Universidad de las Ciencias Informáticas

Título: Implementación del submódulo de Recuperaciones de Información del Módulo de Recuperaciones Dinámicas.

Trabajo de Diploma para optar por el título de Ingeniero Informático.

Autores:
Yasmany Montesino Afonso
Orestes Parra Lubín

Tutor:
Raydel Muñoz Vidal
Cotutor(a):
Yanet Pérez Valcarcel

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) y a la Universidad de 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 ____ de _____.

_____	_____	_____
Firma del Autor	Firma del Autor	Firma del Tutor

Pensamiento

"Si no chocamos contra la razón nunca llegaremos a nada."

Albert Einstein.

Agradecimientos

Agradecer a todos los que ayudaron a la realización de este trabajo, por su tiempo y dedicación.

A nuestros amigos y familiares, por su confianza y apoyo.

A nuestra universidad, a la revolución, y a nuestro comandante en jefe Fidel Castro por hacer posible nuestros sueños.

Agradecimientos especiales para el grupo del submódulo Configuración: Andy y Armando, a nuestro tutores Raydel, Yanet y Yosnier por su amplia colaboración.

Dedicatoria

A mis padres por su confianza y apoyo...
A mis abuelos y hermana...
A todos mis amigos.

Yasmany

Dedico este trabajo a mis padres, por su enorme consagración durante toda mi carrera.
A mi hermana por su apoyo y preocupación.
A mis amigos y a todo aquel que me extendido la mano para hacer posible este gran logro.

Orestes

Resumen

Nuestro país avanza con amplios pasos dentro del proceso de utilización ordenada y masiva de las tecnologías de la información y las comunicaciones. Factor decisivo para el desarrollo de nuestras empresas, economía y de la sociedad cubana. Proceso que busca lograr más eficacia y eficiencia, satisfacer las necesidades de información y conocimiento de todas las personas y esferas de nuestra sociedad.

El Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) también se sumerge en este proceso de información y desarrollo. Por el alto volumen de información que se procesa en dicho ministerio es necesaria la utilización de varias herramientas que permitan su manipulación eficiente y segura. Actualmente se presenta la necesidad de poder recuperar dinámicamente la información almacenada en los diferentes Sistemas de Gestión de Base de Datos (SGBD).

Este trabajo abarca la implementación de un sistema que permite realizar la Recuperación Dinámica de la Información (RDI) de los SGBD del MINFAR de forma integra, segura, y rápida de los datos que sean solicitados.

Índice

Introducción

En la actualidad existen diversos puntos de vistas en cuanto a la forma en que se debe procesar la información, por el desarrollo vertiginoso que ha ido experimentando la informática en la última década y los grandes volúmenes de datos que a diario deben ser consultados. Por tal desarrollo seria pertinente pensar que se vive un momento en la historia de la humanidad en el que la gestión de la información se ve posibilitada y juega un papel fundamental.

Por la relevancia que se le concede al acceso a la información y a la utilidad que se le puede dar a la misma para la toma de decisiones. Constantemente se trabaja para idear los mecanismos que permitan a la misma ser consultada, recuperada y analizada en tiempo real y sin depender de agentes externos.

Este trabajo en particular se centra en dar una solución informática a la Recuperación de la Información (RI), aspecto que juega un papel central en todo sistema de software, ya que permite el acceso objetivo y eficiente a los datos, posibilitando con la correcta aplicación de criterios organizacionales, una elegante manipulación de los datos para obtener en realidad la información que se desea consultar o la base de conocimiento que la misma puede generar.

La RDI da la posibilidad al usuario de elaborar sus propios reportes y esto posibilita la versatilidad al analizar y presentar la información de diferentes reportes sobre los mismos datos. Estos reportes pueden dar diversos puntos de vista y ayudar a tener un mayor entendimiento y posibilitar el análisis del recuperador.

La representación de la información juega un papel significante en el análisis de la misma. Para dar terminación a un sistema informático de RI se debe tener en cuenta como se representará. Es prácticamente un estándar en el caso que se consulta gran cantidad de información representarla mediante listados, donde el usuario puede ver exactamente que es lo que busca; en caso que se quiera dar cierto procesamiento a la información la misma es representada por estadística a través de tablas o gráficas donde el usuario pueda tener una mejor comprensión. Esta última forma de representar en ocasiones puede describir y prever determinados comportamientos.

El MINFAR, por la gran importancia que desde el punto de vista estratégico que se le confiere, no a dado la espalda al vertiginoso desarrollo del campo de la informática y se ha envuelto en la difícil tarea de informatizar la mayor cantidad de procesos que le sea posible para agilizar y mejorara la gestión de los mismos. Por la relevancia que en cualquier sistema informático se le confiere a las recuperaciones dinámicas el MINFAR tiene particular interés en contar con una herramienta que le permita obtener una eficaz representación de la información, lo cual mejoraría enormemente la comprensión de la información a la que se está accediendo, dándole al usuario los elementos para un análisis exhaustivo y al mismo tiempo simple.

Por el volumen de datos que por cuestiones obvias tiene almacenado el MINFAR, se ve en la necesidad de valerse de los adelantos tecnológicos que brinda la informática para almacenar, administrar y gestionar los mismos, la Universidad de las Ciencias Informáticas (UCI) trabajando en conjunto con el MINFAR se han trazado la tarea que nos ocupa, la implementación de un sistema que gestione de una manera rápida y eficaz el trabajo y la información que se maneja a diario, una herramienta altamente configurable y genérica que le permita al usuario personalizar según sus necesidades los reportes que desee generar.

En estos momentos no se cuenta en dicha institución con un sistema potente de RI ya que los que existen son específicos y no brindan una buena representación de la información. Lo que lleva al MINFAR a necesitar un sistema de RDI, altamente configurable y que presente la información de una manera elegante y eficiente.

Por lo antes expuesto el **problema** a resolver queda planteado de la siguiente manera:

La no existencia de un mecanismo para generalizar la recuperación de la información existente en una base de datos, teniendo en cuenta restricciones establecidas por los usuarios.

El **objeto de estudio** lo constituyen las formas de recuperación, por listados y estadísticas, de la información existente en una base de datos.

El **campo de acción** engloba los procesos de recuperación, por listados y estadísticas, de la información almacenada en las bases de datos del cliente.

Como **Hipótesis** se plantea que, si se tuviese un sistema de RDI, altamente configurable y genérico que pudiese brindar en listados, tablas y gráficos la información solicitada por el usuario, se realizará teniendo en cuenta restricciones establecidas, la recuperación de la información existente en una base de datos.

Con el propósito de dar solución al problema anteriormente planteado se ha trazado como **objetivo general**: Implementar un modelo que garantice generalizar las recuperaciones, por listados y por estadísticas, de la información existentes en las bases de datos del cliente.

De acuerdo con esta propuesta se derivan los siguientes **objetivos específicos**:

- Realizar un estudio del diseño de los diferentes casos de uso que conforman el paquete de recuperación.
- Realizar un estudio sobre el lenguaje de programación para la implementación, y ofrecer ventajas y desventajas del mismo.
- Realizar un estudio sobre la arquitectura de desarrollo, así como una detallada descripción de la que se emplee finalmente.
- Realizar un estudio sobre Grafos.
- Realizar un estudio de las diferentes tecnologías a utilizar para el desarrollo del sistema.

Para cumplir con estos objetivos y resolver la situación polémica planteada, se proponen las siguientes **tareas**:

- Análisis del estado del arte de las tecnologías que se utilizan para llevar a cabo herramientas como la que se pretende desarrollar.
- Selección de las herramientas para llevar a cabo la realización del proyecto, así como la elección de la plataforma en la que se desarrollará la aplicación.
- Aplicación de la arquitectura, en el proceso de desarrollo del sistema.
- Implementar una estructura capaz de modelar el funcionamiento de un grafo para nuestro sistema.

Posibles resultados:

El submódulo de Recuperaciones de Información del Módulo de Recuperaciones Dinámicas, está desarrollado en un ambiente Web y está diseñado de manera tal que optimiza lo referente a RI. Esta herramienta brinda la posibilidad de hacer consultas a la información almacenada de forma dinámica.

A continuación se da una breve explicación de la estructuración del contenido de la presente investigación:

El **Capítulo I** titulado "Fundamentación teórica" ofrece los conceptos básicos asociados al dominio del problema. Además brinda el estado del arte en cuanto a los antecedentes históricos del sistema, así como las técnicas, tecnologías, metodologías y software usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta.

El **Capítulo II** denominado "Descripción y análisis de la solución propuesta": se plantea una valoración crítica del diseño propuesto por el analista, un análisis de posibles implementaciones de componentes o módulos ya existentes que puedan ser rehusados y las estrategias de integración, una descripción de los algoritmos no triviales a implementar. Análisis de complejidad de los mismos y selección de las estructuras de datos apropiadas para la implementación de estos algoritmos, finalizando con la descripción de las nuevas clases u operaciones necesarias.

En el **Capítulo III** "Validación de la solución propuesta": se muestran las pruebas realizadas para validar dicha solución y se brindan una serie de recomendaciones a tener en cuenta en futuras iteraciones.

