



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

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO DE SISTEMAS E INFORMÁTICA**

**AUTOR: ARÉVALO PACHALA KATHERINE ALEXANDRA**

**MATANGO IPIALES RENÉ FABRICIO**

**TEMA: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE**

**GESTIÓN DE CALIDAD PARA LOS PROGRAMAS DE POSGRADOS DE LA**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, MEDIANTE LA**

**UTILIZACIÓN DE LA PLATAFORMA JAVA ENTERPRISE EDITION JEE6 WEB**

**APLICANDO LA METODOLOGÍA UWE**

**DIRECTOR: ING. FUERTES, WALTER. PhD**

**CODIRECTOR: ING. RON, MARIO. MSc.**

**SANGOLQUÍ, ABRIL 2014**

**CERTIFICADO**

Ing. Walter Fuertes. PhD

Ing. Mario Ron. MSc

**CERTIFICAN**

Que el trabajo titulado “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE GESTIÓN DE CALIDAD PARA LOS PROGRAMAS DE POSGRADOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA JAVA ENTERPRISE EDITION JEE6 WEB APLICANDO LA METODOLOGÍA UWE” realizado por la Srta. ARÉVALO PACHALA KATHERINE ALEXANDRA y el Sr. MATANGO IPIALES RENÉ FABRICIO, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas “ESPE”.

Sangolquí, abril del 2014

---

Ing. Walter Fuertes PhD

DIRECTOR

---

Ing. Mario Ron MSc

CODIRECTOR

## DECLARACIÓN DE RESPONSABILIDAD

ARÉVALO PACHALA KATHERINE ALEXANDRA  
MATANGO IPIALES RENÉ FABRICIO

DECLARO QUE:

El proyecto de grado denominado “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE GESTIÓN DE CALIDAD PARA LOS PROGRAMAS DE POSGRADOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA JAVA ENTERPRISE EDITION JEE6 WEB APLICANDO LA METODOLOGÍA UWE”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las fuentes que se incorporan en la bibliografía.

Consecuentemente este trabajo es nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, abril del 2014

---

Arévalo Pachala Katherine Alexandra

---

Matango Ipiales René Fabricio

## **AUTORIZACIÓN**

Nosotros, ARÉVALO PACHALA KATHERINE ALEXANDRA  
MATANGO IPIALES RENÉ FABRICIO

Autorizamos a la UNIVERSIDAD DE LA FUERZAS ARMADAS “ESPE”, la publicación, en la biblioteca virtual de la Institución del trabajo “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE GESTIÓN DE CALIDAD PARA LOS PROGRAMAS DE POSGRADOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA JAVA ENTERPRISE EDITION JEE6 WEB APLICANDO LA METODOLOGÍA UWE”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, abril del 2014

---

Arévalo Pachala Katherine Alexandra

---

Matango Ipiales René Fabricio

## DEDICATORIA

*Primero a Dios, por darme la oportunidad de vivir, por sus bendiciones, por proteger y guiar mi camino. A mis padres, Liborio(+) y Aurora por todo lo que han hecho por mí, por estar conmigo en las buenas y malas con sus consejos, brindándome su amor. A mi padre, que a pesar de no contar con su presencia siempre lo llevaré en mi corazón y sé que desde el cielo nos cuida. A mi madre, que me apoyo en todo momento, y a pesar de lo difícil de la vida me ha enseñado a salir adelante, usted es la mejor. A mi hermana, Jajaira por su apoyo y cariño incondicional, por sus sonrisas inesperadas que me han alegrado durante este tiempo. Gracias por todo.*

*Ustedes son mi inspiración y mi fuerza para seguir adelante cumpliendo todas mis metas propuestas. Dios y Papi me las bendiga y proteja siempre, Mami Chula y Jaji las amo.*

**Katherine Arévalo**

*Gracias a Dios por iluminarme y guiarme por los senderos correctos, a mis padres por el infinito apoyo, que me brindaron para que hoy pueda alcanzar este sueño tan anhelado, a mis hermanos Christian y Daysi a mi abuelita Laura, que siempre me apoyaron con sus consejos y me han incentivado con su ejemplo; a mi hija Emily que es la razón de superación, a ustedes por siempre mi más sinceros agradecimientos.*

**René Matango**

## **AGRADECIMIENTOS**

Agradecemos a todas las personas que hicieron posible la culminación de este trabajo. Primordialmente a nuestro Director y Co Director de tesis; a los Ingenieros (as) que nos ayudaron en el proceso de pruebas del sistema. Gracias por su ayuda, apoyo, comprensión y orientación los cuales nos brindaron durante el desarrollo de este trabajo.

**TABLA DE CONTENIDO**

CERTIFICADO.....	i
DECLARACIÓN DE RESPONSABILIDAD .....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTOS.....	v
ÍNDICE DE TABLAS .....	xi
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE ANEXOS.....	xiv
RESUMEN.....	xv
ABSTRACT.....	xvi
GLOSARIO DE NOMENCLATURAS.....	xvii
CAPÍTULO 1 .....	1
1.1 Antecedentes .....	2
1.2 Justificación e Importancia.....	2
1.3 Objetivos .....	4
1.3.1 Objetivo General .....	4
1.3.2 Objetivos Específicos.....	4
1.4 Alcance .....	4
1.5 Factibilidad.....	7
1.5.1 Factibilidad Técnica .....	7
1.5.2 Factibilidad Tecnológica.....	7
1.5.3 Factibilidad Operativa .....	8

1.5.4	Factibilidad Operacional.....	8
1.5.5	Factibilidad Económica .....	8
1.5.6	Factibilidad Legal .....	9
1.6	Presupuesto .....	9
CAPÍTULO 2.....		10
MARCO TEÓRICO DE REFERENCIA .....		10
2.1	Introducción .....	10
2.2	Aplicaciones Web.....	10
2.3	Características .....	12
2.4	Herramientas y Bases de datos .....	12
2.4.1	StarUML.....	12
2.5	Herramientas de Desarrollo .....	14
2.5.1	Lenguaje de Programación Java .....	14
2.5.2	IDE de desarrollo Netbeans .....	14
2.5.3	Java Server Faces (JSF) .....	16
i.	Capa Web: .....	17
ii.	Capa de Negocio.....	17
iii.	Capa de Datos.....	17
2.5.4	Base de Datos MySQL.....	19
2.5.5	Servidor de Aplicaciones Glassfish.....	20
2.6	Metodología UWE .....	21
2.6.1	Definición .....	21
2.6.2	Características .....	21
2.6.3	Fases de la Metodología UWE .....	22



2.6.4	Análisis de Requisitos .....	22
2.6.5	Modelo Conceptual .....	23
2.6.6	Modelo Navegacional.....	23
2.6.7	Modelo de Presentación .....	23
2.7	Artefactos .....	24
2.7.1	Diagramas de Casos de Uso .....	24
2.7.2	Diagrama de Clases .....	25
2.7.3	Diagrama de Secuencia.....	25
2.7.4	Diagrama de Estado .....	26
2.7.5	Diagrama de Despliegue.....	26
2.7.6	Diagrama de Actividades .....	26
2.8	Lenguaje de Modelamiento Unificado (UML) .....	27
2.9	Ingeniería de Requerimientos (Estándar IEEE 830) .....	28
i.	Identificación de Requisitos:.....	28
ii.	Análisis de Requisitos y Negociación: .....	28
iii.	Especificación de requisitos: .....	29
iv.	Modelado del Sistema: .....	29
v.	Validación de Requisitos y gestión de Requisitos: .....	29
2.10	Pruebas de software .....	29
CAPÍTULO 3.....		31
ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....		31
3.1	Introducción .....	31
3.2	Propósito.....	31
3.3	Alcance .....	31

3.4	Visión Global.....	32
3.5	Descripción .....	32
3.5.1	Perspectiva del producto.....	32
3.5.2	Funciones del producto.....	34
3.5.3	Tipos y Funciones del Usuario.....	34
3.5.4	Restricciones generales.....	35
3.5.5	Interfaces de software.....	35
3.5.6	Interfaces de comunicación .....	35
3.5.7	Atributos.....	35
3.6	Metodología UWE .....	36
3.6.1	Modelo Conceptual .....	36
3.6.2	Modelo Físico.....	39
3.6.3	Modelo de Presentación .....	41
3.6.4	Modelo de Paquetes .....	41
3.6.5	Casos de Uso del Paquete Parametrización Datos .....	42
3.6.6	Casos de Uso del Paquete Gestión Evidencias.....	43
3.6.7	Casos de Uso del Paquete Usuarios .....	44
3.6.8	Diagrama de Clases .....	45
3.6.9	Diagrama de Secuencia.....	47
3.6.10	Diagrama de Estados .....	48
3.6.11	Diagrama de Despliegue.....	49
3.6.12	Diagrama de Actividades .....	50
	CAPÍTULO 4.....	52
	PRUEBAS.....	52

4	Introducción .....	52
4.1	Estándares de Implementación.....	52
4.1.1	Declaración de Variables .....	53
4.1.2	Sentencias .....	53
4.2	Pruebas de Aceptación .....	54
4.2.1	Pruebas de Caja Negra.....	54
4.2.2	Pruebas de Caja Blanca .....	56
4.2.3	Pruebas de Stress .....	58
4.3	Manual de Referencias Técnicas .....	59
4.3.1	Prerrequisitos.....	59
i.	Requisitos de Instalación Servidor .....	59
ii.	Características Técnicas Servidor .....	59
iii.	Características Técnicas Cliente .....	59
4.3.2	Configuración de MySql.....	60
4.3.3	Configuración de Glassfish .....	60
	CAPÍTULO 5 .....	62
	CONCLUSIONES Y RECOMENDACIONES .....	62
5.1	Conclusiones .....	62
5.2	Recomendaciones .....	64
	BIBLIOGRAFÍA Y WEBGRAFÍA .....	65

## ÍNDICE DE TABLAS

Tabla 1: Presupuesto aproximado para la implementación del sistema SIIAC-ESPE.....	9
Tabla 2: Lenguajes de Programación utilizados en Netbeans IDE 6.1 .....	16
Tabla 3: Notación UWE para casos de uso. ....	24
Tabla 4: Notación UWE para elementos del diagrama de actividad. ....	27
Tabla 5: Descripción de las Tablas del Sistema .....	36
Tabla 6: Formulario de revisión de funcionalidad del caso de uso Administrar Indicador - Ingreso.....	54
Tabla 7: Formulario de revisión de funcionalidad del caso de uso Administrar Indicador - Actualizar. ....	55
Tabla 8: Formulario de revisión de funcionalidad del caso de uso Administrar Indicador - Eliminar.....	56
Tabla 9: Prueba de Caja Blanca - Buscar Carrera.....	57
Tabla 10: Prueba de Caja Blanca - Eliminar Criterio.....	57

## ÍNDICE DE FIGURAS

Ilustración 1: Interfaz (CGI) tecnología World Wide Web.....	11
Ilustración 2: Arquitectura Multicapas con la capa de presentación JSF	
Fuente: (Mentoring, 2012). .....	18
Ilustración 3: Arquitectura de Implementación Sistema SIIAC-ESPE.	
Fuente: (Almeida, 2013). .....	18
Ilustración 4: Dimensiones Modelamiento UWE. ....	22
Ilustración 5: ISO/IEC 29119 Software Testing Structure. ....	30
Ilustración 6: Modelo Árbol de Acreditación Institucional. ....	33
Ilustración 7: Modelo Árbol de Acreditación Carreras. ....	33
Ilustración 8: Modelo Conceptual del Sistema SIIAC-ESPE. ....	38
Ilustración 9: Modelo Físico del Sistema SIIAC-ESPE.....	40
Ilustración 10: Modelo de Presentación del Sistema SIIAC-ESPE. ....	41
Ilustración 11: Paquetes del Sistema SIIAC-ESPE. ....	42
Ilustración 12: Casos de Uso Parametrización Datos .....	43
Ilustración 13: Casos de Uso Paquete Gestión Evidencias. ....	44
Ilustración 14: Caso de Uso Paquete Usuarios.....	45
Ilustración 15: Diagrama de Clases del Sistema SIIAC-ESPE. ....	46
Ilustración 16: Diagrama de Secuencia - Cargar Evidencia del Indicador. ....	48
Ilustración 17: Diagrama de Estados - Ingreso al Sistema. ....	49
Ilustración 18: Diagrama de Despliegue del Sistema.....	50
Ilustración 19: Diagrama de Actividades - Registrar Carrera. ....	51
Ilustración 20: Estructura del archivo BEAN. ....	52

Ilustración 21: Declaración de variables utilizadas.....	53
Ilustración 22: Sentencias implementadas en el desarrollo. ....	54
Ilustración 23: Programa Jmeter utilizado para prueba de Stress.....	58
Ilustración 24: Configuración del Servidor de Aplicaciones Glassfish.....	58
Ilustración 25: Configuración Acceso Base de Datos MySql. ....	60
Ilustración 26: Archivo Fuente de la Aplicación SIIAC-ESPE. ....	61
Ilustración 27: Carga Archivo /prj_espeAcreditacion .War. ....	61

## ÍNDICE DE ANEXOS

**ANEXO A:** Casos de Uso.

**ANEXO B:** Manual de Usuario del Sistema SIIAC-ESPE.

## RESUMEN

Este proyecto describe la construcción de un sistema denominado SIIAC-ESPE “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE GESTIÓN DE CALIDAD PARA LOS PROGRAMAS DE POSGRADOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA JAVA ENTERPRISE EDITION JEE6 WEB APLICANDO LA METODOLOGÍA UWE”. El sistema gestionará el proceso de Acreditación y Evaluación de las carreras y programas de posgrado, de acuerdo al tipo de acreditación como son carreras, institucional y extensiones. Permitirá la carga de evidencias por cada uno de los indicadores según el criterio de evaluación asignados al coordinador y responsable. Este documento describe el desarrollo de un sistema Web transaccional y aplicación distribuida mediante la metodología UWE que se basa en la elaboración de los modelos de casos de uso, conceptual, y presentación. Los resultados obtenidos para los usuarios finales son los solicitados en el levantamiento de requerimientos tanto para el Coordinador, Evaluador y Responsable; además se realizó pruebas directas con los involucrados en la acreditación y esto ayudó con el afinamiento de la aplicación, se realizó el acta de entrega de recepción del producto solicitado quedando ambas partes satisfechos.

**Palabras Clave:** Acreditación Universitaria, Metodología UWE, Estándar IEEE-830, UML Web Engineering, Lenguaje Unificado de Modelado, Norma ISO: Organización Internacional de Normalización.



## ABSTRACT

This Project describes the creation of a system named SIIAC-ESPE “ANALYSIS, DESIGN AND IMPLEMENTATION OF A WEB SYSTEM OF QUALITY MANAGEMENT FOR THE POSTGRADUATE PROGRAMS OF THE UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, USING JAVA ENTERPRISE EDITION JEE6 WEB PLATFORM APPLYING THE UWE METHODOLOGY”. The system will manage the process of Accreditation and Evaluation of the careers and postgraduate programs, according to the type of accreditation like careers, institutional and extensions. It will allow the load of evidences for each one of the indicators according to the evaluating criteria assigned to the coordinator and responsible. This document describes the development of a transactional Web System and distributed application through the UWE methodology which is based in the elaboration of use case models, conceptual and presentation. The obtained results for the final users are the ones requested in the requirement elicitation along for the Coordinator, Evaluator and Responsible; besides direct test were done with the stakeholders in the accreditation and this helped with the refinement of the application, the deliver – reception act was done and both parts were satisfied.

