

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 9**



Rediseño del Directorio de Personas de la Universidad de las Ciencias Informáticas

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO
EN CIENCIAS INFORMÁTICAS**

**AUTORES: Dayanis Elvia Alcantara Rabí
Alexander López Gouyonnet**

TUTORA: Lic. Isachi Abreu Gil

CO-TUTORA: Lic. Gretchen Guillermo Hernández

CONSULTOR: Ing. Tomás López Jiménez

ASESOR: Ing. Manuel Alejandro Gil Martín

**Ciudad de La Habana, julio del 2007
“Año 49 de la Revolución”**

“Los conceptos y principios fundamentales de la ciencia son invenciones libres del espíritu humano”

Albert Einstein

DEDICATORIA

A mi mamá y a mi abuela por su amor, dedicación y esfuerzo. Sin ellas no hubiera llegado hasta aquí. Las adoro, gracias por estar siempre a mi lado y hacer de mi cada día una mejor persona.

A mi tío Osmani por estar siempre ahí.

A mi papá y a mi hermanito.

A mi familia por confiar en mí.

Dayanis Elvia Alcantara Rabí

A mis padres y hermanos por su amor y dedicación, por haberme ayudado a ser lo que soy.

A toda mi familia.

A todos mis amigos.

Alexander López Gouyonnet

AGRADECIMIENTOS

A toda mi familia.

A mis padres, por su apoyo durante toda mi vida.

A mis hermanos, Aniuska y Enardo.

A todos mis amigos, por la ayuda prestada.

A la Universidad de las Ciencias Informáticas por darme la posibilidad de formar parte de esta generación del futuro.

Alexander López Gouyonnet

A mi tío (papá) Osmani por su amor y dedicación en todo el transcurso de mi vida.

A mi mamá Aurora por apoyarme y guiarme siempre.

A mi abuelita querida Elvia.

A mi papá.

A mi hermanito lindo Frank, te quiero mucho.

A mi novio por su amor y confianza.

A mis vecinos por confiar siempre en mí.

A mi tutora Isachi y al asesor Manuel Alejandro por su dedicación.

A mis grandes amigos pinareños Alexander, Johann, Reinel, Yarisel y Yordanys.

A mis amigas de siempre Natalí y Damaris.

A mis viejos amigos de la vocacional de Camaguey.

A mis compañeros de aula por haber compartido estos 5 años y hacer que cada día en la universidad sea inolvidable.

A todos lo que contribuyeron de una forma u otra en la realización de este trabajo, Rafael, Darlyn, Aliosmi, Yadira, Ricardo (El bala) por su paciencia, Duany.

A todos los profesores que contribuyeron en mi formación profesional.

A la Universidad de las Ciencias Informáticas por darme la posibilidad de formar parte de esta generación del futuro.

A la Revolución Cubana por haberme permitido estudiar.

Dayanis Elvia Alcantara Rabí

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Departamento de Informatización de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2007

Dayanis Elvia Alcantara Rabí

Alexander López Gouyonnet

Lic. Isachi Abreu Gil

DATOS DE CONTACTO

Síntesis de la Tutora Lic. Isachi Abreu Gil

Profesión: Licenciada en Ciencias de la Computación

Categoría docente: Adiestrado

Años de graduada: 2

Síntesis de la Co-Tutora Lic. Gretchen Guillermo Hernández

Profesión: Licenciada en Contabilidad y Finanzas

Categoría docente: Adiestrado

Años de graduada: 1

Síntesis del Consultor Ing. Tomás López Jiménez

Profesión: Ingeniero en Informática

Categoría docente: Especialista Superior

Años de graduada: 37

Síntesis del Asesor Ing. Manuel Alejandro Gil Martín

Profesión: Ingeniero en Informática

Categoría docente: Adiestrado

Años de graduado: 2

RESUMEN

En la Universidad de las Ciencias Informáticas (UCI) la gestión de la información de las personas que pertenecen a este centro es uno de los problemas que presenta la universidad hoy día. Esto se debe a que la información se encuentra distribuida en las diferentes bases de datos que posee la universidad. Para acceder a dicha información en la universidad existe implementado un Directorio de Personas del cual hace uso toda la comunidad universitaria. Este sistema de búsqueda presenta algunas dificultades, tales como que no es capaz de realizar consultas múltiples y complejas, tiene problemas para la sincronización de datos, así como una gran dependencia de otros sistemas, entre otros. Una vez analizado todo lo antes mencionado se hace necesario rediseñar el Directorio de Personas con nuevas funcionalidades y diseñar una base de datos para prestar su servicio conjunto a una aplicación que se encargue de la actualización de los datos que maneja el directorio. Para dicha actualización empleará servicios web. De esta manera se logra un mejor dominio de la información de las personas en la universidad y prestar un mejor servicio de búsqueda.

Palabras claves:

- ✓ Gestión de información
- ✓ Directorio de Persona
- ✓ Aplicación
- ✓ Base de Datos

TABLA DE CONTENIDOS

INTRODUCCIÓN	9
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	15
1.1 Introducción.....	15
1.2 Herramientas de búsquedas.....	15
1.2.1 Metabuscadores.....	15
1.2.2 Buscadores automáticos o motores de búsqueda.....	16
1.2.3 Buscadores basados en directorios.....	17
1.3 Recuperación de Información.....	18
1.4 Aplicación.....	19
1.5 Base de Datos.....	19
1.5.1 Tipos de base de datos.....	21
1.5.2 Normalización de bases de datos.....	23
1.5.3 Transacciones, concurrencia y recuperación.....	26
1.6 Objeto de Estudio.....	30
1.6.1 Descripción General.....	30
1.6.2 Descripción actual del dominio del problema.....	31
1.6.3 Situación problemática.....	32
1.7 Sistemas automatizados vinculados con el campo de acción.....	33
1.8 Fundamentación del objetivo general.....	34
1.9 Propuesta de solución.....	34
1.10 Conclusiones.....	35
CAPITULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR	36
2.1 Introducción.....	36
2.2 Las Tecnologías de la Información y las Comunicaciones (TIC).....	36
2.3 Aplicaciones Web vs. Sitios Web.....	37
2.4 Aplicación de escritorio.....	39
2.5 Servicios Web.....	40
2.6 Modelo Cliente/Servidor.....	41
2.7 Lenguajes de programación para la web.....	43

2.8 Plataforma.Net	49
2.8.1 La aplicación VisualStudio.Net 2005 como software de autor.....	50
2.8.2 Lenguaje de programación a utilizar: C#.....	51
2.9 Sistema de Gestión de Base de Datos (SGBD).....	51
2.10 El Lenguaje Unificado de Modelado como soporte de la modelación de la solución propuesta. ..	55
2.11 El Proceso Unificado de Desarrollo de Software como base en el desarrollo de la solución.....	56
2.12 Rational Rose como herramienta de trabajo.....	58
2.13 Conclusiones.....	59
CAPITULO 3. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.....	60
3.1 Introducción.....	60
3.2 Modelo Conceptual.....	60
3.3 Requerimientos Funcionales.....	62
3.4 Requerimientos No Funcionales.....	65
3.5 Descripción del Sistema Propuesto.....	66
3.5.1 Descripción de los actores.....	67
3.5.2 Diagrama de Casos de Uso del Sistema.....	68
3.6 Conclusiones.....	80
CAPÍTULO 4. CONTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	81
4.1 Introducción.....	81
4.2 Diagramas de clases de diseño.....	81
4.3 Diseño de la base de datos.....	90
4.3.1 Diagrama de clases persistentes.....	90
4.3.2 Modelo de datos.....	91
4.4 Principios de diseño.....	92
4.4.1 Estándares de la interfaz de aplicación.....	92
4.4.2 Formato de reportes.....	93
4.4.3 Concepción de la ayuda.....	93
4.4.4 Tratamiento de excepciones.....	93
4.5 Estándares de la codificación.....	94
4.6 Modelo de despliegue.....	95

4.7 Conclusiones.....	96
CAPITULO 5. ESTUDIO DE FACTIBILIDAD	97
5.1 Introducción.....	97
5.2 Planificación basada en casos de usos.....	97
5.3 Beneficios tangibles e intangibles.....	99
5.3.1 Beneficios tangibles.....	99
5.3.2 Beneficios intangibles.....	99
5.4 Análisis de costos y beneficios.....	99
5.5 Conclusiones.....	100
CONCLUSIONES.....	101
RECOMENDACIONES.....	102
REFERENCIAS BIBLIOGRÁFICAS	103
BIBLIOGRAFÍA	104
GLOSARIO DE TÉRMINOS.....	106
ANEXOS	108

INTRODUCCIÓN

Desde el surgimiento de Internet, la gran Red de redes, la necesidad de obtener información ha marcado una gran pauta en los servicios de búsqueda. La World Wide Web ha pasado de ser un medio para la publicación de información y contenidos, a convertirse en una plataforma para el diseño y desarrollo de aplicaciones informáticas distribuidas. Dado su crecimiento en forma rápida y exponencial se vio en la necesidad de tener un orden, herramienta o algún tipo de clasificación de las páginas y servicios de la web. En este ámbito surgen los Buscadores y Directorios, herramientas capaces de ayudar a localizar y acceder a toda la información distribuida y compartida en la red.

En la actualidad, en el modelo de aplicación web tradicional encontramos una importante limitación: la interacción comienza y termina en dos puntos claramente definidos: la petición del usuario y la respuesta de la aplicación informática. La vía mas efectiva es acudir a un buscador, esta herramienta permite encontrar información detallada e inmediata dentro de la gran gama de bases de datos.

Los directorios como herramientas de búsqueda de Internet organizado por categorías y temas jerárquicamente hasta llegar a la Información deseada por el usuario, agrupan los archivos de datos dependiendo de su contenido y del criterio que decida el usuario. Estos permiten una mayor efectividad en la identificación de la información e incluyen otros elementos que hacen que el resultado de la búsqueda sea más relevante, por ejemplo, gráficos y fotos. En un campo muy amplio se ve el valor que ha alcanzado la información como recurso imprescindible para cualquier organismo y la necesidad de almacenamiento de la misma.

Los sistemas de búsquedas de datos de usuarios y personas en las distintas instituciones están sujetos a diversas situaciones que están dadas por la disponibilidad del sistema. Esto esta relacionado con las Bases de Datos, sus gestores y la cantidad de datos que se puedan manejar.

El manejo de la información es una parte fundamental de cualquier entidad para la realización de sus actividades de manera eficiente. Con los adelantos tecnológicos en el área de la informática las empresas han aprovechado los beneficios que aportan los sistemas computacionales para el procesamiento de la información en forma rápida y confiable. Esto en gran medida ha permitido que reconozcan la importancia de una adecuada administración de la información con el fin de lograr la máxima utilidad de la misma.

Para el almacenamiento de estos volúmenes de datos se han desarrollado sistemas informáticos adaptables a las necesidades de cualquier empresa. En este caso se hace referencia a las de base de datos que facilitan la manipulación y el acceso a la información almacenada, además de brindar medidas de seguridad que permiten mantener la integridad de esta. Trayendo esto un gran aumento en la demanda de estos sistemas.

La necesidad de obtener información ha marcado una gran pauta en los servicios de búsqueda de forma rápida y exponencial. Esta gran necesidad de consumir información ha permitido que la mayoría de las instituciones recurran al uso de los llamados servicios web por las ventajas que trae consigo la utilización de los mismos para lograr un mejor manejo de la información e interoperabilidad entre aplicaciones de software, independientemente de sus propiedades o de las plataformas sobre las que se instalen.

La Universidad de las Ciencias Informáticas (UCI) es una de esas instituciones de las que hacen buenas prácticas del manejo de información por medio de servicios web. La misma presta varios servicios búsqueda mediante los cuales se nutre toda la comunidad universitaria. Uno de los servicios más solicitados de este centro estudiantil lo constituye el de búsqueda de información de personas. Para ello emplea varios mecanismos de almacenamiento y estrategias de búsquedas de la misma.

En la universidad existe un Directorio de Personas asociado con las distintas Bases de Datos existentes en el centro, por ejemplo Akademos, Ciudadano, Trabajadores, en las que se encuentran registrados todos los estudiantes, trabajadores y profesores acreditados en el centro. Este sistema de búsqueda presenta algunas deficiencias al no satisfacer todas las necesidades del usuario y carecer de ciertas funcionalidades importantes además de no contar con una base de datos propia y un sistema para la

actualización de los datos de las personas. Algunas de estas insatisfacciones se ven reflejadas por las siguientes cuestiones:

- Respuesta al usuario demasiado lenta.
- Dificultad para realizar consultas múltiples o complejas.
- Distribución de la información de las personas en distintas bases de datos.
- Dificultad del sistema para la sincronización de datos.
- Dependencia de la base de datos Ciudadano para cualquier operación.
- Poca independencia entre los sistemas.

La recuperación de la información de las personas y el acceso a la misma en la universidad es una tarea compleja. Esto se debe al gran volumen de información que se maneja, además de que la misma no está organizada de la forma más propicia para acceder a ella, que como ya habíamos mencionado anteriormente se encuentra distribuida en las distintas bases de datos que posee la universidad.

En función de lo planteado anteriormente el **problema a resolver** radica en ¿Cómo perfeccionar el proceso de gestión de la información de las personas en la universidad?

Por tanto, se define como **objeto de estudio** de esta investigación la gestión de información de las personas en la Universidad de las Ciencias Informáticas.

De ahí que el **campo de acción** que abarca este trabajo sean los procesos de gestión de la información de las personas en la universidad.

Con la finalidad de dar solución al problema en cuestión el **objetivo general** de este trabajo es rediseñar el Directorio de Personas, diseñar una base de datos y una aplicación para la actualización de los datos de las personas en la Universidad de las Ciencias Informáticas.

Para darle cumplimiento al objetivo propuesto se plantea la siguiente **idea a defender**:

Con el rediseño del Directorio de Personas, el diseño de una base de datos y la modelación de una aplicación de actualización, se tendrá un mejor dominio de la información de las personas en la Universidad, los sistemas serán más independientes y podrán tener vida propia. De esta manera se prestará un mejor servicio de búsqueda de información de usuarios.

Para alcanzar el objetivo propuesto y darle solución a dicha problemática se trazaron las siguientes **tareas**:

- Realizar un estudio de la información de los usuarios en la Universidad y su situación actual.
- Análisis de soluciones similares.
- Estudio de posibles tecnologías a utilizar.
- Rediseñar el Directorio de Personas existente.
- Modelar una Base de Datos que soporte las funcionalidades del nuevo Directorio de Personas.
- Diseñar una aplicación capaz de extraer datos de otros sistemas mediante servicios web haciendo uso del lenguaje de programación C#.

Para el desarrollo de este trabajo se utilizaron como **métodos científicos** de la investigación los que a continuación se hacen referencia.

Se empleó como **métodos teóricos** el histórico-lógico con el fin de investigar sobre las aplicaciones informáticas de este tipo desarrolladas en la universidad, Cuba y el resto del mundo. Su uso en el contexto universitario, así como las ventajas y desventajas de este sistema. El **hipotético-deductivo** en la formulación del planteamiento hipotético lo que nos permitió establecer predicciones a partir del sistema de conocimientos mediante la aplicación de reglas lógicas de la deducción.

Como **métodos empíricos** se utilizó la entrevista a especialista del Departamento de Informatización para recoger toda la información posible acerca de las funcionalidades que debe tener el sistema y de cómo está estructurada la información dentro de la universidad. Se consultó a especialistas del área de gestión académica (Akademos) y del Departamento de Capital Humano. También se utilizan **métodos estadísticos** en el análisis de los costos que generará el desarrollo de la aplicación.

Después de darle cumplimiento a todas las tareas planteadas en función a resolver el problema antes expuesto se espera que la investigación sirva como guía o como material de estudio para futuros sistemas con fines similares en otras instituciones.

Para lograr una mejor comprensión el presente documento se estructura en cinco capítulos de contenidos donde se incluye todo lo relacionado con el trabajo investigativo realizado.

Capítulo 1: Fundamentación Teórica.

En este capítulo se analizan los conceptos asociados al dominio del problema y se describe en sentido general el objeto de estudio. Se describen algunos sistemas automatizados existentes en la universidad vinculados al campo de acción y se plantea una propuesta de solución en función de la situación problemática.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

Se analizan las características más significativas de las principales tecnologías a considerar para la modelación del sistema que se pretende llevar a cabo. Además se detallan algunas características del lenguaje a utilizar en la elaboración del mismo y se explica el porqué de la metodología utilizada.

Capítulo 3: Presentación de la solución propuesta.

Se describe la situación a través de un modelo de dominio, y a partir de esto se comienza a hacer el análisis del sistema a diseñar. Se definen las funcionalidades del sistema y se describen detalladamente usando herramientas de modelación y el lenguaje de UML.

Capítulo 4: Construcción de la solución propuesta.

Se describe cómo será construida la solución que se ha propuesto, se modelan los diagramas de clases necesarios, se especifican los principios para el diseño gráfico.

Capítulo 5: Estudio de Factibilidad.

Contiene el estudio de factibilidad realizado para este sistema, se enumeran los beneficios tangibles e intangibles y se analizan los costos que representa la elaboración de la propuesta de solución.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

En este capítulo se presenta de forma general los principales conceptos asociados al dominio del problema que son necesarios para entender el entorno. Se detalla el objeto de estudio de esta investigación mediante una descripción más amplia. Además se hace el análisis de algunos sistemas automatizados vinculados con el campo de acción.

1.2 Herramientas de búsquedas.

Actualmente nos enfrentamos a un mundo de información distribuida en la red que continúa creciendo y para hacer un mejor uso de esta poderosa fuente de información debemos familiarizarnos con las herramientas de búsqueda. Las herramientas de búsqueda son sistemas que ayudan al usuario a localizar y acceder a la información, en gran medida facilitan la búsqueda en la web. Hoy día los más difundidos son los llamados buscadores automáticos o motores de búsqueda, los metabuscadores y los directorios.

1.2.1 Metabuscadores.

Los metabuscadores son sitios web especializados en consultar varios buscadores al mismo tiempo y presentar la información proporcionada de forma entendible y ordenada. En realidad, no son buscadores, no disponen de bases de datos propias, sino que utilizan los recursos almacenados en las bases de datos de otros índices y buscadores. Lo que hacen, es realizar búsquedas en auténticos buscadores, analizan los resultados de la página, y presentan sus propios resultados. No disponen de estructuras de indexación propias, reenvían las búsquedas a otros sistemas de búsqueda.

Entre las principales ventajas de este sistema de búsqueda se encuentra que combina resultados de múltiples buscadores, elimina los resultados duplicados y confiere mayor peso a aquellos resultados que

aparecen en varios buscadores y como desventaja que es mayor el tiempo de espera, además de que explota los servicios ofrecidos por otros sistemas de búsqueda.

1.2.2 Buscadores automáticos o motores de búsqueda.

Los buscadores automáticos o motores de búsqueda son herramientas de búsqueda que utiliza la World Wide Web para la recuperación de información a partir de palabras claves que identifican el contenido de la pregunta a realizar. Habitualmente localizan las páginas web que mejor se adapten a las palabras introducidas. Algunos ejemplos de este tipo de buscador son: Altavista, HotBot, Lycos, Google, entre otros.

Tres son los elementos que componen un motor de búsqueda:

- Interfaz: Página web a la que accede el usuario y mediante la cual efectúa la búsqueda.
- Robots: Programa que recorre la web de forma automática para la recopilación de información de las páginas encontradas.
- Base de datos: Índice de palabras, frases o datos asociados con la dirección URL de las páginas web.

Los motores de búsqueda realizan su trabajo haciendo referencia al robot y actualizando la base de datos periódicamente. El robot como programa que recorre la web automáticamente en la recopilación de información de las páginas web encontradas procede con la indexación de los datos y guarda en la base de datos los términos indexados, así como la dirección URL de los documentos localizados. La información indexada luego será explorada por medio de un motor de búsqueda con el fin de encontrar términos que coincidan con los de la pregunta realizada y devuelve en el menor tiempo posible una lista con las direcciones de las páginas que más se ajusten a la pregunta y los usuarios podrán acceder a ella a través de una interfaz de consulta.

Una vez que se han analizado los aspectos teóricos y técnicos de los motores de búsqueda es necesario ver cuáles son sus ventajas y desventajas en el proceso de recuperación de información. Dentro de sus ventajas se destaca su utilidad para realizar búsquedas sobre temas o sitios específicos, actualizan sus

bases de datos periódicamente y proveen acceso a gran cantidad de páginas, que aumentan exponencialmente. Pero presenta algunos inconvenientes pues retornan muchos resultados irrelevantes, diferentes motores de búsqueda pueden entregar distintos resultados, además que es costoso mantener actualizada una base de datos.

1.2.3 Buscadores basados en directorios.

Esta es otra variante de buscadores, por ejemplo: un directorio es un sitio donde se almacenan links a otros sitios, estos vínculos están agrupados por categorías y tienen una estructura arbórea, de forma tal que para acceder a un sitio en específico es necesario comenzar a navegar por las distintas categorías en orden descendente hasta llegar a la categoría “hoja” donde se ubican los vínculos a las páginas en concreto en las cuales se encuentra la información deseada.

Es, básicamente, una lista alfabética de encabezamientos de materias que puede ser subdividida más ampliamente. Las búsquedas se hacen descendiendo a través de estos encabezamientos y subencabezamientos hasta llegar a la información deseada.(PULIDO 2000)

Este tipo de buscador presenta la información de la web en forma de una colección de categorías, y las búsquedas se realizan en profundidad, una vez que se selecciona una categoría, se muestran las subcategorías contenidas en ella, además de los sitios que se relacionan directamente con dicha categoría.

Presenta una estructura en forma de árbol, en el que se agrupa en las ramas principales las categorías o temas más generales y amplios, por ejemplo (artes, deportes, computación), estas categorías son capaces de encerrar dentro de ellas a muchas otras. El hecho de que los sitios se incluyan de forma manual implica que los resultados de la búsqueda sean mejores y que el ruido sea casi nulo, esto quiere decir que los resultados se corresponden en un gran % al tema de búsqueda, una desventaja es que el costo de operación es relativamente alto al ser el factor humano fundamental en el funcionamiento del mismo.