1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se esbozan de forma general las características de algunos sistemas informáticos de recuperaciones dinámicas de información, se valoran y comparan con la solución que en el trabajo se plantea. Así como también se analiza el estado en que se encuentran algunas tecnologías de programación, paradigmas y plataformas de desarrollo en el mundo y nuestro país. Además se realiza un estudio comparativo entre las tecnologías que se ajustan a los requerimientos que la confección del producto final requiere con el objetivo de seleccionar una combinación de las mismas: lo cual constituirá el entorno de trabajo que se utilizará para brindar una solución.

1. 2 Recuperación de información

La integridad y robustez de un sistema informático puede estar dada por la calidad de: la recuperación, fácil acceso y versatilidad que el mismo implemente y brinde al usuario. La terminología Recuperación de la Información (RI) engloba los procesos de representación, organización, almacenamiento, algoritmos de búsqueda y acceso a la información.

La principal tarea de la RI es la de permitir al usuario acceder a la información que le responda a sus necesidades, de la manera más eficaz en el momento real que él mismo necesite. Existen varias formas de recuperar y representar la información y en cada caso se recomienda utilizar por el cliente la que pueda saciar sus expectativas.

Recuperación por listados: maneja gran volumen de información y brinda un estado completo de la información haciendo en ocasiones el trabajo con la misma engorroso y de difícil análisis para una toma de decisión, ya que el usuario se abruma ante tanto volumen de datos y se le dificulta encontrar lo que en realidad responde a su interrogante, si bien es cierto que en ocasiones una buena presentación y filtrado de la información pueden hacer más amigable el entendimiento de la misma, es decir, no mostrar una

amalgama de datos que le pueda resultar incoherente al usuario. También es una realidad que en ocasiones por la relevancia de la misma se hace preciso mostrarla íntegramente.
Recuperación por estadísticas: al igual que la recuperación por listados maneja gran volumen de información pero su principal diferencia y mérito radica en la forma que presenta la misma, como se conoce se usa a diario las estadísticas como una manera eficiente y elegante de presentar grandes volúmenes de información, basándose en determinados criterios se puede presentar la misma al usuario de una manera elegante y resumida en la cual el usuario puede analizar el comportamiento y tener una mejor comprensión de la información que solicita.

Recuperaciones por gráfico: presenta gran similitud con las recuperaciones por estadísticas en cuanto al volumen de información que representa ya que se centra en graficar comportamientos de determinados datos para de esa manera darle al usuario final una aproximación real del comportamiento que se grafique, brinda una interfaz amigable que compuesta de una gama de graficas de pastel, barras líneas intermitentes y áreas de ocurrencia proporciona un apoyo en la toma de decisiones.

1.3 Recuperaciones de información en nuestro país

A pesar de que nuestro país prácticamente comienza a enfrentarse en el mundo de la informática, en la mayoría de los sistemas de gestión que se han desarrollado en los últimos años, de una manera u otra se ha trazado como objetivo que dichas implementaciones brinden una forma de RI.

Luego de investigar se determinó la no existencia de un modelo que sintetizara la RI.

1.4 Sistemas internacionales

- **Microsoft Query.**

Microsoft Query es un programa que permite incorporar datos de orígenes externos a otros programas de Microsoft Office, especialmente a Microsoft Excel. Es posible recuperar datos de varios tipos de bases de datos, incluidos Microsoft Access, Microsoft SQL Server y los servicios OLAP de Microsoft SQL Server.

También puede recuperar datos de las listas de Excel y de archivos de texto. Puede recuperar datos de una base de datos creando una consulta, que es una pregunta que se hace acerca de los datos almacenados en una base de datos externa.

- **Crystal Report.**

Crystal Reports es una aplicación de inteligencia empresarial, o bien, inteligencia de negocio, utilizada para diseñar y generar reportes desde una amplia gama de fuentes de datos. Provee más opciones de conectividad a datos que cualquier otra herramienta. Incluye más de 30 drivers para acceso a bases de datos relacionales, fuentes de datos XML y cubos OLAP (Incluyendo sistemas ERP, CRM, Oracle, IBM DB2 y Microsoft SQL Server). También puede acceder a datos personalizados a través de JavaBeans y objetos COM (ADO record sets) para una conectividad más flexible.

- **Agata Report.**

Agata Report es un conjunto de herramientas para sacar lo más sustantivo de las bases de datos. Parecido a Crystal Report pero en versión libre, permite extraer datos de PostgreSQL, MySQL, Oracle, DB2, MS-SQL, Informix, InterBase, Sybase, o Frontbase y exportarlas a PostScript, texte, HTML, XML, PDF, o CSV.

Entre otras funcionalidades interesantes, Agata Report da la posibilidad de generar un completo diagrama Entidad-Relación, proporciona un módulo para crear gráficas en curvas o en bastones y permite cruzar informes con otras bases de datos. Además, el programa soporta parámetros temporales de ejecución (consulta estos valores antes de correr la aplicación).

Es una herramienta estable y adaptada a las necesidades de aquellos que manipulan datos con frecuencia. Con una interfaz intuitiva y disponible en inglés, portugués, español, italiano, francés, alemán y sueco, permite su uso de cualquier persona con unas pocas nociones de SQL.
Bajo licencia GNU, Agata Report funciona tanto para Linux como para Microsoft Windows.

- **Jasper Report**

Jasper Reports es una poderosa herramienta para realizar reportes que tiene como función el llevar documentos ricos en contenido a la pantalla, a la impresora, o a archivos PDF, HTML, XLS, CSV y XML. Es una librería de clases de Java de código abierto que está diseñada para facilitar el agregar capacidades de reporte a las aplicaciones Java.

Permite una diagramación flexible de los reportes, pueden dividirse en secciones opcionales que son: titulo del reporte, el encabezado de página, una sección para los detalles del reporte, el pie de página y una sección de resumen que aparece al final del reporte. Jasper Reports permite generar textos o imágenes de fondo para utilizarlo como marcas de agua con el propósito de identificar el reporte o simplemente por motivos de seguridad.

1.5 Sistemas nacionales

- **SIPRID**

Sistema Informático para la Recuperación de Información Docente. Sistema implementado que permite la gestión de los principales documentos del proceso docente de la Universidad de Matanzas “Camilo Cienfuegos”, Facultad de Informática.

- **SISMED**

Sistema automatizado de información sobre medicamentos para el sector de salud cubano, desarrollado conjuntamente por el CINFA (Centro de Información Farmacéutica) y la Empresa Productora y Comercializadora de Software, cuya integralidad lo convierte en una alternativa informativa de gran vigencia para el sector de salud, se comenzó a introducir en el territorio nacional desde el año 1996.

1.6 Nuestra propuesta

Nuestra propuesta es un modelo que generaliza la RI en cualquier base de datos, el cual en esta primera iteración y a solicitud del cliente enmarcará su alcance a las recuperaciones en bases de datos Postgres, se basa en el establecimiento de una base de datos de configuración en la que el sistema contará con un mapeo estructural de la base de datos a la que se le realizará la recuperación, lo que permite un nivel de abstracción al elaborar las restricciones bajo la que se hará la recuperación, estas restricciones son : las condiciones determinadas por el usuario.

1.7 Tecnologías

1.7.1 Tecnología Cliente – Servidor.

Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o a través de la organización, en múltiples plataformas. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes. Este esquema facilita la integración entre sistemas diferentes y comparte información permitiendo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera, se integra PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.

1.8 Técnicas de programación

1.8.1 Programación estructurada.

Este tipo de programación está basada en una secuencia de instrucciones modificando los datos globales en el transcurso de todo el programa. Donde los saltos y el fin de programa no siguen ninguna estructura. Los saltos pueden apuntar a cualquier punto del código, lo que ocasiona que el algoritmo termine siendo un ovillo indescifrable. Tampoco se puede saber cuándo terminaba.

1.8.2 Programación procedimental.

La programación procedimental es un tipo de programación estructurada en donde el código se divide en porciones llamadas "procedimientos" o "funciones".

1.8.3 Programación modular.

La programación modular está basada en la técnica de diseño descendente, consiste en dividir el problema original en diversos sub-problemas que se pueden resolver por separado, para después recomponer los resultados y obtener la solución al problema.

1.8.4 Programación orientada a objetos.

La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (datos), comportamiento (procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulo más fáciles de escribir, mantener y reutilizar. De esta forma, un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos

pertenecientes a otras clases (e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos).

A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos (de una misma clase o de distintas), y en consecuencia, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan (ni deben separarse) información (datos) y procesamiento (métodos).

1.9 Lenguajes y plataformas de desarrollo.

1.9.1 Lenguajes

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Un lenguaje de programación permite a uno o más programadores especificar de manera precisa: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados, transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural.

1.10 Lenguajes del lado del servidor.

Se les clasifica así a los lenguajes de programación en la tecnología cliente servidor que se ejecutan del lado del servidor y de los que los cuales los usuarios solo obtienen el beneficio del procesamiento de la información.

