

**DESARROLLO DEL MÓDULO DE PERSONALIZACIÓN PARA LOS PORTALES  
UNAB E IMPLEMENTACIÓN MEDIANTE MÓDULO DE EVALUACIÓN  
DOCENTE**

**ANA SLENDY CASTELLANOS GÓMEZ**

**MARIA CLARA PARDO COBOS**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
ESCUELA DE CIENCIAS NATURALES E INGENIERIA  
FACULTAD DE INGENIERIA DE SISTEMAS  
BUCARAMANGA**

**2004**

**DESARROLLO DEL MÓDULO DE PERSONALIZACIÓN PARA LOS PORTALES  
UNAB E IMPLEMENTACIÓN MEDIANTE MÓDULO DE EVALUACIÓN  
DOCENTE**

**ANA SLENDY CASTELLANOS GÓMEZ  
MARÍA CLARA PARDO COBOS**

**Tesis para optar al título de Ingeniero de Sistemas**

**Director  
HUGO VECINO PICO  
Ingeniero de Sistemas**

**Vo. Bo. \_\_\_\_\_**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
ESCUELA DE CIENCIAS NATURALES E INGENIERIA  
FACULTAD DE INGENIERIA DE SISTEMAS  
BUCARAMANGA, SANTANDER**

**2004**

Nota de aceptación

---

---

---

---

Presidente del jurado

---

Jurado

---

Jurado

Bucaramanga, Julio de 2004

## **AGRADECIMIENTOS**

A nuestro director de tesis, Ing. Hugo Vecino Pico ser el gestor de la idea.

A nuestra evaluadora de proyecto la Ing. Sandra Johanna Moreno por su tiempo, interés y orientación ofrecidos desde un comienzo y durante toda la evolución del proyecto.

Al MCC. Daniel Arenas Seleey por la valiosa ayuda que nos prestó en la comprensión del proyecto SIGPEED y en la adquisición y uso de las herramientas Oracle8i y Designer6i.

A nuestros padres que con su apoyo moral y económico nos han impulsado buscando siempre un mejor futuro para sus hijas.

A la Universidad Autónoma de Bucaramanga por proporcionarnos los recursos necesarios y el apoyo de sus docentes.

A Dios, quien a pesar de todas las adversidades se ha manifestado de las formas más maravillosas, y ha estado presente en todo momento. A mi familia e hijo motores de mi vida y motivo de mi lucha y esfuerzo.

***María Clara***

## CONTENIDO

	pág .
<b>RESUMEN</b>	
<b>OBJETIVOS</b>	
<b>1. ASPECTOS GENERALES</b>	<b>13</b>
<b>1.1. ANTECEDENTES</b>	<b>13</b>
<b>1.2. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN</b>	<b>15</b>
<b>2. MARCO TEORICO</b>	<b>16</b>
<b>2.1 PORTALES</b>	<b>16</b>
<b>2.2 INTRANET</b>	<b>18</b>
<b>2.3 PERSONALIZACIÓN</b>	<b>18</b>
2.3.1 Creación De Perfiles.	19
2.3.2 Perfil De Usuario.	19
<b>2.4 CONCEPTOS Y PRINCIPIOS ORIENTADOS A OBJETOS</b>	<b>21</b>
2.4.1 Clases Y Objetos.	22
2.4.2 Atributos.	22
2.4.3 Operaciones, Métodos Y Servicios.	23
2.4.4 Mensajes.	23
2.4.5 Encapsulamiento, Herencia Y Polimorfismo.	23
<b>2.5 APLICACIÓN DEL LENGUAJE UML Y DEL ANÁLISIS Y EL DISEÑO ORIENTADO A OBJETOS</b>	<b>24</b>
2.5.1 El Unified Modeling Language, Uml.	24
2.5.2 Un Proceso Dirigido Por Casos De Uso.	25
2.5.2.1 Ventajas De Los Casos De Uso.	25
2.5.2.2 La Captura De Casos De Uso.	28
2.5.2.3 El Modelo De Casos De Uso Representa Los Requisitos Funcionales.	28
2.5.2.4 Los Actores Son El Entorno Del Sistema.	29

2.5.3 Un Proceso Centrado En La Arquitectura.	30
2.5.4 Un Proceso Iterativo E Incremental.	30
2.5.4.1 Desarrollo En Pequeños Pasos.	30
2.5.5 Diagrama De Secuencia.	31
2.5.6 Diagrama De Colaboración.	32
2.5.7 Diagrama De Clases.	33
<b>2.6 JAVA</b>	<b>33</b>
2.6.1 Servlets	34
2.7.1.1 Características de los Servlets.	34
2.6.2 JAVABEANS	35
2.6.3 JAVA DATA BASE CONNECTION (JDBC)	36
<b>2.7 JAVA SERVER PAGE (JSP)</b>	<b>37</b>
<b>2.8 BASES DE DATOS</b>	<b>37</b>
2.8.1 Definición De Base De Datos.	37
2.8.2 Componentes Principales.	38
2.8.3 El Sistema Organizador De Base De Datos (DBMS).	38
2.8.4 Ventajas En El Uso De Bases De Datos.	39
<b>2.9 ORACLE 8I</b>	<b>40</b>
2.9.1 Arquitectura.	41
2.9.2 Estructuras Internas De La Base De Datos.	41
2.9.2.1 Restricciones De Tablas.	42
2.9.2.2 Usuarios.	42
<b>3. DESARROLLO GENERAL DEL SISTEMA</b>	<b>43</b>
<b>3.1 FASE 1 ESTUDIO Y APRENDIZAJE</b>	<b>43</b>
<b>3.2 SEGUNDA FASE: ANÁLISIS ORIENTADO A OBJETOS</b>	<b>45</b>
3.2.1 Objetivos Del Análisis.	45
3.2.1.1 Actividades Desarrolladas.	46
<b>3.3 TERCERA FASE: DISEÑO ORIENTADO A OBJETOS</b>	<b>56</b>
<b>3.4 CUARTA FASE: IMPLEMENTACIÓN</b>	<b>58</b>
<b>3.5 QUINTA FASE: PRUEBAS</b>	<b>58</b>

<b>4 LOGROS, CONCLUSIONES Y SUGERENCIAS</b>	<b>59</b>
<b>4.1 LOGROS</b>	<b>59</b>
<b>4.2 CONCLUSIONES</b>	<b>60</b>
<b>4.3 SUGERENCIAS</b>	<b>60</b>
<b>BIBLIOGRAFÍA</b>	<b>62</b>

## LISTA DE FIGURAS

	<b>pág.</b>
Figura 1, Historia de UML	25
Figura 2, Ejemplo De Un Diagrama De Caso De Uso	29
Figura 3, Ejemplo Diagrama De Secuencia	31
Figura 4, Ejemplo Diagrama de Colaboración	32
Figura 5, Ejemplo Diagrama De Clases	33
Figura 6, Caso de uso Evaluación Docente y Personalización	50
Figura 7, Diagrama De Clases Personalización	51
Figura 8, Diagrama De Clases Evaluación Docente	52

## LISTA DE ANEXOS

	<b>pág.</b>
ANEXO A. DER del sistema SIGPEED	66
ANEXO B. DER actualizado de la Evaluación Docente	67
ANEXO C. Diagrama de análisis de la Evaluación Docente	68
ANEXO D. Diagramas de Secuencia de la Personalización	70
ANEXO E. Diagramas de Secuencia de la Evaluación Docente.	72
ANEXO F. Diagramas de colaboración de la Personalización.	75
ANEXO G. Diagramas de colaboración de la Evaluación Docente.	78
ANEXO H. Diccionario de datos de la Evaluación Docente.	81
ANEXO I. DER del sistema personalización	89
ANEXO J. Resumen de reuniones	90

## RESUMEN

Con el desarrollo de esta tesis, buscamos crear un módulo de personalización para implementarlo en los portales estudiante y administrativo de la UNAB, los cuales se encuentran en producción, ambos desarrollados por el Departamento de Sistemas de la Universidad Autónoma de Bucaramanga, para que de esta forma se facilite la interacción del usuario con el sitio y además delimite los alcances determinados por la institución para los respectivos usuarios de los portales mediante el uso de reglas de negocio y así evitar accesos no permitidos; información que será canalizada y de interés exclusivo para cada tipo de perfil previamente establecido.

Como prueba de este módulo de personalización se implementará la evaluación docente institucional aplicable a cada una de las facultades a través de la Intranet y a los estudiantes a través del portal del estudiante, para satisfacer una necesidad con la que cuenta en este momento la Universidad Autónoma de Bucaramanga. De esta manera le permitiremos a la UNAB evaluar a sus correspondientes docentes por medio de la Heteroevaluación, para que dicha información muestre su desempeño laboral, desde el punto de vista del estudiante.

Esta herramienta se desarrolló para trabajar en una arquitectura de 3 capas (cliente, aplicación y base de datos), permitiendo a los estudiantes evaluar a los docentes; decanos, acceder a los reportes de la evaluación; administrador, mantenimiento de la aplicación.

Ya que la Universidad ha venido trabajando en la estandarización de los portales UNAB, se tuvieron en cuenta las diferentes plataformas utilizadas en el Departamento de Sistemas de la UNAB, las cuales son: ORACLE como motor de Bases de Datos, Java Server Page (JSP) como interfaz Web en el cliente y Programación Orientada a Objetos usando JAVA.

## OBJETIVOS

### OBJETIVO GENERAL:

Implementar y desarrollar el módulo correspondiente a perfiles para los usuarios de los portales UNAB, usando una metodología RUP (Rational Unified Procces); con la intención de lograr un nivel de personalización en estos portales, y adaptar la evaluación docente al portal para que nos sirva de aplicación y prueba del módulo perfiles a todas las facultades.

### OBJETIVOS ESPECIFICOS:

- Desarrollar el modulo para la creación de perfiles de usuario para los portales UNAB, mediante el análisis de reglas de negocio para la personalización de las opciones a presentar en los mismos.
- Implementar el módulo de evaluación docente para obtener valoraciones, en los contextos de Heteroevaluación del profesorado de la UNAB mediante herramientas ya preestablecidas por la universidad, y así mismo nos sirva de aplicación para el módulo de perfiles.
- Realizar un diseño para el modelamiento del servicio presentado anteriormente utilizando UML para la definición, organización, visualización y construcción de este.
- Se debe entregar funcionando el módulo de la Evaluación Docente.

## **1. ASPECTOS GENERALES**

### **1.1 ANTECEDENTES**

A través del tiempo la Universidad Autónoma de Bucaramanga siempre ha estado en la búsqueda de mejorar su calidad formativa y esto se puede lograr por medio de la evaluación y supervisión de la labor docente para comprobar el cumplimiento de las obligaciones docentes del profesorado y requerir las responsabilidades que se deriven del incumplimiento de las mismas, de esta forma conocer el estado de la calidad de docencia y así utilizar la información resultado de este proceso para el mejoramiento continuo de esta.

En un principio estas evaluaciones se realizaban manualmente y después se construyó para las facultades de ingenierías una herramienta que facilitara este proceso.

Con el tiempo se desarrolló un proyecto de grado llamado SIGPEED\* (Sistema de información para la Gestión del Portafolio de Empleados y la Evaluación Docente en la UNAB), el cual tuvo en cuenta además de la heteroevaluación dos de los tres contextos en los que se debe evaluar, que son la coevaluación y la autoevaluación.

Debido a la imperante necesidad de implementar una evaluación docente que cubra las necesidades de cada una de las facultades de la UNAB, se cree conveniente retomar la heteroevaluación del proyecto SIGPEED para situarla en los portales de la Universidad que se encuentran en producción.

---

\* Sistema de Información para la Gestión del Portafolio de Empleados y la Evaluación docente en la UNAB.

La Universidad Autónoma de Bucaramanga siempre se ha esforzado por la actualización e inclusión de nuevas tecnologías en sus quehaceres organizacionales, para permitir que siga siendo un interlocutor universitario reconocido en los ámbitos nacional e internacional, el departamento de sistemas aplicó un estándar para realizar nuevos portales, ya que los actuales han sido desarrollados y mantenidos por diferentes dependencias, provocando incongruencia en la información y gastos innecesarios de recursos. Poco a poco se han ido desarrollando estos nuevos portales; y ahora se quiere implementar el servicio de la definición de los perfiles de usuarios de estos.

## **1.2. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN**

Debido a que los portales UNAB tendrán una gran variedad de usuarios y administradores con preferencias y necesidades diferentes, surge la necesidad de diseñar un perfil para cada uno de estos, mediante técnicas de personalización y el análisis de reglas de negocio que son “todas las subrutinas creadas con el propósito de regular alguna acción del usuario” [1], las cuales nos permitan identificarlos y definirlos de tal modo que al ingresar al portal cuenten con un sitio fácil de manejar y con cualidades adaptativas para sus respectivos requerimientos.

A su vez la evaluación docente herramienta fundamental en toda Institución Universitaria, ya que el fracaso o triunfo de todo sistema educativo esta basado principalmente en la calidad del desempeño docente, y esta no se encuentra debidamente implementada en la Universidad Autónoma de Bucaramanga, se desarrolló una evaluación docente aplicable para toda la UNAB, teniendo como modelo el módulo de heteroevaluación del proyecto SIGPEED.

## 2. MARCO TEORICO

### 2.1. PORTALES WEB

Un Portal se entiende como: “la página Web que agrega contenidos y funcionalidades, organizados de tal manera que facilitan la navegación y proporcionan al usuario un punto de entrada en la Red con un amplio abanico de opciones” [2]. En ese punto de entrada, el usuario ve concentrados todos los servicios y productos que ofrece, de forma que le permite hacer cuanto necesita sin tener que salir de dicho sitio Web.

Entre los varios usos y funciones que los portales pueden tener se incluyen:

- Conectar diferentes clases de información;
- Proveer acceso a información sobre administración de negocios;
- Facilitar compras al consumidor;
- Brindar capacitación a empleados;
- Elaborar listas y vínculos a recursos, servicios y productos;
- Apoyar la comunicación entre los miembros de una organización en Intranet; y
- Servir como recurso para asesoría especializada, decisiones, opiniones y soporte.[3]

Aunque gran parte de los portales no utilizarán todas estas funciones, aquellos que emplean una combinación de ellos son llamados Portales de Conocimiento.

“La principal función de un Portal de Conocimiento es obtener información específica utilizando un sistema de tecnología de la información y presentarla sobre la base de las preferencias y perfiles del usuario en particular; posibilitar características de personalización y navegación que permitirán al usuario adaptar

sus búsquedas para su mayor beneficio; y facilitar la comunicación y colaboración entre aquellos que disponen de la información y aquellos que la necesitan” [4].

El principal objetivo de la implementación de un portal es crear un ámbito amigable para que el usuario pueda navegar fácilmente y encontrar la información que necesita para realizar sus tareas más rápidamente, tomar decisiones, o utilizar la información obtenida.

Una característica adicional que hace de los portales un recurso valioso es su adaptabilidad a usuarios individuales. De hecho, a través del uso de técnicas de personalización, los portales han evolucionado para entregar precisamente la información específica que se necesita y de la manera en que se la necesita. Por ejemplo, un portal puede proveer de servicios tales como reportes meteorológicos locales o listados de películas con sólo recordar el código postal del usuario.

Información pertinente acerca del portal es obtenida usualmente a través de un proceso de registro por medio del cual el usuario ingresa información personal a un banco de datos. Esta información es usualmente utilizada para "personalizar" las búsquedas que el usuario realiza, o para notificarlo de novedades relevantes basado en el perfil que el usuario ha provisto previamente.

Un portal también puede convertirse en una comunidad virtual a través del uso del e-mail y los foros de Chat. Esto permite a los usuarios de un mismo portal intercambiar ideas acerca del contenido del portal y sus temas sin importar donde están localizados.

## **2.2 INTRANET**

“Intranet es la herramienta que posibilita acumular, administrar, clasificar y presentar en forma amigable la comunicación e información interna de una organización” [5].

Una Intranet es una red empresarial de computadores que nos permite acumular, administrar, clasificar y presentar en forma amigable la comunicación e información interna de una organización desarrollada con tecnologías de Internet, tales como el navegador que permite consultar páginas Web o el programa gestor de correo electrónico; a la vez que utiliza los mismos protocolos y estándares abiertos que permiten que computadores de diferentes tipos y fabricantes se puedan comunicar entre ellos.

## **2.3 PERSONALIZACIÓN DE SITIOS WEB**

Personalización es el conjunto de procesos en el que un sistema provee servicios basados en la información personal de cada usuario. Un componente de personalización debe ser capaz de recomendar documentos, promover productos, aconsejar apropiadamente, etc.

“La personalización ha pasado por varias fases. Inicialmente la personalización fue usada para mantener al visitante en el sitio, explorando más en este, lo que proveía oportunidades de promocionar productos” [6].

La siguiente fase apuntaba a incrementar cuánta plata un visitante gastaba en cada visita ofreciéndole productos relacionados más costosos. Ahora la personalización es cada vez más utilizada como medio de agilizar la entrega de información al usuario, convirtiendo al sistema en más útil y atractivo, de forma que el usuario se sienta estimulado al utilizarlo. Por eso, la personalización se viene convirtiendo en un requisito esencial en los sistemas de hoy en día, y en

particular en los sitios Web de comercio electrónico. La personalización en comercio electrónico permite ofertas de productos, ventas promocionales, publicidad, etc. dirigidas individualmente al usuario, tomando en consideración su historial de navegación, compras y otras interacciones con la tienda electrónica.

**2.3.1 Creación de Perfiles.** “Hay tres métodos principales para crear perfiles: el método **explícito** o manual; el método **colaborativo** o de composición a partir de otros perfiles; por último, el método **implícito**, que utiliza técnicas específicas para extraer las características automáticamente.

En el método explícito o de creación manual, los datos son introducidos por el usuario escribiéndolos directamente en su perfil de usuario, respondiendo a formularios, etc. Este método tiene la ventaja de permitirle al cliente decirle al sitio directamente que quieren ellos ver. Un ejemplo es Myyahoo donde el visitante es cuestionado para especificar la información del perfil, incluyendo, por ejemplo que almacenes rastrea y que categorías nuevas reporta.

También se puede crear y modificar un perfil a partir de la interacción colaborativa con otros perfiles, con los que se relaciona, recurriendo a conocimiento específico del dominio.

En el método implícito, se extraen/crean y modifican/actualizan automáticamente los perfiles, recurriendo normalmente a técnicas de Inteligencia Artificial para realizar estas tareas. Es importante señalar que estos métodos muchas veces se utilizan simultáneamente (métodos híbridos), para producir perfiles más precisos y comprensibles” [7].

**2.3.2 Perfil de Usuario.** Un perfil de usuario es un modelo de usuario creado en la memoria del computador, que es utilizado como representante del usuario en las tareas computacionales. El perfil de usuario es el perfil más importante y es

utilizado en muchas situaciones, entre ellas la personalización de aplicaciones y servicios, en la automatización de tareas, etc.