**KeyWords:** University Accreditation, IEEE-830 Standard, UML Web Engineering, Unified Modeling Language, ISO: International Organization for Standardization.

## GLOSARIO DE NOMENCLATURAS

- SIIAC: Sistema Interno Institucional de Acreditación.
- CEAACES: Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior.
- UDI: Unidad de Desarrollo Institucional.
- ERS: Especificación de Requerimientos de Software.
- UGP: Unidad de Gestión de Posgrados.
- IES: Instituciones de Educación Superior.
- CGI: Common Gateway Interface.
- HTTP: Protocolo de transferencia de hipertexto.
- UML: Lenguaje Unificado de Modelado.
- MDA: Arquitectura Dirigida por Modelos.
- UWE UML: UML-Based Web Engineering.
- GNU/GPL: GNU is Not Unix General Public License.
- IDE: Entorno de Desarrollo Integrado.
- COCOMO: Modelo Constructivo de Costos.
- MVC: Modelo Vista Controlador.
- EJB: Enterprise Java Beans.
- JSF: Java Server Faces.
- JDBC: Java Database Connectivity.
- JPA: Java Persistence API.
- BSD: Berkeley Software Distribution.
- ACID: Atomicity, Consistency, Isolation, Durability.

## **CAPÍTULO 1**

### **ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE GESTIÓN DE CALIDAD DE LOS PROGRAMAS DE POSGRADOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA JAVA ENTERPRISE EDITION JEE6 WEB APLICANDO LA METODOLOGÍA UWE.**

El constante avance de los sistemas de información, comunicación, tecnología en el Ecuador y en el mundo, facilita la automatización y estandarización de los procesos, alcanzando de esta manera la integridad, confiabilidad y disponibilidad de la información.

Con la utilización de las herramientas tecnológicas las empresas start-up, instituciones públicas y privadas logran un mayor desenvolviendo de sus actividades, hallando nuevas formas de mejoramiento para lograr cada uno de sus objetivos a través de la innovación.

En la actualidad la información es un activo estratégico de toda organización, en la que se invierte cantidades asombrosas de dinero para la obtención de sistemas seguros y confiables, controlando de esta manera toda la información con la que cuenta cada organización. Algunas entidades no comprenden aún la importancia que tiene la información, por tal motivo sucumben por su baja competitividad.

## **1.1 Antecedentes**

Como parte de su misión, la Universidad de las Fuerzas Armadas “ESPE” ha prestado servicios de formación en cuarto nivel, a través de la Unidad de Gestión de Posgrados (UGP); que coordina y apoya el desarrollo de los programas de cuarto nivel ofertados por la escuela.

Actualmente, como parte de la oferta académica, se ofrecen 24 programas de maestría, que dependen académicamente de cada departamento que ha creado ese posgrado.

La ley de educación superior dispone que en el lapso de cinco años de cuando fue expedida, todas las carreras y programas alcancen la acreditación nacional, para que continúen su funcionamiento; por tal motivo la UGP, ha organizado una Comisión de Acreditación de Posgrado, la que ha iniciado su trabajo metodológico para cumplir con los requerimientos que el Órgano de control de la Educación Superior así lo determina.

## **1.2 Justificación e Importancia**

Actualmente la Universidad de las Fuerzas Armadas “ESPE”, ubicada en la ciudad de Quito con sede en Sangolquí, no cuenta con un sistema que satisfaga los requerimientos para la gestión de calidad específicamente con las evidencias de los programas de posgrado, que son necesarias para la: **EVALUACIÓN, ACREDITACIÓN Y ASEGURAMIENTO DE LA CALIDAD DE LA EDUCACIÓN SUPERIOR.**

Los procesos para ejecutar la “evaluación de desempeño institucional de las universidades y escuelas politécnicas del Ecuador” (Ecuador, 2008); como ordena el Mandato Constituyente No 14 de 2008, consiste en la evaluación de las carreras presenciales y semi-presenciales ofrecidas por estas instituciones de educación superior (IES); dicho proceso demanda la elaboración de un modelo genérico a partir del cual se construirá modelos específicos complementarios para las carreras, por lo tanto como institución académica pionera en la educación superior, se observa la necesidad de normalizar la información de los procesos de programas de posgrado, tomando en cuenta que el registro de la información se lo realiza en archivos Excel; esta situación es propensa a pérdidas de la información y que no haya disponibilidad de acceso a todos los usuarios gestores involucrados.

La información es el valor estratégico de toda institución para cumplir de forma eficiente y eficaz sus actividades, por tanto se debe establecer políticas de protección contra cualquier desastre informático.

Adelantándose a las posibles problemáticas que pueden surgir al momento de la evaluación, acreditación y aseguramiento de la calidad de los procesos de los programas de posgrados, es necesario implementar un sistema que tenga la capacidad de obtener toda la información íntegra, confiable y disponible.

El sistema ayudará a la parametrización, administración, dirección, evaluación, seguimiento y generación de reportes de la información de los programas de posgrados.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Realizar el diseño e implementación de un sistema Web de Gestión de Calidad Evidencias de los programas de posgrados para la Universidad de las Fuerzas Armadas “ESPE”, utilizando la metodología UWE y la plataforma Java Enterprise Edition JEE6.

### **1.3.2 Objetivos Específicos**

- Elaborar la especificación de requerimientos del sistema utilizando el estándar IEEE 830-1998.
- Codificar mediante el lenguaje y herramienta Java Enterprise Edition JEE6.
- Utilizar la metodología UWE para el desarrollo de la aplicación.
- Construir un ambiente Web donde los usuarios gestores (administrador, coordinador, responsable, evaluador) de los programas de posgrado de la Universidad de las Fuerzas Armadas “ESPE”, puedan disponer de la información para operar un sistema de gestión de la calidad de sus programas.
- Probar e implementar el sistema con todas las funcionalidades que requieren los usuarios del sistema.

## **1.4 Alcance**

El tema “Análisis, Diseño e Implementación de un sistema Web de Gestión de Calidad de los programas de posgrados de la Universidad de las Fuerzas Armadas “ESPE”, mediante la utilización de la plataforma Java Enterprise

Edition JEE6 Web aplicando la metodología UWE” comprende de los siguientes módulos:

- **Módulo: Parámetros para Evaluación Administrador**

- Administrar Usuarios (Coordinador, Evaluador).
- Administrar Tipo de Evaluación.
- Administrar Período de Evaluación para la acreditación general de la universidad.
- Administrar Criterio.
- Administrar Variables de Evidencia.
- Administrar Indicadores.
- Administrar Estados del Indicador.
- Cargar Matriz Documento de Información.
- Inicializar Nuevo Período.

- **Información Universidad**

- Administrar Sedes.
- Administrar Tipo Carrera/Programa.
- Administrar Departamentos.
- Administrar Carrera/Programa.
- Administrar Tiempo de Evaluación por Carrera/Programa.

- **Módulo: Coordinador**

- Administrar Responsables.
- Asignar Responsable al Indicador.
- Actualizar Datos Personales.

- Actualizar Contraseña.
- Cargar Evidencia.
- **Módulo: Responsable**
  - Cargar Evidencia.
  - Actualizar Datos Personales.
  - Actualizar Contraseña.
- **Módulo: Evaluador**
  - Evaluar Evidencia.
  - Actualizar Datos Personales.
  - Actualizar Contraseña.
- **Módulo: Reportes**
  - Generar reportes de los indicadores de acuerdo al estado que asigna el evaluador, con el porcentaje respectivo.
  - Generar reportes de las evidencias revisadas y cargadas en el servidor.
  - Generar reportes por Carrera o Programa de los estados del indicador, el responsable y el evaluador.
- **Módulo: Seguridad de Acceso al Sistema: Control de los niveles de acceso de los usuarios al Sistema.**
  - Perfiles de usuario.
  - Asignación de Usuarios.



El Sistema se implementará en los programas de posgrado y en la Unidad de Gestión de Posgrado de la Universidad de las Fuerzas Armadas “ESPE”, de acuerdo a los requerimientos establecidos.

## **1.5 Factibilidad**

### **1.5.1 Factibilidad Técnica**

Los conocimientos requeridos para el desarrollo del sistema se detallan a continuación:

- Administración de Base de Datos, programación PL/SQL (MySql).
- Programación en Java con JSF.
- Manejo de servidores Linux.
- Manejo de herramientas CASE (StarUML, Power Designer).

Los desarrolladores cuentan con estos conocimientos y aquellos que se requieran serán investigados sobre la marcha.

### **1.5.2 Factibilidad Tecnológica**

El área de posgrado de la Universidad de las Fuerzas Armadas “ESPE” proporcionará toda la información necesaria para el desarrollo del sistema, dando a conocer cada uno de los requerimientos.

- Cumplimiento de los requerimientos del sistema a desarrollar la herramienta IDE de desarrollo a utilizar será Netbeans 7.3.
- Seguridad, estabilidad y escalabilidad como herramienta de Base de Datos se utilizará MySql 5.0.

- El Sistema operativo el cual se utilizará será Windows 7 por su estabilidad y en los cuales se procederá a instalar las herramientas de desarrollo.

### **1.5.3 Factibilidad Operativa**

Donde se llevará a cabo el proyecto será en las instalaciones de la Universidad de las Fuerzas Armadas “ESPE”, oficinas y aulas de la Unidad de Desarrollo Institucional (UDI); los coordinadores asignados por la Universidad serán quienes realicen el seguimiento del proyecto, con experiencia en sistemas de esta índole. Adicionalmente, el sistema se encontrará disponible 24/7 desde la intranet de la Universidad, ya sea por conexión TCP/IP o Wireless con su respectiva autenticación de acceso.

### **1.5.4 Factibilidad Operacional**

El sistema a desarrollar busca mejorar el proceso de evaluación para la acreditación de la Universidad de las Fuerzas Armadas “ESPE”; para lo cual se cuenta con el apoyo suficiente por parte de los encargados del proceso y los usuarios finales del sistema, con los que se llevará a cabo las pruebas previas a la implantación; consecuentemente la aplicación será fácil de usar para los usuarios involucrados en este proceso.

### **1.5.5 Factibilidad Económica**

Para llevar a cabo el proyecto, se utilizará software libre por tanto la Universidad de las Fuerzas Armadas “ESPE”; no tendrá que sobrellevar gastos por la adquisición de licencias en programas de desarrollo.

### 1.5.6 Factibilidad Legal

Con respecto a la implantación y operación del sistema no infringe ninguna ley o norma vigente en la actualidad a nivel nacional. No se solicitará, ni pagará ninguna patente o licencia; no obstante se listan a continuación las normas y metodologías utilizadas en el desarrollo.

- Metodología UWE UML.
- Norma IEEE- 830.
- ISO/IEC 29119 Software Testing Structure.

### 1.6 Presupuesto

A continuación se detalla de manera general los costos de implementación del sistema a desarrollarse, como se puntualiza en la Tabla 1.

**Tabla 1: Presupuesto aproximado para la implementación del sistema SIIAC-ESPE.**

#	DESCRIPCIÓN	CANTIDAD	V. UNITARIO	V. TOTAL
<b>Hardware</b>				
1	Laptop	2	1000	2000
	Impresora	1	150	150
<b>Software</b>				
	Licencia Netbeans	2	0	0
2	Licencia MySql	1	0	0
	Licencia Servidor de Aplicaciones Glassfish	1	0	0
<b>3 Recurso Humano</b>				
	Desarrolladores	2	2300	4600
<b>Otros</b>				
	Servicios Básicos			200
4	Internet			180
	Movilización			300
<b>Total USD</b>				<b>7430</b>

Fuente: (Autores, 2014)

## **CAPÍTULO 2**

### **MARCO TEÓRICO DE REFERENCIA**

#### **2.1 Introducción**

Este capítulo presenta la fundamentación teórica que respalda este trabajo, servirá de guía para el desarrollador del sistema SIIAC-ESPE se considerarán los siguientes conceptos:

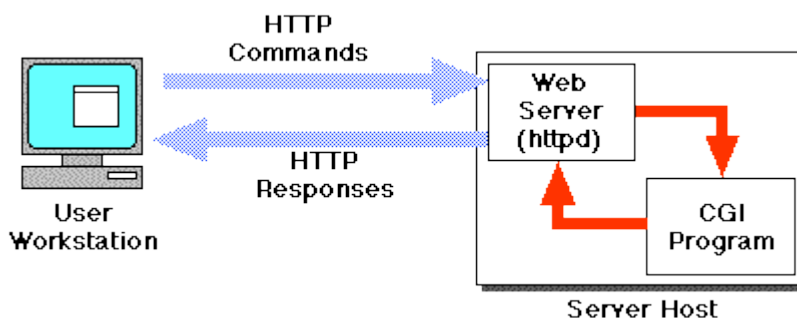
Aplicación Web, permite interactuar con los datos del sistema desde cualquier computador conectado a la intranet de la Universidad de las Fuerzas Armadas “ESPE”, desde un navegador Web al momento que desee el usuario, sin necesidad de instalar un software predeterminado. Arquitectura, cliente-servidor y cliente-servidor multicapas.

Herramientas de desarrollo a utilizar como son: base de datos MySQL, servidor Glassfish, lenguaje de programación Netbeans, framework de desarrollo de interfaces JSF. Metodología UWE UML, es un marco tecnológico dirigido para el desarrollo de Sistemas Web basado en UML (Lenguaje de Modelado Unificado).

#### **2.2 Aplicaciones Web**

Al principio la Web era sencillamente un listado de páginas estáticas para su consulta o descarga. Con el paso del tiempo la Web incluyó métodos para elaborar páginas que permitieran desplegar información dinámica, es decir que la información se genere a partir de una petición (Mora, 2001).

El primer método que se diseñó para la elaboración de la Web dinámica fue conocido como CGI (Common Gateway Interface), este era un mecanismo por el cual se podía pasar información entre el servidor y la aplicación según (Mora, 2001). La ilustración 1 muestra la petición que realiza un cliente para recibir una respuesta desde el servidor mediante el protocolo HTTP.



**Ilustración 1: Interfaz (CGI) tecnología World Wide Web.**

Debido a que los CGI tenían graves problemas de concurrencia, que eran causados por la excesiva carga que asumía un servidor al atender una petición, se empezaron a desarrollar soluciones alternativas a los CGI.

Para solucionar el problema de los CGI se dieron dos propuestas que son:

- i. Diseñar sistemas mejor integrados con el servidor, que eviten la instanciación y ejecución de varios programas.
- ii. Proveer a los servidores la capacidad de trabajar con algún lenguaje de programación que permita añadir código en las páginas; a partir de estas dos propuestas, se produce un aumento del número de arquitecturas y lenguajes para desarrollar aplicaciones Web.