1.10.1 C#

C# es uno de los lenguajes base de su plataforma .NET. Este lenguaje tiene como característica fundamental el hecho de ser muy versátil en su uso y eficiente en su aplicación conformando un lenguaje

que reúne las mejores características de los lenguajes más utilizados, agrupándolas en un solo lenguaje mejorado: C#. Las ***ventajas*** que ofrece C# frente a otros leguajes de programación son:

- *Declaraciones en el espacio de nombres:* al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.
- *Tipos de datos:* en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- *Atributos:* cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.
- *Pase de parámetros:* aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada ref, la cual indica que el pase es por referencia.
- *Métodos virtuales y redefiniciones:* antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra override.
- *Propiedades:* un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.
- *Inicializador:* un inicializador es como una propiedad, con la diferencia de que en lugar de un nombre de propiedad, un valor de índice entre corchetes se utiliza en forma anónima para hacer referencia al miembro de una clase.
- *Control de versiones:* C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

Las ***desventajas*** que se derivan del uso de este lenguaje de programación son que en primer lugar se tiene que conseguir una versión reciente de Visual Studio .NET, por otra parte se tiene que tener algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como contar con Windows NT 4 o superior, tener alrededor de 4 gigas de espacio libre para la pura instalación. Para quien no está familiarizado con ningún lenguaje de programación, le costará más trabajo iniciarse en su uso, y si se

quiere consultar algún tutorial más explícito sobre la programación en C# tendría que contar además con una conexión a Internet.

1.10.2 JAVA

Java es un lenguaje de programación con el se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si se hace un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina Virtual de Java para cada sistema, que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente. La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

1.10.3 PHP

PHP: lenguaje de programación de scripts, usado para la programación de páginas Web dinámicas. Es un lenguaje interpretado, que soporta la orientación a objeto, clases y herencia. Es posible crear aplicaciones con una interfaz gráfica para los usuarios. PHP es multiplataforma, corre en casi cualquier plataforma utilizando el mismo código fuente, puede interactuar con muchos motores de bases de datos tales como MySQL, Postgres y Oracle; soporta ODBC lo que lo hace compatible con l a mayoría de los SGBD existentes, también puede interactuar con los servidores Web más populares, generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en

C, así que se ejecuta rápidamente utilizando poca memoria. También es código abierto. Es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Desde su surgimiento en 1994 cuando fue concebido por Rasmus Lerdorf a evolucionado hasta convertirse en una poderosa herramienta para los desarrolladores, ya que al contar con un grupo de desarrolladores tan extenso se mantiene al día en la compatibilidad tecnológica.

1.11 Lenguajes del lado del cliente.

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Pero nuestra página no se verá bien si el ordenador cliente no tiene instalados los plug-in adecuados. El código, tanto del hipertexto como de los scripts, es accesible a cualquiera y ello puede afectar a la seguridad.

1.11.1 HTML

HTML es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Marcas Hipertextuales. Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Gracias a Internet y a los navegadores como Internet Explorer, Opera, Firefox, Netscape o Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para Web.

1.11.2 XML

XML, sigla en inglés de eXtensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos (de la misma manera que HTML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos,

editores de texto, hojas de cálculo y casi cualquier cosa imaginable.XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato. El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan bugs y se acelera el desarrollo de la aplicación. Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

1.11.3 JavaScript

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del DOM. Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página Web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript se puede crear diferentes efectos e interactuar con nuestros usuarios. Javascript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.

1.11.4 Tecnología AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. AJAX es una combinación de tecnologías ya existentes como son: HTML y hojas de estilos CSS para generar estilos, implementaciones ECMAScript (uno de ellos es el

lenguaje Javascript) y XMLHttpRequest que es una de las funciones más importantes que incluye, esta permite intercambiar datos asincrónicamente con el servidor Web, puede ser mediante PHP, ASP, JSP entre otros.

1.12 Control de versiones

Una versión, revisión o edición de un producto, es el estado en el se encuentra en un momento dado en su desarrollo o modificación. Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas. Un sistema de control de versiones debe proporcionar un mecanismo de almacenaje de los elementos que deba gestionar y un registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente brindando la posibilidad de volver o extraer un estado anterior del producto) entre otros aspectos.

Todos los sistemas de control de versiones se basan en disponer de un repositorio, que es el conjunto de información gestionada por el sistema. Este repositorio contiene el historial de versiones de todos los elementos gestionados. Cada uno de los usuarios puede crearse una copia local duplicando el contenido del repositorio para permitir su uso. Es posible duplicar la última versión o cualquier versión almacenada en el historial.

1.12.1 Subversion

Subversion es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

1.13 Navegadores

Un navegador Web o explorador Web es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Internet. Cualquier navegador actual permite mostrar o ejecutar gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

1.13.1 MOZILLA FIREFOX

Mozilla Firefox es un navegador, con interfaz gráfica de usuario desarrollado por la Corporación Mozilla. El programa es multiplataforma y está disponible en versiones para Microsoft Windows, Mac OS X y GNU/Linux. El código fuente de Firefox está disponible libremente bajo la **triple licencia** de Mozilla como un programa libre y de código abierto. Firefox incorpora bloqueo de ventanas emergentes, navegación por pestañas, marcadores dinámicos, compatibilidad con estándares abiertos, y un mecanismo para añadir funciones mediante extensiones.

1.14 IDE de desarrollo

Un entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE') es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Delphi, Visual Basic, Object Pascal, Velneo, etc. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, que mediante pluggins se le puede añadir soporte de lenguajes adicionales.

1.14.1 ECLIPSE

Es una plataforma de software de código abierto independiente de la plataforma en la que se desarrolla lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos integrados de desarrollo (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente. Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++, Python y PHP, permite a Eclipse trabajar con sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente. En cuanto a las aplicaciones clientes, eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones Web. Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre.

1.14.2 Zend Studio

Zend Studio es una herramienta software profesional excelente para trabajar proyectos PHP. Es concebido con el fin de crear aplicaciones altamente fiables, proporciona una facilidad de uso inigualable, escalabilidad, fiabilidad, y la extensión que los programadores profesionales y de empresas requieren para desarrollar, distribuir, depurar y administrar aplicaciones PHP críticas de negocios. Permite conectarse directamente con la bases de datos profesionales más utilizadas tales como IBM DB2/Cloudscape/ Derby/, MySQL, Oracle, Microsoft SQL Server, PostgreSQL y SQlite. Tiene características de depuración avanzadas, incluyendo: condiciones límites, visualización de errores, vistas avanzadas, variables y buffer de salida. Asegura la protección máxima de ubicaciones de proyectos o en Internet con depuradores

remotos. Facilita el desarrollo y colaboración en equipo mediante la administración efectiva de su código fuente utilizando CVS o Subversión directamente desde Zend Studio. Tiene soporte para PHP 5 completo, analizador de código, carpeta de código, completado de código, coloreado de sintaxis, administrador de proyecto, editor de código, depurador de gráficos y asistentes.

1.15 Gestor de base de datos

MySQL y PostgreSQL son, sin duda, las dos bases de datos libres más empleadas.

1.15.1 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones y con la ventaja ser *software libre, varios* lenguajes de programación que pueden usar este gestor de datos entre ellos (C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), FreeBASIC, y Tcl), cada uno de ellos utiliza un API específica, también cuenta con una interfaz ODBC, llamado MyODBC permitiendo así a cualquier lenguaje que soporte ODBC comunicarse con este SGBD.

1.15.2 PostgreSQL

PostgreSQL es un servidor libre, liberado bajo la licencia BSD. Que ofrece un potencial adicional al incorporar cuatro conceptos básicos mediante los cuales los usuarios pueden extender fácilmente los sistemas: clases, herencia, tipos y funciones. Nativamente provee soporte para: números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC y Arrays. A lo largo de su desarrollo se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers), se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias,

identificadores entrecomillados, forzado de tipos cadena literal, conversión de tipos y entrada de enteros binarios y hexadecimales. Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales. La velocidad del código del motor de datos ha sido incrementada aproximadamente en 40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fuera lanzada, otra de las ventajas incorporada de Postgres es el bloqueo de registros. Algunos de los lenguajes que se pueden usar con el PostgreSQL son los siguientes: un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de Oracle), C, C++, Gambas, Java PL/Java Web, PL/Perl, plPHP, PL/Python, PL/Ruby,

1.16 Conclusiones

Condicionado por las características del polo productivo y la investigación sobre el estado del arte de las tecnologías, paradigmas y conceptos en los que se basará la solución que se propone, se ha determinado seleccionar las siguientes herramientas:

Como ID de desarrollo **Eclipse** por la flexibilidad que supone su tecnología de plugin que lo hace extensible a prácticamente cualquier lenguaje y lo amigable de su área de trabajo. Para mantener un control de versiones se seleccionó **Subversion** dada la superioridad que supone ante la alternativa de **CSV** y por el ID de desarrollo seleccionado contar con un plugin que brinda un interfaz potente y amigable para interactuar con el mismo. Como gestor de bases de datos **PostgreSQL** por ser de los estudiados el que brinda mayores prestaciones y ser el utilizado en el polo productivo. Todas las herramientas seleccionadas para el desarrollo de la aplicación están bajo licencias Open Source.