Los directorios se centran en la calidad y no en la cantidad. Como principal ventaja se destaca la gran calidad de sus contenidos y desventaja con respecto al crecimiento desmedido que tiene la web y la velocidad de clasificación.

1.3 Recuperación de Información.

La información no es más que un conjunto de datos ordenados, organizados manejados por el usuario o por sistemas, para poder procesar esta es necesario guardarla lógicamente en archivos.

Los sistemas de información se definen como un conjunto de funciones o componentes interrelacionados que forman un todo, es decir, obtienen, procesan, almacenan y distribuyen información (datos manipulados).

La Recuperación de Información es una actividad compleja que realizan tanto los sistemas informáticos como los seres humanos. Se desarrolla mediante un conjunto de tareas las cuales permiten localizar y acceder a los recursos de información pertenecientes a la resolución del problema planteado. Cuando a un usuario le surge la necesidad de obtener nueva información sobre un asunto o materia de su interés, está manifestando una carencia, una situación irregular planteada bajo la necesidad de resolver una duda, o de documentar una afirmación o estudio, estas son expresiones clásicas de los procesos de recuperación de información.

De ahí se define la Recuperación de Información (RI) como la representación, almacenamiento, organización y el acceso a la información deseada. El campo de RI envuelve un conjunto bastante grande de conceptos, estructuras y métodos. Conceptualmente la Recuperación de Información es una operación que analiza la necesidad de información del usuario y selecciona los archivos más significativos que den solución a esta petición, es decir, localiza los documentos que muestren mayor asociación a la pregunta formulada.

La recuperación de información engloba las acciones encaminadas a identificar, seleccionar y acceder a los recursos de información útiles al usuario. Con el desarrollo de los sistemas digitales de procesamiento de datos y de tratamiento de información, las técnicas de recuperación de información han ido desarrollando un conjunto de teorías y aplicaciones prácticas que subyacen en la actualidad a

cualquier búsqueda y recuperación de información en Internet. Un sistema de recuperación de información proporciona acceso rápido automático a grandes colecciones de objetos que contienen información de cualquier tipo.

El proceso de recuperación de información está compuesto por una interfaz de usuario y por operaciones de consultas. Mediante la interfaz de usuario se gestiona toda la interacción con el usuario así como las entradas de las consultas y la visualización de los resultados y a través de las operaciones de consulta se transforma esta para mejorar la búsqueda, actividad en la cual entran a jugar su papel los lenguajes de programación donde se transforma la consulta que se realiza en lenguaje natural para que el sistema entienda lo que se quiere buscar.

1.4 Aplicación.

Para mejor entendimiento de lo que se abordará más adelante es necesario interiorizar qué es una aplicación.

Aplicación es:

Un programa informático que lleva a cabo una función con el objeto de ayudar a un usuario a realizar una determinada actividad. (*Glosario de Definiciones*)

Luego una aplicación se define como un programa informático o software almacenado y ejecutado en una computadora capaz de realizar las funciones para las cuales fue creado con el fin de satisfacer una necesidad del usuario.

1.5 Base de Datos.

Una base de datos es el lugar donde se guardan los datos en reposo y al cual acceden las diferentes aplicaciones (sistemas o programas) de una organización. (ROZIC 2004)

Conjunto organizado e integrado de datos almacenados en computadora, con el fin de facilitar su uso para aplicaciones con múltiples finalidades. (AGRICULTURA 1997)

Conjunto de datos accesible mediante medios informáticos y organizados de tal modo que la adición, consulta y actualización (el llamado mantenimiento) resulten ser operaciones rápidas y sencillas. (IGLESIAS 1999)

Entonces una base de datos se define como un conjunto de datos organizados en forma tal que su contenido sea fácilmente accesible, administrado y actualizado. La filosofía de las bases de datos es la de almacenar grandes cantidades de datos de una manera no redundante y que permita las posibles consultas de acuerdo a los derechos de acceso.

Las bases de datos se encuentran estructuradas en forma de tablas, en las que cada fila, o registro, se corresponde con el concepto de ficha, en el que están contenidos todos los campos (es decir, las columnas) que describen las características de los objetos a los que hace referencia la base de datos.

Al diseñar una base de datos óptima existen varias consideraciones a tener en cuenta:

- La velocidad de acceso.
- El tamaño de la información.
- El tipo de la información.
- Facilidad de acceso a la información.
- Facilidad para extraer la información requerida.
- El comportamiento del manejador de bases de datos con cada tipo de información.

El diseño de una base de datos es un proceso complejo, se descompone en diseño conceptual, diseño lógico y diseño físico.

El diseño conceptual parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla. Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño

conceptual es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información. (ANDRÉS 2001)

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. Un esquema lógico es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD. Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.). El diseño lógico depende del tipo de SGBD que se vaya a utilizar, no depende del producto concreto.(ANDRÉS 2001)

El diseño físico parte del esquema lógico y da como resultado un esquema físico. Un esquema físico es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.(ANDRÉS 2001)

A continuación se exponen algunas de las principales ventajas que proporciona el uso de las bases de datos en sentido general:

- Normaliza los datos de la base de datos.
- Evita la redundancia e inconsistencia de datos.
- Garantiza la integridad de los datos.
- Garantiza la seguridad de los datos.
- Comparte los datos.
- Facilita la modificación los datos.

1.5.1 Tipos de base de datos.

Las bases de datos por su función se pueden clasificar de acuerdo a su administración de datos. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Las bases de datos jerárquicas son los que almacenan su información en una estructura jerárquica. También existen las bases de datos de red que es un modelo ligeramente distinto del jerárquico, en estas se permite que un mismo nodo tenga varios padres. Las bases de datos relacionales son los modelos que más se usan para modelar problemas reales y administrar datos dinámicamente, con la ventaja de que el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red).

En cambio, las bases de datos orientadas a objetos es un modelo bastante reciente que trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Incorpora todos los conceptos importantes del paradigma de objetos: Encapsulación, Herencia y Polimorfismo. Otro de los modelos que se utilizan para modelar bases de datos es el modelo de gestión de bases de datos distribuida, el cual se basa en que la información está almacenada en varias computadoras conectadas en red.

Una Base de Datos Distribuida es, una base de datos construida sobre una red computacional y no por el contrario en una máquina aislada. Surgen debido a la existencia física de organismos descentralizados. La información que constituye la base de datos esta almacenada en diferentes sitios en la red, y las aplicaciones que se ejecutan accesan datos en distintos sitios.

La idea de distribución de los datos trajo consigo la superación de una serie de dificultades tecnológicas entre las que se pueden mencionar:

- Asegurar que el acceso entre diferentes sitios o nodos y el procesamiento de datos se realice de manera eficiente, presumiblemente óptima.
- Transformar datos e integrar diferentes tipos de procesamiento entre nodos de un ambiente distribuido.
- Distribuir datos en los nodos del ambiente distribuido de una manera óptima.
- Soportar la recuperación de errores de diferentes módulos del sistema de manera segura y eficiente.

- Soportar el diseño de sistemas de información distribuidos. Por ejemplo, ayudar a decidir donde localizar algún dato particular o donde es mejor ejecutar un programa de aplicación.
- Considerar la competencia que habrá por el uso de los recursos entre nodos diferentes.

La principal desventaja de estos sistemas se refiere al control y manejo de los datos. Dado que éstos residen en muchos nodos diferentes y se pueden consultar por nodos diversos de la red, así como la probabilidad de violaciones de seguridad es creciente si no se toman las precauciones debidas. La otra desventaja consiste en la habilidad para asegurar la integridad de la información en presencia de fallas no predecibles tanto de componentes de hardware como de software es compleja. La integridad se refiere a la consistencia, validez y exactitud de la información. También que los datos pueden estar replicados, el control de concurrencia y los mecanismos de recuperación son mucho más complejos que en un sistema centralizado.

Otra desventaja importante es el coste de desarrollo del software. La complejidad añadida que es necesaria para mantener la coordinación entre nodos hace que el desarrollo de software sea más costoso. Además tiene mayor probabilidad de errores y la sobrecarga de procesamiento es mayor. El intercambio de mensajes y ejecución de algoritmos para el mantenimiento de la coordinación entre nodos supone una sobrecarga que no se da en los sistemas centralizados.

1.5.2 Normalización de bases de datos.

Uno de los pasos más importantes en el diseño de una base de datos consiste en asegurarse de que los datos se distribuyan correctamente entre las tablas. Si las estructuras de datos son correctas, el resto de la aplicación (las consultas, los formularios, los informes, el código, etc.) se verá simplificada en gran medida. El nombre formal que recibe el diseño de una tabla es el de normalización de bases de datos.

El proceso de normalización es un estándar que consiste, básicamente, en un proceso de conversión de las relaciones entre las entidades, evitando:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

Técnicas de almacenamiento de la información

Las técnicas empleadas para almacenar bases de datos son sumamente importantes para la velocidad de acceso y recuperación de datos. Las técnicas dependen del tipo de almacenamiento, el uso que se le da o se le dará a la base de datos, la estructura de la misma, el SGBD empleado, etc.

Esta dependencia no significa necesariamente que haya que cambiar la estructura de la base de datos si se cambian las técnicas empleadas. Las técnicas de almacenamiento son independientes de la base de datos, pero, de todas maneras, las mejores técnicas muchas veces pueden determinarse viendo la estructura de la base de datos, entre otras características.

Los encargados de elegir estas técnicas son los diseñadores y administradores de bases de datos, y dependen también de las capacidades del SGBD. En general, el SGBD ofrece diferentes opciones y técnicas para organizar los datos.

La idea es que los encargados de la base de datos encuentren las técnicas idóneas, o sea, aquellas que permitan la mayor velocidad posible de acceso a los datos. Una mala decisión en esta área puede resultar en una menor velocidad de acceso a la base de datos, o en un uso excesivo del espacio de almacenamiento, o incluso, puede aumentar la velocidad de consulta de una base de datos, pero disminuir la velocidad de actualización de la misma.

Las bases de datos se almacenan en ficheros o archivos. Existen diferentes formas de organizaciones primarias de archivos que determinan la forma en que los registros de un archivo se colocan físicamente en el disco y, por lo tanto, cómo se accede a éstos.

Las distintas formas de organizaciones primarias de archivos son:

- Archivos de montículos (o no ordenados): esta técnica coloca los registros en el disco sin un orden específico, añadiendo nuevos registros al final del archivo.
- Archivos ordenados (o secuenciales): mantiene el orden de los registros con respecto a algún valor de algún campo (clave de ordenación).
- Archivos de direccionamiento calculado: utilizan una función de direccionamiento calculado aplicada a un campo específico para determinar la colocación de los registros en disco.
- Árboles B: se vale de la estructura de árbol para las colocaciones de registros.

Existe una segunda forma de acceder a los datos llamada organización secundaria o estructura de acceso auxiliar. Estas permiten que los accesos a los registros de un archivo basado en campos alternativos, sean más eficientes que los que han sido utilizados para la organización primaria de archivos.

Los avances en la explotación y manejo de la información (acceso, consulta, recuperación, análisis) son inherentes a los sistemas de información actuales y futuros. Hoy día, acceder a grandes volúmenes de información es una realidad; mañana, los sistemas de información permitirán a grandes comunidades de usuarios acceder de manera transparente a recursos (datos, documentos, servicios) pre-existentes, autónomos, distribuidos y heterogéneos.

Los métodos de extracción o recuperación se enfocan hacia la especificación de algoritmos de búsqueda precisos y eficientes que apliquen varios modelos matemáticos o aspectos semánticos al proceso de búsqueda y de recuperación de información.

1.5.3 Transacciones, concurrencia y recuperación.

Transacciones

Una transacción es una unidad de la ejecución de un programa que accede y posiblemente actualiza varios elementos de datos. Se delimita dependiendo del lenguaje por las sentencias inicio transacción y fin transacción y se compone de todas las instrucciones que se encuentran entre estos dos delimitadores. (NEIRA 2002)

Implementar transacciones puede ser el modo más significativo para aumentar la velocidad de acceso a datos en la aplicación de base de datos. Esta técnica es ideal para acelerar las consultas, insertar, actualizar, anexar o eliminar datos de una base de datos. Es responsabilidad del programador definir adecuadamente las diferentes transacciones, de tal manera que cada una preserve la consistencia de la base de datos.

Para asegurar la consistencia de los datos se necesita que el sistema de base de datos tengan las propiedades llamadas ACID: (Atomicidad, Consistencia, Aislamiento, Durabilidad).

- Atomicidad: asegura que o bien todos los efectos de la transacción se reflejan en la base de datos, o bien ninguno de ellos; un fallo no puede dejar a la base de datos en un estado en el cual una transacción se haya ejecutado parcialmente.
- Consistencia: asegura que si la base de datos es consistente inicialmente, la ejecución de la transacción deja la base de datos en un estado consistente.
- Aislamiento: asegura que en la ejecución concurrente de transacciones, estas están aisladas unas de otras, de tal manera que cada una tiene la impresión de que ninguna otra transacción se ejecuta concurrentemente con ella.
- Durabilidad: asegura que una vez que la transacción se ha comprometido, las actualizaciones hechas por la transacción no se pierden, incluso si hay un fallo en el sistema.

La operación **COMMIT** indica el término exitoso de una transacción: le dice al administrador de transacciones que se ha finalizado con éxito, que la base de datos está o debería estar de nuevo en un estado consistente y que se puede comprometer o hacer permanentes todas las modificaciones efectuadas.

La operación **ROLLBACK**, en cambio, nos indica el término no exitoso de una transacción: le dice al administrador de transacciones que algo salió mal, que la base de datos podría estar en un estado no consistente y que todas las modificaciones realizadas hasta el momento deben retroceder o anularse.

Asegurar las propiedades de atomicidad y durabilidad es responsabilidad del propio sistema de base de datos, específicamente del componente de gestión de transacciones. En ausencia de fallos, toda transacción completada con éxito y atómica se archiva fácilmente.

El sistema de base de datos debe realizar la recuperación de fallos, es decir, detectar los fallos del sistema y restaurar la base de datos al estado que existía antes de que ocurriera el fallo.

Finalmente cuando varias transacciones actualizan la base de datos concurrentemente, la consistencia de los datos puede no ser preservada, incluso aunque cada transacción individualmente sea correcta. Es responsabilidad del gestor de control de concurrencia controlar la interacción entre las transacciones concurrentes para asegurar la consistencia de la base de datos. (PROAÑO 2006)

En esencia, lo que se persigue con el procesamiento de transacciones es, por una parte, obtener una transparencia adecuada de las acciones concurrentes a una base de datos y por otra, manejar adecuadamente las fallas que se puedan presentar en una base de datos. (Referencia comparativa)

Concurrencia

El control de concurrencia en las bases de datos permite que la información se maneje en forma eficiente, permite además la ejecución de transacciones en paralelo, accediendo a información compartida y, por lo tanto, interfiriendo potencialmente unas con otras. (PROAÑO 2006)

Existen varios esquemas de control de concurrencia que aseguran la secuencialidad, todos estos esquemas o bien retrasan una operación o bien abortan la transacción que ha realizado la operación. Los más conocidos son los protocolos de bloqueo, el esquema de ordenación por marcas temporales, las técnicas de validación, el esquema de granularidad múltiple y los esquemas multiversión.

Un protocolo de bloqueo es un conjunto de reglas que indican el momento en que una transacción puede bloquear o desbloquear un objeto de la base de datos. El protocolo de bloqueo de dos fases permite que una transacción bloquee un objeto sólo después de que haya desbloqueado otro objeto distinto, este método asegura la secuencialidad. (NEIRA 2002)

El esquema de ordenación por marcas temporales asegura la secuencialidad seleccionando previamente un orden en todo par de transacciones. Existen dos formas de implementar este esquema, uno es por medio de valores timestamp (dependientes del reloj del sistema) y por medio de un contador lógico que se incrementa cada vez que asigna una nueva marca temporal. Este protocolo asegura la secuencialidad en cuanto a conflictos y la ausencia de interbloqueos, si una de las transacciones viola la norma la transacción se retrasa y se le asigna una nueva marca temporal. Por ejemplo, una operación leer se puede retrasar porque todavía no se ha escrito el valor apropiado o incluso rechazar si ha sobrescrito el valor que supuestamente se iba a leer.

Un esquema de validación es un método de control de concurrencia adecuado para transacciones de sólo lectura y en las cuales la tasa de conflictos es baja. Se basa en dividir una transacción en tres etapas (lectura, validación y escritura) y trabajar con el esquema de marcas temporales sobre la etapa de validación. De esta manera se quita una sobrecarga en la etapa de lectura, en la cual no se necesita un esquema de control de concurrencia dado que toda lectura lleva a la base de datos al mismo estado de consistencia.

Una manera distinta manejar la concurrencia es por medio de la granularidad, este concepto permite agrupar varios elementos de datos y definirlos como un todo para efectos de sincronización. Se define como una jerarquía de distintos niveles, donde el nivel superior puede representar toda la base de datos, se esquematiza como estructura de árbol donde los nodos hijos de un nodo interno representan las

dependencias de datos asociadas. Se utilizan los tipos de bloqueos Compartidos y Exclusivos más un nuevo tipo de bloqueo llamado el bloqueo intencional, el cual indica que existen nodos descendientes que tienen bloqueos compartidos o exclusivos.

El esquema multiversión se basa en la creación de nuevas versiones de los elementos de datos cada vez que una transacción vaya a escribir dicho elemento. Cuando se va a realizar una escritura el sistema elige una de las versiones para que se lea. El esquema de control de versiones garantiza que la versión que se va a leer se elige de forma que asegure la secuencialidad por medio de marcas temporales. En este esquema una operación de lectura tiene éxito siempre, sin embargo, una operación de escritura puede provocar el retroceso de una transacción.

Recuperación

El objetivo de la recuperación es proteger la base de datos contra fallas lógicas o físicas (fallos de disco, cortes de energía o fallos en el software) que destruyan los datos en forma total o parcial. Estas fallas pueden afectar al correcto almacenamiento de los datos.

Para asegurar que la base de datos siempre esté en un estado consistente, cada base de datos tiene un proceso para obtener copias de seguridad, esto ayuda a mantener un registro confiable de los datos ante desastres o posibles fallas del sistema.

Existen varios tipos de fallas, a considerar:

- Fallo en la transacción, que a su vez se dividen en errores lógicos y errores del sistema. Un error lógico ocurre cuando una transacción no puede continuar con su ejecución normal a causa de una condición interna como lo es un desbordamiento o un exceso de recursos. Un error del sistema ocurre cuando el sistema se encuentra en un estado no deseado.
- Caída del sistema, provocado ya sea por el hardware, el sistema operativo o por el software de base de datos. Comúnmente causa la pérdida del contenido de la memoria primaria y aborta el procesamiento de una transacción.
- Fallo de disco, para el cual sólo sirve la recuperación por medio de copias existentes en medios de almacenamiento secundarios.

1.6 Objeto de Estudio.

Como objeto de estudio de esta investigación se había definido la gestión de información de las personas en la Universidad de las Ciencias Informáticas. Pero ¿qué es la gestión de información?

Gestión de información: Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada. (QUIROGA 2002)

Luego, partiendo del concepto mencionado anteriormente la gestión de información no es más que el proceso mediante el cual podemos recopilar, analizar y almacenar determinada cantidad de información de un tema en específico.

1.6.1 Descripción General.

La información es un agente importante en la modificación de las conductas existentes en la organización, su correcta gestión es una herramienta fundamental para la toma de decisiones, la formación del personal, la evaluación de los productos, la determinación de los errores y el control de los procesos. (QUIROGA 2002)

La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas. (QUIROGA 2002)

Sistema de gestión de información: Es el conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información.

Las estrategias actuales para la gestión de la información y el conocimiento deben responder los nuevos tipos de demandas, resultantes de la aparición de tendencias gerenciales más modernas en las organizaciones. En la creación de los nuevos sistemas de gestión de la información es imprescindible considerar las fuentes factográficas (datos), documentales y no documentales, los sistemas informáticos, la cultura de información, los modelos de comunicación, entre otros elementos.

En la Universidad de las Ciencias Informáticas la mayoría de la información se encuentra distribuida en toda la red. Esto nos da la medida del gran valor que posee la información como recurso imprescindible para esta institución, ya que el buen funcionamiento de la misma gira en torno a ella, además del personal que labora en este centro.

La necesidad de información en toda la comunidad universitaria es cada vez mayor, existe mucha información a la que acceder pero hay que saber cómo manejarla y administrarla, por el significado que poseen los datos que aquí se manipulan. En la actualidad la gestión de información es uno de los problemas que presenta la universidad.

Debido a que la mayor parte de la información se encuentra compartida en la red, como se mencionaba anteriormente, en la universidad se prestan varios servicios de búsqueda que permiten el desarrollo de esta institución como ciudad digital. Estos sistemas dedicados a la gestión y almacenamiento de la información en la UCI deben estar disponibles a cualquier hora del día, lo que hace que se requiera un excelente funcionamiento de los mismos.

1.6.2 Descripción actual del dominio del problema.

El acceso a la información de las personas en la Universidad de las Ciencias Informáticas constituye uno de los servicios de primer orden. Para el almacenamiento de dicha información cuenta con varias Bases de Datos en las que se encuentra recopilada toda la información con datos propios de las personas que pertenecen a este centro estudiantil, ya sea estudiantes, trabajadores u otras personas vinculadas al mismo.

La gestión de la información académica se realiza a través de un sistema implementado en la propia universidad, Akademos. La información que compete a los trabajadores y a la residencia es controlada por medio de otros sistemas, los cuales laboran independientemente y gestionan los procesos por los que debe pasar la misma, mas adelante hacemos referencia a estos sistemas.