- iii. Las aplicaciones Web son soluciones informáticas que se encuentran codificadas en algún lenguaje de programación y que permiten al usuario interactuar con la información por medio de un navegador Web (Mozilla, Internet Explorer, Chrome, etc.).

## **2.3 Características**

Las características que poseen las Aplicaciones Web son:

- Evitan la dependencia del Sistema Operativo, ya que se pueden ejecutar mediante un navegador.
- Optimizan el tiempo al momento de llevar a cabo el proceso de actualización debido a que no se necesita hacerlo en cada terminal.
- Al no tener que instalarse ningún software no consumen ningún espacio en el disco. Además tampoco consumen recursos, ya que el encargado de ejecutar los procesos es el servidor.

## **2.4 Herramientas y Bases de datos**

### **2.4.1 StarUML**

StarUML es una herramienta para el modelamiento de software basado en los estándares UML (Lenguaje Unificado de Modelado) y MDA (Arquitectura Dirigida por Modelos), que en un principio era un producto comercial y ahora es un producto de licencia abierta GNU/GPL (StartUML, 2006).

El software heredó todas las características de la versión comercial y poco a poco ha ido mejorando sus características, entre las cuales se encuentran:

- Soporte completo al diseño UML mediante el uso de:

- Diagrama de casos de uso.
- Diagrama de clase.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividad.
- Diagrama de componentes.
- Diagrama de despliegue.
- Diagrama de composición estructural (UML 2.0).
- Definir elementos propios para los diagramas, que no necesariamente pertenezcan al estándar de UML. La capacidad de generar código a partir de los diagramas y viceversa, actualmente funcionando para los lenguajes c++, c# y java.
- Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas.
- Plantillas de proyectos.
- Posibilidad de crear plugins para el programa.

En definitiva esta es una de las mejores alternativas gratis que hay en Internet para el modelamiento de software y ayuda para el diseño de sistemas robustos aplicando las metodologías de desarrollo para este caso UWE UML.

## **2.5 Herramientas de Desarrollo**

### **2.5.1 Lenguaje de Programación Java**

“Java es un lenguaje de Programación Orientado a Objetos.” (Byous, 2005). Las ideas fundamentales en la creación de Java fueron que sea orientado a objetos, independiente del sistema operativo en el cual se esté trabajando, además de ser seguro para trabajar en red al ejecutar el código de sistemas remotos.

La principal característica de este lenguaje es que todo programa desarrollado en java debe ser compilado transformándolo en código bytecode y a la vez este es interpretado por una máquina virtual.

En java se usa el manejo de excepciones para evitar problemas durante el tiempo de ejecución del programa. Estas excepciones son procedimientos para tratar errores. Cuenta con un cargador de clases que permite diferenciar entre las clases que se están accediendo de forma remota y las clases locales.

### **2.5.2 IDE de desarrollo Netbeans**

Netbeans es un entorno de desarrollo integrado (IDE), kit de desarrollo de software, con licencia libre, dirigido hacia el lenguaje de programación Java.

La plataforma de desarrollo Netbeans crea las aplicaciones en base a un conjunto de componentes de software denominados módulos. Estos son archivos que poseen clases Java las cuales interactúan con la interfaz de programación de Netbeans, además dichos módulos poseen un archivo de identificación para diferenciarse unos de otros.



Netbeans permite a los desarrolladores administrar las interfaces de usuario de una manera visual haciéndolas más sencillas, dejando mayor cabida al enfoque de la lógica específica de su aplicación. Además permite utilizar un sin número de tipos de datos para el procesamiento de información, su interfaz gráfica se basa en ventanas para una mejor navegabilidad, posee la característica del uso de asistentes para la creación de componentes.

Netbeans está escrito en Java pero brinda la posibilidad de trabajar con otros lenguajes de programación para la creación de aplicaciones de escritorio, Web, móviles, etc. Al poseer tecnología Java es portable, es decir que no se limita a un sistema operativo en particular.

Los datos mencionados a continuación se obtuvieron usando SLOCCount (del inglés cómputo de líneas de código fuente) es un proyecto de software libre (Wheeler, 2001).

La tabla 2 muestra los lenguajes de programación utilizados para la creación de Netbeans IDE 6.1 contiene aproximadamente 1.990.915 líneas de código de las cuales el 99.19% son de Java. El esfuerzo para producir un software de este tamaño según el Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés Constructive Cost Model) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detalla es de aproximadamente 582 persona / año. (Domínguez-Dorado, 2005).

**Tabla 2: Lenguajes de Programación utilizados en Netbeans IDE 6.1**

<b>LENGUAJE</b>	<b>LÍNEAS DE CÓDIGO</b>	<b>%</b>
<b>JAVA</b>	1.990.915	99,19%
<b>JSP</b>	7.917	0.40%
<b>Haskell</b>	3.138	0,16%
<b>CPP</b>	1.761	0.09%
<b>Yacc</b>	1.123	0,06%
<b>Sh</b>	1.080	0,05%
<b>Lex</b>	506	0,03%
<b>Perl</b>	350	0,02%
<b>Objc</b>	288	0,01%
<b>Ansic</b>	20	0,00%

**Fuente: (Wikipedia, 2013)**

### 2.5.3 Java Server Faces (JSF)

JSF es un framework que se centra en el desarrollo de interfaces gráficas de usuario. Permite desarrollar aplicaciones bien estructuradas, ya que realiza una separación entre el comportamiento y presentación, además posee su propio servlet como controlador, implementando así el patrón Modelo Vista Controlador (MVC).

Java Server Faces es un framework de desarrollo de aplicaciones Web que implementa el patrón MVC cuyo objetivo es el diseño ágil de interfaces de usuarios (Mora, 2001).

Una aplicación empresarial en Java se compone de distintas capas, cada capa tiene una función muy específica. Dividir una aplicación en capas tiene

varias ventajas, como son separación de responsabilidades, un mejor mantenimiento a la aplicación, especialización de los programadores en cada capa, entre muchas más.

La versión empresarial de Java brinda un API distinta para cada capa de una aplicación empresarial, desde la capa de presentación, la capa de negocio y la capa de datos.

A continuación se menciona cada una de las capas de una aplicación multicapas.

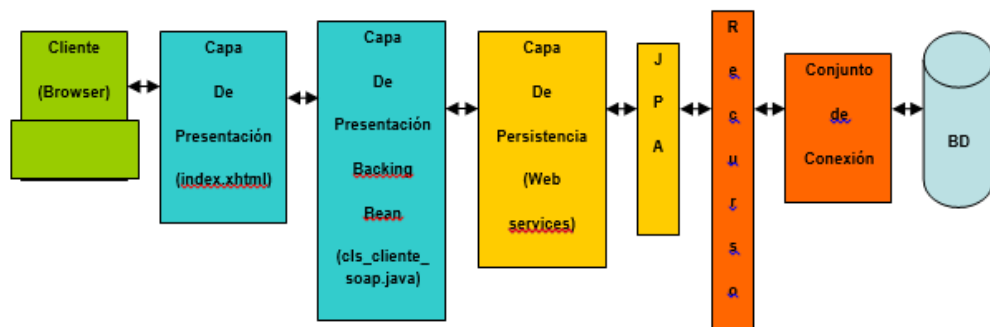
- i. **Capa Web:** La capa del Cliente es donde el cliente interactúa por medio de un navegador Web, un cliente móvil, una aplicación de escritorio, entre otros. También puede residir en un servidor web, las tecnologías más básicas que se pueden encontrar en este servidor web son los JSP's y los Servlets o Java Server Faces.
- ii. **Capa de Negocio:** en esta capa se puede encontrar tecnología como son los Enterprise Java Beans (EJBs).
- iii. **Capa de Datos:** Aquí se encuentran tecnologías como JDBC, o JPA.

La ilustración 2 muestra la tecnología JSF da la posibilidad de fabricar aplicaciones Web que separan la lógica del negocio de la presentación, esto permite la distribución de tareas para el diseño de la aplicación y proporciona un modelo sencillo de programación para la integración.



**Ilustración 2: Arquitectura Multicapas con la capa de presentación JSF Fuente: (Mentoring, 2012).**

JSF permite crear sus propios componentes personalizados, así como generar salidas para diferentes dispositivos clientes. La ilustración 3 muestra la arquitectura implementada en el sistema SIIAC-ESPE; como el cliente desde un navegador web interactúa con la capa de presentación, la capa de persistencia, el recurso y con la conexión de base de datos MySQL.



**Ilustración 3: Arquitectura de Implementación Sistema SIIAC-ESPE. Fuente: (Almeida, 2013).**

#### 2.5.4 Base de Datos MySQL

MySQL es el Sistema Gestor de Bases de Datos de código abierto, los cimientos de MySQL fueron hechos por académicos y alumnos universitarios quienes pretendieron desarrollar un gestor de base de datos que brinde una solución a los problemas que poseían los demás gestores de base de datos.

A través del tiempo se desarrolla una comunidad, cuya labor ha sido trabajar sobre los cimientos de MySQL y desarrollar un avanzado Sistema Gestor de Base de Datos que brinde solución y eficiencia a los problemas del mundo empresarial. Con ese fin, se han desarrollado y añadido a MySQL múltiples y útiles características que antes sólo se podían hallar en sistemas manejadores de bases de datos comerciales cuyos costos son muy elevados.

Es un poderoso sistema manejador de bases de datos publicado bajo licencia BSD, el cual se puede ejecutar en la mayoría de los Sistemas Operativos más utilizados incluyendo, Linux, Unix y Windows. Algunas de las características que posee MySQL son:

MySQL cumple la prueba ACID (Atomicity, Consistency, Integrity, Durability) y tiene soporte completo para:

- Llaves foráneas.
- Joins.
- Vistas.
- Subconsultas.
- Triggers.

- Procedimientos almacenados (en varios lenguajes).
- Incluye la mayoría de los tipos de datos de los estándares (integer, numeric, boolean, char, varchar, date, interval, timestamp, entre otros).
- También soporta almacenamiento de objetos grandes (imágenes, sonido y video).
- Posee un lenguaje nativo llamado (PL/PGSQL) que es muy similar al PL/SQL de Oracle.

### **2.5.5 Servidor de Aplicaciones Glassfish**

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito, de código libre y se distribuye bajo un licenciamiento dual a través de la GNU GPL. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server).

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation; este último proporcionó el módulo de persistencia; tiene como base al servidor Sun Java System Application Server de Oracle Corporation, un derivado de Apache Tomcat.

Glassfish posee varias características entre las cuales se destaca su capacidad de procesar funciones de alta velocidad, escalabilidad, manejo de instancias, bajo consumo de memoria, posee una consola de administración teniendo así una administración centralizada. (Wikipedia, 2013).

## **2.6 Metodología UWE**

### **2.6.1 Definición**

UWE (UML Web Engineering) es un método de ingeniería del software para el desarrollo de aplicaciones Web basado en UML (Lenguaje Unificado de Modelado). Cualquier tipo de diagrama UML puede ser usado, porque UWE es una extensión de UML (Perdita Stevens, 2002).

### **2.6.2 Características**

Las principales características de la metodología UWE son las siguientes:

- Uso de una notación estándar (UML).
- Definición de métodos.
- Especificación de restricciones.

La ilustración 4 muestra las dimensiones del modelamiento de la metodología de desarrollo UWE. Direccionada en 3 ejes X, Y, Z, donde se puede observar en el eje X las fases de análisis, diseño e implementación; en el eje Y las vistas como son contenido, estructura de navegación y presentación; en el eje Z muestra los aspectos estructura y comportamiento.



**Ilustración 4: Dimensiones Modelamiento UWE.**

### 2.6.3 Fases de la Metodología UWE

Esta metodología consta de una notación basada en UML y un método que consta de 5 fases que son:

- Análisis de Requerimientos.
- Modelo Conceptual.
- Modelo Navegacional.
- Modelo de Presentación.

Para la implementación del sistema SIIAC-ESPE, se utilizará las fases de Análisis de Requisitos, Modelo Conceptual que es la base para la construcción y recomendación en el desarrollo.

### 2.6.4 Análisis de Requisitos

El modelo de requerimientos tiene como objetivo principal comprender los procesos a realizarse en el sistema y delimitar su alcance; los requerimientos se



pueden clasificar en funcionales y no funcionales. Entre los funcionales existen los de contenido, presentación y usuarios; para obtener estos requisitos se usan medios de recolección de información como entrevistas y cuestionarios.

### **2.6.5 Modelo Conceptual**

Este modelo muestra cómo se encuentran relacionados los contenidos del sistema, aquí se especifican las clases y sus relaciones dentro del sistema Web; este modelo usa los diagramas de clases para definir la estructura de los datos que se encuentran alojados en el sitio Web.

En los cuales se representan los conceptos, las unidades de información y usuarios; se usan los diagramas de secuencia y de estado para visualizar los mensajes entre objetos y las acciones que llevan a una transición de estado.

### **2.6.6 Modelo Navegacional**

Este modelo indica como las páginas Web del sitio están relacionadas internamente; el modelo de navegación se enfoca en los puntos donde el usuario, puede llevar a cabo una acción y estos puntos se los llama nodos. Además estudia los enlaces que llevan a un nodo (acción), o que se originaron en el mismo. El objetivo del modelo de navegación es representar el diseño y estructura de las rutas de navegación al usuario.

### **2.6.7 Modelo de Presentación**

En este modelo se representan las clases y los procesos que pertenecen a cada página Web. Este modelo permite crear una vista de la interfaz de usuario de la aplicación Web.

## 2.7 Artefactos

### 2.7.1 Diagramas de Casos de Uso

Un diagrama de casos de uso es una colección de situaciones que se documentan a partir del punto de vista del usuario. Por lo tanto los casos de uso es una de las herramientas más importantes, en la conversión de los requisitos vistos desde el lado del usuario en requisitos a ser implementados por los profesionales del desarrollo de Software (Rossi G., 2000).

La ventaja principal es que debido a la facilidad para interpretarlos, también es útil en la comunicación con el cliente. Es decir el cliente no debe preocuparse por tecnicismos al momento de ayudar en el desarrollo del sistema.

En la tabla 3 muestra la notación UWE para los diagramas de caso de uso, los requerimientos funcionales desde el punto de vista de usuario y usuario registrado.

**Tabla 3: Notación UWE para casos de uso.**

Tipo de caso de uso	Estereotipo UWE	Símbolo
De navegación	«navegación»	
De proceso	UML tradicional	UML tradicional
Personalizado	«personalizado»	

**Fuente: (Rossi G., 2000)**

### **2.7.2 Diagrama de Clases**

Es un diagrama estático el cual se encarga de describir en forma gráfica la estructura de un sistema usando objetos conceptuales. A los diagramas de clases se les utiliza durante la etapa de análisis y diseño de los sistemas.

El elemento fundamental en este diagrama son las clases, las cuales son la representación conceptual del mundo real. Estas clases también deben tener relaciones, las mismas que deben evidenciar la interacción de los objetos en el mundo real.