CAPÍTULO 2: ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2.1 Valoración crítica del diseño propuesto en el análisis

Del diseño propuesto por los analistas se pueden identificar las clases y las funcionalidades a implementar para que el sistema funcione correctamente, basándose en la descripción detallada de los casos de uso, se puede establecer una estrategia de trabajo que permita la implementación de la capa lógica de la aplicación. Tomando como base la propuesta de diseño de interfaz, cabe puntualizar que la misma no cumplía con las exigencias del cliente ya que fraccionaba el área de trabajo y hacia un poco engorroso el trabajo al definir las condiciones.

2.2 Análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser rehusados

2.2.1 Ext

Ext.: es un framework JavaScript del lado del cliente para el desarrollo de aplicaciones Web. Tiene un sistema dual de licencia: Comercial y Open Source. Este framewok puede correr en cualquier plataforma que pueda procesar POST y devolver datos estructurados (PHP, Java, .NET y algunas otras) en tiempo de ejecución carga y crea todos los objetos html a través del uso intenso de DOM. Los datos son obtenidos mediante mensajes AJAX a través de XML y/o JSON. El diseño está completamente separado de la funcionalidad. Presenta funciones comunes como validación, comboxes editables, ventanas arrastables y grillas editables. Se determinó utilizar para la implementación de la interfaz de usuario este framework en su versión estable Ext.-2.0, por las facilidades que desde el punto de vista del diseño brinda y lo amigable que resulta la interfaz resultante para el usuario final.

2.2.2 JSON

JSON: acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La

simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador semántico de JSON.

2.2.3 Arquitectura

La arquitectura que se empleará para el desarrollo del sistema es una Arquitectura de tres capas, la cual está definida en el polo productivo, la misma proporciona una magnifica organización y estructuración entre los diferentes niveles de abstracción, ya que un cambio en uno de estos niveles no traerá modificaciones significativas en los restantes. Este modelo arquitectónico es el más utilizado en la actualidad sobre todo por el nivel de adaptabilidad que supone.

- **Capa de presentación:** es la capa mediante la cual el sistema brinda al usuario una interfaz con la que el mismo puede interactuar intercambiando información, pues su funcionamiento básico enmarca los procesos de mostrar información y capturar la misma para su posterior procesamiento, en ella se realiza una primera validación con el objetivo de evitar que se introduzcan valor no deseados por el cliente que puedan peral errores en la lógica, es importante destacar que dicha capa solo se comunica con la Lógica de Negocio con la que realiza el intercambio de información.

- **Capa de Lógica de negocio:** es la capa más importante ya que es ella la que se encarga de procesar la parte lógica, recibe las peticiones de la capa de presentación y envía las respuestas a la misma, es en ella donde se establecen todas las reglas que deben cumplirse y se manipula toda la información. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y mostrar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él, para el acceso a datos esta capa se relaciona con una clase que implementa la interfaz del modelo de persistencia que responde a un patrón llamado Factory la cual es una puerta de enlace entre la capa de Acceso a Datos y la capa de Lógica de Negocio.

- **Capa de Acceso a datos:** es la encargada de manejar todo el flujo de información que entra y sale de la fuente de datos y la conexión con la misma. En ella se implementa una interfaz que se encarga de establecer un enlace con la lógica de negocio, como es la encargada de establecer la conexión con la base de datos implementa un patrón de diseño llamado Singlenton cuyo objeto es restringir la creación de objetos pertenecientes a una clase logrando así que una clase solo tenga una instancia.

2.3 Principales clases generales a utilizar

Breve descripción de las clases en la arquitectura propuesta

Factoría Típica: Clase que implementa la interfaz del modelo de persistencia con el resto de los subsistemas. A través de esta clase se crean y se manipulan los objetos de las típicas simples, los nomencladores y las demás típicas. Es una puerta entre la capa de Acceso a Datos y la capa de Lógica de Negocio.

Típicas*: C*lase que representa a las clases típicas de la aplicación, por cada entidad de la base de datos existe una clase típica que implementa generalmente un método de Validación, Insertar, Modificar y Eliminar. Para la implementación de esta clase se decidió aplicar el patrón de diseño Table Data Gateway, que consiste en crear una instancia por cada tabla existente en la BD. Sus métodos consisten en las operaciones básicas que se realizan sobre estas tablas, insertar, modificar y eliminar. Hereda de la clase abstracta meBase.

Típica Simple*:* Es una clase que representa a las clases típicas (nomencladores simples) en general de la aplicación. Estas típicas son de una implementación muy sencilla, pues la mayoría de las líneas que normalmente habían que codificar quedaron encapsuladas en la clase base de las mismas. Para la implementación de esta clase se decidió aplicar el patrón de diseño Table Data Gateway, que consiste en crear una instancia por cada tabla existente en la BD. Sus métodos consisten en las operaciones básicas que se realizan sobre estas tablas, insertar, modificar y eliminar. Hereda de la clase abstracta meSimple.

cClaseconsulta*: C*lase que representa a las clases de la aplicación. Existe una clase consulta para cada entidad de la base de datos, estas heredan de la clase abstracta consBase.

meSimple: Clase abstracta heredera de meBase, y la vez base para la implementación de las típicas que responderán a los nomencladores simples del modelo de persistencia dado. Redefine las operaciones básicas con la funcionalidad de Validación dada. Redefine las operaciones básicas que pudieran realizarse a una entidad (insertar, eliminar, modificar) para los nomencladores simples.

meBase: Clase abstracta que hereda de consBase, es la base para el resto de las que implementan funcionalidades para el trabajo con las entidades del sistema a implementar. Implementa las operaciones básicas que pudieran realizarse a una entidad (insertar, eliminar y modificar). Y hereda de consBase la operación de Consulta.

consBase: Esta clase es la base en toda la jerarquía de Acceso a Datos y es empleada para aportar contenido dinámico a las plantillas. Encapsula el objeto conexión. Implementa la operación de Consulta.

Conexión: Esta clase es la encargada de establecer la conexión con el servidor de la base de datos a través de un objeto PDO de la librería de PHP.

PDO*:* Es un modelo de acceso a datos para php que brinda una capa de abstracción para el acceso a Base de Datos desde PHP.

2.4 Bases de datos relacionales

Una **base de datos relacional** es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos poseen las mismas características, a este campo generalmente se le denomina, identificador o llave. A esta manera de construir bases de datos se le denomina **modelo relacional**.

2.4.1 Características

Una base de datos relacional es una base de datos que cumple con el modelo relacional, y se refiere a una base de datos y base de datos de esquema lógico (la estructura de la base de datos). Estrictamente, una base de datos relacional, es un conjunto de relaciones (frecuentemente llamadas tablas). Cada tabla a su vez es un conjunto de registros, filas o tuplas. Y cada una de éstas es un conjunto de campos, columnas o atributos. Adicionalmente todas las filas poseen el mismo número de campos y el mismo campo, sin importar la fila a la que pertenece, debe cumplir con el dominio del campo que se encuentra definido en el conjunto de requerimientos.

2.5 Optimización de consultas

Existen varias técnicas con las que los SGBD procesan, optimizan y ejecutan las consultas. Las consultas son analizadas por un analizador léxico que identifica los componentes del lenguaje. Después un análisis sintáctico que revisa la sintaxis y por último la consulta debe ser validada para lo que ha de comprobarse que los nombres de las relaciones, los atributos son válidos.

Los SGBD crean una representación interna de las consultas, por lo regular en forma de árbol o grafo (árbol o grafo de consultas). Lo siguiente que debe hacer el SGBD es crear una estrategia de ejecución para obtener el resultado de la consulta a partir de los archivos internos. El proceso de elegir la alternativa más adecuada para procesar una consulta se denomina optimización de consultas.

El módulo optimizador de consultas se encarga de producir un plan de ejecución y el generador de código genera el código necesario para ejecutarlo. El procesador de base de datos en tiempo de ejecución se encarga de ejecutar el código de la consulta.

El término optimización no es del todo correcto ya que en algunos casos, el plan de ejecución elegido no es la estrategia óptima (la mejor) es tan sólo una estrategia razonablemente eficiente. En general, encontrar la mejor solución consume demasiado tiempo.

2.5.1 Estrategias para la optimización

Existen técnicas que mejoran los tiempos de respuesta de un SGBD relacional, teniendo en cuenta que la complejidad de algunas consultas puedan tomar un tiempo considerable, obteniendo no siempre una respuesta óptima. Muchas veces no basta con especificar una sentencia SQL correcta, sino que además, hay que indicarle como tiene que hacerlo si se quiere que el tiempo de respuesta sea el mínimo. Algunas de las formas de mejorar el tiempo de respuesta de nuestro intérprete son:

- **Gestión y elección de los índices**

Los índices son campos elegidos arbitrariamente por el constructor de la base de datos que permiten la búsqueda a partir de dicho campo a una velocidad notablemente superior. Sin embargo, esta ventaja se ve contrarrestada por el hecho de ocupar mucha más memoria (el doble más o menos) y de requerir para su inserción y actualización un tiempo de proceso superior.
Un caso en el que los índices pueden resultar muy útiles es cuando se realiza peticiones simultáneas sobre varias tablas. En este caso, el proceso de selección puede acelerarse sensiblemente si se indexa los campos que sirven de nexo entre las dos tablas.