Para acceder a los datos de la persona existe implementado un sistema de búsqueda de información de personas de la UCI, el cual se encuentra en actual función como Directorio de Personas; de este sistema puede hacer uso cualquier tipo de persona de la universidad. La aplicación consta de dos interfaces para efectuar la búsqueda en la que el usuario es quién decide cuál es la interfaz que va a utilizar para realizar dicha búsqueda. La primera interfaz presenta en su conjunto un formulario con una región de texto y un botón, en ella se introduce una palabra o un patrón para la búsqueda; la segunda muestra también un formulario pero con varias regiones de inserción o selección de texto a partir de las que el sistema construye automáticamente el patrón para la búsqueda.

Los resultados generados de la búsqueda se presentan al usuario divididos en dos bloques, un primer bloque para estudiantes y otro para trabajadores; de un elemento de cualquiera de los bloques se puede ver además de la información presentada inicialmente, una ficha con datos más específicos de la persona.

Debido a que la información de las personas se encuentra almacenada en bases de datos distribuidas el sistema gestiona los datos mediante la base de datos Ciudadano. Esta recupera la información que desea mostrar al usuario extrayendo la misma de la base de datos Akademos y Trabajadores de acuerdo a los criterios de búsqueda establecidos.

1.6.3 Situación problemática.

En la universidad existe un Directorio de Personas asociado con las distintas Bases de Datos existentes en el centro, por ejemplo Akademos, Ciudadano, Trabajadores, en las que se encuentran registrados todos los estudiantes, trabajadores y profesores acreditados en el centro. Este sistema de búsqueda presenta algunas deficiencias al no satisfacer todas las necesidades del usuario y carecer de ciertas funcionalidades importantes, además de no contar con una base de datos propia y un sistema para la actualización de los datos de las personas. Algunas de estas insatisfacciones se ven reflejadas por las siguientes cuestiones:

1. Respuesta al usuario demasiado lenta.
2. Dificultad para realizar consultas múltiples o complejas.
3. Distribución de la información de las personas en distintas bases de datos.
4. Dificultad del sistema para la sincronización de datos.
5. Dependencia de la base de datos Ciudadano para cualquier operación.
6. Poca independencia entre los sistemas.

1.7 Sistemas automatizados vinculados con el campo de acción.

En la universidad existen sistemas vinculados a la gestión de la información de las personas. Algunos de los que se encuentran relacionados con nuestro campo de acción son los que a continuación describimos.

➤ **Sistema de gestión académica: Akademos**

Akademos es el sistema que se encarga de toda la gestión de la información académica en la Universidad de las Ciencias Informáticas. Este sistema permite llevar un control de la asistencia a clases, las evaluaciones, la relación de notas de los estudiantes y otras actividades y en su diseño presenta diferentes secciones; secciones accesibles para profesores y secciones accesibles para estudiantes. A través de la sección para profesores los mismos pueden llevar un control académico de los grupos a los cuales imparten clases así como los estudiantes en su sección pueden observar la relación de notas obtenidas en cursos anteriores y las evaluaciones alcanzadas hasta el momento.

➤ **Sistema de gestión de alojamiento**

El sistema de gestión de alojamiento controla todas las actividades relacionadas con la ubicación de las personas en la Universidad de las Ciencias Informáticas. La zona de residencia, constituida por un gran número de edificios está distribuida por áreas en dependencia de las diez facultades que tiene la universidad; en la residencia se encuentran alojados los estudiantes y trabajadores del centro. Este sistema además de llevar un control de las actividades de alojamiento en la universidad permite al personal que labora en la residencia realizar un trabajo más eficiente.

➤ **Sistema de gestión de información de trabajadores.**

El sistema de gestión de información de trabajadores controla todos los procesos relacionados con los recursos humanos. Los trabajadores que son los que contribuyen al correcto funcionamiento de la universidad se encuentran distribuidos en diferentes áreas en relación con su rol como trabajadores de este centro, así como en los comedores, facultades, entre otros. Este sistema para la gestión de información de trabajadores y en general de los recursos humanos permite además del control la actualización inmediata de la misma.

1.8 Fundamentación del objetivo general.

Partiendo de la situación actual de la información de la persona en la Universidad de las Ciencias Informáticas y con la finalidad de dar solución al problema que existe para la gestión de la información el objetivo general propuesto es rediseñar el Directorio de Personas, diseñar una base de datos y una aplicación para la actualización de los datos de las mismas en la Universidad de las Ciencias Informáticas mediante servicios web, manifestando así el propósito de este trabajo de investigación.

1.9 Propuesta de solución.

Luego de analizar la manera en que se encuentra distribuida la información de las personas en la universidad y haber estudiado detalladamente los rasgos del objeto de estudio, es necesario rediseñar el Directorio de Personas, diseñar una base de datos y una aplicación que se encargue de la actualización de la misma en la Universidad de las Ciencias Informáticas que sea capaz de resolver los problemas existente.

La base de datos propuesta será a la que accederá el Directorio de Personas para brindar sus servicios. El diseño de la misma estará orientado a garantizar una mayor velocidad de acceso, de esta manera la búsqueda será más rápida y eficiente.

La aplicación de actualización será capaz de extraer información de las bases de datos existentes en la universidad mediante servicios web para así actualizar la base de datos del directorio. Durante el proceso

de actualización de la base de datos el servicio de búsqueda de personas nunca será detenido. Esto debe ser garantizado por el control de concurrencia.

Contando con una base de datos propia y una aplicación para la actualización de los datos, el directorio propuesto ofrecerá un mejor servicio de búsqueda, con nuevas funcionalidades que permitirán realizar consultas múltiples y complejas.

1.10 Conclusiones.

En este capítulo se detallaron los aspectos más significativos que rodean el objeto de estudio, se determinaron las condiciones esenciales alrededor al problema a través de los conceptos y las definiciones planteadas, en función a esto se obtuvo el objetivo general para este trabajo. Además se analizaron algunos sistemas vinculados con el campo de acción y finalmente se elaboró la propuesta de solución al problema en cuestión.

CAPITULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR

2.1 Introducción.

En el presente capítulo se analizan las características de las principales tecnologías a considerar para el desarrollo del sistema que se pretende llevar a cabo. En algunos casos se establecen comparaciones que nos ayudan a tomar una decisión final sobre qué tecnología es más conveniente emplear en la confección de esta aplicación. Se explica el porqué de la metodología utilizada y el lenguaje a utilizar en la elaboración del mismo y se detallan algunas características que fundamentan tal elección.

2.2 Las Tecnologías de la Información y las Comunicaciones (TIC).

Las TIC son las tecnologías que se necesitan para la gestión y transformación de la información con el apoyo de programas que permitan crear, modificar, almacenar, proteger y recuperar información. La introducción de estas tecnologías implica un cambio de la sociedad, representan un elemento fundamental en la Sociedad de la información o Sociedad del Conocimiento pues habilitan la capacidad universal de acceder y contribuir a la información, las ideas y el conocimiento, además de promover el intercambio y el fortalecimiento de los conocimientos mundiales en favor del desarrollo.

Permiten un acceso equitativo a la información para actividades económicas, sociales, políticas, sanitarias, culturales, educativas y científicas dando acceso a la información que está en el dominio público. El uso de las mismas genera múltiples ventajas en todas las esferas de la sociedad, pero esto dependerá, en gran medida, de cómo se use en una determinada comunidad y cuánta importancia le otorgue en su desarrollo.

Desde el punto de vista educacional elevan la calidad del proceso educativo derribando las barreras del espacio y del tiempo, permitiendo la interacción y colaboración entre las personas para la construcción colectiva del conocimiento y de fuentes de información de calidad. En las empresas el empleo de las TIC

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

atribuye mejoras en los procesos productivos internos al proporcionar herramientas que facilitan la clasificación, organización, manejo y filtro de la información, y de cara al exterior mejoran los procesos de interacción con los agentes externos: clientes, proveedores y socios, permitiendo el acceso desde cualquier lugar a cualquier hora, abriendo así la posibilidad de nuevos negocios en la red.

2.3 Aplicaciones Web vs. Sitios Web.

Un sitio Web es una página o conjunto de páginas accesibles a través de Internet o de una intranet y convenientemente enlazadas que se identifica con una dirección (URL) que es la de la página principal o portada, puede contener información sobre un tema en específico o muchos en general. La propiedad y administración de cada sitio web corresponde a un individuo, empresa u organización.

Por lo general, aunque no necesariamente, las páginas de un determinado sitio web suelen almacenarse en un solo servidor. (GLOSARIO DEL NAVEGANTE 2006)

De ahí que la arquitectura sea sencilla, sus principales componentes un servidor Web, una conexión de red, y uno o más clientes (browsers). El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP.

Una aplicación web es aquella que es accedida por los usuarios desde un servidor web de Internet o de una intranet. La habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

Actualmente, lo más importante en el desarrollo de aplicaciones Web han sido las herramientas. Poco se ha dicho sobre el proceso de desarrollo. La fácil creación de hojas HTML y en general de sitios Web, usando herramientas simples, ha hecho que el desarrollo de este tipo de aplicaciones se haga sin un trabajo serio de análisis y diseño. Las aplicaciones Web, al igual que otras aplicaciones, necesitan métodos formales de análisis y diseño.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

Las Aplicaciones Web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Una aplicación Web es un sitio Web donde la navegación a través del sitio, y la entrada de datos por parte de un usuario, afectan o modifica el estado de la lógica del negocio. En esencia, una aplicación Web usa un sitio Web como entrada a una aplicación típica. Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web sino Sitio Web (GUERRERO 2005)

La información mostrada en las páginas está típicamente almacenada en archivos. Sin embargo, muchas veces esta información está contenida en una base de datos, y las páginas son creadas dinámicamente. Los sitios Web que usan este esquema, son llamados sitios dinámicos.

Las aplicaciones web presentan una serie de ventajas, a continuación se mencionan algunas de ellas:

- Necesita solo una computadora con navegador Web.
- Son fáciles de usar (no requieren conocimientos avanzados de computación).
- Alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde tenga acceso a Internet y a cualquier hora.
- No requieren instalación, pues usan tecnología Web, lo cual permite el aprovechamiento de todas las características del Internet.
- Compatibilidad multiplataforma. Se puede cambiar de sistema operativo o cambiar el hardware sin afectar el funcionamiento de las aplicaciones del servidor.
- Inmediatez de acceso. Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas.

Una vez analizado las características y facilidades que brinda la web se propone para el desarrollo del Directorio de Personas de la universidad una aplicación web, de esta manera podrá ser uso de dicho directorio todos los usuarios del centro desde cualquier terminal.

2.4 Aplicación de escritorio.

Windows es un entorno basado en mensajes. Una aplicación Windows no es más que un intérprete de mensajes. Los mensajes, enviados por el sistema o por otras ventanas, se acumulan en la cola de mensajes de la aplicación y se van tratando secuencialmente.

La clave de las aplicaciones de escritorio está en la sencillez y el mecanismo de funcionamiento se adapta al paradigma de programación orientada a objetos. Este tipo de aplicación almacena tradicionalmente sus datos en un archivo, aunque algunos tienen la capacidad de colaborar con otros usuarios, o almacenar sus datos en una base de datos central. Estas aplicaciones se basan en la eficacia del equipo de escritorio al procesar y presentar el contenido de alto rendimiento. En este caso hay que tener la aplicación instalada en cada uno de los clientes.

Para el desarrollo de la aplicación de actualización de los datos de las personas en la Universidad de las Ciencias Informáticas se decidió recurrir a este tipo de aplicación por las características que debe poseer dicha aplicación.

La aplicación de escritorio brinda una serie de bondades que para la aplicación que se propone, en la web se encuentra un poco limitado. A través de la aplicación de escritorio la respuesta es más rápida y para este sistema de actualización se requiere rapidez en el acceso a los datos. Además posibilita el uso de interfaces enriquecidas, con controles inteligentes que dan mayor fluidez al trabajo del usuario y son aplicaciones en tiempo real, mientras que en la web la interactividad no se produce de esta forma, cada acción del usuario conlleva un tiempo de espera excesivo hasta que se obtiene la reacción del sistema.

En la web los elementos de interacción son muy limitados. En comparación con las aplicaciones de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones web (mediante formularios principalmente) son muy escasas.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

La aplicación que se propone no necesita ser web ya que no será accedida por un grupo de personas, será accedido únicamente por el administrador del sistema, y residirá en uno de los servidores de la universidad.

Las aplicaciones de escritorio tienen un campo enorme (sistemas CAD, CAM, suite de oficina, aplicaciones gráficas, juegos, utilidades o el mismo sistema operativo), no todo está en web, hay cosas que se necesitan que se ejecuten estrictamente en su máquina para aprovechar el poder de alcance que tiene.

2.5 Servicios Web.

Un Servicio Web puede definirse como una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. También como una unidad de aplicación capaz de ofrecer datos o servicios de procesamiento a otras aplicaciones informáticas. Así, una aplicación podría ofrecer distintos servicios a otras aplicaciones. Estas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutadas sobre cualquier plataforma pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Estos pueden ofrecer varios servicios tales como:

- Realizar un simple cálculo con unos datos que se le envían como parámetro.
- Acceder a una base de datos para recuperar un conjunto de registros.
- Validar la corrección de una información o contrastarla frente a otros datos, etc.

El servicio web podrá ser solicitado desde otro programa informático que se ejecute en un ordenador conectado al web. Junto a la solicitud de la ejecución, se pueden enviar al ordenador que ofrece el servicio unos parámetros que el servicio web remoto tomará como base para el cálculo o la función.

Las características de estos servicios web es que se solicitan y los resultados de su ejecución son devueltos a través de la web y se transmiten según un modelo estandarizado. Estos estándares señalan

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

cómo se deben hacer las peticiones de servicio a servidores remotos, la forma en la cual éstos deben enviar los resultados, y cómo se deben publicar o dar a conocer los servicios que están accesibles a través de un servidor web. Ninguno de estos estándares trata la forma en la que debe implementarse o programarse el servicio en sí mismo. En este punto, se deja libertad absoluta a los fabricantes y proveedores para elegir el lenguaje de programación que deseen utilizar.

Para el intercambio de mensajes en los Servicios Web se utilizan varios estándares; cada uno con sus responsabilidades bien definidas. Estos son el XML, SOAP, WSDL, y UDDI.

XML (Extensible Markup Language) es un estándar para definir los nombres y propiedades de los items de datos que se intercambian entre el servicio solicitante y el servicio proveedor, mientras que el **SOAP** (Simple Object Access Protocol) representa el conjunto de normas que define como deben ser las solicitudes y respuestas entre un usuario Web y un Servicio Web y constituye la base principal de estos servicios. **WSDL** (Web Services Description Language) describe pormenorizadamente la funcionalidad de un Servicio Web y los protocolos necesarios, tales como **SOA**, para interactuar con el mismo. **UDDI** (Universal Description Discovery and Integration) no es mas que un registro que permite al proveedor de un Servicio Web, anunciarlo y describirlo para que otros usuarios lo puedan encontrar y hacer uso de él.

La interoperabilidad es una de sus ventajas y se logra mediante la adopción de estándares abiertos, esta interoperabilidad es entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen. Los servicios Web con protocolos basados en texto, hacen más fácil acceder a su contenido y entender su funcionamiento. Pueden beneficiarse de los sistemas de seguridad firewall al apoyarse en Http sin necesidad de cambiar las reglas de filtrado. Estos también tienen algunas desventajas o inconvenientes ya que al compararse con otros modelos de computación distribuida su rendimiento es bajo.

2.6 Modelo Cliente/Servidor.

La arquitectura Cliente-Servidor, es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema, en una arquitectura *monolítica* no hay distribución; los

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

tres niveles tienen lugar en el mismo equipo, en el modelo cliente-servidor, en cambio, el trabajo se reparte entre dos ordenadores. (ROMERO and CORALES 2005)

Se puede decir que todas las aplicaciones tienen la misma arquitectura básica y se pueden subdividir en tres partes:

- Interfaz de Usuario: La presentación del usuario, con las entradas de dato y las pantallas de consultas.
- Reglas de Negocio: Sería el procesamiento de la información.
- Accesos a Dato: El control del almacén de datos.

Ventajas del modelo Cliente/Servidor:

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. El cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.
- El sistema es fácil de escalar.

Modelo Cliente/Servidor de tres capas

Cada uno de los componentes de la aplicación en una arquitectura de tres capas se separa en una sola entidad. Esto te permite implementar componentes de una manera más flexible, es decir, la aplicación tiene que estar preparada para los posibles cambios que el cliente pueda pedir sin tener que reescribir totalmente la aplicación. Este tipo de arquitectura es la más compleja.

En esta Arquitectura todas las peticiones de los clientes se controlan en la capa correspondiente a la lógica del negocio. Cuando el cliente necesita hacer una petición se la hace a la capa en la que se encuentra la lógica del negocio. Esto es bastante importante pues eso quiere decir que:

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

- El cliente no tiene que tener drivers ODBC ni la problemática consiguiente de instalación de los drivers por tanto se reduce el costo de mantener las aplicaciones cliente.
- El cliente y el Gestor de Reglas de negocio tienen que hablar el mismo lenguaje (COM, CORBA, SOAP).
- El Gestor de Reglas de Negocio y el Servidor de Datos tienen que hablar el mismo lenguaje (ODBC).

2.7 Lenguajes de programación para la web.

Los lenguajes de programación para la web se encuentran divididos en dos grupos fundamentalmente, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente/Servidor: los lenguajes de programación del lado del Servidor y los lenguajes de programación del lado del Cliente.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, precisamente antes de que la página sea enviada al cliente a través de Internet. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

Dentro del grupo de los lenguajes de programación del lado del Servidor se destacan los llamados Common Gateway Interface (CGI) habitualmente escritos en Perl, Practical Extracting and Reporting Language (Perl), Active Server Pages (ASP), Java Server Pages (JSP) y Personal Home Page (PHP).

Del lado del cliente se encuentran principalmente el JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

A continuación se hace una breve descripción de algunos de estos lenguajes:

CGI

Es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor. Actualmente se encuentra un poco desfasado por diversas razones entre las que destaca la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta. (ALVAREZ)

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ o Visual Basic pueden ser también empleados para construirlos.

Algunas desventajas de la programación en CGI son las siguientes:

- Los resultados se escriben directamente con el CGI, así que el código del programa se mezcla con el del HTML haciendo difícil su comprensión y mantenimiento.
- Cada programa CGI que se pone en marcha lo hace en un espacio de memoria propio. Así, si tres usuarios ponen en marcha un CGI a la vez se multiplicará por tres la cantidad de recursos que ocupe ese CGI. Esto significa una grave ineficiencia.

Perl

Es un lenguaje de programación web usado para construir y desarrollar aplicaciones CGI para ser ejecutadas en la WWW, principalmente para realizar consultas a bases de datos como Oracle, SQL-Server.

Es un lenguaje para manipular textos, archivos y procesos, extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Es un lenguaje libre de uso, es decir, es gratuito y fue creado con la finalidad de administrar las tareas propias de sistemas UNIX, pero actualmente esta generalizadas para todos los sistemas operativos.

La ventaja más importante de PERL es que no necesita ser recompilado, es un lenguaje interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP.

ASP

Es una tecnología desarrollada por Microsoft para la creación de páginas dinámicas para sus servidores de páginas Web. Una página ASP es una página html que incluye uno o más scripts (pequeños

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

programas). ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (JavaScript de Microsoft) y componentes de servidor ActiveX reutilizables. (ALVAREZ)

Los servidores que emplean este lenguaje son los que funcionan con sistemas operativos de la familia Windows NT, pero también se puede utilizar en una PC con windows 98 si instalamos un servidor denominado Personal Web Server. Incluso en sistemas Linux también podemos trabajar con las ASP si instalamos un componente denominado Chilisoft, aunque sería conveniente trabajar sobre el servidor web para el que está pensado: Internet Information Server.

Actualmente se ha presentado ya la segunda versión de ASP, el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro.

ASP.NET

ASP.NET es más que una nueva versión de las páginas Active Server (ASP); proporciona un modelo de desarrollo Web unificado que incluye los servicios necesarios para que los programadores creen aplicaciones Web para la empresa. Si bien ASP.NET es en gran medida compatible con la sintaxis de ASP, proporciona también un modelo de programación y una estructura nuevos para crear aplicaciones más escalables y estables que ayuden a proporcionar mayor protección. Las aplicaciones ASP se pueden ampliar agregándoles funcionalidad de ASP.NET.

ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

- **Mejor rendimiento.** ASP.NET es un código de Common Language Runtime compilado que se ejecuta en el servidor. ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento.
- **Compatibilidad.** Con herramientas de primer nivel. El marco de trabajo se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación de Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y colocar y la

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.

- **Eficacia y flexibilidad.** Debido a que ASP.NET se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma .NET se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes.
- **Simplicidad.** ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios.
- **Facilidad de uso.** ASP.NET emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Una aplicación ASP.NET Framework se implementa en un servidor sencillamente mediante la copia de los archivos necesarios al servidor. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.
- **Escalabilidad y disponibilidad.** ASP.NET se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

- **Posibilidad de personalización y extensibilidad.** ASP.NET presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado.
- **Seguridad.** Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

JSP

Es una tecnología orientada a crear páginas web con programación en Java. Con JSP se pueden crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java.

PHP

Es el acrónimo de Hipertext Preprocesor. Este lenguaje es gratuito y multiplataforma, rápido, con una gran librería de funciones y mucha documentación.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como está desarrollado en política de código abierto, ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, lanzada con un gran propósito: mejorar los mecanismos de la Programación Orientada a Objetos (POO) para solucionar las carencias de las versiones anteriores.(ALVAREZ)

Es un lenguaje de programación que por poseer una extensa librería de funciones tiene la capacidad de realizar diversos tipos de aplicaciones web. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por citar algunos ejemplos y permite el acceso a su código fuente.