**¿Que contiene un Perfil de Usuario?** El perfil de usuario contiene información modelada sobre el usuario, representada explícita o implícitamente, y cuya explotación permite al sistema incrementar la calidad de sus adaptaciones.

Con relación al dominio, un perfil se compone de las siguientes partes: una parte **independiente del dominio**, como son el nombre, la edad, el nivel educativo, etc.; otra **dependiente del dominio**, como por ejemplo los intereses del usuario, sus conocimientos en una determinada área, etc.; y una tercera, que no cae en la estricta definición de ninguna de las dos anteriores, como son las preferencias del usuario, el ambiente, etc.

Otra forma de clasificar el perfil de usuario, es considerar su comportamiento con relación al cambio. Alguna información del perfil de usuario es **estática**, como la fecha de nacimiento, el nombre, etc.; y normalmente es introducida manualmente, de una sola vez por el usuario. Otra sin embargo es **dinámica**, como por ejemplo los intereses del usuario, que cambian y por consiguiente es aconsejable que sean determinados automáticamente. Esto significa que para obtener un perfil más actual y preciso, es necesario acompañar las acciones del usuario de la forma más cercana posible. Por eso se recoge, procesa y guarda información de las acciones del usuario, que sirve para, entre otras cosas, determinar que perfiles de otros componentes del sistema interactúan con el perfil actual, así como para proceder a las depuraciones y actualizaciones que se tengan que realizar. También hay otro conjunto de aspectos que pueden condicionar este proceso, como son: los objetivos del *perfil* de usuario, el dominio de aplicación, etc. Esto refuerza la necesidad de emplear técnicas que automaticen de forma inteligente las tareas de creación y gestión de los perfiles de usuario.

## **2.4 CONCEPTOS Y PRINCIPIOS ORIENTADOS A OBJETOS**

“Vivimos en un mundo de objetos. Estos objetos existen en naturaleza, en entidades hechas por el hombre, en los negocios y en los productos que usamos. Pueden ser clasificados, organizados, combinados, manipulados y creados. Por esto no es sorprendente que se proponga una visión orientada a objeto en la producción del software, una abstracción que modela el mundo de forma tal que nos ayuda a entenderlo y gobernarlo mejor. La primera vez que se propuso un enfoque orientado a objetos para el desarrollo de software fue a finales de los años sesenta. Sin embargo, las tecnologías de objetos han necesitado casi veinte años para llegar a ser ampliamente usadas. Durante los años 90, la ingeniería del software orientada a objetos se convirtió en el paradigma de elección para muchos productores de software y para un creciente número de sistemas de información y profesionales de la ingeniería. A medida que pasa el tiempo, las tecnologías de objetos están sustituyendo a los enfoques clásicos de desarrollo de software. Las tecnologías de objetos llevan a reutilizar, y la reutilización lleva a un desarrollo de software más rápido y a programas de mejor calidad. El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente poco acoplada. Esto lleva a menores efectos colaterales cuando se deben hacer cambios y provoca menos frustración en el ingeniero del software y en el cliente. Además, los sistemas orientados a objetos son más fáciles de adaptar y más fácilmente escalables”. [8]

“Durante muchos años el término orientado a objetos (OO) se usó para referirse a un enfoque de desarrollo de software que usa uno de los lenguajes orientados a objetos” [9]. El proceso OO se mueva a través de una espiral evolutiva que comienza la comunicación con el usuario. Es aquí donde se define el dominio del problema y se identifican las clases básicas del problema.

**2.4.1 Clases y objetos.** “Los conceptos fundamentales que llevan a un diseño de alta calidad son igualmente aplicables a sistemas desarrollados usando métodos convencionales y orientados a objetos. Por esta razón, un modelo OO de software de computadora debe exhibir abstracciones de datos y procedimientos que conducen a una modularidad eficaz. Una clase es un concepto OO que encapsula las abstracciones de datos y procedimientos que se requieren para describir el contenido y comportamiento de alguna entidad del mundo real” [10]. Un objeto encapsula datos (atributos) y los métodos (operaciones, métodos o servicios) que manipulan esos datos.

“Las abstracciones de datos (atributos) que describen la clase están encerradas por una << muralla >> de abstracciones procedimentales capaces de manipular los datos de alguna manera. La única forma de alcanzar los atributos es ir a través de algunos métodos que forman la muralla” [11]. Esto posibilita la ocultación de información y reduce el impacto de efectos colaterales asociados a cambios. Como estos métodos tienden a manipular un número limitado de atributos, esto es alta cohesión, y como la comunicación ocurre sólo a través de los métodos que encierra la muralla, la clase tiende a un desacoplamiento con otros elementos del sistema. Puesto de otra manera, una clase es una descripción generalizada que describe una colección de objetos similares. Por definición, todos los objetos que existen dentro de una clase heredan sus atributos y las operaciones disponibles para la manipulación de los atributos.

**2.4.2 Atributos.** Ya hemos visto que los atributos están asociados a clases y objetos, y que describen la clase o el objeto de alguna manera.

**2.4.3 Operaciones, Métodos Y Servicios.** “Un objeto encapsula datos y los algoritmos que procesan estos datos. Estos algoritmos son llamados operaciones, métodos o servicios y pueden ser vistos como módulos en un sentido

convencional. Cada una de las operaciones encapsuladas por un objeto proporciona una representación de uno de los comportamientos del objeto” [12].

**2.4.4 Mensajes.** “Los mensajes son el medio a través del cual interactúan los objetos. Usando la terminología presentada en la sección precedente, un mensaje estimula la ocurrencia de cierto comportamiento en el objeto receptor. El comportamiento se realiza cuando se ejecuta una operación“ [13].

#### **2.4.5 Encapsulamiento, Herencia Y Polimorfismo.**

- “Los detalles de implementación interna de datos y procedimientos están ocultos al mundo exterior. Esto reduce la propagación de efectos colaterales cuando ocurren cambios.
- Las estructuras de datos y las operaciones que las manipulan están mezcladas en una entidad sencilla: la clase. Esto facilita la reutilización de componentes.
- Las interfaces entre objetos encapsuladas están simplificadas. Un objeto que envía un mensaje no tiene que preocuparse de los detalles de las estructuras de datos internas en el objeto receptor, lo que simplifica la interacción y hace que el acoplamiento del sistema tienda a reducirse.

La herencia es una de las características claves entre de los sistemas OO. Una subclase **Y** hereda todos los atributos y operaciones asociadas con su superclase **X**. Esto significa que todas las estructuras de datos y algoritmos originalmente diseñados e implementados para **X** están inmediatamente disponibles para **Y**.

Cualquier cambio en los datos u operaciones contenidas dentro de una superclase es heredado inmediatamente por todas las subclases que se derivan de la superclase” [14].

## **2.5 APLICACIÓN DEL LENGUAJE UML Y DEL ANÁLISIS Y EL DISEÑO ORIENTADO A OBJETOS**

¿Qué significa contar con un sistema orientado a objetos que esté bien diseñado? El proverbio “el hábito hace al monje” se aplica perfectamente a la tecnología de objetos. El hecho de conocer un lenguaje orientado a objetos (Java, por ejemplo) y además tener acceso a una rica biblioteca (como la de java) es un primer paso necesario pero insuficiente para crear sistemas de objetos. Se requiere además analizar y diseñar un sistema desde la perspectiva de los objetos.

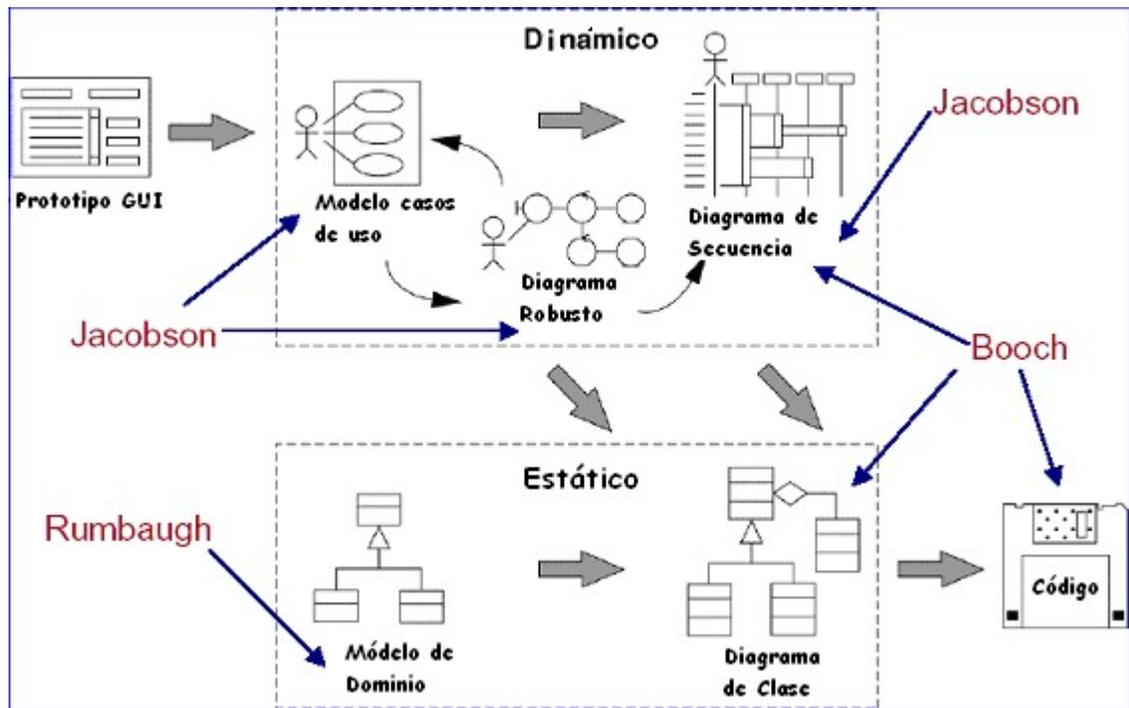
**2.5.1 El Unified Modeling Language, UML.** El UML (Lenguaje Unificado Para La Construcción de modelos) se define como un “lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software” [15]. Es un sistema notacional destinado a los sistemas de modelado que utilizan conceptos orientados a objetos.

El UML es un estándar incipiente de la industria para construir modelos orientados a objetos. Nació en 1994 por iniciativa de Grady Booch y Jim Rumbaugh para combinar sus dos famosos métodos: el de Booch y el OMT (Object Modeling Technique, Técnica de Modelado de Objetos). Más tarde se les unió Ivar Jacobson, creador del método OOSE (Object- Oriented Software Engineering, Ingeniería de software Orientada a Objetos). En respuesta a una petición de OMG para definir un lenguaje y una notación estándar del lenguaje de construcción de modelos, en 1997 propusieron el UML como candidato.

Prescindiendo de la aceptación que pueda tener, este lenguaje recibió la aprobación de facto en la industria, pues sus creadores representan métodos muy difundidos de la primera generación del análisis y diseño orientado a objetos. Muchas organizaciones dedicadas al desarrollo de software y los proveedores de herramientas CASE lo adoptaron, y muy probablemente se convertirá en el

estándar mundial que utilizarán los desarrolladores, los autores y proveedores de herramientas CASE. La Figura 1, resume la creación del UML.

Figura 1. Historia de UML



**2.5.2 Un Proceso Dirigido Por Casos De Uso.** “El objetivo del Proceso Unificado es guiar a los desarrolladores en la implementación y distribución eficiente de sistemas que se ajusten a las necesidades de los clientes. La eficiencia se mide en términos de coste, calidad, y tiempo de desarrollo. El paso desde la determinación de las necesidades del cliente hasta la implementación no es trivial. En primer lugar, las necesidades del cliente no son fáciles de discernir. Esto nos obliga a que tengamos algún modo de capturar las necesidades del usuario de forma que puedan comunicarse fácilmente a todas las personas implicadas en el proyecto. Después, debemos ser capaces de diseñar una implementación funcional que se ajuste a esas necesidades. Por último, debemos verificar que las necesidades del cliente se han cumplido mediante la prueba del

sistema. Debido a esta complejidad, el proceso se describe como una serie de flujos de trabajo que construyen el sistema gradualmente” [16].

“El proceso de desarrollo unificado está dirigido por los casos de uso, centrado en la arquitectura y es iterativo e incremental” [17].

“Los casos de uso han sido adoptados casi universalmente para la captura de requisitos de sistemas software en general, y de sistemas basados en componentes en particular, pero los casos de uso son mucho más que una herramienta para capturar requisitos. Dirigen el proceso de desarrollo en su totalidad. Los casos de uso son la entrada fundamental cuando se identifican y se especifican clases, subsistemas e interfaces. Cuando se identifican y especifican casos de prueba y cuando se planifican las iteraciones del desarrollo y la integración del sistema. Para cada iteración, nos guían a través del conjunto completo de flujos de trabajo, desde la captura de requisitos, pasando por el análisis, diseño e implementación, hasta la prueba, enlazando estos diferentes flujos de trabajo” [18].

“La captura de requisitos tiene dos objetivos: encontrar los verdaderos requisitos, y representarlos de un modo adecuado para los usuarios, clientes y desarrolladores. Entendemos por “verdaderos requisitos” aquellos que cuando se implementen añadirán el valor esperado para los usuarios con “representarlos de un modo adecuado para los usuarios, clientes y desarrolladores” queremos decir en concreto que la descripción obtenida de los requisitos debe ser comprensible por usuarios y clientes. Este es uno de los retos principales del flujo de trabajo de los requisitos.

Normalmente un sistema tiene varios tipos de usuarios. Cada tipo de usuario se representa por un actor. Los actores utilizan el sistema interactuando con los casos de uso. Un caso de uso es una secuencia de acciones que el sistema lleva

a cabo para ofrecer algún resultado de valor para un actor. El modelo de casos de uso está compuesto por todos los actores y todos los casos de uso de un sistema. El modelo de análisis es una especificación detallada de los requisitos y funciona como primera aproximación del modelo de diseño, aunque es un modelo con entidad propia. El modelo de análisis se diferencia del de diseño, en que es un modelo conceptual, en lugar de ser un esquema de la implementación. El modelo de análisis puede ser transitorio y sobrevivir solo al primer par de iteraciones. Sin embargo, en algunos casos, especialmente para sistemas grandes y complejos, el modelo de análisis debe mantenerse durante toda la vida del sistema. En estos casos, existe una relación directa (mediante dependencias de trazas) entre una realización de casos de uso en el modelo de análisis y la correspondiente realización de casos de usos en el modelo de diseño. Cada elemento del modelo de análisis es trazable a partir de elementos del modelo de diseño que lo realiza” [19].

“El modelo de diseño posee las siguientes características:

El modelo de diseño es jerárquico, pero también contiene relaciones que atraviesan la jerarquía. Las relaciones son las habituales en UML: asociaciones, generalizaciones y dependencias.

La realización de los casos de usos son estereotipos de colaboraciones. Una colaboración representa cómo los clasificadores participan y desempeñan papeles en hacer algo útil, como la realización de un caso de uso.

El modelo de diseño es también un esquema de la implementación. Existe una correspondencia directa entre subsistemas del modelo de diseño y componentes del modelo de implementación” [20].

Los Diagramas de Casos de uso:

- Muestran la granularidad del sistema en piezas de funcionalidad reutilizables.

- Muestran la interacción de los Actores con la funcionalidad del Sistema
- Organizan visualmente los requerimientos del usuario.
- Permiten certificar contractualmente la funcionalidad.
- Formalizan el mapa de procesos de negocio.

#### **2.5.2.1 Ventajas De Los Casos De Uso.**

- “Lenguaje de comunicación entre usuarios y desarrolladores.
- Comprensión detallada de la funcionalidad del Sistema.
- Acotación precisa de las habilitaciones de los usuarios.
- Trazabilidad es de los requerimientos al código ejecutable.
- Gestión de riesgo para gobernar la complejidad de un sistema.
- Planificación de iteraciones para su implementación.
- Estimación precisa del esfuerzo para su implementación.
- Documentación orientada al usuario: Manual de Procedimientos & Reglas de negocio” [21].

**2.5.2.2 La Captura De Casos De Uso.** “Durante los flujos de trabajo de los requisitos identificamos las necesidades de usuarios y clientes como requisitos. Los requisitos funcionales se expresan como casos de uso en un modelo de casos de uso, y los demás requisitos o bien se “adjuntan” a los casos de uso que afectan, o bien se guardan en una lista aparte o se describen de alguna otra forma” [22].

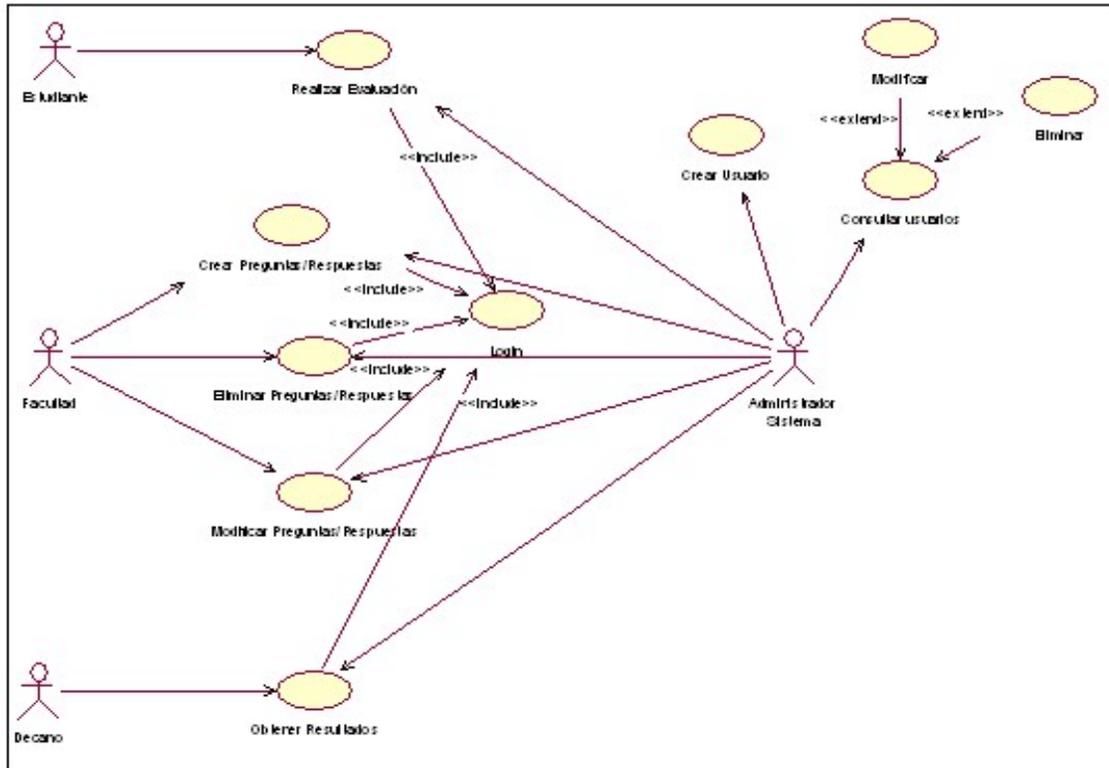
#### **2.5.2.3 El Modelo De Casos De Uso Representa Los Requisitos Funcionales.**

El modelo de casos de uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. “La mayoría de los sistemas tienen muchos tipos de usuarios. Cada tipo de usuario se representa mediante un actor. Los actores utilizan el sistema al interactuar con los casos de uso. Todos los actores y casos de uso del sistema forman un modelo de casos de uso. Un

diagrama de casos de uso describe parte del modelo de casos de uso y muestran un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan” [23].

