado sesión
<b>Postcondiciones</b>	Se actualizará la tabla de registro usuario.
<b>Efectos Colaterales</b>	Ninguno.
<b>Prioridad</b>	Media
<b>Rol que lo ejecuta</b>	Usuario

**Tabla 12**  
**Requerimiento funcional 4 Registrar título**

<b>Id. Requerimiento</b>	<b>RE04 Registrar título</b>
<b>Descripción</b>	Permite al usuario registrar su título de tercer nivel en el aplicativo WEB.
<b>Entradas</b>	Universidad, tipo de título, carrera a la que pertenece, nombre del título.
<b>Salidas</b>	Mensaje de confirmación del proceso.
<b>Proceso</b>	El usuario accederá a su perfil donde se le dará la opción de registrar su título de tercer nivel llenando sus datos.
<b>Precondiciones</b>	Haberse registrado en la aplicación
<b>Postcondiciones</b>	Se actualizará la información del usuario añadiendo sus datos del título de tercer nivel.
<b>Efectos Colaterales</b>	Ninguno.
<b>Prioridad</b>	Alta
<b>Rol que lo ejecuta</b>	Usuario

**Tabla 13**  
**Requerimiento funcional 5 Desarrollar el test de intereses vocacionales CIPSA**

<b>Id. Requerimiento</b>	<b>RE05 Desarrollar el test de intereses vocacionales CIPSA</b>
<b>Descripción</b>	Permite al usuario contestar el test de intereses vocacionales CIPSA en el aplicativo WEB.
<b>Entradas</b>	Contestación de preguntas de opción múltiple.
<b>Salidas</b>	Mensaje de confirmación del proceso.
<b>Proceso</b>	El usuario después de iniciar sesión podrá acceder por medio del menú a la página donde se encuentra el test.
<b>Precondiciones</b>	Haber iniciado sesión en la aplicación
<b>Postcondiciones</b>	Se actualizará la información del usuario añadiendo datos sobre las carreras afines para el usuario.
<b>Efectos Colaterales</b>	Ninguno.

<b>Prioridad</b>	Alta
<b>Rol que lo ejecuta</b>	Usuario

**Tabla 14****Requerimiento funcional 6 Buscar programas de maestrías**

<b>Id. Requerimiento</b>	<b>RE06 Buscar programas de maestrías</b>
<b>Descripción</b>	El usuario puede realizar búsqueda de programas de maestrías según 3 criterios: por su título de tercer nivel, por el resultado del test de intereses vocacionales CIPSA o por palabras claves
<b>Entradas</b>	Selección del tipo de búsqueda que desee el usuario
<b>Salidas</b>	Despliegue de nueva información en pantalla
<b>Proceso</b>	El usuario después de iniciar sesión y de haber realizado el test CIPSA, podrá acceder a una página donde podrá escoger entre las opciones ya mencionadas para la búsqueda de posibles maestrías a seguir.
<b>Precondiciones</b>	Tener una sesión iniciada y haber realizado el test CIPSA
<b>Postcondiciones</b>	Tener conexión con el aplicativo “AQUINO-CRAWLER” el cual se encargará de traer la información resultante de la búsqueda
<b>Efectos Colaterales</b>	Ninguno.
<b>Prioridad</b>	Alta
<b>Rol que lo ejecuta</b>	Usuario

**Tabla 15****Requerimiento funcional 7 Agregar comentarios**

<b>Id. Requerimiento</b>	<b>RE07 Agregar comentarios</b>
<b>Descripción</b>	El usuario podrá agregar comentarios a las opciones escogidas
<b>Entradas</b>	Texto
<b>Salidas</b>	Mensaje de confirmación de guardado de la información

<b>Proceso</b>	El usuario después de haber seleccionado el tipo de búsqueda tendrá la opción de guardar comentarios en las opciones que el desee guardar para revisarlas posteriormente
<b>Precondiciones</b>	Haber buscado programas de maestrías
<b>Postcondiciones</b>	Ninguna
<b>Efectos Colaterales</b>	Ninguno.
<b>Prioridad</b>	Baja
<b>Rol que lo ejecuta</b>	Usuario

**Tabla 16**  
**Requerimiento funcional 8 Seleccionar**

<b>Id. Requerimiento</b>	<b>RE08 Seleccionar la maestría escogida</b>
<b>Descripción</b>	El usuario podrá escoger entre todas las opciones una única opción como la maestría que deseé seguir
<b>Entradas</b>	Selección de una opción
<b>Salidas</b>	Mensaje de confirmación de la selección
<b>Proceso</b>	El usuario podrá escoger entre todas las opciones una única opción como la maestría que deseé seguir
<b>Precondiciones</b>	Tener una sesión iniciada y haber buscado programas de maestrías
<b>Postcondiciones</b>	Ninguna
<b>Efectos Colaterales</b>	Ninguno.
<b>Prioridad</b>	Alta
<b>Rol que lo ejecuta</b>	Usuario

### 3.4.2 RESTRICCIONES GENERALES

Para poder acceder al sistema “AQUINO”, el usuario deberá contar con una conexión a internet y a su vez tener instalado un navegador web actualizado hasta la fecha

### **3.4.3 INTERFACES DE SOFTWARE**

El sistema AQUINO ha sido desarrollado bajo las siguientes características:

- Plataforma Java Enterprise Edition 7
- Java Development Kit 1.8
- Lenguaje de programación Java ServerFaces (JSF)
- Librería de componentes JSF PrimeFaces 5.3
- Librería de componentes JSF PrimeFaces Extensions 4.0.0
- Servidor de aplicaciones Glassfish 4.1
- Base de datos MySQL 5.6

### **3.4.4 INTERFACES DE COMUNICACIÓN**

Dado que el sistema AQUINO es desarrollado para la web esta implementado en un servidor para aplicaciones web, adicionalmente el sistema AQUINO se comunica por medio de Web Services tanto para el envío como para la recolección de información con un sistema complementario, que se encargara de buscar información relacionada con las posibles becas.

## **3.5 METODOLOGÍA UWE**

El desarrollo del sistema AQUINO se lo ha realizado usando la metodología de desarrollo web UWE la cual está basada en UML, dicha metodología cuenta con los siguientes aspectos:

### **3.5.1 DEFINICIÓN DE ACTORES**

En el sistema CIPSA existirá un único tipo de usuario, el cual, contará con los siguientes permisos:

- Posibilidad de iniciar de sesión a su cuenta
- Dentro de su cuenta podrá ver información relacionada con sus datos

- Si no ha realizado el Test De Intereses Vocacionales CIPSA, se le desplegará en el menú la opción para poder realizarlo, caso contrario esta opción se deshabilitara.
- Búsqueda de maestrías y postgrados desde su cuenta.
- Calificación de las opciones resultantes para las maestrías.

### 3.5.2 DEFINICIÓN DE LOS CASOS DE USO

Los casos de uso permiten describir los pasos a seguir para realizar una actividad o proceso dentro de un sistema, es por esta razón que en la Figura 6 se muestra el diagrama global de los casos de uso del sistema AQUINO.

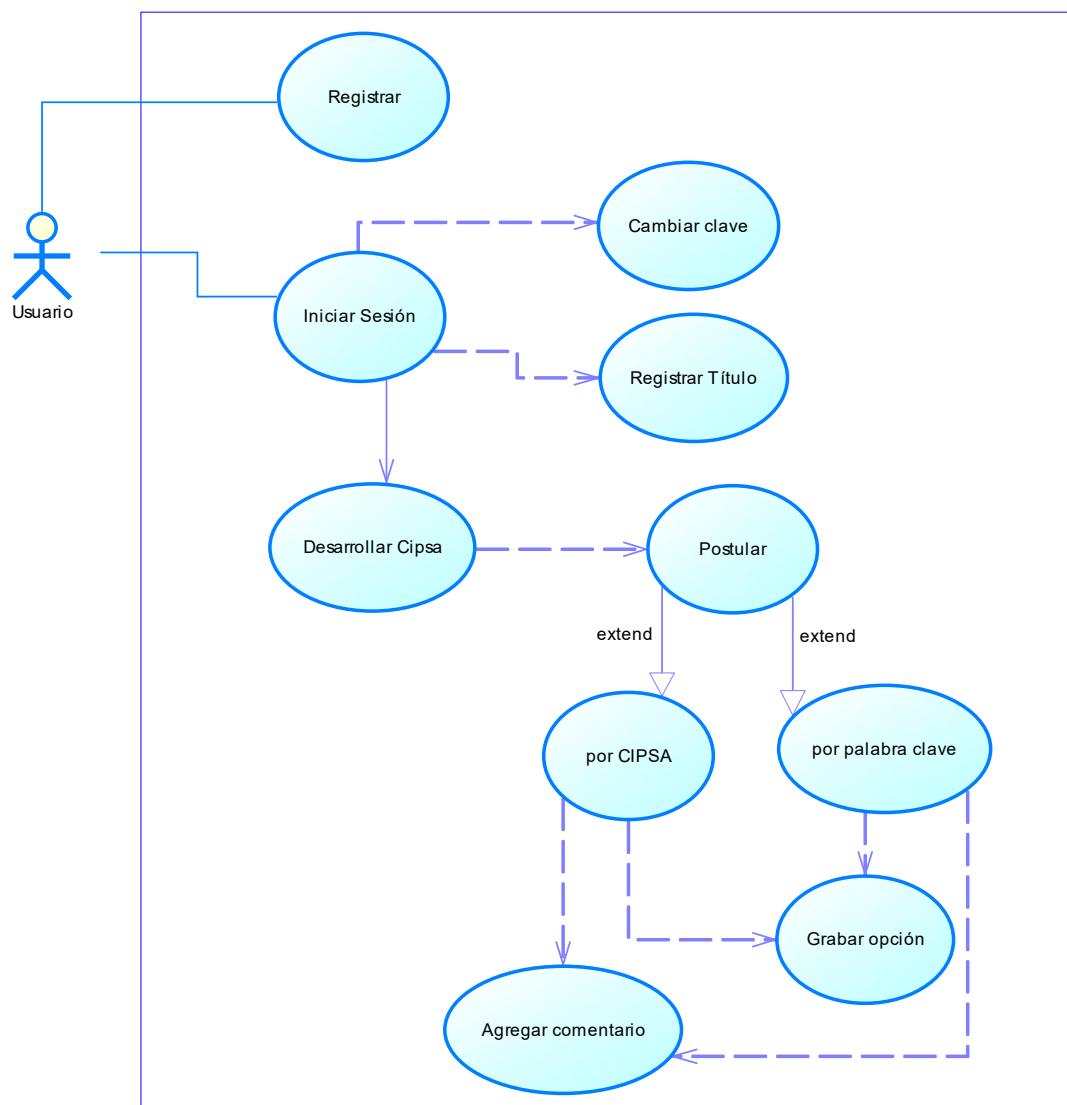


Figura 6 Diagrama global de casos de uso del sistema AQUINO

Para una mejor comprensión de los casos de uso del sistema AQUINO se va a realizar la especificación de uno de los casos de uso, el cual servirá como modelo guía de los demás casos de uso.