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el ASP de Microsoft, pero con algunas ventajas sobre este debido a su gratuidad, independencia de

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

plataforma, rapidez y seguridad. Funciona más rápido que ASP si está montado sobre servidores UNIX o LINUX.

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC. Incluye funciones para el envío de correo electrónico, upload de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales. Con PHP también se pueden realizar y consumir Servicios Web.

JavaScript

Es un lenguaje muy sencillo de aprender, compatible con la mayoría de los navegadores y tiene la capacidad de trabajar con muchas propiedades de los exploradores Web, páginas Web y en ocasiones con el propio sistema donde se ejecuta el explorador.

Es un lenguaje que tiene muchas posibilidades, permite la programación de pequeños script y grandes programas. Además, pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Se puede hacer cargo de gran parte de las funciones del cliente de las cuales se encarga el servidor y uno de los mejores ejemplos es la validación. Con JavaScript es posible validar los elementos antes de que el usuario se los envíe al servidor y de esta forma reduce la cantidad de transacciones que se efectúan a través de http y las posibilidades de que se genere un error durante la inserción de datos. También puede leer y escribir cookies, una operación que hasta ahora únicamente podía desarrollar el servidor Web.

No necesita una fase de compilación como Java o C y el explorador no ha de cargar ninguna máquina virtual para ejecutar el código. Solo hay que crear el código y cargarlo. JavaScript utiliza una arquitectura orientada a objetos parecida a la de Java o C++.

Visual Basic Script.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Está basado en Visual Basic, tanto su sintaxis como la manera de trabajar están muy inspirados en él. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.

El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas web es muy similar al utilizado en Javascript y los recursos a los que se puede acceder también son los mismos: el navegador.

2.8 Plataforma.Net

.NET permite escribir programas o aplicaciones para entornos distribuidos. Para lograr esto, .NET facilita la creación de Servicios Web y Aplicaciones Web. La Arquitectura .NET no solo ayuda a crear nuevos programas, sino que también ofrece la posibilidad de mejorar los programas ya existentes.

La Arquitectura.NET facilita la creación de complejas aplicaciones puesto que proporciona clases y métodos predefinidos dentro de la librería básica de clases .NET y también gestiona la ejecución de las aplicaciones creadas. Además ofrece un entorno de desarrollo de aplicaciones llamado Visual Studio .NET que consta de varios lenguajes de programación como Visual Basic .NET, Visual C#, Visual FoxPro y Visual C++ .NET. Estos lenguajes de programación combinan las características de los lenguajes existentes con nuevas posibilidades para proporcionar un potente sistema de desarrollo.(FERGUSON *et al.* 2003)

El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son:

- Lenguajes de compilación
- Biblioteca de clases de .Net
- CLR (Common Language Runtime)

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

Entre sus características mas significativas se destaca la interoperabilidad con otros entornos, además de que en su conjunto presenta soporte para desarrollar aplicaciones independientes del lenguaje, soporte para Programación Orientada a Objetos (OOP), soporte para aplicaciones Web y para servicios Web.

Es importante señalar que esta plataforma presenta una serie de inconvenientes. Algunos de estos se ven reflejados en la administración del código pues se introducen factores de sobrecarga que repercuten en la demanda de otros requisitos del sistema. También podemos decir que en el nivel de administración del código depende en gran medida del lenguaje que se utilice para programar. Por ejemplo, mientras que Visual Basic .Net es un lenguaje totalmente administrado, C# permite la administración de código de forma manual, siendo por defecto también un lenguaje administrado. Mientras que C++ es un lenguaje no administrado en el que se tiene un control mucho mayor del uso de la memoria que hace la aplicación.

2.8.1 La aplicación VisualStudio.Net 2005 como software de autor.

Visual Studio es un Sistema de Desarrollo Integrado (IDE) creado por Microsoft. Éste permite a los desarrolladores crear programas, sitios de internet, aplicaciones para el internet, y servicios web que pueden ser usados en varios sistemas operativos,

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J#. Admite diseñar la aplicación de forma visual, permitiendo en cada momento acceder al código generado y sirviendo además como herramienta de compilación y depuración.

La ventaja de utilizar Visual Studio .NET es que proporciona herramientas que hacen que el desarrollo de aplicaciones sea mucho más rápido, sencillo y confiable. Entre estas herramientas se incluyen:

- Diseñadores visuales para formularios Windows Forms con controles de arrastrar y colocar.
- Editores de código inteligentes que incluyen finalización de instrucciones, comprobación de sintaxis y otras características de IntelliSense.
- Compilación y depuración integradas.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

- Utilidades de administración de proyectos para la creación y administración de archivos de aplicación, incluida la implementación en servidores de intranet o Internet.

2.8.2 Lenguaje de programación a utilizar: C#.

C# es un lenguaje de programación orientado a objetos desarrollado por la Corporación Microsoft como parte de su iniciativa titulada .NET. C# es algo más que un simple lenguaje de programación, ya que reúne todas las características que son demandadas en un lenguaje de programación moderno, y el cuál está diseñado especialmente pensando en los desarrolladores actuales.

C# es una versión avanzada de C y de C++ y se ha diseñado especialmente para el entorno .NET. Es un nuevo lenguaje orientado a objetos empleado por programadores de todo el mundo para desarrollar aplicaciones que se ejecuten en la plataforma .NET. Este lenguaje constituye una solución ideal para el desarrollo de aplicaciones de alto nivel. Con C# se puede desarrollar todo tipo de proyectos de aplicaciones cliente / servidor. Amplia las capacidades de C, C++, Visual Basic (VB) y Java para proporcionar un completo entorno de desarrollo. C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

Por sus características establece una línea totalmente atractiva para el desarrollo de grandes aplicaciones informáticas. Posee un amplio sistema de clases y librerías para el tratamiento de cadenas de texto, las variables disponen de seguridad de codificación estricta de tipos, presenta un recolector de basura, es decir, que no es necesario incluir instrucciones de destrucción de objetos, así como que tiene compatibilidad con conceptos de metadatos extensibles, entre otras.

2.9 Sistema de Gestión de Base de Datos (SGBD).

Un Sistema de Gestor de Base de Datos (SGBD) es un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Su principal objetivo es evitar la redundancia de los datos, eliminando así la inconsistencia de los mismos, mejorar los mecanismos de

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

seguridad de los datos y la privacidad. Se caracteriza fundamentalmente por permitir una descripción centralizada de los datos y por la posibilidad de definir vistas parciales de los mismos para los diferentes usuarios. Sus funciones principales se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario y el cumplimiento de las normas de tratamiento de datos.

Entre los SGBD comúnmente utilizados en el mundo tenemos Microsoft SQL Server, MySQL, PostgreSQL, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

SQL (Structured Query Language)

El SQL es un lenguaje de programación interactivo y estandarizado, declarativo de acceso, extracción y actualización de bases de datos relacionales, permite especificar diversos tipos de operaciones sobre las mismas. Agrupa características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla. Es el Sistema de Administración de Bases de Datos Relacionales de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software. El SQL Server permite lograr una gran velocidad en el procesamiento de transacciones y agilidad en todas sus operaciones.

Ventajas

- Soporta la configuración automática y la auto-optimización.
- Administración multiservidor para un gran número de servidores.
- Gran variedad de opciones de duplicación de cualquier base de datos.
- Acceso universal a los datos (Universal Data Access).
- Fácil de usar.
- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

- Posee los mejores registros de los benchmarks independientes (TCP) tanto en transacciones totales como en coste por transacción.
- Gestión: Con una completa interfaz gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

Desventajas

- Licencias con costos altos.
- Plataformas Windows.

MySQL

MySQL es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto. (Open Source). Es un gestor de Bases de Datos multiusuario que gestiona bases de datos relacionales poniendo las tablas en ficheros diferenciados. Es muy criticado porque carece de muchos elementos vitales en bases de datos relacionales y no es posible lograr una integridad referencial verdadera. Es más utilizado en plataformas Linux ya que fue diseñado para los sistemas Unix aunque puede usarse en otras plataformas. Su uso en un servidor web es gratuito salvo en los casos que se necesite el uso de aplicaciones especiales.

Ventajas

- Diseñado con el objetivo de aumentar la velocidad.
- Consume muy pocos recursos de CPU y memoria. Muy buen rendimiento.
- Tamaño del registro sin límite.
- Buena integración con PHP.
- Utilidades de administración (PhpMyAdmin).
- Buen control de acceso usuarios-tablas-permisos.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

Desventajas

- Es gratis para aplicaciones de código abierto, de lo contrario hay que pagar licencia comercial.

PostgreSql

Está considerado como el SGBD de código abierto más avanzado del mundo. PostgreSQL es totalmente libre. Recomiendan la utilización de PostgreSQL para la elaboración de un sistema robusto y para lograr mayor escalabilidad. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales de alto calibre intentando estar a la altura de Oracle, Sybase o Interbase.

Es un SGBD objeto-relacional altamente extensible, soporta operadores y tipos de datos definidos por el usuario y es capaz de manejar complejas rutinas y reglas de modo que su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas, el control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y valores no atómicos (atributos basados en vectores y conjuntos). Además cuenta con un mejor soporte para subselects, triggers, vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos. Este sistema cuenta con una API (Application Program Interface) flexible lo cual ha permitido dar soporte para el desarrollo con PostgreSQL en diversos lenguajes de programación entre los que se incluyen: Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

Se hace necesario señalar que a diferencia de la mayoría de otros sistemas de bases de datos que usan bloqueos para el control de concurrencia, Postgres utiliza un modelo multiversión MVCC o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), mediante el cual evita el problema de los bloqueos por completo. Esto significa que mientras se consulta una base de datos, cada transacción ve una imagen de los datos (una versión de la base de datos) como si fuera tiempo atrás, sin tener en cuenta el estado actual de los datos que hay por debajo.

Esto evita que la transacción vea datos inconsistentes que pueden ser causados por la actualización de otra transacción concurrente en la misma fila de datos, proporcionando aislamiento transaccional para

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

cada sesión de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Los bloqueos de tabla han sido sustituidos por el control de concurrencia multiversión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.

Ventajas

- Soporta transacciones y desde la versión 7.0, llaves foráneas (integridad referencial).
- Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL.
- Escala muy bien al aumentar el número de CPUs y la cantidad de RAM.

Desventajas

- Consume bastantes recursos y carga más el sistema.

Como SGBD a utilizar para el desarrollo de la base de datos se seleccionó el PostgreSQL 8.2 por las ventajas antes mencionadas y por requerimientos del cliente.

2.10 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta.

UML es un lenguaje estándar de modelado para software que es utilizado por RUP para la representación de sus modelos. Es un lenguaje para la visualización, especificación, construcción y documentación de los artefactos de un sistema que involucra una gran cantidad de software.

Básicamente, UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. Además les proporciona un vocabulario y mecanismos de extensibilidad que le permiten al usuario refinar su sintaxis y su semántica, por tanto, puede ajustarse a un sistema, proyecto o proceso de desarrollo específico si es necesario.

2.11 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado, y desarrolladores aún más insatisfechos.

Escoger la metodología que va a guiar el proceso de desarrollo de un sistema es fundamental. Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos.

En la actualidad existen diversas metodologías para el desarrollo de software. Las más significativas por su alcance y desarrollo son el Proceso Unificado de Modelado (RUP), Extreme Programming (XP) y Microsoft Solution Framework (MSF).

Para controlar, y planificar la propuesta que presenta este trabajo, se decidió utilizar como metodología el Proceso Unificado de Modelado (RUP), por sus características y las facilidades que aporta a todo el proceso.

RUP es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Es la metodología que dentro de un proyecto de desarrollo de software define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Se encuentra preparada para desarrollar grandes y complejos proyectos. Fundamentalmente se caracteriza por ser un proceso dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- **Iterativo e Incremental:** Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla algunos en mayor grado que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Como su enfoque está basado en modelos utiliza UML como lenguaje de representación visual, siendo este una parte esencial de este proceso.

La metodología RUP divide el ciclo de desarrollo de un producto software en 4 fases fundamentales, cada una con un propósito bien definido.

Fases	Propósito
Inicio o Conceptualización	Definir el alcance del proyecto.
	Entender que se va a construir.
Elaboración	Definir la arquitectura del sistema.
	Entender cómo se va a construir.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

Construcción	Completar el esqueleto del sistema con funcionalidad. Construir una versión Beta del producto.
Transición	Disponibilizar el sistema para los usuarios finales. Construir una versión final.

Tabla 1. Fases de RUP.

RUP comprende varios flujos de trabajo en toda su integración. Algunos de estos flujos son los que a continuación describimos brevemente.

1. Modelamiento del negocio: describir los procesos de negocio.
2. Requerimientos: definir qué es lo que exactamente el sistema debe hacer, con las restricciones que se imponen.
3. Análisis y Diseño: describir cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas, indica con precisión lo que se debe programar.
4. Implementación: implementar el sistema en término de componentes.
5. Prueba: verificar que el comportamiento del software obtenido es el correcto y el deseado por el cliente.

2.12 Rational Rose como herramienta de trabajo.

Las herramientas CASE (Computer Aided Software Engineering), son herramientas muy completas y ofrecen amplias potencialidades para modelar visualmente la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Estas herramientas ayudan al analista, o a todo el equipo de trabajo para administrar el proceso durante una o más fases del desarrollo de software y facilitar el mantenimiento o la creación de nuevas versiones posteriores.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar

Para ello se usará el Rational Rose: herramienta de Modelación por excelencia muy usada mundialmente por las facilidades que brinda haciendo uso de la notación estándar en la arquitectura de software (UML).

2.13 Conclusiones.

En este capítulo se hizo un análisis de las principales tecnologías a utilizar en el desarrollo de la propuesta de solución. Se fundamentó la elección del lenguaje de programación, el sistema gestor de bases de datos, el servidor Web, y la metodología de desarrollo de software con elementos específicos.

CAPITULO 3. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción.

En el presente capítulo se realiza la descripción de la solución propuesta y para ello se utiliza un modelo de dominio. Además, se enumeran los requisitos funcionales y no funcionales que debe tener el sistema, lo que permite hacer una concepción general del sistema, e identificar mediante Diagramas de Casos de Uso del Sistema, las relaciones de los actores que interactúan con el mismo, y las secuencias de acciones con las que interactúan.

3.2 Modelo Conceptual.

Un modelo conceptual es una representación de conceptos en un dominio del problema. Además este modelo nos muestra asociaciones entre conceptos y atributos de conceptos. En UML ilustramos el modelo conceptual como un diagrama de estructura estática, donde no se define ninguna operación. Es importante comprender que un modelo conceptual no es una descripción del diseño del software, como una clase de java o C++.

Además de descomponer el espacio del problema en unidades comprensibles (conceptos), la creación de este modelo contribuye a esclarecer la terminología o nomenclatura del dominio. Se puede ver como un modelo que comunica cuales son los términos importantes y como se relacionan entre si. Esto ayuda a los usuarios, clientes, desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

Se decidió realizar solamente un modelo de dominio, puesto que la aplicación no da lugar a ningún tipo de negocio en la universidad y en vista de las bondades que ofrece la Web el sistema se apoyará en la misma para el desarrollo del Directorio de Personas.

Capítulo 3. Presentación de la solución propuesta

Primeramente se identifican todos los conceptos que se utilizarán en el modelo conceptual, mediante un glosario de términos sobre los nombres:

- **Usuario:** persona que interactúa con la aplicación de búsqueda de persona: Directorio de Persona.
- **Administrador:** persona encargada de configurar la aplicación que actualiza la información de las personas para que se ejecute a una hora determinada, como una tarea del sistema.
- **Directorio_Persona:** aplicación de búsqueda con la que el usuario interactúa directamente, encargado de mostrar los resultados a estos de acuerdo con los criterios de búsqueda establecidos por el usuario.
- **Aplicación_Actualización:** aplicación encargada del proceso de actualización de los datos de las personas que pertenecen a la universidad.
- **Reporte_Persona:** resultado de la búsqueda de persona, puede estar compuesto por fichas de personas.
- **Ficha_Persona:** contiene información mas detallada de cada persona.
- **Reporte_Estadísticas:** resultado de la búsqueda de los reportes estadísticos, tales como la cantidad de usuarios que hicieron uso del Directorio de Personas, cantidad de consultas, tipo y cantidad de navegadores y sistemas operativos bajo el cual el usuario se conecta desde su PC.
- **Proveedor de información:** sistemas que suministran información, tales como Akademos y Trabajadores.
- **Información:** datos de las personas, ya sea estudiantes o trabajadores del centro.
- **BD_Directorio:** base de datos donde se encuentra almacenada toda la información actualizada de las personas.

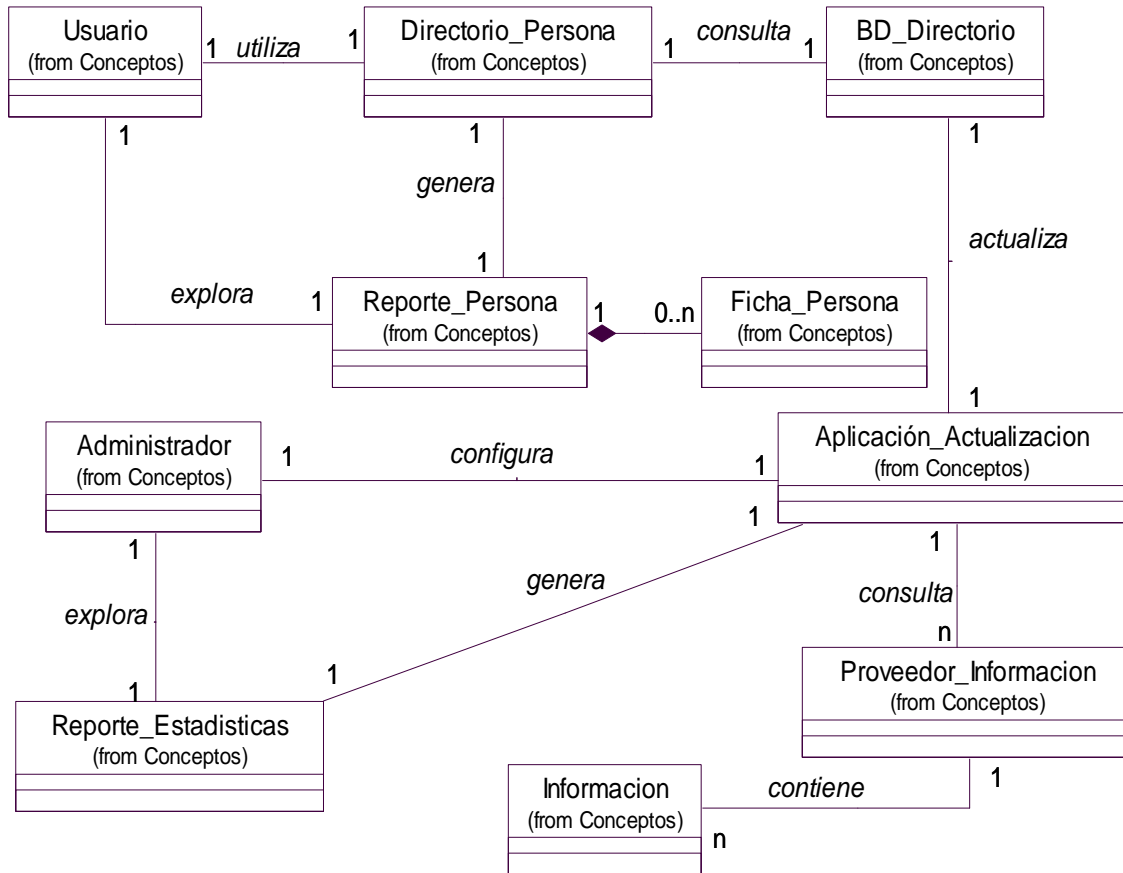


Figura 1. Modelo conceptual

3.3 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. En función del objetivo planteado el sistema debe ser capaz de:

R1- Buscar persona

- Mostrar interfaz para realizar la búsqueda: Búsqueda simple o con asistente.

Realizar búsqueda simple

- Crear patrones de búsqueda.
- Recuperar datos utilizando el patrón de búsqueda.
- Mostrar listado de personas que cumplan con los requisitos con la posibilidad de ver información mas detallada de la persona en cada caso.
- Mostrar mensaje en caso de que la búsqueda no produzca ningún resultado.
- Permitir la reformulación de la búsqueda desde la misma página de resultados.
- Acceder a la ficha de la persona seleccionada en la lista.
- Presentar al usuario la interfaz de la ficha de la persona seleccionada.
- Integrar en la ficha toda la información existente de la persona.

Realizar búsqueda utilizando asistente.

- Mostrar interfaz para la búsqueda utilizando asistente.
- Seleccionar los requisitos en el asistente para realizar la búsqueda.
- Crear patrón de búsqueda con los requisitos.
- Recuperar datos utilizado el patrón de búsqueda.
- Mostrar listado de personas que cumplan con los requisitos con la posibilidad de ver información mas detallada de la persona en cada caso.
- Mostrar mensaje en caso de que la búsqueda no produzca ningún resultado.
- Permitir la reformulación de la búsqueda desde la misma página de resultados.
- Acceder a la ficha de la persona seleccionada en la lista.
- Presentar al usuario la interfaz de la ficha de la persona seleccionada.
- Integrar en la ficha toda la información existente de la persona.

R2- Establecer conexión.

- Establecer conexión con el servicio web.

- Testear la conexión.

R3-Configurar aplicación.

- Establecer horario de actualización.

R4- Actualizar datos estudiantes.

- Establecer conexión con la base de datos.
- Limpiar registros correspondientes a los estudiantes en la base de datos.
- Obtener datos de estudiantes.
- Llenar la base de datos con los datos actualizados de los estudiantes.

R5- Actualizar datos trabajadores.

- Establecer conexión con la base de datos.
- Limpiar registros correspondientes a los trabajadores de la base de datos.
- Obtener datos de trabajadores.
- Llenar la base de datos con los datos actualizados de los trabajadores.

