

Universidad de las Ciencias Informáticas



InfoWeb: Sistema de Gestión de Información del Ciudadano
Modulo:(Buscador de Personas)



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERÍA INFORMÁTICA**

Autores

Miguel Ángel Martínez Acosta
Julián Valdés Santiuste

Tutor

Ing. Renier Pérez García

Ciudad de la Habana
Abril 2006

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ___ días del mes de abril del 2005.

Firma del Autor

Firma del Autor

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado: "InfoWeb: Sistema de Gestión de Información del Ciudadano. (Módulo: Buscador de Personas)", fue realizado en la Universidad de las Ciencias Informáticas, a petición del Departamento de Seguridad. Esta Dirección considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta Universidad los beneficios siguientes:

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____.

Y para que así conste, se firma la presente a los ____ días del mes de diciembre del año 2005.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: InfoWeb: Sistema de Gestión de Información del Ciudadano. (Módulo: Buscador de personas).

Autores: Miguel Ángel Martínez Acosta y Julián Valdés Santiuste..

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
 - Originalidad
 - Creatividad
 - Laboriosidad
 - Responsabilidad >

< Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que las estudiantes están aptas para ejercer como Ingenieras Informáticas y propongo que se le otorgue al Trabajo de Diploma la calificación de ____ puntos. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

Firma

Fecha

***“La experiencia más bella que tenemos a nuestro alcance es el misterio.
Esta es la emoción sustancial que se halla en la cuna del auténtico arte
y de la verdadera ciencia”.***

Albert Einstein.

AGRADECIMIENTOS

A Fabio y a Perdomo, por creer siempre en nosotros.

A todos nuestros compañeros y compañeras de aula (y de año) por compartir con nosotros todos estos años.

A todos nuestros amigos

A todos ellos, gracias.

DEDICATORIA

A toda mi familia.

A mis padres, por apoyarme y creer en mi todos estos años.

A mi hermana

A mis abuelos y en especial a mi abuelo Francisco, sin el no hubiera llegado hasta aquí.

A todos mis amigos, los que están y los que ya no se encuentran aquí.

Miguel Ángel Martínez Acosta

A mis padres por confiar en mi todos estos años.

A mi hermana por los ratos juntos.

A mi abuela por toda la paciencia que ha tenido.

Muy en particular a mi bisabuela.

Julián Valdés Santiuste.

RESUMEN

Este trabajo se centra en las herramientas de búsqueda, dentro de ellas las especializadas en buscar información de la persona, las que permiten acceder a datos de valor tanto para personas como para entidades de todo tipo. Actualmente en la Universidad de las Ciencias Informáticas el acceso a la información del ciudadano se ha convertido en uno de los servicios de vital importancia dentro de la comunidad universitaria, puesto que dicha información es utilizada en múltiples tareas dentro de las que se encuentran las relacionadas con el control de la calidad del funcionamiento del centro; aunque por la ausencia de algunas funcionalidades este servicio no satisface totalmente a los usuarios.

Luego se puede plantear que de manera general **el objetivo** de este trabajo es la creación de una aplicación Web que permita la recuperación de la información de la persona, auxiliándose de varias bases de datos ya existentes en la universidad y mediante el uso de avanzadas técnicas de búsqueda de la que se va a implementar solamente el módulo de búsqueda.

En este documento se plasman los resultados de la investigación realizada; se incluyen conceptos vinculados con los sistemas de búsqueda, se realiza un análisis de las tecnologías más usadas para posteriormente seleccionar la apropiada; finalmente, se muestran los resultados del análisis y diseño de la propuesta del sistema y se dejan algunas recomendaciones para el desarrollo futuro del mismo.

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA.....	5
1.1 INTRODUCCIÓN.....	5
1.2 HERRAMIENTAS DE BÚSQUEDA.....	5
1.2.1 <i>Funcionamiento de los buscadores automáticos o motores de búsqueda.....</i>	<i>5</i>
1.2.2 <i>Buscadores basados en directorios y catálogos.....</i>	<i>8</i>
1.2.3 <i>Buscadores híbridos.....</i>	<i>9</i>
1.2.4 <i>Buscadores especializados.....</i>	<i>10</i>
1.3 RECUPERACIÓN DE DATOS.....	12
1.3.1 <i>Recuperación datos de contra recuperación de información.....</i>	<i>12</i>
1.4 BASES DE DATOS.....	14
1.4.1 <i>Base de datos distribuida.....</i>	<i>15</i>
1.5 CAMPO DE ACCIÓN.....	18
1.5.1 <i>Descripción actual de los procesos de negocio.....</i>	<i>19</i>
1.5.2 <i>Situación problemática.....</i>	<i>20</i>
1.6 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.....	21
1.6.1 <i>Sistema de gestión académica: Akademos.....</i>	<i>21</i>
1.6.2 <i>Sistema de gestión de alojamiento.....</i>	<i>22</i>
1.6.3 <i>Sistema de gestión de información de los trabajadores.....</i>	<i>22</i>
1.7 PROPUESTA DE SOLUCIÓN.....	22
1.8 OBJETIVOS PROPUESTOS.....	23
1.8.1 <i>Objetivo general.....</i>	<i>23</i>
1.8.2 <i>Objetivos específicos.....</i>	<i>24</i>
1.9 CONCLUSIONES.....	24
CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR. ..	25
2.1 INTRODUCCIÓN.....	25
2.2 LA POLÍTICA DE MIGRACIÓN HACIA SOFTWARE LIBRE.....	25
2.3 APLICACIONES WEB.....	26
2.4 LA SEGURIDAD DE LAS TRASMISIONES EN LA WEB.....	27
2.4.1 <i>Protocolo SSL.....</i>	<i>27</i>
2.4.2 <i>OpenSSL.....</i>	<i>29</i>
2.5 SERVIDOR WEB.....	30
2.6 FUNDAMENTACIÓN DE LA METODOLOGÍA UTILIZADA.....	32
2.6.1 <i>El proceso Unificado de Rational: RUP.....</i>	<i>32</i>

2.6.2	<i>Programación Extrema: XP.</i>	33
2.6.3	<i>Desarrollo Guiado por la Funcionalidad: FDD.</i>	34
2.6.4	<i>Selección de la metodología a utilizar.</i>	35
2.7	Lenguajes de Programación para la Web.	36
2.7.1	<i>Lenguajes del lado del servidor.</i>	37
2.7.2	<i>Lenguajes del lado del cliente.</i>	39
2.7.3	<i>Selección del lenguaje a utilizar.</i>	40
	<i>Cascade Style Sheets: CSS.</i>	41
2.7.4	<i>Sistemas Gestores de Bases de Datos: SGBD.</i>	42
2.7.5	<i>Selección del SGBD a utilizar en la UCI.</i>	44
2.8	OTRAS HERRAMIENTAS.	45
2.9	NUESTRA PROPUESTA.	46
2.10	CONCLUSIONES.	46
CAPÍTULO 3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCION.		47
3.1	INTRODUCCIÓN.	47
3.2	MODELO DE NEGOCIO PROPUESTO.	47
3.2.1	<i>Actores del negocio.</i>	48
3.2.2	<i>Trabajadores del negocio.</i>	48
3.3	DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIO PROPUESTOS.	49
3.4	REQUISITOS FUNCIONALES.	54
3.5	REQUISITOS NO FUNCIONALES.	56
3.6	DESCRIPCIÓN DEL SISTEMA PROPUESTO.	57
3.7	MODELO DE CASOS DE USO DEL SISTEMA PROPUESTO.	58
3.7.1	<i>Actores del sistema.</i>	59
3.7.2	<i>Diagrama de casos uso del sistema.</i>	59
3.7.3	<i>Expansión de los casos de uso del sistema.</i>	61
3.8	CONCLUSIONES.	65
CAPÍTULO 4. CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN.		66
4.1	INTRODUCCIÓN.	66
4.2	MODELO DE DISEÑO.	66
4.2.1	<i>Diagrama de clases Web.</i>	66
4.2.2	<i>Diseño de base de datos.</i>	70
4.3	PATRONES DE DISEÑO.	72
4.3.1	<i>Estándares de la interfaz de la aplicación.</i>	73
4.3.2	<i>Formato del reporte de búsqueda.</i>	74
4.3.3	<i>Concepción general de la ayuda.</i>	75

4.3.4	<i>Tratamiento de excepciones.</i>	76
4.4	ESTÁNDARES DE CODIFICACIÓN.	77
4.5	MODELO DE DESPLIEGUE.	78
4.6	MODELO DE IMPLEMENTACIÓN.	80
4.7	DESCRIPCIÓN PRELIMINAR DEL MODELO DE PRUEBAS.	81
4.8	CONCLUSIONES.	81
CAPÍTULO 5. ESTUDIO DE FACTIBILIDAD.		82
5.1	INTRODUCCIÓN.	82
5.2	ESTIMACIÓN DE COSTO.	82
5.3	ANÁLISIS DE COSTO / BENEFICIO.	83
5.4	CONCLUSIONES.	83
CONCLUSIONES GENERALES		84
RECOMENDACIONES		86
BIBLIOGRAFÍA		87
GLOSARIO DE TÉRMINOS		88
ANEXOS		90

INTRODUCCIÓN

Con el desarrollo alcanzado, actualmente por las nuevas tecnologías de la informática y las comunicaciones, se han desarrollado nuevos mecanismos de almacenamiento y nuevas estrategias de búsqueda para proporcionar el acceso a los grandes volúmenes de datos que existen en formato digital. Entre las instituciones que más usan estas nuevas tendencias, para controlar su funcionamiento interno, se encuentran las universidades encabezando el gran grupo de centros que utilizan la información para el bien social.

Las universidades cubanas, dentro de ellas la Universidad de las Ciencias Informáticas (UCI) como centro universitario de nuevo tipo, proveen a su comunidad de una serie de servicios de información que hacen que funcione eficientemente desde la preparación de los estudiantes hasta la comunicación dentro de la institución. A través de estos se hace cumplir su misión como centro académico: contribuir a la formación de profesionales de alto calibre. Uno de los servicios más importantes dentro de una universidad es el de búsqueda de información, concretamente el que está relacionado con la recuperación de información de valor correspondiente a una persona determinada.

El acceso a la información de las personas dentro de la UCI es uno de los servicios considerados de primer orden dentro de este centro estudiantil. Actualmente existe una aplicación para la búsqueda de datos de las personas, el **Directorio de búsqueda**, la que permite a través de una herramienta para la búsqueda que los usuarios puedan ver datos básicos de una persona o de grupos de estas. Este sistema implementado sobre la plataforma .NET carece de algunas funcionalidades que hacen que su uso no sea completamente satisfactorio. Entre las deficiencias se encuentra por ejemplo; que no se realiza correctamente la búsqueda por nombre completos, no se realiza la paginación de los resultados, no se controla correctamente los espacios en las cadenas de búsqueda, etc. Todo lo antes mencionado propicia que se

obtengan a veces resultados no deseados como producto de la recuperación de información formulada a partir de una petición hecha por el usuario.

Partiendo de la situación anterior se puede decir que el **problema** a resolver queda formulado a modo de interrogante de la siguiente manera: ¿Como recuperar y mostrar la información del ciudadano que se encuentra diseminada en las distintas bases de datos de los sistemas informáticos de la UCI, que controlan información de la persona, de forma correcta y eficiente?

Conjuntamente con el desarrollo de las redes de computadoras e Internet se han implementado disímiles de aplicaciones cuyo propósito es la exposición, construcción y actualización de la información para presentarla a través de la red con algún formato determinado previamente. La recuperación de información es el conjunto de tareas mediante las cuales el usuario localiza y accede a los datos que son pertinentes para la resolución de un problema planteado.

La recuperación de información es una tarea que ocupa a todos en la actualidad; desde la aparición de Internet. Cada día se adicionan nuevas posibilidades a la red de redes, nuevos volúmenes de datos y nuevos servicios comienzan a operar. La gran cantidad de datos se convierte en el principal obstáculo para determinar la información deseada en el menor tiempo posible y una de las variantes más utilizadas para resolver este problema es la creación de herramientas de búsqueda.

Con este trabajo se propone la construcción del módulo de búsqueda de una herramienta para la búsqueda de personas con el fin de facilitar el acceso a la información del ciudadano, permitiendo asimismo que el sistema se convierta en una ayuda indispensable y útil para todos.

Por tanto, definimos como **objeto de estudio** la gestión de búsqueda de información dentro de la Universidad de las Ciencias Informaticas.

De ello se deriva que el **campo de acción** que abarca este trabajo es la gestión de búsqueda de información del ciudadano dentro de la Universidad para ponerla a disposición de los estudiantes, trabajadores y personas en general de la Universidad de las Ciencias Informáticas.

El **objetivo general** es: desarrollar el módulo de búsqueda como una Aplicación Web que permita la búsqueda y recuperación de información del ciudadano en la Universidad de las Ciencias Informáticas mediante avanzadas técnicas de gestión de búsqueda y auxiliándose de las base de datos de los sistemas que gestionan la información del ciudadano en la universidad.

Partiendo de lo planteado anteriormente los **objetivos específicos** son:

- Realizar un estudio sobre las formas y mecanismos de búsqueda de manera general.
- Realizar un estudio sobre la situación de la información de la persona en la universidad.
- Diseñar la propuesta de solución utilizando una metodología de desarrollo de software de las actuales.
- Diseñar una interfaz amigable para la propuesta de solución en la que el usuario con conocimientos mínimos de informática pueda navegar sin problema.
- Implementar un mecanismo para la recuperación de datos de las bases de datos ya existentes en la universidad.

En aras de cumplir los objetivos y solucionar la situación problemática planteada se realizaran las siguientes **tareas**:

- Estudio y descripción de los sistemas de búsqueda existentes en la Intranet de la Universidad actualmente.
- Selección de la metodología de Análisis y Diseño de sistemas informáticos, que facilite la creación y garantice la calidad del sistema.

- Selección de las herramientas para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrollará la aplicación. Fundamentando la elección.
- Implementación del módulo de búsqueda de la nueva aplicación Web que gestionará la búsqueda de información de la persona en la Universidad.

El presente documento consta de cinco capítulos:

El **Capítulo 1** describe algunas de las características de las herramientas de búsqueda y el proceso de búsqueda actual, lo que hace también necesario una referencia a los sistemas automatizados que existen en la universidad y que están vinculados con nuestro trabajo; se mencionan los principales problemas que generaron la necesidad del cambio; y finalmente, se obtienen los objetivos generales y específicos a cumplir por el trabajo.

En el **Capítulo 2** se trata la situación de las tecnologías a utilizar en el desarrollo de la propuesta, se comparan y seleccionan las mejores propuestas para el trabajo, y se explican los conceptos principales que se van a tratar a lo largo del trabajo.

El **Capítulo 3** describe la situación a través de un modelo de Negocio, y a partir de esto se comienza a hacer el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente utilizando herramientas de modelación.

En el **Capítulo 4** se describen como será construida la solución que se ha propuesto, se modelan los diagramas de clases Web y se especifican los principios para el diseño gráfico y la implementación.

El **Capítulo 5** 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 DEL TEMA.

1.1 Introducción.

En este capítulo se expondrán de manera general los aspectos relacionados con los sistemas de búsqueda, los conceptos necesarios para el estudio y clasificación de los mismos; y algunas características de cada tipo de herramienta.

Además se describen claramente los procesos del negocio que se relacionan con el campo de acción de este trabajo. Se identifican los principales problemas que fundamentan la propuesta de solución, y se marcan los objetivos generales y específicos.

1.2 Herramientas de búsqueda.

Las herramientas de búsqueda, también llamadas buscadores, son sistemas que hacen fácil para los usuarios la búsqueda de información a través de la red, permitiendo el acceso a información almacenada en una o varias bases de datos. Actualmente las más difundidas son la que realizan búsquedas en la Web que pueden ser: buscadores automáticos, buscadores basados en directorios y catálogos o buscadores híbridos. Cualquiera de ellos puede estar especializado en la búsqueda dentro de un campo determinado por ejemplo recursos o personas.

1.2.1 Funcionamiento de los buscadores automáticos o motores de búsqueda.

Los buscadores automáticos son herramienta de búsqueda en Internet que permite buscar información a partir de palabras claves que identifican el contenido de la pregunta a realizar. Ejemplos de este tipo de buscador son Altavista, HotBot, Lycos.

El funcionamiento de un motor debe estudiarse desde dos perspectivas complementarias: la recopilación y la recuperación de información. Un motor compila de forma automática las direcciones de las páginas que van a formar parte de su índice tras realizar sobre su contenido un proceso de indización. Una vez que se encuentren estos registros debidamente depositados en la base de datos del motor de búsqueda, los usuarios buscarán en su índice por medio de una interfaz de consulta, que puede ser más o menos avanzada en función del grado de desarrollo del sistema. Al módulo encargado de la recopilación de las páginas se le conoce comúnmente como robot, es un programa que rastrea la estructura hipertextual de la Web, recogiendo información sobre las páginas que encuentra. Esa información se indiza y se introduce en una base de datos que será explorada posteriormente utilizando un motor de búsqueda.

Esos robots pueden recopilar varios millones de páginas por día, y actualizar la información recopilada en los índices en períodos de tiempo extremadamente pequeños. Por regla general, se parte de una lista inicial de direcciones de sitios Web, que son visitadas por el robot, y a partir de ahí cada robot rastrea a su manera la Web, de ahí que la información almacenada en cada base de datos de cada motor sea diferente.

Arquitectura.

La mayoría de los motores de búsqueda emplean una arquitectura de tipo robot-indexador centralizada, que se muestra en la figura 1.1. A pesar de lo que puede incluir su nombre de una amplia serie de definiciones incorrectas, el robot no se mueve por la red, ni se ejecuta sobre las máquinas remotas que visita, ya que realmente el robot funciona sobre el sistema local del motor de búsqueda y envía una serie de peticiones a los servidores Web remotos (donde se alojan las páginas a analizar). El índice también es gestionado localmente.

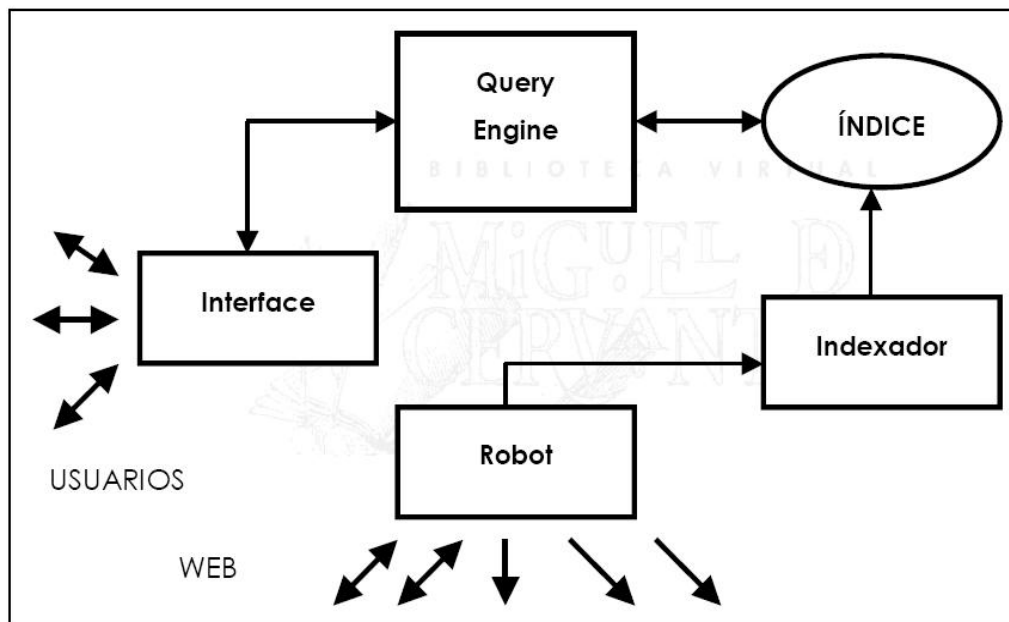


Fig. 1.1 arquitectura robot-indexador centralizada.

También existe la arquitectura distribuida para recopilar y distribuir los datos, que es más eficiente que la arquitectura centralizada y cuya estructura se muestra en la figura 1.2. El principal inconveniente es la necesidad de contar con varios servidores para implementarla. Esta arquitectura está formada por dos componentes principales: el recopilador y los brokers. El primero de ellos es el módulo que colecciona páginas y extrae de ellas toda la información necesaria para crear el índice del motor de búsqueda. El segundo de estos componentes, el broker, es el módulo encargado de proporcionar el mecanismo de indexación de las páginas recopiladas y la interfaz de consulta para los datos recopilados. Al mismo tiempo, los brokers son los servidores de búsquedas, recuperan información desde uno o varios recopiladores o desde uno o varios brokers, actualizando constantemente sus índices.