**Tabla 17**  
**CU01 Registrar usuarios**

<b>Nombre</b>	CU01 Registrar usuarios
<b>Descripción</b>	Permite registrar un nuevo usuario en el aplicativo WEB.
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1) Ingreso del usuario a la página.</li> <li>2) Clic en la pestaña de registro de usuarios.</li> <li>3) Llenar la información solicitada para crear al usuario</li> <li>4) Aceptar el envío de datos para crear el usuario</li> </ol>
<b>Flujo Alternativo</b>	Ninguno
<b>Post condiciones</b>	Se actualizará la tabla de registro usuario.

### 3.5.3 MODELO LÓGICO

En la tabla 19 se describen cada una de las entidades que tiene la base de datos y que se las puede ver tanto en la representación del modelo lógico como en el modelo físico.

**Tabla 18**  
**Descripción de las entidades de la base de datos**

ENTIDAD	DETALLE
<b>USUARIO</b>	Encargada de recolectar la información personal del usuario para la aplicación, adicionalmente se la puede considerar como la tabla maestra ya que todas las demás dependerán de la misma.

---

<b>SEGURIDAD_USUARIO</b>	Encargada de guardar información acerca del inicio de sesión de un usuario, considerada como una tabla de registro de seguridad.
<b>PAIS</b>	En esta tabla se encuentra almacenado el nombre de cada uno de los países del mundo con su respectivo código.
<b>UNIVERSIDAD</b>	Encargada de guardar la información sobre las universidades de nuestro país Ecuador.
<b>TITULO</b>	Encargada de recolectar la información de cada uno de los títulos de tercer nivel por parte de los usuarios obtenidos en las universidades del Ecuador.
<b>CIPSA</b>	Encargada de guardar la información de los resultados del test de intereses vocacionales CIPSA.
<b>OPCIONES</b>	Encargada de guardar la información resultante de las posibles becas que el usuario le gustaría seguir.
<b>COMENTARIO</b>	Encargada de guardar información adicional que el usuario desee agregar a cada una de las opciones de posibles becas.

---

En la Figura 7 se puede ver cómo se encuentran estructuradas las relaciones entre las tablas de la base de datos, dicho modelo se lo ha autogenerado partiendo del modelo físico (Figura 8).

### 3.5.4 MODELO FÍSICO

En la Figura 8 se puede ver la estructura del modelo físico de la base de datos para la aplicación AQUINO, para la generación de la base de datos se realizó el modelado de la misma partiendo del modelo físico, el cuál fue realizado siguiendo los estándares de normalización y optimización de una base de datos relacional. Como se mencionó en la Tabla 9 la entidad principal de todo sistema es usuario el cual se encarga de unir a todas las demás entidades.

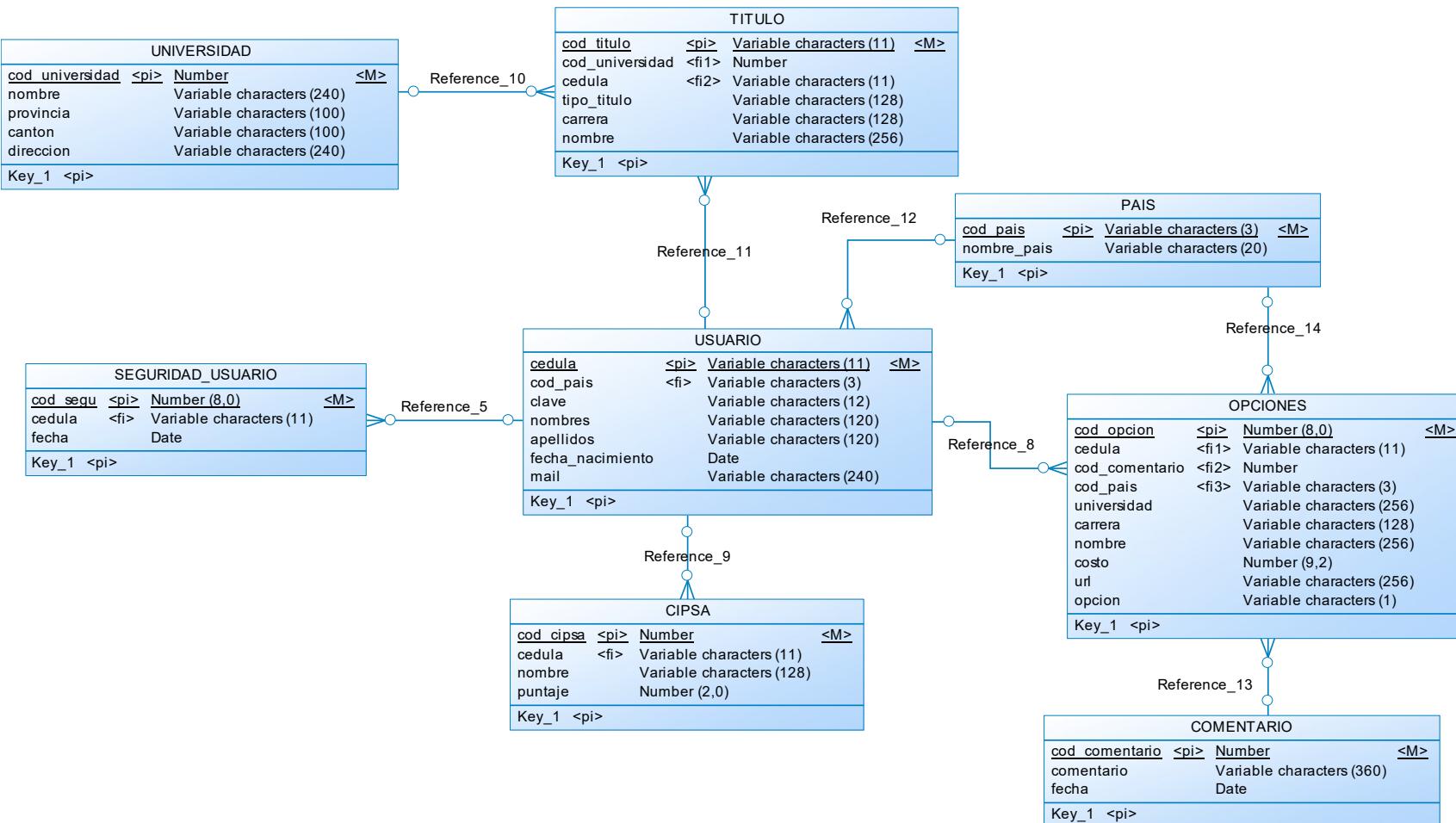


Figura 7 Modelo lógico de la base de datos

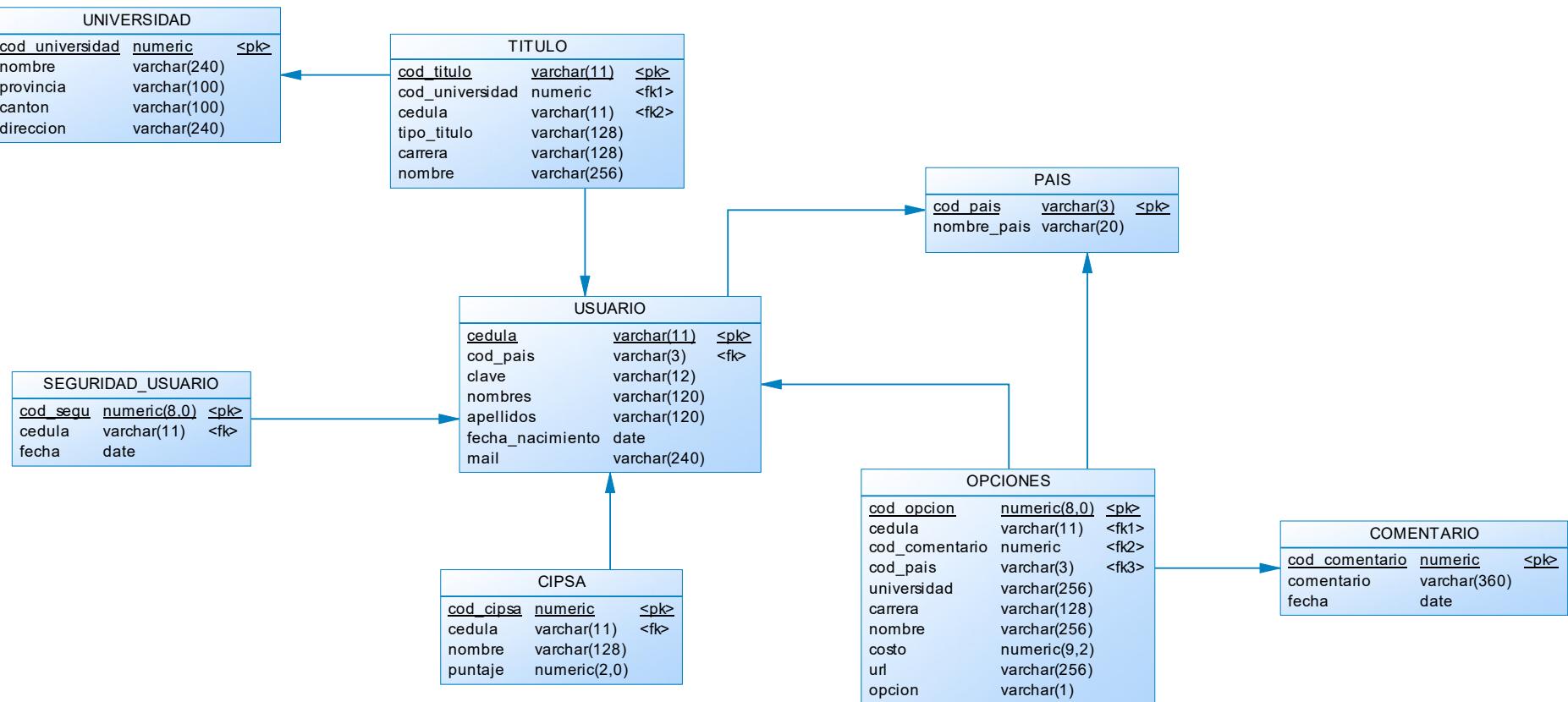


Figura 8 Diagrama Físico de la base de datos

### **3.5.5 SCRIPT DE LA BASE DE DATOS**

Después de la generación de los modelos de la base de datos se ha procedido a generar el script SQL que permitirá la creación y uso de la base, dicho script se lo puede encontrar como parte del anexo A.