**2.5.2.4 Los Actores Son El Entorno Del Sistema.** “No todos los actores representan a personas. Pueden ser actores otros sistemas o hardware externo que interactuará con el sistema. Cada actor asume un conjunto coherente de papeles cuando interactúa con el sistema. Un usuario físico puede actuar como uno o varios actores, desempeñando los papeles de esos actores en su interacción con el sistema. Varios usuarios concretos pueden actuar como diferentes ocurrencias del mismo actor” [24].

Figura 2. Ejemplo De Un Diagrama De Caso De Uso



**2.5.3 Un Proceso Centrado En La Arquitectura.** “La arquitectura software abarca decisiones importantes sobre:

- La organización del sistema software.
- Los elementos estructurales que compondrían el sistema y sus interfaces, junto con sus comportamientos, tal y como se especifican en las colaboraciones entre estos elementos.
- La composición de los elementos estructurales y del comportamiento en subsistemas progresivamente más grandes.
- El estilo de la arquitectura que guía esta organización: los elementos y sus interfaces, sus colaboraciones y su composición.

Sin embargo, la arquitectura software está afectada no sólo por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y compromisos económicos” [25].

**2.5.4 Un Proceso Iterativo E Incremental.** “Lleva a los desarrolladores a garantizar que el sistema se ajuste a las necesidades reales del usuario. El estar centrado en la arquitectura significa que el trabajo de desarrollo se centra en obtener el patrón de la arquitectura que dirigirá la construcción del sistema en las primeras fases, garantizando un proceso continuo no sólo para la versión en curso del producto, sino para la vida entera del mismo” [26].

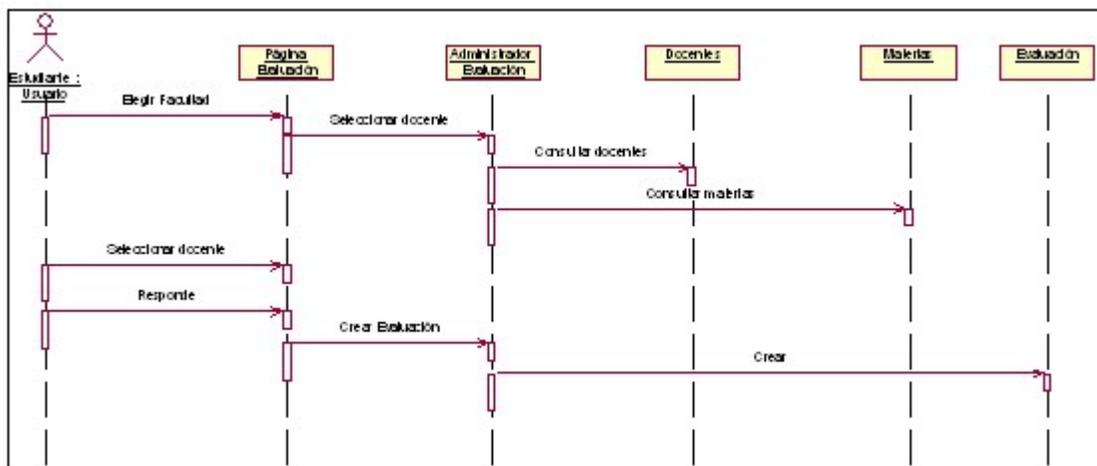
**2.5.4.1 Desarrollo En Pequeños Pasos.** “Si estamos contentos con un paso, continuamos con el siguiente. Entre cada paso, obtenemos retroalimentación que nos permite ajustar nuestros objetivos para el siguiente paso. Después se da el siguiente paso, y después otro. Cuando se han dado los pasos que habíamos planificado, tenemos un producto desarrollado que podemos distribuir a nuestros clientes y usuarios.

Las iteraciones en las primeras fases tratan en su mayor parte con la determinación del ámbito del proyecto, la eliminación de los riesgos críticos, y la creación de la línea base de la arquitectura. Después a medida que avanzamos a lo largo del proyecto y vamos reduciendo gradualmente los riesgos restantes e implementando los componentes. La forma de las iteraciones cambia, dando incrementos como resultados” [27].

### 2.5.5 Diagrama De Secuencia.

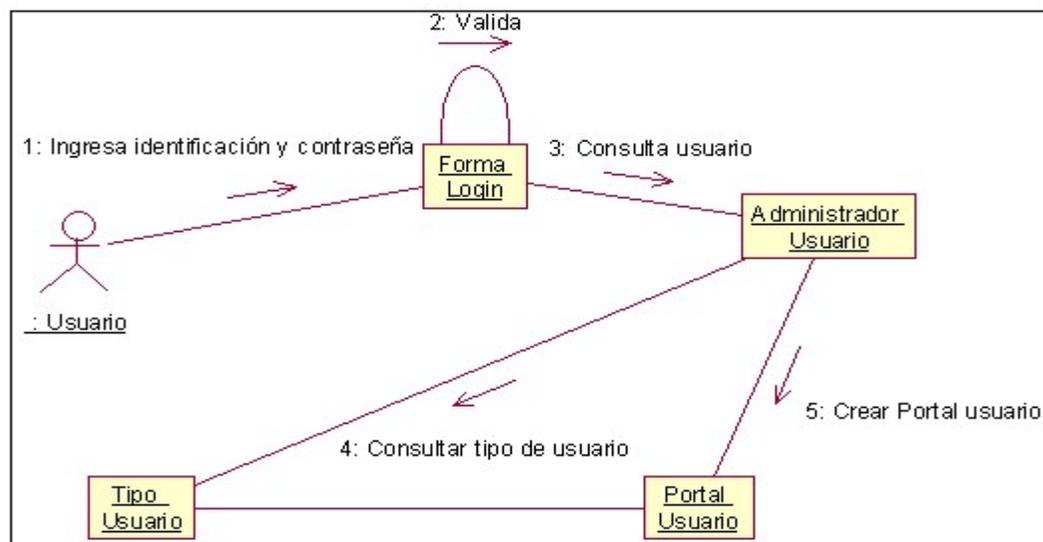
- “Describe la interacción de objetos que requiere la funcionalidad de los distintos escenarios de un Caso de Uso.
- Los objetos son representados con su ciclo de vida dentro de una serie temporal.
- Cada posible escenario de un Caso de Uso puede representarse con un diagrama de secuencia” [28].

Figura 3. Ejemplo Diagrama De Secuencia



**2.5.6 Diagrama De Colaboración.** “Muestra lo mismo que un diagrama de secuencia cómo interaccionan los objetos dentro de un Caso de Uso. A diferencia de un diagrama de secuencia no hay referencia a una serie temporal. Su propósito es mostrar la topología del proceso distribuido entre los distintos objetos” [29].

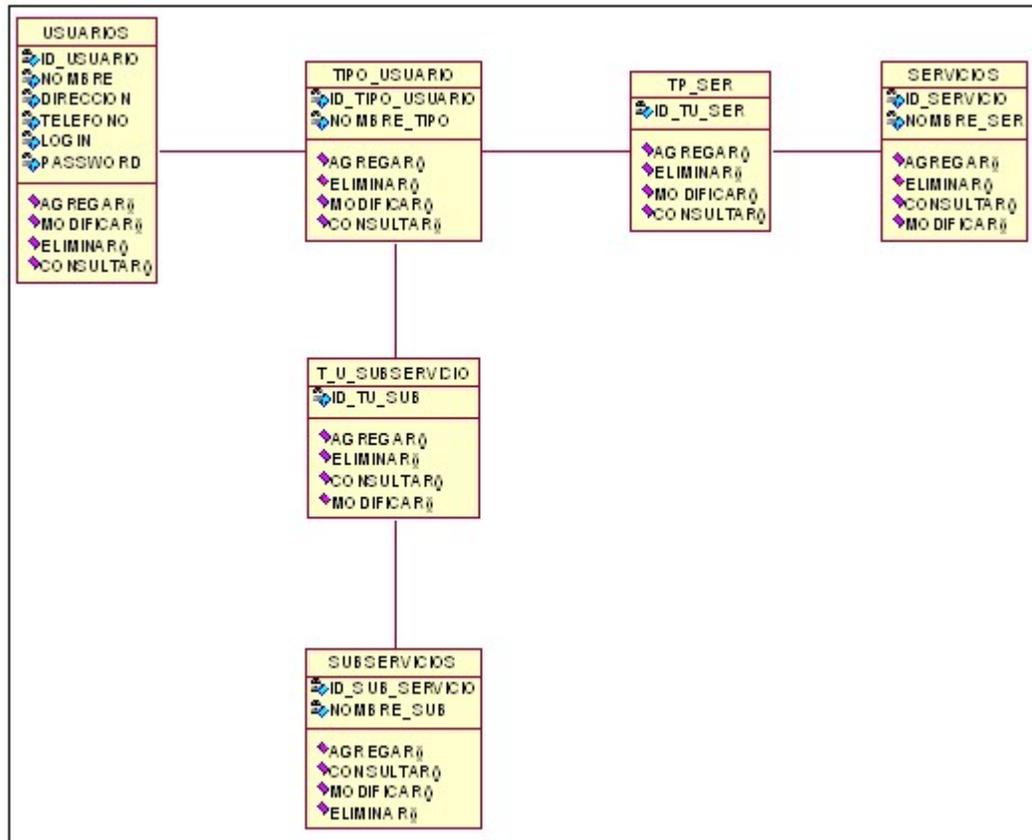
Figura 4. Ejemplo Diagrama de Colaboración



**2.5.7 Diagrama De Clases.** “Una Clase representa a un tipo de objetos que comparten:

- Objetos que comparte las mismas propiedades (Atributos).
- El mismo comportamiento (Métodos).
- Las mismas relaciones con otros objetos (asociaciones y agregaciones).
- La misma semántica dentro del sistema “[30]

Figura 5. Ejemplo Diagrama De Clases



## 2.6 JAVA

“Lenguaje de programación orientado a objetos desarrollado por la empresa Sun Microsystems en 1995 que permite diseñar aplicaciones para ser ejecutadas en las páginas de la Web. Es un lenguaje de alto nivel y propósito general similar a C++, con marcadas características de seguridad y transportabilidad. Es útil para realizar desde tareas sencillas como cálculos matemáticos hasta versiones completas de bases de datos para consultarlas a través de Internet. Tanto sus desarrolladores como gran cantidad de sectores involucrados con Internet aseguran que, en virtud de su característica de lenguaje multiplataforma (capaz de operar sobre cualquier plataforma a través, normalmente, de navegadores WWW),

es el lenguaje de programación del futuro. Este lenguaje define una máquina virtual independiente de la plataforma donde se ejecuta, que procesa programas, llamados Applets, descargados desde el servidor Web. Además, debido al modo de ejecución de los Applets, este lenguaje es muy seguro frente a la presencia y ataque de virus informáticos”. [31]

**2.6.1 SERVLETS.** Los Servlets son módulos que extienden los servidores orientados a petición-respuesta, como los servidores Web compatibles con Java. Por ejemplo, un servlet podría ser responsable de tomar los datos de un formulario de entrada de pedidos en HTML y aplicarle la lógica de negocios utilizada para actualizar la base de datos de pedidos de la compañía.

Los servlets pueden ser incluidos en muchos servidores diferentes porque el API Servlet, el que se utiliza para escribir Servlets, no asume nada sobre el entorno o protocolo del servidor. Los servlets se están utilizando ampliamente dentro de servidores HTTP; muchos servidores Web soportan el API Servlet.

#### **2.6.1.1 Características de los Servlets.**

1. “Es independiente de la plataforma en la que se este ejecutando.
2. Ejecución multihilo. Cada una de las peticiones sobre el servlet creará una instancia que se ejecutará de manera independiente. A no ser de que le indiquemos lo contrario. El servlet permanece cargado en memoria por lo que atiende rápidamente las peticiones.
3. Un servlet puede ejecutarse en un sandbox. Que lo que hace es limitar los privilegios del servlet a un modelo controlado como el de los applets. Salvaguardando la integridad del host donde se ejecuta.

4. El servlet puede obtener información acerca de la máquina que ha realizado la petición (IP, puerto, tipo de método de envío: get o post,...).
5. Conexión a Bases de Datos. A través de los servlets podemos establecer conexiones a diferentes tipos de bases de datos. Esta característica acopla perfecta a los servlets dentro de una arquitectura cliente/servidor en 3 capas (cliente - servidor - datos).
6. Proxy para applets. Dentro del desarrollo de applets nos encontramos con un gran número de limitaciones, dentro de las cuales encontramos el acceso al sistema de archivos. Para subsanar dicha carencia podemos interponer un servlet entre el applet y el sistema de archivos, de tal manera que el applet se comunicaría con el servlet, que sería el encargado de acceder al sistema de archivos.
7. Recursos compartidos entre usuarios. Los servlets pueden definir estructuras o información que va a ser compartida por diferentes usuarios que utilicen el servlet. A la hora de utilizar esta información compartida o global deberemos de tomar las precauciones oportunas para que siempre sea una información correcta, íntegra y fiable”. [32]

**2.6.2 JAVABEANS.** “El API JavaBeans define un modelo de componentes de *Java*. Un *JavaBean* es un componente de software reutilizable que puede ser manipulado visualmente por una herramienta de programación en lenguaje *Java*” [33].

Los *JavaBeans* están escritos totalmente en *Java*, por lo que pueden ser utilizados en cualquier plataforma que soporte *Java*. Ésta es una de las principales cualidades de la arquitectura *JavaBeans*: la de proporcionar una arquitectura de componentes independiente de la plataforma. Además, también existen puentes

para poder utilizar los componentes JavaBean en otros modelos de componentes, como son ActiveX, OpenDoc, etc.

Una cualidad de estos componentes es que son ligeros, ya que no tienen que llevar múltiples librerías alrededor. Son, además, fáciles de crear y de usar, y se ejecutan en el mismo espacio de memoria que su contenedor.

**2.6.3 JAVA DATA BASE CONNECTION (JDBC).** “Es una especificación de la interfaz de aplicación de programa (application program interface, API) para conectar los programas escritos en Java a los datos en bases de datos populares. La interfaz de aplicación de programa nos permite codificar órdenes de solicitud de acceso en lenguaje estructurado de solicitud (structured query language, SQL) que luego pasan al programa que administra la base de datos. Devuelve los resultados a través de una interfaz similar. La JDBC es muy similar a la Conectividad Abierta de Base de Datos (Open Database Connectivity, ODBC) de Microsoft y, con un pequeño programa "puente" se puede usar la interfaz JDBC para acceder a bases de datos a través de la interfaz ODBC de Microsoft. “[34]

La JDBC tiene actualmente dos niveles de interfaz. Además de la interfaz principal hay también una API de un "administrador" JDBC que a su vez se comunica con los "controladores" de producto de las bases de datos individuales, si es necesario, el puente JDBC-ODBC bridge y a un controlador de red JDBC cuando el programa Java está corriendo en un entorno de red (es decir, accediendo a una base de datos remota).

**Modelo de dos y tres capas.** “JDBC soporta arquitectura de dos y tres capas:

- En dos capas: Un applet de Java o aplicación habla directamente con la B.D.
- Requiere un manejador de JDBC que pueda comunicarse con B.D.

- En tres capas: los comandos son mandados a la capa intermedia la cual a su vez manda sentencias SQL a B.D., ésta los procesa y los manda a la capa intermedia, la cual las manda a su vez al usuario.”[35]

## **2.7 JAVA SERVER PAGE (JSP)**

“Java Server Page -- JSP (Página de Servidor Java) Se refiere a un tipo especial de páginas HTML, en las cuales se insertan pequeños programas que corren sobre Internet (comúnmente denominados scripts), se procesan en línea para finalmente desplegar un resultado final al usuario en forma de HTML. Por lo general dichos programas hacen consultas a bases de datos y dependiendo del resultado que se despliegue será la información que se muestre a cada usuario de manera individual. Los archivos de este tipo llevan la extensión ".jsp".”[36]

## **2.8 BASES DE DATOS**

“Para diseñar unas Bases de Datos se debe establecer un proceso partiendo del mundo real, de manera que sea posible plasmarlo mediante una serie de datos. La imagen que se obtiene del mundo real se denomina modelo conceptual consiste en una serie de elementos que definen lo que se quiere plasmar del mundo real en una Bases de Datos. La definición de este modelo se denomina esquema conceptual y está compuesto de una parte estática llamada lenguaje de definición de datos ó DDL (Data Definition Language), donde se define la estructura de la Base de Datos y una parte dinámica denominada lenguaje de manipulación de datos o DML (Data Manipulation Language).”[37]

**2.8.1 Definición de Base de Datos.** “Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas. Por este motivo es importante conocer la forma en que están estructuradas las bases de datos y su manejo.”[38]

### **2.8.2 Componentes principales.**

- Datos: Los datos son la Base de Datos propiamente dicha.
- Hardware: El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- Software. Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DMBS: Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.
- Usuarios. Existen tres clases de usuarios relacionados con una Base de Datos:

*El programador de aplicaciones*, quien crea programas de aplicación que utiliza la base de datos.

*El usuario final*, quien accesa la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.

*El administrador de la Base de Datos (DBA: Data Base Administrator)*, quien se encarga del control general del Sistema de Base de Datos.

**2.8.3 El sistema organizador de Base de Datos (DBMS).** “El DBMS es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos (DDL: Data Definition Language), de un lenguaje de manipulación de datos (DML:

Data Manipulation Language) y de un lenguaje de consulta (SQL: Structured Query Language).

El lenguaje de definición de datos (DDL) es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una base de datos.

El lenguaje de manipulación de datos (DML) es utilizado para escribir programas que crean, actualizan y extraen información de las bases de datos.

El lenguaje de consulta (SQL) es empleado por el usuario para extraer información de la base de datos. El lenguaje de consulta permite al usuario hacer requisiciones de datos sin tener que escribir un programa, usando instrucciones como el SELECT, el PROJECT y el JOIN.