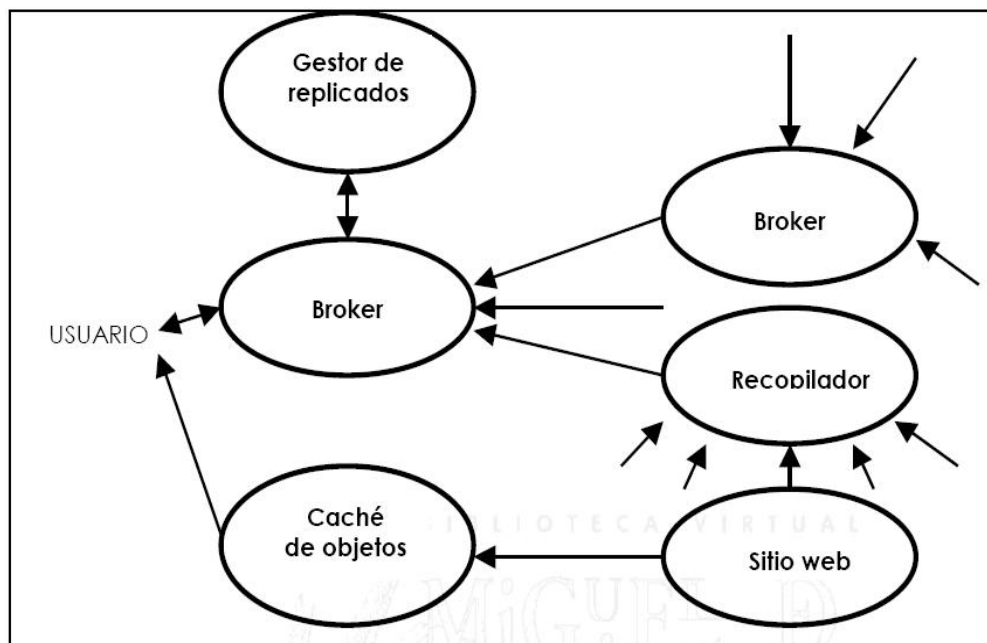


Fig. 1.2 arquitectura distribuida.

1.2.2 Buscadores basados en directorios y catálogos.

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.

Por otro lado los buscadores basados en los catálogos dependen por completo de los moderadores del sistema en cuestión, o sea personas “expertas” en las distintas materias que son las encargadas de construir y mantener el catálogo. Como variante, el catálogo puede subdividirse en subcatálogos que, a su vez, tienen sus catálogos, etc.

Como el registro se efectúa por una persona y no por un programa, la búsqueda por catálogos arroja mejores resultados en comparación con otros tipos de buscadores, además es posible obtener más datos que los obtenidos en un buscador automático, ya que las palabras claves no se seleccionan de forma automática sino que la efectúan los editores del catálogo, y como es revisado por personas se garantiza que la descripción se ajuste mejor a los intereses de los usuarios. Sin embargo, el costo de mantenimiento es muy elevado puesto que las personas encargadas deben registrar e incluir manualmente las descripciones de cada recurso de información, además de revisar constantemente las ya incluidas para detectar los cambios que puedan haber ocurrido.

Para el registro en un catálogo es necesario navegar hasta la categoría deseada y entonces solicitar la inclusión del recurso en cuestión, adicionando un conjunto de elementos que describen el contenido del mismo. Luego de un período de tiempo los editores revisan el recurso y comprueban que efectivamente corresponda con las características registradas, de ser así el recurso es publicado junto con la descripción y las palabras claves, pero de lo contrario puede ser que el registro sea denegado, o que el recurso sea colocado en otra categoría más apropiada según la opinión de los moderadores o sea incluido cambiando la descripción y/o palabras claves.

1.2.3 Buscadores híbridos.

Algunos buscadores disponen de ambos métodos. Los índices para el buscador se captan por la araña, y el catálogo se llena por los moderadores del sistema. De forma tal que los usuarios puedan realizar las búsquedas por palabras claves o navegar a través de las distintas categorías hasta llegar a las páginas deseadas. Este tipo de sistemas aunque hereda las ventajas de cada

uno también requiere de un gran costo de mantenimiento debido al factor humano que presenta. El Google es un ejemplo claro de este tipo de buscadores, su interfaz se muestra en la figura 1.3.

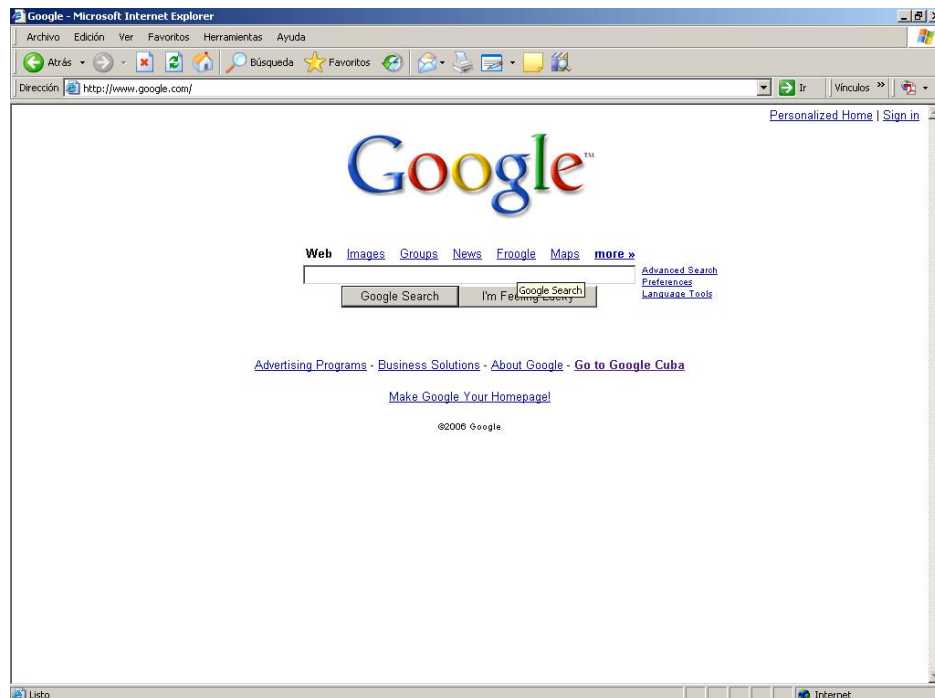


Fig. 1.3 Buscador Google.

1.2.4 Buscadores especializados.

Un buscador especializado viene a ser un sistema de manera basada en los buscadores anteriormente mencionados y que realizan la búsqueda dentro de un campo determinado. Aunque en centros como las grandes universidades existen sencillos buscadores más bien que recuperan datos y que están solamente constituidos por una herramienta para la búsqueda y una base de datos sencilla a partir de la que se obtiene la información que se desea buscar. Un

ejemplo de estos buscadores es el que se muestra en la figura 1.4 correspondiente a la Universidad Austral de Chile. Este sistema implementado en php realiza una búsqueda sencilla utilizando como criterio solamente el nombre y(o) el apellido de la persona que se desea buscar, a partir de los que se obtienen los resultados; donde cada elemento resultante tiene un enlace a la ficha de persona en la que se muestran algunos datos de la persona. Además los datos que son mostrados pueden ser actualizados por los usuarios a través de un formulario sencillo logrando así que la información siempre sea lo más actual posible.

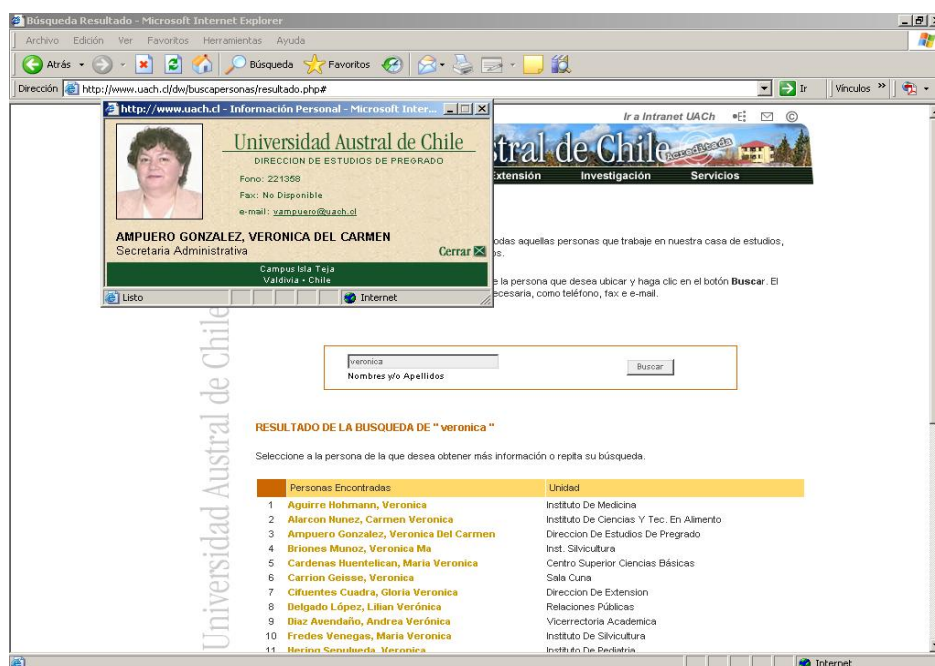


Fig. 1.4 Buscador de personas de la Universidad Austral de Chile.

1.3 Recuperación de datos.

La Recuperación de datos, RD, puede definirse como la representación, almacenamiento, organización y el acceso a determinada información. El campo de RD envuelve un conjunto bastante grande de conceptos, estructuras y métodos. Conceptualmente, la Recuperación de datos (RD) es una operación en la que se satisface una necesidad de información de un usuario y se seleccionan los datos que exactamente son capaces de solucionarla, es decir, consiste en buscar la información que exhiba la respuesta precisa a la pregunta formulada. La velocidad de respuesta de esa información depende entre otras cosas de la forma de almacenar los datos. Una de esas formas es usar la caché de la memoria como almacén temporal.

La recuperación de datos de una base de datos es ciertamente más lenta que la recuperación de datos almacenados en caché de la memoria. No obstante, el almacenamiento en caché no es una opción válida para todos los tipos de aplicaciones. Una aplicación que deba proporcionar información actualizada apenas si puede utilizar el almacenamiento en caché o, como mínimo, sólo se puede permitir almacenar los datos en caché durante un período de tiempo limitado. Cuanto menos tiempo se mantengan los datos en la memoria, menos ventajas se obtendrán del almacenamiento en caché. Sin embargo, cuanto más tiempo se almacene en caché un valor, más obsoleto se volverá. Para tomar una decisión es preciso considerar un equilibrio que dependerá de la aplicación concreta.

1.3.1 Recuperación datos de contra recuperación de información.

En la Recuperación de Datos se usa un emparejamiento exacto, pues se chequea si un elemento está o no está presente en un fichero. Sin embargo, en la Recuperación de Información buscamos aquellos que emparejen parcialmente con el pedido y de entre los seleccionados, tomamos los mejores. En la Recuperación de Datos el proceso es determinista, pues la pregunta

consiste en un par atributo-valor que coincide o no con los registros en una base de datos. En la Recuperación de Información el proceso es probabilístico, pues no se puede estar seguro de que en un documento extraído se recoge la información exacta que necesita el usuario; algunos documentos relevantes se pierden y otros no relevantes son recuperados.

En la Recuperación de Datos se usa una inferencia deductiva simple, es decir, si aRb y bRc entonces aRc . En la Recuperación de Información la inferencia es inductiva, pues las relaciones son expresadas con un grado de certidumbre y nuestra confianza en la inferencia es variable.

Por otro lado, en la Recuperación de Datos estamos interesados en una clasificación monotemática, es decir, las clases se definen por la posesión de determinados atributos, pero todos ellos son necesarios y suficientes para la pertenencia de un elemento a esa clase. En la Recuperación de Información la clasificación es politemática. Cada individuo de una clase posee sólo una proporción de todos los atributos que poseen todos los miembros de esa clase, por lo que no existe ningún atributo que sea necesario y suficiente para la pertenencia a dicha clase. El lenguaje de consulta en la Recuperación de Datos es artificial, por ejemplo, SQL y hay que especificarla de forma completa. En la Recuperación de Información preferimos usar el lenguaje natural y no hay un vocabulario restringido para especificar la consulta, la cual es incompleta. Además aquí sólo estamos interesados en recuperar los documentos relevantes.

Un sistema de Recuperación de Datos es más sensitivo al error en el sentido de que un error en el emparejamiento no recupera el elemento deseado y falla el sistema. En los de Recuperación de Información pequeños errores en el emparejamiento no afectan significativamente al sistema.

En la Recuperación de Datos la rapidez de respuesta depende del soporte físico y de la perfección del algoritmo de búsqueda y de los índices y el criterio de éxito a emplear es la corrección y la exactitud. En Recuperación de Información, por el contrario, la rapidez de

respuesta depende de las decisiones y acciones del usuario durante el proceso de interrogación y el único criterio de éxito a emplear es la satisfacción del usuario, basada en un criterio personal de utilidad.

Para diferenciar claramente ambos enfoques, piense en los resultados de formular una consulta a un sistema de gestión de Bases de Datos. En ellos una consulta exige un dato en un campo o atributo y la respuesta no deja lugar a dudas. Es un sistema cerrado, determinado y de respuesta exacta. Si no se obtiene respuesta es porque no la hay o porque el usuario ha introducido los datos de forma errónea. En cambio, en un sistema de recuperación de información textual, no basta con obtener las respuestas, hay que valorar si éstas son adecuadas o no, lo que debe hacer el usuario y volver a formular la búsqueda en caso de que sea necesario.

1.4 Bases de datos.

Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD).

1.4.1 Base de datos distribuida.

Es una colección de datos (base de datos) construida sobre una red y que pertenecen, lógicamente, a un solo sistema distribuido, la cual cumple las siguientes condiciones:

- La información de la base de datos esta almacenada físicamente en diferentes sitios de la red.
- En cada sitio de la red, la parte de la información, se constituye como una base de datos en sí misma.
- Las bases de datos locales tienen sus propios usuarios locales, sus propios DBMS y programas para la administración de transacciones, y su propio administrador local de comunicación de datos.
- Estas base de datos locales deben de tener una extensión, que gestione las funciones de sociedad necesarias; la combinación de estos componentes con los sistemas de administración de base de datos locales, es lo que se conoce como Sistema Administrador de Base de Datos Distribuidas.
- Este gestor global permite que usuarios puedan acceder a los datos desde cualquier punto de la red, como si lo hicieran con los datos de su base de datos local, es decir, para el

usuario, no debe existir diferencia en trabajar con datos locales o datos de otros sitios de la red.

En consecuencia, la base de datos distribuida, es como una unidad virtual, cuyas partes se almacenan físicamente en varias bases de datos "reales" distintas, ubicadas en diferentes sitios.

Ventajas de las bases de datos distribuidas.

Descentralización: En un sistema centralizado/distribuido, existe un administrador que controla toda la base de datos, por el contrario en un sistema distribuido existe un administrador global que lleva una política general y delega algunas funciones a administradores de cada localidad para que establezcan políticas locales y así un trabajo eficiente.

Economía: Existen dos aspectos a tener en cuenta.

- El primero son los costes de comunicación; si las bases de datos están muy dispersas y las aplicaciones hacen amplio uso de los datos puede resultar más económico dividir la aplicación y realizarla localmente.
- El segundo aspecto es que cuesta menos crear un sistema de pequeños ordenadores con la misma potencia que un único ordenador.

Mejora de rendimiento: Pues los datos serán almacenados y usados donde son generados, lo cual permitirá distribuir la complejidad del sistema en los diferentes sitios de la red, optimizando la labor.

Mejora de fiabilidad y disponibilidad: La falla de uno o varios lugares o el de un enlace de comunicación no implica la inoperatividad total del sistema, incluso si tenemos datos duplicados puede que exista una disponibilidad total de los servicios.

Crecimiento: Es más fácil acomodar el incremento del tamaño en un sistema distribuido, por que la expansión se lleva a cabo añadiendo poder de procesamiento y almacenamiento en la red, al añadir un nuevo nodo.

Flexibilidad: Permite acceso local y remoto de forma transparente.

Disponibilidad: Pueden estar los datos duplicados con lo que varias personas pueden acceder simultáneamente de forma eficiente. El inconveniente, el sistema administrador de base de datos debe preocuparse de la consistencia de los mismos.

Control de Concurrencia: El sistema administrador de base de datos local se encarga de manejar la concurrencia de manera eficiente.

Desventajas de las bases de datos distribuidas.

El rendimiento que es una ventaja podría verse contradicho, por la naturaleza de la carga de trabajo, pues un nodo puede verse abrumado, por las estrategias utilizadas de concurrencia y de fallos, y el acceso local a los datos. Se puede dar esta situación cuando la carga de trabajo requiere un gran número de actualizaciones concurrentes sobre datos duplicados y que deben estar distribuidos.

La confiabilidad de los sistemas distribuidos, esta entre dicha, puesto que, en este tipo de base de datos existen muchos factores a tomar en cuenta como: La confiabilidad de los ordenadores, de la red, del sistema de gestión de base de datos distribuida, de las transacciones y de las tazas de error de la carga de trabajo.

La mayor complejidad, juega en contra de este tipo de sistemas, pues muchas veces se traduce en altos gastos de construcción y mantenimiento. Esto se da por la gran cantidad de componentes Hardware, muchas cosas que aprender, y muchas aplicaciones susceptibles de fallar. Por ejemplo, el control de concurrencia y recuperación de fallos, requiere de personal muy especializado y por tal costoso.

El procesamiento de base de datos distribuida es difícil de controlar, pues estos procesos muchas veces se llevan a cabo en las áreas de trabajo de los usuarios, e incluso el acceso físico no es controlado, lo que genera una falta de seguridad de los datos.

1.5 Campo de acción.

La Universidad de las Ciencias Informáticas como todo centro académico cubano, esta orientada a la superación y formación de estudiantes y profesores, que sean dignos representantes de nuestro país en todos los aspectos. Al igual que las demás instituciones, está constituida por personas que interactúan cotidianamente intercambiando conocimiento o experiencias. Cada una de estas personas poseen datos propios que la identifican como un ciudadano que pertenece al centro docente. A través de esta información se controla por ejemplo para los estudiantes todos los datos relacionados con la docencia o con la ubicación en residencia, luego podemos decir que el acceso a esta información permite el correcto funcionamiento de un gran número de tareas dentro de la universidad. Razón por la cual en se ha prestado especial atención a la gestión de búsqueda de información del ciudadano dentro de la Universidad de las Ciencias Informáticas.

1.5.1 Descripción actual de los procesos de negocio.

La Universidad está formada por estudiantes, trabajadores en sentido general y otras personas que residen en el centro, de todos ellos actualmente existe un volumen bastante grande de información. La información general del ciudadano esta almacenada en la base de datos Ciudadano, en la que se pueden encontrar los datos propios de cualquier tipo de persona. La parte académica es gestionada por un sistema que permite el control de la asistencia, las evaluaciones y la relación de notas de los estudiantes, entre otras cosas. Las informaciones vinculadas con la residencia y los trabajadores son controladas por sistemas independientes que gestionan todos los procesos por los que debe pasar la misma.

Cuando un usuario desea acceder a información de la persona, ya sea estudiantes, trabajador o persona en general hace uso del sistema para la búsqueda de personas de la UCI (Directorio de búsqueda). A través de esta aplicación se puede acceder a información básica de la persona. Esta aplicación Web básicamente presenta dos interfaces para realizar la búsqueda, una para la búsqueda simple y otra para la búsqueda avanzada llamada, asistente para búsqueda. La primera interfaz conformada por un formulario contiene un región de texto y un botón, en ella se introduce una palabra o un patrón para la búsqueda; la segunda esta constituida también por un formulario pero este contienen varias regiones de inserción o selección de texto a partir de las que el sistema construye automáticamente el patrón para la búsqueda. Los resultados de la búsqueda de personas son divididos en dos grupos uno para estudiantes y otro para trabajadores; de un elemento de cualquiera de los grupos se puede ver además de la información preliminar, una ficha con la información un poco mas detallada.

1.5.2 Situación problemática.

El primer problema que podemos plantear es que la aplicación para la búsqueda de personas de la Universidad de las Ciencias Informáticas está soportada sobre la plataforma .NET que es software no libre (también llamado software propietario) lo que significa que es un programa informático donde el usuario tiene limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones) o que su código fuente no está disponible o el acceso a éste se encuentra restringido.

En el software no libre una persona física o jurídica (compañía, corporación, fundación, etc.) posee los derechos de autor sobre un software negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito; de estudiar cómo funciona el programa y adaptarlo a las propias necesidades; de distribuir copias; o de mejorar el programa y hacer públicas las mejoras.

El otro problema, podemos decir que está relacionado con el uso que hace la aplicación de los servicios Web para recibir la información de los proveedores de datos como Akademos por ejemplo. A través de estos servicios la aplicación recibe la información del sistema que realmente se dedica al procesamiento y control de la información solicitada en un momento determinado por la herramienta de búsqueda.