- **Diseño de las tablas.**

1. Normaliza las tablas, al menos hasta la tercera forma normal, para asegurar que no hay duplicidad de datos y se aprovecha al máximo el almacenamiento en las tablas.

2. Los primeros campos de cada tabla deben ser aquellos campos requeridos y dentro de los requeridos primero se definen los de longitud fija y después los de longitud variable.

3. Ajusta al máximo el tamaño de los campos para no desperdiciar espacio.

- **Campos a Seleccionar.**

1. Evitar que las sentencias SQL estén embebidas dentro del código de la aplicación. Es mucho más eficaz usar vistas o procedimientos almacenados porque el gestor los guarda compilados. Si se trata de una sentencia embebida el gestor debe compilarla antes de ejecutarla.

2. Seleccionar exclusivamente aquellos elementos que se necesiten.

3. Utilizar la referencia del esquema y la tabla en las consulta siempre a que se haga referencia a un campo especifico, se le ahorra al gestor el tiempo de localizar a que tabla y a que esquema pertenece el campo.

- **Diseño de consultas.**

    1. Combinar ciertas selecciones con un producto cartesiano anterior, para realizar una asociación o un JOIN.

    2. Detectar subconsultas con resultados equivalentes que puedan reutilizarse a lo largo de una misma consulta.

    3. Mediante cascada de selecciones, sustituir una selección compuesta en varias simples.

2.6 Selección de las estructuras de datos apropiadas para la implementación de estos algoritmos. Descripción de las clases que se utilicen para representar computacionalmente dicha estructura.

2.6.1 Estructura de datos

Algoritmo para llenar el grafo

1. Contar la cantidad de tablas que conformaran los nodos (vértices del grafo).
   1.1 Este valor se almacena en una variable para controlar el ciclo que recorrerá todas las tablas.

2. Seleccionar los id y tipos de cada una de estas tablas.

   2.1 Estos valores se guardan en arreglos para la posterior creación de los nodos.

3. Se crea un grafo vacio.

4. Se procede a llenar el grafo con el arreglo de id y tipos (cada posición en el arreglo del grafo representa un nodo). Se conforman todos los nodos del grafo pero sin conexiones (aristas) ni costo de caminos (peso) y sin el tipo de relación que se establece.

5. Se recorre el arreglo que contiene todos los nodos (tablas transformadas a nodos).

6. Se obtiene el id que representa a cada nodo para buscar su campo llave.

7. Se determinan la cantidad de llaves de ese nodo.

7.1 En caso que la cantidad campos llave sea mayor que 1, se realiza un ciclo con el caso 8 hasta el número de iteraciones igual a la cantidad de llaves encontradas.

7.2 En caso que la cantidad de campos llave sea igual a 1, ir al paso #8.

8. Se determina con cuantos campos se encuentra relacionado este campo.

8.1 En caso que la cantidad de campos con que se relaciona sea igual a 0, se procede a evaluar que dicha tabla representada por dicho campo no tiene relación, es decir, su llave primaria no pasa como llave foránea a otra tabla, ir al paso #12.

8.2 Cuando sea mayor que 0 se obtiene el arreglo de campos con que se relaciona.

9. Se procede a determinar el arreglo anterior de campos, a que tabla pertenece cada uno.

9.1 Se guarda un arreglo con las tablas a las que pertenecen cada uno de los campos almacenados en paso #8.

10. A cada nodo representado en el arreglo del paso #9 se procede a la asignación reversa de sgts (array) , cost (array) y rel (array): Esta es una asignación para dar una doble orientación a las relaciones, un nodo puede acceder a otros tanto como esos podrán acceder a él. Esta asignación se realiza directamente a los nodos persistentes en el grafo

11. Se procede a la actualización del nodo, seleccionado en el paso #5, se le asigna su sgts (array), cost (array) y rel (array) determinados.

12. En caso de no tener elementos el arreglo que conforman los campos a lo cual otro campo puede relacionarse, se le asigna valores null a su sgts (array) , cost (array) y rel (array) .

Ver Anexo 1: Diagrama de la representación de transición de una iteración hacia la construcción del grafo.

cost (array): Arreglo de valores que almacena de forma paralela con el sgts (array) el peso de un nodo a otro. El costo se determina en un función que realiza una comparación entre los tipos de nodos que van a relacionarse (creación del arista).

type: Representa el identificador de un nomenclador para determinar a que tipo de tabla se hace referencia. Estos tipos de tablas toman como valor: informativa, especificación y nomenclador.

sgts (array): Representa el arreglo de nodos con que una tabla se relaciona, guarda el id de las tablas que se relaciona y se identifica un nodo en el grafo.

flag (integer): Representa un estado , variable cuyo propósito tiene definido ser utilizada con bandera en los algoritmos de búsqueda.

rel (array): Representa un arreglo con el tipo de relación establecida entre el nodo y su nodo siguiente (sgts).

2.7 Clases

Nombre: Node	
Atributo	**Tipo**
$id	
$type	
$sgts	Array
$cost	Array
$flag	
$rel	Array
Para cada responsabilidad:	
Nombre:	__construct($newId,$newType,$newSgts,$newCost,$newFlag,$newRel)
Descripción:	Constructor de la clase.

Nombre:	getSgts
Descripción:	Funcion para obtener el valor del atributo $sgts.
Nombre:	getCost
Descripción:	Funcion para obtener el valor del atributo $cost.
Nombre:	getFlag
Descripción:	Funcion para obtener el valor del atributo $flag.
Nombre:	getId
Descripción:	Funcion para obtener el valor del atributo $id.
Nombre:	getType
Descripción:	Funcion para obtener el valor del atributo $type.
Nombre:	setSgts($newSgts)
Descripción:	Funcion para cambiar el valor del atributo $sgts.
Nombre:	setCost($newCost)
Descripción:	Funcion para cambiar el valor del atributo $cost.
Nombre:	setFlag($newFlag)
Descripción:	Funcion para cambiar el valor del atributo $flag.
Nombre:	setId($newId)
Descripción:	Funcion para cambiar el valor del atributo $id.
Nombre:	setType($newType)
Descripción:	Funcion para cambiar el valor del atributo $type.
Nombre:	getRel()
Descripción:	Funcion para obtener el valor del atributo $rel.
Nombre:	setRel($newRel)
Descripción:	Función para cambiar el valor del atributo $rel.

Nombre: Graf	
Atributo	**Tipo**
$nodes	Array
$cant	
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase.
Nombre:	getNodes
Descripción:	Función para obtener el valor del atributo $nodes.
Nombre:	setNodes($new_nodes)
Descripción:	Función para cambiar el valor del atributo $nodes.
Nombre:	getCant
Descripción:	Función para obtener el valor del atributo $cant.
Nombre:	setCant($newCant)
Descripción:	Función para cambiar el valor del atributo $cant.
Nombre:	insertNode($node,$pos)
Descripción:	Función para insertar un nodo en una posición indicada.
Nombre:	updateNode($idnode,$sgts,$cost)
Descripción:	Función para actualizar el valor de un nodo.
Nombre:	deleteNode($pos)
Descripción:	Función para eliminar un nodo en la posición indicada.
Nombre:	posNode($id)
Descripción:	Segun el id del nodo devuelve el pos asignado en el arreglo.
Nombre:	devSgts($pos)

Descripción:	Devuelve el arreglo de sgts según el pos asignado en el grafo.
Nombre:	countNode()
Descripción:	Función para determinar la cantidad de nodos en el grafo.
Nombre:	getNode($pos)
Descripción:	Devuelve el elemento que se encuentra en una posición del grafo.
Nombre:	nodeExist($idN)
Descripción:	Devuelve dado un $id de un nodo si el mismo se encuentra en el grafo.
Nombre:	isEmptyGraf()
Descripción:	Devuelve si el grafo esta vacio.
Nombre:	searchElement($elemnt,$array)
Descripción:	Busca un elemento en un arreglo dado, y si lo encuentra devuelve la posición donde lo encontró.
Nombre:	devType($id)
Descripción:	Devuelve el type de un nodo que se encuentra en el grafo según su $id.
Nombre:	devCost($id1,$id2)
Descripción:	Devuelve el costo de recorrido de un nodo a otro.
Nombre:	devArrCost($id)
Descripción:	Dado un id devuelve el arreglo de costo del mismo
Nombre:	vecinos($id1,$id2)
Descripción:	Determina si 2 nodos son vecinos o no.

2.8 Operaciones necesarias

Función getDatos ($idplant, & $sess = array ()): método estándar definido para todas las clases de negocio que se encargada de procesar mediante el ‘’accion’’ el mensaje enviado por la interfaz de usuario, en dependencia de la opción que esta represente se pasará a la ejecución de las funciones que contienen la lógica para resolver la petición.

Función getArray (): se le asigna la responsabilidad de de pasar a un formato especifico la información que proviene de la interfaz, permitiendo así un mejor manejo de la misma.