La secuencia conceptual de operaciones que ocurren para acceder cierta información que contiene una base de datos es la siguiente:

El usuario solicita cierta información contenida en la base de datos.

El DBMS intercepta este requerimiento y lo interpreta.

El DBMS realiza las operaciones necesarias para acceder y/o actualizar la información solicitada” [39].

**2.8.4 Ventajas en el uso de Bases de Datos.** “Globalización de la información. Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.

- Eliminación de información redundante. Duplicada
- Eliminación de información inconsistente. Si el sistema esta desarrollado a través de archivos convencionales, dicha cancelación deberá operarse tanto en el archivo de facturas del Sistema de Control de Cobranza como en el archivo de facturas del Sistema de Comisiones.

- Permite compartir información. Varios sistemas o usuarios pueden utilizar una misma entidad.
- Permite mantener la integridad en la información. Solo se almacena la información correcta.
- Independencia de datos. La independencia de datos implica un divorcio entre programas y datos; es decir, se pueden hacer cambios a la información que contiene la base de datos o tener acceso a la base de datos de diferente manera, sin hace cambios en las aplicaciones o en los programas.” [40]

## **2.9 ORACLE 8I**

Oracle8i es la base de datos para el Internet, esta cambia la forma en que la información es manejada y accesada para satisfacer las demandas del Internet, mientras que proporciona nuevas características significativas para el tradicional proceso transaccional en línea (OLTP) y para los datos de aplicaciones “warehouse”. Proporciona avanzadas herramientas para manejar todos los tipos de datos de sitios del Web, pero también ofrece el funcionamiento, la escalabilidad, y la disponibilidad necesaria para soportar bases de datos muy grandes (VLDB).

Oracle8i es mucho más que una simple base de datos relacional. Introduce a IFS, el file system del Internet que permite que los usuarios muevan fácilmente todos sus datos a una base de datos de Oracle8i, en donde pueden ser guardados y manejados más eficientemente en forma integrada. Oracle8i introduce la nueva ayuda para Java incluyendo una robusta, integrada, y escalable máquina virtual de Java dentro del servidor. Esto amplía la ayuda de Oracle para Java en todas las aplicaciones, permitiendo a los programas de Java ser desplegados, donde ellos funcionan en la mejor forma posible -- en el cliente, en el servidor, o en el medio -- sin recompilar o modificar el código de Java.

En resumen, Oracle8i fue diseñado para tener acceso y para manejar todos los datos usando el estilo e infraestructura del Internet. Oracle8i es la más completa y comprensiva plataforma para desarrollar, desplegando y manejando al Internet y a las aplicaciones tradicionales.

- \*  Simplifica el desarrollo de aplicaciones.
- \*  Simplifica la administración del contenido del Internet.
- \*  Simplifica el despliegue de aplicaciones.

**2.9.1 Arquitectura.** Una Base de Datos Oracle es un conjunto de datos almacenado y accesible según el formato de tablas relacionales. Una tabla relacional tiene un nombre y unas columnas, su definición. Los datos están almacenados en las filas. Las tablas pueden estar relacionadas con otras. Una Base de Datos Oracle está almacenada físicamente en ficheros, y la correspondencia entre los ficheros y las tablas es posible gracias a las estructuras internas de la BD, que permiten que diferentes tipos de datos estén almacenados físicamente separados. Esta división lógica se hace gracias a los espacios de tablas, *tablespaces*.

**2.9.2 Estructuras Internas De La Base De Datos.** Tablas y Columnas: Los datos son almacenados en la BD utilizando tablas. Cada tabla está compuesta por un número determinado de columnas.

Las tablas propiedad del usuario SYS son llamadas tablas del diccionario de datos. Proveen el catálogo del sistema que permite que la BD se gestione a sí misma.

Las tablas se pueden relacionar entre ellas a través de las columnas que las componen. La BD se puede utilizar para asegurar el cumplimiento de esas relaciones a través de la integridad referencial, que se concreta en las restricciones de tablas.

**2.9.2.1 Restricciones de Tablas.** Una tabla puede tener asociadas restricciones que deben cumplir todas las filas. Entre las restricciones que se pueden fijar algunas reciben nombres especiales.: *clave primaria, clave ajena*.

“La clave primaria de una tabla está compuesta por las columnas que hacen a cada fila de la tabla una fila distinta.

La clave ajena se utiliza para especificar las relaciones entre tablas. De modo que un conjunto de columnas declaradas como clave ajena de una tabla deben tener valores tomados de la clave primaria de otra tabla.” [41]

**2.9.2.2 Usuarios.** Una cuenta de usuario no es una estructura física de la BD, pero está relacionada con los objetos de la BD: los usuarios poseen los objetos de la BD. Existen dos usuarios especiales: SYS y SYSTEM. El usuario SYS posee las tablas del diccionario de datos; que almacenan información sobre el resto de las estructuras de la BD. El usuario SYSTEM posee las vistas que permiten acceder a las tablas del diccionario, para el uso del resto de los usuarios de la BD.

Todo objeto creado en la BD se crea por un usuario, en un espacio de tablas y en un fichero de datos determinado. Toda cuenta de la BD puede estar unida a una cuenta del S.O., lo que permite a los usuarios acceder a la cuenta de la BD sin dar la clave de acceso.

Cada usuario puede acceder a los objetos que posea o a aquellos sobre los que tenga derecho de acceso.

### **3. DESARROLLO GENERAL DEL SISTEMA.**

“Los objetos concurren en nuestro entorno, en los negocios, en los productos y servicios que usamos. Pueden ser catalogados, coordinados, compuestos, manipulados y creados. Por esto la programación orientada a objetos a la hora de crear software es de vital importancia, ya que esto ayuda a modelar lo abstracto para poder entenderlo y gobernarlo mejor.”[42]

Es por esa razón que se realizó el análisis, diseño e implementación de los módulos correspondiente a la evaluación docente y perfiles para los usuarios de los portales UNAB usando la metodología Rational Unified Procces (RUP) con la notación de Unified Modeling Language (UML).

Las fases para el desarrollo de estos módulos son los siguientes:

#### **3.1 FASE 1 ESTUDIO Y APRENDIZAJE.**

En esta fase se realizó una investigación de tipo exploratoria\* y al mismo tiempo descriptiva\*\* en la cual se definieron los perfiles de usuarios de los portales UNAB por medio de una combinación de técnicas de filtración de información como lo son las reglas de negocio y la explícita.

A su vez se realizó un estudio de tipo descriptivo al proyecto SIGPEED donde se retomó toda la estructura que en este manejaron para el desarrollo de la evaluación docente. Las herramientas que se utilizaron para formar toda esta

---

\* Investigación exploratoria: ayudar a obtener, con relativa rapidez, ideas y conocimientos en una situación.

---

\*\* Investigación descriptiva: exigen que el investigador identifique de antemano las preguntas específicas que desea contestar, cómo las responderá y las implicaciones que posiblemente tengan.

metodología y su correspondiente implementación están compuestas especialmente por libros, la tesis de SIGPEED, documentos, actas que tenían establecidas algunas normas para la implementación de la evaluación docente, la documentación que tiene el departamento de sistemas en el desarrollo de la estandarización de los portales e Internet.

#### **Actividades Realizadas:**

- Investigación del análisis de las reglas de negocio para la creación del módulo de los perfiles de usuarios para los portales de la UNAB.
- Estudio al proyecto SIGPEED, se pudieron reutilizar algunos elementos que nos sirvieron para la implementación de la evaluación docente de la UNAB como el Diagrama Entidad Relación (DER) y la programación en JSP.

La conclusión del estudio se puede resumir en lo siguiente:

Desde principios del mes de Junio se empezaron a realizar lecturas y un análisis breve del proyecto SIGPEED, se identificaron conceptos claves para el desarrollo del módulo de la Evaluación Docente tales como el manejador de Bases de Datos Oracle, el cual fue utilizado para el desarrollo del Sistema SIGPEED, aquí se utilizó una versión más avanzada la cual es Oracle 8i.

Inconvenientes presentados:

El Departamento de Sistemas tenía una copia de los scripts pero no pudieron recuperarla ya que el servidor se dañó.

En el momento de empezar a conectar la bases de datos con la programación realizada en Java se detectaron ciertas falencias entre el análisis y el diseño de el sistema SIGPEED, como por ejemplo el nombre de las tablas que eran llamadas en las clases de Java eran totalmente diferentes a las del DER\*. Lo que ocasionó problemas a la conexión. (Ver Anexo A)

---

\* Diagrama Entidad - Relación

El análisis y la programación de SIGPEED la realizaron por aparte sin tener en cuenta que se podrían presentar inconvenientes por la inconsistencia existente en el desarrollo del Sistema.

### **Descripción del Proceso**

- El modelamiento del DER de la Evaluación Docente actualizado en Designer 6i, para no perder el análisis realizado en el proyecto. (Ver Anexo B)
- Generar los scripts nuevamente.
- Cambiar los nombres de las tablas y sus atributos en la Base de Datos por medio de Sql Navigator.
- Un correcto análisis del Sistema para que pudiese funcionar coherentemente, esto por medio del Diccionario de Datos. (Ver Anexo D )
- Aprendizaje de las herramientas JSP, JAVA, ORACLE 8I, DESIGNER 6I.
- Estudio de las normas o reglas que existen en la UNAB para así poder realizar la evaluación docente, por medio de reuniones con la Vicerrectora Académica de la Universidad, con las diferentes facultades, con el Departamento de Sistemas, Director y evaluadores. (Ver Anexo F)

## **3.2 SEGUNDA FASE: ANÁLISIS ORIENTADO A OBJETOS**

No se puede construir software (orientado a objetos o de otro tipo) hasta que se tiene un conocimiento razonable del sistema o producto. El AOO nos proporciona una forma concreta de representar el conocimiento de los requisitos y una forma de probar dicho conocimiento enfrentándolo con la percepción que el cliente tiene del sistema a construir.

Es necesario abstraer primero las características importantes del mundo real.

**3.2.1 Objetivos del Análisis.** “El objetivo del Análisis Orientado a Objetos (AOO) es desarrollar una serie de modelos que describan el software de computadora al trabajar para satisfacer un conjunto de requisitos definidos por el cliente. El AOO, como los métodos de análisis convencionales, forma un modelo de análisis

multiparte para satisfacer este objetivo. El modelo de análisis ilustra información, funcionamiento y comportamiento dentro del contexto de los elementos del modelo de objetos". [43]

El resultado denominado Lenguaje de Modelado Unificado (UML), se ha convertido en el método más utilizado por la industria. UML permite a un ingeniero del software expresar un modelo de análisis utilizando una notación de modelamiento con unas reglas sintácticas, semánticas y prácticas. En UML, un sistema viene representado por cinco vistas diferentes que lo describen desde diferentes perspectivas. Cada vista se representa mediante un conjunto de diagramas.

### **3.2.1.1 Actividades Desarrolladas.**

**Análisis de la estructura de los portales de la UNAB:** para establecer las reglas de negocio para la implementación de perfiles de usuarios.

Los usuarios de la Web se encuentran clasificados como:

*Público interno:* Estudiantes, administrativos y docentes. En este nivel se encuentran usuarios administrativos con privilegios para la publicación de documentos en el Web.

*Público externo:* Visitantes interesados en conocer la Universidad y cursar algún tipo de programa. Visitantes de las Universidades Asociadas a la Red Mutis.

**MAPA DEL SITIO UNAB VIRTUAL:** Es un portal enfocado al servicio de los estudiantes y la promoción de los programas en esta modalidad. Desarrolla espacios de interacción y comunicación para el desarrollo de la educación virtual. La coordinación de este canal se encuentra actualmente a cargo de Producciones UNAB y UNAB Virtual. El objetivo de UNAB VIRTUAL es ser un portal de servicios y comunicación para el desarrollo de la educación virtual de la UNAB.

***Los usuarios que podemos encontrar en este sitio son los siguientes:***

- Profesores: con privilegios sobre los cursos para crear y desarrollar su materia.

- Estudiantes: con privilegios para acceder a los cursos y consultar sus materiales y desarrollar actividades de discusión, enviar sus tareas.
- Administrativos: con privilegios sobre el Web para publicación y actualización de cursos.
- Visitantes: En los niveles de postgrado y educación superior.

**Definición del dominio de información:** el análisis del dominio puede verse como la actividad de cobertura para el proceso del software. Con esto se quiere decir que el análisis del dominio es una actividad en curso de la ingeniería del software no ligada a ningún proyecto de software. Esto se realiza con entrevistas con el cliente (Coordinador del proyecto, Vicerrectora Académica, Departamento de Sistemas, usuarios, profesores etc.), identificando los requisitos y las clases básicas para realizar la evaluación docente y el perfil de los usuarios de los portales UNAB, dentro del documento visión podemos encontrar lo siguiente:

- El objetivo final del módulo de personalización de perfiles para los usuarios de los portales UNAB es facilitar al estudiante, docente y administrativo el acceso a la información preestablecida por el Departamento de Sistemas según las normas que rigen el sistema académico de la Universidad. En esta etapa se procedió a realizar reuniones con la persona encargada en el Dpto. de Sistemas para hacer el levantamiento de los requerimientos de la personalización de perfiles que según las pautas suministradas se realizó un prototipo el cual aplica el concepto de la personalización, este fue desarrollado en JSP y se utilizó Oracle 8i como manejador de una Base de datos SQL.

Este prototipo crea dinámicamente el portal con los servicios asignados o permitidos para cada perfil.

Los perfiles son los siguientes: administrativo, estudiante y visitante; estos perfiles se crearon por medio del método explícito o creación manual, los datos en el caso de nuestro prototipo son los servicios asignados por la

UNAB a cada uno de los perfiles los cuales fueron introducidos por nosotras escribiéndolos directamente en su perfil de usuario.

El usuario suministra en una página de ingreso un login y password que el sistema reconoce, con este realiza una búsqueda y selecciona el o los perfiles (un mismo usuario puede tener varios perfiles) del usuario, de igual manera realiza otra búsqueda en la que selecciona los servicios permitidos para el tipo de perfil y luego crea el menú de la página dinámicamente. En caso que el usuario tenga más de un perfil, el sistema realiza una selección de los servicios no repetidos en los diferentes perfiles a los cuales pertenece y de esta forma crea el menú en una misma página con todos los servicios para los diferentes perfiles del mismo usuario. Esto es muy importante ya que un usuario que es estudiante y a la vez administrativo, no tendría que ingresar a portales diferentes para realizar actividades distintas entre los dos perfiles, sino que en un mismo portal tendrá acceso a los servicios designados para ambos perfiles.

En la página de ingreso se encuentra un botón “visitantes”, el cual genera un portal especial para aquellos que no hacen parte de la comunidad UNAB.

- Se realizó el módulo de evaluación docente para que sirva de implementación de la personalización de perfiles de usuarios para los portales UNAB. La evaluación docente se tomó del proyecto SIGPEED (Sistema De Información Para La Gestión Del Portafolio De Empleados y La Evaluación Docente En la Unab) implementado en oracle 8i, java y jsp. Es un sistema adaptativo ya que se va a poder implementar en un motor de mayor versión.

Este módulo permitirá a todos los estudiantes de Pregrado evaluar a sus correspondientes docentes, donde todas las facultades realizarán una evaluación institucional. Las preguntas realizadas se encuentran clasificadas por diferentes áreas tales como (personal, académica, profesional, ética etc.); estas (áreas y preguntas) van a ser dinámicas (el usuario “administrador”

tiene la posibilidad de cambiarlas) según su criterio, lo propio con el tipo de respuestas.

El estudiante podrá realizar la evaluación a su profesor desde el portal del estudiante.

Para empezar a evaluar el usuario ingresa un login y password.

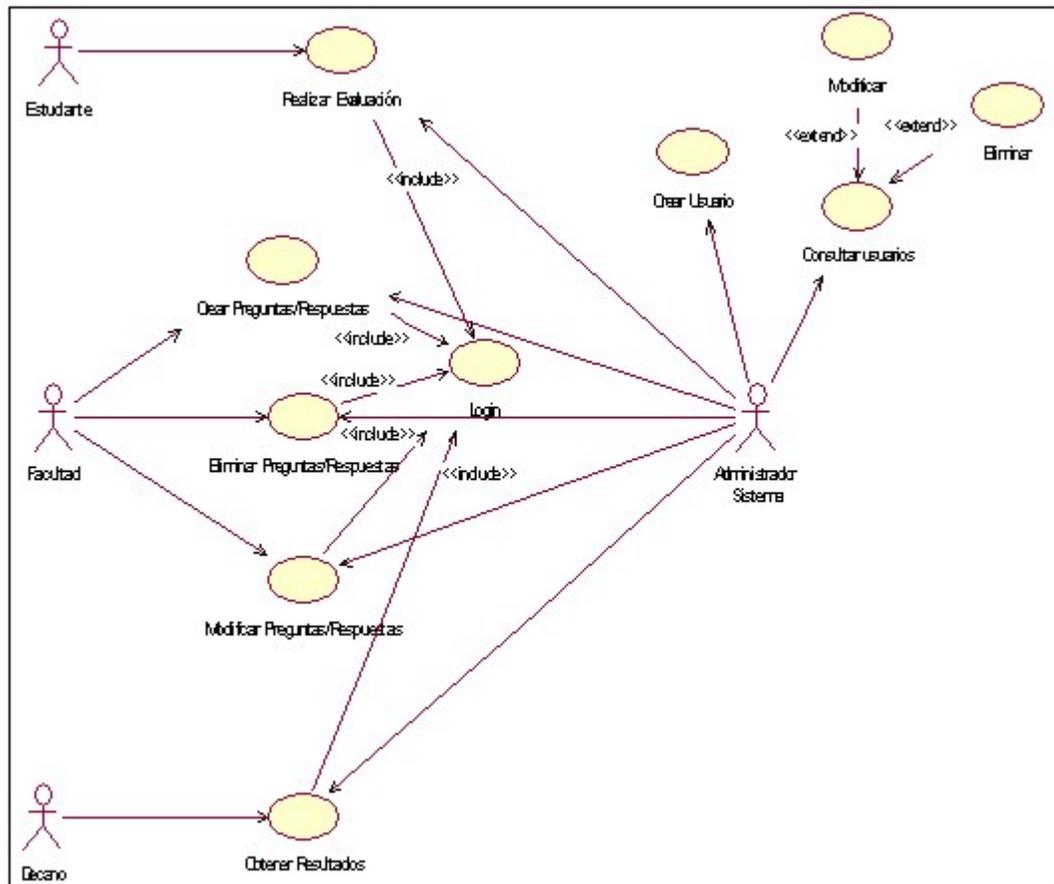
Una vez el estudiante ingresa a la heteroevaluación tendrá que dar clic en el botón de la facultad a la cual el está inscrito después el proseguirá a realizar la evaluación de cada uno de los docentes con los que el está viendo las diferentes asignaturas, el alumno al finalizar correctamente dicha evaluación dar clic en aceptar, para que estos datos puedan ser ingresados a la Base de Datos del sistema y así sucesivamente hasta que termine de evaluar todos los docentes que le corresponden.