Los servicios Web son colecciones de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos mediante la red. Esta idea anterior podemos decir que es verdadera de manera general porque los servicios Web implementados en UCI utilizan la autenticación NTLM de Microsoft que de cierto modo no puede ser usada por aplicaciones implementadas con

herramientas que son software libre como php; lo que no permite hacer una migración sencilla hacia software libre de las aplicaciones exigentes en la universidad.

Finalmente, los otros problemas están relacionados con el mecanismo de funcionamiento del sistema de búsqueda que hacen que la aplicación proporcione resultados a veces no deseados. Un ejemplo de ello es la no correcta manipulación de los espacios en la cadena de búsqueda.

1.6 Sistemas automatizados existentes vinculados al campo de acción.

En la UCI existen un conjunto de sistemas que gestionan información de la persona dentro de un área determinada. La descripción de alguno de los que están relacionados con nuestro sistema se hará a continuación.

1.6.1 Sistema de gestión académica: Akademos.

Akademos es el sistema que gestiona toda la información académica dentro de la Universidad de las Ciencias Informáticas, a través de el se controla la asistencia y puntualidad de los estudiantes y el desempeño docente de los mismos. Este sistema esta conformado por secciones accesible para profesores y secciones que son accesibles para estudiantes. Actualmente esta solo accesible la sección para los profesores donde los mismos pueden llevar el control del registro de los grupos donde dan clases, documento donde se plasma toda la situación académica de cada estudiante. La parte destinada para los estudiantes es un módulo que todavía se encuentra en desarrollo, razón por la todavía no esta disponible.

1.6.2 Sistema de gestión de alojamiento.

El sistema de gestión de alojamiento controla todas las tareas relacionadas con la ubicación de las personas de la UCI. La parte de residencia de la universidad esta constituida por un gran número de edificios entre los que se distribuyen estudiantes y trabajadores. La distribución de las personas se hace por áreas que tienen cierta relación con la ubicación de las diez facultades de la universidad. Este sistema que controla todos los procesos en residencia permite que de una forma sencilla se lleven datos y estadísticas sobre el alojamiento en la Universidad de las Ciencias Informaticas haciendo eficiente el trabajo del personal de residencia.

1.6.3 Sistema de gestión de información de los trabajadores.

El sistema de gestión de información de los trabajadores controla todo lo que esta vinculado con los recursos humanos. En la UCI existen, además de los estudiantes, un gran grupo de personas, los trabajadores, que a través sus acciones contribuyen al correcto funcionamiento de la universidad. Estas personas trabajan en diferentes departamentos dentro del centro como facultades, comedores, entre otros. A través de la aplicación para la gestión de los datos de estas personas se puede tener siempre actualizada la información de estas personas y realizar otras operaciones relacionadas con el manejo de la información de valor sobre los recursos humanos.

1.7 Propuesta de solución.

Posterior al análisis realizado sobre el tema de las búsqueda de información en el que se comentaron detalles de alguno de los tipos buscadores y la descripción de la descripción actual sobre el objeto de estudio de este trabajo, concluimos que se hace necesario implementar un

nuevo sistema que permita buscar la información de la persona existente en la Universidad de forma eficiente y lo mas rápidamente posible.

Para la gestión de búsqueda de información del ciudadano de la Universidad de las Ciencias Informaticas se propone un sistema que utilizando programas que sean software libre integre toda la información del ciudadano. Este sistema a través de una herramienta para la búsqueda permitirá el acceso a la información de las personas. Por razones de tiempo y complejidad nos vamos a centrar en le módulo de búsqueda, tema que se tratará a lo largo del trabajo. Este sistema utilizará varias bases de datos que ya fueron creadas en la universidad a las que nosotros accederemos directamente.

1.8 Objetivos propuestos.

Teniendo en cuenta la situación de la información de la persona en la UCI se ha definido a modo de generalidad una propuesta de solución a partir de la que se espera que se solucionen unas cuantas dificultades. Para lograr la realización de la propuesta planteada se debe cumplir con una serie de objetivos divididos en generales y específicos a través de los que reflejan los fines y propósitos del proyecto y que además están determinados por la necesidad de resolver un problema: la gestión de búsqueda de información de ciudadano en la Universidad de las Ciencias Informaticas.

1.8.1 Objetivo general.

El objetivo general es desarrollar el módulo de búsqueda de una propuesta de Aplicación Web que permita la búsqueda y recuperación de información del ciudadano en la Universidad de las Ciencias Informaticas mediante avanzadas técnicas de gestión de búsqueda y auxiliándose

de las base de datos de los sistemas que gestionan la información del ciudadano en la universidad.

1.8.2 Objetivos específicos.

Para lograr el cumplimiento del objetivo general planteado en la sección anterior se deben cumplir los siguientes objetivos específicos:

- Realizar un estudio sobre las formas y mecanismos de búsqueda de búsqueda de manera general.
- Realizar un estudio sobre la situación de la información de la persona en la universidad.
- Diseñar la propuesta de solución utilizando una metodología de desarrollo de desarrollo de software de las actuales.
- Diseñar una interfaz amigable para la propuesta de solución en la que el usuario con conocimientos mínimos de informática pueda navegar sin problema.
- Implementar un mecanismo para la recuperación de datos de las bases de datos ya existentes en la universidad.

1.9 Conclusiones.

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que rodean al problema y en base a esto se obtuvieron los objetivos generales y específicos para este trabajo; aunque en esta etapa solo se habla de ideas, es correcto que estén bien fundamentadas, porque estas constituyen la base para el posterior desarrollo de este trabajo.

CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.

2.1 Introducción.

En el presente capítulo; se hace un análisis de alguna de las características de las tecnologías que pueden ser adecuadas para llevar a cabo el sistema que se pretende desarrollar. Y en algunos casos son necesarias comparaciones que fundamentarán la propuesta final.

Como elemento indispensable se analiza la posible metodología a utilizar para el análisis y diseño del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Y finalmente se exponen las propuestas de herramientas y tecnologías a utilizar para el desarrollo de la aplicación, teniendo en cuenta el resultado de los análisis anteriores.

2.2 La política de migración hacia software libre.

Recientemente se ha venido observando una tendencia en la Universidad de las Ciencias Informáticas, y en el país en general, hacia la utilización en grado creciente de software libre. Cada vez se promociona más la migración desde los sistemas con licencia comercial que están en poder de unos pocos gigantes de la rama de la informática (Microsoft es el mas representativo), y que se reservan el derecho de vender sus productos a quienes consideren pertinente, hacia aquellos que se denominan “libres”, es decir, aquellos cuyo uso por parte de cualquiera que esté interesado en hacerlo está exento de pago.

Como institución de avanzada en el campo de la informática, la UCI está prácticamente obligada a llevar a cabo, y cuánto antes mejor, esta migración. El presente trabajo parte de esa premisa y se propone la construcción de un sistema que satisfaga las necesidades que lo originaron, haciendo uso de herramientas y tecnologías libres.

2.3 Aplicaciones Web.

La plataforma Web tiene una arquitectura Cliente/Servidor. Funciona en modo “desconectado”, lo que significa que un usuario, usando un navegador o cliente, hace a través de la red, una petición de una página Web a un servidor Web, el servidor recepciona dicha petición, la procesa y le envía la respuesta al cliente (en formato HTML, también a través de la red), este la recepciona y se desconecta.

Se puede apreciar que los componentes de la arquitectura Web son: el servidor Web, la red física que permite la comunicación y un navegador o cliente. Existen casos que esta arquitectura es un poco más compleja, o sea, incluye un nuevo elemento: una aplicación que se ejecuta en el servidor. Este tipo de arquitectura permite manejar lógica de negocio a través de una “aplicación Web”, que es como se denomina a este tipo de sistemas Web. La aplicación que se ejecuta en el servidor se encarga de controlar el estado del negocio y de gestionar los datos almacenados con ayuda de algún Sistema Gestor de Bases de Datos (SGBD).

En la actualidad se ha generalizado el uso de aplicaciones Web dada las grandes posibilidades que brindan, y dado que los clientes sólo necesitan un navegador, capaz de interpretar código con formato HTML, para hacer uso de ellas: no tienen que instalar ningún componente de software adicional. Por tanto se propone que el sistema a desarrollar consista en una aplicación Web.

2.4 La seguridad de las transmisiones en la Web.

El reciente aumento del uso de Internet ha dirigido la atención del mundo entero a un problema crucial: la privacidad. Se ha tornado cada vez más importante el hecho de tener una protección real que garantice que los mensajes que se envían o se reciben no sean interceptados, leídos o incluso alterados por algún desconocido.

2.4.1 Protocolo SSL.

En la actualidad la mayoría de los navegadores utilizan el protocolo **SSL** (Secure Sockets Layer) para transmitir información de manera segura. Es un sistema diseñado y propuesto por Netscape Communications Corporation. En la pila OSI, se encuentra entre los niveles de TCP/IP y el de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen del y van al servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea reventada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones. MD5 se usa como algoritmo de hash.[1] Proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP. En la figura 2.1 se muestra la estructura del protocolo SSL

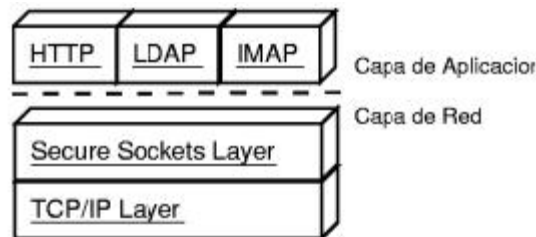


Fig. 2.1 Protocolo SSL.

Estas capacidades apuntan a las preocupaciones fundamentales sobre las comunicaciones sobre Internet y otras redes TCP/IP:

- La **autenticación del servidor SSL** permite al usuario confirmar la identidad del servidor. Los clientes SSL pueden utilizar técnicas de criptografía de llave pública para chequear que el certificado de un servidor y su identificación pública son válidas y han sido avalados por una autoridad certificadora (CA18) aceptada en la lista de CA confiables del cliente.
- La **autenticación del cliente SSL** permite a un servidor confirmar la identidad de un usuario. Usando las mismas técnicas que las usadas por la autenticación del servidor, un servidor SSL puede chequear que un certificado del cliente y su identificación pública son válidas y certificadas por un CA de confianza.
- Una **conexión SSL encriptada** requiere que toda la información enviada entre un cliente y un servidor sea encriptada por el software emisor y desencriptada por el software receptor, proveyendo un alto grado de confidencialidad. Además, toda la información enviada sobre una conexión SSL encriptada es protegida con un mecanismo para detectar si ésta fue alterada durante su tránsito en la red.

El protocolo SSL incluye dos sub-protocolos: el protocolo *SSL record* y el protocolo *SSL handshake*. El protocolo SSL record define el formato usado en la transmisión de los datos. El protocolo SSL handshake involucra el uso del protocolo SSL record para intercambiar una serie

de mensajes entre el servidor SSL y el cliente SSL cuando inician una conexión SSL. Este intercambio de mensajes esta diseñado para facilitar las siguientes acciones:

- Que el cliente autentique al servidor.
- Permitir al cliente y al servidor seleccionar los algoritmos criptográficos que son soportados entre ambos.
- Opcionalmente que el servidor autentique al cliente.
- Utilizar técnicas de encriptación de llave publicar para generar una llave secreta compartida.
- Establecer una conexión SSL encriptada.

2.4.2 OpenSSL.

El proyecto **OpenSSL** es un esfuerzo de colaboración para desarrollar un grupo de programas y rutinas robusto, de código abierto, que implemente los protocolos Secure Sockets Layer (SSL v2/v3) y Transport Layer Security (TLS v1), que es una versión estandarizada por el IETF (Internet **E**ngineering **T**ask **F**orce) del protocolo SSL y que pretende abarcar toda la capa de transporte de la pila OSI, así como una librería criptográfica potente de propósito general. Este proyecto es dirigido por una comunidad mundial de voluntarios que se comunican a través de Internet, y por esta misma vía planean y desarrollan la herramienta OpenSSL y su documentación.

OpenSSL se basa en la excelente librería SSLeay desarrollada por Eric A. Young y Tim J. Hudson y tiene licencia al estilo del servidor Apache, o sea, es completamente libre para su uso con fines comerciales y no comerciales, siempre que se respeten algunas sencillas condiciones que establece la licencia.

2.5 Servidor Web.

Entre los distintos tipos de servidores Web que existen en la actualidad, se seleccionó Apache-SSL para utilizarlo en la propuesta de solución.

Apache-SSL es un servidor Web seguro, basado en Apache y SSLeay / OpenSSL. Tiene licencia BSD, lo que significa que su uso tanto para fines comerciales como no comerciales es libre, siempre que se mantengan los avisos de derechos de reproducción (copyright notices).

El **servidor HTTP Apache** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor *parcheado*).

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

En la actualidad (2005), Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado. El núcleo 2.x de Apache tiene varias mejoras clave sobre el núcleo de Apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPV6.

La arquitectura del servidor Apache es muy modular. El servidor consta de un sección *core* y mucha de la funcionalidad que podría considerarse básica para un servidor Web es provista por módulos. Algunos de estos son:

- mod_ssl - Comunicaciones Seguras vía TLS.
- mod_rewrite - reescritura de direcciones servidas (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a como fueron desarrolladas estas páginas).
- mod_dav - Soporte del protocolo WebDAV (RFC 2518).
- mod_deflate - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- mod_auth_ldap - Permite autenticar usuarios contra un servidor LDAP.
- mod_proxy_ajp - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- mod_perl - Páginas dinámicas en Perl.
- mod_php - Páginas dinámicas en PHP.
- mod_python - Páginas dinámicas en Python.
- mod_aspdotnet - Páginas dinámicas en .NET_de_Microsoft.

Entre sus principales características se encuentran, entre otras:

- Libre para uso comercial y no comercial.
- Encriptación de 128 bits.
- Autenticación del cliente.
- Código fuente completo.

- API para extensiones modulares.

2.6 Fundamentación de la metodología utilizada.

Uno de los temas más comunes en el mundo de la informática hoy en día es el de las metodologías de desarrollo de software: cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos. Una metodología tiene como objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo, por medio de una mayor transparencia y control sobre el proceso; “producir lo esperado en el tiempo esperado y con el coste esperado”. [9]

Durante los últimos años se han desarrollado dos corrientes en lo referente a las metodologías de desarrollo de software, las llamadas “pesadas” y las llamadas “ligeras o ágiles”. Las primeras se basan en la idea de conseguir el objetivo común por medio de orden y documentación, mientras que las segundas tratan de lograrlo por medio de la comunicación directa e inmediata entre aquello que intervienen en el proceso. Analizaremos a continuación tres de las más conocidas, sus características, ventajas y desventajas.

2.6.1 El proceso Unificado de Rational: RUP.

RUP son las siglas en inglés de Rational Unified Process. Esta es una de las metodologías más generales de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto, no sólo de software. Es el producto final de treinta años de desarrollo y uso práctico. Se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal.

RUP “se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes”. [10] Cada uno de estos ciclos se divide en cuatro fases (Inicio, Elaboración, Construcción y Transición) en cada una de las cuales se llevan a cabo una o varias iteraciones, cada una de las cuales resulta un incremento de la precedente, por lo que se dice que RUP es iterativo e incremental.

Esta metodología también tiene la ventaja de venir acompañada de una potente herramienta que soporta todos los procesos básicos de RUP: Rational Rose Enterprise Edition 2003.

2.6.2 Programación Extrema: XP.

La Programación Extrema, o eXtreme Programming, es otra de las metodologías de desarrollo de software que existen en la actualidad. “Mientras que el RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.” [9]

Supone la disposición en todo momento, por parte del equipo de trabajo, de un representante competente del cliente, que debe estar en condiciones de dar una respuesta rápida y correcta a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.

La base para el desarrollo del software que usa esta metodología son las llamadas User Stories, historias escritas por el cliente en las que describe escenarios sobre el funcionamiento del sistema y que no sólo están limitados a la interfaz de usuario, sino que también pueden describir modelos, dominio, etc. Estas User Stories junto a la arquitectura que se persigue, sirve de base para crear un plan de “entregas de software” entre el equipo de desarrollo y el cliente,

para cada una de las cuales se definen objetivos y las iteraciones (generalmente cortas) necesarias para cumplirlos. Las User Stories y los casos de pruebas son la base sobre la que se asienta el trabajo del desarrollador. Esta metodología apuesta por iteraciones cortas que generan software que el cliente puede ver.

La codificación del software se realiza siempre en parejas (dos programadores, un ordenador), las cuales no son fijas sino que rotan a lo largo del proyecto, y el código que escriben no les pertenece sólo a ellos sino al equipo completo. El objetivo ideal sería que cada integrante del equipo trabaje al menos una vez con cada uno de los demás integrantes y con cada componente software, de forma que el conocimiento de la aplicación completa lo posea el equipo entero y no unos pocos miembros. Se programa sólo la funcionalidad requerida para la entrega en curso, se trabaja en función de las necesidades del momento, por lo que no se le da importancia al análisis como fase independiente.

2.6.3 Desarrollo Guiado por la Funcionalidad: FDD.

Se podría considerar que FDD (por las siglas en inglés de Feature Driven Development) está a medio camino entre RUP y XP, aunque en realidad es más bien una metodología ligera. Está pensada para proyectos con un tiempo de desarrollo relativamente corto (menos de un año). Se basa en un proceso iterativo con iteraciones cortas de aproximadamente dos semanas que producen un software funcional, el cual puede ser examinado por el cliente y la dirección de la empresa. Cada iteración se define en términos de funcionalidades (de ahí su nombre) que son pequeñas partes del sistema con significado para el cliente.

Un proyecto que siga esta metodología estará dividido en cinco fases: desarrollo de un modelo general, construcción de la lista de funcionalidades, plan de entregas sobre la base de las funcionalidades a implementar, diseño basado en las funcionalidades y implementación basada en las funcionalidades. Todo el trabajo se realiza en grupo, aunque siempre hay un responsable,

que generalmente tiene mayor experiencia, que dice la última palabra en caso de no llegar a un acuerdo.

Las funcionalidades de cada entrega se dividen entre los distintos subgrupos del equipo y se implementan. El código escrito (las clases) tiene propietario, o sea, sólo quien lo crea puede modificarlo, lo que no ocurre en XP. Es por eso que en un subgrupo deben estar todos los propietarios de las clases implicadas, pudiendo un desarrollador pertenecer a varios subgrupos. También se contemplan como parte del proceso de implementación, la preparación y ejecución de pruebas, las revisiones de código y la integración de las partes que componen el software.

2.6.4 Selección de la metodología a utilizar.

Las tres metodologías analizadas tienen pocas similitudes entre si, aunque XP y FDD tienen algunas más al ser ambas del tipo de metodologías ágiles, orientadas al cliente y de iteraciones cortas. Por otra parte, es importante mencionar que algunos autores consideran que dado el carácter general de RUP, todas las otras metodologías son casos particulares de esta. Pero siempre se pueden comparar si se definen determinados criterios.

En cuanto a la forma en que se capturan los requisitos, RUP y XP crean como base los Use Cases (casos de uso) y las User Stories respectivamente, los cuales describen los requerimientos de la aplicación desde el punto de vista del cliente, definen los requisitos técnicos sin abordar detalles de implementación. FDD, por el contrario, no define la parte del proyecto en la que se obtienen los requisitos y sólo define el proceder a partir del momento que ya estos han sido identificados. La captura de requisitos es una parte vital de cualquier proceso de desarrollo de software, puesto que garantiza la eficacia de éste: que el producto final sea lo que espera el cliente.

En cuanto a la relación del equipo con el cliente, XP se basa en la disposición de un representante competente que reúna una serie de condiciones, pero en la práctica, es poco probable que el cliente pueda prescindir de una persona con esas características por un tiempo, ya que le resulta poco menos que imprescindible.

Por otra parte, XP es un proceso muy orientado a la implementación, en el que se genera poca documentación y en que la funcionalidad exacta del sistema final no se define nunca formal y contractualmente. Es por eso que este método es más aplicable para desarrollos internos.

En el caso particular de este proyecto, se decidió la utilización de la metodología RUP, por todas las ventajas de organización que brinda y porque las otras metodologías estudiadas presentan ciertas debilidades que, a juicio del equipo de trabajo, representan riesgos considerables, como es el caso de una posible captura de requisitos no adecuada.

2.7 Lenguajes de programación para la Web.

Entre los distintos lenguajes de programación para la Web que existen en la actualidad, se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente / Servidor característica de los sistemas Web. El primer grupo está formado por los lenguajes que se ejecutan en el servidor (en inglés, server side languages). Como ejemplos más sobresalientes tenemos algunos como PERL, ASP, PHP, Java, JSP, los módulos CGI, etc. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la información, etc. Dentro del segundo grupo se encuentran aquellos lenguajes que se ejecutan en el cliente (client side languages). En este caso están el JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

Es importante seleccionar el lenguaje a utilizar, tanto del lado del servidor como del lado del cliente. A continuación se hace un breve análisis de los lenguajes de uso más común en la actualidad, con vistas a hacer una apropiada selección.