Entre las relaciones que se pueden añadir a un diagrama de clases están:

- Herencia.
- Composición.
- Agregación.
- Asociación.

### **2.7.3 Diagrama de Secuencia**

El diagrama de secuencia muestra la secuencialidad de las tareas que se lleva a cabo para cumplir con la funcionalidad de un caso de uso. En este diagrama se puede ver la distribución de los módulos y como el sistema va a actuar cuando un usuario requiera algún caso de uso en especial.

Los diagramas de secuencias se componen de los elementos siguientes:

- Los objetos.
- Los mensajes.
- Los métodos.

#### **2.7.4 Diagrama de Estado**

El diagrama de estado es aquel que muestra el dinamismo de una aplicación. Este diagrama refleja los cambios de estados que sufre un objeto en función a los sucesos y al tiempo.

Este diagrama es usado para encontrar cada una de los caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Los componentes existentes en el diagrama de estados son:

Eventos.

- Acciones.
- Actividades.
- Transiciones.
- Estados.



#### **2.7.5 Diagrama de Despliegue**

Define la relación existente entre los componentes, archivos y elementos de hardware que conforman el sistema.

#### **2.7.6 Diagrama de Actividades**

La tabla 4 muestra la notación UWE para el diagrama de actividades, permite ver el comportamiento de cada proceso de la aplicación Web y las interfaces que permiten manejar dichos procesos; representan los flujos operacionales de los componentes de un sistema Web (Rossi G., 2000).

**Tabla 4: Notación UWE para elementos del diagrama de actividad.**

Elemento	Estereotipo UWE	Símbolo
Contenido	«contenido»	
Nodo	«nodo»	
Interfaz de usuario	«WebUI»	
Navegar	«navegar»	

**Fuente: (Rossi G., 2000)**

## 2.8 Lenguaje de Modelamiento Unificado (UML)

UML es un lenguaje de modelado de sistemas de software. Entre sus funciones permite especificar, construir y documentar un sistema. Permite describir los procesos de negocio, funciones del sistema, esquema de bases de datos, etc., de manera que se permita cumplir con todos los requerimientos del sistema. Al ser UML un lenguaje de modelado se utiliza para documentar el sistema creando los artefactos para las diferentes metodologías de desarrollo de software, es decir ayuda a diagramar los requerimientos.

UML es una herramienta visual que permite la creación de artefactos más puntuales como son los diagramas de modelos, esquemas de bases de datos, componentes reusables de software, etc.

Pero UML no solo se enfoca en el diseño de diagramas sino también ayuda a una comprensión detallada del sistema, el cual se lo realiza mediante los casos de uso. Las bases en las que se enfoca UML son modelar sistemas de software con conceptos de orientación a objetos, utilizar un lenguaje de

modelado entendible a nivel de máquina y de personas, poseer una fuerte relación entre los conceptos y los artefactos creados.

Los diagramas UML ayudan en el modelado de los sistemas en lo que corresponde a la definición de clases, objetos, atributos, operaciones y relaciones; además de las colaboraciones existentes entre los diferentes objetos, así como los cambios de estado que este objeto presenta en el transcurso del sistema.

## 2.9 Ingeniería de Requerimientos (Estándar IEEE 830)

La ingeniería de requisitos es una herramienta cuyo objetivo es satisfacer las expectativas del usuario en base al análisis de las necesidades y negociando una solución razonable libre de ambigüedades.

El proceso de ingeniería de requerimientos se divide en 5 etapas:

- i. **Identificación de Requisitos:** Es la primera etapa de la ingeniería de requerimientos, la cual aunque parece ser extremadamente simple, es una de las etapas más críticas y complejas. En esta etapa se hace una recolección de todos los requisitos vistos desde el lado del usuario.
- ii. **Análisis de Requisitos y Negociación:** En esta etapa se analizan cada uno de los requisitos agrupándolos por categorías y en función a las necesidades de los clientes. Además cada requisito debe cumplir las propiedades de consistencia y completitud. Los requisitos deben ser delimitados y libres de dobles interpretaciones.

- iii. **Especificación de requisitos:** Se define un documento o modelo gráfico en el cual se describen las funciones, limitaciones y características que han de controlar el desarrollo del sistema. Es recomendable que en el desarrollo de sistemas grandes se utilicen documentos escritos y modelos gráficos.
- iv. **Modelado del Sistema:** Se desarrollan modelos del sistema, los cuales sirven para comprender y evaluar los componentes y relaciones que existen en el sistema.
- v. **Validación de Requisitos y gestión de Requisitos:** la validación de requisitos es una etapa de verificación, la cual se asegura de que todos y cada uno de los requisitos hayan sido establecidos, sin ambigüedades, sin inconsistencias y que los errores detectados en la etapa de análisis de requisitos, hayan sido corregidos.

La etapa de Gestión administra cada uno de los requisitos mediante la asignación de un identificador, el cual permitirá identificar, controlar, dar seguimiento a los requisitos y sus cambios en cualquier momento.

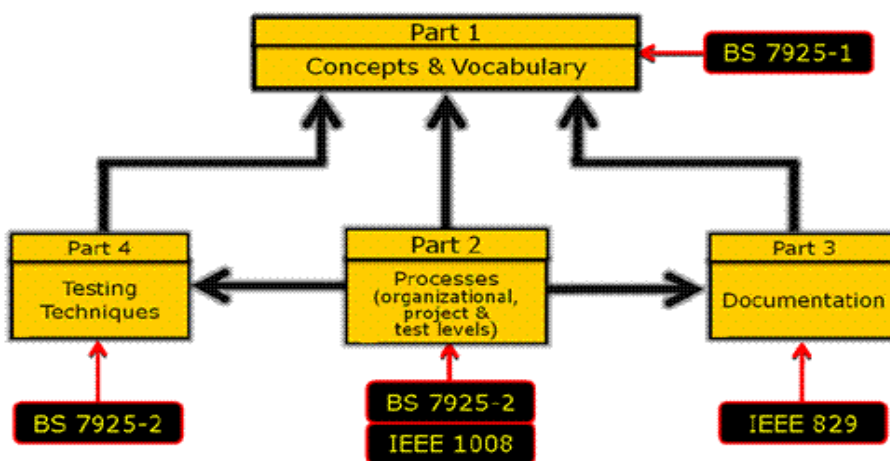
## 2.10 Pruebas de software

El objetivo de la norma ISO / IEC 29119 Pruebas de software es proporcionar una norma definitiva para las pruebas de software que define el vocabulario, procesos, documentación, técnicas y un modelo de evaluación del proceso de pruebas de software que se puede utilizar dentro de cualquier ciclo de vida de desarrollo (Tuya, 2011).

La norma se centra en un modelo de proceso de tres niveles basado en el riesgo para las pruebas de software que proporciona orientación sobre el desarrollo de estrategias de prueba organizativas y políticas, la gestión de proyectos de prueba incluyendo el diseño de estrategias de prueba del proyecto nivel, seguimiento, control de pruebas, y un proceso de prueba dinámica que proporciona una guía para el análisis y diseño de prueba, entorno de prueba de configuración y mantenimiento, la ejecución de prueba y se informa.

La ilustración 5 muestra las 4 partes que contempla la ISO / IEC 29119:

- Parte 1: Definiciones y Vocabulario
- Parte 2: Proceso de Prueba
- Parte 3: Documentación de prueba
- Parte 4: Técnicas de ensayo



**Ilustración 5: ISO/IEC 29119 Software Testing Structure.**

Las pruebas de software también denominadas testing se realizan cuando se dispone del código ejecutable del sistema, este es evaluado a través de la ejecución en circunstancias controladas, para la búsqueda de errores.



## CAPÍTULO 3

### ANÁLISIS Y DISEÑO DE LA APLICACIÓN

#### 3.1 Introducción

Este capítulo presenta la Especificación de Requerimientos de Software (ERS) para el desarrollo del Sistema SIIAC-ESPE, el cual ha sido elaborado tomando en cuenta las necesidades presentes en la Universidad de las Fuerzas Armadas “ESPE” y en las directrices dadas por el estándar IEEE 830.

#### 3.2 Propósito

El propósito de esta especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema a construir. El documento va dirigido tanto a los desarrolladores como a los usuarios de la Universidad de las Fuerzas Armadas “ESPE”. Después de su aprobación, este documento constituirá la base sobre la cual se establecen las características funcionales y requerimientos de hardware y software, siendo esta la guía del proceso de desarrollo del sistema SIIAC-ESPE.

#### 3.3 Alcance

El sistema tiene como meta solucionar los problemas de organización de gestión de evidencias para la acreditación de los programas de posgrado de la Universidad de las Fuerzas Armadas “ESPE” de la siguiente manera:

- El sistema incluirá el ingreso de los gestores de evaluación, control y

gestión de carga de evidencias por programa de posgrado, reportes de carga de evidencias para el administrador, coordinador, evaluador, ayuda para la acreditación del CEACES (Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior), avance y observaciones de cada programa de posgrado con respecto a las evidencias de acuerdo a los criterios de evaluación.

### **3.4 Visión Global**

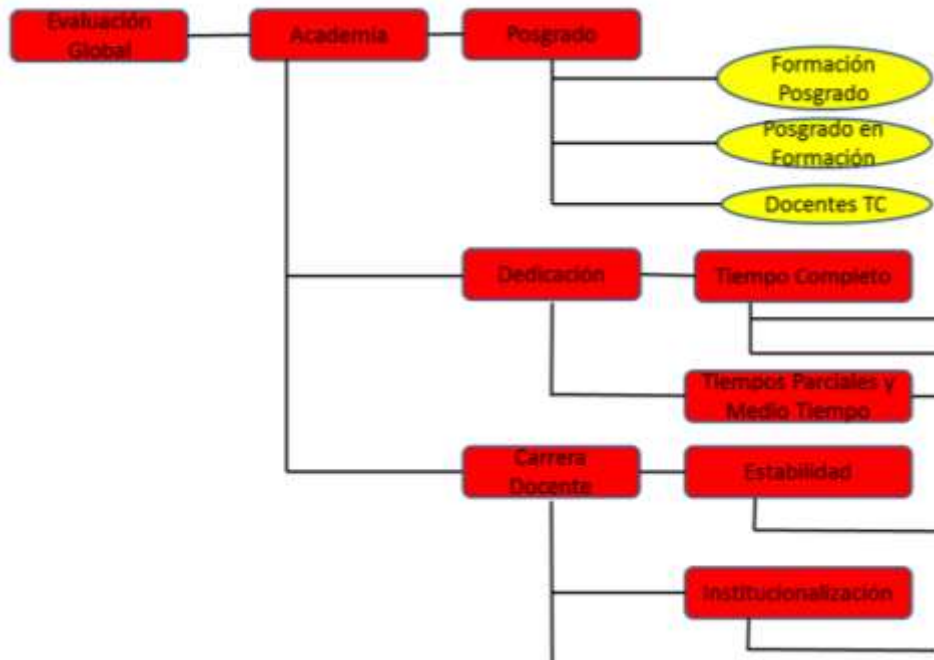
El documento se divide en tres secciones: La primera sección (sección actual) consta de introducción, propósito y alcance, dando una visión general de la ERS. La segunda sección consta de la perspectiva del producto, funciones del producto, características del usuario y restricciones generales, para obtener las funciones que debe realizar, los datos asociados. La tercera y última sección se divide en requerimientos de interfaces externas, interfaces de software, interfaces de comunicación, requerimientos funcionales, restricciones de diseño y atributos; los cuales definen de manera detallada los requisitos que debe satisfacer el sistema.

### **3.5 Descripción**

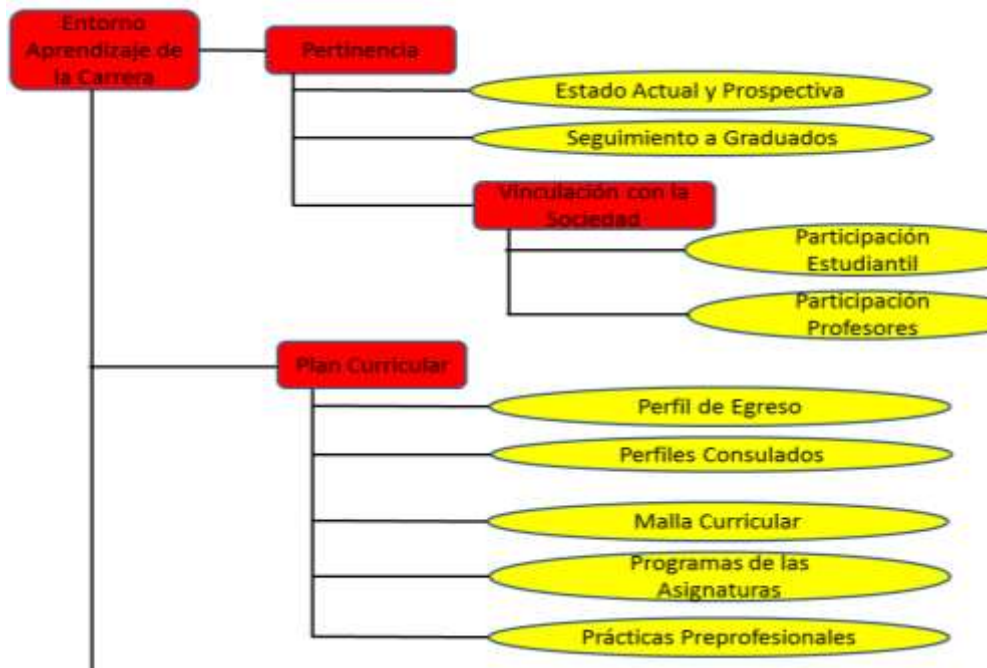
#### **3.5.1 Perspectiva del producto**

El sistema a desarrollar agrupará en un entorno macro de la carga de evidencias dividiéndole de acuerdo al tipo de acreditación ya sea este institucional, carrera o extensiones el sistema automáticamente crea un path para la generación de evidencias de acuerdo al árbol de acreditación como se

muestra en la ilustración 6 y 7.



**Ilustración 6: Modelo Árbol de Acreditación Institucional.**  
Fuente: (Ecuador, 2008).



**Ilustración 7: Modelo Árbol de Acreditación Carreras.**  
Fuente: (Ecuador, 2008).

### **3.5.2 Funciones del producto**

Sistema SIIAC-ESPE:

- Parametrizar todas las matrices de evaluación tanto para Carreras, Extenciones e Institucional.
- Cargar documentación que ayuden a cumplir con el proceso de evaluación y acreditación de la Universidad.
- Difundir la información de las carreras en proceso de acreditación, avances de la carga de evidencias y el tiempo estimado de las mismas.
- Administrar las cuentas de usuarios para restringir el ingreso a zonas no permitidas según sea su cargo, de esta manera se asegura la confiabilidad e integridad de los datos.

### **3.5.3 Tipos y Funciones del Usuario**

Se definen 4 tipos de usuarios, los cuales son:

- Administrador del Sitio: este usuario se encarga de la administración, mantenimiento y actualización del Sitio.
- Coordinador: son las personas que se encargan del registro de los responsables; asignación del indicador o indicadores al responsable y carga de evidencias.
- Responsable: este usuario tendrá acceso a cargar evidencias designados a su cargo.
- Evaluador: estos usuarios podrán calificar las evidencias de acuerdo al estado de la evidencia.