Como usuario Decano esté tendrá el permiso para acceder a los resultados de las diferentes evaluaciones sin poder modificarla ni alterarla, dentro de la información desplegada nunca se va hallar los nombres, ni códigos de los estudiantes que evaluaron a dicho profesor.

- Identificación de las clases de análisis, que son las clases de interfaz, de control y de entidades. Para ver el diagrama de análisis dirigirse al Anexo C.
- Crear un conjunto de escenarios (casos de usos) iniciales que nos facilite una descripción de cómo van a ser usados los dos módulos cuando se requiera.
- Identificar los diferentes tipos de personas que los van a utilizar.

En la figura 6, se observa el Caso de uso realizado para la Evaluación Docente y la Personalización.

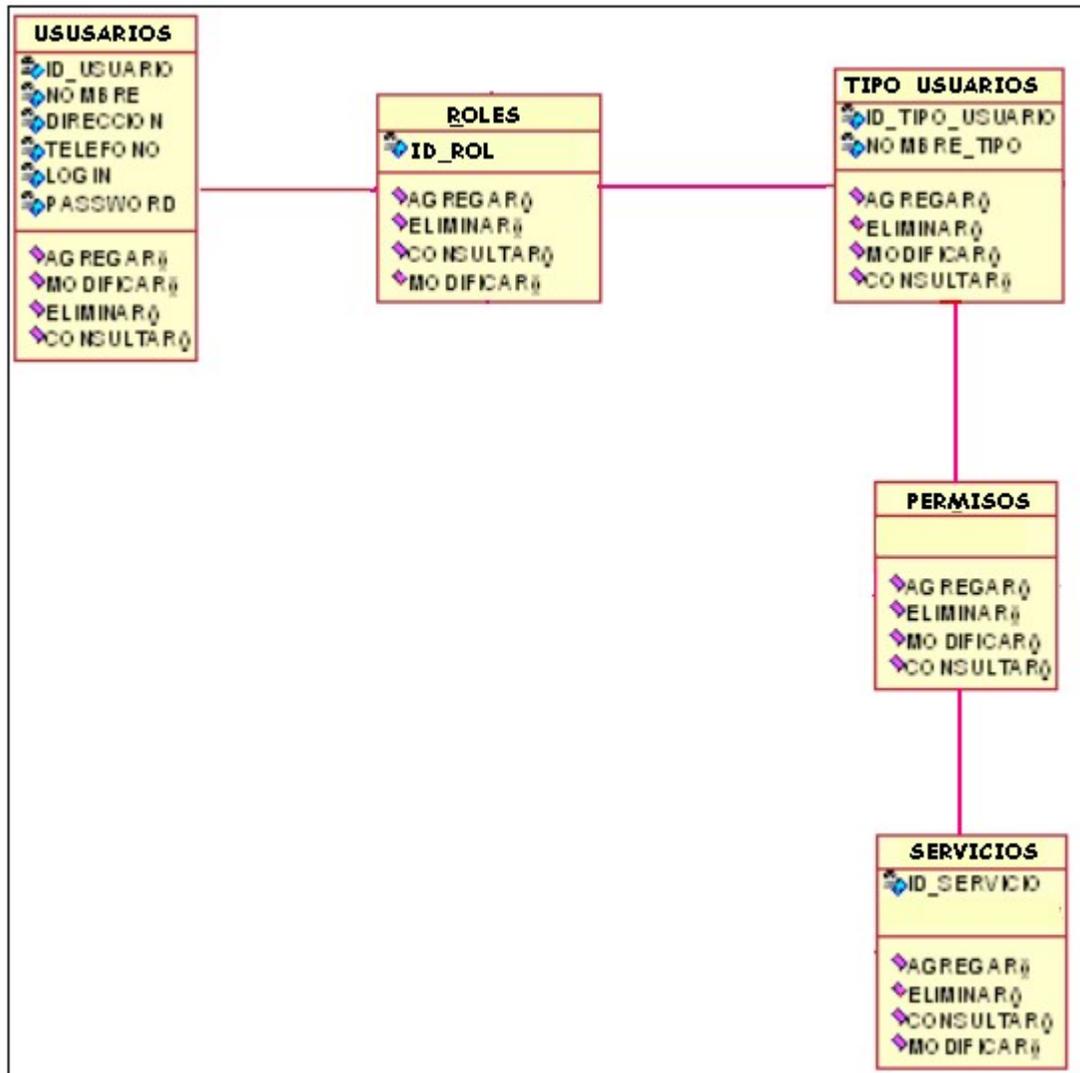
Figura 6. Caso de uso Evaluación Docente y Personalización



- Análisis del dominio: el objetivo es encontrar o crear aquellas clases u objetos ampliamente aplicados de tal manera que sean reutilizables.

En las figuras 7 y 8 se encuentran los diagramas de clases para cada uno de los módulos Personalización y Evaluación Docente.

Figura 7. Diagrama De Clases de la Personalización





- Realizar un modelado de Clases-Responsabilidades-colaboraciones (CRC), la cual se aplica para documentar las clases y sus atributos y operaciones, hacer una clasificación de objetos y una jerarquía de clases.

### **Evaluación Docente:**

REALIZAR EVALUACIÓN DOCENTE: El Estudiante ingresa al su respectivo portal e ingresa al módulo evaluación docente.

→ Descripción corta: Inicia cuando el usuario ingresa al módulo evaluación docente y da clic en “INGRESAR”. Finaliza cuando el sistema le desactiva al estudiante la evaluación docente, es decir cuando no aparecen más docentes por evaluar.

→ Precondición: Que el usuario ingrese al portal estudiante, luego al módulo evaluación docente y por último ingrese el login y password que le corresponde.

→ Flujo de eventos: el usuario ingresa al portal, ingresa al módulo evaluación docente, escribe su Login y Password y le da clic en el botón “INGRESAR”, configura el despliegue del numero de facultades y materias a la que el está matriculado.

→ Flujo alternativo: que el portal del estudiante no funcione.-

→ Poscondición: el usuario hace clic en finalizar.

CREAR EVALUACIÓN DOCENTE :El usuario Facultad ingresa al portal administrativo y se dirige Gestión de Evaluación docente.

→ Descripción corta: inicia cuando la Facultad da clic en gestión de la evaluación docente. Finaliza cuando el usuario configura, actualiza, crea o elimina la evaluación docente según sus criterios.

→ Precondición: Que el usuario ingrese al portal y de clic en Gestión de Evaluación docente.

- Flujo de eventos: el usuario ingresa al portal, la facultad da clic en Gestión de evaluación docente y el usuario ingresa a la interfaz de crear, actualizar o modificar evaluación docente.
- Flujo alternativo: que el portal no funcione.
- Poscondición: el usuario utiliza el servicio.

**CONSULTAR EVALUACIÓN DOCENTE:** El Decano tiene facilidad para ingresar a la información estadística de la evaluación docente, cada vez que lo requiera necesario para obtener sus debidos reportes.

- Descripción corta: inicia cuando el Decano de cada facultad necesita información estadística de los resultados de la evaluación docente. Finaliza cuando el Decano recibe reportes acerca de la información solicitada.
- Precondición: ser verdaderamente el Decano y necesitar los reportes de la evaluación docente.
- Flujo de eventos: el Decano ingresa al portal Administrativo, ingresa login y password del Decano- selecciona la información que necesita ver las estadísticas.
- Flujo alternativo: no coincida el login y password, el sistema no valida el ingreso del administrador.
- Poscondición: el Decano imprime y analiza la información obtenida, concluye y toma decisiones para beneficio de la Universidad.

**INGRESAR USUARIO:** El usuario ingresa al portal e ingresa su login y password.

- Descripción corta: Inicio el usuario ingresa login y password y da clic en el Botón "INGRESAR". Finaliza confirmación por parte del sistema.
- Precondición: Usuario ingresa al módulo evaluación docente.
- Flujo de eventos: El usuario ingresa al portal, el usuario ingresa el login y password y da clic en "INGRESAR", el sistema confirma si el usuario existe

en la base de datos de los usuarios, el registro es aceptado y el usuario ingresa a los servicios específicos.

→ Flujo alternativo: usuario ingresa el login y password incorrectos, se consulta a la base de datos y se dará un aviso de acceso denegado, y el usuario intentará de nuevo ingresar datos.

→ Poscondición: El usuario puede acceder el servicio deseado.

VALIDAR USUARIO: Cuando el usuario ingresa login y password el sistema consulta la base de datos la validez de este.

→ Descripción corta: Inicia cuando el usuario da clic en ingresar o registrar. Finaliza con la confirmación o negación de login/password por parte del sistema.

→ Precondición: el usuario ingresa login/password y de clic en ingresar o registrar.

→ Flujo de eventos: el usuario ingresa al portal, ingresa login y password, el usuario da clic en ingresar o registrar, el sistema consulta base de datos y este confirma validez de login/password al usuario.

→ Flujo alternativo: el usuario ingresa login/password incorrecto, el sistema consulta base de datos, el sistema niega ingreso al usuario y el usuario intenta registrarse nuevamente.

→ Poscondición: el usuario esta apto para ingresar a cierta información que no todo internauta puede hacerlo.

- Interfaz Decano: la interfaz de decano le permite acceder a los resultados de la evaluación docente y al mismo tiempo permite imprimir este informe.
- Interfaz Estudiante: el estudiante solamente tendrá acceso a la función de evaluar a los docentes según a la facultad y a las materias a la que el este inscrito.

- Interfaz Facultad: el interfaz facultad tendrá acceso solamente a la gestión de la evaluación con el privilegio de crear, modificar, actualizar y borrar la evaluación correspondiente a cada facultad.
- Interfaz Administrador: el administrador le permite acceder a todas las opciones que representa la aplicación.

### **3.3 TERCERA FASE: DISEÑO ORIENTADO A OBJETOS**

Se transformó el modelo de análisis orientado a objetos, en un modelo de diseño que sirvió como anteproyecto para la construcción de los módulos.

ACTIVIDADES DESARROLLADAS: La actividad principal de este periodo fue la definición de una arquitectura de software multicapa, realizando los siguientes pasos:

- Diseño de los dos módulos: el cual creó la arquitectura del producto y encerró las siguientes actividades:
  - Desarrollar un diseño para la interfaz de usuario.
- Diseño de objetos, el cual se centra en los detalles internos de cada clase, definición de atributos, operaciones y detalles de los mensajes.
- Modelo de clases, describiendo las clases de los módulos y las relaciones que existen entre ellas.
- Documentación casos de uso en UML.
- Diagramas de secuencia para cada uno de los casos de uso.

### **3.3.1 Diagramas de Secuencia de la Personalización.**

Para observar los diagramas de secuencia de la personalización dirigirse al Anexo D.

### *3.3.2 Diagramas de Secuencia de la Evaluación Docente.*

Para observar los diagramas de secuencia de la Evaluación Docente dirigirse al Anexo E.

### **3.3.3. Diagramas de colaboración de la Personalización.**

Para observar los diagramas de colaboración de la Personalización dirigirse al Anexo F.

### *3.3.4 Diagramas de colaboración de la Evaluación Docente.*

Para observar los diagramas de colaboración de la Evaluación Docente dirigirse al Anexo G.

Además en esta fase se realizó la arquitectura del software, descripción de la interfaz de usuario, componentes de gestión de datos, mecanismos de administración de tareas y descripción detallada de cada una de las clases usadas en los módulos.

### **3.4 CUARTA FASE: IMPLEMENTACIÓN**

Esta fase se centra en la transformación del diseño en código utilizando la metodología orientado a objetos (UML).

#### **ACTIVIDADES DESARROLLADAS**

- Realizar la programación, las clases deben ser convertidas

El resultado será el producto final, los módulos de evaluación docente y perfil de usuarios con análisis orientado a objetos.

### **3.5 QUINTA FASE: PRUEBAS**

Se realizaron una secuencia de pruebas, en una variedad de niveles diferentes, en un esfuerzo por descubrir errores para corregirlos

#### **ACTIVIDADES DESARROLLADAS**

- Revisión de los modelos de análisis, diseño e implementación
- Pruebas a las diferentes clases, y sus operaciones para así examinar los errores existentes.
- Pruebas de los módulos desarrollados.

Como resultado final se espera que los módulos se implementen sin errores y robusto.

## 4. LOGROS, CONCLUSIONES Y SUGERENCIAS

### 4.1 LOGROS

- Con el transcurrir de la tesis se han obtenido una gran cantidad de logros, los que han sido de beneficio tanto para el área profesional como personal.
- Se aprendió en primer lugar a trabajar en equipo, del mismo modo se pusieron en práctica los conocimientos básicos adquiridos durante todo el proceso de la carrera.
- Se corroboró por medio de los inconvenientes presentados que en cuanto al desarrollo de software se refiere, no significa que la única tarea por hacer sea programar sino que se debe realizar un buen análisis, debe empezarse con una serie de tareas de modelado para llegar a una representación de alta calidad del diseño general del software.
- De igual manera se aprendió que para todo objetivo se debe realizar un plan de contingencia, no menos importante que se debe ser claro y específico en el momento de realizar el ámbito o el alcance del proyecto con el cliente.
- A pesar de la poca experiencia por no decir nula, y de la gran cantidad de problemas que se presentaron, se solucionaron, por ejemplo aunque no existía una documentación sobre la Base de Datos leyendo el Diagrama Entidad Relación se pudo comprender la estructura y la función del sistema.

- Se adquirieron conocimientos en gestión y manejo de las herramientas Oracle 8i, y designer6i, al igual sobre Java y JSP (Java Server Page).
- Por otro lado la investigación llevada a cabo sobre personalización de perfiles deja más claro cómo trabajan los portales en la Web sitios como lo son amazon.com, Myyahoo y se adquirieron conocimientos sobre las diferentes técnicas de filtración de información, como lo es la explícita que se utilizó para este módulo conjuntamente con las reglas de negocio.

## **4.2 CONCLUSIONES**

El trabajo realizado en esta tesis fue tedioso en el sentido que se utilizó mucho del tiempo que se tenía destinado para investigación y estudio de herramientas en recopilación de la información y documentación que debería haber tenido la tesis en la cual nos basamos, tiempo con el cual no contábamos.

La creación de la documentación de un producto siempre se considera un aspecto secundario en relación con la programación y la producción, lo que supone retrasarlo hasta última hora, de modo que se genera nerviosismo y, por tanto, manuales de mala calidad. La mala calidad de la documentación es, en muchas ocasiones, motivo de insatisfacción en relación con un producto, lo que pone en riesgo que se renueven los pedidos.

## **4.3 SUGERENCIAS**

Evitar en lo posible trabajar con software realizado por terceros a menos que esté muy bien documentado, y de ser posible hacer un estudio de éste antes de comprometerse a realizar las modificaciones.

Hablando en términos de ingeniería del software:

Producto Acabado = Programa Software + Manual de Usuario

- La calidad de un producto software se ve mermada si carece de la adecuada documentación.
- Producto no documentado = poco útil al usuario.

En cuanto al desarrollo de la Evaluación Docente, podría haber cabida para nuevas tesis que se encaminaran a desarrollar los otros tipos de evaluación como lo son la Autoevaluación y la Coevaluación que serían muy útiles para la Universidad.

A través del uso de técnicas de personalización, los portales han evolucionado para entregar precisamente la información específica que se necesita y de la manera en que se le necesita; podrían estudiarse en otros proyectos de grado algunas de las técnicas de Inteligencia Artificial de soporte a perfiles de usuario como lo son la Programación en Lógica Inductiva (PLI), y las Redes Neuronales Artificiales (RNA).

## BIBLIOGRAFÍA

- [1] VERGARA ZAPATA, David Esteban. Introducción a la programación Multicapas [http://www.elguille.info/colabora/puntoNET/jevergara\\_Multitier.htm](http://www.elguille.info/colabora/puntoNET/jevergara_Multitier.htm)
- [2] GLOSARIO DE INTERNET [on line] [citado el 10 de julio de 2003]. Disponible en Internet:<http://gti1.edu.um.es:8080/portales/PORTALES-cv-ocio.pdf>
- [3] GLOSARIO DE INTERNET [on line] [citado el 10 de julio de 2003]. Disponible en Internet: <http://www.edudistan.com/ponencias/Carlos%20E%20Paldao.htm>
- [4] GLOSARIO DE INTERNET [on line] [citado el 10 de julio de 2003]. Disponible en Internet: <http://www.edudistan.com/ponencias/Carlos%20E%20Paldao.htm>
- [5] GLOSARIO DE INTERNET [on line] [citado el 12 de agosto de 2003]. Disponible en Internet: [http://www.csvarnet.com/modulaweb\\_intranet.asp](http://www.csvarnet.com/modulaweb_intranet.asp)
- [6] High Volume Web Site Team. Web Site Personalization. [citado el 12 de agosto de 2003]. Disponible en Internet: <http://www.ibm.com/websphere/developer/zones/hvws>
- [7] P. P. DA CRUZ, Rui Alexandre. Perfiles de usuario: en la senda de la personalización. [citado el 14 de agosto de 2003]. Disponible en Internet: <http://tejo.usal.es/inftec/2003/DPTOIA-IT-2003-001.pdf>
- [8] PRESSMAN, Rooger S. Ingeniería Del Software Un enfoque Práctico. Madrid : McGraw-Hill, 2002. p. 343.
- [9] Ibid., p.344.

[10] Ibid., p.346.

[11] PRESSMAN, Op. Cit., p.346.

[12] PRESSMAN, Op. Cit., p.347.

[13] PRESSMAN, Op. Cit., p.347.

[14] PRESSMAN, Op. Cit., p.348.

[15] RUMBAUGH, James. El Lenguaje Unificado De Modelado. Madrid. Addison Wesley Iberoamericana, 2000. p.11.

[16] JACBSON, Ivar. UML . El Proceso Unificado De Desarrollo De Software. Madrid. Addison Wesley, 2000. p.31.

[17] Ibid., p.7.

[18] Ibid., p.32.

[19] JACBSON, Op. Cit., p.33.

[20] JACBSON, Op. Cit., p.34.

[21] JACBSON, Op. Cit., p.37.

[22] JACBSON, Op. Cit., p.38.

[23] JACBSON, Op. Cit., p.38.

[24] JACBSON, Op. Cit., p.39.

[25] JACBSON, Op. Cit., p.57.

[26] JACBSON, Op. Cit., p.81.

[27] JACBSON, Op. Cit., p.83.

[28] RUMBAUGH, Op. Cit., p.213.

[29] RUMBAUGH, Op. Cit., p.215.

[30] RUMBAUGH, Op. Cit., p.94.