Función crearArbol (): se encarga de devolver la estructura que la interfaz utilizará para mostrar al usuario la estructura de la base de datos a la que puede hacer recuperaciones.

Función getCampos (): se encarga de obtener de los datos que provienen de la interfaz los que forman la estructura de la consulta.

Función insertar (): se encarga de utilizando el método getCampos (), para guardar la estructura de la consulta en la base de datos.

Todas estas funcionalidades interactúan con las clases de acceso a datos para el procesamiento de la información.

2.9 Descripción de la implementación

Para la implementación de la aplicación en primer lugar se definieron una serie de pasos a seguir con el objetivo de humanizar la experiencia del usuario, llevando a este a un nivel de abstracción en el cual sin contar con conocimientos sobre bases de datos el mismo pueda realizar de manera organizada, consultas sobre la información a la que el mismo tiene acceso. Como se trata de un sistema dinámico se debe

garantizar un modelo que generalice la formación de las consultas que a bajo nivel serán hechas por el mismo al gestor de bases de datos.

Para dar al sistema la independencia de crear la consulta tomando como base los datos entrados por el usuario se siguen los siguientes pasos:

Como una consulta tiene su estructura formada por: “SELECT [campo1], [campo2], [campo3] FROM [esquema.tabla1], [esquema.tabla2], [esquema.tabla3] WHERE [campo2]=’valor’”, se determino para general la consulta que: los campos afectados por el SELEC serían los pasados por el usuario, lo afectado por el FROM las tablas a las que pertenecen estos campos y el WHERE la condición especificada por el usuario, este modelo resuelve la problemática en su caso de menor complejidad. Cuando las consultas ganan en complejidad por la cantidad de JOIN y de tablas que se deben vincular para conformar una consulta entra a jugar su papel la estructura de datos que fue seleccionada para dar solución al problema que se plantea, que no es otro que la elaboración de una consulta que puede ser de mediana o de alta complejidad, utilizando la información contenida en el grafo se elaboran un grupo de reglas para la confección de la consulta, las que ayudan a determinar los caminos a seguir para obtener un resultado final. La utilización de la estructura grafo tiene como fin la representación de bases de datos para ayudar a la toma de decisión para el diseño de las consultas a realizar.

2.9.1 Realización de la recuperación

- **El recuperador visualiza los esquemas sobre los que tiene permiso en la base de datos.**
  Ver anexo 2.

- **El recuperador selecciona determinado esquema, el árbol se despliega y muestra las tablas a las que tiene permiso contenidas en el mismo.**
  Ver anexo 3.

- **Se seleccionan los campos de los que se desea visualizar información, los mismos aparecen en el área de selección.**
  Ver anexo 4

- **Se determinan las condiciones (los filtros) de la recuperación.**
  Ver anexo 5

- **Se escoge el campo por el cual se va a ordenar la información y el orden de la misma (ascendente o descendente).**
  Ver anexo 6

- **Se ejecuta la recuperación en el botón Ejecutar, de esta manera se muestra un listado en el grid con la información resultante.**
  Ver anexo 7

2.10 Estándares de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, que de la impresión de que fue escrito por un solo programador, esta importante determinación se debe tomar al iniciar un proyecto haciendo que los implementadores trabajen de forma coordinada. La legibilidad del código fuente repercute directamente en el entendimiento que pueda tener otro programador del mismo, aspecto crucial ya que todo software tiene que someterse constantemente a mantenimiento y mejora de sus funcionalidades. El mejor método para lograr que un grupo de desarrolladores mantenga un código de calidad es establecer un estándar de codificación sobre el cual se realizarán revisiones rutinarias.

2.11 Conclusiones

En este capitulo se mostraron los componentes, las estructuras de datos así como algunas de las funcionalidades que permitieron el desarrollo de la aplicación, se esbozó un conglomerado de conceptos que se tuvieron en cuenta al proponer y ejecutar la solución técnica y se mostró además brevemente las pantallas con que cuenta la aplicación en esta primera iteración. Con la implementación de las estructuras de datos seleccionadas se avanzo en el marco de brindar en próximas iteraciones una solución más completa.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

Con los resultados arrojados en el capitulo anterior se comenzarán a testear la aplicación, se mostrará un diagrama de despliegue y se procurarán recomendaciones para las próximas iteraciones.

Casos de Pruebas:

1. No hay registros que cumplan con las condiciones
2. No se encuentran los registros solicitados
3. Se muestra el informe solicitado.

Recuperar Información		
Entrada	**Resultados**	**Condiciones**
El recuperador solicita: el nombre, primer apellido y edad de todos los hombres, de color de piel blanca y más de 200 libras de peso.	El sistema muestra un mensaje informativo “No se encuentran los registros solicitados”	No existe la combinación de datos.
El recuperador solicita la edad de las personas de sexo femenino y nombre Ramón.	El sistema muestra un mensaje informativo “No se encuentran los registros solicitados”	No existe esta información en la base de datos.
El recuperador solicita el nivel escolar de las personas que tengan el color de los	El sistema muestra un resultado con el listado de las personas que cumplen las restricciones establecidas.	Existen los datos en la bases de datos

ojos verdes, una edad mayor que 50 años y nombre Pedro.		

3.2 Diagrama de despliegue

En el siguiente diagrama de despliegue se representa la distribución física del sistema en términos de cómo se distribuirán la funcionalidades entre los nodos, cada nodo representa un recurso de cómputo, siendo estos procesadores o dispositivos hardware que se necesitarán para el despliegue del sistema.

Ver anexo 8: Diagrama de Despliegue

3.3 Conclusiones

En este capítulo se obtuvieron importantes resultados sobre el comportamiento del sistema a partir de la realización de los procedimientos de pruebas. Además de obtener la distribución física del sistema mediante el diagrama de componentes y presentar un grupo de recomendaciones, que a entender de los autores deben ser tenidas en cuenta en futuras iteraciones.

CONCLUSIONES

El resultado de esta investigación está avalado por la implementación del submódulo de Recuperaciones de Información del Módulo de Recuperaciones Dinámicas, de esta forma se da cumplimiento a los objetivos trazados, así como a las tareas descritas para llevar a cabo los objetivos.

Con el estudio de la forma en que se desarrollan las recuperaciones de información por otros sistemas que se orientan a casos específicos se llega a la conclusión que nuestra solución será de gran aceptación por el cliente y se convertirá en una herramienta agregada a los módulos del polo productivo.

RECOMENDACIONES

1 Generalizar la aplicación a otros SGBD, para lo que se propone el uso de ODBC.

2 Incluir mayor diversidad en las salidas de los reportes generados por los usuarios.

CONSULTADAS

[*] Wikipedia La enciclopedia libre. Sistema de gestión de base de datos, 2008. [http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos]

[*] Wikipedia La enciclopedia libre. Ajax, 2008. [http://es.wikipedia.org/wiki/AJAX]

[*] Wikipedia La enciclopedia libre.Teoría de grafo, 2008. [http://es.wikipedia.org/wiki/Teor%C3%ADa_de_grafos]

GLOSARIO DE TÉRMINOS

Grafo: En matemáticas y ciencias de la computación, la teoría de grafos estudia las propiedades de los grafos, que son colecciones de objetos llamados vértices (o nodos) conectados por líneas llamadas aristas (o arcos) que pueden tener orientación (dirección asignada). Típicamente, un grafo está diseñado por una serie de puntos (los vértices) conectados por líneas (las aristas).

OLAP: OLAP es el acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing). Es una solución utilizada en el campo de la llamada Inteligencia empresarial (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos.

ERP: Los sistemas de planificación de recursos empresariales (ERP) son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía comprometida en la producción de bienes o servicios.

CRM: La administración de la relación con los clientes, CRM, es parte de una estrategia de negocio centrada en el cliente. Una parte fundamental de su idea es, precisamente, la de recopilar la mayor cantidad de información posible sobre los clientes, para poder dar valor a la oferta. La empresa debe trabajar para conocer las necesidades de los mismos y así poder adelantar una oferta y mejorar la calidad en la atención.

COM: Component Object Model (COM) es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología.

DB2: Es una marca comercial, propiedad de IBM, bajo la cual se comercializa el sistema de gestión de base de datos.