### **3.5.4 Restricciones generales**

Es necesario para los usuarios del sistema SIIAC-ESPE tener instalado un navegador web y acceso a la intranet de la Universidad de las Fuerzas Armadas “ESPE”; cada usuario del Sistema SIIAC-ESPE necesita un nombre de usuario y una contraseña.

### **3.5.5 Interfaces de software**

- Plataforma JEE6.
- Lenguaje de programación Java.
- Base de datos MySQL.
- Servidor de Aplicaciones Glassfish 3.1.2.

### **3.5.6 Interfaces de comunicación**

El sistema SIIAC-ESPE es una aplicación orientada a la web. La Universidad de las Fuerzas Armadas “ESPE”, posee un servidor para la implementación del sistema. Por lo tanto todo el trabajo a desarrollar se lo alojará en el servidor de la UDI (Unidad de Desarrollo Institucional).

### **3.5.7 Atributos**

Para el desarrollo se requerirá de software de código abierto como lo son: Plataforma Java Enterprise Edition JEE6, Netbeans, base de datos MySQL y servidor de aplicaciones Glassfish; se desarrollará en plataforma web. El diseño debe ser, agradable al usuario y fácil de interactuar con los usuarios.

### 3.6 Metodología UWE

#### 3.6.1 Modelo Conceptual

En la tabla 5 se describen las entidades con las que cuenta el modelo conceptual y físico. A sí mismo, la ilustración 8 muestra, el modelo conceptual que se realizó en el sistema SIIAC-ESPE la relación de las tablas que se utilizó para la generación del script de la base de datos; cabe mencionar que este modelo es la parte medular para la implementación en este se obtiene toda la lógica del negocio y la parte transaccional del sistema. Se podrá observar que el modelo no cuenta con lazos cíclicos o cierres como recomiendan los estándares de diseño de base de datos.

**Tabla 5: Descripción de las Tablas del Sistema**

ENTIDADES	
Tabla de Datos	Detalle
<b>UNIVERSIDAD</b>	Almacena los datos de la Universidad. Se encuentran por defecto en el sistema.
<b>DEPARTAMENTO</b>	Almacena los datos de los departamentos existentes en la Universidad
<b>CARRERA</b>	Almacena los datos de las carreras existentes en la Universidad.
<b>TIPOCARRERA</b>	Almacena los tipos de carrera existentes en la Universidad.
<b>ARBOLEVALUACION</b>	Almacena los nombres de los árboles de evaluaciones existentes. Se encuentran por defecto en el sistema.
<b>EVALUACION</b>	Almacena los datos de la evaluación global que se realizará en la Universidad.
<b>CRITERIO</b>	Almacena los datos de los criterios que intervienen en el proceso de evaluación y acreditación.

CONTINÚA



<b>INDICADOR</b>	Almacena los datos de los indicadores involucrados en el proceso de evaluación y acreditación.
<b>EVIDENCIA</b>	Almacena los datos de las evidencias cargadas por cada uno de los responsables.
<b>VARIABLE</b>	Almacena las variables involucradas en el proceso de evaluación y acreditación.
<b>DOCUMENTO</b>	Almacena los nombres de los documentos de información de cada árbol de evaluación.
<b>COORDINADOR</b>	Almacena los datos personales de los coordinadores de la Universidad.
<b>EVALUADOR</b>	Almacena los datos personales de los evaluadores de la Universidad.
<b>RESPONSABLE</b>	Almacena los datos personales de los responsables de la Universidad.
<b>RESPONSABLEINDICADOR</b>	Almacena los datos del responsable con su indicador involucrados en el proceso de evaluación y acreditación.
<b>TIEMPOEVALUACION</b>	Almacena los datos del tiempo de evaluación por carrera o programa con las que cuenta la Universidad.

Fuente: (Autores, 2014)

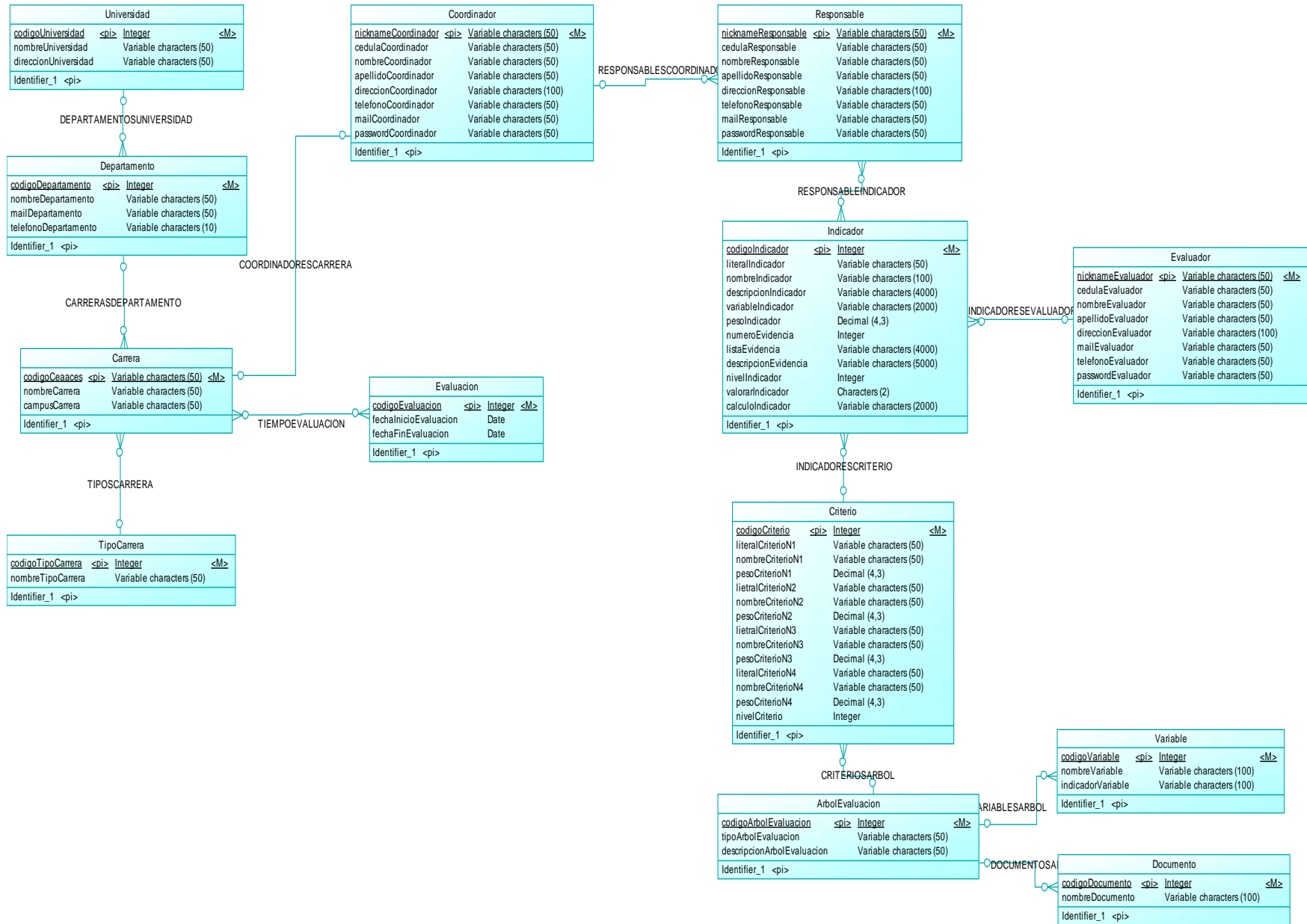


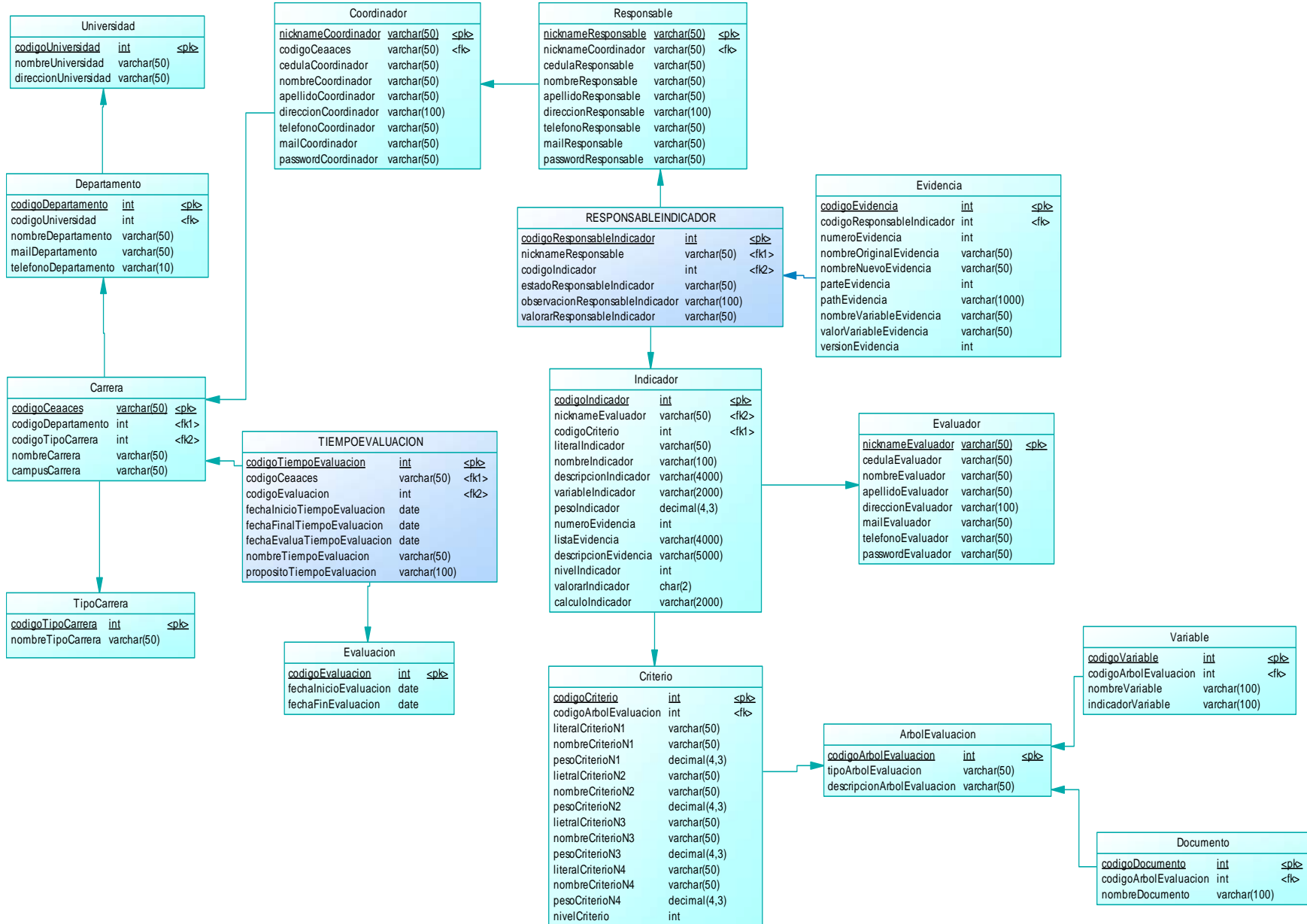
Ilustración 8: Modelo Conceptual del Sistema SIAC-ESPE.



### 3.6.2 Modelo Físico

La ilustración 9 muestra el modelo físico obtenido para el sistema SIIAC-ESPE, este se genera a partir del modelo conceptual, donde se cuenta con 16 entidades. Aquí se rompen las relaciones de varios a varios formando de esta manera nuevas entidades como: TIEMPOEVALUACION y RESPONSABLEINDICADOR, las cuales son las más importantes para el desarrollo del sistema. Se logró diseñar la base de datos en tercera forma normal ya que en este tipo de sistemas distribuidos la parte más medular es la funcionalidad y el nivel de transaccionalidad de la base de datos.

Las tablas se encuentran relacionadas de la siguiente manera: Una universidad puede tener varios departamentos; un departamento puede tener varias carreras y una carrera puede tener un tipo de carrera. Cada carrera tiene una fecha de tiempo de evaluación y este se encuentra dentro de una evaluación. Por otro lado una carrera tiene un coordinador y este tiene varios responsables. Un responsable tiene varios indicadores a su cargo, este indicador pertenece a un criterio y este pertenece a un árbol de evaluación. Un árbol de evaluación puede tener varias variables y documentos asociados. Además un evaluador tiene varios indicadores a su cargo. Un responsable del indicador puede tener varias evidencias cargadas.



**Ilustración 9: Modelo Físico del Sistema SIAC-ESPE.**

### 3.6.3 Modelo de Presentación

La ilustración 5 muestra, el modelo de presentación que se utilizó en el sistema SIIAC-ESPE; en la parte superior el logotipo de la institución en la parte inferior el pie de página, a la izquierda el menú desplegable y a la derecha el lugar de trabajo.

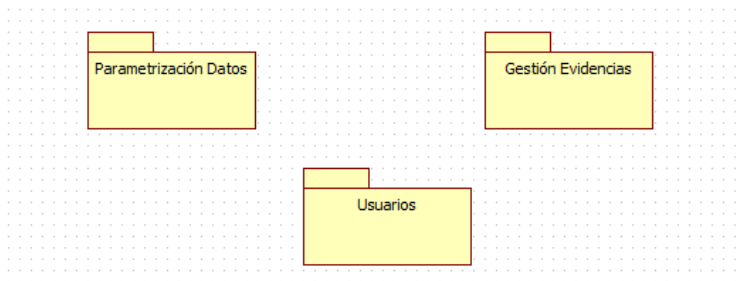


**Ilustración 10: Modelo de Presentación del Sistema SIIAC-ESPE.**

### 3.6.4 Modelo de Paquetes

La ilustración 11 muestra los paquetes que contiene el sistema SIIAC-ESPE; se describe los siguientes paquetes:

- Parametrización Datos: casos de uso del ingreso de la información sobre la acreditación; como: criterio, indicador, variable, estado.
- Gestión de Evidencias: casos de uso acerca de las evidencias de cada indicador; como: asignación de indicador, carga de evidencia.
- Usuarios: casos de uso de los datos de los usuarios; como: actualización de sus datos personales.