[31] GLOSARIO DE INTERNET [on line] [citado el 19 de julio de 2003]. Disponible en Internet:

<http://www.inst-informatica.pt/v20/ibst/estandar/1/ibjava.htm>

[32] GLOSARIO DE INTERNET [on line] [citado el 19 de julio de 2003]. Disponible en Internet:

[http://www.programacion.com/java/tutorial/servlets\\_basico](http://www.programacion.com/java/tutorial/servlets_basico)

[33] GLOSARIO DE INTERNET [on line] [citado el 19 de julio de 2003]. Disponible en Internet:

<http://www.plan9.escet.urjc.es/who/nemo/export/java.ps>

[34] GLOSARIO DE INTERNET [on line] [citado el 19 de julio de 2003]. Disponible en Internet:

<http://www.gamarod.com.ar/recursos/glosario/mostrar.asp?letra=J>

[35] GLOSARIO DE INTERNET [on line] [citado el 25 de julio de 2003]. Disponible en Internet: <http://www.gamarod.com.ar/recursos/glosario>

[36] GLOSARIO DE INTERNET [on line] [citado el 25 de julio de 2003]. Disponible en Internet: <http://www.utem.cl/web/dicciojfm.htm>

[37] GLOSARIO DE INTERNET [on line] [citado el 22 de agosto de 2003]. Disponible en Internet: <http://www.rinconprog.metropoliglobal.com/CursosProg/BDatos/IntroBD>

[38] GLOSARIO DE INTERNET [on line] [citado el 22 de agosto de 2003]. Disponible en Internet: <http://www.monografias.com/trabajos12/basdat/basdat.shtml>

[39] GLOSARIO DE INTERNET [on line] [citado el 22 de agosto de 2003]. Disponible en Internet: <http://www.monografias.com/trabajos5/dissist/dissist.shtml>

[40] GLOSARIO DE INTERNET [on line] [citado el 27 de agosto de 2003]. Disponible en Internet: <http://www.monografias.com/trabajos12/basdat/basdat.shtml>

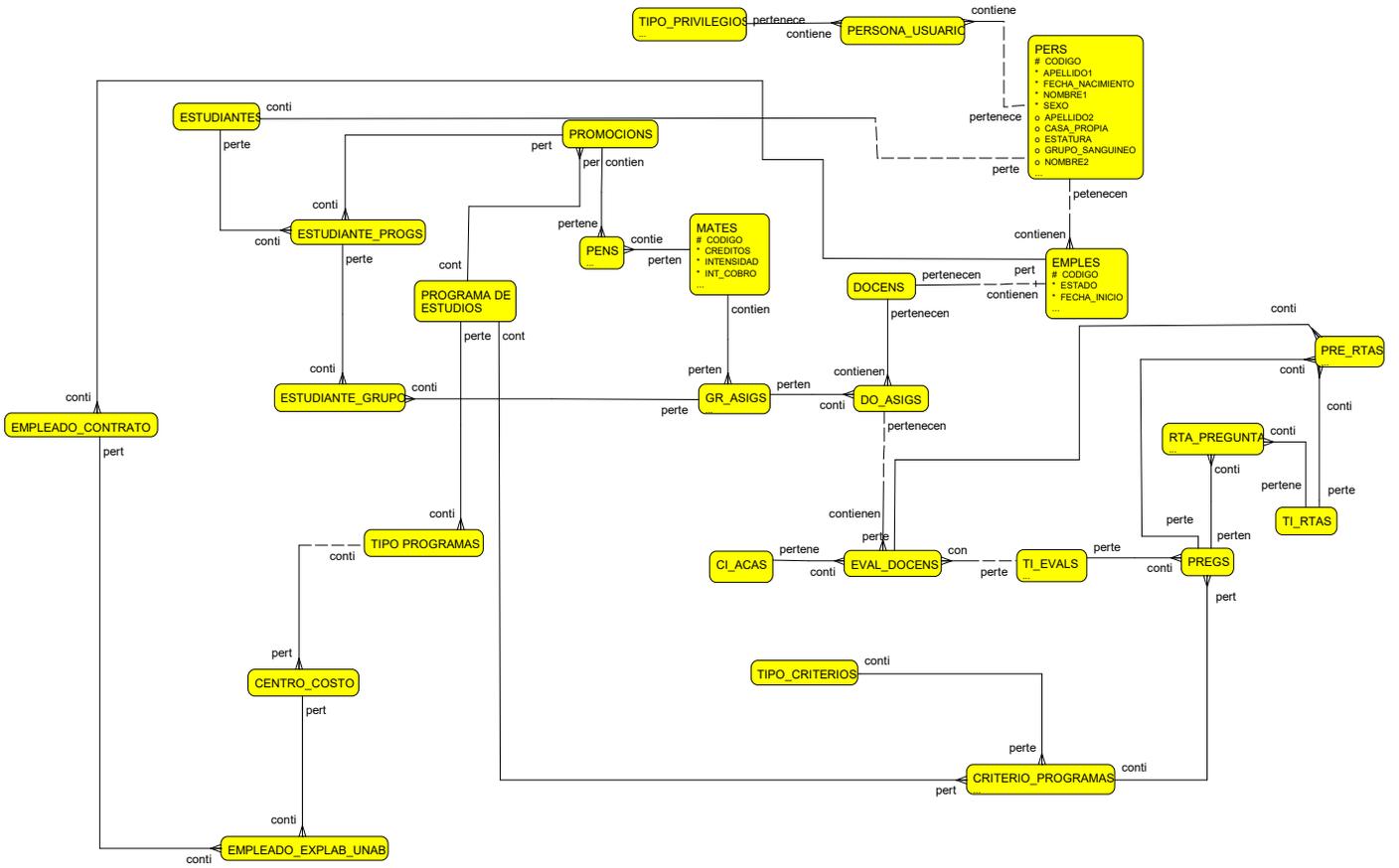
[41] GLOSARIO DE INTERNET [on line] [citado el 27 de agosto de 2003]. Disponible en Internet: [http://www.itver.edu.mx/comunidad/material/tallerbd/apuntes/1.2\\_Txt.htm](http://www.itver.edu.mx/comunidad/material/tallerbd/apuntes/1.2_Txt.htm)

[42] PRESSMAN, Op. Cit., p.343.

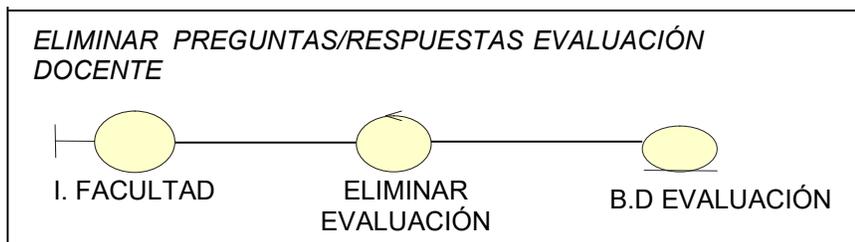
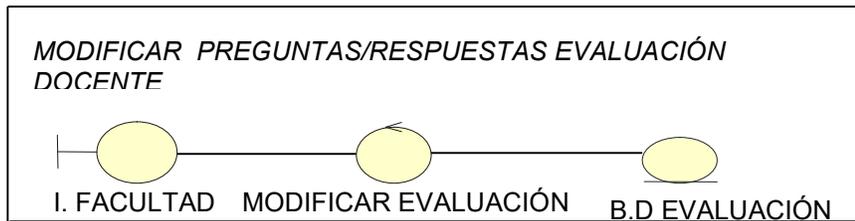
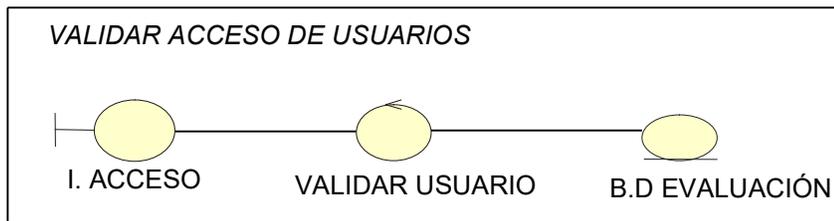
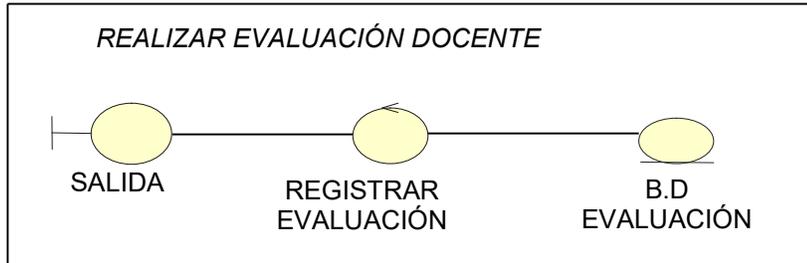
[43] PRESSMAN, Op. Cit., p.362.



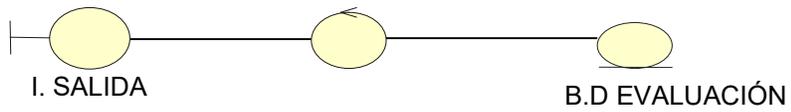
## ANEXO B. DER actualizado de la Evaluación Docente



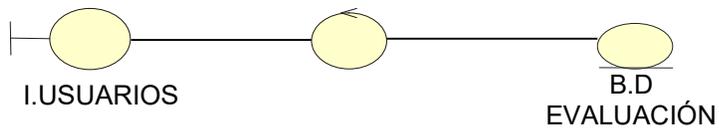
### ANEXO C. Diagrama de análisis de la Evaluación Docente