2.7.1 Lenguajes del lado del servidor.

Practical Extracting and Reporting Language: Perl.

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el Web, y es considerado el lenguaje perfecto para este fin dadas sus facilidades en cuanto a la manipulación de texto. Es un lenguaje de libre uso. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP.

Active Server Pages: ASP.

Es una tecnología propietaria de Microsoft. Se utiliza exclusivamente en los servidores Web de Microsoft (Internet Information Server y Personal Web Server), lo cual constituye su principal desventaja. Los scripts ASP se ejecutan, por lo tanto, en el servidor y puede utilizarse conjuntamente con HTML y JavaScript para realizar tareas interactivas y en tiempo real con el cliente. Con ASP se pueden realizar fácilmente páginas de consulta de bases de datos, funciones sencillas como obtener la fecha y la hora actual del sistema servidor, cálculos matemáticos simples, etc. 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, y que presenta algunas diferencias en cuanto a sintaxis, de modo que se ha de tratar de distinta manera uno de otro. Para implementar ASP.NET es necesario instalar en el Servidor la Plataforma .NET.

Personal Home Page: PHP.

Es el acrónimo de Hipertexto Preprocessor. Es un lenguaje de programación del lado del servidor gratuito, de código abierto, e independiente de plataforma, muy rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML. Su sintaxis es muy parecida a la del lenguaje C, por lo que para cualquier programador que esté familiarizado con dicha sintaxis será fácil aprender a programar en PHP. Actualmente se encuentra en su versión 5. Es evidente que se ha convertido en la gran tendencia en el mundo de Internet. Las estadísticas arrojan que cada mes su uso crece en un 15% [2] y cuenta con una de las comunidades más grandes de Internet, lo cual facilita encontrar ayuda, documentación, y otros recursos relacionados.

PHP cuenta con un motor de plantillas denominado SMARTY que permite separar la lógica de la programación de la presentación, es decir, el código PHP del código HTML. Por tanto se puede modificar uno sin afectar el otro. SMARTY es sumamente rápido, entre otros aspectos porque cada plantilla se compila sólo una vez y sólo recompila aquellas que han sido modificadas. Permite que una misma plantilla sea utilizada por varias páginas PHP que muestren el contenido en el mismo formato, independientemente de que los procedimientos usados para obtener la información a mostrar sean diferentes.

Otra de las ventajas de utilizar PHP, es poder usar la librería ADOdb, una capa de abstracción de base de datos, de alta velocidad, y que tiene características avanzadas, como la gestión de sesiones, generación automática del código SQL, simulación de SELECT LIMIT para todas las bases de datos y monitorización del rendimiento. Su aprendizaje resulta muy fácil, sobre todo si se está familiarizado con la programación de Windows, ya que utiliza muchas convenciones de ADO. Esta capa lleva siendo utilizada desde el año 2000, actualmente, con una comunidad amplia de usuarios. Su licencia es más que flexible (BSD). Esto significa que se puede incorporar (incluso compilar) en las aplicaciones libre de tasas, sin necesidad de solicitar el

permiso del autor. Con PHP también se pueden realizar y consumir Servicios Web mediante la utilización de la librería NuSOAP .Las principales ventajas de PHP son:

- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una muy buena documentación en su página oficial.
- Es Libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.

2.7.2 Lenguajes del lado del cliente.

JavaScript.

Lenguaje de programación del lado del cliente mas utilizado por ser el navegador el que soporta la carga de procesamiento y además por ser compatible con la mayoría de los navegadores modernos. Es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Permite realizar efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Además permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se

puede crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Visual Basic Script.

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Está basado en Visual Basic. 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.

2.7.3 Selección del lenguaje a utilizar.

PHP, con todas las ventajas que presenta por su propia esencia (es el más rápido de todos los analizados, es multiplataforma, con una sintaxis familiar a los estudiantes de la universidad, y cuenta con gran disponibilidad de recursos en Internet); y por el hecho de contar con un motor de plantillas como SMARTY y una capa de abstracción de datos como ADOdb, se convierte en la más atractiva de las opciones analizadas. Por ello se propone su uso como lenguaje del lado del servidor.

Como lenguaje del lado del cliente se propone el uso de JavaScript, que es un lenguaje orientado a eventos e interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente. En sus inicios fue desarrollado por Netscape y la primera versión JavaScript 1.0 fue por primera vez introducida en su navegador Netscape Navigator 2.0. Luego surgió la versión JavaScript 1.1 que se introdujo en el Netscape Navigator 3.0. Sin embargo hasta este momento no era soportado por el navegador de Microsoft, Internet Explorer. Luego de algunos esfuerzos se logró total compatibilidad entre JavaScript 1.3, que está incluido en Netscape Navigator 4.06 y posteriores versiones, y ECMA-262, un estándar para JavaScript introducido por Microsoft en el Internet Explorer.

Cascade Style Sheets: CSS.

En términos de páginas Web se puede establecer una diferencia entre los términos estilo lógico y estilo físico. El estilo lógico se refiere a la lógica del documento: cabeceras, párrafos, etc., no se preocupa de la apariencia final, sino de la estructura del documento. Por el contrario, el estilo físico no se preocupa de la estructura del documento, sino por la apariencia final: párrafos con un cierto tipo de letra, tablas con un determinado color de fondo, etc.

La finalidad de las hojas de estilo es crear unos estilos físicos, separados de las etiquetas HTML (en lugar de como parámetros de las etiquetas), y aplicarlos en los bloques de texto en los que se quieran aplicar. Estos estilos podrán ser modificados en algunas ocasiones desde JavaScript, lo que brinda mayor interactividad. En resumen, las hojas de estilo permiten separar el formato visual de las páginas de contenido como se muestra en la figura 2.2. Por la ventaja que representan, se propone su utilización en la propuesta de solución.

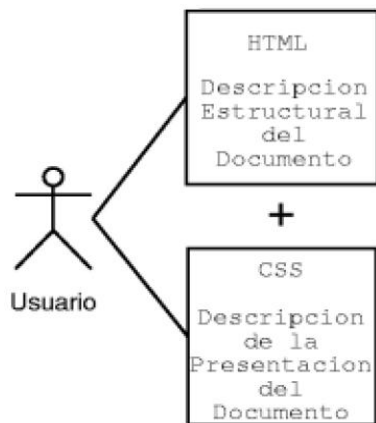


Fig. 2.2 HTML+CSS.

2.7.4 Sistemas Gestores de Bases de Datos: SGBD.

“Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y Data Base Management System, su expresión inglesa.” [4]

En la actualidad existe una gran variedad de SGBD, tanto de tipo comercial como libre. Entre los más usados dentro del grupo de los comerciales se encuentra **Oracle**, el cual es considerado el SGBD más completo que existe. Sus características más destacadas son el soporte de transacciones, su gran estabilidad y seguridad, su escalabilidad, así como que es un sistema multiplataforma, entre otras ventajas. “Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias).” [5]

En sus inicios fue muy revolucionario dado que usaba la filosofía de bases de datos relacionales, algo que por los años 70, fecha en que surge Oracle, era todavía desconocido. Hasta hace poco su dominio en el mercado de los servidores de bases de datos empresariales era casi total, pero recientemente está sufriendo la competencia del MS SQL Server de Microsoft y de la oferta de otros SGBD libres.

SQL Server es un potente SGBD que está totalmente habilitado para Web. “Ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de bases de datos de gran tamaño. El SQL Server permite lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones.” [2] A pesar de todas las ventajas que presenta este SGBD, tiene el inconveniente de que, al igual que Oracle, no es un sistema libre.

MySQL “implementa funcionalidades Web que permiten un acceso a los datos, seguro y fácil, desde Internet.” [6] “Es uno de los SGBD más populares, desarrollado bajo la filosofía de código abierto.” [7]

La licencia GPL de MySQL obliga a distribuir cualquier producto derivado (aplicación) bajo esa misma licencia. Por tanto MySQL tiene sus restricciones: sólo es gratis si se está dispuesto a distribuir la aplicación que se quiere desarrollar bajo esa misma licencia GPL. Si se desea distribuir la aplicación comercialmente, entonces se debe pagar la licencia comercial de MySQL que permite hacer exactamente eso.

MySQL tiene como una de sus principales ventajas la velocidad en la lectura de datos, pero a costa de eliminar un conjunto de facilidades que presentan otros SGBD: integridad referencial, bloqueo de registros, procedimientos almacenados, entre otros. En recientes versiones de MySQL (la versión 4 y la 5) se incluyen algunas de estas características, pero indudablemente esto va en detrimento de la velocidad.

Por otra parte está **PostgreSQL** que está considerado como el SGBD de código abierto más avanzado del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales de alto calibre tales como DB2 u Oracle. [8]

Es un SGBD objeto-relacional, aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. 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).

Es altamente extensible: soporta operadores y tipos de datos definidos por el usuario. Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92. Cuenta con una API (del inglés 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. Tiene soporte para lenguajes procedurales internos, incluido un lenguaje nativo denominado PL/pgSQL, el cual es comparable con el lenguaje procedural de Oracle PL/SQL.

A diferencia de MySQL que, como se explicó anteriormente, tiene sus restricciones en cuanto al tema de las licencias, PostgreSQL es totalmente libre. Las polémicas entre los partidarios de MySQL y los defensores de PostgreSQL pueden clasificarse como del tipo “Guerra Santa”, junto a otras como Linux vs. Windows, Mac. Vs. PC, etc. Muchos desarrolladores en sus discusiones a través de la Web en torno al tema de qué es mejor: MySQL o PostgreSQL, recomiendan la utilización de PostgreSQL para la elaboración de un sistema robusto y para lograr mayor escalabilidad. La mayoría coincide en que cada SGBD tiene sus ventajas y desventajas, y que la elección de uno de los dos depende de lo que se quiera construir. Se destaca sobre todo que MySQL ha avanzado vertiginosamente comparado con PostgreSQL que ya lleva alrededor de 15 años de desarrollo.

2.7.5 Selección del SGBD a utilizar en la UCI.

Nuestra aplicación no posee una base de datos propia, pero pensamos que dadas las características de la misma, un buscador, que accede a todas las existentes en la Universidad y que va a ser utilizado por miles de usuarios simultáneamente es conveniente proponer la migración hacia un gestor de bases de datos potente y que sea capaz de responder a todo el tráfico generado. En la actualidad se utiliza Microsoft SQL Server 2000, un sistema potente y probado, pero tiene el inconveniente de ser un software propietario y caro. Por tanto se considera que la mejor decisión que puede tomarse es la de migrar todas las bases de datos existentes en

la universidad hacia PostgreSQL (específicamente la versión 8.0.1) ya que reúne las dos importantes características de ser un sistema robusto con variadas funcionalidades y de ser un sistema de código abierto.

2.8 Otras herramientas.

Como vamos a utilizar una aplicación Web para confeccionar la propuesta de este trabajo, se hace necesario tener en cuenta la utilización de un editor de páginas Web, y una herramienta para el trabajo con las imágenes.

Para estas funciones la elección no ha sido muy difícil, ya que la herramienta de creación de sitios Web más utilizada en la actualidad es el Macromedia Dreamweaver MX 2004, que aúna los editores Web Dreamweaver 4 (sencillez) y Dreamweaver UltraDev (amplias funcionalidades). Con esa herramienta podremos desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos, como es nuestro caso; además de contar con un soporte para aplicaciones PHP. También cuenta con un amplio soporte para la creación y utilización de CSS, para lograr un diseño fácil y óptimo.

Finalmente se escogió el Adobe Photoshop 7, como herramienta principal para crear las imágenes del Sistema, ya que se considera la aplicación estándar para el tratamiento digital de imágenes. Las continuas mejoras han hecho de este programa uno de los más profesionales para la edición y retoque fotográfico. Tiene un enfoque dirigido hacia los gráficos para la Web, y posee una total integración con su avanzada herramienta de producción Web: Adobe ImageReady 3.0.

2.9 Nuestra propuesta.

Luego del análisis llevado a cabo, se puede plantear una propuesta que consiste en desarrollar una aplicación sobre plataforma Web, que siga el modelo de un buscador y sea capaz de extraer información simultáneamente de varias bases de datos, además de otras funcionalidades, utilizando como lenguaje de programación del lado del servidor el PHP dada su portabilidad y eficiencia, apoyándose en el uso del generador de plantillas SMARTY dada la facilidad de trabajo que brinda, y el JavaScript para lograr una interactividad con el usuario en el navegador. Se propone el uso de ApacheSSL como servidor Web. Como metodología de desarrollo se utilizará RUP.

2.10 Conclusiones.

En este capítulo se hizo un análisis de las tecnologías a utilizar en el desarrollo de la propuesta de solución, así como la descripción de algunas tendencias vinculadas con el tipo de propuesta que se desea desarrollar. Se fundamentó la elección del lenguaje de programación, el servidor Web, y la metodología de desarrollo de software. Finalmente se planteó la propuesta que incluye dichos aspectos. A partir de este punto se comenzará el desarrollo de la propuesta de sistema como tal.

CAPÍTULO 3. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCION.

3.1 Introducción.

En el presente capítulo se hace la descripción de la propuesta que trae este trabajo, para ello y con el fin de poder entender el contexto en que se emplaza el sistema se describen los procesos del negocio que tiene que ver con el objeto de estudio, de acuerdo a esto se modela el negocio propuesto, identificándose los actores, trabajadores y los casos de uso correspondientes, para capturar correctamente los requisitos y poder construir un sistema correcto.

Además se enumeran los requisitos funcionales y no funcionales que debe tener el sistema que proponemos, lo que nos permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

3.2 Modelo de negocio propuesto.

Como primer paso para el mejoramiento de los procesos vinculados al campo de acción se propone la realización de algunas modificaciones al negocio. Una de estas modificaciones consiste en dar la posibilidad de que los resultados que se muestren producto de la recuperación de información sean paginados en grupos de una cantidad que puede ser seleccionada por el usuario. Además esto garantiza que la acceso a la información sea ordenado y la disminución de la velocidad de respuesta. A continuación se modela el negocio propuesto.

3.2.1 Actores del negocio.

Los actores del negocio son aquellas personas o sistemas que obtienen un resultado de valor de uno o varios procesos del negocio. Los actores del negocio estudiado se definen en la siguiente tabla. También puede verse su representación en UML en la Tabla 3.1.

ACTORES	JUSTIFICACIÓN
Usuario	Representa a una persona que va a utilizar el sistema para buscar información de una personas determinada; puede ser un estudiante, un profesor o un trabajador de la Universidad.

Tabla 3.1 Actores del negocio

3.2.2 Trabajadores del negocio.

Los trabajadores del negocio son aquellas personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos, pero no obtienen ningún resultado de valor. Los trabajadores del negocio estudiado se definen en la siguiente tabla. Su representación en UML puede consultarse en la Tabla 3.2.

TRABAJADORES	JUSTIFICACIÓN
Sistema	Representa a la aplicación que realiza las operaciones de búsqueda y recuperación de información de las personas de la Universidad de las Ciencias Informaticas.

Tabla 3.2 Trabajadores del negocio

3.3 Descripción de los procesos de negocio propuestos.

A continuación se describen textualmente los procesos del negocio propuesto. Después se muestran los diagramas de casos de uso y los diagramas de actividad y de objetos de cada proceso del negocio. En ellos se muestran todos los procesos de negocio a través de su flujo de actividades.

CU Autenticar Usuario: El caso de uso inicia cuando el usuario desea hacer uso de la aplicación para la búsqueda de personas; generando esta última, para permitir el acceso, una ventana de autenticación en la que la persona debe introducir sus credenciales de dominio UCI para acceder al sistema, el cual realizará internamente una validación de la información introducida a través del directorio activo; si verifica que la información es válida se muestra al usuario la interfaz para realizar la búsqueda; en caso contrario el sistema redirige a la ventana de autenticación para que sean introducidas nuevamente las credenciales.

CU Buscar Información: El caso de uso inicia cuando el usuario, frente a la interfaz de búsqueda, selecciona el tipo de búsqueda que va a realizar: simple o utilizando asistente. Si el usuario escoge la simple, introduce directamente el patrón que va a ser utilizado por el sistema para realizar la búsqueda y ordena a la aplicación la realización de la operación; en caso de ser seleccionada la opción utilizar asistente el sistema genera una interfaz en la que el usuario configura las características que desea que tenga la búsqueda que va a realizar y cuando se da la orden de realizar la búsqueda el sistema confecciona automáticamente el patrón a partir del que va a realizar la búsqueda utilizando las características proporcionadas por el usuario. En cualquiera de los dos casos el sistema de manera general realiza un proceso de recuperación de información, presentándola al usuario en un reporte de la búsqueda compuesto por una sección de estudiantes y una de trabajadores donde de cada persona de una sección tiene una foto y algunos datos básicos. Dentro del reporte el usuario tiene la posibilidad de ver información más

detallada de una persona cualquiera a través de la selección de la opción más, a partir de la que el sistema genera una ventana mas pequeña que contiene, de la persona seleccionada, la información dada en el reporte y otros datos adicionales.

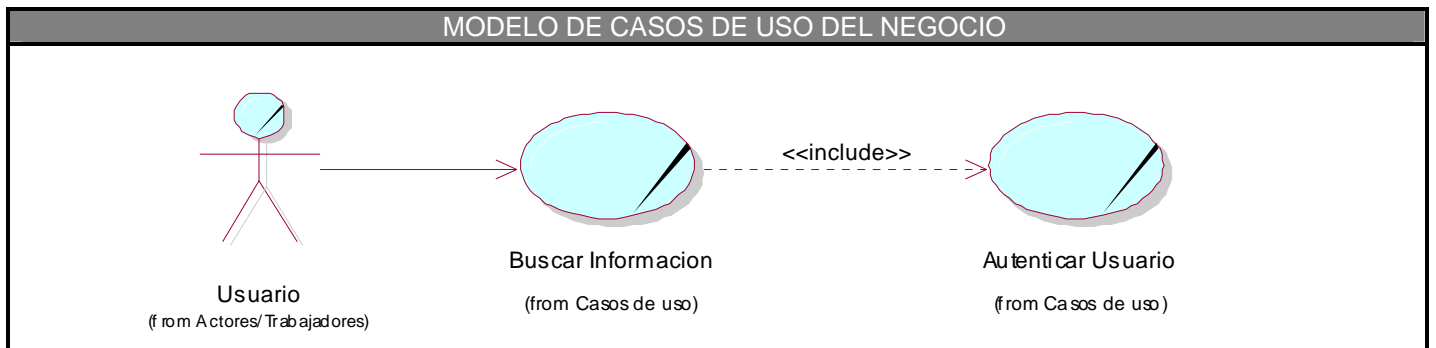


Fig. 3.1 Modelo de casos de uso del negocio.

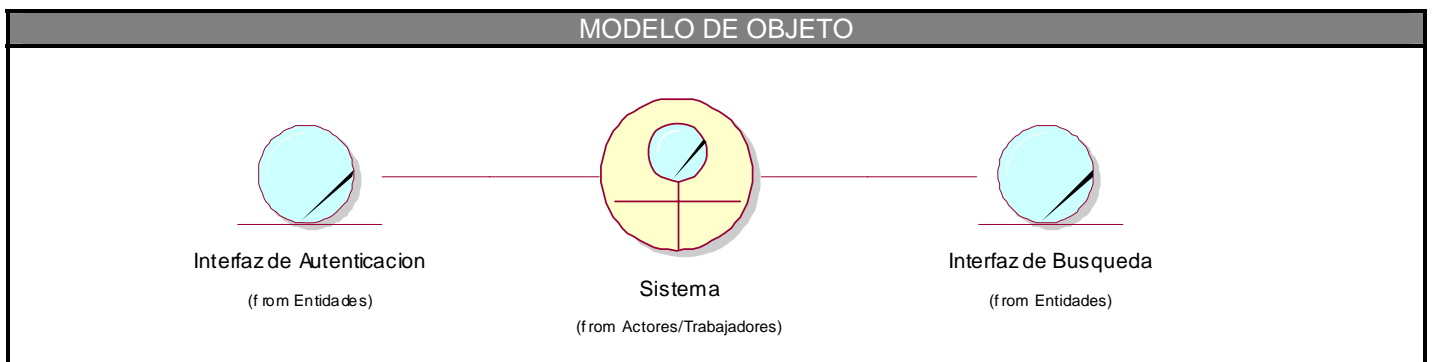


Fig. 3.2 Modelo de objeto CU Autenticar Usuario.