### **3.5.6 MODELO DE PRESENTACIÓN**

En la Figura 9 se puede apreciar la estructura general de los templates usados para el despliegue de información de la interfaz en la página web, la página web está compuesta por 4 partes principales, las cuales son:

- Menú: En esta sección estará disponible un menú de acceso a las principales páginas del sistema web AQUINO, adicionalmente dependiendo si el usuario ha iniciado sesión o no se le dará la opción de hacerlo o a su vez de cerrar su sesión.
- Logo: En esta sección estará presente el logo del sistema AQUINO.
- Contenido: En esta sección se incluirá todo el contenido sobre la página que se está visitando.
- Pie de página: En esta sección se encuentran links de interés sobre el Gobierno de Ecuador, ministerios y secretarías. Adicionalmente también se muestra una marca de agua con la firma del autor del sistema muestra, el logo de la campaña “Ecuador Ama la Vida” y la información sobre la Universidad de las Fuerzas Armadas “ESPE”.

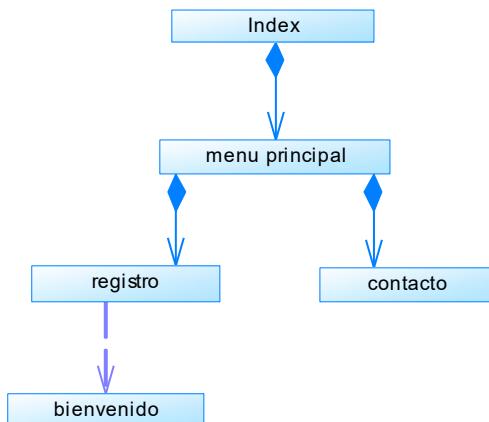


**Figura 9 Modelo de presentación**

### 3.5.7 DIAGRAMA NAVEGACIONAL

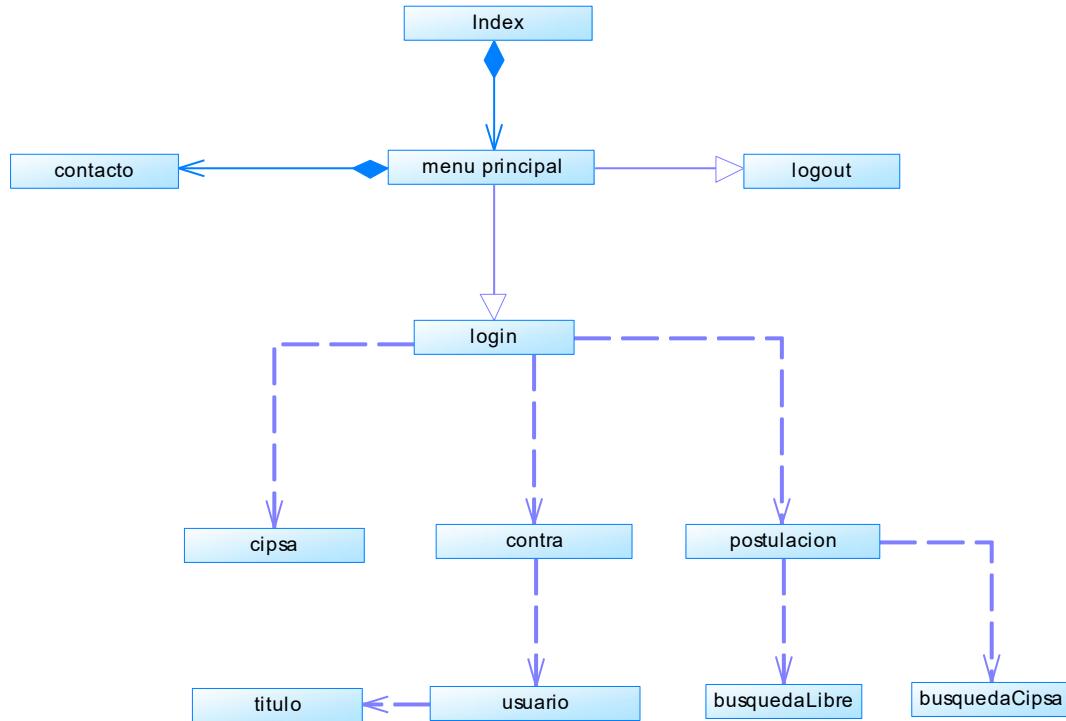
El diagrama navegacional como su nombre indica, es la representación en forma gráfica de cómo se va a poder navegar por todo el aplicativo web. Para el sistema AQUINO se han realizado dos escenarios para la construcción de este diagrama, los cuales son:

#### 3.5.7.1 Usuario no registrado



**Figura 10 Diagrama navegacional, usuario no registrado**

### 3.5.7.2 Usuario registrado



**Figura 11 Diagrama navegacional, usuario registrado**

### 3.5.8 DIAGRAMA DE CLASES

En la Figura 12 se muestra el diagrama de clases, el cual indica la estructura, relaciones y tipos de datos con los que cuenta el sistema AQUINO.

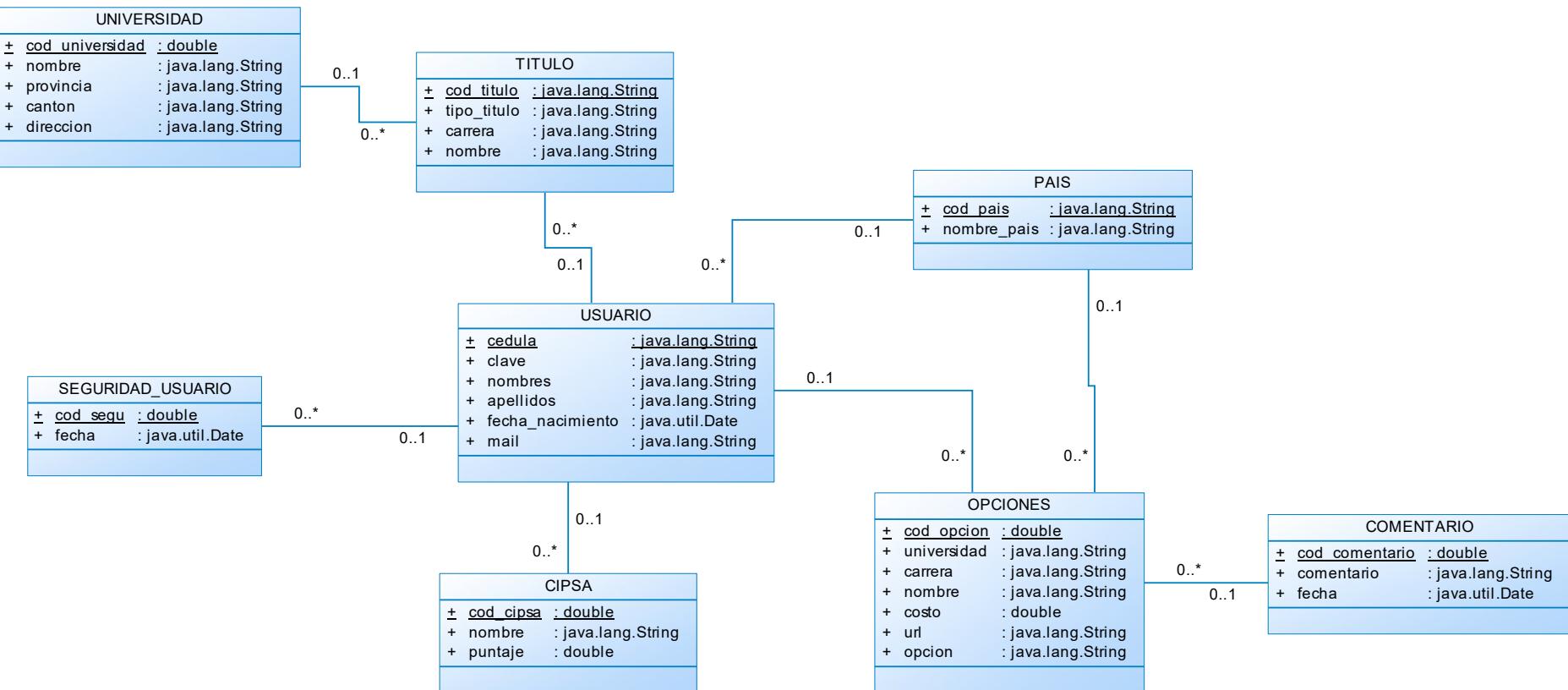
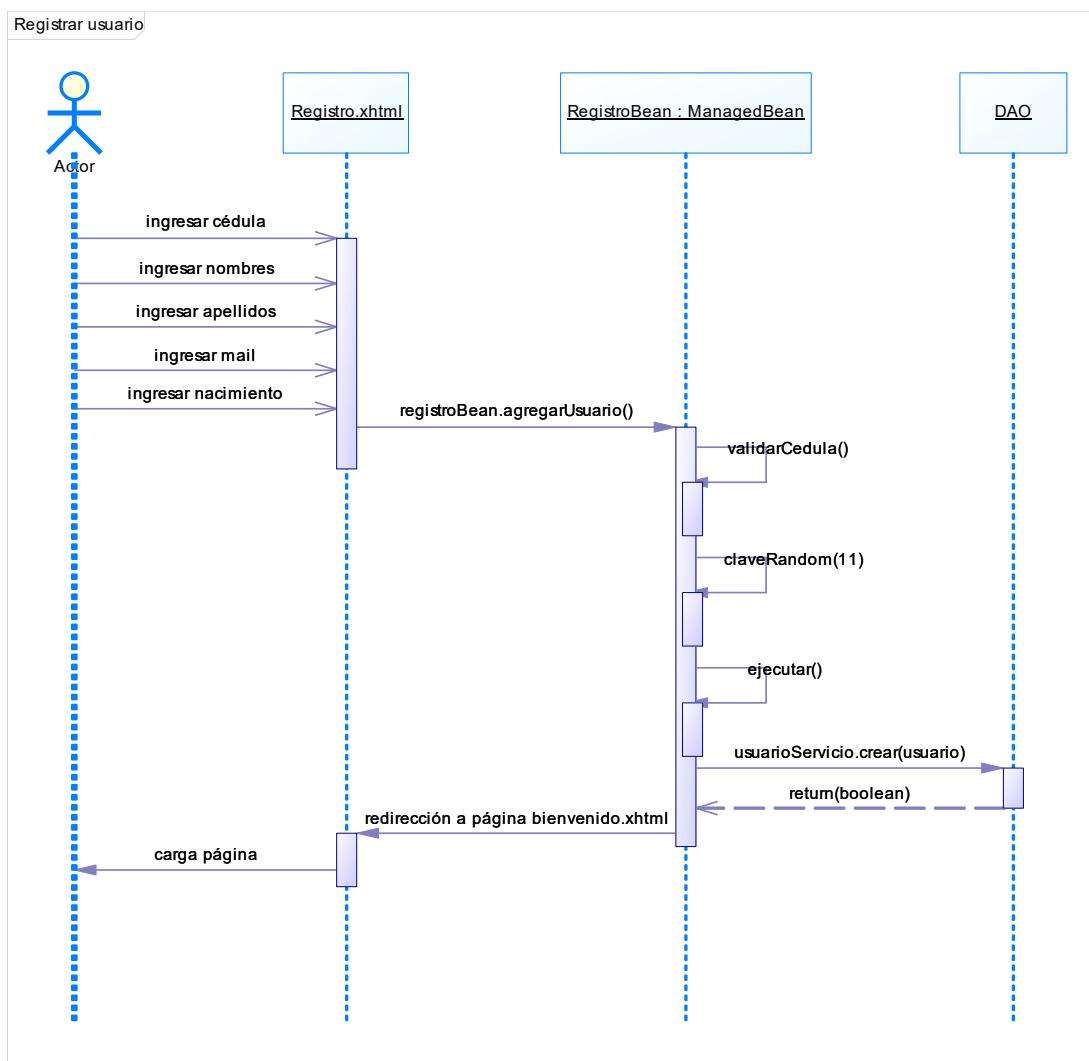