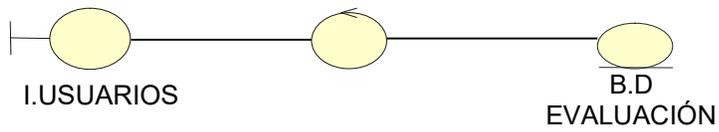
*CONSULTAR RESULTADOS EVALUACIÓN DOCENTE*



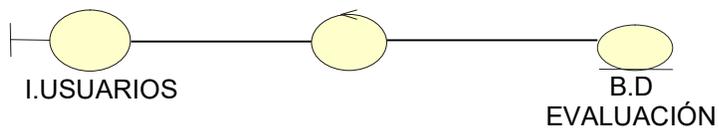
*CREAR USUARIOS*



*ELIMINAR USUARIOS*

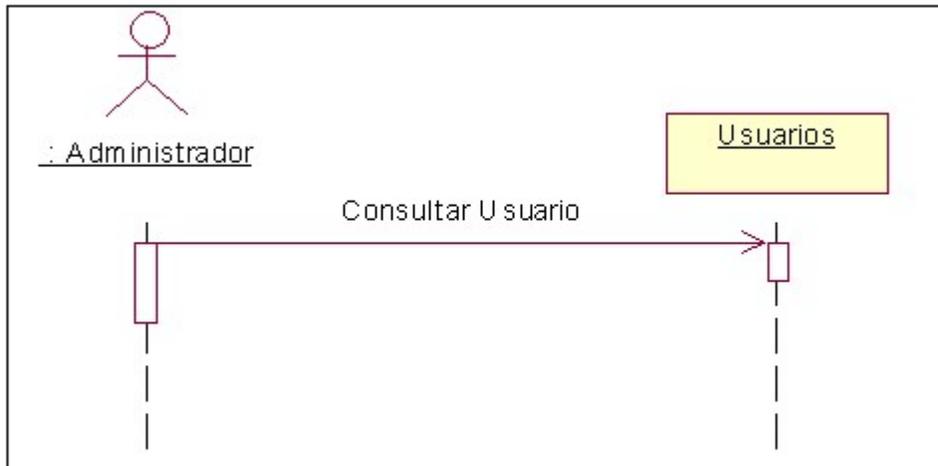


*CONSULTAR USUARIOS*

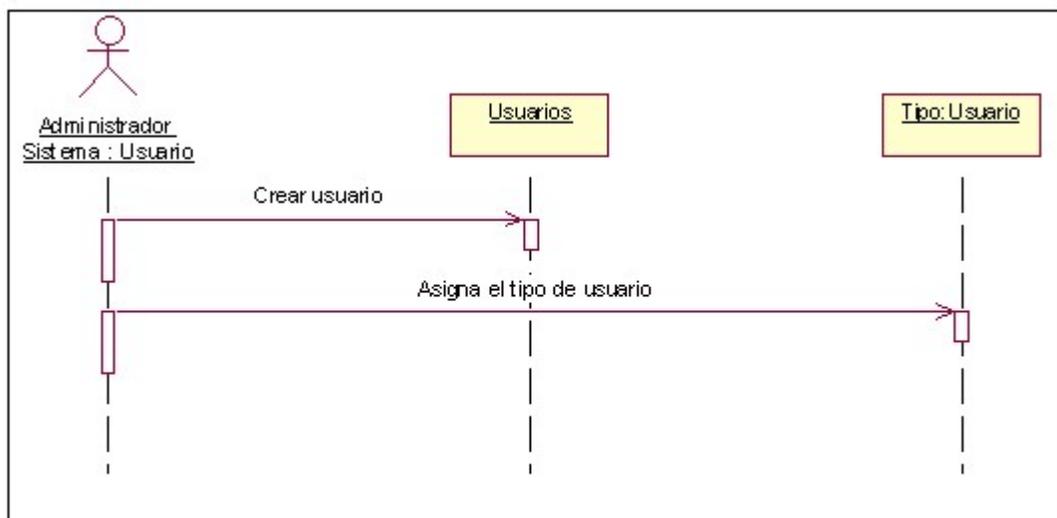


## ANEXO D. Diagramas de Secuencia de la Personalización

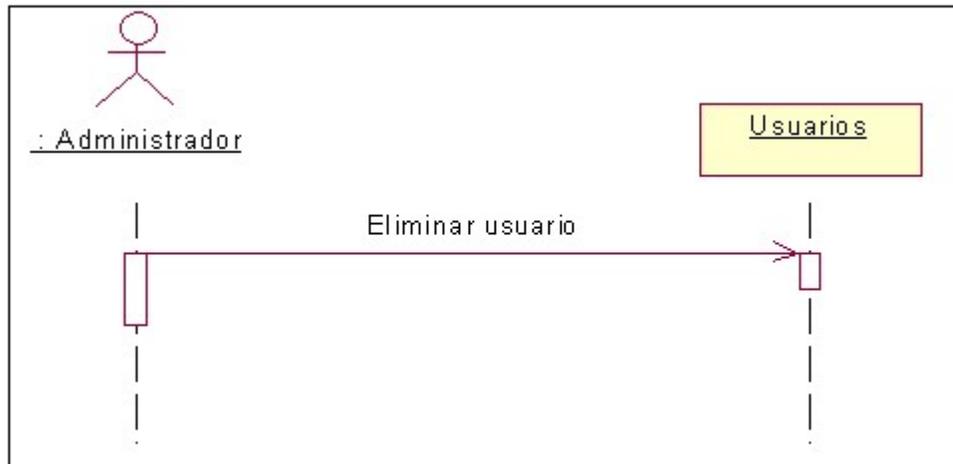
### Consultar Usuarios



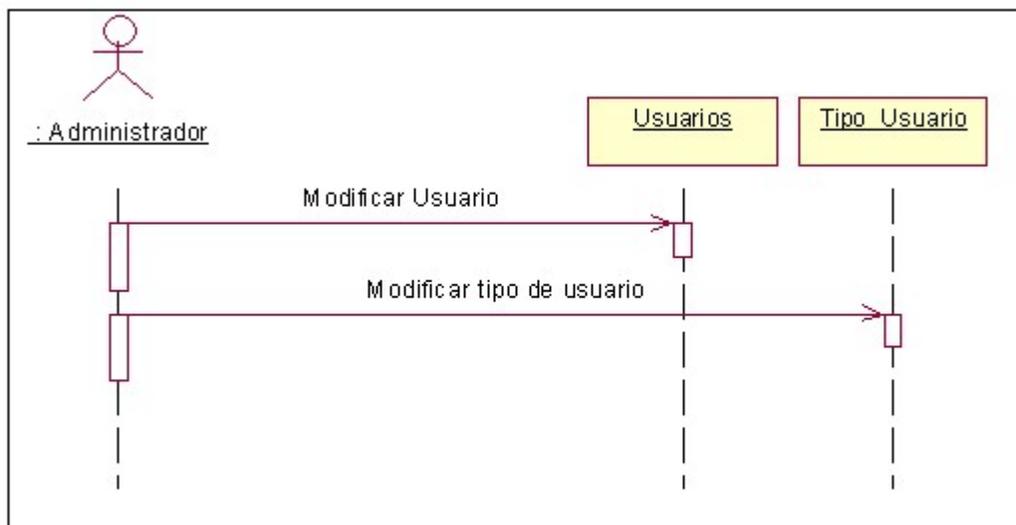
### Crear Usuario



## Eliminar usuario

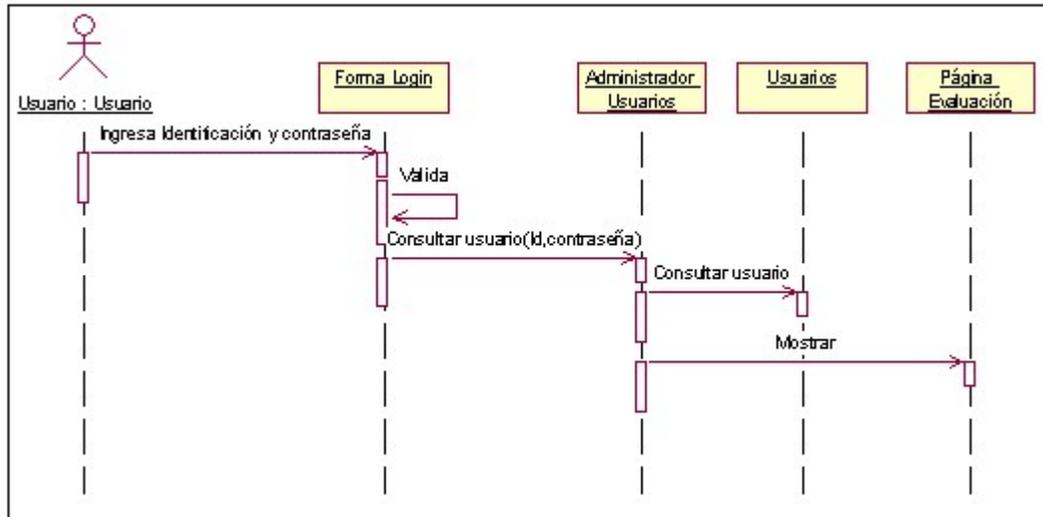


## Modificar-usuario

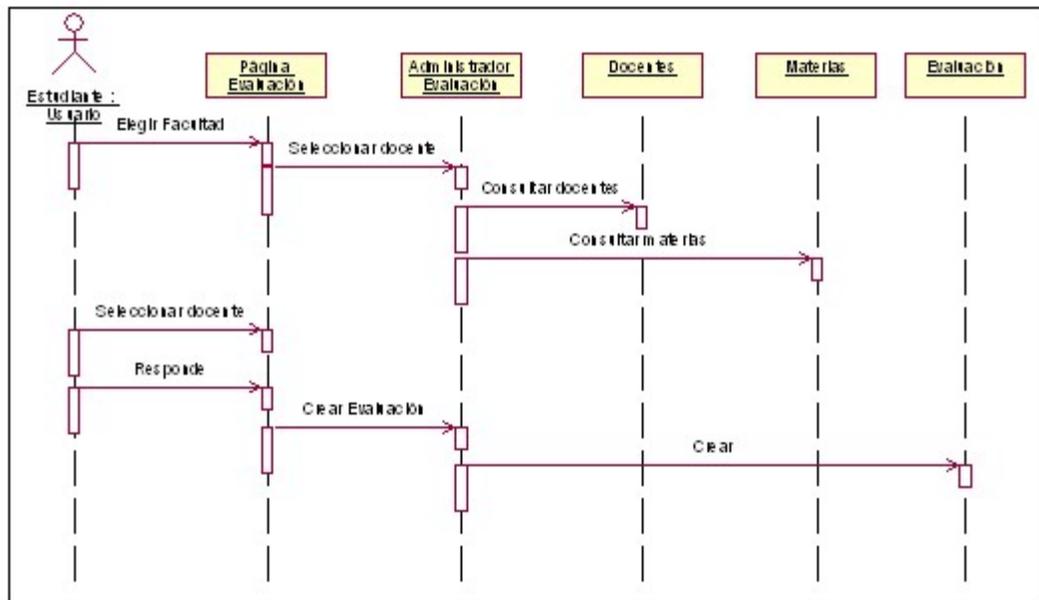


## ANEXO E. Diagramas de Secuencia de la Evaluación Docente.

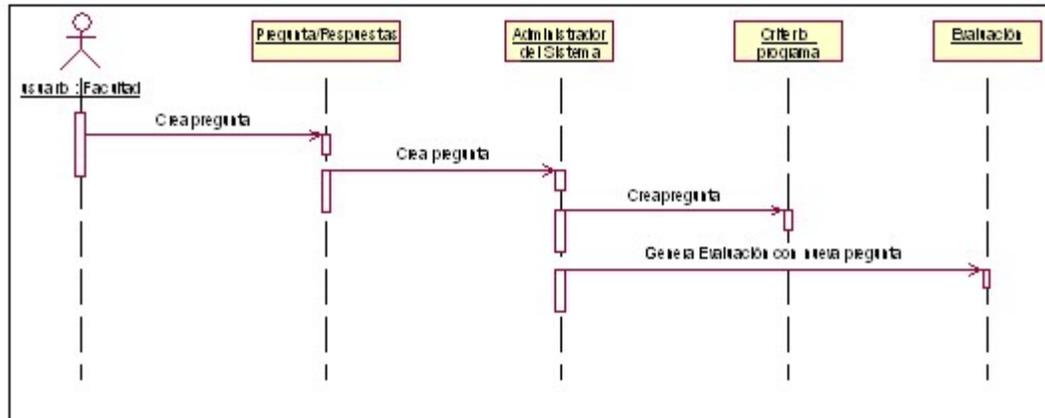
### Login



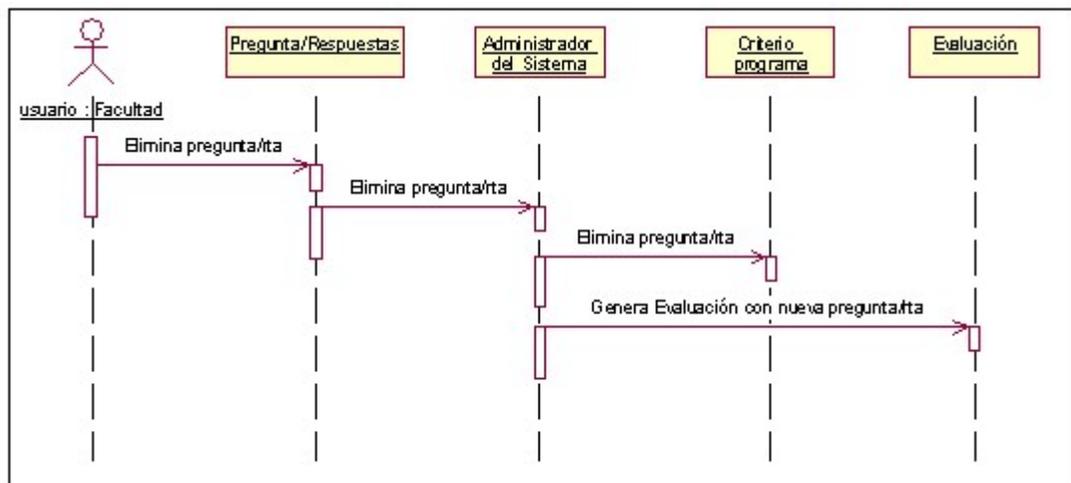
### Estudiante Realiza evaluación



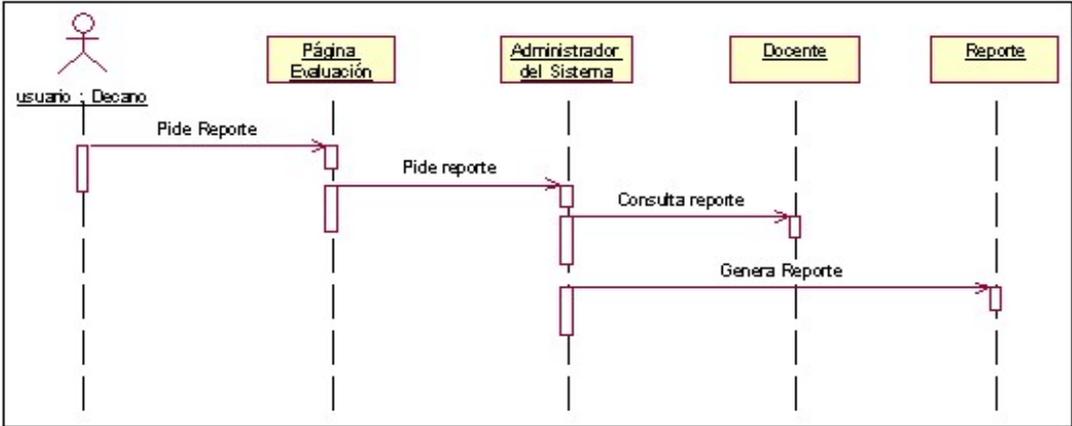
Facultad Crea pregunta-respuesta.



Facultad Elimina pregunta-respuesta

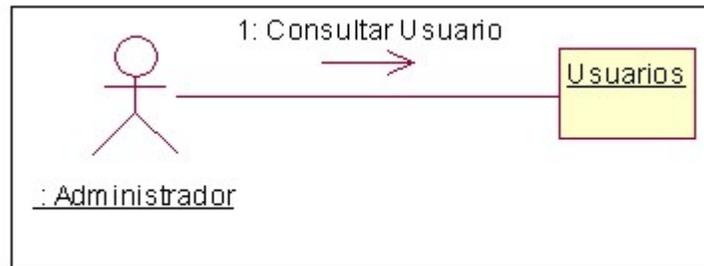


Decano pide reporte

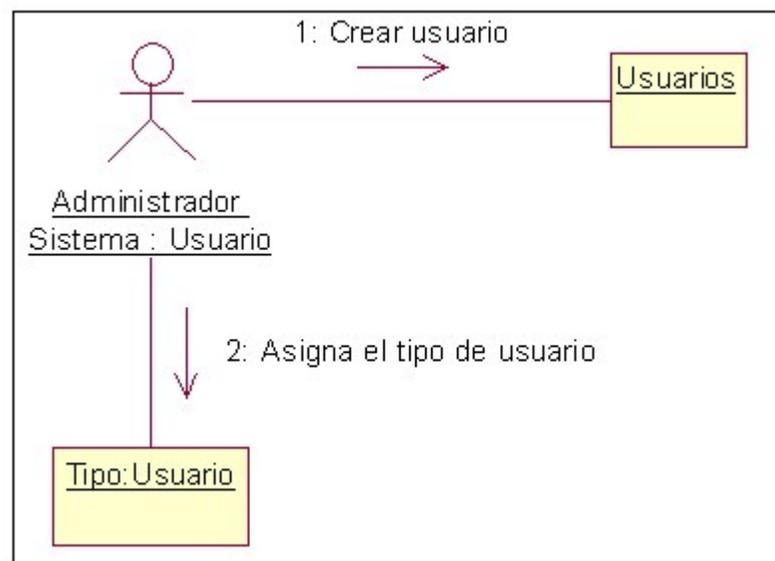


## ANEXO F. Diagramas de colaboración de la Personalización.

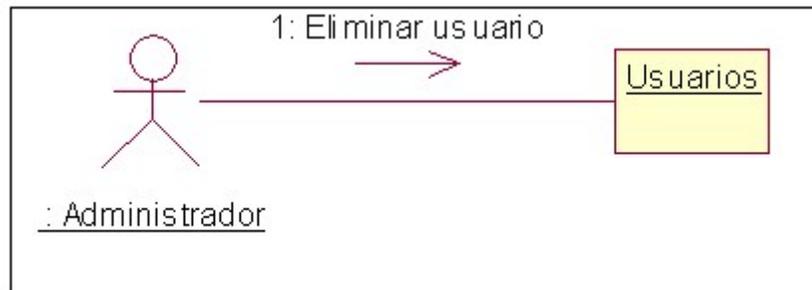
Consultar Usuario



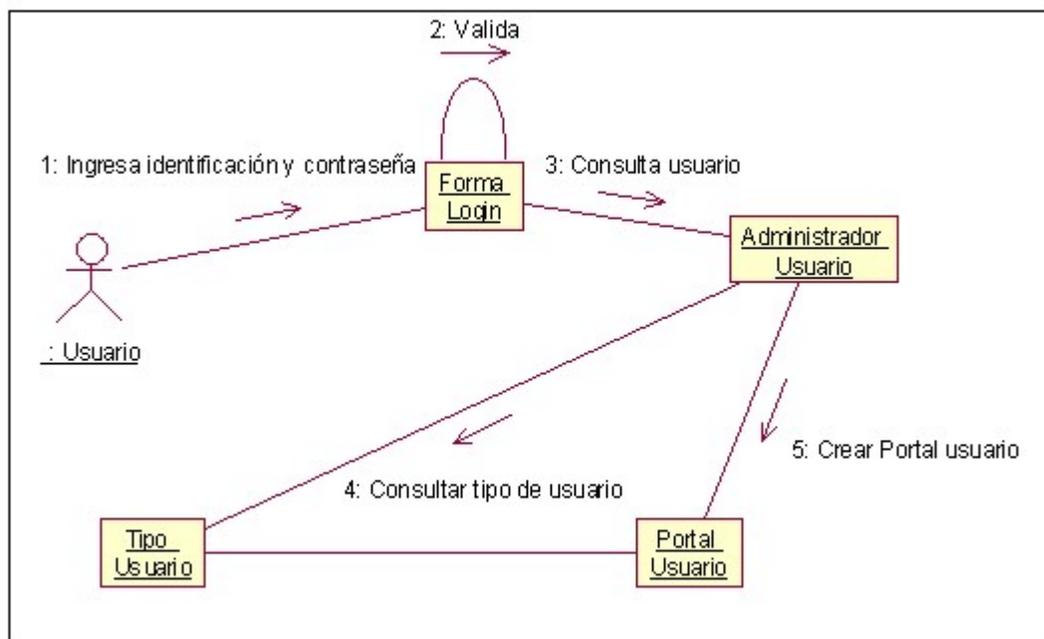
Crear-Usuario



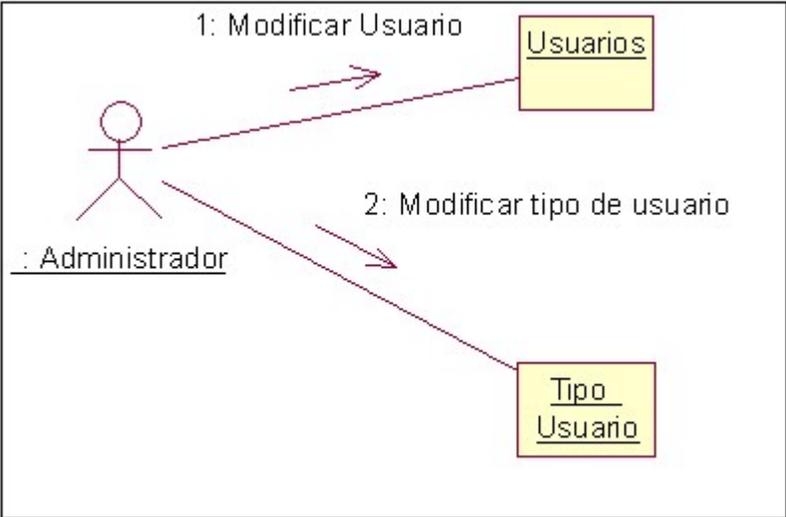
## Eliminar usuario



## Login

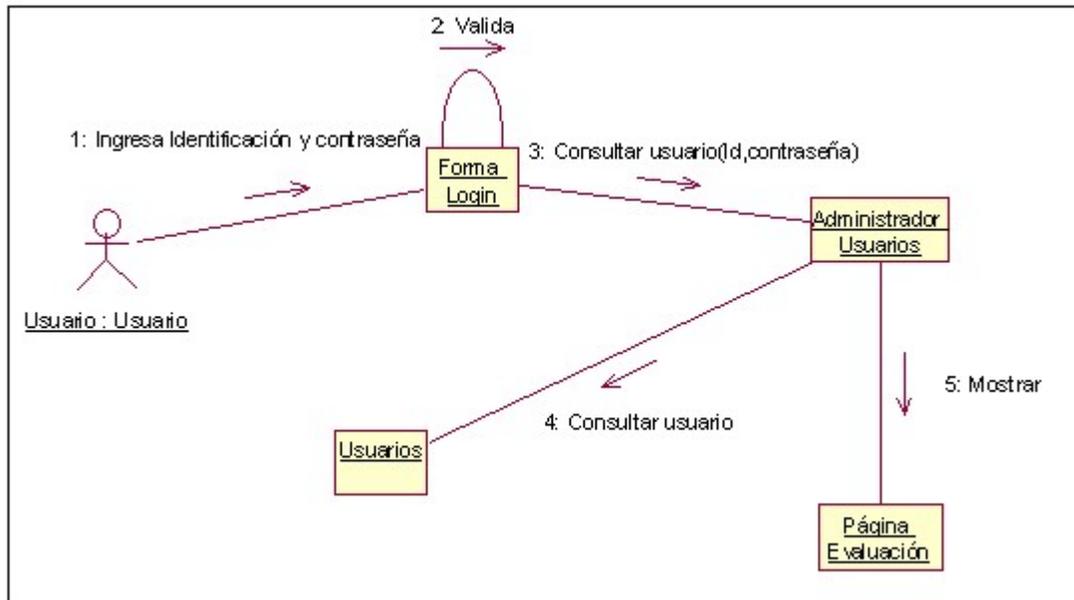


Modificar-usuario

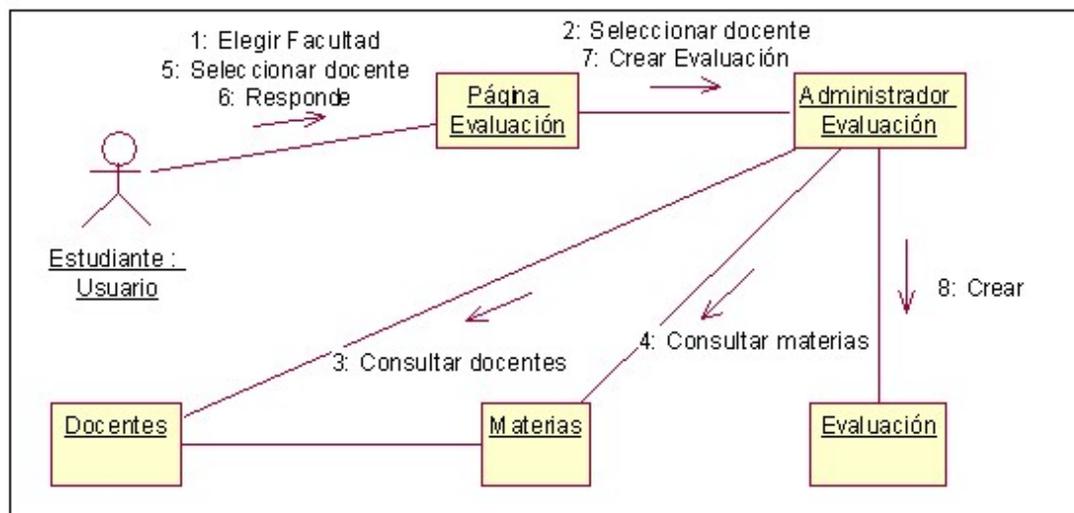


ANEXO G. Diagramas de colaboración de la Evaluación Docente.

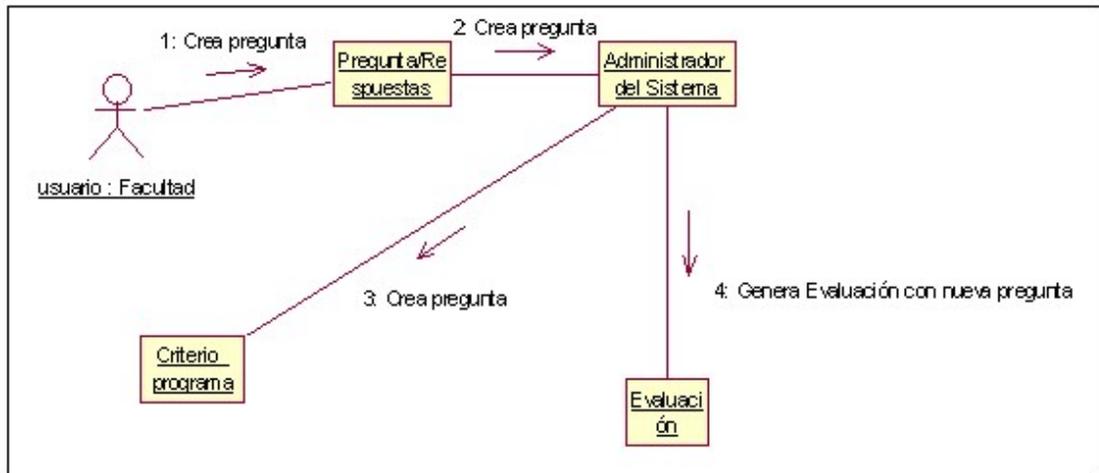
Login.