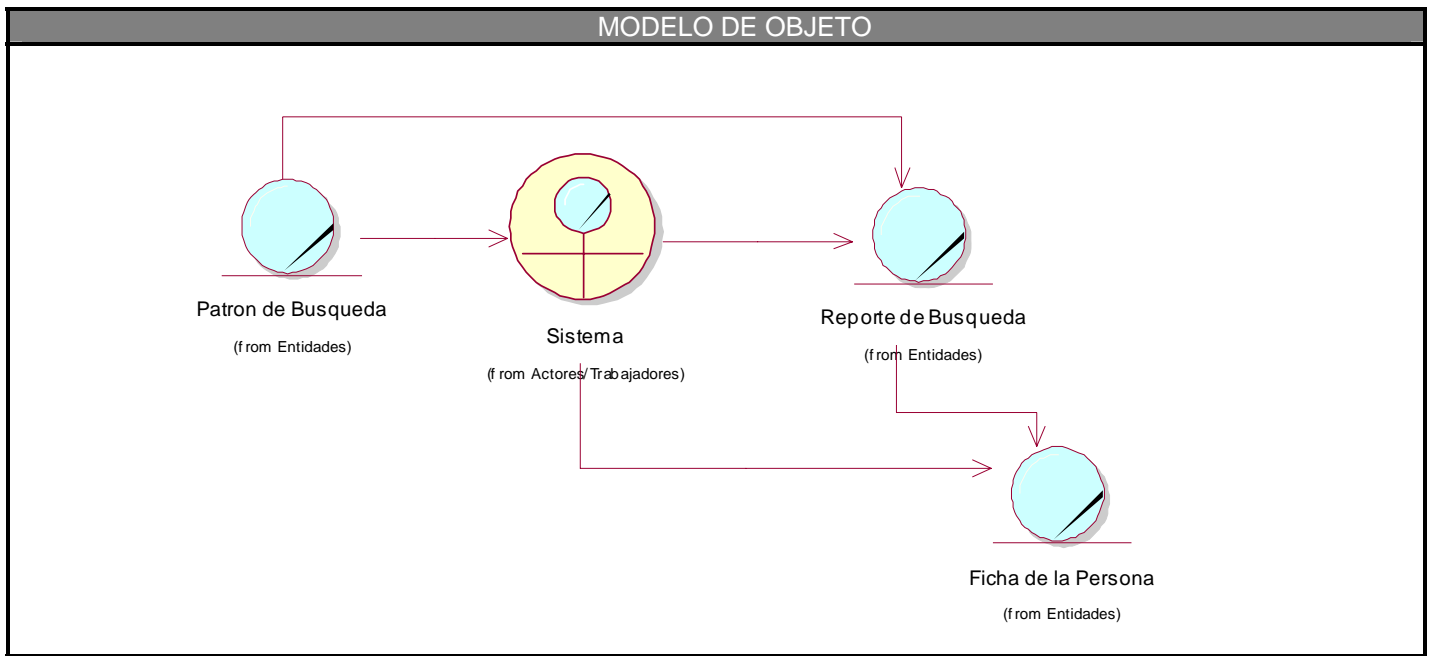


Fig. 3.3 Modelo de objeto CU Buscar Información.

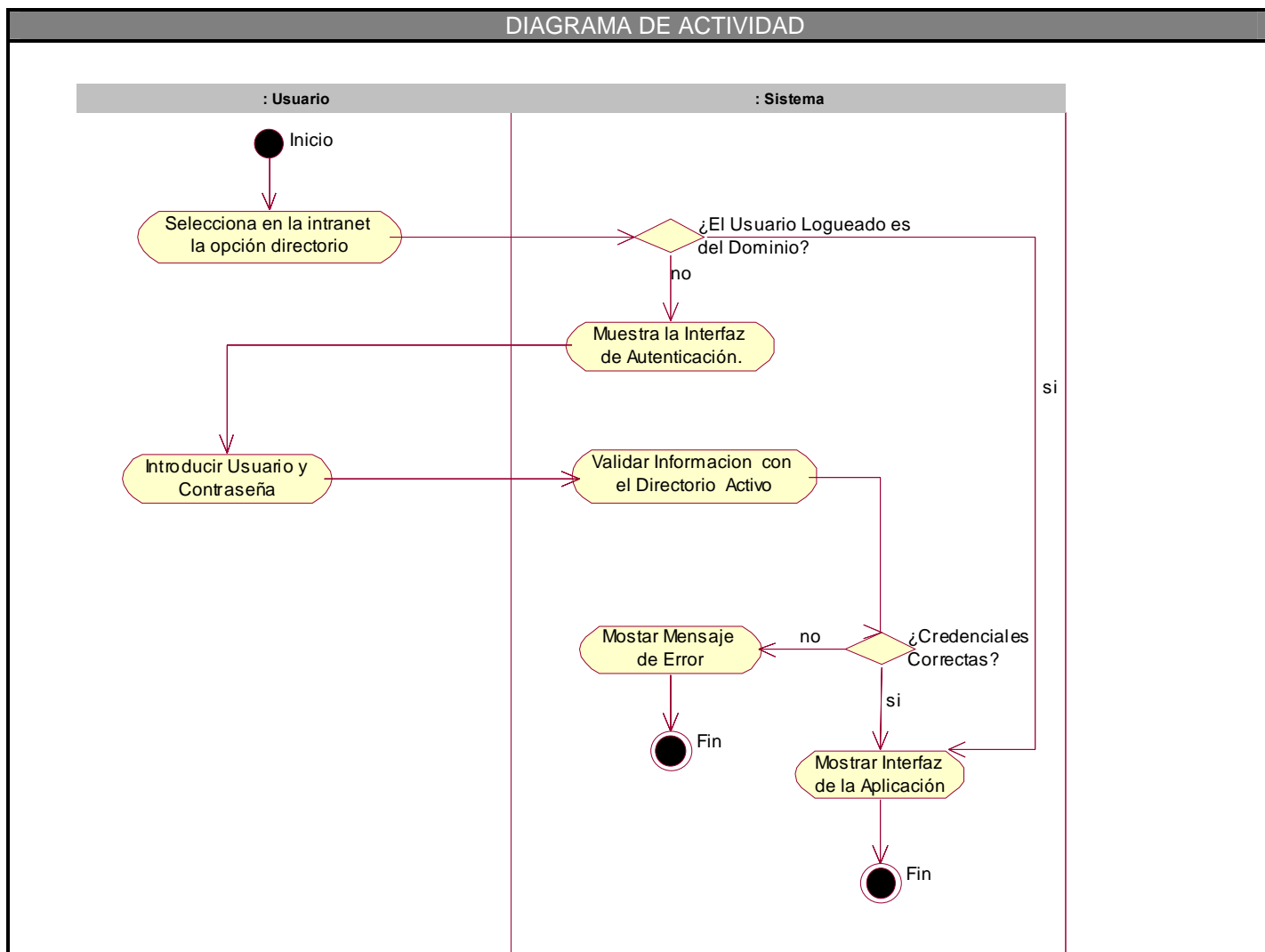


Fig. 3.4 Diagrama de actividad CU Autenticar Usuario

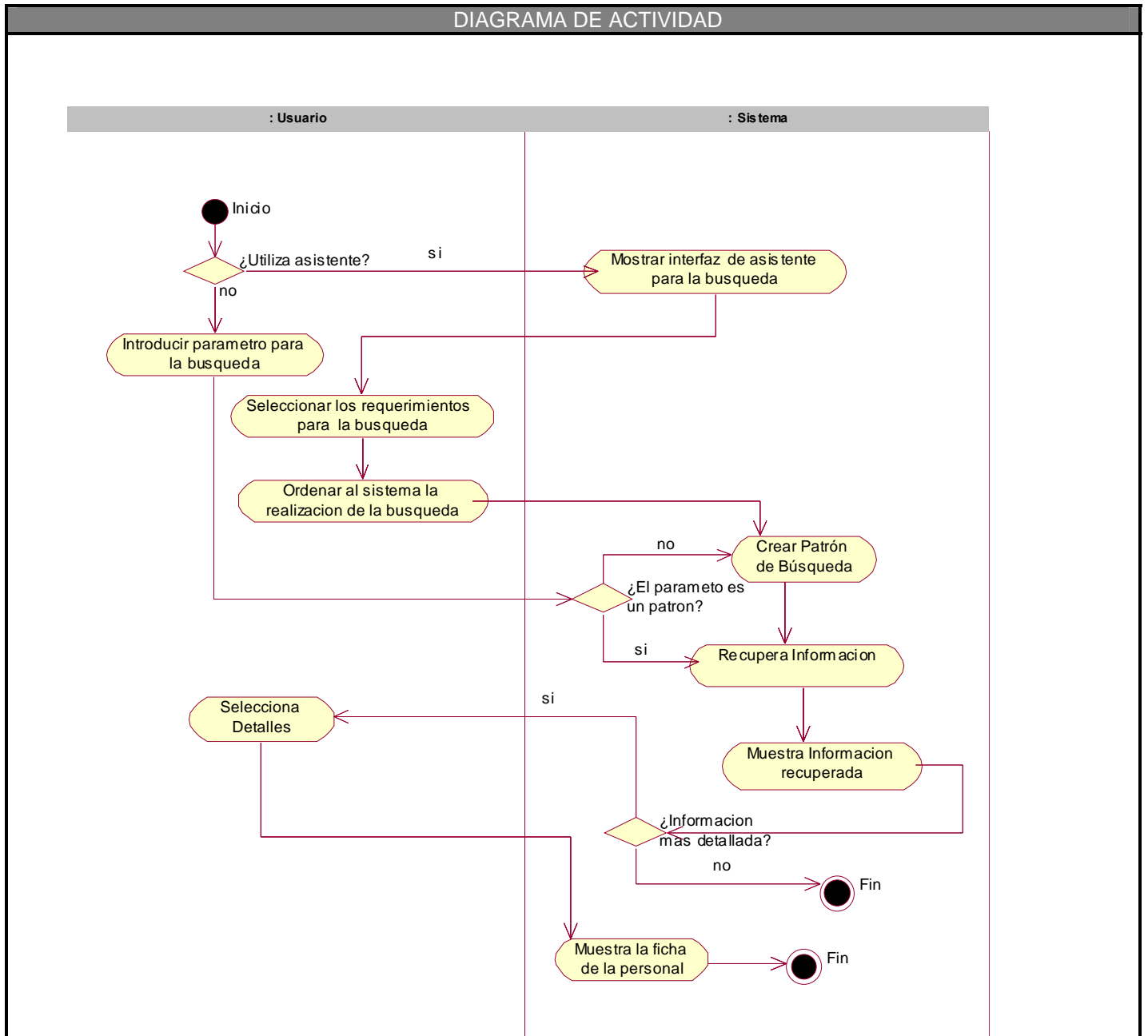


Fig. 3.5 Diagrama de actividad CU Buscar Información.

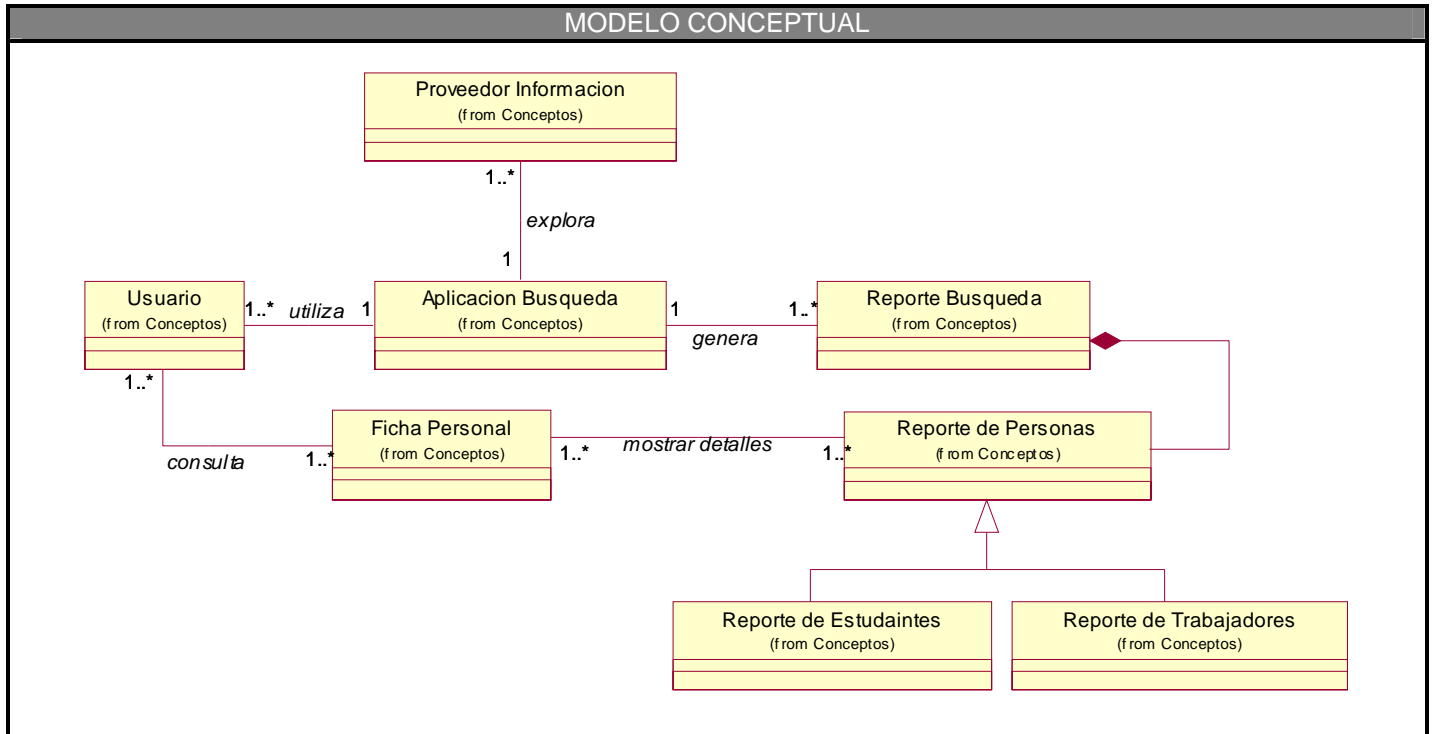


Fig. 3.5 Modelo conceptual.

3.4 Requisitos funcionales.

Una vez conocidos los conceptos que rodean al objeto de estudio, podemos empezar a analizar ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello enumeraremos a través de requerimientos funcionales las funciones que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones

extremas a determinar por el sistema. De acuerdo con los objetivos planteados el sistema debe ser capaz de:

R1-Autenticar usuario.

R1.1 Mostrar interfaz de autenticación al usuario.

1.1.1 Introducir nombre de usuario y contraseña del dominio.

1.1.2 Validar datos introducidos usando el directorio activo.

R1.2 Presentar al usuario la interfaz correspondiente a la búsqueda de información.

R 2-Gestionar búsqueda de información.

R2.1 Mostrar interfaz con las opciones de búsqueda: Simple y Utilizando Asistente.

R2.2 Utilizar la búsqueda simple.

2.2.1 Introducir la información a partir de cual se va a realizar la búsqueda.

2.2.1.1 Introducir información que será buscada exactamente con las características dadas como por ejemplo: nombre completo o nombre y algún apellido, etc..

2.2.2 Creación del patrón de búsqueda con la información.

2.2.3 Recuperar datos utilizando el patrón de búsqueda.

2.2.4 Mostrar listado utilizando paginación de personas que cumplen con los requisitos formado por la foto, el nombre el grupo en el caso de los estudiantes y la foto y el nombre en el caso de los trabajadores con la posibilidad en cada caso de ver información mas detallada de la persona en la ficha.

2.2.5 Permitir Reformulación de la búsqueda desde la misma página.

R2.3 Acceder a la búsqueda utilizando asistente.

2.3.1 Mostrar interfaz para la búsqueda utilizando asistente.

2.3.2 Seleccionar los requisitos en el asistente para realizar la búsqueda.

2.3.3 Crear patrón de búsqueda con los requisitos.

2.3.4 Recuperar datos utilizando el patrón de búsqueda.

2.3.5 Mostrar listado utilizando paginación de personas que cumplen con los requisitos formado por la foto, el nombre el grupo en el caso de los estudiantes y la foto y el nombre en el caso de los trabajadores con la posibilidad en cada caso de ver información mas detallada de la persona en la ficha.

R2.4 Acceder a la ficha de la persona seleccionada en la lista.

2.4.1 Presentar al usuario la interfaz de la ficha de la persona seleccionada.

2.4.2 Integrar en la ficha toda la información existente de la persona en diferentes bases de datos.

3.5 Requisitos 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.

Apariencia o interfaz externa:

- Diseño sencillo, permitiendo la utilización del sistema sin mucho entrenamiento.
- Diseño encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.

Soporte:

- Garantía de instalación y prueba del sistema.

Portabilidad:

- Independencia de la plataforma.

Seguridad:

- Transmisión de datos por la red a través de un protocolo seguro.
- Utilización de transacciones para el trabajo con los datos, para garantizar la integridad de los mismos.

Confiabilidad:

- Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

3.6 Descripción del sistema propuesto.

Para cumplimentar los objetivos propuestos al inicio de este trabajo, y teniendo en cuenta todos los requerimientos planteados, el sistema que proponemos debe tener un módulo de búsqueda que interactúa con diferentes sistemas que le proveen información, de acuerdo a esto, la efectividad de este depende de la calidad de la información existente en los proveedores.

Para utilizar todas las funcionalidades del sistema, se considera la existencia de 1 rol, o sea, la persona que utiliza la aplicación se comporta como el usuario que realiza la búsqueda. La aplicación utilizará para controlar el acceso el directorio activo, que es un servidor que utiliza para la comunicación el protocolo LDAP, en el que están registradas las credenciales de los usuarios que existen en la universidad.

El módulo del buscador puede ser utilizado por todos los usuarios de la universidad, y pretende utilizar la información existente en las bases de datos de los sistemas que le proveen información a través de un permiso de lectura a las mismas debido a la no existencia de un estándar de los servicios prestados por estos sistemas, tema en el que el departamento de informatización se encuentra trabajando, por lo que la aplicación debe ser fácilmente adaptable a los cambios en cuanto a la forma de pedir y recibir los datos que serán presentados a los

usuarios. Se especializarán las búsquedas en simple y utilizando asistente como lo hacen algunos buscadores dando la posibilidad de buscar por nombre, apellidos, nombre exacto, grupo, área etc., y de filtrar los resultados para obtener respuestas mas específicas. Gracias a los proveedores de información el resultado puede ser más específico en cuanto a la información de las personas recuperada por la búsqueda, e inclusive puede mostrar otros datos vinculados con la persona que puedan ser de interés para el usuario.

En resumen, con este sistema podemos realizar la búsqueda de información de la persona, sin importar la ubicación física de dicha información. De esta forma desde la misma aplicación podemos abarcar el universo de información de la persona en la Universidad de las Ciencias Informaticas. Y en la medida que la calidad de la información sea mayor, los resultados de las búsqueda serán mejores, permitiendo que dicha información pueda ser de mayor utilidad para toda la comunidad universitaria.

3.7 Modelo de casos de uso del sistema propuesto.

Utilizando las facilidades que nos brinda el UML, vamos a capturar los requisitos funcionales del sistema y a representarlos mediante un diagrama de casos de uso. Para ello tenemos que definir de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema, y los casos de uso que me van a representar las funcionalidades.

Un caso de uso es un documento narrativo que describe la secuencia de un actor (agente externo) que utiliza un sistema para completar un proceso. Un actor no es parte del sistema, es un rol de un usuario, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. En este caso con el sistema interactúan cinco actores que se definen a continuación:

3.7.1 Actores del sistema.

Los actores de un sistema son agentes externos: aquellas personas o sistemas que interactúan con él. En el Diagrama se puede observar la representación UML de los actores del sistema propuesto, que se describen en la siguiente tabla.

ACTORES	JUSTIFICACIÓN
Usuario del Sistema	Representa a una persona que va a utilizar el sistema para buscar información de una persona determinada; puede ser un estudiante, un profesor o un trabajador de la Universidad.
Ciudadano	Representa al sistema que controla la información general del ciudadano, y provee información al sistema de búsqueda.
Akademos	Representa al sistema de gestión de información académica, y provee información al sistema de búsqueda.
Trabajadores	Representa al sistema de gestión de información de los trabajadores, y provee información al sistema de búsqueda.
Residencia	Representa al sistema gestión de información de la beca, y provee información al sistema de búsqueda.

Tabla. 3.3 Actores del Sistema.