Figura 12 Diagrama de clases del sistema AQUINO

### 3.5.9 DIAGRAMA DE SECUENCIA

Los diagramas de secuencia muestran el flujo de actividades que van a suceder durante todo el proceso para el cumplimiento de un requerimiento o de un caso de uso. A continuación se va a listar cada uno de los diagramas de secuencias de los requerimientos del sistema AQUINO.

#### 3.5.9.1 Registrar usuario



**Figura 13 Diagrama de secuencia para registrar usuario**

### 3.5.9.2 Iniciar sesión

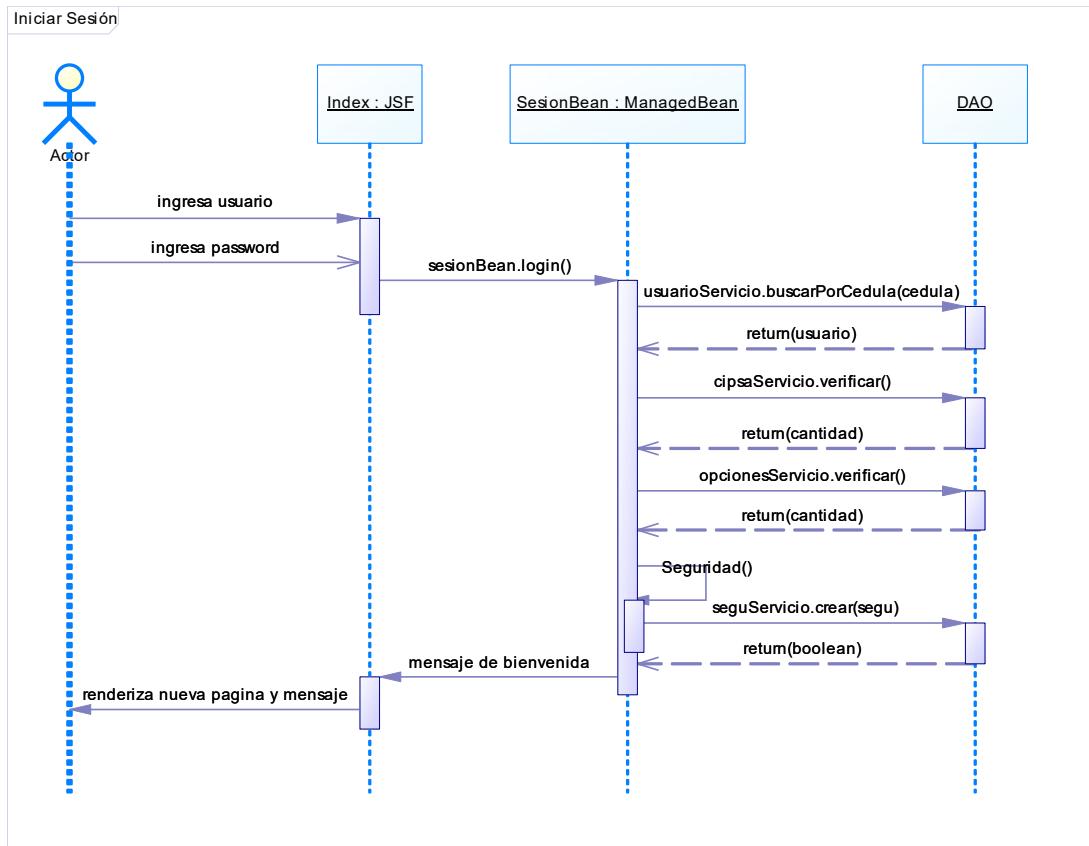


Figura 14 Diagrama de secuencia para iniciar sesión

### 3.5.9.3 Cambiar clave

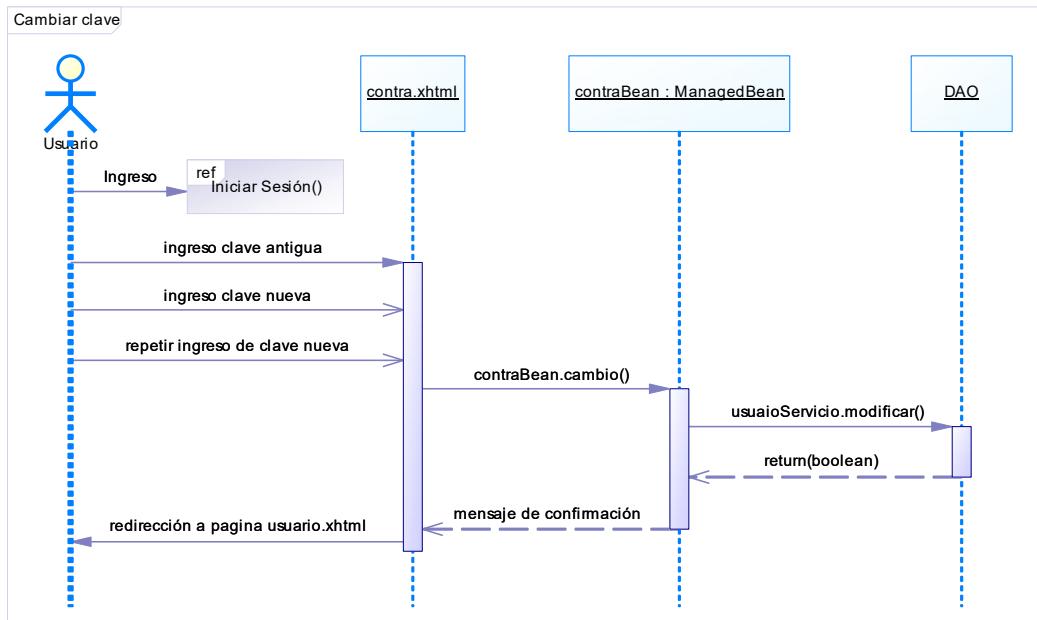


Figura 15 Diagrama de secuencia para cambiar la clave

### 3.5.9.4 Registrar título

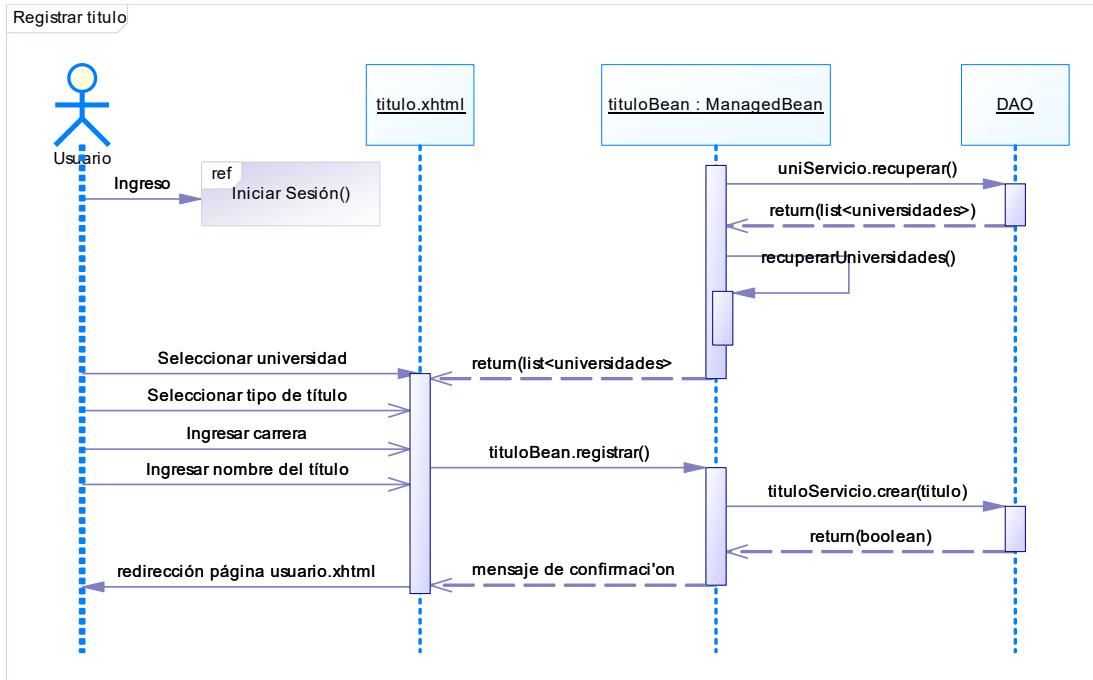


Figura 16 Diagrama de secuencia para Registrar título

### 3.5.9.5 Desarrollar el test de intereses vocacionales CIPSA

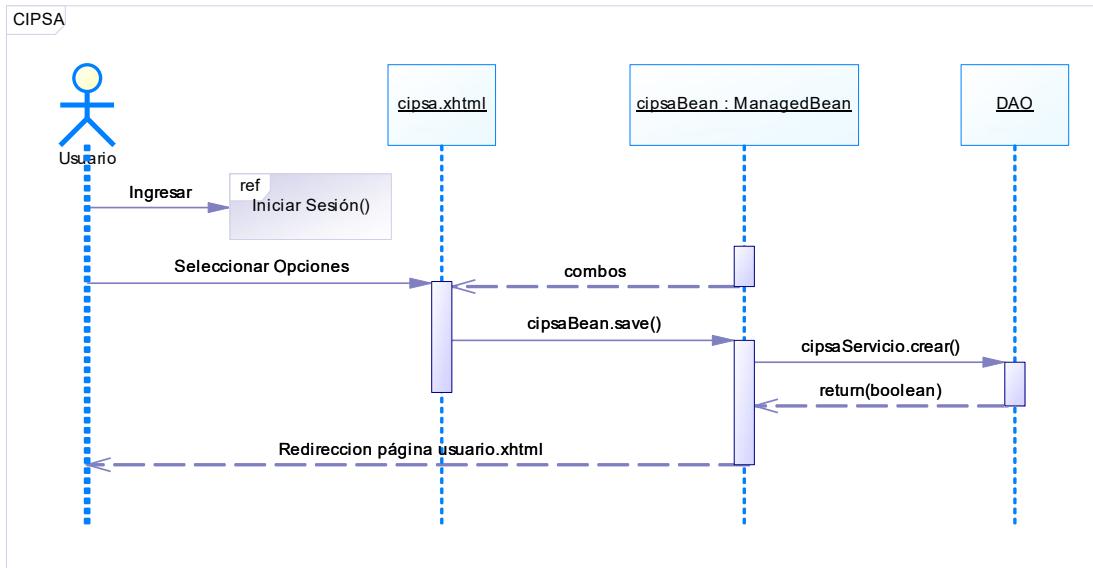
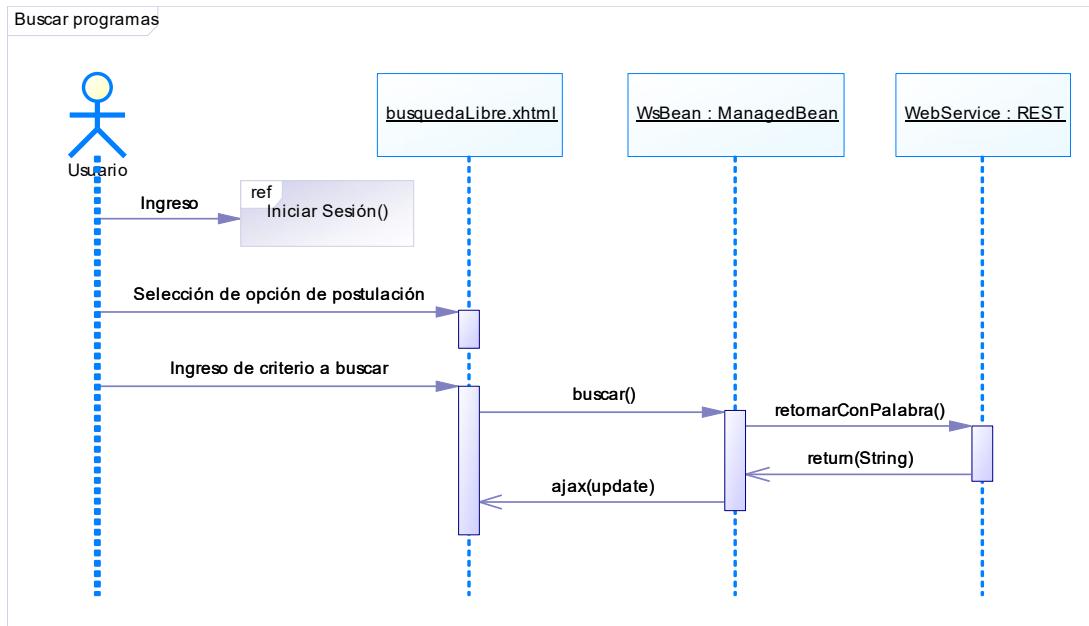


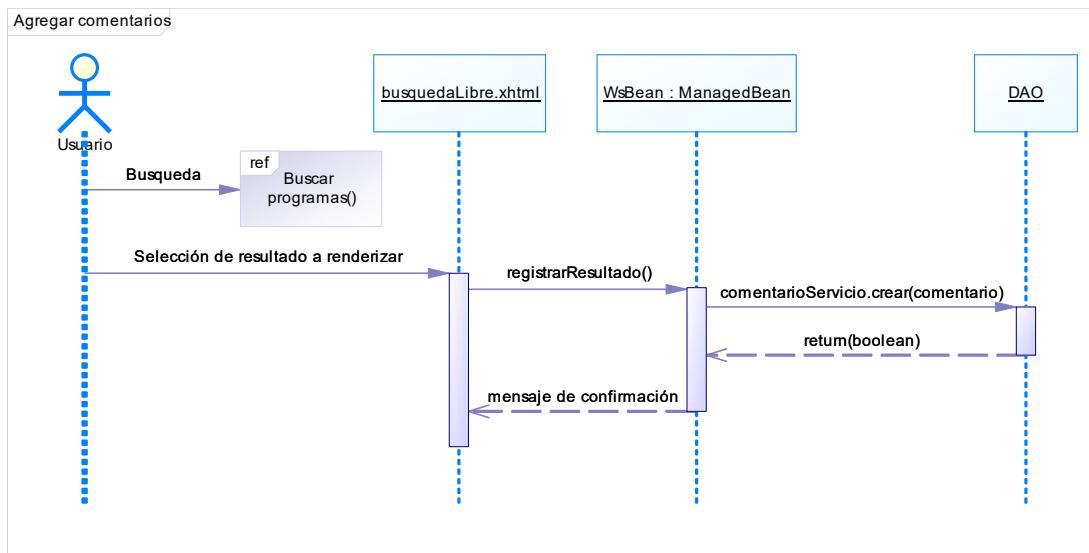