R6- Visualizar estatus de la operación

- Visualizar progreso de la transacción y la actualización.
- Visualizar mensaje de fin de la actualización.
- Visualizar mensaje error en caso de no haber sido satisfactoria la actualización.

R7- Visualizar reportes estadísticos.

- Visualizar cantidad de usuarios que hicieron uso del Directorio de Personas.
- Visualizar búsquedas realizadas por los usuarios.
- Visualizar cantidad de consultas realizadas al Directorio de Personas.
- Visualizar cantidad de cada tipo de sistema operativo bajo el cual el usuario del Directorio de Personas se conecta desde su PC.
- Visualizar cantidad de cada tipo de navegador bajo el cual el usuario del nuevo directorio se conecta desde su PC.

R8- Visualizar manual de usuario

R9- Visualizar ayuda

3.4 Requerimientos No Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que los clientes y los usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Requerimiento de Apariencia o Interfaz externa

- Diseño sencillo, que no requiera de mucho conocimiento para utilizar el sistema.

Requerimiento de Usabilidad

- El sistema de actualización podrá ser usado solamente por el administrador.
- El Directorio de Personas podrá ser usado por todos los usuarios del dominio UCI.

Requerimiento de Rendimiento

- El sistema de actualización manejará grandes volúmenes de información.
- El Directorio de Persona soportará varias conexiones al mismo tiempo.

Requerimiento de Soporte

- Se requiere de un servidor de Base de Datos Postgres capaz de almacenar grandes volúmenes de información, con velocidad de procesamiento eficiente y tiempo de respuesta rápido.
- Plataforma window.

Requerimiento de Hardware

- Requiere estar instalada en una PC Pentium IV o superior, más de 1Gb de RAM.

Requerimiento de Software

- Se debe disponer en el servidor con Windows XP, Windows 2003 Server. Se utilizará como lenguaje de programación: C# y como gestor de Base de Datos: PostgreSQL

Requerimiento de Funcionalidad

- Capacidad de Búsqueda y actualización de los datos en el menor tiempo posible.

3.5 Descripción del Sistema Propuesto.

Basándose en los requerimientos planteados anteriormente y en el objetivo de este trabajo, el sistema que se propone debe tener dos módulos, un módulo para la búsqueda de personas y un módulo para la actualización de los datos de las personas de la universidad que interactúa con diferentes sistemas que le proveen información.

El módulo de búsqueda de personas es una aplicación web, donde todos los usuarios de la UCI tendrán acceso, ya sea estudiantes, trabajadores o cualquier personal vinculado a la institución. A través de este sistema se pueden hacer búsquedas simples y complejas. Esta aplicación hará uso de una base de datos propia la cual se actualizará cada cierto tiempo.

Capítulo 3. Presentación de la solución propuesta

Para utilizar las funcionalidades del sistema de actualización se considera la existencia de un administrador, que no es más que el encargado de configurar la ejecución de la aplicación que inicia el proceso de la actualización de los datos. El administrador podrá ver cuando lo desee reportes estadísticos, tales como la cantidad de usuarios que hicieron uso del directorio, cantidad de consultas, etc. También podrá ver el estado de las transacciones o si estas han sido ejecutadas y finalizadas satisfactoriamente.

Mientras ocurre el proceso de actualización, el servicio del directorio no será detenido, los accesos a la base de datos se manejarán mediante el gestor de control de concurrencia, en el caso de PostgreSQL (MVCC), Control de Concurrencia Multiversión. En caso de ocurrir algún problema en la conexión con los servicios web de los proveedores de información, fallo de energía o cualquier otra anomalía o incidente el sistema intentará en un lapso de tiempo iniciar nuevamente el proceso de actualización. Si no resulta satisfactorio el proceso de actualización, se registra en una tabla de la base de datos el incidente. Esta información podrá ser vista posteriormente por el administrador, se le notificará que ha ocurrido algún error en la actualización mediante un correo electrónico y un aviso de Windows.

Este sistema se encontrará instalado en un servidor de la universidad y el mismo se ejecutará de acuerdo a la hora y fecha establecida por el administrador. Para el almacenamiento de los datos el sistema cuenta con una base de datos propia, que en este caso será la que utilizará el nuevo Directorio de Personas para ofrecer sus servicios. La aplicación estará ejecutándose como un servicio de Windows y podrá ser minimizada, maximizada y cerrada desde la barra de tareas.

Este módulo pretende utilizar la información existente en las bases de datos de los sistemas que le proveen de la misma a través de los servicios web que ofrecen los sistemas de Akademos, Trabajadores, entre otros.

3.5.1 Descripción de los actores.

Los actores de un sistema son agentes externos: aquellas personas o sistemas que interactúan con él. En la siguiente tabla se describen los actores del sistema.

ACTOR	JUSTIFICACIÓN
Usuario	Representa a una persona que interactúa con la aplicación de búsqueda de personas (Directorio de Personas), ya sea estudiante, trabajador u otros.
Administrador	Representa a la persona encargado de configurar la ejecución de la aplicación de actualización de la información de las personas de la universidad.
Reloj	Evento que inicia los casos de usos de actualización.

Tabla 2. Actores del sistema.

3.5.2 Diagrama de Casos de Uso del Sistema.

Los casos de uso son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación.

3.5.2.1 Paquetes del Sistema.



Figura 2. Paquetes del sistema.

3.5.2.2 Diagramas por paquetes.

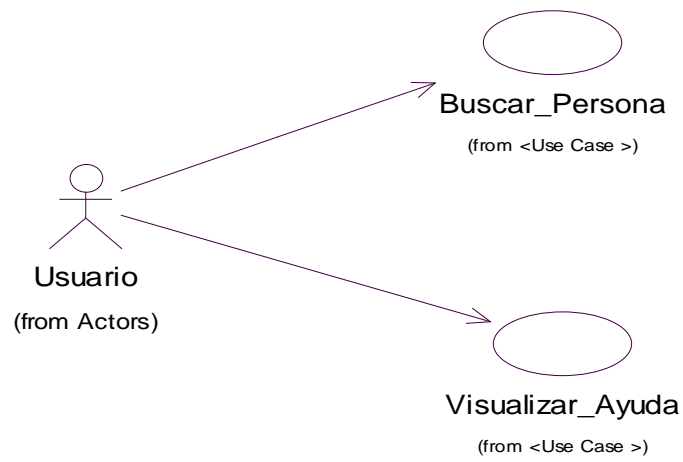


Figura 3. Diagrama CU del paquete: Buscar_Persona

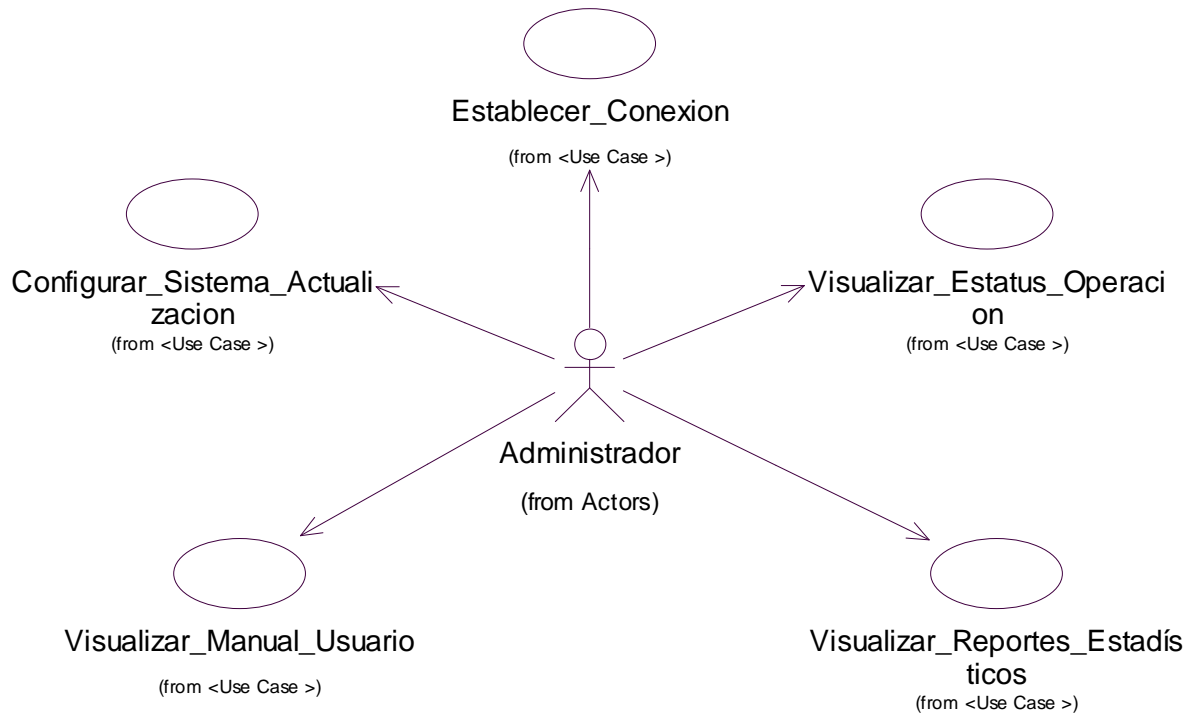


Figura 4. Diagrama CU del paquete: Administración

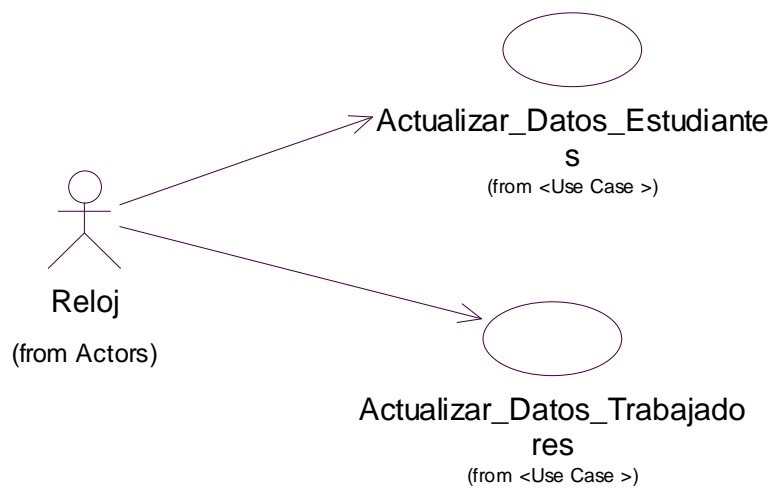


Figura 5. Diagrama CU del paquete: Actualización_Información_Persona

3.5.3 Expansión de los Casos de Uso.

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. Este sería el último paso en el análisis, para pasar a la construcción de la solución propuesta.

Caso de uso:	Buscar_Personas	
Actor(es):	Usuario (inicia).	
Propósito:	Mostrar la información detallada de las personas	
Resumen:	El caso de uso se inicia cuando el Usuario decide buscar información de cualquier tipo de persona, introduce los parámetros para efectuar la búsqueda en caso de que vaya a hacer una Búsqueda Simple o selecciona las características de la búsqueda si usa el Asistente. Luego se muestra una lista con las personas que cumplan con los requisitos establecidos por el Usuario junto con algunos datos básicos. El caso de uso no termina hasta que el usuario lo decida.	
Referencias:	R1	
Precondiciones:	El usuario se debe haber autenticado contra el dominio UCI.	
Acción del Actor	Respuesta del Sistema	
	1. Muestra interfaz para realizar la búsqueda. a) Si selecciona la opción Búsqueda Simple, ver sección: "Búsqueda Simple" b) Si selecciona la opción Utilizar Asistente, ver sección: "Búsqueda Utilizando Asistente".	
Sección: "Búsqueda Simple"		
1. El usuario introduce el parámetro	2. El sistema consulta la base datos del Directorio de	

Capítulo 3. Presentación de la solución propuesta

para realizar la búsqueda y presiona el botón Buscar .	Personas y le muestra al usuario el resultado de la búsqueda. Ver Sección: "Resultado de la Búsqueda".
Sección: "Búsqueda Utilizando Asistente"	
	1. El sistema muestra la interfaz de búsqueda usando el asistente
2. El usuario selecciona el tipo de búsqueda que desea hacer, ya sea estudiantes o trabajadores. Para ello selecciona los requisitos para la búsqueda y presiona el botón Buscar.	3. El sistema consulta la base datos Directorio de Personas y le muestra al usuario el resultado de la búsqueda. Ver Sección: "Resultado de la Búsqueda".
Sección: "Resultado de la Búsqueda"	
	1. El sistema lista los resultados recuperados por la búsqueda de la Sección: "Búsqueda Simple" o "Búsqueda Utilizando Asistente".
2. El Usuario tiene dos opciones, mostrar información mas detallada de una persona en específico o reformular la búsqueda.	3. a) Si elige mostrar información mas detallada. Ver Sección: "Mostrar Ficha" b) Si elige reformular la búsqueda. Ver Sección: "Reformular Búsqueda"
Sección: "Reformular Búsqueda"	
1. El usuario puede reformular la búsqueda a través de la búsqueda simple o utilizando el asistente para la búsqueda.	2. El sistema consulta la base datos Directorio de Personas y le muestra al usuario el resultado de la búsqueda. Ver Sección: "Resultado de la Búsqueda".
Sección: "Mostrar Ficha"	
	1. El sistema muestra información mas detallada de la persona en una ventana mas pequeña.
Poscondiciones:	Las búsquedas de información se han realizado.
Prioridad:	Crítico

Tabla 3. Descripción CU Buscar_Personas: Paquete Buscar_Personas

Caso de uso:	Establecer Conexión	
Actor(es):	Administrador (inicia).	
Propósito:	Garantizar que los parámetros de conexión sean los correctos	
Resumen:	El caso de uso se inicia cuando el administrador decide establecer conexión con los servicios web, en caso de que no establezca la conexión el sistema debe mostrar una notificación de error y el administrador podrá intentar nuevamente establecer la conexión.	
Referencias:	R2	
Precondiciones:		
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona en el menú Conexión la opción Establecer conexión.	2. El sistema muestra una ventana con las URL de los servicios web con los que debe establecer conexión (Las URL pueden ser editadas)	
3. El administrador puede seleccionar con que servicio establecer conexión o Introducir una URL nueva. Luego presiona el botón testear.	4. El sistema comprueba si existe la URL enviada por el administrador. a) En caso que sea correcta se establece la conexión. b) En caso que sea incorrecta, ver CA1	
Cursos alternos		
CA1	Muestra mensaje error e indica regresar a la acción 3	
Poscondiciones:	Se ha establecido la conexión.	
Prioridad:	Crítico	

Tabla 4. Descripción CU Establecer Conexión: Paquete Administración

Caso de uso:	Configurar aplicación
Actor(es):	Administrador (inicia).

Capítulo 3. Presentación de la solución propuesta

Propósito:	Garantizar que la aplicación actualice la información de las personas a una hora determinada por el administrador.	
Resumen:	El caso de uso se inicia cuando el administrador decide establecer la hora de actualización de la información de las personas.	
Referencias:	R3	
Precondiciones:		
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona en el menú Configuración la opción Establecer actualización	2. El sistema muestra la interfaz correspondiente para que el administrador establezca la fecha y hora de actualización.	
3. El administrador introduce la fecha y hora de actualización y acepta la operación presionando el botón Fijar actualización.	4. El sistema valida los datos introducidos por el administrador. a) En caso de ser correctos el sistema recepciona la fecha y la hora de actualización. b) En caso de ser incorrectos , Ver CA1	
Cursos alternos		
CA1	Muestra mensaje error e indica regresar a la acción 3	
Poscondiciones:	Se ha establecido la fecha y la hora de actualización.	
Prioridad:	Crítico	

Tabla 5. Descripción CU Configurar aplicación: Paquete Administración

Caso de uso:	Actualizar datos estudiantes.
Actor(es):	Reloj (inicia).
Propósito:	Actualizar los datos de los estudiantes de la universidad en la base de datos que utiliza el Directorio de Personas.
Resumen:	El caso de uso se inicia cuando el evento Reloj empieza el proceso de actualización de los datos de los estudiantes en la base de datos que el

Capítulo 3. Presentación de la solución propuesta

	directorio utiliza para prestar sus servicios, el caso de uso no finaliza hasta que no se haya actualizado todos los datos de los estudiantes.
Referencias:	R4
Precondiciones:	Se estableció la conexión con los servicios web.
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Reloj inicia el proceso de actualización de los datos de los estudiantes.	2. El sistema establece la conexión con la base de datos del Directorio de Personas. a) En caso de no haber conexión Ver CA1
	3. El sistema elimina los datos de los estudiantes en la base de datos del Directorio de Personas.
	4. El sistema obtiene los datos de los estudiantes por medio de servicios web. a) En caso fallo en la conexión el sistema intentará restablecer la conexión en un lapso de tiempo. b) En caso de no establecerse la conexión después del fallo en el tiempo establecido, se inserta en la tabla Excepciones el la fecha y el tipo de excepción y se le notifica al administrador mediante un correo electrónico y un aviso de Windows.
	5. El sistema inserta los datos en la base de datos.
Cursos alternos	
CA1	Muestra mensaje error e indica regresar a la acción 2
Poscondiciones:	Se ha actualizado los datos de los estudiantes.
Prioridad:	Crítico

Tabla 6. Descripción CU Actualizar datos estudiantes: Paquete Administración

Capítulo 3. Presentación de la solución propuesta

Caso de uso:	Actualizar datos trabajadores.	
Actor(es):	Reloj (inicia).	
Propósito:	Actualizar los datos de los trabajadores de la universidad en la base de datos que utiliza el Directorio de Personas.	
Resumen:	El caso de uso se inicia cuando el evento Reloj empieza el proceso de actualización de los datos de los trabajadores en la base de datos que el directorio utiliza para prestar sus servicios, el caso de uso no finaliza hasta que no se haya actualizado todos los datos de los trabajadores.	
Referencias:	R5	
Precondiciones:	Se estableció la conexión con los servicios web.	
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Reloj inicia el proceso de actualización de los datos de los trabajadores.	2. El sistema establece la conexión con la base de datos del Directorio de Personas. a) En caso de no haber conexión Ver CA1	
	3. El sistema elimina los datos de los trabajadores en la base de datos del Directorio de Personas.	
	4. El sistema obtiene los datos de los trabajadores por medio de servicios web. a) En caso fallo en la conexión el sistema intentará restablecer la conexión en un lapso de tiempo. b) En caso de no establecerse la conexión después del fallo en el tiempo establecido, se inserta en la tabla Excepciones el la fecha y el tipo de excepción y se le notifica al administrador mediante un correo electrónico y un aviso de Windows.	
	5. El sistema inserta los datos en la base de datos.	

Capítulo 3. Presentación de la solución propuesta

Cursos alternos	
CA1	Muestra mensaje error e indica regresar a la acción 2
Poscondiciones:	Se ha actualizado los datos de los trabajadores.
Prioridad:	Crítico

Tabla 7. Descripción CU Actualizar datos trabajadores: Paquete Administración

Caso de uso:	Visualizar status de la operación	
Actor(es):	Administrador (inicia).	
Propósito:	Conocer el estado de la operación de actualización.	
Resumen:	El caso de uso se inicia cuando el administrador desea ver el estado de las operaciones de la actualización de los datos de las personas.	
Referencias:	R6	
Precondiciones:	Se debe estar ejecutando algún proceso de actualización.	
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona en el menú la opción Ver estatus de la Operación.	2. El sistema muestra en una ventana el estado de la operación. a) Si el proceso de actualización no ha finalizado muestra el progreso de la transacción y actualización. b) Si el proceso de actualización ha finalizado y ha sido satisfactorio muestra mensaje de confirmación de actualización. c) Si el proceso de actualización ha finalizado y no ha sido satisfactorio muestra mensaje de confirmación de error en la actualización.	
Poscondiciones:	Se ha mostrado el estado de la actualización.	
Prioridad:	Secundario	

Tabla 8. Descripción CU Visualizar status de la operación: Paquete Administración

Capítulo 3. Presentación de la solución propuesta

Caso de uso:	Visualizar reportes estadísticos.
Actor(es):	Administrador (inicia).
Propósito:	Conocer los reportes estadísticos de las consultas al Directorio de Personas.
Resumen:	El caso de uso se inicia cuando el administrador decide conocer los reportes estadísticos de las consultas realizadas al Directorio de Personas de la universidad. El mismo escoge los reportes que desea conocer, por ejemplo cantidad de consultas, cantidad de usuarios, etc.
Referencias:	R7
Precondiciones:	El administrador ha sido autenticado.
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción Ver estadísticas.	2. Muestra interfaz correspondiente a los reportes para que el administrador seleccione que tipo de reporte desea ver.
3. Selecciona en el combobox que tipo de reporte desea ver. Los reportes pueden ser los siguientes: <ul style="list-style-type: none"> • Cantidad de usuarios • Búsquedas realizadas • Cantidad de consultas • Sistema Operativo • Excepciones • Navegadores Luego presiona el botón Ver reportes.	4. El sistema consulta la base de datos Directorio y le presenta al administrador el reporte de acuerdo al escogido por el mismo.
Poscondiciones:	La visualización de los reporte se ha efectuado.
Prioridad:	Secundario

Tabla 9. Descripción CU Visualizar reportes estadísticos: Paquete Administración

Capítulo 3. Presentación de la solución propuesta

Caso de uso:	Visualizar manual de usuario
Actor(es):	Administrador (inicia).
Propósito:	Consultar el manual de la aplicación.
Resumen:	El caso de uso se inicia cuando el administrador desee consultar el manual de la aplicación para conocer como funciona el sistema.
Referencias:	R8
Precondiciones:	
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona en el menú Ayuda la opción Manual de usuario.	2. El sistema muestra en una ventana una descripción detallada como funciona el sistema.
Poscondiciones:	Se ha visualizado el Manual de Usuario.
Prioridad:	Auxiliar

Tabla 10. Descripción CU Visualizar manual de usuario: Paquete Administración

Caso de uso:	Visualizar ayuda
Actor(es):	Usuario (inicia).
Propósito:	Consultar la ayuda del Directorio de Personas.
Resumen:	El caso de uso se inicia cuando un usuario (estudiante o trabajador) del Directorio de Persona decide consultar la ayuda para saber cómo realizar una búsqueda.
Referencias:	R9
Precondiciones:	El usuario se ha autenticado
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario presiona el vinculo Consultar Ayuda.	2. El sistema muestra en una ventana una descripción detallada cómo puede efectuar una búsqueda.