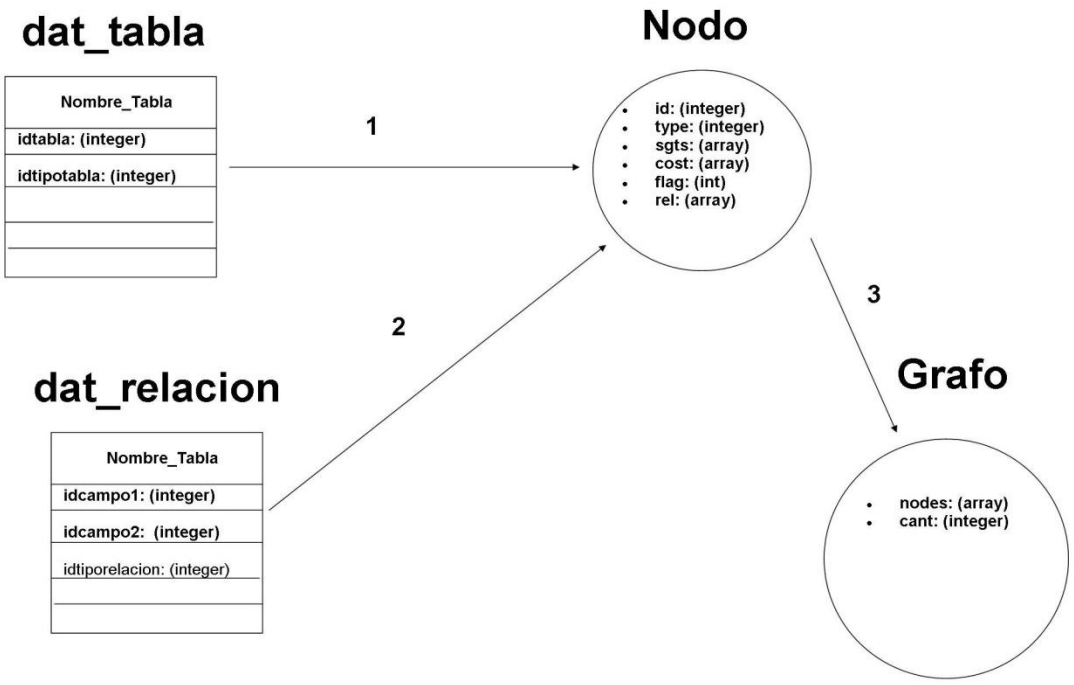
Open Source: Código abierto, es el término con el que se conoce al software distribuido y desarrollado libremente. Fue utilizado por primera vez en 1998 por algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (free software).

Bugs: Un defecto de software, es el resultado de un fallo o deficiencia durante el proceso de creación de programas de ordenador o computadora (software). Dicho fallo puede presentarse en cualquiera de las etapas del ciclo de vida del software aunque los más evidentes se dan en la etapa de desarrollo y programación. Los errores pueden suceder en cualquier etapa de la creación de software.

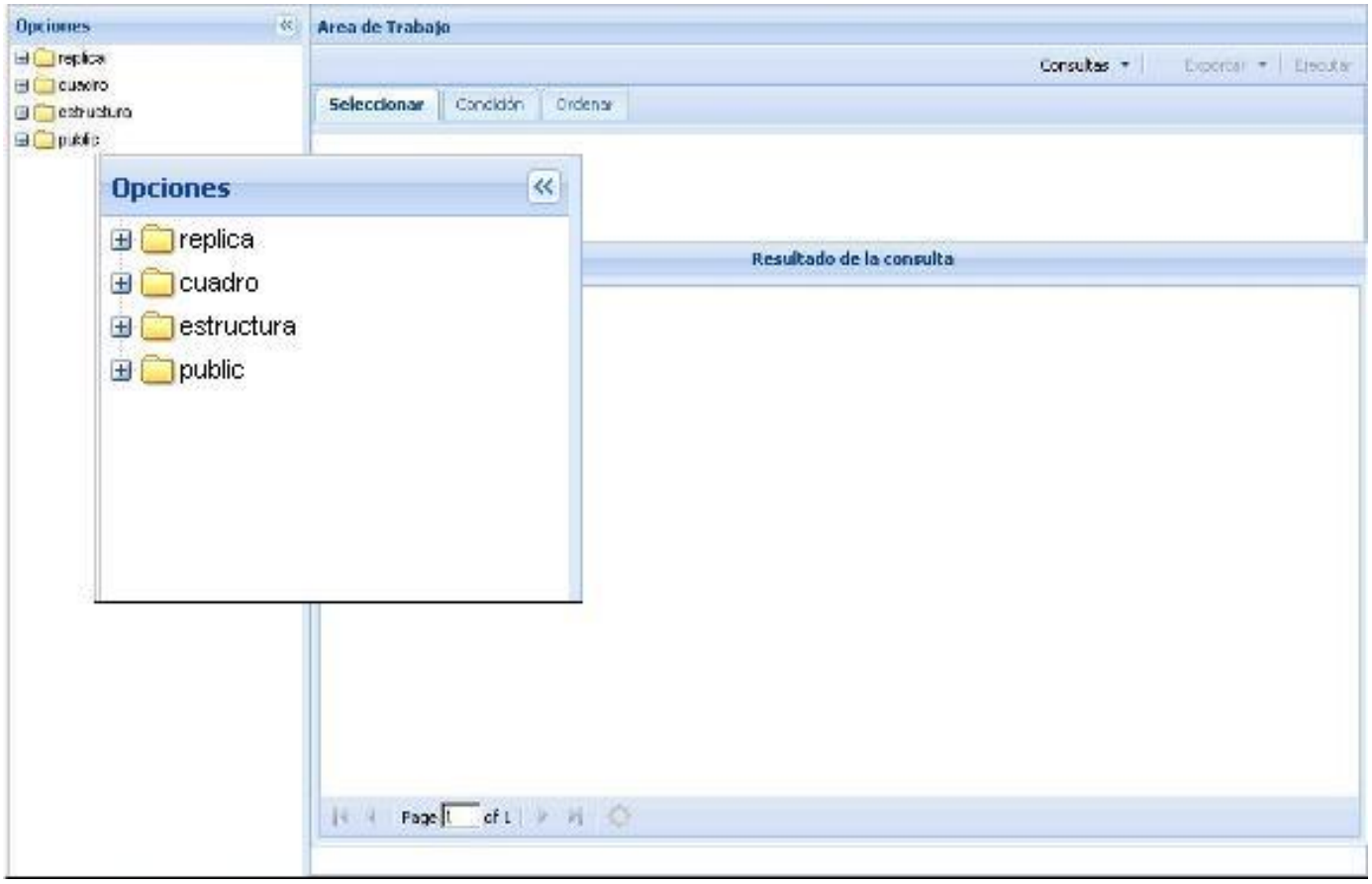
Triple licencia: El uso de una licencia dual o doble licenciamiento es la práctica de conceder dos o más licencias para el mismo producto intelectual.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