Figura 17 Diagrama de secuencia para desarrollar el test CIPSA

### 3.5.9.6 Buscar programas de maestrías



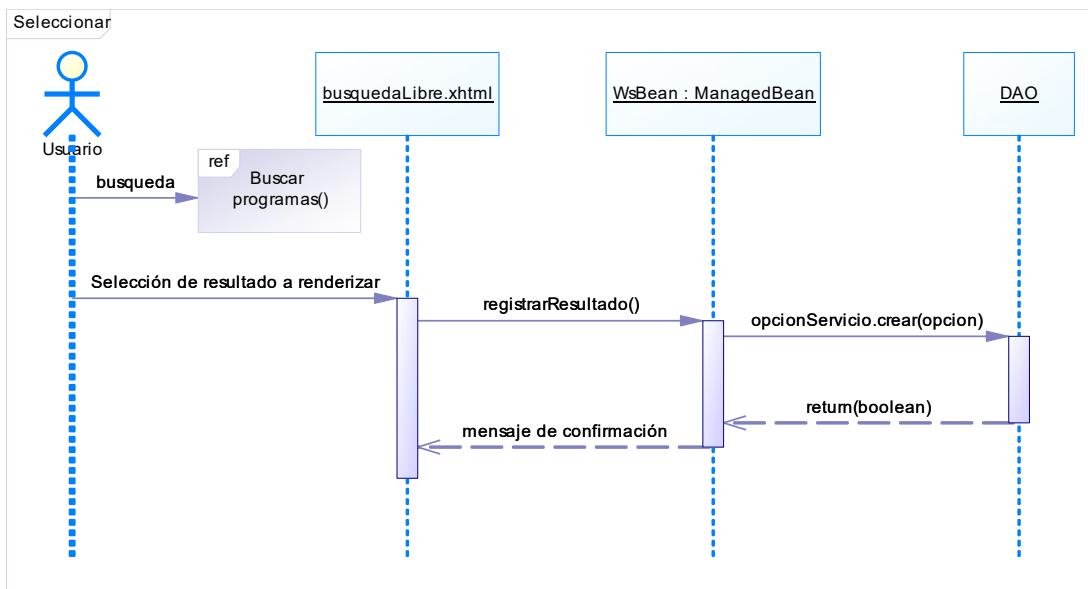
**Figura 18 Diagrama de secuencia para buscar programas**

### 3.5.9.7 Agregar comentarios



**Figura 19 Diagrama de secuencia para agregar comentarios**

### 3.5.9.8 Seleccionar



**Figura 20 Diagrama de secuencia para seleccionar el programa escogido**

## 3.6 ARQUITECTURA

Para el desarrollo del sistema AQUINO se ha usado una arquitectura de tipo MVC (Model View Control) partiendo desde una aplicación de tipo JEE (Java Enterprise Edition). Por las características de este tipo de aplicaciones se maneja usando 4 capas principales, las cuales son:

### 3.6.1 EAR

Los archivos EAR (Enterprise Archive) es un formato utilizado en la arquitectura JEE para desplegar de manera coherente y simultánea varios módulos en un servidor de aplicaciones. Contiene archivos XML denominados descriptores de despliegue que describen como desplegar los módulos contenidos en el paquete EAR.