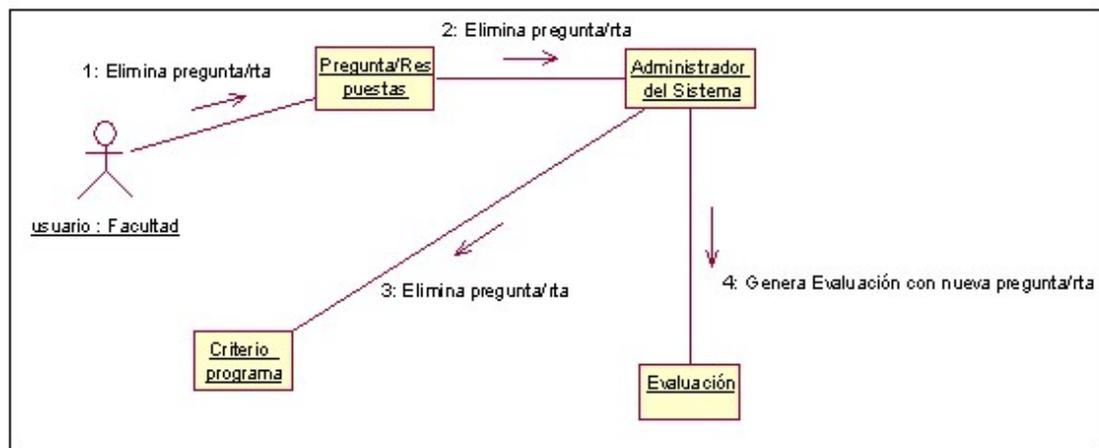
Estudiante Realiza Evaluación



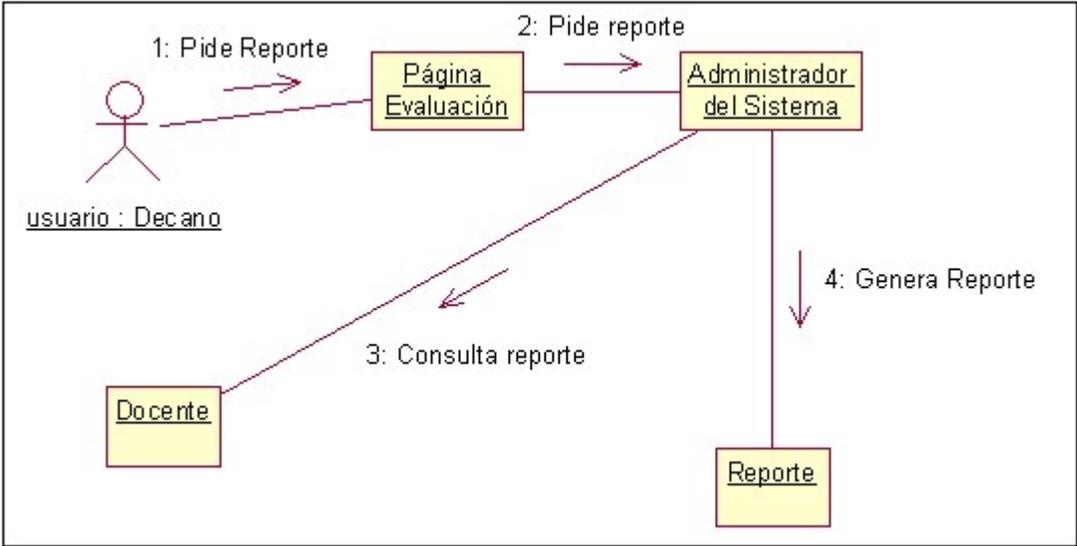
## Facultad Crea pregunta-respuesta.



## Facultad Elimina pregunta-respuesta



Decano pide reporte.



## **ANEXO H. Diccionario de datos de la Evaluación Docente.**

El Diccionario de Datos de la evaluación docente esta estructurado de la siguiente manera:

1. Ira sobre cada tabla el nombre de la entidad o tabla a la cual se va a hacer referencia. Este nombre ira escrito en Mayúsculas. Por ejemplo: Nombre de la Tabla: PERSONA.

2. Debajo del nombre de la entidad o tabla se encontrara una tabla que contiene las siguientes columnas:

Columna 1. Nombre de Atributos: En esta columna se encontraran los atributos o campos de las entidades, los cuales estarán escritos en mayúsculas.

Columna 2. Descripción: En esta columna ira una descripción de cada uno de los atributos o campos a los cuales sé hace referencia en la columna 1.

Columna 3. Tipo: En esta columna se escribe el tipo de dato que contendrá ese atributo o campo, que puede ser carácter, numérico, fecha, alfanumérico, etc.

Columna 4. Tamaño: En esta columna sé escribirá el número de caracteres que puede almacenar ese atributo o campo.

Columna 5. Ejemplo: En esta columna se dará un ejemplo del contenido que puede ser almacenado en ese atributo o campo.

**Tabla 3. Diccionario de Datos Evaluacion Docente**

<b>Nombre de la tabla : PERS(persona)</b>				
<b>Nombre Atributos</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Ejemplos</b>
CODIGO	Código de la Persona	Number	32	Dígitos (0-9)
NOMBRE1	Nombre de la Persona	Varchar2	256	Dígitos (1-2)
NOMBRE2	Nombre de la Persona	Varchar2	256	Letras del alfabeto A-Z
APELLIDO1	Apellido de la Persona	Varchar2	256	Letras del alfabeto A-Z
APELLIDO2	Apellido de la Persona	Varchar2	256	Letras del alfabeto A-Z
SEXO	Sexo de la Persona	Varchar2	256	M(masculino), F(Femenino)
FECHA_NACIMIENTO	Fecha en que nació la persona	Date		Fechas válidas
GRUPO_SANGUINEO	Grupo sanguíneo	Varchar2	32	Positivo, Negativo
RH	Factor RH de la Persona	Varchar2	256	A, B, AB, O
ESTATURA	Estatura de la Persona			
CASA_PROPIA	Si la Persona tiene Casa Propia o no	Varchar2	1	Si, No
PUBLICIDAD	Si la Persona desea recibir Publicidad o no	Varchar2	1	Si, No

<b>Nombre de la tabla : PERSONA_USUARIO</b>				
<b>Nombre</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Ejemplos</b>
CODIGO	Codigo identificador del Usuario	Number	5	Dígitos (0-9)
LOGIN	Nombre asignado al Usuario del sistema	Varchar2	10	

PASSWORD	Clave de acceso para el Usuario del sistema	Varchar2	10
----------	---	----------	----

Nombre de la tabla : ACTIVIDAD_EXT_LAB				
Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador de la Actividad Extra Laboral	Number	5	Dígitos (0-9)
ESTADO	Si se encuentra activo o no la Actividad Extra Laboral	Varchar2	1	Si, No
FECHA_INICIO	Fecha en que se comenzo con la Actividad Extra Laboral	Date		Fechas válidas
FECHA_FINAL	Fecha en que se termino con la Actividad Extra Laboral	Date		Fechas válidas
INTENSIDAD_HORARIA	Número de horas que le dedica a esa Actividad Extra Lab	Number	2	Dígitos (0-9)

Nombre de la tabla : CARGO_GENERAL				
Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador del Cargo General	Number	5	Dígitos (0-9)
NOMBRE	Nombre asignado al Cargo General	Varchar2	250	Administrativo, Docente, etc

Nombre de la tabla : DOCENS				
-----------------------------	--	--	--	--

Nombre	Descripción	Tipo	Tamaño	Ejemplos

Nombre de la tabla : DOCEN\_ASIGS

Nombre	Descripción	Tipo	Tamaño	Ejemplos

Nombre de la tabla : EMPLES

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Código del Empleado	Number	5	Dígitos (0-9)
FECHA_INICIO	Fecha en que se comenzo a ser Empleado UNAB	Date		Fechas válidas
FECHA_FINAL	Fecha en que termino de ser Empleado UNAB	Date		Fechas válidas
ESTADO	Si se encuentra activo o no como Empleado	Varchar2	1	Si, No
JORNADA	Jornada en la cual se encuentra laborando	Varchar2	1	Mañana, Tarde o Noche
PENSIONADO	Si es o no Pensionado	Varchar2	1	Si, No
SALARIO_INTEGRAL	Si cuenta con Salario Integral o no	Varchar2	1	Si, No

Nombre de la tabla : EMPLEADO\_CONTRATO

Nombre	Descripción	Tipo	Tamaño	Ejemplos
NO_CONTRATO	Número del Contrto del Empleado	Number	5	Dígitos (0-9)

FIRMA_CONTRATO	Si ya Firmo o no el Contrato	Varchar2	1	Si, No
----------------	------------------------------	----------	---	--------

**Nombre de la tabla :**

**EMPLEADO\_EXPLAB\_UNAB**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo asignado a la Experiencia en la UNAB	Number	5	Dígitos (0-9)
FECHA_INICIO	Fecha en que se comenzo a ser Empleado Administrativo	Date		Fechas válidas
FECHA_FINAL	Fecha en que termino de ser Empleado Administrativo	Date		Fechas válidas
CARGO_ESPECIFICO	Cargo Especifico que esta o desempeño	Varchar	250	Secretaria de la Facultad de Musica
ESTADO	Si se encuentra activo o no como Empleado Administrativo	Varchar2	1	Si, No
UNIDAD_BASICA	Cual es el Cento de costos Basico al cual pertenece	Varchar2	1	Facultade Derecho, Musica, etc

**Nombre de la tabla : ESTUDIANTE\_ASIG**

Nombre	Descripción	Tipo	Tamaño	Ejemplos

Nombre de la tabla : ESTUDIANTE_PROG				
Nombre	Descripción	Tipo	Tamaño	Ejemplos

Nombre de la tabla : GR_ASIGS				
Nombre	Descripción	Tipo	Tamaño	Ejemplos
ANO	Año del Grupo de esa Asignatura	Number	4	1999, 2000, etc
PERIODO	Periodo Academico	Number	2	Dígitos (0-9)
GRUPO	Nombre del Grupo	Vachar2	2	A, B, C, Aa, AA, etc

Nombre de la tabla : MATES				
Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador de la Materia	Number	5	
PES_CODIGO	Codigo identificador de la facultad	Number	32	

Nombre de la tabla : PENS				
Nombre	Descripción	Tipo	Tamaño	Ejemplos
NO_CAMBIO	Número de cambios de el Pensum	Number	2	Dígitos (0-9)
NIVEL	Nivel en el que se encuentra una materia en el	Varchar2	1	Primero, Segundo, Tercero, etc

	Pensum			
FECHA_CAMBIO	Fecha en la que se realizo el cambio del Pensum	Date		Fechas válidas
ESTADO	Si se encuentra activo o no el Pensum	Varchar2	1	Si, No

**Nombre de la tabla: PREGS**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador de la Pregunta	Number	5	Dígitos (0-9)
CONTENIDO	Contenido de la Pregunta	Varchar2	250	Contenido ¿?

**Nombre de la tabla: PREG\_RTA**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
FECHA_INICIO	Fecha en que se respondió la Evaluación	Date		Fechas válidas
ESTADO	Estado de la Respuesta Escogida y de la Pregunta	Varchar2	1	Activa, Inactiva

**Nombre de la tabla: PROMOCIONS**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador de la Promoción	Number	5	Dígitos (0-9)

**Nombre de la tabla: TIPO\_CAR\_GENERAL**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador del Tipo de Cargo General	Number	5	Dígitos (0-9)
NOMBRE	Nombre asignado al Tipo de Cargo General	Varchar2	250	Sercretaria General de Vicerrectoria

**Nombre de la tabla : TIPO\_CONTRATO**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador del Tipo de Contrato	Number	5	Dígitos (0-9)
NOMBRE	Nombre asignado al Tipo de Contrato	Varchar2	250	

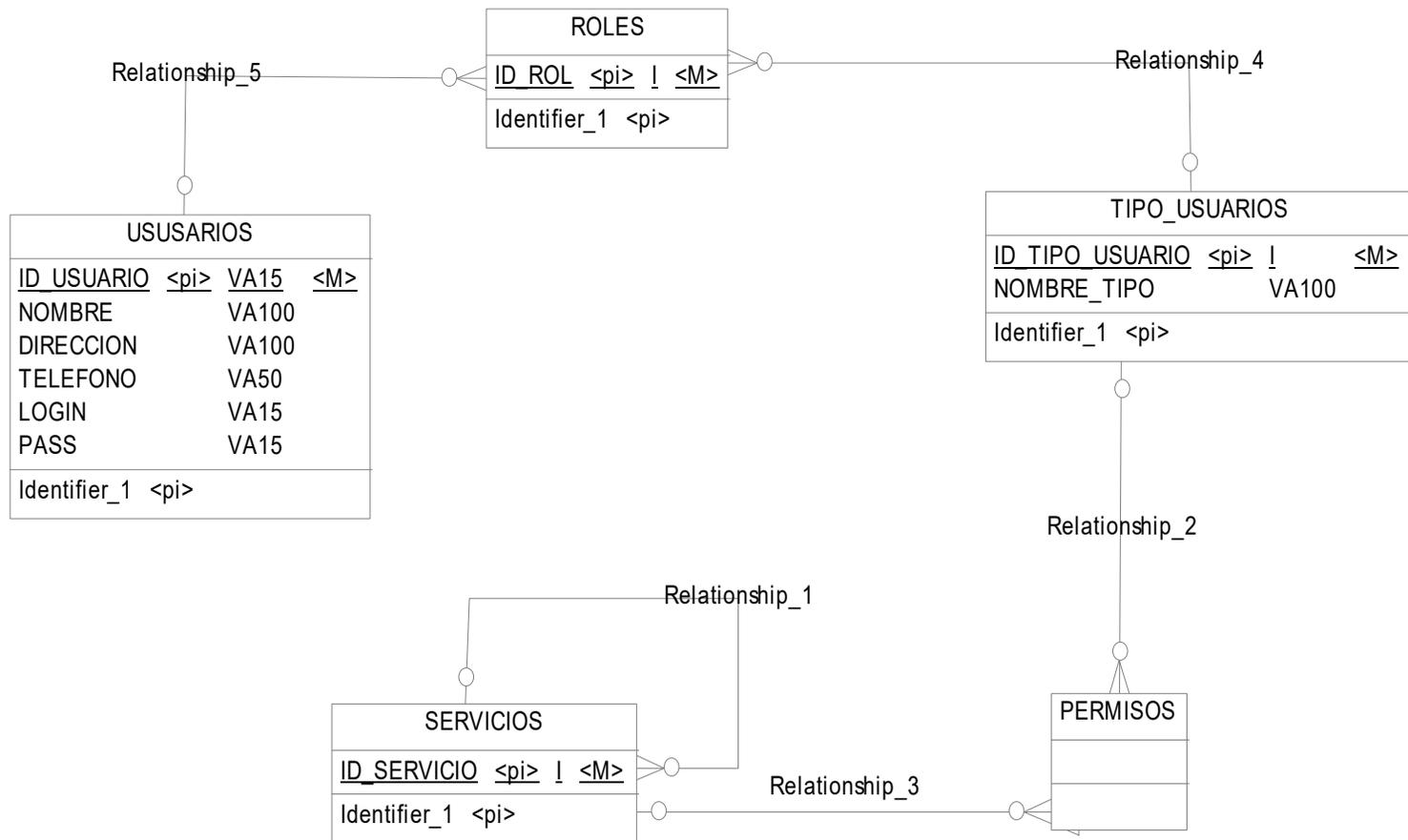
**Nombre de la tabla : TI\_EVALS**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador del Tipo de Evaluación	Number	5	Dígitos (0-9)
NOMBRE	Nombre asignado al Tipo de Evaluación	Varchar2	250	HETEROEVALUACION

**Nombre de la tabla : TI\_RTA**

Nombre	Descripción	Tipo	Tamaño	Ejemplos
CODIGO	Codigo identificador del Tipo de Respuesta	Number	5	Dígitos (0-9)
NOMBRE	Nombre asignado al Tipo de Respuesta	Varchar2	250	1, 2, 3, etc

## ANEXO I. DER del sistema personalización



## ANEXO J. RESUMEN DE REUNIONES

1. IDENTIFICACIÓN DE NECESIDADES. Es el primer paso del análisis del sistema en este proceso el analista se reúne con el cliente y/o usuario, e identifican las metas globales, se analizan las perspectivas del cliente, sus necesidades y requerimientos.

En esta etapa se procedió a realizar reuniones con los diferentes coordinadores de la evaluación docente o decanos de las diferentes facultades hasta ahora solo hemos logrado realizar citas con la Vicerrectora Académica, Graciela Moreno y con las facultades de Comunicación Social y su respectivo decano el doctor Rodrigo Velasco quienes realizaron una prueba piloto de la evaluación docente con todos los lineamientos institucionales como lo son la coevaluación, heteroevaluación y una Jefe evaluación, pero la realizaron manualmente, con una calificación de desempeño de 0-100, que según estándares formados por la facultad se tomaron en cuenta para el mejoramiento de la docencia. Las diferentes evaluaciones tienen un porcentaje diferente, por ejemplo la de los colegas vale un 25%, la de los estudiantes 25%, y la del jefe 50%. Esta información fue tabulada por una de las secretarias de la facultad.

Luego tuvimos una reunión el 6 de Junio con la Dra. Alba Rosa decana de la facultad de Educación, donde nos encontramos que realizan la evaluación también manualmente y la decana se encargaba de analizar evaluación por evaluación y tener reuniones personales con cada uno de los docentes. También tuvimos un encuentro con la facultad de Ingeniería de Mercados y su respectivo decano el Dr. Torres y pudimos encontrar que esta facultad siempre ha tenido una relación estrecha con la facultad de Ingeniería de Sistemas, pero debido a que el administrador de la evaluación

docente se había marchado y se presentaron muchos inconveniente para realizar la evaluación docente, se tomó la decisión de no hacerla este semestre que terminó.

De todas las reuniones las conclusiones que pudimos sacar son:

- Que se necesita una evaluación estándar para implementarla en todas las facultades.
- Que se presentan inconvenientes al momento de realizar la evaluación docente vía Internet, ya que a los alumnos les da pereza realizarla debido a que las preguntas son muy largas y muchas, además ellos sacan alguna excusa para demorarse en el proceso de evaluación para perder clases.
- Los profesores están reacios a llevar a los estudiantes a realizar la evaluación docente debido a que ha habido pérdida de tiempo.
- No hay confidencialidad en la información.
- A las personas les molesta que los evalúen entonces hay que buscar estrategias para incentivar a realizar esta evaluación (Comunicación Social)