3.7.2 Diagrama de casos 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.

Los casos de uso del sistema están organizados en una serie de paquetes, según su área de aplicación. En la figura 3.6 se muestra el diagrama de casos de uso del sistema, donde se puede apreciar que está dividido en cuatro subsistemas principales:

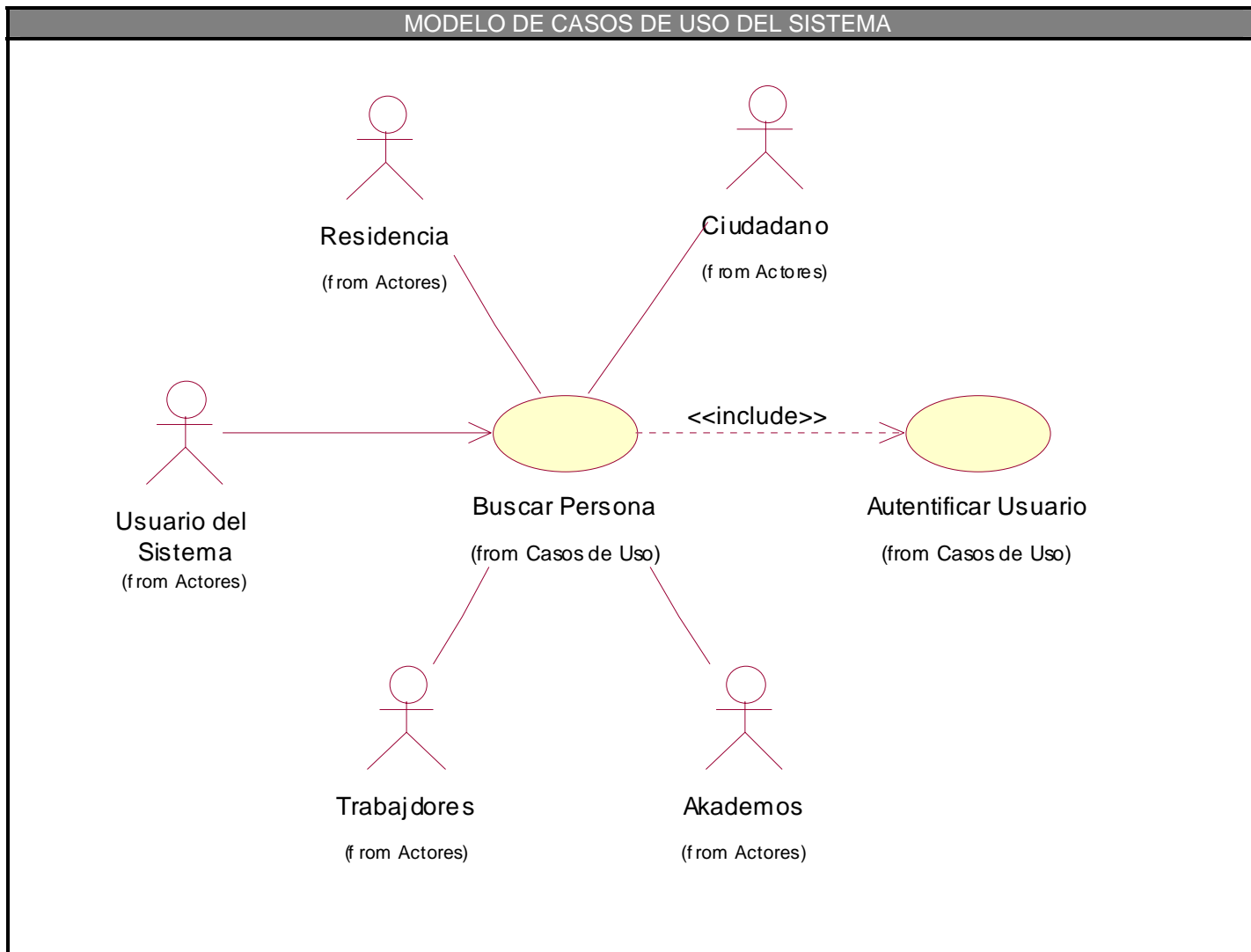



Fig. 3.6 Modelo de casos de uso del sistema.


3.7.3 Expansión de los casos de uso del sistema.

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. En este caso se van a describir los casos de uso que representan al módulo de busca de búsqueda de información; tema central de la investigación.

Caso de Uso:	Autenticar usuario.
Actor(es):	Usuario del Sistema (inicia).
Propósito:	Garantizar que en todo momento se controle el acceso al sistema.
Resumen:	El caso de uso se inicia cuando el usuario envía al sistema sus datos (nombre de usuario y contraseña). Si los datos fueron correctos el usuario accederá al sistema en caso contrario se negara el acceso y termina el caso de uso.
Referencias:	R1
Precondiciones:	El usuario ha introducido sus datos.
 <p>Pantalla 1</p>	
Acción del Actor	Respuesta del Sistema
	Muestra los controles necesarios para que el usuario introduzca su usuario y contraseña de dominio UCI.

CAPÍTULO 3. DESCRIPCION DE LA PROPUESTA DE SOLUCION

Introduce los datos requeridos y acepta la operación.	Valida la Identidad del usuario. En caso de que los datos no sean correctos, ver CA1.
Cursos Alternos	
<p>Mensaje de Error: Usuario o Contraseña Incorrecta... por favor: intentelo otra vez.</p> <p>Pantalla 2</p>	
CA1	Muestra mensaje de error embebido en la página.
Poscondiciones:	Mostrar al usuario la interfaz de la aplicación para la búsqueda.

Caso de Uso:	Buscar Personas
Actor(es):	Usuario del Sistema (inicia), Ciudadano, Residencia, Trabajadores, Avademos.
Propósito:	Recuperar recursos de información.
Resumen:	El caso de uso se inicia cuando el usuario decide buscar información, introduce el parámetro para la búsqueda si usa la búsqueda estándar o selecciona las características de la búsqueda si usa el asistente y se muestra una lista con las personas que lo cumplan junto con algunos datos básicos. El caso de uso no termina hasta que el usuario lo especifique explícitamente.
Referencias:	R2
Precondiciones:	El usuario se debe haber autenticado.
 <p>Pantalla 1</p>	
Acción del Actor	Respuesta del Sistema
1. El Usuario elige la operación a realizar.	2. <ul style="list-style-type: none"> a) Si elige la opción Búsqueda Simple, ver sección: "búsqueda simple". b) Si elige la opción Utilizar Asistente, ver sección: "Búsqueda Utilizando Asistente"
Sección: "Búsqueda Simple"	
1. Utilizando la Pantalla 1, el Usuario selecciona introduce el parámetro que desea para realizar la búsqueda y ordena que se le muestre el resultado de su	2. El sistema consulta la base de datos del Ciudadano y le presenta al usuario la Pantalla 3, con los resultados de la búsqueda. Ver Sección: "Resultado de Búsqueda".

búsqueda al presiona el botón Buscar.

Sección: “Búsqueda Utilizando Asistente”



Asistente de Búsqueda

Buscar:

Estudiantes

Nombre:

Provincia:

Municipio:

Apartamento:

Sexo:

Usuario:

Facultad:

Grupo:

Buscar

Buscador



Asistente de Búsqueda

Buscar:

Trabajadores

Nombre:

Provincia:

Municipio:

Sexo:

Usuario:

Area:

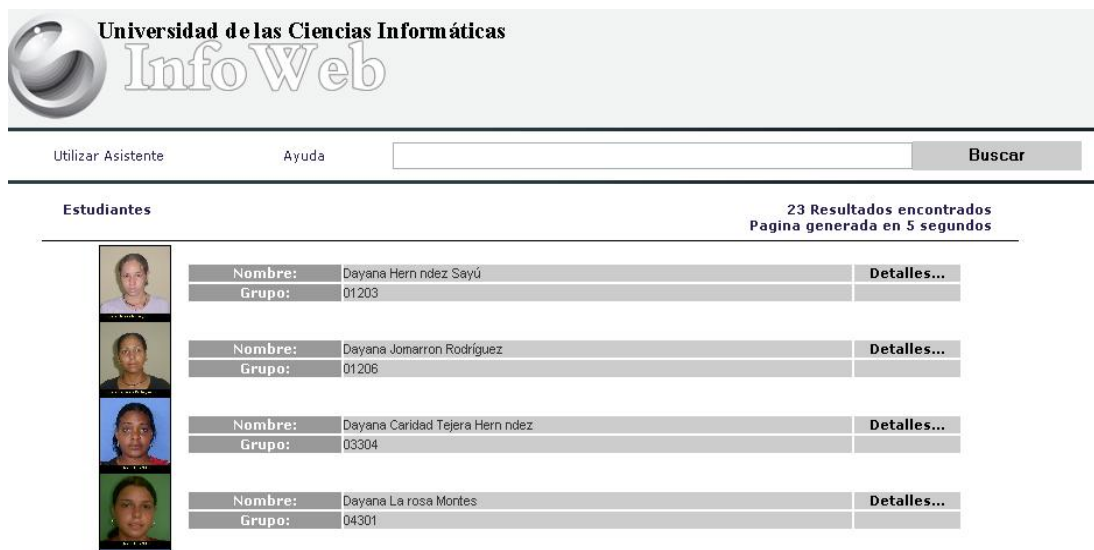
Buscar

Pantalla 2

1. El Sistema muestra la Pantalla 2, con los valores de los filtros de la búsqueda para estudiantes, nombre, provincia, municipio, sexo, usuario, facultad y grupo en blanco por defecto; además da la posibilidad de cambiar a la búsqueda, para trabajadores y mostrar los filtros para la misma; nombre, provincia, municipio, usuario y área, con su valor por


<p>2. El Usuario escoge el tipo de búsqueda que desea hacer, de estudiantes o trabajadores, a partir de ellas seleccionas los filtros que le quiere aplicar a la búsqueda y presiona el botón Buscar.</p>	<p>defecto. 3. El sistema consulta la base de datos del Ciudadano y le presenta al usuario la Pantalla 3, con los resultados de la búsqueda. Ver Sección: “Resultado de Búsqueda”.</p>
---	--

Sección: “Resultado de Búsqueda”



The screenshot shows the 'Universidad de las Ciencias Informáticas InfoWeb' interface. It features a search bar with 'Ayuda' and 'Utilizar Asistente' buttons, and a 'Buscar' button. Below the search bar, the 'Estudiantes' section displays '23 Resultados encontrados' and 'Pagina generada en 5 segundos'. A list of four student profiles is shown, each with a photo, name, group number, and a 'Detalles...' link.

Nombre:	Grupo:	Detalles...
Dayana Hernández Sayú	01203	
Dayana Jomarrón Rodríguez	01206	
Dayana Caridad Tejera Hernández	03304	
Dayana La rosa Montes	04301	




The screenshot shows the 'Universidad de las Ciencias Informáticas InfoWeb' interface. It features a search bar with 'Ayuda' and 'Utilizar Asistente' buttons, and a 'Buscar' button. Below the search bar, the 'Trabajadores' section displays '7 Resultados encontrados'. A list of two worker profiles is shown, each with a photo, name, and a 'Detalles...' link.

Nombre:	Detalles...
Alina Fernandez velazquez	
Alina Ruiz jhones	

Pantalla 3

	<p>1. El sistema lista los resultados recuperados por la búsqueda de la Sección: “Búsqueda Simple” o “Búsqueda Utilizando Asistente”.</p>
<p>2. El usuario puede elegir dos operaciones, mostrar información mas detallada de</p>	<p>3. a) Si elige mostrar información mas</p>

una persona seleccionada, o reformular la búsqueda.	detallada, ver sección: “Mostrar Ficha”. b) Si elige, reformular la búsqueda ver sección: “Reformular Búsqueda”.														
Sección: “Reformular Búsqueda”															
1. Utilizando la Pantalla 3, el usuario puede realizar nuevamente la búsqueda simple o utilizar el asistente para la búsqueda y presiona el botón Buscar	2. El sistema consulta la base de datos del Ciudadano y le presenta al usuario la Pantalla 3, con los resultados de la búsqueda. Ver Sección: “Resultado de Búsqueda”.														
Sección: “Mostrar Ficha”															
 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Nombre</td><td>Yaira Morales Aguilar</td></tr> <tr><td>Grupo</td><td>01205</td></tr> <tr><td>Municipio</td><td>Santa Clara</td></tr> <tr><td>Provincia</td><td>Villa Clara</td></tr> <tr><td>Correo</td><td>ymaguilar@estudiantes.uci.cu</td></tr> <tr><td>Apartamento</td><td>Apto 27203</td></tr> <tr><td>Teléfono</td><td></td></tr> </table>		Nombre	Yaira Morales Aguilar	Grupo	01205	Municipio	Santa Clara	Provincia	Villa Clara	Correo	ymaguilar@estudiantes.uci.cu	Apartamento	Apto 27203	Teléfono	
Nombre	Yaira Morales Aguilar														
Grupo	01205														
Municipio	Santa Clara														
Provincia	Villa Clara														
Correo	ymaguilar@estudiantes.uci.cu														
Apartamento	Apto 27203														
Teléfono															
Pantalla 4															
	1. El sistema muestra información mas detallada de la persona en una ventana mas pequeña.														
Poscondiciones:	Las búsquedas de información se han realizado														

3.8 Conclusiones.

En este capítulo se hizo una descripción de la propuesta de solución a través de la modelación del negocio propuesto: la identificación de los actores, trabajadores y los casos de uso correspondientes; el planteamiento de los requisitos funcionales y no funcionales de la aplicación que se va a desarrollar y la modelación de la misma en términos de casos de uso de sistema. A partir de este punto se puede comenzar a construir el sistema que constituye la propuesta de solución.

CAPÍTULO 4. CONSTRUCCIÓN DE LA PROPUESTA DE SOLUCIÓN.

4.1 Introducción.

En este capítulo se modelan los artefactos que ayudan a manejar las complicaciones que implican la construcción de aplicaciones Web. Para ello los componentes de la aplicación se tratan como clases, y utilizando las extensiones del UML, podemos presentarlas a través de diagramas de clases Web. Además se explica como el sistema recupera la información y el modelo relacional de uno de los proveedores de información del sistema. Finalmente después de modelar la lógica del negocio a través de las clases Web, se tratan los principios del diseño de la aplicación.

4.2 Modelo de diseño.

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales, creándose así una entrada apropiada para las actividades de implementación.

4.2.1 Diagrama de clases Web.

El diagrama de clases para las Aplicaciones Web, difiere un poco del resto de las aplicaciones que estamos acostumbrados a construir, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Para obtener un nivel correcto de abstracción y detalle que nos permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir: modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, una vez que estén

en el navegador del cliente; estos son los artefactos que necesitamos modelar para que el desarrollador los implemente luego y obtener así nuestro producto final.

De acuerdo a la forma en que hemos organizado el contenido del trabajo, consideramos que se debe presentar los modelos organizados por paquetes (que coinciden con los casos de uso), y subpaquetes, de forma que pueda entenderse mejor la lógica del negocio.

1. **Paquete 1.** Buscar Información: A-Generar búsqueda, B- Mostrar Ficha.
2. **Paquete 2.** Autenticación de Usuario.

Casi todas las páginas del servidor tienen relación con algunos componentes; como pueden ser componentes de acceso a la base de datos, de búsqueda, etc. Para una mejor comprensión de los modelos se explicará el propósito de cada clase.

Clase	Propósito
class_GENERAL	Fichero de configuración general, para el funcionamiento de las demás clases, y por tanto de la aplicación en general.
class_ACCESO_DATOS	Clase para la conexión con bases de datos Microsoft SQL Server, incluye el fichero de class_GENERAL. Crea un objeto conexión que permite hacer consultas, y recuperar los resultados.
class_MOTOR_BUSQ	Clase que representa el motor de búsqueda que se utiliza en el sistema, incluye el fichero de configuración general, y el de acceso a datos.
class_PATRON	Esta clase contiene todas las funciones utilizadas para construir el patrón que utiliza el motor de búsqueda para recuperar y presentar la información.
class_Autenticacion	Fichero de inclusión para el tratamiento de las sesiones. Contiene la creación, acceso, y manipulación de las variables de sesión usadas en el sistema además de controlar el acceso a la aplicación.

A continuación se presentan los diagramas de clases Web que representan los procesos más importantes, ocultamos los atributos y las operaciones de las clases para facilitar la comprensión de los mismos.

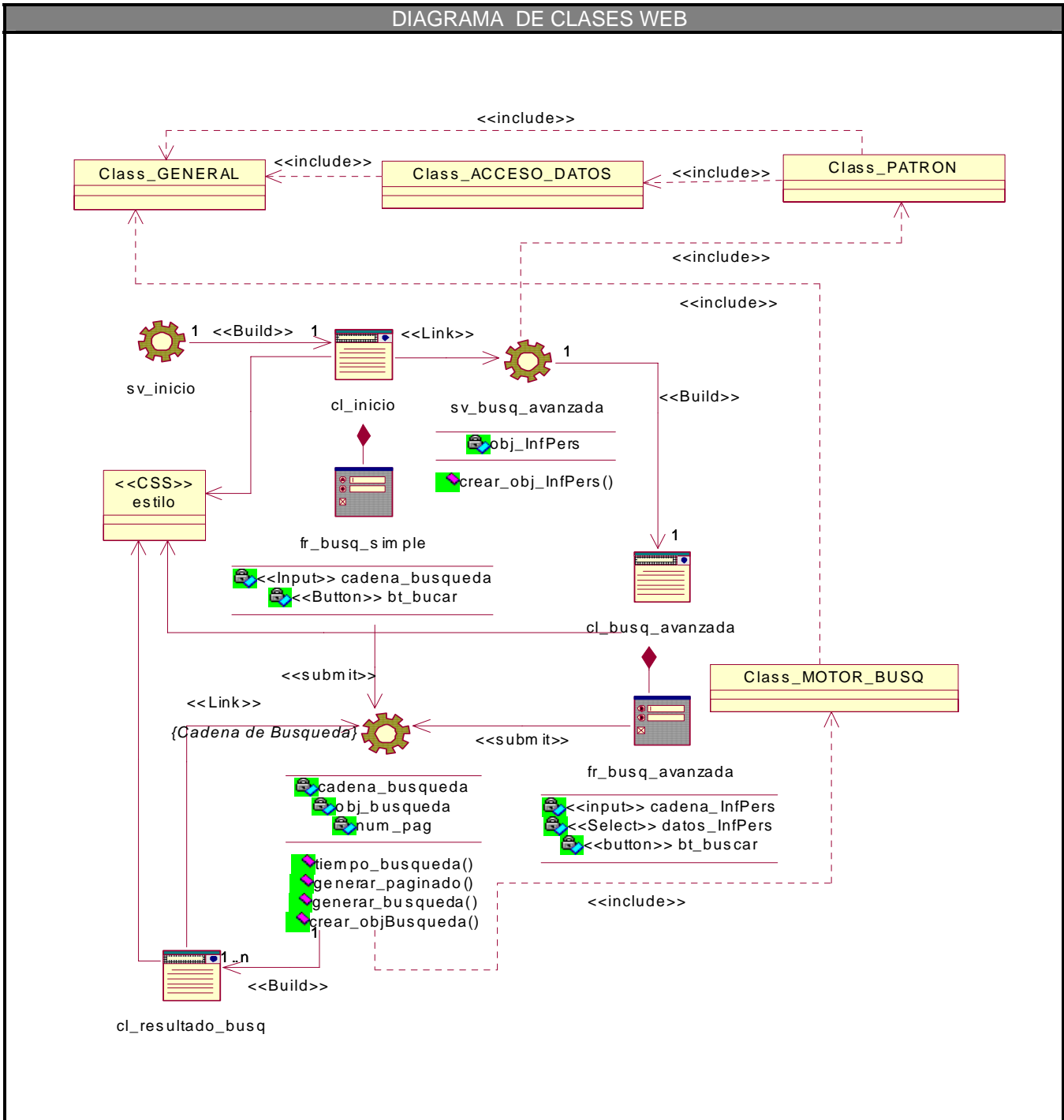


Fig. 4.1 Diagrama de clases Web CU Buscar Información: Generar búsqueda.