### 3.6.2 EJB

Este módulo es la primera capa de programación donde se programará todo lo que involucra la lógica del negocio, ya sean servicios, clases de modelo, dao (Data Access Object).

Principalmente en este módulo se encuentra la capa de Modelo, dicha capa está compuesta por 3 subpaquetes o también denominados paquetes, los cuales son:

- **Modelo:** En este paquete se puede encontrar todas las clases que representan a cada uno de las entidades de la base de datos.
- **DAO:** Este paquete es el encargado de comunicar el sistema con la base de datos, permitiendo crear, recuperar, modificar y eliminar información de la base de datos.
- **Servicios:** Este paquete es la intermediadora entre el paquete DAO y la capa Controlador.

### **3.6.3 WEB**

Como su nombre lo especifica este módulo es donde se programará todo la parte web o en otras palabras lo que involucra a la parte visual que interactuará con el usuario.

Este módulo se divide en dos capas siendo la primera la parte visual de la aplicación o en otras palabras la Vista y la segunda corresponde a los Backing Beans los que controlan y permiten que funcionen adecuadamente la parte visual de la aplicación, también llamada Controlador.

## **3.7 PRUEBAS**

Como parte de todo ciclo de vida de software un sistema siempre tiene un etapa donde se realizan las llamadas pruebas, las cuales sirven para comprobar que el sistema funcione correctamente, dichas pruebas dependiendo del ámbito que se deseé comprobar pueden ser de caja blanca o de caja negra, adicionalmente para asegurar una correcta implementación de código se debe de seguir ciertas normas de buenas prácticas dentro de la programación de nuestro sistema, las cuales son:

### **3.7.1 ESTÁNDARES DE IMPLEMENTACIÓN**

Dentro de la codificación de un programa todos los desarrolladores escriben su código de diferentes formas, algunos de forma menos comprensible que otros, es por

esta razón que se ha decidido seguir una de las guías de buenas prácticas, las cuales ayudan dando consejos para estructurar y codificar de forma comprensible y correcta

### 3.7.1.1 ESTRUCTURA DE LOS ARCHIVOS

Entre las diferentes recomendaciones de buenas prácticas dentro de la programación se tiene la forma de la estructura de los cada uno de los archivos, en este caso serían de los Backing Bean, dicha estructura se muestra en la figura 21, donde se recomienda que como parte inicial del archivo se declare el nombre de los paquetes a utilizar, declaración de imports, e implementación de clases y librerías a utilizar.

```

2  * To change this license header, choose License Headers in
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package ec.edu.programasbecas.bean;
7
8 import ec.edu.programasbecas.modelo.Comentario;
9 import ec.edu.programasbecas.modelo.Opciones;
10 import ec.edu.programasbecas.modelo.Usuario;
11 import ec.edu.programasbecas.servicios.ComentarioServicio;
12 import ec.edu.programasbecas.servicios.OpcionesServicio;
13 import ec.edu.programasbecas.util.Util;
14 import ec.edu.programasbecas.wS.AquinoLucene;
15 import java.io.Serializable;
16 import java.sql.Date;
17 import java.util.ArrayList;
18 import java.util.Calendar;
19 import java.util.HashMap;
20 import java.util.Map;
21 import javax.annotation.PostConstruct;
22 import javax.ejb.EJB;
23 import javax.faces.application.FacesMessage;
24 import javax.faces.bean.ManagedBean;
25 import javax.faces.bean.ViewScoped;
26 import javax.faces.context.FacesContext;
27
28 /**
29 *
30 * @author Ramt

```

**Figura 21 Estructura de los Backing Bean**

### 3.7.1.2 DECLARACIÓN DE VARIABLES

Dentro de la guía de buenas prácticas se recomienda que para declarar variables, estas deben declararse una sola por línea, además de instanciarlas de ser necesario y que sus nombres sean descriptivos para lo que fueron creadas, esto se puede apreciar en la figura 22.

```

    * @author Ramt
    */
@ManagedBean
@ViewScoped
public class RegistroBean implements Serializable {

    @EJB
    private UsuarioServicio usuarioServicio;
    @EJB
    private PaisServicio paisServicio;

    @Resource(name = "mail/aquino")
    private Session mailSession;

    private Usuario usuario;
    private List<Pais> pais;
    private Date nacimiento;

    @PostConstruct
    public void init() {
        usuario = new Usuario();
    }
}

```

**Figura 22 Declaración de Variables**

### 3.7.1.3 DECLARACIÓN DE MÉTODOS

Las funciones o métodos deben de estar declaradas con nombres descriptivos, los cuales comienzan con letra minúscula, si son nombres compuesto se deberá usar una letra mayúscula para el comienzo de la siguiente palabra del nombre además de ir unida a la anterior palabra.

### 3.7.2 PRUEBAS DE CAJA NEGRA

Las pruebas de caja negra fueron diseñadas para comprobar el funcionamiento externo del sistema más no como se realiza el proceso internamente, es por esa razón que al ser una aplicación de tipo web donde el usuario únicamente interactúa con la interfaz gráfica, este tipo de pruebas es la que más conviene usar.

Las pruebas de caja negra que a continuación se van a realizar son el resultado de la evaluación de cada uno de los requerimientos.

**Tabla 19:**  
**Prueba de Caja Negra 1 Registrar usuario**

<b>Código de caso de prueba</b>	<b>PCN01 Registrar usuarios</b>
<b>Código de requerimiento</b>	RE01 Registrar usuarios
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto proceso de registro de un usuario al sistema AQUINO.
<b>Condiciones de Ejecución</b>	Un nuevo usuario sin cuenta debe entrar al sistema AQUINO
<b>Entrada/Pasos de Ejecución</b>	Ingreso al sistema. Clic en el link “Registro” Llenar la información solicitada Hacer clic en aceptar
<b>Resultado Esperado</b>	Se desplegará una nueva página web donde se le dará la bienvenida al sistema, adicionalmente se le enviará un correo electrónico al mail que el usuario indicó con su nueva contraseña
<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.

**Tabla 20**  
**Prueba de Caja Negra 2 Iniciar sesión**

<b>Código de caso de prueba</b>	<b>PCN02 Iniciar sesión</b>
<b>Código de requerimiento</b>	RE02 Iniciar sesión
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto proceso para el inicio de sesión del usuario.
<b>Condiciones de Ejecución</b>	Un usuario que ya se registró anteriormente dentro del sistema intenta iniciar sesión
<b>Entrada/Pasos de Ejecución</b>	Ingreso al sistema. Llenar el campo donde se le solicita su cédula Llenar el campo donde se le solicita su contraseña Hacer clic en “Iniciar sesión”

Continua →

<b>Resultado Esperado</b>	Si los datos ingresados son correctos se le desplegará un cartel informativo en la esquina superior derecha dándole la bienvenida seguido con el nombre del usuario, caso contrario el mensaje indicara que la contraseña es incorrecta o que no se encuentra registrado el número de cedula ingresado
<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.

**Tabla 21**  
**Prueba de Caja Negra 3 Cambiar clave**

<b>Código de caso de prueba</b>	<b>PCN03 Cambiar clave</b>
<b>Código de requerimiento</b>	RE03 Registrar usuarios
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto proceso para el cambio de clave.
<b>Condiciones de Ejecución</b>	El usuario debe iniciar sesión
<b>Entrada/Pasos de Ejecución</b>	Ingreso al sistema. Iniciar sesión Si es la primera vez que inicia sesión se le solicitará que cambie la contraseña, caso contrario deberá ingresar al link que tiene su nombre y hacer clic en el botón “Cambiar contraseña” Llenar la información solicitada Hacer clic en aceptar
<b>Resultado Esperado</b>	Si los datos ingresados son correctos se le desplegará un cartel informativo en la esquina superior derecha informándole sobre el correcto cambio de la contraseña, caso contrario se le desplegará información para solucionar el error
<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.

**Tabla 22**  
**Prueba de Caja Negra 4 Registrar Título**

<b>Código de caso de prueba</b>	<b>PCN04 Registrar título</b>
<b>Código de requerimiento</b>	RE04 Registrar título
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto proceso del registro del título de tercer nivel del usuario
<b>Condiciones de Ejecución</b>	El usuario debe iniciar sesión
<b>Entrada/Pasos de Ejecución</b>	<p>Ingreso al sistema.</p> <p>Al ser la primera vez que el usuario inicia sesión se le solicitará después de cambiar su contraseña registrar su información del título de tercer nivel</p> <p>Llenar la información solicitada</p> <p>Hacer clic en aceptar</p>
<b>Resultado Esperado</b>	Si los datos ingresados son correctos se le desplegará un cartel informativo en la esquina superior derecha informándole sobre el correcto registro del título, caso contrario se le desplegará información para solucionar el error
<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.