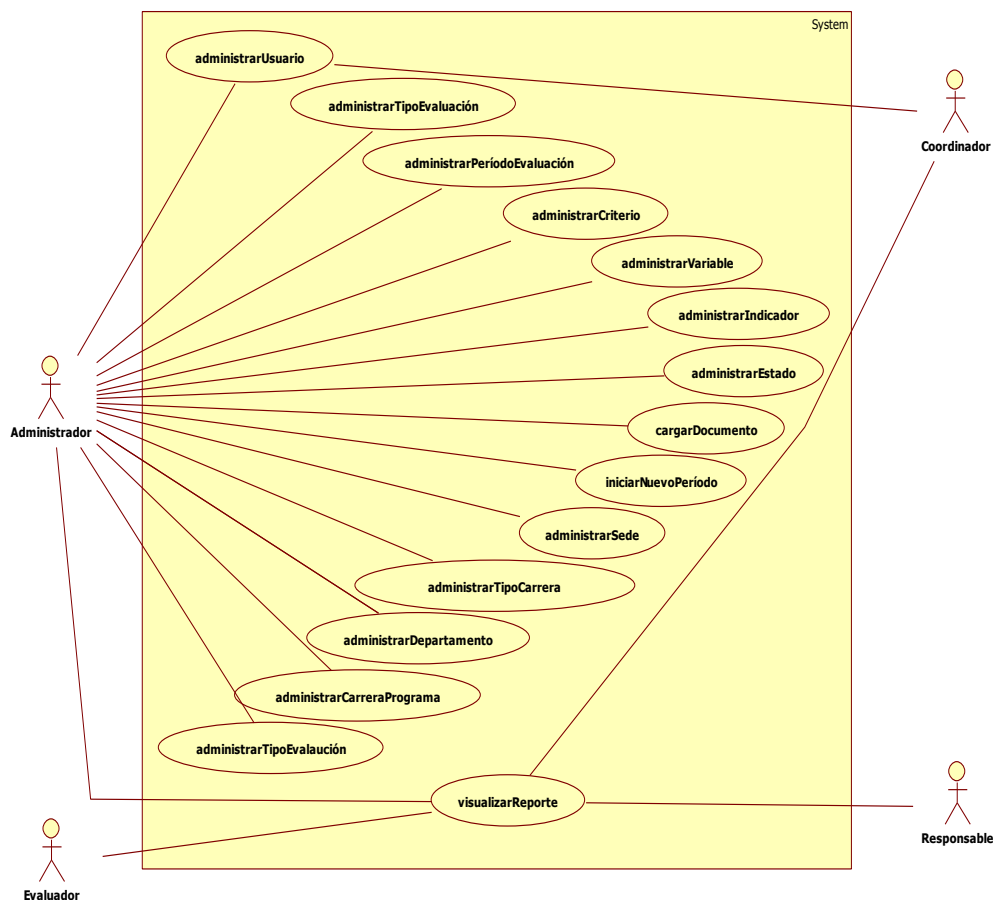
**Ilustración 11: Paquetes del Sistema SIAAC-ESPE.**

### **3.6.5 Casos de Uso del Paquete Parametrización Datos**

La ilustración 12 muestra los casos de uso del paquete parametrización datos en los cuales interactúan los actores del sistema administrador, coordinador, evaluador y responsable; se describen a continuación:

- Administrar Usuario.
- Administrar Tipo de Evaluación.
- Administrar Período de Evaluación.
- Administrar Criterio.
- Administrar Variable.
- Administrar Indicador.
- Administrar Estado.
- Cargar Documento.
- Iniciar Nuevo Período.
- Administrar Sede.
- Administrar Tipo Carrera.
- Administrar Departamento.
- Administrar Carrera/Programa.

- Administrar Tipo Evaluación.
- Visualizar Reporte.



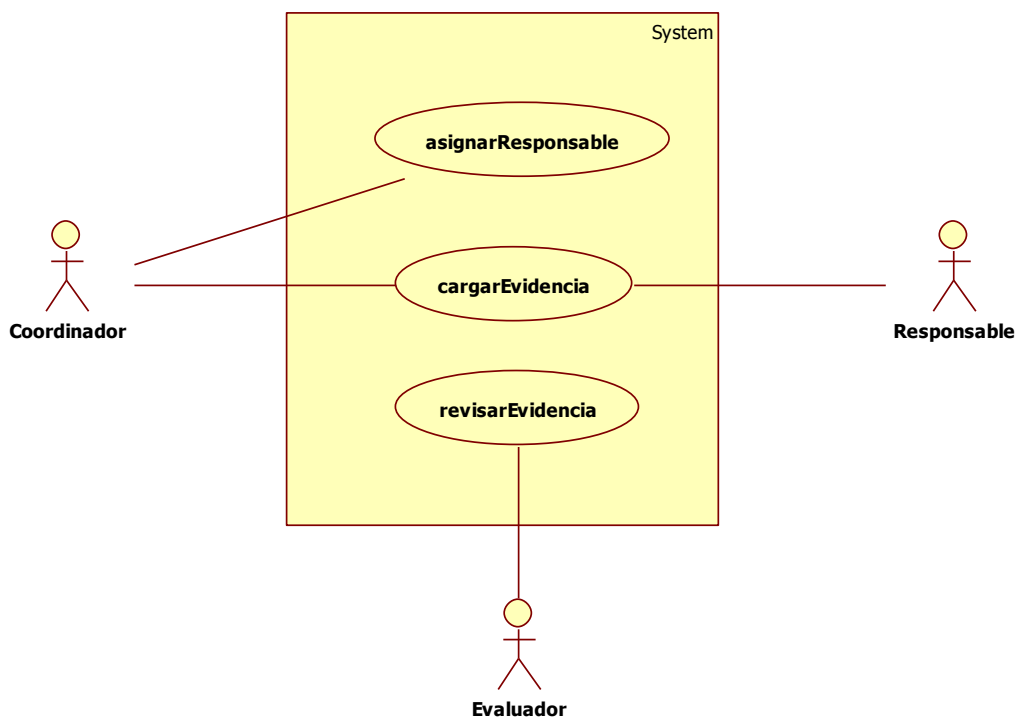
**Ilustración 12: Casos de Uso Parametrización Datos**

### 3.6.6 Casos de Uso del Paquete Gestión Evidencias

La ilustración 13 muestra los casos de uso del paquete gestión de evidencias en los cuales interactúan los actores del sistema coordinador, responsable y evaluador se describen a continuación:

- Asignar Responsable.
- Cargar Evidencia.

- Revisar Evidencia.

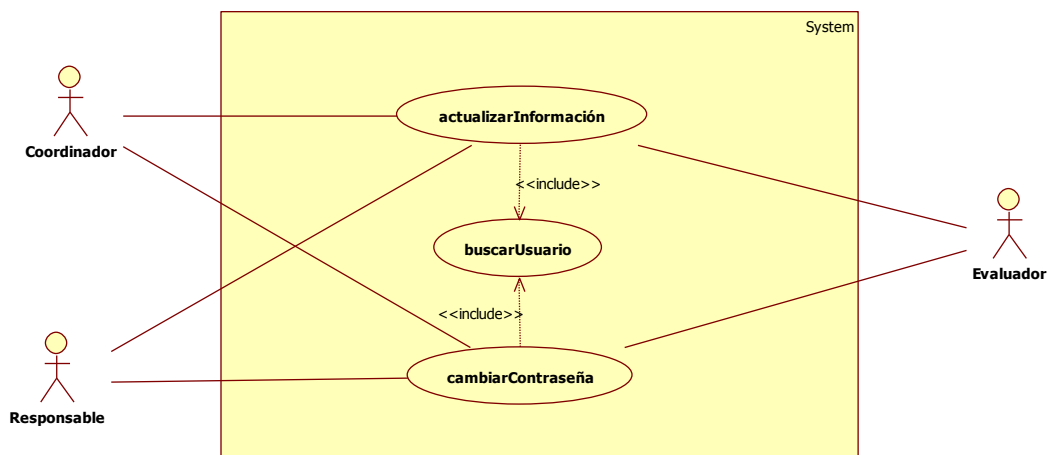


**Ilustración 13: Casos de Uso Paquete Gestión Evidencias.**

### 3.6.7 Casos de Uso del Paquete Usuarios

En la ilustración 14 muestra los casos de uso del paquete usuarios en los cuales interactúan los actores del sistema coordinador, responsable y evaluador se describen a continuación:

- Actualizar Información.
- Buscar Usuario.
- Cambiar Contraseña.



**Ilustración 14: Caso de Uso Paquete Usuarios.**

### 3.6.8 Diagrama de Clases

La ilustración 15 se muestra el diagrama de clases del sistema SIIAC-ESPE, en la cual se describe las entidades y relaciones existentes, cada uno de los campos cuenta con el tipo de dato. Aquí se obtuvo la lógica del negocio para la implantación del sistema; las principales entidades son las siguientes:

- Coordinador.
- Responsable.
- Responsable Indicador.
- Indicadores Evidencia.
- Indicador Evaluador.
- Criterio.
- Evidencia.

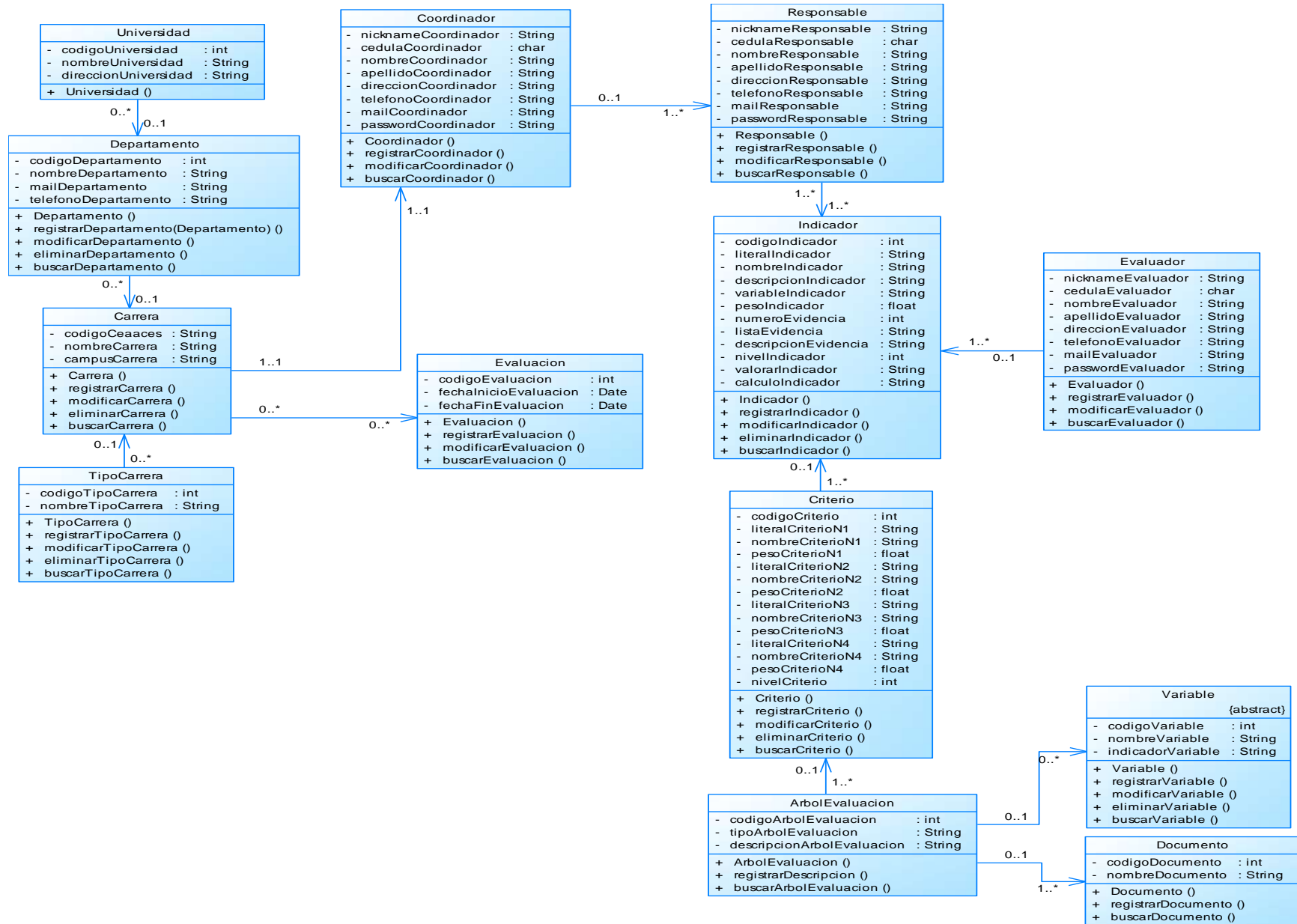


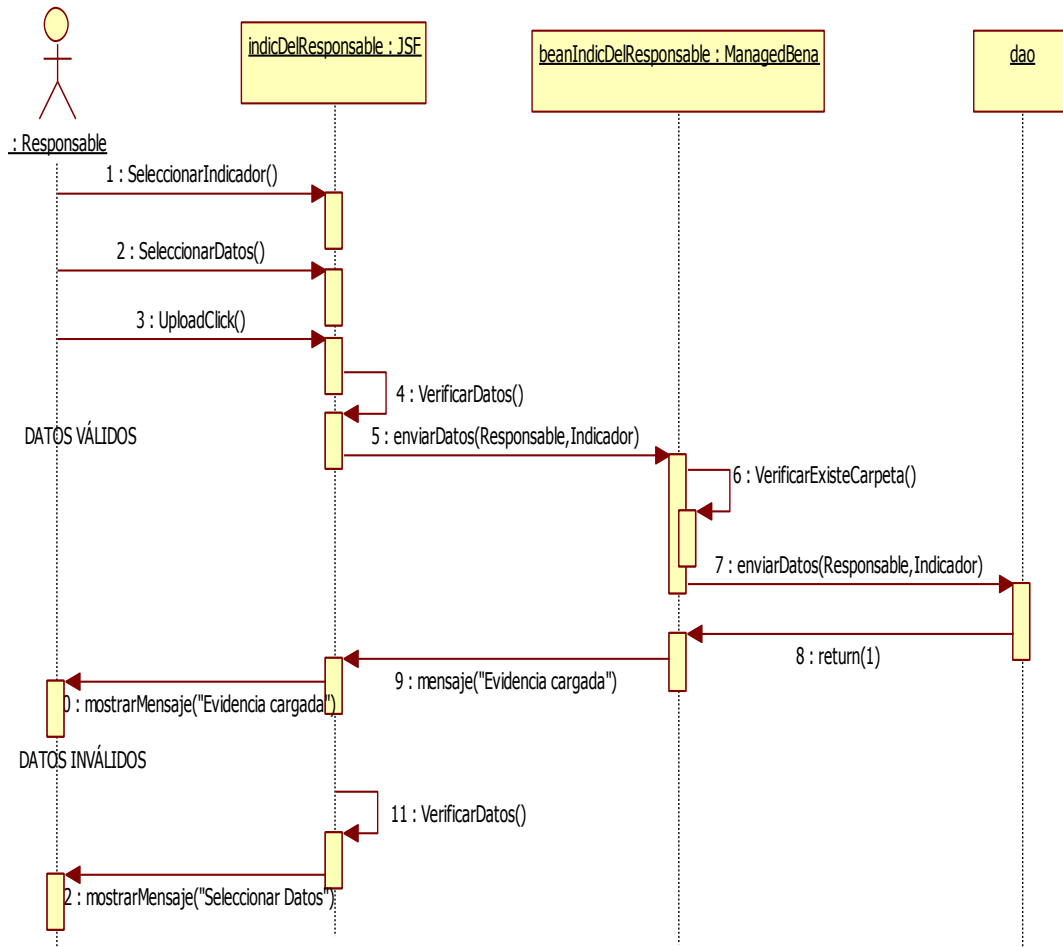
Ilustración 15: Diagrama de Clases del Sistema SIIAC-ESPE.