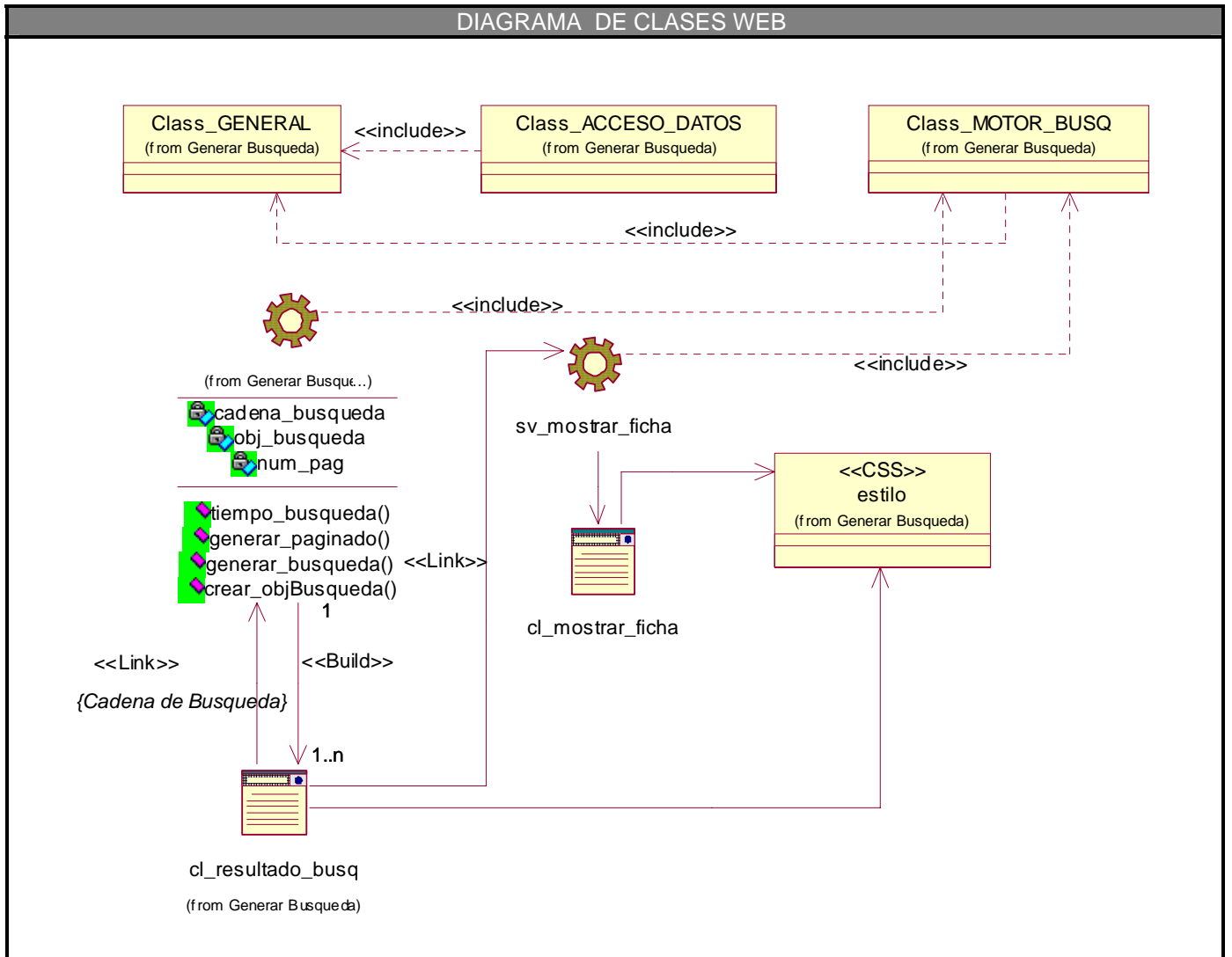


Fig. 4.2 Diagrama de clases Web CU Buscar Información: Mostrar ficha.

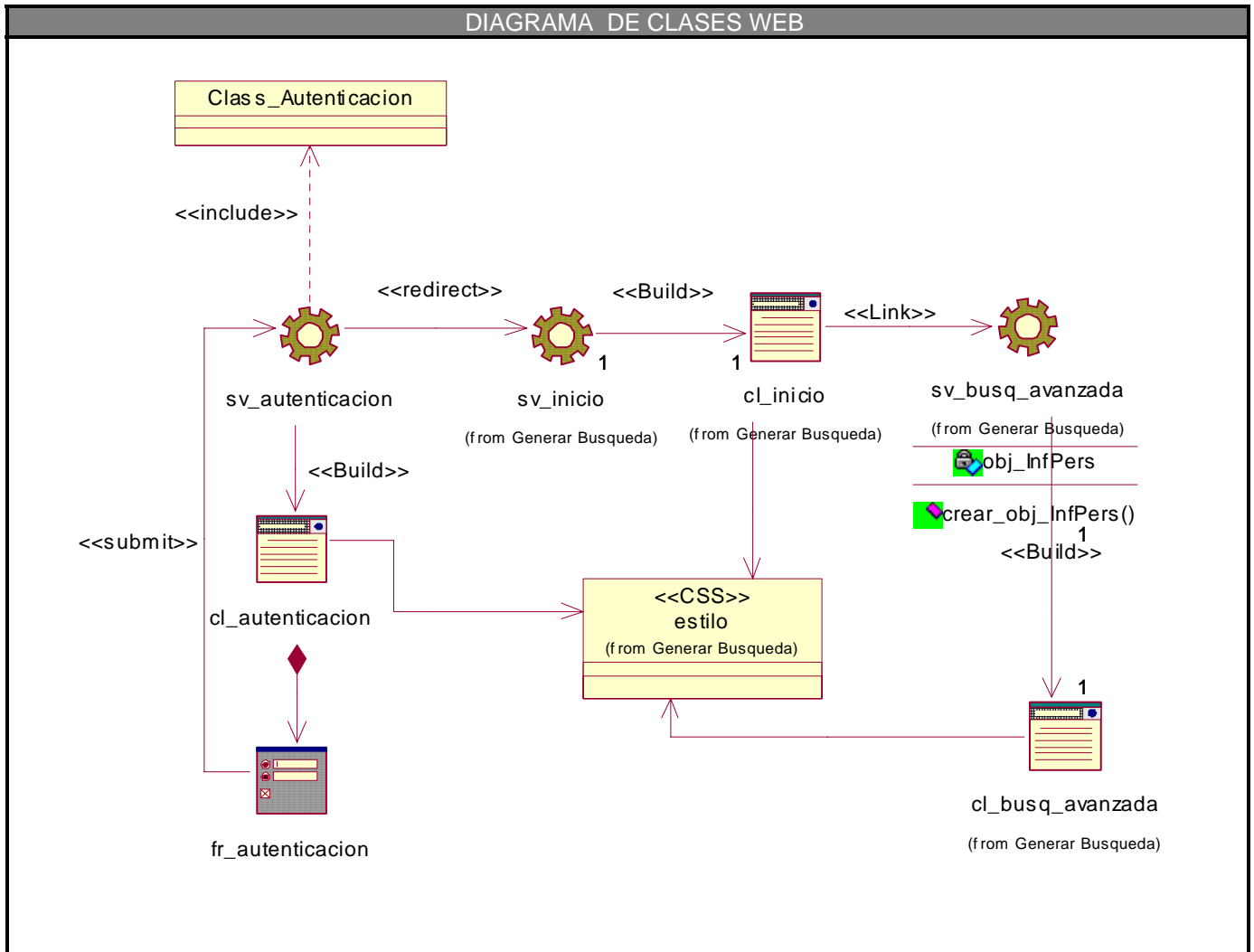


Fig. 4.3 Diagrama de clases Web CU Autenticación de Usuario.

4.2.2 Diseño de base de datos.

En el caso de InfoWeb se tiene un sistema que busca información de la persona pero que no usa una base de datos como repositorio, a partir de la que se haría el proceso de recuperación

de información por medio de una consulta creada por el sistema usando las entradas de usuario sino que se tienen varios proveedores de información a los que se les hacen pedidos para realizar el proceso de recuperación. En la figura 4.4 se muestra el modelo relacional de la base de datos de uno de estos proveedores.

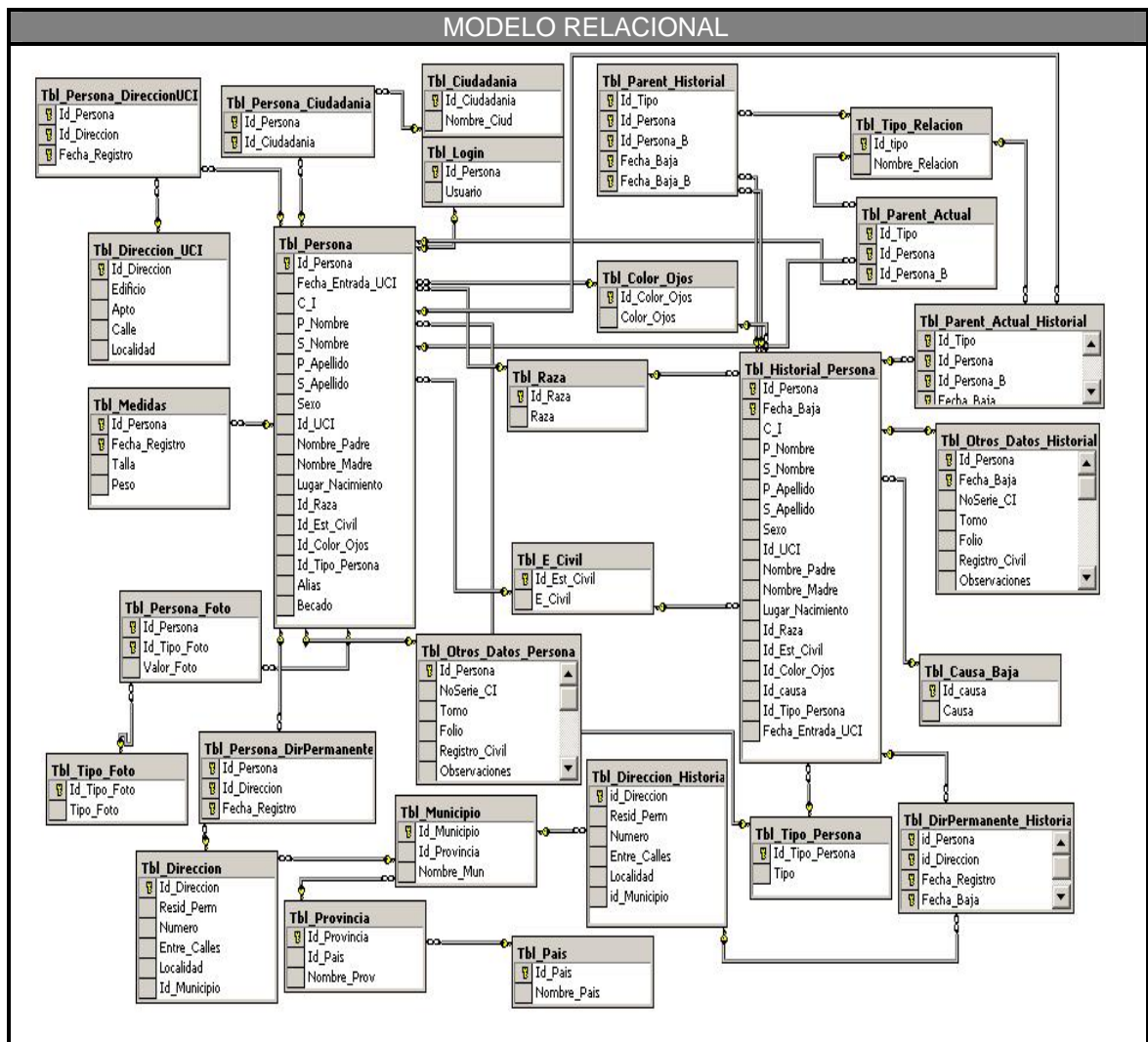


Fig. 4.4 Modelo relacional.

4.3 Patrones de diseño.

“En la terminología de objetos, el patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas. (...) Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software. Todo lo contrario: intentan codificar el conocimiento, las expresiones y los principios *ya existentes*: cuanto más trillados y generalizados, tanto mejor.” [11]

El diseño Web no se limita a la apariencia estética, a la combinación de colores, o a un logo más o menos acertado. De él depende que la información sea útil, que los servicios se puedan usar, es decir, el diseño convierte a una aplicación Web en algo atractivo para el usuario, por su estética y su utilidad.

Una aplicación con una interfaz bien diseñada debe tener, además de un buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos. Persiguiendo este objetivo se han seguido, para el diseño de la interfaz de usuario de InfoWeb, los siguientes principios:

- Requerir un mínimo proceso de aprendizaje y permitir su utilización desde el primer momento, por cualquier persona que tenga un mínimo dominio de la computación.
- Evitar el uso inadecuado o excesivo de las metáforas, que puede dificultar enormemente el aprendizaje del usuario.
- Garantizar la legibilidad, el color de los textos debe contrastar con el del fondo, y el tamaño de fuente debe ser suficientemente grande.

- Evitar elementos invisibles de navegación que han de ser inferidos por los usuarios, menús desplegados, indicaciones ocultas, etc.
- Requerir de los usuarios un mínimo esfuerzo para alcanzar sus objetivos.
- Cumple con el principio de la flexibilidad, donde se ofrezcan posibilidades de elección en los métodos de uso, que facilite al usuario la exactitud y precisión, y se adapte al paso o ritmo del usuario.

En el diseño de la aplicación se tuvieron en cuenta principalmente los patrones Experto, y Creador. El primero plantea que siempre se debe asignar una responsabilidad al experto en información, o sea, la clase con toda la información necesaria para llevarla a cabo. El segundo expresa que la responsabilidad de crear una instancia de una determinada clase debe asignarse a otra clase, siempre que esta agregue, contenga, registre o utilice específicamente los objetos de aquella.

4.3.1 Estándares de la interfaz de la aplicación.

Con vistas a lograr un diseño consistente de la interfaz de la aplicación, en todas las páginas se respeta el esquema formado por una cabecera, un navegador y el contenido. La cabecera contiene el logotipo de la aplicación en la esquina superior izquierda (el primer lugar, según estadísticas, que mira el usuario al visitar una página Web) la estructura está formada por una pequeña barra de navegación con enlaces de texto para cada una de las secciones de la aplicación. En el navegador, según la sección activa, se incluyen los enlaces a las correspondientes secciones restantes. En el área del contenido se muestran los formularios de entrada, los reportes de al búsqueda, etc.

Se utilizan para el diseño las tablas y plantillas, dado que son 100% compatibles con todos los navegadores, hasta en sus versiones más antiguas, a diferencia de los marcos. Se utiliza

también una hoja de estilos para guardar la configuración del diseño de todas las páginas. Esta hoja de estilos establece el tipo y tamaño de fuente de los distintos elementos de cada página. Se utiliza en general la familia de fuentes Sans-Serif, de tamaño entre 9 y 12 píxeles, según la importancia de la información mostrada. La hoja de estilos también establece el color de los vínculos, el color de fondo, el formato de los controles de formulario y las tablas, entre otros.

4.3.2 Formato del reporte de búsqueda.

El reporte se muestra en formato de tabla dividida en dos secciones una para estudiantes y otra para trabajadores, los elementos en cada sección se presentan ordenadamente. Dichas secciones muestran solo la información básica de cada elemento del reporte, como se muestra en la figura 4.5. En caso de que el usuario desee ver información mas detallada de una personas, se muestra junto a los datos la misma la palabra **Detalles..**, que es un vínculo a otra página donde se pueden ver los detalles de la personas en cuestión.



Fig. 4.5 Formato de un reporte de búsqueda.

4.3.3 Concepción general de la ayuda.

La aplicación InfoWeb va dirigida a la comunidad universitaria formada por estudiantes, trabajadores entre otras personas; donde una parte tiene poca calificación en el tema de la informática y la otra, conformada por la gran mayoría, con buenos o aceptables conocimientos del tema. Por este motivo se considera que una ayuda formal, sería lo más funcional, donde el usuario puede ver de cuantas formas puede realizar la búsqueda, como acceder al asistente para la búsqueda, etc.. En la figura 4.6 se puede ver un ejemplo de parte de la ayuda.

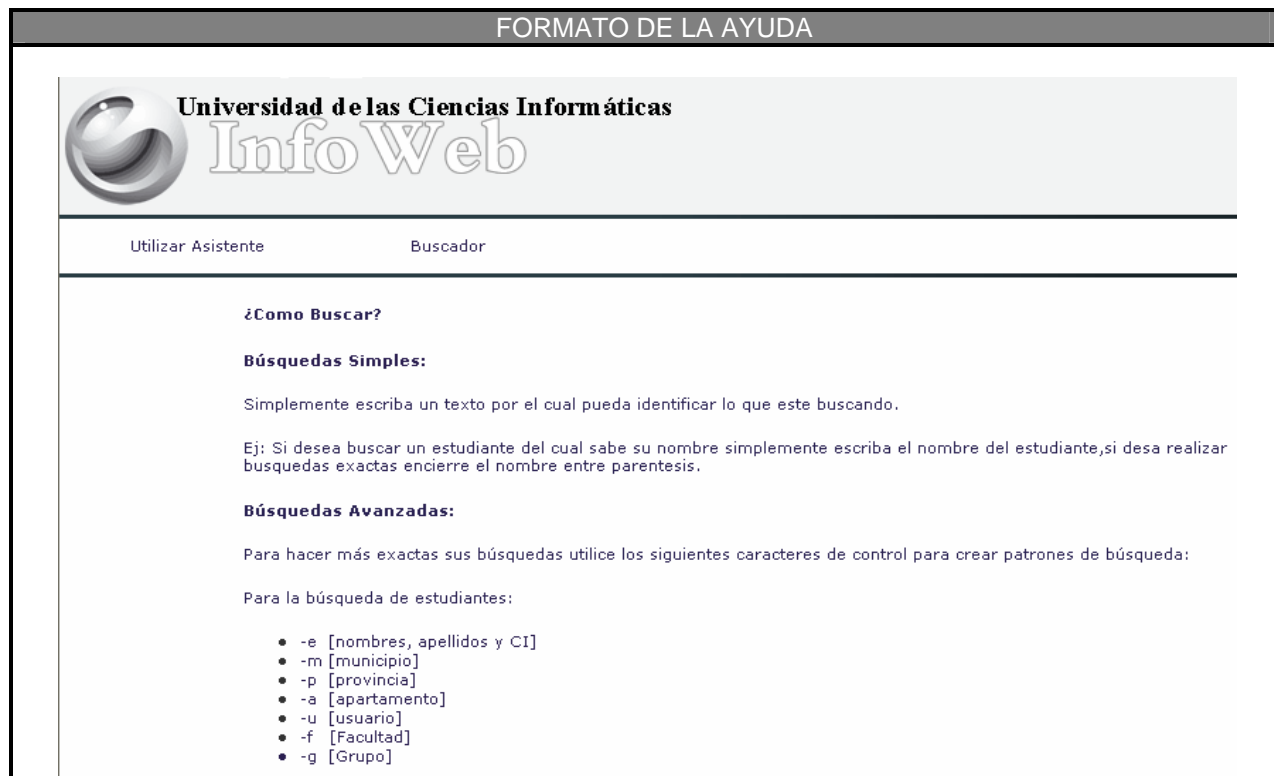


Fig. 4.6 Formato de la ayuda.

4.3.4 Tratamiento de excepciones.

Los errores en la aplicación se tratan de dos maneras diferentes. La primera consiste en chequear posibles situaciones excepcionales en el nodo cliente, utilizando el lenguaje JavaScript y sus mensajes de alerta. Esto se utiliza sobre todo en la validación de las entradas del usuario, para evitar que se ejecute nuevamente la página en el servidor con datos inconsistentes o incorrectos. La segunda forma se aplica cuando no es posible detectar el error en el cliente y se ejecuta la comprobación en el servidor. En este caso se muestra un mensaje embebido en el cuerpo de la página similar al de la figura 4.7.

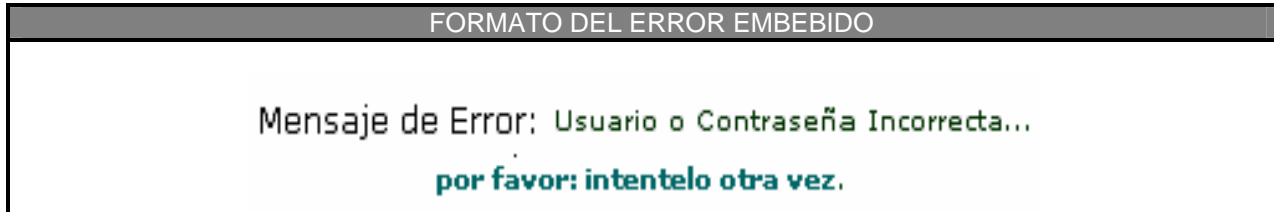


Fig. 4.7 Formato del error.

4.4 Estándares de codificación.

Con vistas a garantizar la homogeneidad del código dentro del grupo de desarrollo, se establece el estilo de código descrito a continuación.

Indentación: La indentación es a una tabulación. Ejemplo:

```
if($a != 1)
{
  echo("a es distinto de 1");
}
```

Comentarios: Los comentarios se definen comenzando con los caracteres */** y terminando con **/* para los comentarios de varias líneas, y comenzando con los caracteres *//* para los de una sola línea.

Además se deberá incluir en cada declaración de clase el siguiente comentario:

```
//-----//
// Clase: <Nombre de la clase>
// Propósito: < Propósito para el cual se diseño la clase>
//-----//
```

Y, para una mejor delimitación, antes y después de la implementación de cada método se debe incluir el siguiente comentario:

```
//-----
```


Declaraciones:

Las constantes se declaran en mayúsculas. Ejemplo: **define** ('CONSTANTE', 'ejemplo');

Las variables deben ser explícitas. Ejemplo: **var** \$NumeroDeResultados;

Para declarar una clase se comenzará con una C mayúscula y luego un sustantivo comenzando con mayúscula. Ejemplo: **class** CGenerales.

Espacios en blanco: Se colocarán espacios en blanco entre operadores lógicos-aritméticos y sus operandos. Ejemplo:

```
if($a == $b)
{
  $x = 5;
  $x = ($a + 1) * $b;
  hazAlgoConX($x);
}
```

Miscelánea: Las llaves para abrir y cerrar un método o un bloque de control de flujo se indentan al mismo nivel del bloque al que pertenecen. Ejemplo:

```
if($bandera != 1)
{
  while(!vacio())
  {
    hacerAlgo();
  }
}
```

4.5 Modelo de despliegue.

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware como se muestra en la figura 4.8.