**Tabla 23**  
**Prueba de Caja Negra 5 Desarrollar el test de intereses vocacionales CIPSA**

<b>Código de caso de prueba</b>	<b>PCN05 Desarrollar el test de intereses vocacionales CIPSA</b>
<b>Código de requerimiento</b>	RE05 Desarrollar el test de intereses vocacionales CIPSA
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto proceso del Desarrollo del test de intereses vocacionales CIPSA

Continua →

<b>Condiciones de Ejecución</b>	El usuario deberá iniciar sesión
<b>Entrada/Pasos de Ejecución</b>	Ingreso al sistema. Clic en el link “CIPSA” Llenar la información solicitada Hacer clic en aceptar
<b>Resultado Esperado</b>	Si los datos ingresados son correctos se le desplegará un cartel informativo en la esquina superior derecha informándole sobre el correcto registro de datos, caso contrario se le desplegará información para solucionar el error
<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.

**Tabla 24****Prueba de Caja Negra 6 Buscar programas de maestrías**

<b>Código de caso de prueba</b>	PCN06 Buscar programas de maestrías
<b>Código de requerimiento</b>	RE06 Buscar programas de maestrías
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto proceso para la búsqueda de programas de maestrías
<b>Condiciones de Ejecución</b>	El usuario debe de iniciar sesión
<b>Entrada/Pasos de Ejecución</b>	Ingreso al sistema. Clic en el menú “Postulación” Clic en la opción “Búsqueda Libre” Llenar el campo con una frase o palabra relevante sobre el programa de postgrado a buscar Seleccionar si desea una búsqueda avanzada o normal. Hacer clic en aceptar
<b>Resultado Esperado</b>	Si los datos ingresados son correctos se le desplegará información sobre el criterio buscado

Continua →

<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.
--------------------------------	---

**Tabla 25****Prueba de Caja Negra 7 Agregar Comentarios**

<b>Código de caso de prueba</b>	PCN07 Agregar Comentarios
<b>Código de requerimiento</b>	RE07 Agregar Comentarios
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto proceso para agregar comentarios dentro de las opciones de postgrados a seguir
<b>Condiciones de Ejecución</b>	El usuario con una sesión iniciada debe realizar la búsqueda de un programa de postgrado
<b>Entrada/Pasos de Ejecución</b>	Ingreso al sistema. Buscar un programa de postgrado según un criterio relevante Seleccionar el resultado deseado Ingresar el contenido del comentario que desea guardar sobre el programa de postgrado Hacer clic en guardar.
<b>Resultado Esperado</b>	Si los datos ingresados son correctos se le desplegará un cartel informativo en la esquina superior derecha informándole que el comentario ha sido guardado correctamente, caso contrario se le desplegará información para solucionar el error
<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.

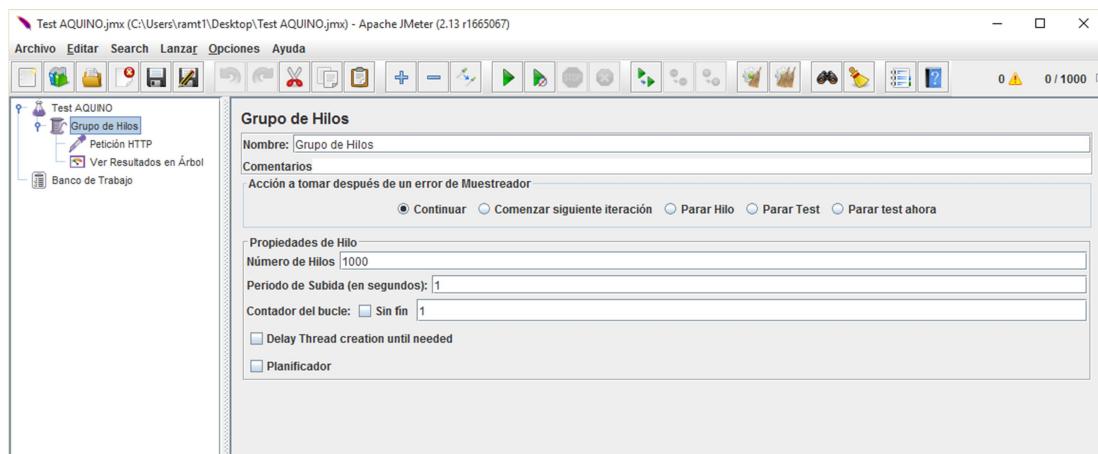
**Tabla 26****Prueba de Caja Negra 8 Seleccionar**

<b>Código de caso de prueba</b>	PCN07 Seleccionar
<b>Código de requerimiento</b>	RE07 Seleccionar maestría escogida
<b>Descripción de la prueba</b>	Se pretende realizar una evaluación del correcto

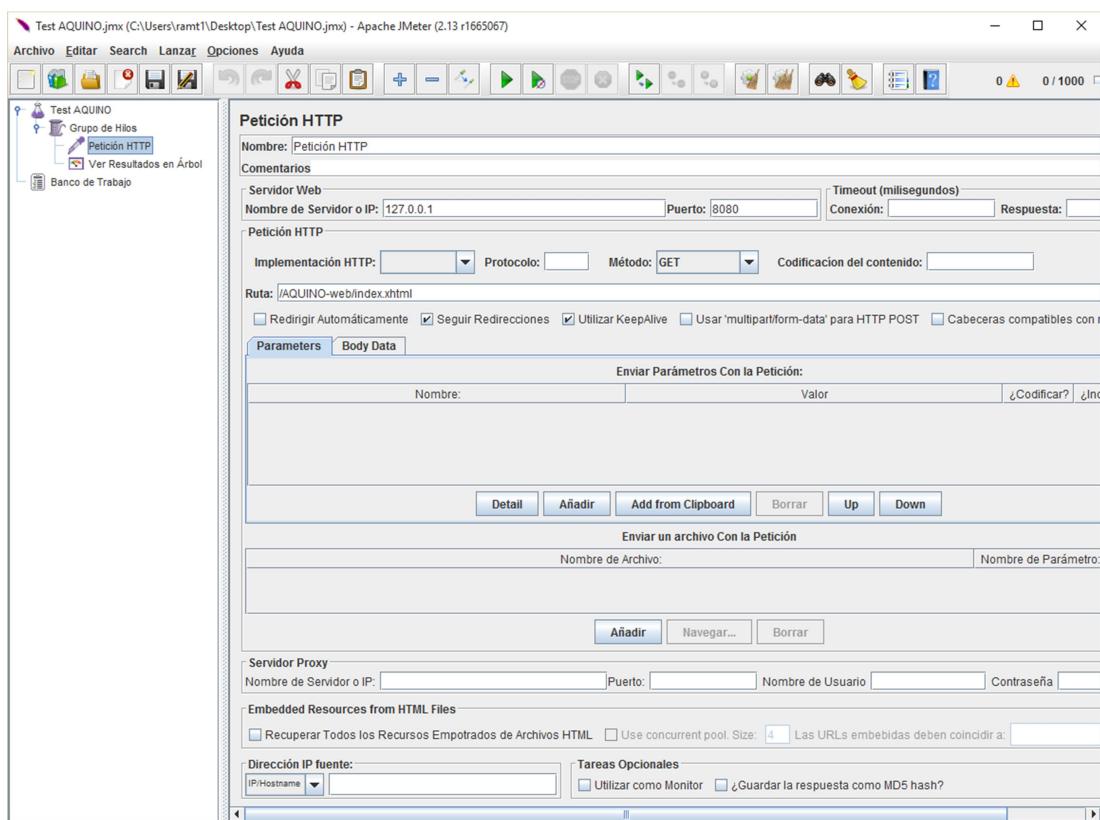
	proceso para seleccionar una de las opciones de postgrado como el programa escogido para la beca
<b>Condiciones de Ejecución</b>	El usuario con una sesión iniciada debe realizar la búsqueda de un programa de postgrado
<b>Entrada/Pasos de Ejecución</b>	Ingreso al sistema. Buscar un programa de postgrado según un criterio relevante Seleccionar el resultado deseado Seleccionar la opción “escogido” Hacer clic en guardar información.
<b>Resultado Esperado</b>	Si los datos ingresados son correctos se le desplegará un cartel informativo en la esquina superior derecha informándole que se ha guardado correctamente, caso contrario se le desplegará información para solucionar el error
<b>Evaluación de la prueba</b>	Los resultados esperados han sido los correctos según los requerimientos del usuario.

### 3.7.3 PRUEBAS DE ESTRÉS

Una de las características fundamentales de una aplicación web es poder soportar toda la carga de usuarios conectados en un mismo momento, es por esta razón que una de las principales pruebas que se debe de realizar a una aplicación web son las pruebas de estrés, dicha prueba fue realizada con la herramienta Apache Jmeter, donde la herramienta simula la conexión de n números de usuarios conectados al mismo tiempo realizando una petición. Para la prueba se ha implementado un pool de conexiones de máximo 400 accesos, pero facilitando la apertura de ser necesario a más. Como se puede apreciar en las figuras 23 y 24, se encuentra la configuración para la prueba la cual se ha realizado para recibir mil peticiones al mismo tiempo.



**Figura 23 Configuración de cantidad de peticiones en Jmeter para prueba de estrés**



**Figura 24 Configuración del host en Jmeter para la prueba de estrés**

Los resultados de dicha prueba se los puede observar en la figura 25, mostrando que las mil peticiones se las han realizado correctamente.