Capítulo 3. Presentación de la solución propuesta

Poscondiciones:	Se ha visualizado la ayuda del Directorio de Persona.
Prioridad:	Auxiliar

Tabla 11. Descripción CU Visualizar ayuda: Paquete Buscar_Personas.

3.6 Conclusiones.

Con los elementos que tenemos luego de concluir este capítulo se está en condiciones de pasar a la próxima fase de propuesta de solución del problema, puesto que se han analizado todos elementos necesarios para la elaboración de la aplicación.

CAPÍTULO 4. CONTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN.

4.1 Introducción.

En el presente capítulo se modela el sistema y se describe la arquitectura del mismo. Se muestra el diagrama de clases persistentes y el modelo de datos para comprender mejor que parte de la información ha de ser almacenada.

Se detallan las pautas a seguir para el diseño de la aplicación y los estándares a emplear en la interfaz de la misma. También se describe que elementos se deben considerar para la confección de la ayuda y el manual del usuario y los reportes.

Además, se especifican los estándares a seguir en la codificación, para lograr una mejor organización y un estilo propio del software. Y por último se presenta un modelo de despliegue donde se definen los componentes que deben conformar la estructura física de la aplicación.

4.2 Diagramas de clases de diseño.

El sistema se ha modelado usando el estilo de Capas sobre una arquitectura cliente servidor, específicamente la variante de las tres capas. Cada una puede contener paquetes, en dependencia del nivel de complejidad que presenta, el propósito de dividir en subpaquetes, posibilita mayor entendimiento, independencia, legibilidad y organización a la hora de modelar y entender el sistema. Se decidió por este estilo arquitectónico ya que es el usado por la Dirección de Informatización y el que el cliente nos pidió concebir en el modelado del sistema.

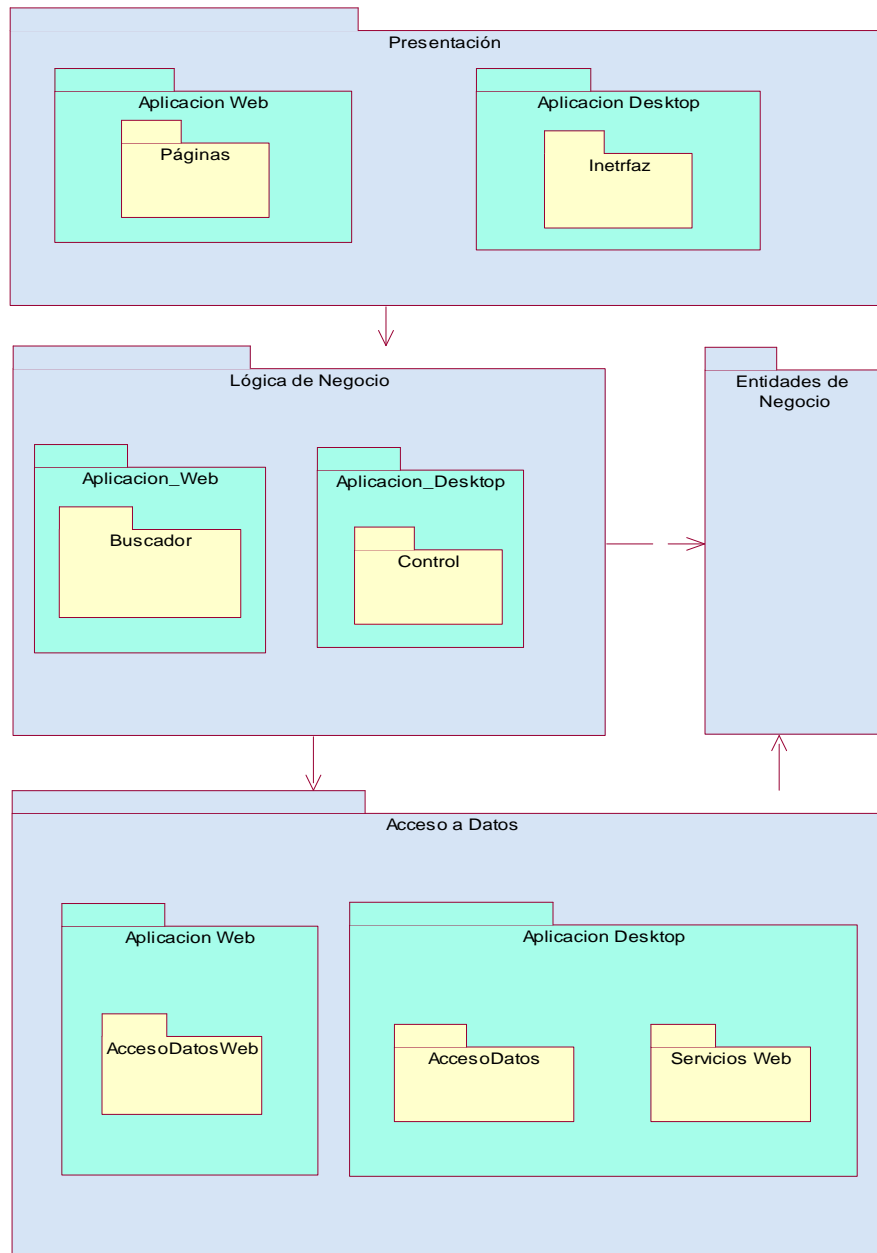


Figura 6. Arquitectura del sistema

Capítulo 4. Construcción de la propuesta de solución

El paquete **Presentación** contiene las clases de presentación del sistema, en este caso, una interfaz web y una interfaz de aplicación desktop. El paquete de **Aplicación Web** contiene un subpaquete **Páginas**, que contiene las páginas para la gestión de información de las personas. El paquete **Aplicación Desktop** contiene un subpaquete **Interfaz**, donde se encuentran las interfaces que se utilizan en el manejo de los datos en el proceso de actualización.

El paquete **Lógica de Negocio** está formado por dos subpaquetes. El subpaquete **Buscador** de la **Aplicación Web** que contiene las clases para establecer los criterios de búsquedas de personas. Por último el subpaquete Control de la **Aplicación Desktop** que contiene las clases para el proceso de actualización de la información de las personas.

El paquete **Entidades de Negocio** contiene clases que no tienen comportamiento, sólo propiedades y son representaciones de entidades reales del dominio, todas son clases persistentes que son accedidas por las clases de los paquetes **Lógica de Negocio y AccesoDatos**.

El paquete **Acceso a Datos** contiene las clases para hacer posible la persistencia y recuperación de objetos. En la **Aplicación Web** tenemos un subpaquete **Acceso datos Web**, contiene las clases encargadas de acceder a la base de datos para manipular la persistencia de las entidades. La **Aplicación Desktop** presenta dos subpaquetes, el paquete Servicios Web que abarca las clases para acceder a servicios web de otros sistemas en la universidad, necesario para obtener diversas informaciones sobre personas y actualización de la misma. El otro subpaquete comprendido en esta aplicación es el subpaquete **Acceso Datos** que se desarrolla bajo las mismas funcionalidades que el subpaquete Acceso Datos Web. El Paquete **Acceso a Datos** en general permite a la aplicación abstraerse del origen de los datos y de la lógica de su persistencia, logrando un bajo acoplamiento entre sus componentes.

Paquete Presentación

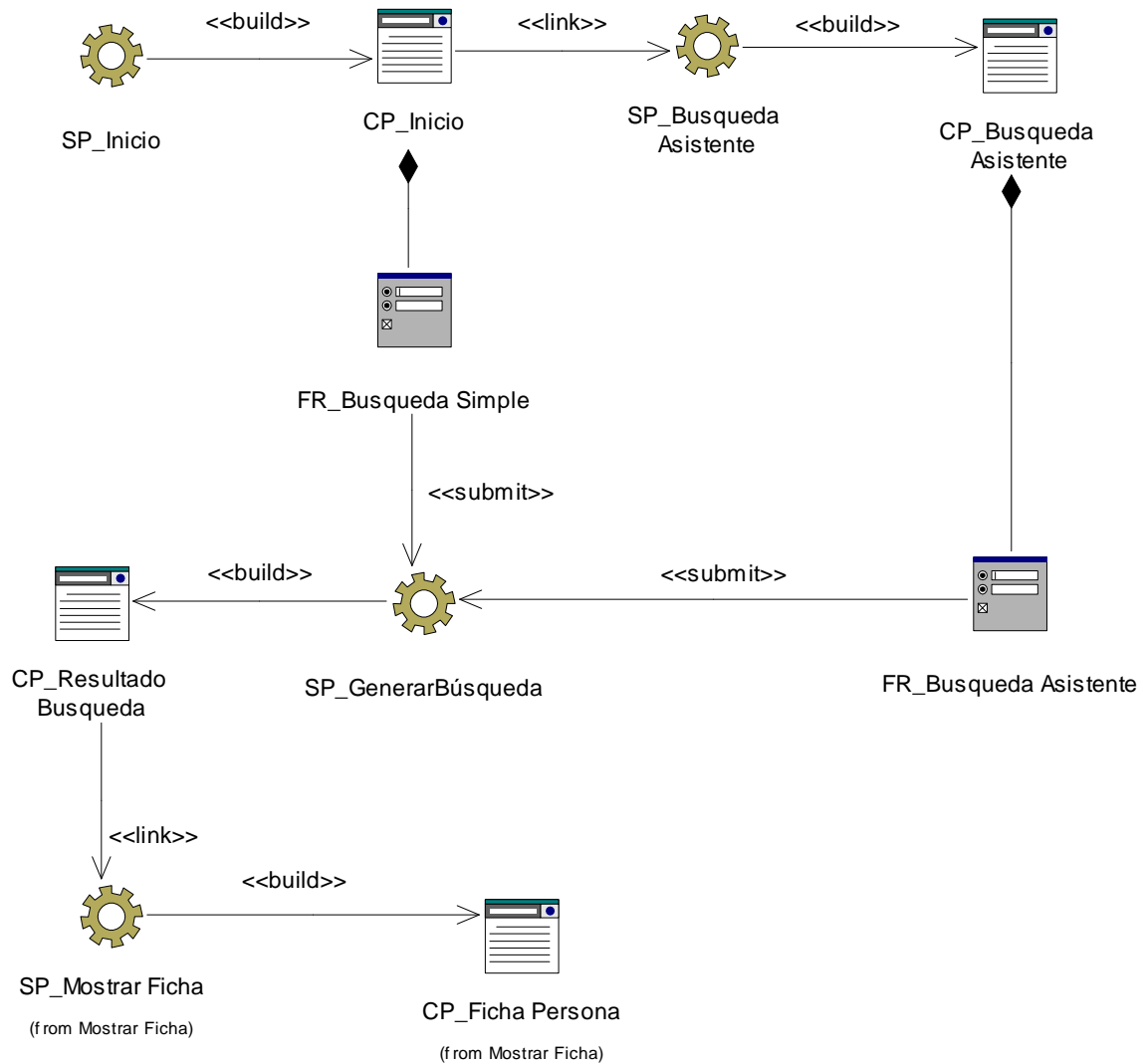


Figura 7. Diagrama Subpaquete Páginas. Ver Anexo 2

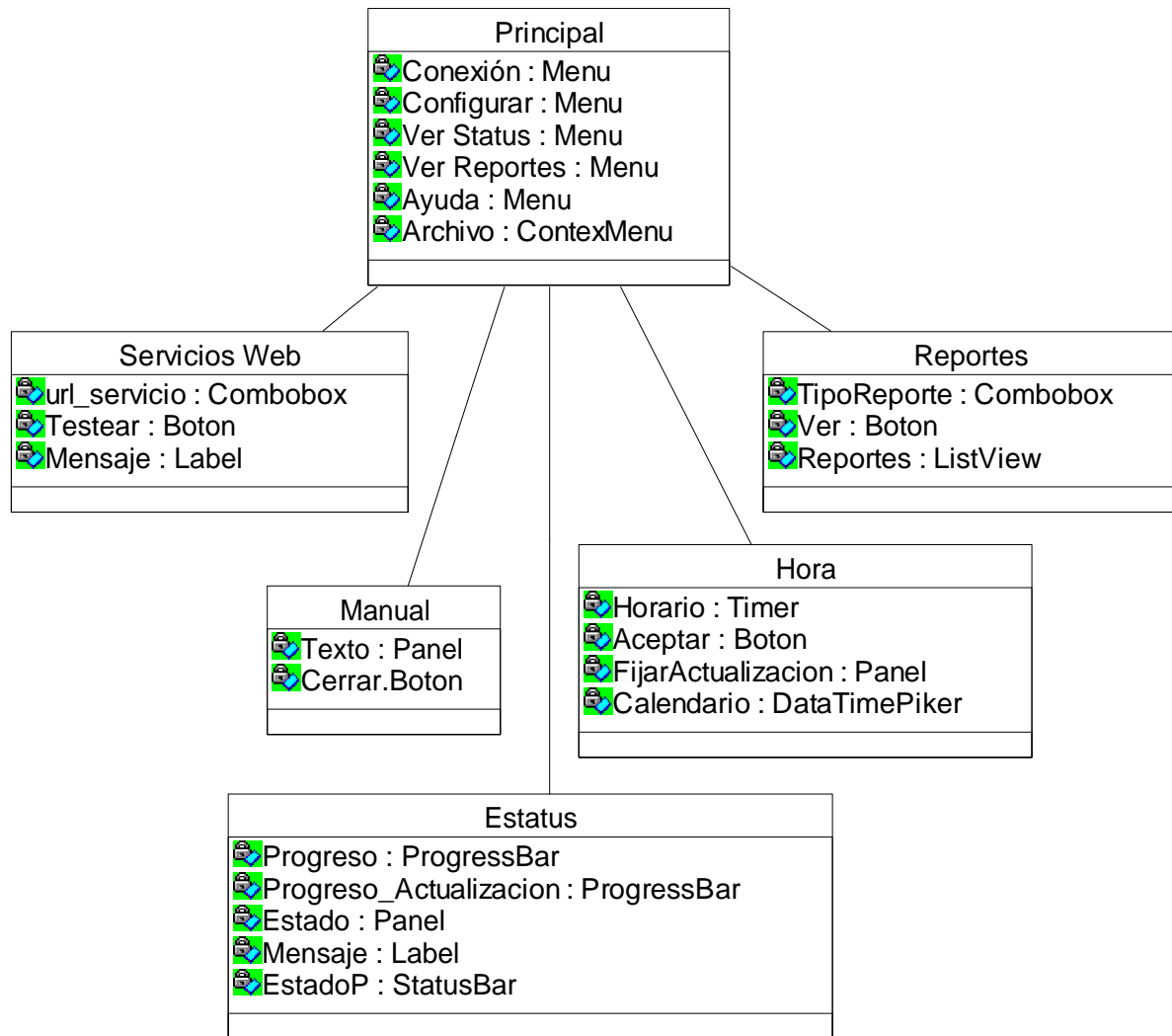


Figura 8. Diagrama Subpaquete Interfaz (Aplicación Desktop)