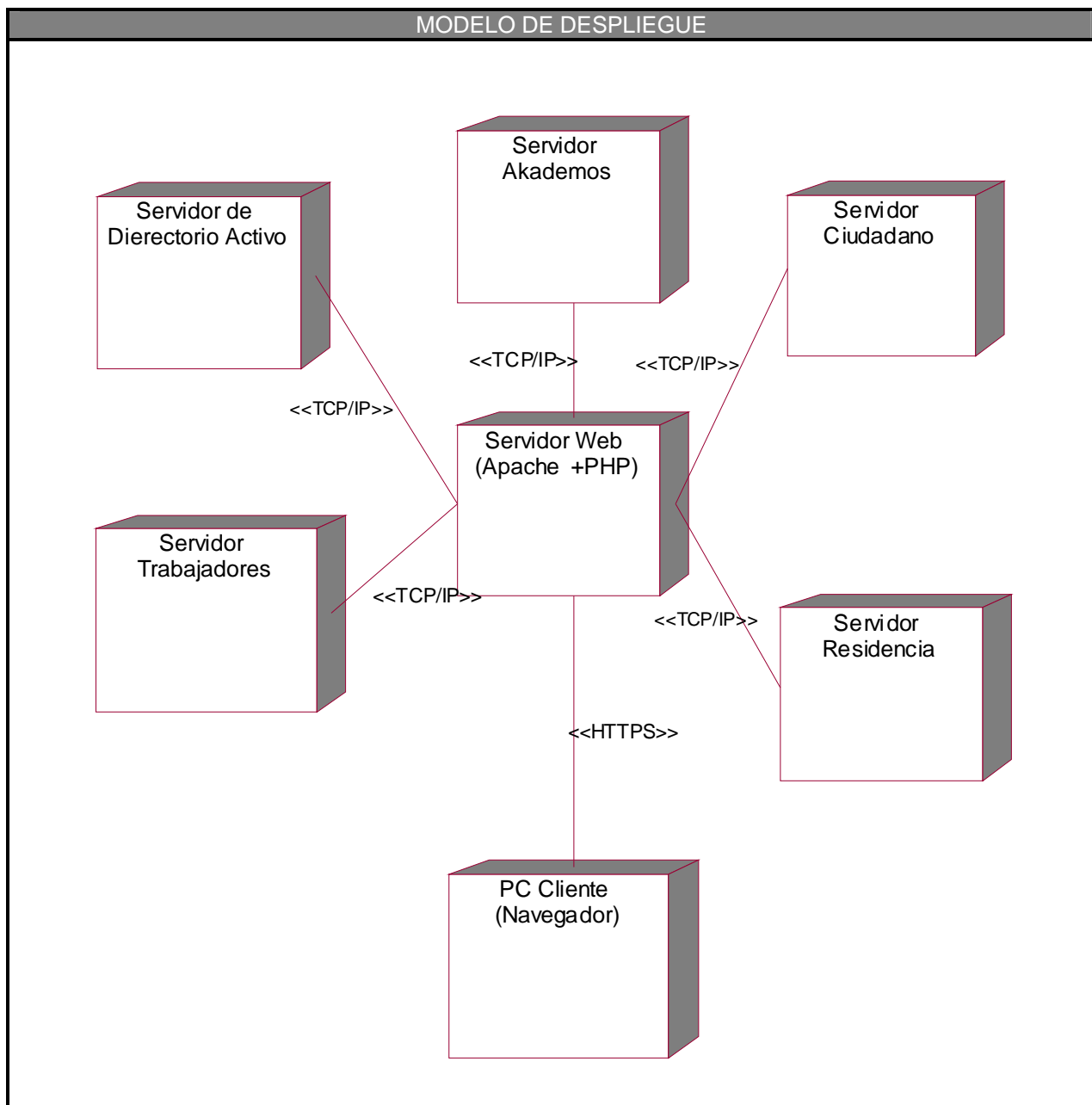


Fig. 4.8 Modelo de despliegue.

4.6 Modelo de implementación.

El modelo de implementación describe cómo se implementan los elementos del modelo de diseño. En la figura 4.9 se puede observar una vista general del modelo de implementación, en la que se puede observar una vista de los ficheros mas importantes en los que se ha dividido la aplicación y la interacción entre los mismos para lograr una mayor claridad y comprensión de la estructura del sistema.

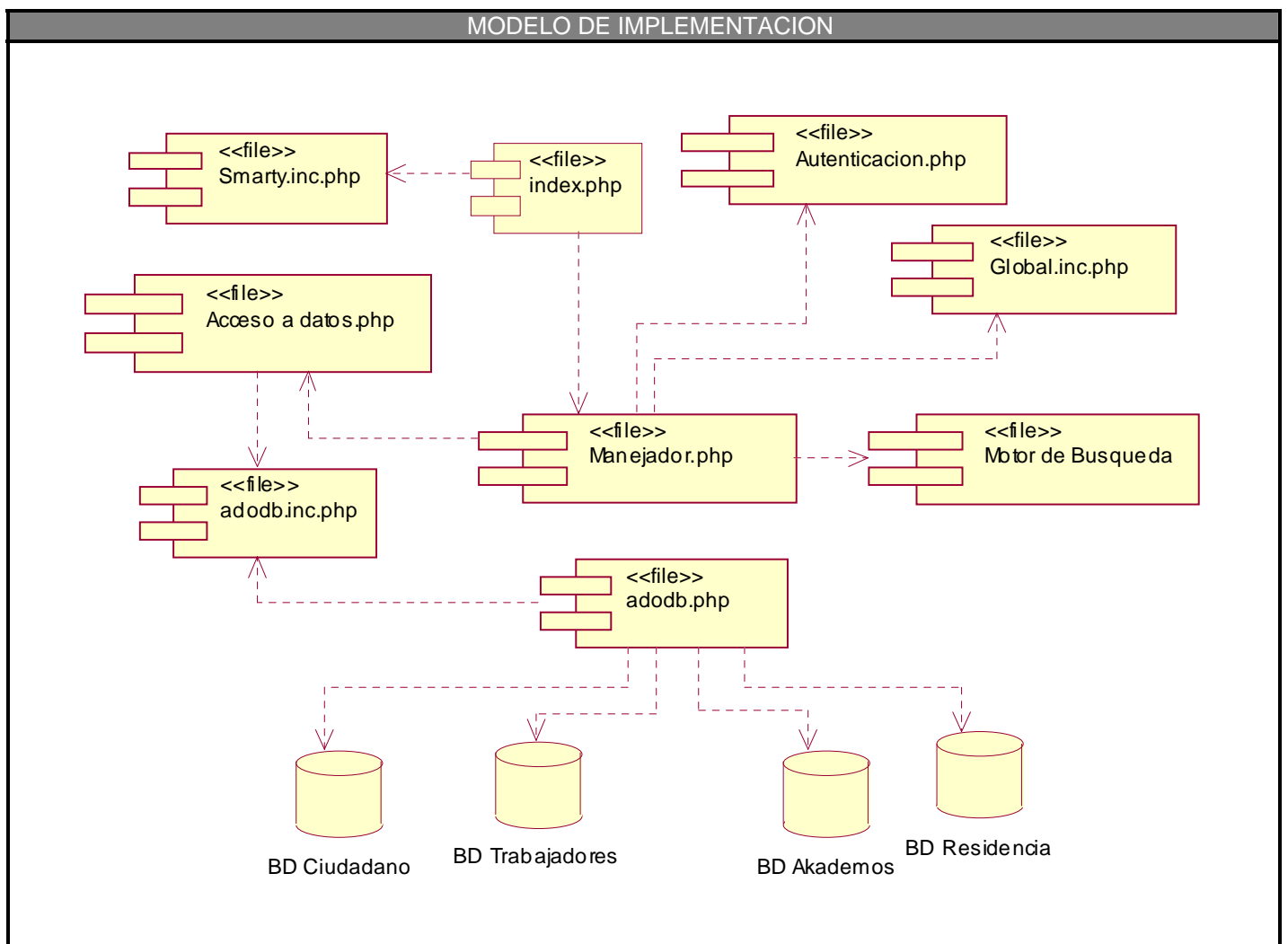


Fig. 4.9 Modelo de implementación.

4.7 Descripción preliminar del modelo de pruebas.

El modelo de pruebas es una colección de casos de prueba, procedimientos de prueba y componentes de prueba, que permite probar los componentes ejecutables en el modelo de implementación. Entre los casos de prueba se puede distinguir dos tipos comúnmente utilizados: las llamadas pruebas de “caja negra” y las de “caja blanca”.

Una prueba de “caja negra” es una prueba del comportamiento observable externamente del sistema, mientras que una prueba de “caja blanca” prueba la interacción interna entre los componentes del sistema.

En el caso del sistema InfoWeb, se propone la realización de una prueba de “caja negra” para cada caso de uso, para probar la interacción entre el usuario y el sistema, que se satisfagan las precondiciones y poscondiciones, y que se siga la secuencia de acciones intermedias especificadas por el caso de uso.

También se propone la realización de una prueba de “caja blanca” para cada realización de caso de uso.

4.8 Conclusiones.

En este capítulo se modeló detalladamente la solución propuesta, a través del modelo de diseño, de implementación y de despliegue. Se analizaron los principios de diseño de la interfaz de la aplicación, los estándares de codificación a emplear durante la fase de construcción y se hizo una breve descripción del modelo de pruebas que se propone realizar al sistema para comprobar que cumpla los requerimientos que le dieron origen.

CAPÍTULO 5. ESTUDIO DE FACTIBILIDAD.

5.1 Introducción.

En el presente capítulo se hace un análisis de los costos y los beneficios que generará el desarrollo de la aplicación y la relación entre ellos, con el objetivo de demostrar la factibilidad del proyecto, así como la conveniencia de su explotación en la universidad.

5.2 Estimación de costo.

Existen diversos modelos para estimar los costos de los proyectos de software. Entre ellos uno de los más aceptados internacionalmente en la actualidad es el modelo COCOMO v2 (del inglés Constructive Cost Model). Este modelo expresa el esfuerzo de desarrollo en términos de Personas Mes (cantidad de tiempo que una persona dedica a trabajar sobre el proyecto de desarrollo durante un mes). Finalmente se hace una estimación del costo monetario del proyecto teniendo en cuenta el tiempo estimado de desarrollo, la cantidad de personas involucradas y el salario de estas personas.

En el caso del modulo de búsqueda de InfoWeb, al ser una aplicación a desarrollar en el marco de la Universidad de las Ciencias Informáticas, no se hace necesario calcular el costo monetario, pues la elaboración del sistema no supone gastos adicionales por concepto de salario para la institución, ya que cuenta con el capital humano necesario (estudiantes y profesores vinculados al proyecto) y no necesita contratar otro personal. Por otra parte la universidad cuenta con la infraestructura tecnológica necesaria para llevar a cabo el proyecto.

Se realizó un estudio del esfuerzo necesario para el desarrollo del sistema (las funcionalidades principales), el cual se incluye en el ANEXO I, y que arrojó un resultado de 22 Personas Mes.

Posteriormente se hizo un análisis de los beneficios que aportará la explotación de este sistema en relación con los costos que pueda generar.

5.3 Análisis de costo / beneficio.

El sistema InfoWeb no requiere inversión en software pues todas las herramientas que se han empleado en su desarrollo, así como las librerías empleadas, son libres y de código abierto. Esto ha permitido incluso hacerles pequeñas mejoras para hacerlas más adaptables a las necesidades del producto. Por ejemplo, la capa de acceso a datos para PHP, ADODB, tiene implementadas una serie de funciones, pero fue necesaria la implementación de algunas nuevas que se le adicionaron y que ahora pueden ser utilizadas por otros desarrolladores.

El sistema está orientado al usuario, es de fácil aprendizaje, por lo que no reporta grandes gastos por concepto de entrenamiento.

Por todo ello se considera que es factible el desarrollo de la aplicación y que el esfuerzo de desarrollo (22 Personas Mes) está plenamente justificado.

5.4 Conclusiones.

Del estudio realizado en este capítulo, a partir del análisis de los beneficios, se concluye que es factible el desarrollo de esta aplicación, por todo el aporte social que genera.

CONCLUSIONES GENERALES

En este trabajo se demostró la necesidad de diseñar e implementar un sistema completamente nuevo que dé soporte a los procesos de búsqueda y recuperación de información relativa a las personas que viven y/o trabajan en la UCI.

Luego de un análisis de las tecnologías más usadas en la actualidad para la construcción de sistemas informáticos similares, se elaboró la propuesta de utilizar el lenguaje PHP, con su motor de plantillas: Smarty y su capa de acceso a datos: ADOdb, el gestor de bases de datos PostgreSQL y el servidor Web ApacheSSL para la construcción de la solución propuesta, junto a otras tecnologías como son el formato XML y las hojas de estilo CSS.

Se llegó a la conclusión también de que la metodología idónea para llevar a cabo el proceso de desarrollo es RUP (Proceso Unificado de Rational).

Se modeló el negocio propuesto, destacándose su superioridad con respecto al que existe en la actualidad, se identificaron los actores y trabajadores, así como las actividades que son objeto de automatización.

Se definieron los requerimientos del sistema, tanto funcionales como no funcionales, y posteriormente se estructuró el modelo de casos de uso del sistema, describiéndose cada caso de uso para una mejor comprensión de la funcionalidad que brindan.

Se diseñó el sistema, a través de diagramas de clases Web, el diagrama de clases persistentes, entre otros. Posteriormente se elaboró el modelo de despliegue y el de implementación.

Se plantearon los principios a seguir en el diseño de la interfaz de usuario y algunas convenciones a respetar durante la escritura del código fuente.

Finalmente, se analizaron los beneficios y costos que genera el sistema, se compararon y se concluyó que el proyecto es factible.

CONCLUSIONES

Luego de todo este proceso de trabajo se puede concluir que infoWEB es un sistema que da solución a la situación problemática que lo originó y que su explotación significará una mejora considerable en la calidad y eficiencia de los procesos que automatiza.

RECOMENDACIONES

Se recomienda el desarrollo de otros módulos para el buscador InfoWeb, como por ejemplo la implementación de la ficha del ciudadano, que se encargaría de gestionar la información que se muestra y donde cada usuario tendría permiso para ver la información que le esta permitida consultar según niveles de acceso definidos por la dirección de la Universidad. Es de nuestra consideración que esto seria un gran apoyo a todas las organizaciones y estructuras que necesiten consultar información del ciudadano UCI para desarrollar su trabajo y tomar decisiones lo mas acertadas posibles.

También se recomienda la construcción de un modulo de administración que se encargue de gestionar todos los procesos que desarrolla el sistema en general. Y como recomendación final pensamos que seria de gran utilidad diseñar e implementar un portal de búsquedas para la intranet universitaria donde se pudiera recuperar y mostrar cualquier recurso de información existente en la red, y que por supuesto su acceso sea publico e integrar el sistema infoWEb a este portal.

BIBLIOGRAFÍA

- [1] Álvarez Marañón, Gonzalo. *Web Seguro*. <http://www.iec.csic.es/criptonomicon/ssl.html> (15/6/05).
- [2] Soto López, Nilet María y Saborit Ramírez, Yunier. *Propuesta para un sistema de Catalogación y Recuperación de Recursos de Información*. Trabajo para optar por el título de Ingeniería en Informática, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de La Habana, junio 2004.
- [3] Bankhacker.com. *Web Services*. <http://web-services.bankhacker.com/> (2/7/05).
- [4] Wikipedia, la enciclopedia libre. *Sistemas Gestores de Bases de Datos*. <http://es.wikipedia.org/wiki/DBMS> (15/04/05).
- [5] Wikipedia, la enciclopedia libre. *Oracle*. <http://es.wikipedia.org/wiki/Oracle> (15/04/05).
- [6] González, Carlos D. *Curso: Integral de diseño. Programación de sitios dinámicos con MySQL y PHP*. http://www.usabilidadweb.com.ar/x_int.php (15/04/05).
- [7] Wikipedia, la enciclopedia libre. *MySQL*. <http://es.wikipedia.org/wiki/MySQL> (15/04/05).
- [8] González, Carlos D. *Curso: Sitios Web dinámicos con Base de Datos PostgreSQL y PHP*. <http://www.usabilidadweb.com.ar/postgre.php> (15/04/05).
- [9] Molpeceres, Alberto. *Procesos de desarrollo: RUP, XP y FDD*. <http://www.javahispano.org/articles.article.action?id=76> (15/04/05).
- [10] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Editorial Félix Varela, La Habana, 2004.
- [11] Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Editorial Félix Varela. La Habana, 2004.

GLOSARIO DE TÉRMINOS

A continuación, en orden alfabético, se muestra el significado de algunos términos usados en este documento cuyo uso no es común y que pueden dificultar la comprensión del mismo:

1. **Ciudadano UCI:** Se considera a la persona que estudia y/o vive o trabaja en la Universidad de las Ciencias Informaticas.
2. **Open Source:** Se le llama al software desarrollado bajo la filosofía de código abierto, es decir se pone el código a disposición de cualquier persona que quiera seguir desarrollándolo y mejorándolo.
3. **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.
4. **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.
5. **LDAP:** LDAP (*Lightweight Directory Access Protocol*) en sí es un protocolo de red que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP puede considerarse una base de datos (aunque su sistema de almacenamiento puede ser otro diferente) al que pueden realizarse consultas. Habitualmente, almacena la información de login (usuario y contraseña) y es

utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados...)En conclusión, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red.

6. **Proveedor de información:** Todos los sistemas que suministran información de l ciudadano a la aplicación.
7. **Ficha Personal:** Estructura digital en la que se muestra información detallada del ciudadano.

ANEXO I. CÁLCULO DE ESFUERZO DE DESARROLLO.

Tabla -1. Entradas externas.

Nombre de la entrada externa	Cantidad Ficheros	Cantidad Elementos Datos	Clasificación
Autenticación	1	2	Simple
Cadenas de Búsqueda	1	1	Simple
Búsqueda de Estudiantes	1	9	Simple
Búsqueda de Trabajadores	1	7	Simple

Tabla -2. Salidas externas.

Nombre de la salida externa	Cantidad Ficheros	Cantidad Elementos Datos	Clasificación
Reporte de Búsqueda	1	6	Simple
Ficha Personal	1	9	Simple

Tabla -3. Peticiones.

Nombre de la petición	Cantidad Ficheros	Cantidad Elementos Datos	Clasificación
Búsqueda de Estudiantes	1	3	Simple
Búsqueda de Trabajadores	1	1	Simple
Detalles de Estudiante	1	7	Simple
Detalles de Trabajador	1	9	Simple

Tabla -4. Ficheros lógicos internos.

Nombre del fichero interno	Cantidad Registros	Cantidad Elementos Datos	Clasificación
Autenticación	1	5	Simple
Motor de Búsqueda	1	2	Simple
adodb	1	3	Simple
Acceso a datos	1	5	Simple
Manejador	1	2	Simple

Tabla -5. Puntos de función desajustados.

Elementos	Simple		Media		Compleja		Subtotal puntos de función
	No	x Peso	No	x Peso	No	x Peso	
Ficheros lógicos internos	5	7	0	10	0	15	35
Entradas externas	4	3	0	4	0	6	12
Salidas externas	2	4	0	5	0	7	2
Peticiones	4	3	0	4	0	6	4
Total (UFP)							53

Tabla -6. Líneas de instrucciones fuentes.

Características	Valor
Puntos de función desajustados	53
Lenguaje(s)	PHP (80 %) SQL (20%)
Instrucciones fuentes por puntos de función	(10) (20)
Instrucciones fuentes por lenguaje (miles de instrucciones)	(3,102) (2,0313)
Instrucciones fuentes (miles de instrucciones)	5,1333

Tabla -7. Factores de escala.

Factor	Clasificación	Valor
PREC	Nominal	3,72
FLEX	Nominal	3,04
TEAM	Muy Alto	1,10
RESL	Nominal	4,24
PMAT	Bajo	6,24
Suma		18,34

Tabla -8. Multiplicadores de esfuerzo.

Multiplicador	Clasificación	Valor
RELY	Nominal	1,00
DATA	Alta	1,14
CPLX	Nominal	1,00
RUSE	Nominal	1,00
DOCU	Alta	1,11
PERS	Nominal	1,00
SCED	Nominal	1,00
Producto		1,25

La ecuación que plantea COCOMO para calcular el esfuerzo de desarrollo es la siguiente:

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

donde $E = B + 0.01 \times \sum_{j=1}^5 SF_j$

$$E = 0,91 + 0,01 * 18,34 = 1,0934$$

$$PM = 2,94 * (5,1333)^{1,0934} * 1,25 = 21.98 \approx 22 \text{ Personas - Mes}$$