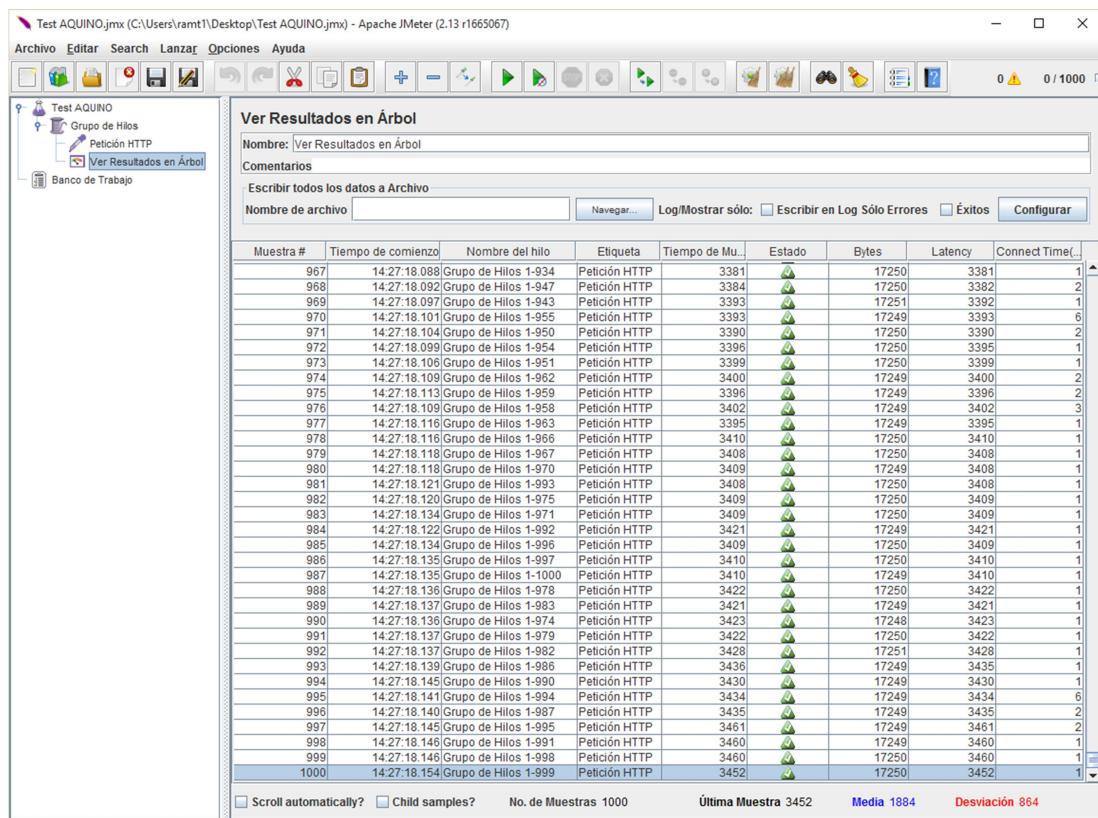


Figura 25 Resultado de la prueba de estrés en Jmeter

## CAPÍTULO 4

# CONCLUSIONES Y RECOMENDACIONES

### 4.1 CONCLUSIONES

- El desarrollo del sistema AQUINO ha logrado satisfacer los requerimientos solicitados por la SENESCYT, los cuales ayudan en automatización del proceso de postulación a becas.
- Con el sistema AQUINO el proceso de postulación a becas redujo su tiempo a minutos, ya que este proceso antiguamente era desarrollado a mano el postulante se podía demorar días para poder escoger un programa de postgrado adecuado para el postulante.
- La etapa de especificación de requerimientos fue una de las etapas más complicadas ya que como en la mayoría de los desarrollos de software existen peticiones que son difíciles de realizar o que a su vez pueden ser implementadas como un nuevo sistema ya que están fuera del ámbito del mismo.
- El haber seguido la metodología UWE y a su vez la guía de buenas prácticas para el desarrollo de sistemas, permitió que la codificación del sistema se realice con el enfoque correcto para una aplicación web fiable y que a su vez cumpla con cada una de las fases del ciclo de vida del software.

### 4.2 RECOMENDACIONES

- A pesar de que el sistema AQUINO es intuitivo y fácil de usar para los usuarios e recomienda que exista una persona encargada en la SENESCYT que pueda dar ayuda a los nuevos postulantes que no logren comprender el funcionamiento del sistema.
- Se recomienda que tanto el sistema AQUINO como el módulo Lucene-AQUINO se encuentren instalados y funcionando en el mismo servidor de aplicaciones, esto facilitará el acceso a la información sin tener riesgos de interrupción de la comunicación por la no disponibilidad de alguno de los dos o a su vez por un simple cambio de IP.

- Se recomienda realizar backups de la información de la base de datos del sistema AQUINO, ya que en el peor de los casos al existir alguna avería del servidor donde se va a encontrar alojado el sistema se puede levantar de nuevo la información con el sistema sin problema alguno de perdida de información.
- Se recomienda que el proceso batch del módulo Lucene-AQUINO se ejecute cada tres meses, de esta forma podemos asegurar que la información recolectada para desplegar es la más actual.

## REFERENCIAS BIBLIOGRÁFICAS

- Colección de Tesis Digitales.* (s.f.). Recuperado el 18 de 06 de 2014, de Universidad de las Américas Puebla:  
[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/viveros\\_s\\_ca/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros_s_ca/capitulo3.pdf)
- Addictos al trabajo.* (13 de 03 de 2006). Recuperado el 18 de 06 de 2014, de  
<http://www.addictosaltrabajo.com/tutoriales/tutoriales.php?pagina=convValidJSF>
- Athula, G., & San, M. (2001). *Ingeniería Web: Una introducción*. IEEE.
- Brandon, D. (2007). *Ingeniería de software para aplicaciones Web modernas*.
- Buenas Tareas.* (01 de 05 de 2012). Recuperado el 19 de 06 de 2014, de  
<http://www.buenastareas.com/ensayos/Tecnologia-Jsf/4304445.html>
- Definición de.* (17 de Junio de 2015). Obtenido de <http://definicion.de/orientacion-vocacional/>
- Dolors, C. (28 de 06 de 2015). *Introducción al diseño de bases de datos*. Obtenido de  
[http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02150.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02150.pdf)
- El blog de Ana Cortés.* (25 de 09 de 2011). Recuperado el 18 de 06 de 2014, de  
[http://puesenmiordenadorfunciona.blogspot.com/2011/10/jsf-20-managed-beans-iii\\_25.html](http://puesenmiordenadorfunciona.blogspot.com/2011/10/jsf-20-managed-beans-iii_25.html)
- Fuseau García, A. S., & Silva Enríquez, E. F. (21 de 06 de 2015). *Repositorio Digital de la ESPE.* Obtenido de  
<http://repositorio.espe.edu.ec/bitstream/21000/362/1/T-ESPE-029496.pdf>
- Guerrero, H., & Vásquez, J. (28 de 06 de 2015). *Monografías.com*. Obtenido de  
[http://www.monografias.com/trabajos84/modelamiento-base-datos.shtml](http://www.monografias.com/trabajos84/modelamiento-base-datos/modelamiento-base-datos.shtml)

Israel, A. (12 de Septiembre de 2013). *Academia.edu*. Obtenido de <https://www.academia.edu/9456729/Mvc>

*nDeveloper*. (s.f.). Recuperado el 19 de 06 de 2014, de [http://www.ndeveloper.com/nDeveloperDocuments/documents/nDeveloper\\_JavaServerFaces.pdf](http://www.ndeveloper.com/nDeveloperDocuments/documents/nDeveloper_JavaServerFaces.pdf)

*No todo código*. (04 de 07 de 2013). Recuperado el 19 de 06 de 2014, de <http://www.notodocodigo.com/introduccion-a-jsf/facelets/>

*PrimeFaces*. (01 de 07 de 2015). Obtenido de <http://www.primefaces.org/showcase/>

Quiroga, A. (23 de Marzo de 2015). *Proyecto de Grado Ingenieria Sistemas*. Obtenido de <http://proyectogradoingenieriasistemas.blogspot.com/2015/03/metodologia-uwe-uml-uml-based-web.html>

Sánchez, J. (28 de 06 de 2015). *Diseño Conceptual de Bases de Datos*. Obtenido de <http://www.jorgesanchez.net/bd/disenoBD.pdf>

Silberschatz, A., Korth, H., & Sudarshan, S. (2002). INTRODUCCIÓN. En *FUNDAMENTOS DE BASES DE DATOS* (pág. 1). Madrid: McGRAW-HILL. Obtenido de <https://unefazuliasistemas.files.wordpress.com/2011/04/fundamentos-de-bases-de-datos-silberschatz-korth-sudarshan.pdf>

*Sinnexus*. (28 de 06 de 2015). Obtenido de [http://www.sinnexus.com/business\\_intelligence/datamining.aspx](http://www.sinnexus.com/business_intelligence/datamining.aspx)

*Un aporte a Java*. (19 de 05 de 2009). Recuperado el 19 de 06 de 2014, de <http://ungranoparajava.blogspot.com/2009/05/distintas-implementaciones-de-jsf.html>

*UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH*. (28 de 06 de 2015). Obtenido de Normalizacion de Bases de Datos y Técnicas de diseño: <http://www.cs.upc.edu/~bcasas/docencia/pfc/NormalitzacioBD.pdf>

*UWE.* (17 de Abril de 2015). Obtenido de <http://uwe.pst.ifi.lmu.de/aboutUwe.html>

*Wikipedia.* (14 de 12 de 2013). Recuperado el 18 de 06 de 2014, de [http://es.wikipedia.org/wiki/Expression\\_Language](http://es.wikipedia.org/wiki/Expression_Language)

*Wikipedia.* (22 de 01 de 2014). Recuperado el 18 de 06 de 2014, de [http://es.wikipedia.org/wiki/JavaServer\\_Faces](http://es.wikipedia.org/wiki/JavaServer_Faces)

*Wikipedia.* (19 de Junio de 2015). Obtenido de [https://es.wikipedia.org/wiki/Metodolog%C3%A3a\\_de\\_desarrollo\\_de\\_software](https://es.wikipedia.org/wiki/Metodolog%C3%A3a_de_desarrollo_de_software)

*Wikipedia.* (04 de Abril de 2015). Obtenido de [http://es.wikipedia.org/wiki/Miner%C3%A3a\\_de\\_datos](http://es.wikipedia.org/wiki/Miner%C3%A3a_de_datos)

*Wikipedia.* (09 de Julio de 2015). Obtenido de <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>