Paquete Lógica de Negocio

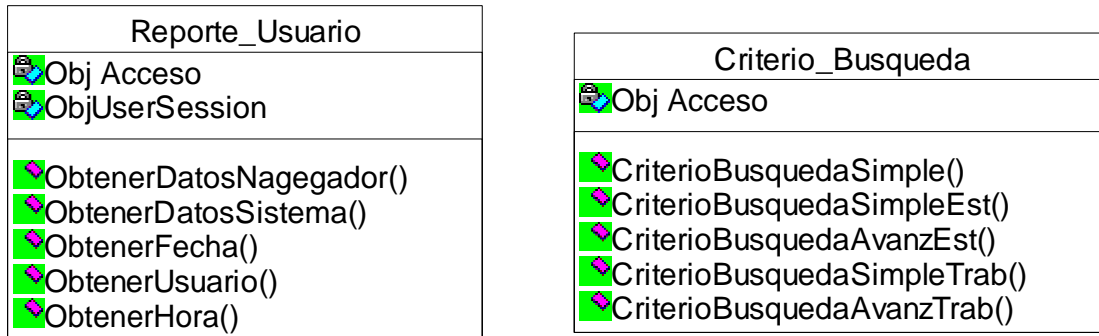


Figura 9. Diagrama Subpaquete Buscador

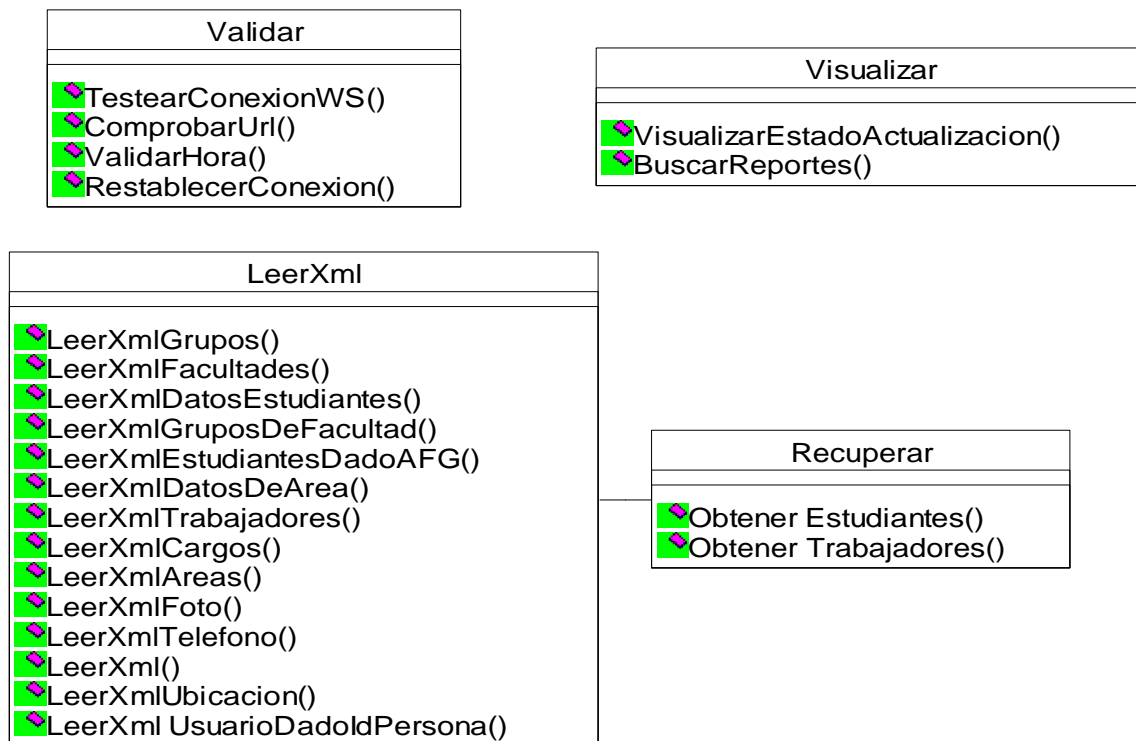


Figura 9. Diagrama Subpaquete Control. Ver Anexo 3

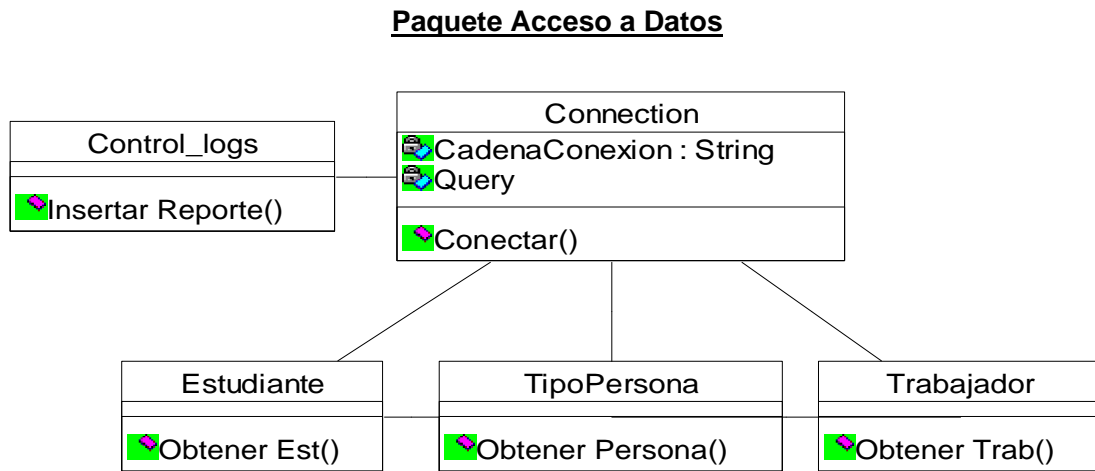


Figura10. Diagrama Subpaquete Acceso a Datos Web

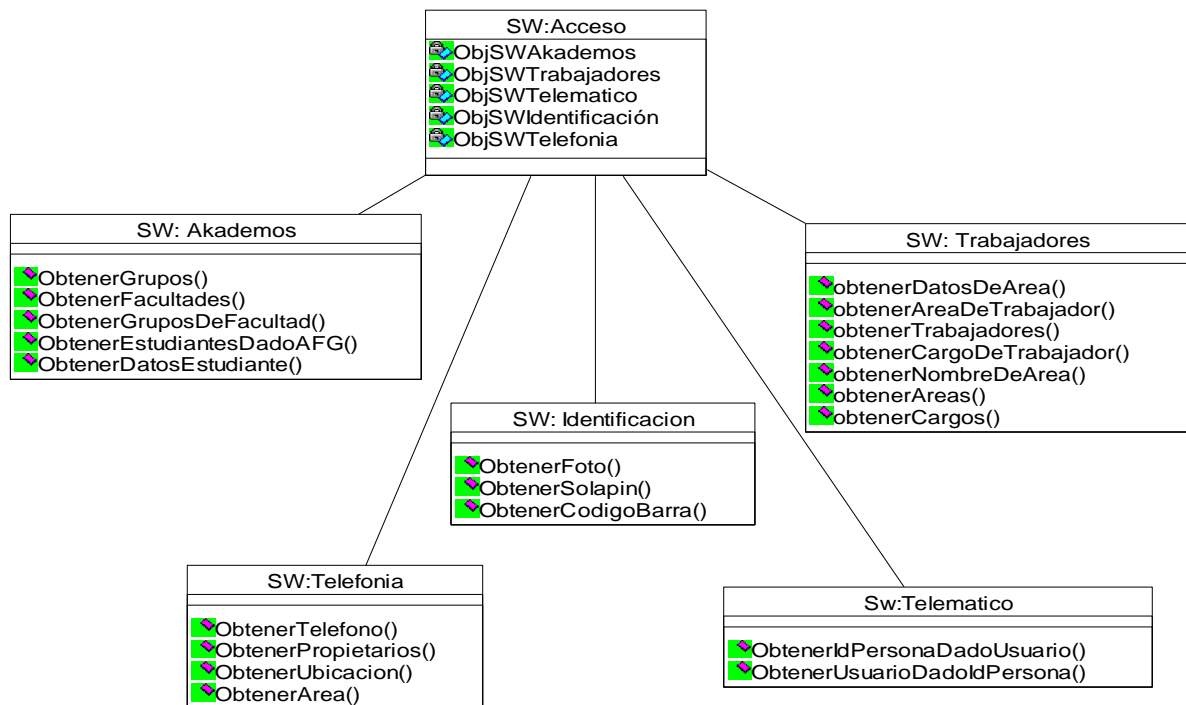


Figura11. Diagrama Subpaquete Servicios Web

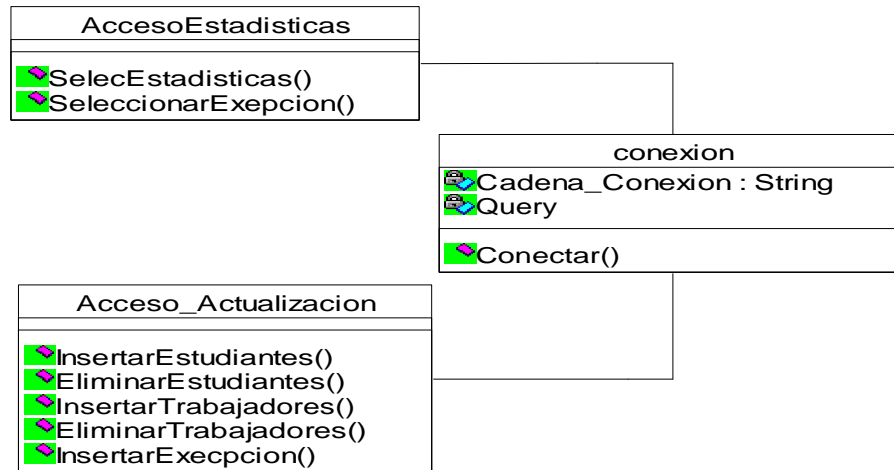


Figura12. Diagrama Subpaquete AccesoDatos (Aplicación Desktop)

Se publicará un servicio web que contará con métodos para acceder a la información contenida en la base de datos y deberá seguir una arquitectura de 3 capas.

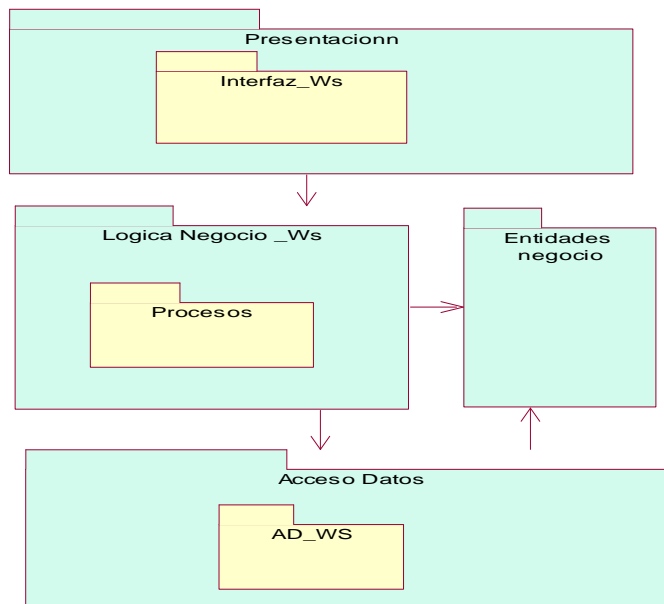


Figura13. Arquitectura del servicio web.

Paquete Presentación

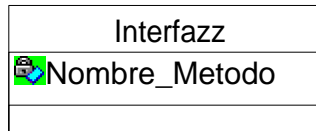


Figura14. Diagrama Subpaquete Interfazz

Paquete Lógica Negocio ws

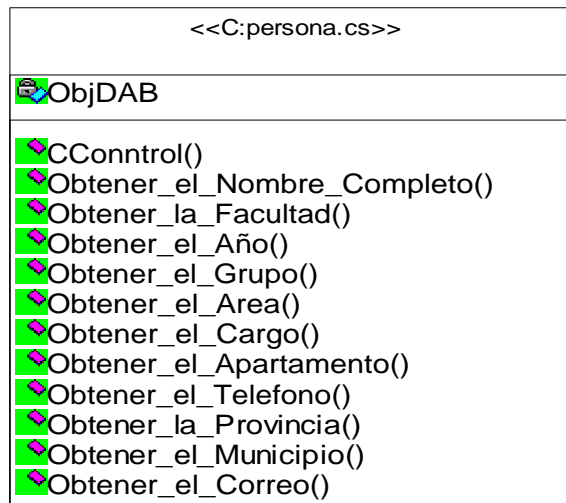


Figura15. Diagrama Subpaquete Procesos

Paquete Acceso Datos

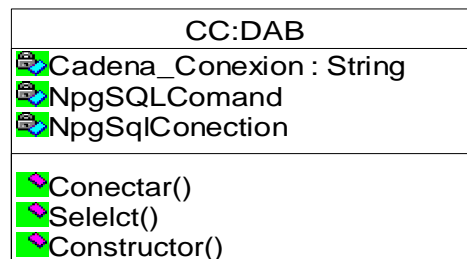


Figura16. Diagrama Subpaquete AD_WS

4.3 Diseño de la base de datos.

Para diseñar la base de datos del sistema, utilizamos el diagrama de clases persistentes y el modelo de datos. Los datos reflejados en las clases entidad son los que se capturan y almacenan durante el proceso que describe la aplicación, y estas a su vez son las que componen del diagrama de clases persistentes del cual se deriva el modelo de datos.

4.3.1 Diagrama de clases persistentes.

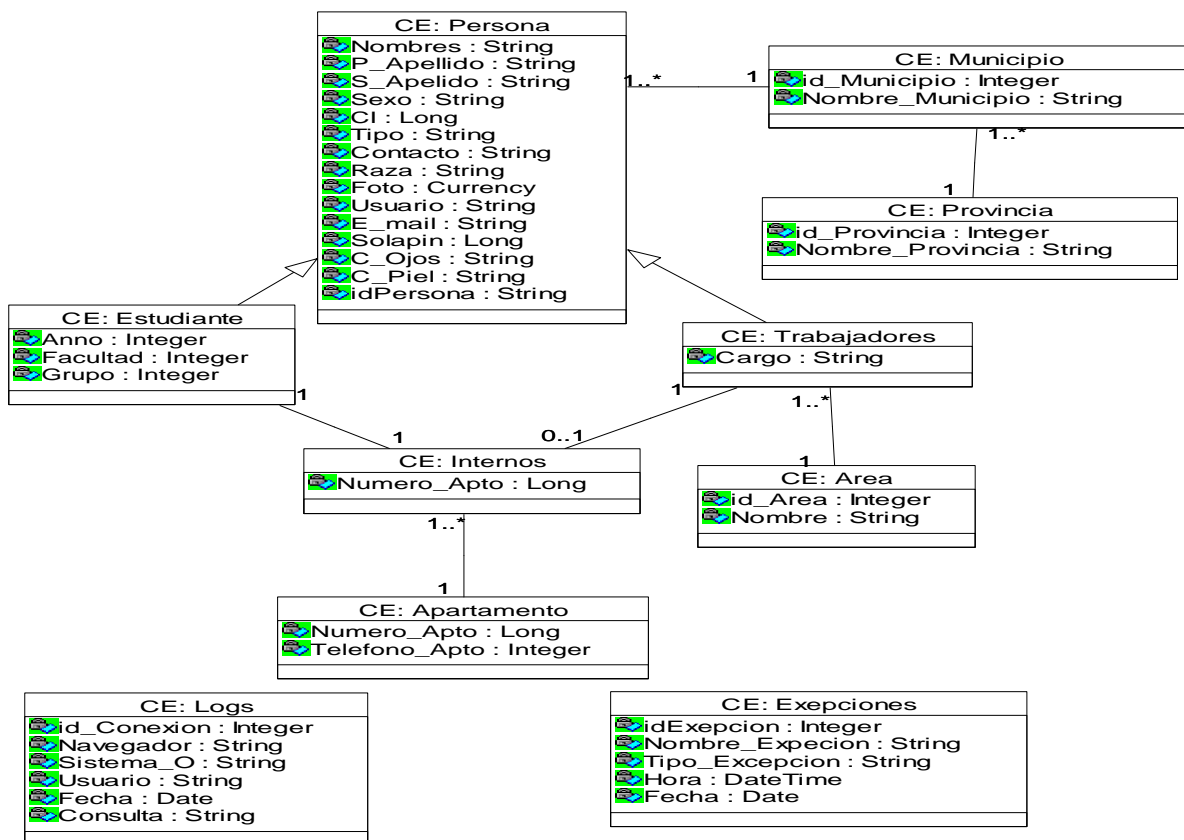


Figura13. Diagrama de clases persistentes.

4.3.2 Modelo de datos.

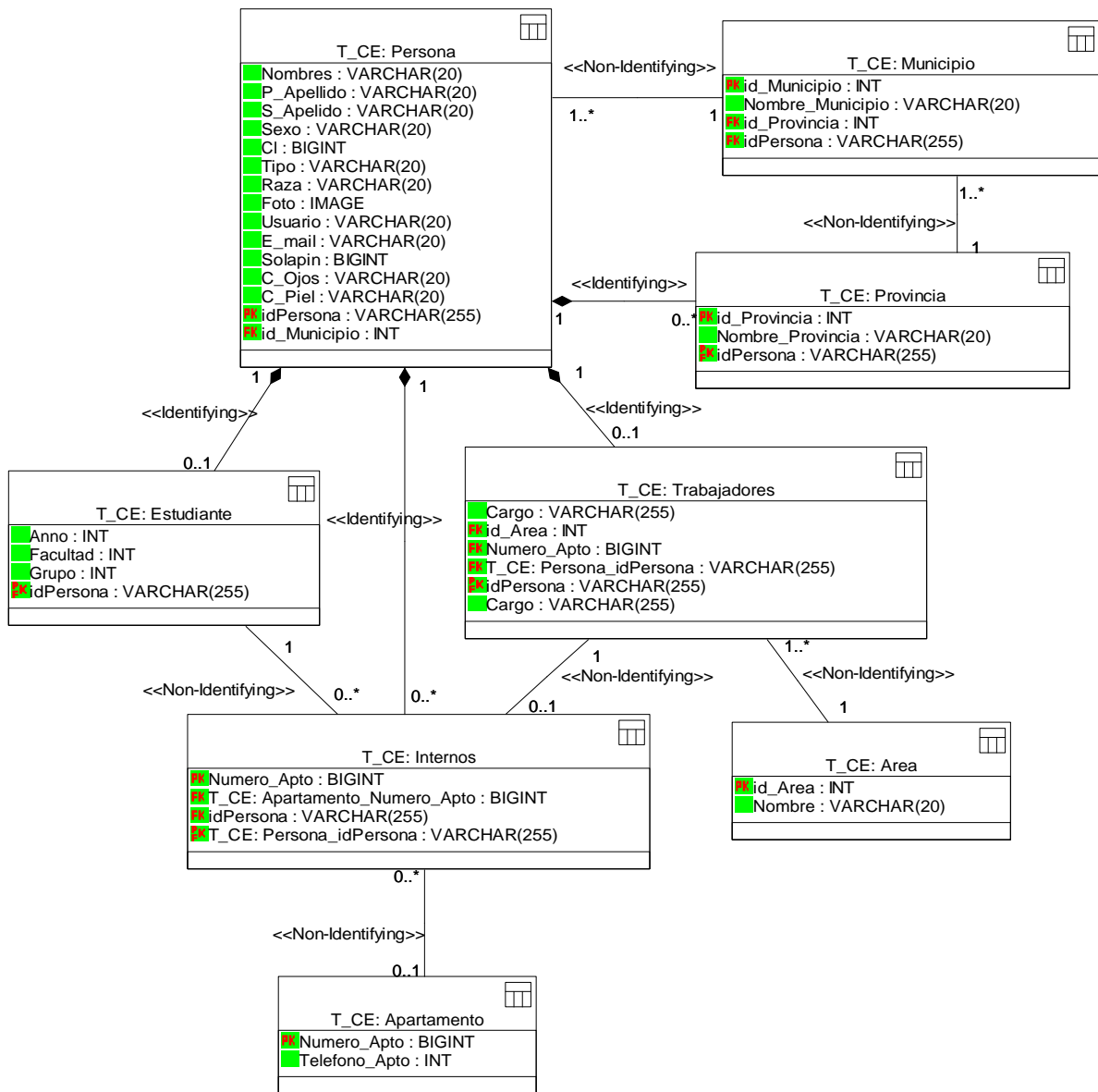


Figura14. Modelo de datos.

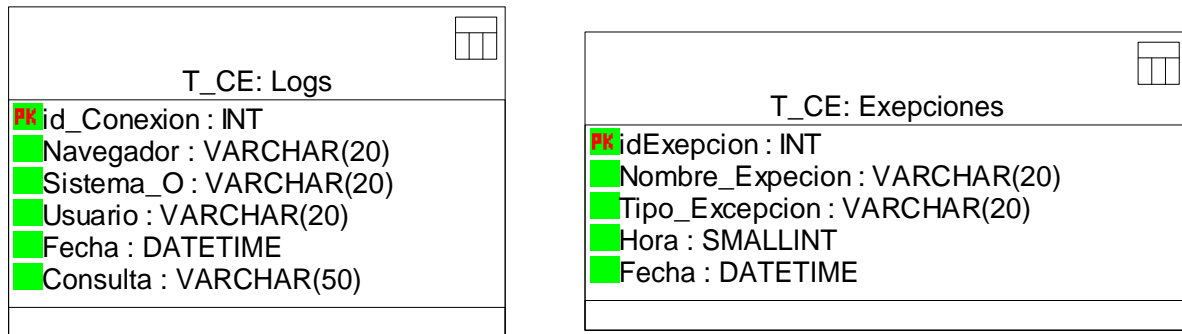


Figura 14. Modelo de datos. Cont.

4.4 Principios de diseño.

Para el diseño se propone que se tenga en cuenta para quien va dirigido el sistema, es decir los usuarios finales que serán: estudiantes, profesores y trabajadores, los cuales deben tener conocimientos de computación. La interfaz será amigable e intuitiva. Con un diseño consistente en todas las páginas, para lograr que el usuario se sienta cómodo y logre acostumbrarse rápidamente a la aplicación del Directorio.

El diseño de la aplicación de actualización será sencillo, aunque el usuario que interactuará con este será un administrador quien deberá tener bastos conocimientos de computación.

4.4.1 Estándares de la interfaz de aplicación.

En el diseño la forma en la que aplicamos el color es un aspecto de máxima importancia, a través de los colores somos capaces de comunicar sensaciones o sentimientos empleando muy poco espacio y muy poco tiempo.

Se propone para el Directorio de Persona que los colores a utilizar sean en su mayoría los de tonos claros, colores agradables y adaptables a la configuración de la intranet de la universidad, y que todas las

Capítulo 4. Construcción de la propuesta de solución

páginas tengan una estructura similar, con una imagen identificativa del sitio. Se debe ofrecer suficiente contraste entre texto y fondo para no dificultar la lectura.

4.4.2 Formato de reportes.

El sistema de actualización deberá brindar varios reportes. Un reporte de estadísticas detallado, en forma de tablas contendrá la cantidad de usuarios, tipo y cantidad de sistemas operativos, de navegadores, listado de usuarios conectados. También deberá mostrar una tabla con la fecha, el usuario y la consulta realizada. Los reportes de ejecución el administrador los necesitará saber constantemente, podrá ver el transcurso del proceso de actualización.

Los reportes de personas deberán permitir paginado, de forma que por cada búsqueda sean visibles solamente un número limitado de registros, permitiendo moverse adelante y hacia atrás, gracias a pequeñas flechas que irían en la parte inferior de las tablas indicando la dirección de las páginas.

4.4.3 Concepción de la ayuda.

El sistema contará con un manual de usuario que estará disponible en un menú de la aplicación de actualización, en dicho manual debe ir explicado el funcionamiento del sistema, requerimientos, y como trabajar con la aplicación. La aplicación web contará con una ayuda y en esta debe explicarse como realizar la búsqueda de las personas y conformar los criterios y patrones de búsqueda.

4.4.4 Tratamiento de excepciones.

Se debe garantizar mediante una validación que los datos suministrados por el administrador sean los correctos y se verificará los campos obligatorios y el tipo de datos de estos.

Las excepciones que podrían ocurrir durante el tiempo de ejecución de la aplicación, se manejarán generándose un registro en la base de datos del sistema, para la revisión por parte del administrador.

4.5 Estándares de la codificación.

Los estándares de codificación son reglas específicas a una lengua que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran. Durante el desarrollo, los estándares de codificación ayudan a los ingenieros a producir un código de alta calidad y a entender y a utilizar el código de sus colegas. Pero también realzan considerablemente la capacidad de mantenimiento y reutilización a largo plazo del producto final.

Utilizar un estándar para abrir código nos aporta varias ventajas, entre ellas están:

1. Reducir errores
2. Escribir un código comprensible y fácil de leer
3. Garantizar una buena comunicación entre los programadores del equipo
4. Facilitar el mantenimiento del software
5. Aumentar la robustez y la confiabilidad

Para esta aplicación se propone que se utilice el estándar de codificación “Camel Case” que se basa en la capitalización de los caracteres. Así mismo, se propone seguir el estilo de codificación propuesto por Microsoft para programar con C#. Se debe tener en cuenta al nombrar clases, variables, y demás elementos, precediendo cada nombre con un prefijo para su fácil identificación.

Para el diseño de la base de datos, se propone que su nombre siga la siguiente regla: anteponer las siglas UCI al nombre de la base de datos. Ejemplo: UCI_Directorio. Para las tablas que sigan la siguiente regla de nombre: Tb_<Nombre>.

4.6 Modelo de despliegue.

El modelo de despliegue contiene los nodos que conforman la topología de hardware sobre la que se ejecuta el sistema.

Para la construcción del sistema se propone que siguiendo la arquitectura de tres capas, o sea, la separación de los componentes en Lógica de Negocio, Acceso a Datos, Cliente se tome la estructura que a continuación describimos.

Un servidor de base de datos PostgreSQL 8.2 que contenga la base de datos. El servidor Web y el servidor de aplicación residirán en servidores separados, este último debe ser potente y debe tener instalada la plataforma .NET 2.0.

El nodo Servidor WS contiene los proveedores de los servicios Web a través de los cuales la aplicación de actualización extraerá la información.

Los clientes son los distintos usuarios desde sus terminales, accediendo al sistema mediante un navegador.

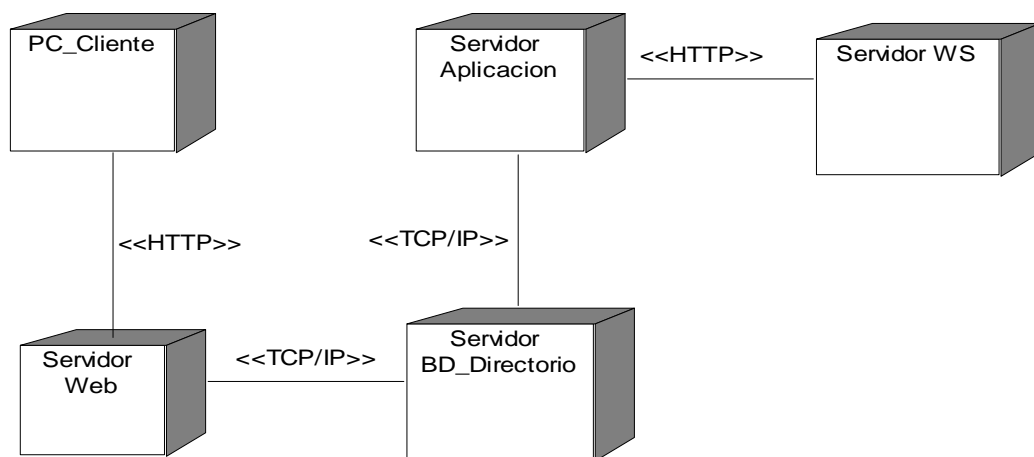


Figura 15. Modelo de despliegue

4.7 Conclusiones.

En este capítulo se ha llevado a cabo la descripción de las clases y demás elementos necesarios para la futura implementación. Se obtuvo el diagrama de clases del sistema, separado por paquetes. Se definieron, cuáles son las clases persistentes, luego, a partir de esto, se construyó el modelo de datos. Se expusieron las pautas seguidas para el diseño de la interfaz, y se explicó cómo quedará estructurada la aplicación físicamente, mediante el modelo de despliegue.