### 3.6.9 Diagrama de Secuencia

La ilustración 16 muestra la relación entre el actor responsable y los paquetes del sistema en el caso de uso CARGAR EVIDENCIAS. Primero el actor responsable selecciona el indicador, luego selecciona la información correspondiente como son la evidencia, la variable en caso de tenerla y selecciona el o los archivos a cargar. Luego presiona sobre el botón UPLOAD, esto se realiza en el paquete de presentación. Aquí se realiza una verificación de los datos si están vacíos o llenos. De aquí la información del responsable y el indicador pasa al paquete de negocio, donde se realiza una verificación de si existe o no la carpeta a la cual se subirá el archivo. De aquí la información de responsable e indicador pasa al paquete de datos, donde se realiza todas las acciones en la base de datos. Como respuesta positiva retorna un 1 al paquete de negocio, este retorna un mensaje al paquete de presentación y este muestra el mensaje al actor como "Evidencia cargada".

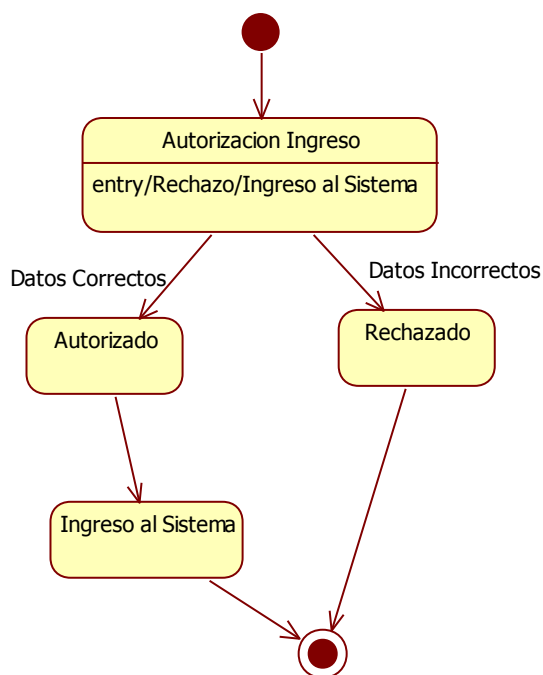
En caso de datos inválidos en el paquete de presentación este muestra un mensaje al actor como "Seleccionar datos". Si esto sucede a nivel de la capa de negocio muestra el siguiente mensaje "Carpeta no existente". Finalmente si esto ocurre a nivel de la capa de datos el mensaje a mostrar es "No se realizó la acción solicitada."



**Ilustración 16: Diagrama de Secuencia - Cargar Evidencia del Indicador.**

### 3.6.10 Diagrama de Estados

La ilustración 17 muestra el proceso de ingreso del usuario al sistema: inicia con el ingreso de los datos, la validación de datos, en caso de ser correcto un mensaje de autorizado y el ingreso al sistema, caso contrario un mensaje de rechazado.

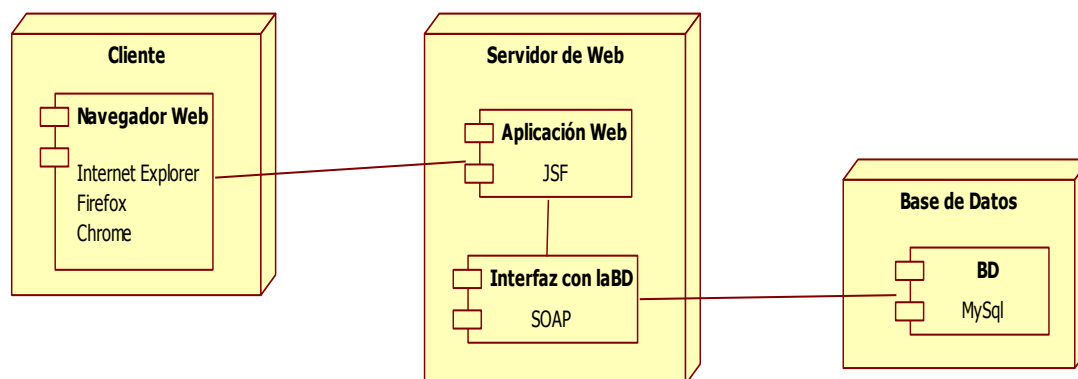


**Ilustración 17: Diagrama de Estados - Ingreso al Sistema.**

### 3.6.11 Diagrama de Despliegue

La ilustración 18 muestra la arquitectura de implementación utilizada y la interacción con los diferentes servicios y capas como son:

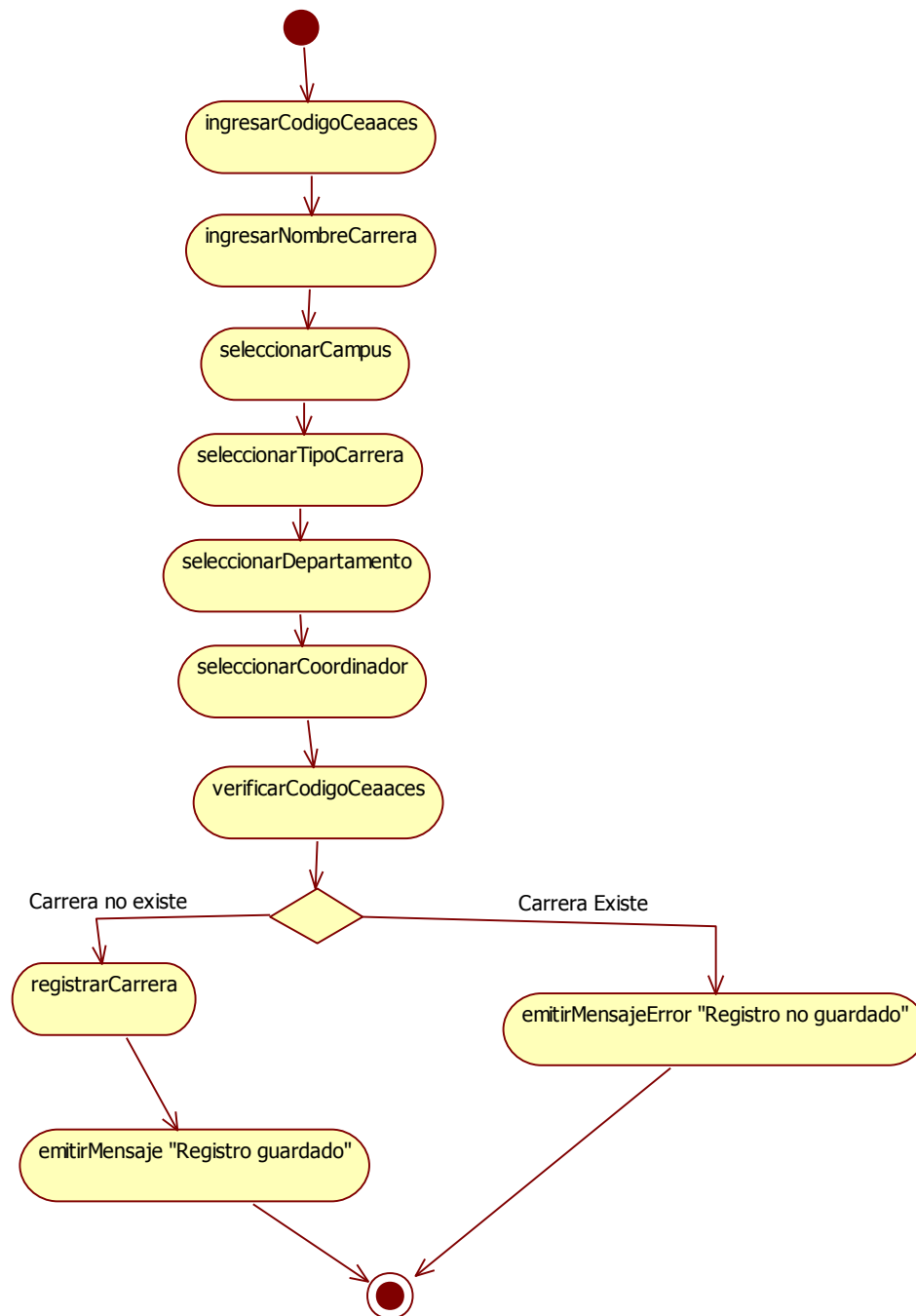
- Cliente.- usuario que puede ingresar al sistema desde cualquier navegador.
- Servidor Web.- compuesta por la Aplicación Web que es el desarrollo del sistema donde se utilizó JSF. La Interfaz con la BD que es la conexión con la BD donde se utilizó la tecnología SOAP.
- Base de Datos.- la herramienta que se utilizó para guardar los datos del sistema fue MySql 5.0.



**Ilustración 18: Diagrama de Despliegue del Sistema.**

### 3.6.12 Diagrama de Actividades

La ilustración 19 muestra el diagrama de actividades correspondiente al registro de una carrera, el administrador del sistema debe ingresar los campos correspondientes como son: código cececes, nombre carrera, seleccionar campus, seleccionar tipo carrera, seleccionar departamento, seleccionar coordinador, verificar código cececes; si la carrera no existe aparece un mensaje “registro guardado” caso contrario “registro no guardado”.



**Ilustración 19: Diagrama de Actividades - Registrar Carrera.**

## CAPÍTULO 4

### PRUEBAS

#### 4 Introducción

El presente capítulo muestra las pruebas realizadas al sistema SIIAC-ESPE. La Norma utilizada para evaluar las pruebas de software que guiaron el tratamiento del mismo fueron la Norma ISO / IEC 29119, que es una guía de buenas prácticas para la evaluación del software dentro de cualquier ciclo de desarrollo.

#### 4.1 Estándares de Implementación

En la Ilustración 20 muestra la estructura del archivo Bean este debe contener un comentario inicial, el nombre de los paquetes a utilizar, declaración de imports, e implementación de clases y librerías a utilizar.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.ViewScoped;
import daoSoap.parametrizacionServicioWeb;
import java.util.ArrayList;
import java.util.List;
import javax.annotation.PostConstruct;
import modelo.Estado;
import java.io.Serializable;
import javax.faces.application.FacesMessage;
import org.primefaces.event.SelectEvent;

@ManagedBean
@ViewScoped
public class beanEstado implements Serializable {

```

**Ilustración 20: Estructura del archivo BEAN.**

### 4.1.1 Declaración de Variables

Para la declaración de variables se debe realizar una declaración por línea, dar un valor inicial a la variable y utilizar nombres descriptivos de acuerdo al sistema que se está desarrollando como muestra en la Ilustración 21.

```
public class beanEstado implements Serializable {  
  
    parametrizacionServicioWeb parametrizacion;  
    List<Estado> calificaciones;  
    Estado calificacion;  
    Estado calificacionSeleccionada;  
    String habilitado = "true";  
  
    @PostConstruct  
    public void init() {  
        calificacion = new Estado();  
        calificacionSeleccionada = new Estado();  
        calificaciones = new ArrayList<>();  
        List<Estado> estados = parametrizacion.calificaciones();  
        Estado estado = new Estado();  
  
    }  
}
```

**Ilustración 21: Declaración de variables utilizadas.**

### 4.1.2 Sentencias

Para la creación de métodos se debe utilizar siempre corchetes y paréntesis se recomienda implementar en el desarrollo sentencias try; estas evitan excepciones generadas por la ausencia de valores en la base de datos como se muestra en la Ilustración 22.