CAPITULO 5. ESTUDIO DE FACTIBILIDAD

5.1 Introducción.

En este capítulo hacemos un análisis del costo y beneficios que tendría la realización del proyecto utilizando la planificación basada en casos de uso con el objetivo de saber si la realización del mismo es factible o no.

5.2 Planificación basada en casos de usos.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	2	6
UAW = S(Factor * Actores)			UAW	6

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	4	20
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	5	50
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
UUCW = Sumatoria(Factor * CantCU)			UUCW	70
UUCP = UAW + UUCW			UUCP	76

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	3	3

Capítulo 5. Estudio de factibilidad

T4	Funcionamiento Interno complejo	1	4	4
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0,5	5	2,5
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	1	2
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	2	2
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	0	0
			Sumatoria	29
$TCF = 0.6 + 0.01 * \text{Sumatoria(Peso * Valor)}$			TCF	0,89

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	4	6
E2	Experiencia en la aplicación	0,5	3	1,5
E3	Experiencia en la orientación a objetos.	1	3	3
E4	Capacidad del analista líder.	0,5	4	2
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part-Time	-1	4	-4
E8	Dificultad del lenguaje de programación	-1	3	-3
			Sumatoria	17,5
$EF = 1.4 - 0.03 * \text{Sumatoria(Peso * Valor)}$			EF	0,875
Factor de conversión			CF	20
$UCP = UUCP * TCF * EF$			UCP	59,19
$E = UCP * CF$			E	1184

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	295,9
Diseño	20	591,9
Implementación	40	1184
Pruebas	15	443,9

Sobrecarga (otras actividades)	15	443,9
Total	100	2959

Esfuerzo Total (Horas--Hombre)	ET1	2959
Esfuerzo Total (Mes--Hombre)	ET2	20,55
Salario Promedio	SM	100
Cantidad de Hombres	CH	2
Costo Hombre--Mes	CHM	200
Costo Total	Costo	2055

5.3 Beneficios tangibles e intangibles.

5.3.1 Beneficios tangibles.

Partiendo de que la aplicación no es elaborada con el objetivo de ser comercializada sino para resolver un problema existente en el Directorio de Personas de la Universidad de las Ciencias Informáticas (UCI), no se puede afirmar que existen beneficios económicos, sociales, políticos ni resultados palpables y cuantificables.

5.3.2 Beneficios intangibles.

Se facilitan todos los procesos relacionados con la búsqueda y uso de los datos de las personas en la universidad, los que estarán mejor organizados y controlados. Esto en gran medida ayuda a prestar un mejor servicio de búsqueda de personas, logrando la satisfacción total de los usuarios del centro.

5.4 Análisis de costos y beneficios.

El rediseño del Directorio de Personas existente no constituye un gasto considerable a la universidad ya que se desarrolla con herramientas de programación poco costosas, además de que el centro posee la infraestructura informática necesaria para la ejecución de la aplicación. El uso de dichas herramientas disminuye los gastos implicados en el proceso de desarrollo de software. Con la implantación de este

sistema se tendrá un mejor dominio de la información, logrando de esta manera prestar un mejor servicio a la comunidad universitaria.

5.5 Conclusiones.

Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de aproximadamente nueve meses para su construcción por dos hombres y su costo asciende a 2055 pesos. El sistema propuesto trae consigo una serie de beneficios para la organización, ya que el mismo va a contribuir a mejorar sus servicios de búsqueda de información de personas, lo que indica que es factible implementar el sistema propuesto.

CONCLUSIONES

Con este trabajo se propone la solución al problema de la gestión de información de las personas en la Universidad de las Ciencias Informáticas. Para ello se hace un estudio detallado de las tecnologías existentes para así seleccionar cual es más conveniente utilizar en la implementación del sistema propuesto.

La modelación del sistema propuesto cumple con todos los requerimientos planteados por el cliente, lo que nos permite arribar a las siguientes conclusiones:

- La aplicación modelada facilita la gestión de la información de las personas en la Universidad de las Ciencias Informáticas.
- Con el diseño planteado se reduce el tiempo de respuesta a las necesidades del usuario.
- Se ha diseñado correctamente una base de datos para el Directorio de Personas propuesto, lo que permitirá que los sistemas de la UCI sean más independientes.
- El diseño de una aplicación para la actualización de la información de las personas en la universidad facilitará que el sistema de búsqueda de personas propuesto brinde un mejor servicio.

Por todo lo anteriormente planteado se concluye que el objetivo de este trabajo se ha cumplido satisfactoriamente.

RECOMENDACIONES

El desarrollo de este trabajo ha cumplido con el objetivo trazado, pero en el transcurso de la modelación del sistema han ido surgiendo ideas que serán muy útiles para el desarrollo de aplicaciones futuras en la universidad. Por ello se recomienda:

- Migrar la arquitectura del sistema propuesto hacia plataformas de software libres, para lograr soluciones de menor costo económico desde el punto de vista funcional para los desarrolladores de software en la universidad.
- Integrar sistemas de búsquedas como este a otras instituciones del país.
- Continuar investigando sobre sistemas de búsquedas basados en directorios.
- Agregar nuevas funcionalidades al directorio propuesto, tales como mapas de ubicación de la zona de residencia y áreas de trabajo.
- Agregar nuevas funcionalidades a la aplicación de actualización de las personas tales como mostrar las estadísticas a través de gráficos.

REFERENCIAS BIBLIOGRÁFICAS

- 1-AGRICULTURA, D. D. Zonificación agro-ecológica: Guía general..., 1997. [Disponible en: <http://www.fao.org/docrep/W2962S/w2962s0l.htm>]
- 2-ALVAREZ, M. A. Qué es ASP.
- . Qué es CGI.
- . Qué es PHP.
- 3-ANDRÉS, M. M. M. Metodología de diseño de bases de datos, 2001.
- 4-FERGUSON, J.; B. PATERSON, et al. La Biblia de C#. Anaya Multimedia, 2003. p.
- 5-Glosario de Definiciones. Disponible en: www.definicion.org/aplicacion
- 6-GUERRERO, L. A. Modelando aplicaciones Web con UML, 2005. [Disponible en: www.dcc.uchile.cl]
- 7-IGLESIAS, F. J. S. Glosario de Ciencia Ficción, 1999. [Disponible en: <http://www.ciencia-ficcion.com/glosario/index.html>]
- 8-MARKETING, I. P. D. GLOSARIO DEL NAVEGANTE, 2006. [Disponible en: <http://www.ipm.com.pe/glosarionave.htm>]
- 9-NEIRA, M. C. METODOS DE OPTIMIZACION DE CONSULTAS PARA EL LENGUAJE SQL. SANTIAGO - CHILE, UNIVERSIDAD DE SANTIAGO DE CHILE, 2002. p.
- 10-PROAÑO, D. J. B. ANALISIS COMPARATIVO DE BASES DE DATOS DE CODIGO ABIERTO VS CODIGO CERRADO (DETERMINACION DE INDICES DE COMPARACION), 2006.
- 11-PULIDO, J. R. V. 2000 MOTORES VERSUS DIRECTORIOS: EL DILEMA DE LA BÚSQUEDA: 5.
- 12-QUIROGA, L. A. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones, 2002.
- 13-ROMERO, Y. P. and Y. S. CORALES. Registro de Partos y Nacimientos para el Sistema Integral de Salud. Ciudad de la Habana, INSTITUTO SUPERIOR POLITECNICO "JOSE ANTONIO ECHEVERRIA", 2005. p.
- 14-ROZIC, S. E. Bases de datos y su aplicación con SQL, 2004.

BIBLIOGRAFÍA

- 1- ACOSTA, M. Á. M. and J. V. SANTIUSTE. InfoWeb: Sistema de Gestión de Información del Ciudadano
- 2- Modulo:(Buscador de Personas). Ciudad de la Habana, Universidad de las Ciencias Informaticas, 2006. p.
- 3- AGRICULTURA, D. D. Zonificación agro-ecológica: Guía general... , 1997. [Disponible en: <http://www.fao.org/docrep/W2962S/w2962s0l.htm>
- 4- ALVAREZ, M. A. Qué es ASP.
---. Qué es CGI.
- 5- ANDRÉS, M. M. M. Metodología de diseño de bases de datos, 2001.
- 6- BOOCH, G., JACOBSON, I. Y RUMBAUGH, J. The UML specification documents. 1997. p.
- 7- BRANDENBAUGH, J. Programación de aplicaciones Web con Java Script Madrid, Anaya Multimedia, 2000. 532 p.
- 8- BRUN, R. E. XML y Servicios Web, 2002. p. C., B. G. XML: el lenguaje de los Servicios Web.
- 9- FERGUSON, J.; B. PATERSON, et al. La Biblia de C#. Anaya Multimedia, 2003. p.
- 10- Glosario de Definiciones. Disponible en: www.definicion.org/aplicacion
- 11- GRADY BOOCH, I. J. A. J. R. El proceso Unificado de Desarrollo de Software. Editorial. Félix Varela. La Habana, 2004. pag 507 p. I-S-B-M
- 12- GUERRERO, L. A. Modelando aplicaciones Web con UML, 2005. [Disponible en: www.dcc.uchile.cl
- 13- HANSEN, G. W. and J. V. HANSEN. Diseño y Administracion de Bases de Datos. 2da Edicion Prentice Hall, p.
- 14- IGLESIAS, F. J. S. Glosario de Ciencia Ficción, 1999. [Disponible en: <http://www.ciencia-ficcion.com/glosario/index.html>
- 15- INFORMACION, S. D. L. Preguntas más frecuentes sobre la Sociedad de la Información, 2007.
- 16- KORTH, H. F. and A. SILVERSCHATZ. Fundamentos de bases de datos. . 3ª Ed. Mc Graw-Hill, p.
- 17- KROENKE. Procesamiento de Bases de Datos 5ª Ed. Ed. Prentice Hall, p.
- 18- LARMAN, C. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Félix Varela. Habana, 2004, 2004. 507 p.
- 19- MANUEL-PERALTA. Sistema de información. Monografía.com, 1997.

- 20- MARKETING, I. P. D. GLOSARIO DEL NAVEGANTE, 2006. [Disponible en: <http://www.ipm.com.pe/glosarionave.htm>]
- 21- MICROSOFT, E. Base de datos. CORPORATION., M., Reservados todos los derechos, 1993-2006.
- 22- MSDN. Documentación del Visual Studio 2005. Disponible en:
- 23- NODARSE, F. A. F. La Red Cubana de Ciencia desde una perspectiva de su integración y componentes, 2005. [Disponible en: http://www.informaticahabana.com/evento_virtual/files/MUL001.pdf]
- 24- PERALTA, M. Estimación del esfuerzo basada en casos de uso. Disponible en:
- 25- PRESSMAN, R. S. Ingeniería del Software. Un enfoque Práctico. Quinta Edición. Habana, 2002, 2005. pág 601 p.
- 26- PULIDO, J. R. V. Estrategias de búsqueda en Internet.
- . Los Motores de Búsqueda en Internet: Un atajo a la información.
- Motores de búsqueda: hacia una normalización terminológica.
- 27- QUIROGA, L. A. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones, 2002.
- 28- ROMERO, Y. P. and Y. S. CORALES. Registro de Partos y Nacimientos para el Sistema Integral de Salud. Ciudad de la Habana, INSTITUTO SUPERIOR POLITECNICO "JOSE ANTONIO ECHEVERRIA", 2005. p.
- 29- ROZIC, S. E. Bases de datos y su aplicación con SQL, 2004.
- 30- SUÁREZ, R. I. Araña del Buscador Cuba Search. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2006. 104. p.
- 31- TORRE, A. D. L. Lenguajes del lado servidor o cliente, 2006.

GLOSARIO DE TÉRMINOS

Camel Case: Estilo de escritura de código, se basa en capitalizar las iniciales de las todas las palabras que forman el nombre de un método o variables, menos la primera.

COMMIT: En el contexto de informática y administración de los datos, COMMIT se refiere a la idea de hacer permanente un juego de cambios provisionales, como al final de una transacción. Una operación COMMIT es (comprometer o confirmar).

EF: Factor de ambiente.

HTTP: El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares.

NET Framework SDK: Conjunto de herramientas.

ROLLBACK: Un término usado en un proceso de transacción que cancela una transacción propuesta modificando uno o más registros y deshace los cambios. Es una operación que devuelve la base de datos a algún estado anterior. Son importantes para la integridad de la base de datos. La base de datos puede restaurarse a una copia limpia después de que se realizan operaciones erróneas, restaurándose a un estado consistente.

TCF: Factor de complejidad técnica.

TCP/IP: La familia de protocolos de Internet es un conjunto de protocolos de red que implementa la pila de protocolos en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se la denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia.

UAW: Factor de peso de los actores sin ajustar.

UCP: Puntos de casos de uso ajustados.

UML: Lenguaje Unificado para la Construcción de Modelos.

URL: Acrónimo de Universal Resource Locator (localizador universal de recursos), método de identificación de documentos o lugares en Internet, que se utiliza principalmente en World Wide Web (WWW). Un URL es una cadena de caracteres que identifica el tipo de documento, la computadora, el directorio y los subdirectorios en donde se encuentra el documento y su nombre.

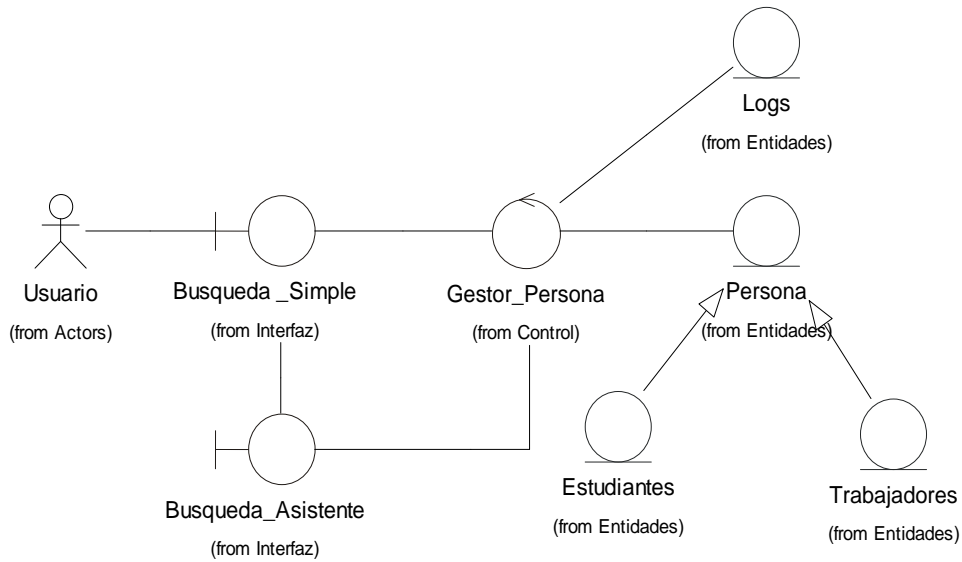
UUCP: Puntos de casos de uso sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar

XML: Lenguaje de marcas extensible.

ANEXOS

ANEXO 1. DIAGRAMA DE CLASES DEL ANÁLISIS



Dia

Diagrama de clases del análisis Paquete Buscar_Persona

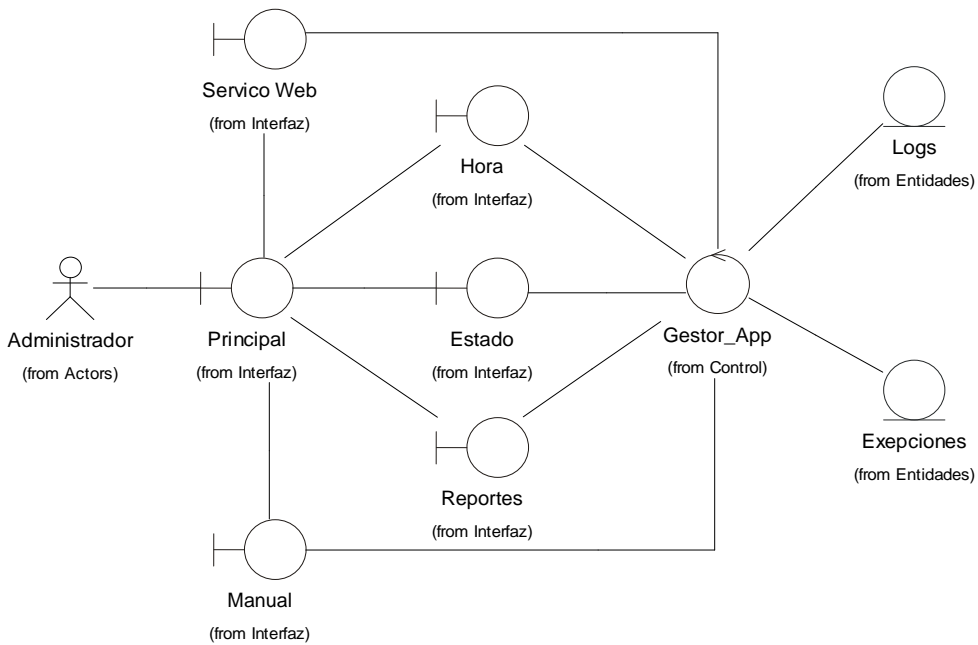


Diagrama de clases del análisis Paquete Administración

Anexo 1. Diagrama de clases del análisis

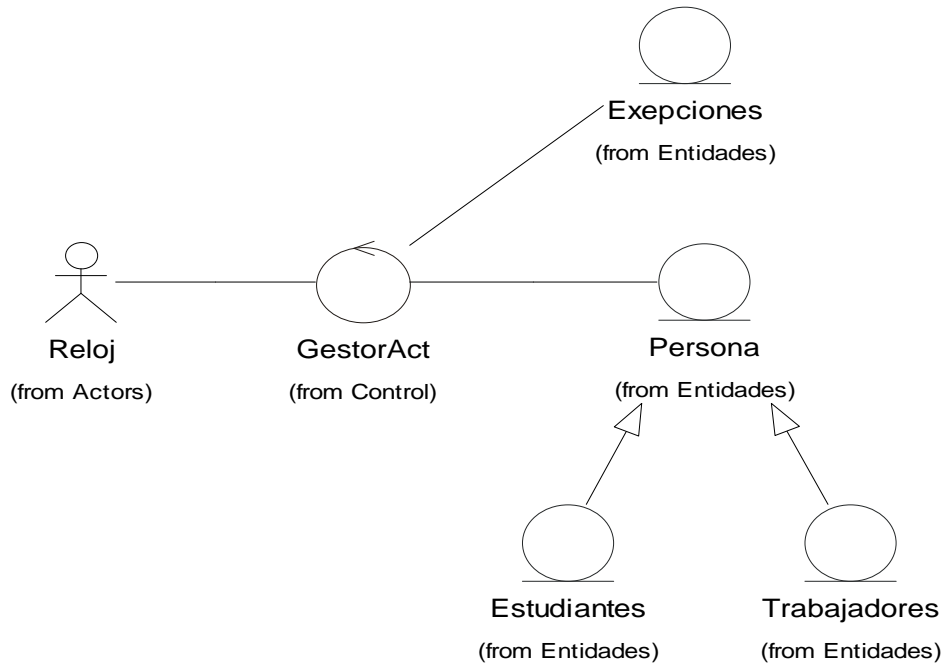







Diagrama de clases del análisis Paquete Actualización_Información_Personas

Anexo 2. Tabla de atributos y operaciones web

ANEXO 2. TABLA DE ATRIBUTOS Y OPERACIONES WEB.

Estereotipo web	Atributos	Operaciones
 SP_Inicio		UserSession()
 SP_GenerarBúsqueda	Datos_Busqueda ObjBusqSimple ObjBusqAsist ObjLogicaN CantPag	CrearObj BusqSimple CrearObj BusqAsist CrearObjLogicaN CrearPaginado
 FR_Busqueda Simple	Datos_busqueda: textbox Buscar: Button Tipo_Persona: RadioButon Ayuda: Label Utilizar_Asistente: Label	
 FR_Busqueda Asistente	Buscar: Button Nombre: textbox FiltraeEst_Trab: Combobox Provincia: Combobox Andprovincia: Combobox Municipio: Combobox AndMunicipio: Combobox Ayuda: Label Usuario: TextBox Sexo: Combobox Ojos: Combobox Area: Combobox Cargo: Combobox Raza: Combobox Facultad: Combobox Año: Combobox Grupo: Combobox	

Anexo 2. Tabla de atributos y operaciones web

 <p>CP_Resultado Busqueda</p>	Nombre: TextBox Buscar: Buton Tipo_Persona: RadioButton Ayuda: Label Asistente: Label Estudiante(Cant): Label Trabajador(Cant): Label Filtro: Combobox Imagen: Image VerDatalles: Label Paginador: Button NombreCompleto: Label Grupo: Label	
--	--	--

Anexo 3. Estructura de los XML

ANEXO 3. ESTRUCTURA DE LOS XML.

Método	ObtenerEstudiantesDadoAFG
Descripción	Obtiene todos los estudiantes dado Año, Facultad y Grupo. Se devuelve solamente el identificador de cada persona. (IdPersona). Puede especificarse solo el parámetro deseado, si se quiere a todos los estudiantes, se dejan los tres parámetros vacíos.
Parámetros de entrada	Año, Facultad, Grupo
Devuelve	XML con los identificadores de los estudiantes que cumplan con el criterio de búsqueda
Ejemplo	<pre><ArrayOfestudiante> <estudiante> <idPersona> 418d4c4b-d950-4e21-836c- 5e4213916723 </idPersona> </estudiante> <estudiante> <idPersona> 314d4c4b-B950-4e21-836c- 5e4213916725 </idPersona> </estudiante> </ArrayOfestudiante></pre>