```

public void insertarCalificacion() {
    if (!"".equals(calificacion.getCampoestado()) && calificacion.getColorestado() != null) {
        int respuesta;
        calificacion.setVisible(1);
        respuesta = parametrizacion.accionesCalificacion(1, calificacion);
        if (respuesta == 1) {
            Funciones.sendMessage("Registro guardado.", FacesMessage.SEVERITY_INFO);
            init();
        } else {
            Funciones.sendMessage("No se guardo el registro.", FacesMessage.SEVERITY_ERROR);
        }
    } else {
        Funciones.sendMessage("Ingrese datos.", FacesMessage.SEVERITY_ERROR);
    }
}
}

```

**Ilustración 22: Sentencias implementadas en el desarrollo.**


## 4.2 Pruebas de Aceptación

Para la realización de las pruebas de aceptación se contó con la presencia de los actores de cada módulo del sistema, proceso mediante el cual se valida que la aplicación cumpla con las funciones escritas en la especificación de requerimientos de software (ERS).

### 4.2.1 Pruebas de Caja Negra

En la tabla 6,7 y 8 se muestra las pruebas de caja negra realizadas al caso de uso Administrar Indicador, el cual implica el ingreso, actualización y eliminación del indicador.

**Tabla 6: Formulario de revisión de funcionalidad del caso de uso Administrar Indicador - Ingreso.**

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código Caso de Prueba:</b> PRU1	<b>Código Caso de Uso:</b> (CU006)
<b>Descripción de la Prueba:</b> La prueba consiste en ingresar todos los datos de un indicador.	
<b>Condiciones de Ejecución:</b>	
CONTINÚA 	



El usuario debe ingresar al sistema autenticado como Administrador.

**Entrada / Pasos de Ejecución:**

El actor elige la opción Información de Evaluación - Indicadores.

Seleccionar la opción Registrar.

Seleccionar datos del indicador como:

Árbol de Evaluación.

Nivel del Indicador.

Criterio al que pertenece.

Número de evidencias

Valoración.

Evaluador encargado.

Nombre de la variable.

Ingresar datos del indicador como:

Literal.

Nombre.

Descripción.

Variables.

Peso.

Despliegue de un mensaje de confirmación o de fallo.

**Resultado Esperado:**

El indicador es almacenado en la base de datos con todos los atributos y parámetros.

**Evaluación de la Prueba:**

Los resultados esperados han sido los correctos según los requerimientos de los usuarios.

**Fuente: (Autores, 2014)**

**Tabla 7: Formulario de revisión de funcionalidad del caso de uso  
Administrar Indicador - Actualizar.**

**CASO DE PRUEBA DE ACEPTACIÓN**

**Código Caso de Prueba:** PRU1

**Código Caso de Uso:** (CU006)

**Descripción de la Prueba:**

La prueba consiste en actualizar los datos de un indicador.

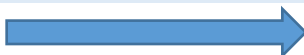
**Condiciones de Ejecución:**

El usuario debe ingresar al sistema autenticado como Administrador.

**Entrada / Pasos de Ejecución:**

El actor elige la opción Información de Evaluación-Indicadores.

CONTINÚA



Seleccionar ítem de la tabla del indicador.  
 Actualizar los campos del indicador.  
 Despliegue de un mensaje de confirmación o de fallo.

**Resultado Esperado:**

El indicador es actualizado en la base de datos con todos los atributos y parámetros necesarios para realizar la evaluación.

**Evaluación de la Prueba:**

Los resultados esperados han sido los correctos según los requerimientos de los usuarios.

**Fuente: (Autores, 2014)**

**Tabla 8: Formulario de revisión de funcionalidad del caso de uso  
 Administrar Indicador - Eliminar.**

**CASO DE PRUEBA DE ACEPTACIÓN**

**Código Caso de Prueba: PRU1**      **Código Caso de Uso: (CU006)**

**Descripción de la Prueba:**

La prueba consiste en eliminar un indicador.

**Condiciones de Ejecución:**

El usuario debe ingresar al sistema autenticado como Administrador.

**Entrada / Pasos de Ejecución:**

El actor elige la opción Información de Evaluación - Indicadores.  
 Seleccionar ítem de la tabla del indicador.  
 Eliminar el indicador.  
 Despliegue de un mensaje de confirmación o de fallo.

**Resultado Esperado:**

El indicador es eliminado de la base de datos; si este no está asociado a otro objeto.

**Evaluación de la Prueba:**

Los resultados esperados han sido los correctos según los requerimientos de los usuarios.

**Fuente: (Autores, 2014)**

#### **4.2.2 Pruebas de Caja Blanca**

En la tabla 9 y 10 se muestra las pruebas de caja blanca realizadas a las siguientes acciones: Buscar Carrera y Eliminar Criterio. Aquí se comprobarán si

los resultados que se esperan son los resultados obtenidos, después de ejecutar la acción respectiva.

**Tabla 9: Prueba de Caja Blanca - Buscar Carrera.**

CASO DE PRUEBA	RESULTADO ESPERADO	RESULTADO OBTENIDO
<b>Carrera Existente</b>	codigoCeaaces="Sist001", codigoDepartamento="1", codigoTipoCarrera="1", nombreCarrera="Ing. Sistemas e Informática", campusCarrera"Sangolquí".	codigoCeaaces="Sist001", codigoDepartamento="1", codigoTipoCarrera="1", nombreCarrera="Ing. Sistemas e Informática", campusCarrera"Sangolquí".
<b>Carrera no Existente</b>	null	null

Fuente: (Autores, 2014)

**Tabla 10: Prueba de Caja Blanca - Eliminar Criterio**

CASO DE PRUEBA	RESULTADO ESPERADO	RESULTADO OBTENIDO
<b>Criterio Existente</b>	true	true
<b>Criterio no Existente</b>	false	false

Fuente: (Autores, 2014)

### 4.2.3 Pruebas de Stress

Para la siguiente prueba de Stress se utiliza el Programa Apache Jmeter 2.11, el cual permitirá simular la interacción de 150 usuarios concurrentes para validar la disponibilidad y medir el nivel de fiabilidad del sistema ver ilustración 23,24.

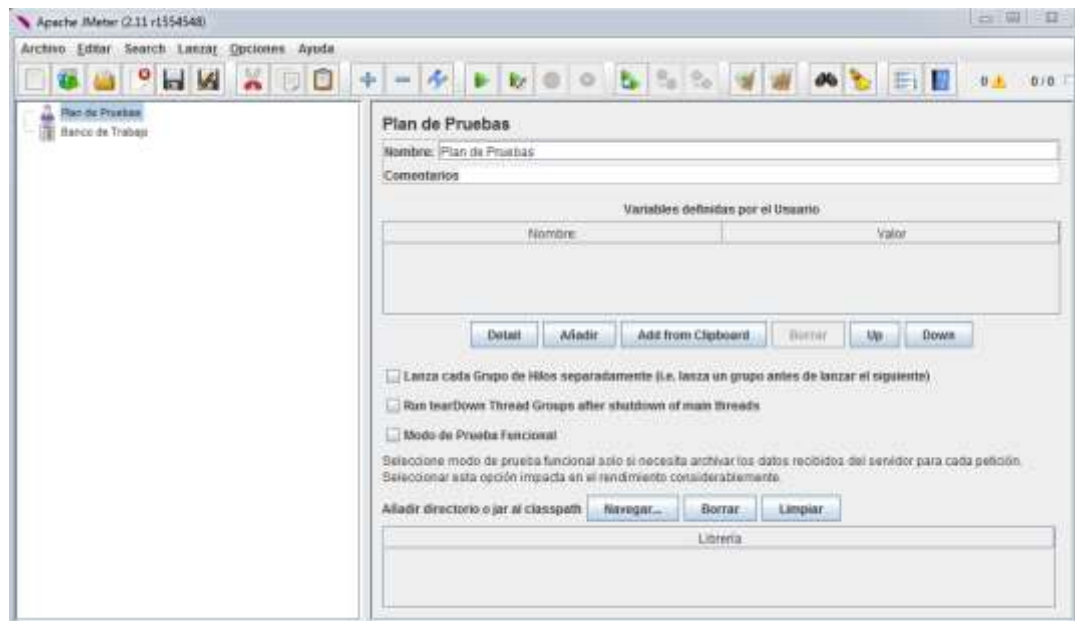


Ilustración 23: Programa Jmeter utilizado para prueba de Stress.



Ilustración 24: Configuración del Servidor de Aplicaciones Glassfish.

## **4.3 Manual de Referencias Técnicas**

### **4.3.1 Prerrequisitos**

A continuación se define los prerrequisitos técnicos para proceder con la instalación del sistema SIIAC-ESPE, adicionalmente se detalla las características técnicas del servidor y las características técnicas del cliente.

#### **i. Requisitos de Instalación Servidor**

- Glassfish 3.1.2 - <https://glassfish.java.net/es/>
- Jdk-7u25-linux-i586 -  
<http://www.oracle.com/technetwork/es/java/javase/downloads/jdk7-downloads-1880260.html>
- Base de datos MySql 5.1 o superior - <http://www.mysql.org/>

#### **ii. Características Técnicas Servidor**

- Nombre del Servidor: UFASQLUTICUDI.espe.int
- IP Servidor: 10.1.0.182
- Plataforma del Servidor: Linux i386 Intel Xeon 2.67Ghz
- Memoria RAM: 8GB
- Disco Duro: 600 Gb

#### **iii. Características Técnicas Cliente**

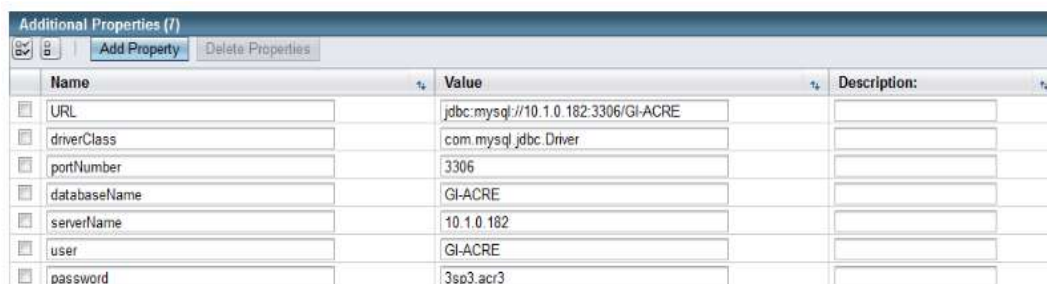
- Requisitos de Hardware
  - a. Computador Dual Core 1.5 GHz o superior.
  - b. Conexión a la intranet de la Universidad de las Fuerzas Armadas “ESPE”.

- Requisitos de Software
  - a. Sistema Operativo Windows, Linux, Mac u otros.
  - b. Navegador (Firefox, Google Chrome o Internet Explorer)

### 4.3.2 Configuración de MySQL

Para iniciar la configuración del sistema SIIAC-ESPE, se debe crear un usuario en la base de datos con todos los privilegios y permisos para poder administrar la misma se detalla la configuración y creación de la base de datos utilizada.

```
create user 'acreditacion'@'localhost' identified by 'acreditacion';
create DATABASE acreditacion GRANT ALL PRIVILEGES ON acreditacion.* TO
'acreditacion'@'localhost' IDENTIFIED BY 'acreditacion' WITH GRANT
OPTION;
```





Name	Value	Description:
URL	jdbc:mysql://10.1.0.182:3306/GI-ACRE	
driverClass	com.mysql.jdbc.Driver	
portNumber	3306	
databaseName	GI-ACRE	
serverName	10.1.0.182	
user	GI-ACRE	
password	3sp3.acr3	

**Ilustración 25: Configuración Acceso Base de Datos MySQL.**

### 4.3.3 Configuración de Glassfish

Para configurar el servidor de Aplicaciones Glassfish se utilizó el puerto por defecto 4848, luego de la instalación se procede a ingresar mediante la url <https://10.1.0.182:4848/common/index.jsf>, para cargar el archivo fuente ubicado en la ruta /opt/Acreditacion/Fuentes/prj\_espeAcreditacion/dist como muestra la ilustración 25 y 26.

/opt/Acreditacion/Fuentes/prj_espeAcreditacion/dist					
Nombre	Ext	Tamaño	Modificado	Permisos	Propiet..
			04/04/2014 12:41:55	rwxr-xr-x	root
 prj_espeAcreditacion.war		7.547 KiB	24/03/2014 15:51:52	rw-r--r--	root

**Ilustración 26: Archivo Fuente de la Aplicación SIIAC-ESPE.**

Name: prj\_espeAcreditacion

Status:  Enabled

Virtual Servers:

Associates an Internet domain name with a physical server.

Context Root: /prj\_espeAcreditacion  
Path relative to server's base URL.

Description:

Location: \${cm.sun.aas.instanceRootURL}/applications/prj\_espeAcreditacion/

Libraries:

Modules and Components (6)					
Module Name	Engines	Component Name	Type	Action	
prj_espeAcreditacion	[jpa, web, webservices]	-----	-----	Launch	
prj_espeAcreditacion		SenicioUsuario	Servlet	View Endpoint	
prj_espeAcreditacion		Faces Servlet	Servlet		
prj_espeAcreditacion		default	Servlet		
prj_espeAcreditacion		jsp	Servlet		
prj_espeAcreditacion		SenicioParametrizacion	Servlet	View Endpoint	

**Ilustración 27: Carga Archivo /prj\_espeAcreditacion .War.**

## CAPÍTULO 5

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones

- Con el sistema SIIAC-ESPE la Universidad de las Fuerzas Armadas “ESPE” logró satisfacer los requerimientos para la gestión de calidad de las evidencias; con el cual los usuarios involucrados en el proceso de evaluación y acreditación pueden tener fácil acceso al sistema y disponibilidad en cualquier momento.
- El sistema ha permitido automatizar el proceso evaluación y acreditación de los programas y carreras, reduciendo las actividades a los gestores de la acreditación, brindando un servicio fiable para la parametrización de las matrices de evaluación.
- El levantamiento de información y el análisis de requerimientos fueron las etapas más arduas del proyecto y las que más tiempo tomaron, se realizó reuniones con cada uno de los gestores de la acreditación para unificar el proceso y conceptos de la evaluación.
- Los resultados que se obtuvieron son los esperados por los usuarios finales como son: Administrador, Coordinador, Responsable y Evaluador. Logrando unificar esfuerzos y enfoques sobre la evaluación se estandarizó las matrices de evaluación para la correcta parametrización de las mismas.
- El método de desarrollo UWE, sirvió para la ejecución de cada una de las fases del ciclo de desarrollo de software; permitió siempre el enfoque de



una aplicación Web fiable y disponible para los gestores de la evaluación y acreditación.

## 5.2 Recomendaciones

- Se recomienda que el encargado de administrar el sistema SIIAC-ESPE tenga el perfil y experiencia en el manejo de aplicaciones Web, además se debería realizar una capacitación antes de utilizar el sistema; para que pueda ser manejado de manera eficiente y eficaz.
- Se garantizará el producto de software siempre y cuando se realice manteniendo preventivo tanto de la base de datos como la aplicación; se recomienda realizar trimestralmente este tipo de mantenimiento.
- En caso de tener dudas acerca de la manipulación del sistema se recomienda acudir al Manual de Usuario, donde encontrará seccionado las funcionalidades dependiendo del tipo de usuario.
- Se recomienda que los únicos usuarios que deben ingresar en el sistema sean aquellos involucrados en el proceso de evaluación y acreditación; evitar entregar los usuarios y claves de acceso a personas no involucradas en este proceso.
- En caso de existir dudas acerca de los ingresos al sistema; acudir a la entidad Proceso de la base de datos, en la cual se guarda: autor, fecha y descripción es un modelo de auditoría para revisar y monitorear, lo que los usuarios realizan en la base de datos.
- Una vez finalizada la acreditación y evaluación de un periodo determinado se recomienda realizar un backup de la base de datos; este mantenimiento debe realizarlo el personal encargado de las bases de datos, de la Unidad de Tecnologías de la Información (UTIC'S).

## BIBLIOGRAFÍA Y WEBGRAFÍA

- Almeida, P. (2013). *Aplicaciones Distribuidas II*. Quito: Clases Pensúm Espe Ingeniería en Sistemas e Informática.
- Autores, L. (2014) Arévalo Katherine, Matango René: SISTEMA WEB DE GESTIÓN DE CALIDAD DE LOS PROGRAMAS DE POSTGRADOS, MEDIANTE LA UTILIZACIÓN DE LA PLATAFORMA JAVA ENTERPRISE EDITION JEE6 WEB APLICANDO LA METODOLOGÍA UWE .
- Byous, J. (2005). *Java technology: The early years*. California: Sun Developer Network.
- Domínguez-Dorado, M. (2005). *Todo Programación*. Madrid: Iberprensa (Madrid).
- Ecuador, C. N. (2008). *Evaluación de Desempeño Institucional de las Universidades y Escuelas Politécnicas del Ecuador*. Quito: Mandato Constituyente N.-14.
- Mentoring, G. (12 de Julio de 2012). *Global Mentoring El Blog*. Obtenido de <http://globalmentoring.com.mx/cursos-java/java-empresarial/arquitectura-multicapas/>
- Mora, S. L. (2001). *Programación de servidores web con CGI, SSI e IDC*. España: Club Universitario.

Perdita Stevens, R. P. (2002). *Utilización de UML en Ingeniería del Software con Objetos y Componentes*.

Rossi G., S. D. (2000). *Web Applications Models are More than Conceptual*. In Proceedings of the Web Engineering Workshop at WWWCM'99.

StartUML. (19 de Agosto de 2006). *Black Byte*. Obtenido de <http://black-byte.com/review/staruml/>

Tuya, J. (2011). *ISO/IEC 29119 Software Testing*. Obtenido de <http://www.javiergarzas.com/wp-content/uploads/2012/01/iso29119.gif>

Wheeler, D. A. (2001). *Red Hat Linux*. California: SourceForge.

*Wikipedia*. (06 de Marzo de 2013). Obtenido de <http://es.wikipedia.org/wiki/NetBeans>

*Wikipedia*. (23 de Mayo de 2013). *Wikipedia*. Obtenido de <http://es.wikipedia.org/wiki/GlassFish>