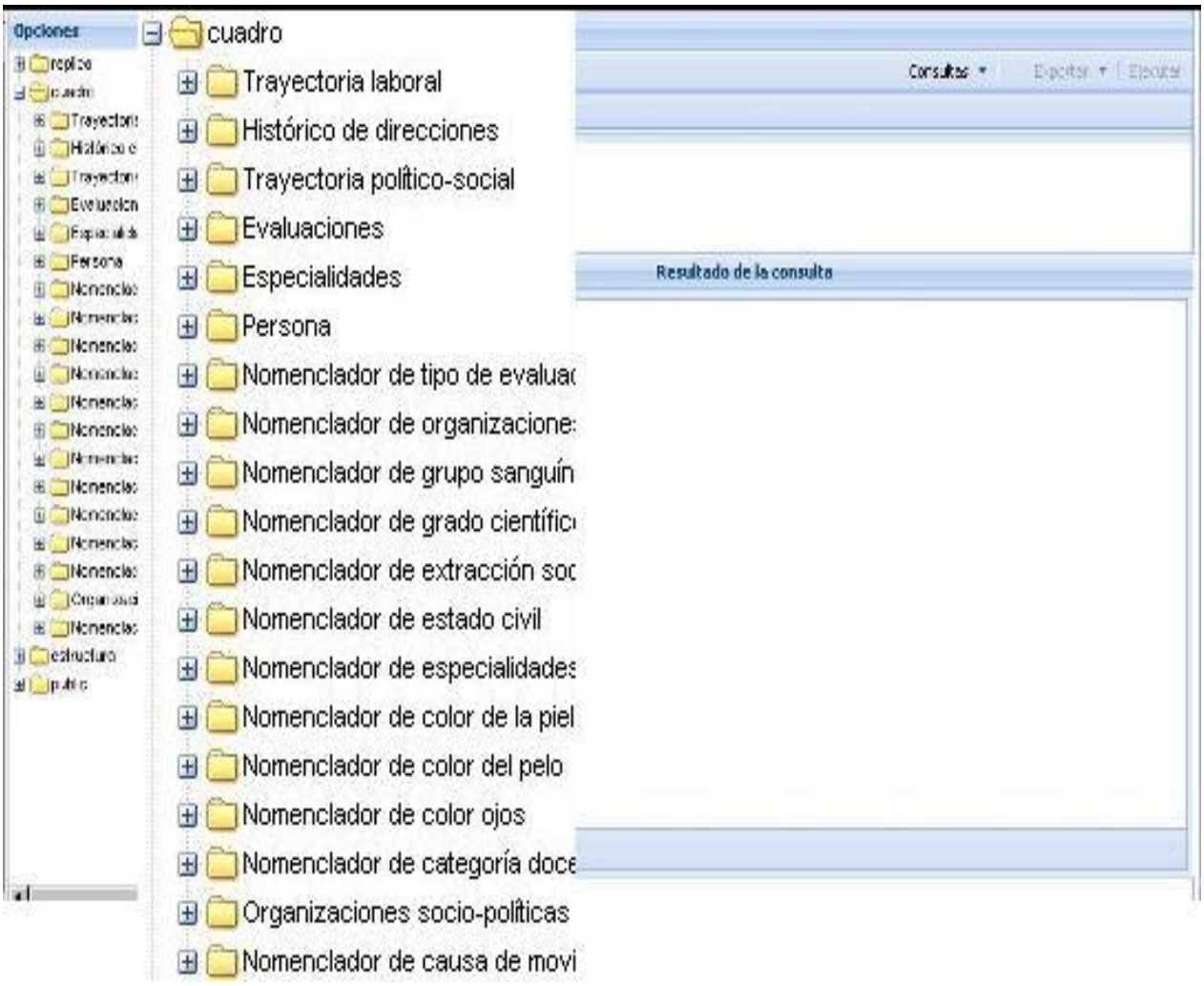
ANEXOS



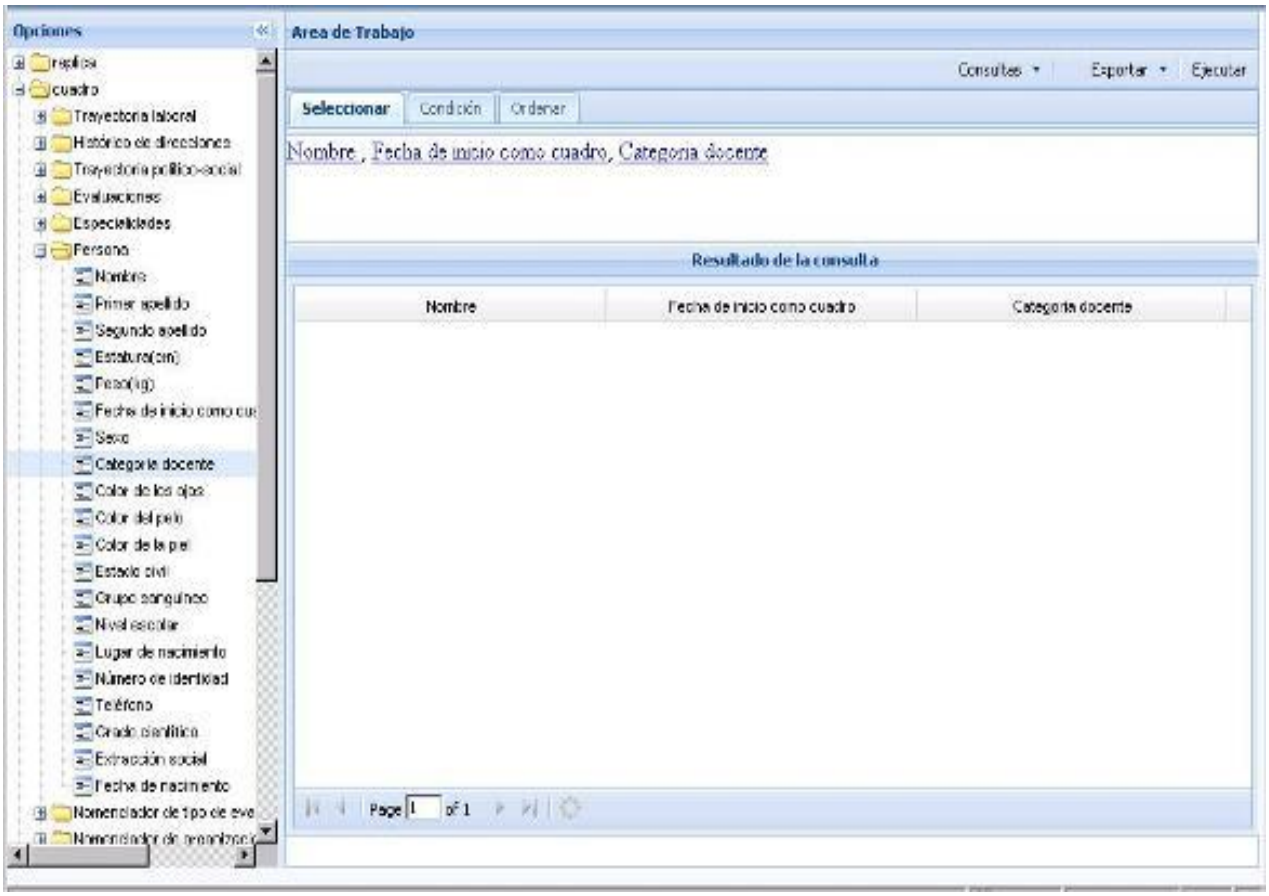
Anexo 1: Diagrama de la representación de transición de una iteración hacia la construcción del grafo.


Opciones
replica
cuadro
estructura
public
Area de Trabajo
Consultas
Exportar
Ejecutar
Seleccionar
Condición
Ordenar
Resultado de la consulta
Page 1 of 1

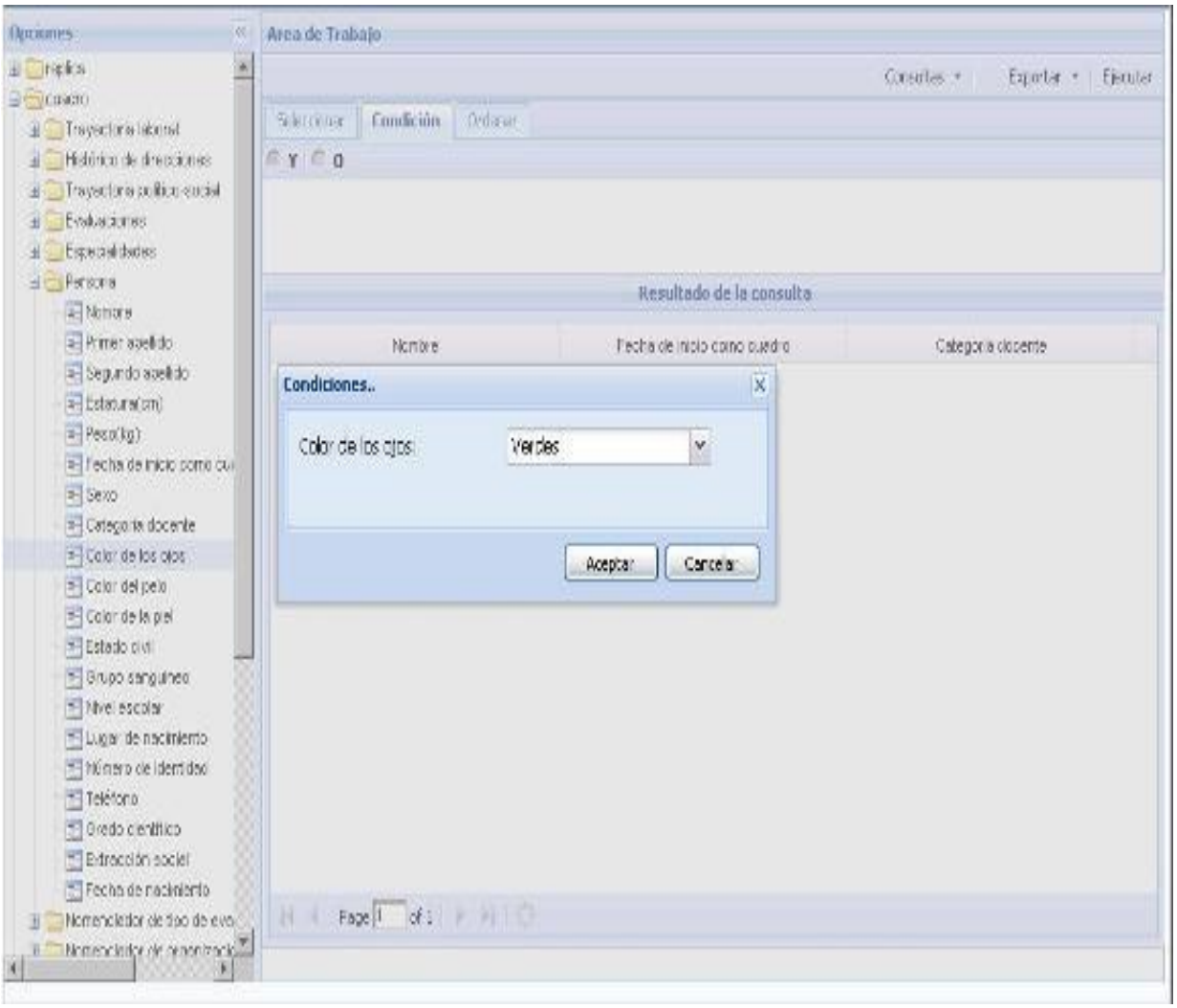
Anexo 2.


Opciones
cuadro
Trayectoria laboral
Histórico de direcciones
Trayectoria político-social
Evaluaciones
Especialidades
Persona
Nomenclador de tipo de evaluac
Nomenclador de organizacione
Nomenclador de grupo sanguín
Nomenclador de grado científic
Nomenclador de extracción soc
Nomenclador de estado civil
Nomenclador de especialidades
Nomenclador de color de la piel
Nomenclador de color del pelo
Nomenclador de color ojos
Nomenclador de categoría doce
Organizaciones socio-políticas
Nomenclador de causa de movi
Consultas
Exportar
Ejecutar
Resultado de la consulta

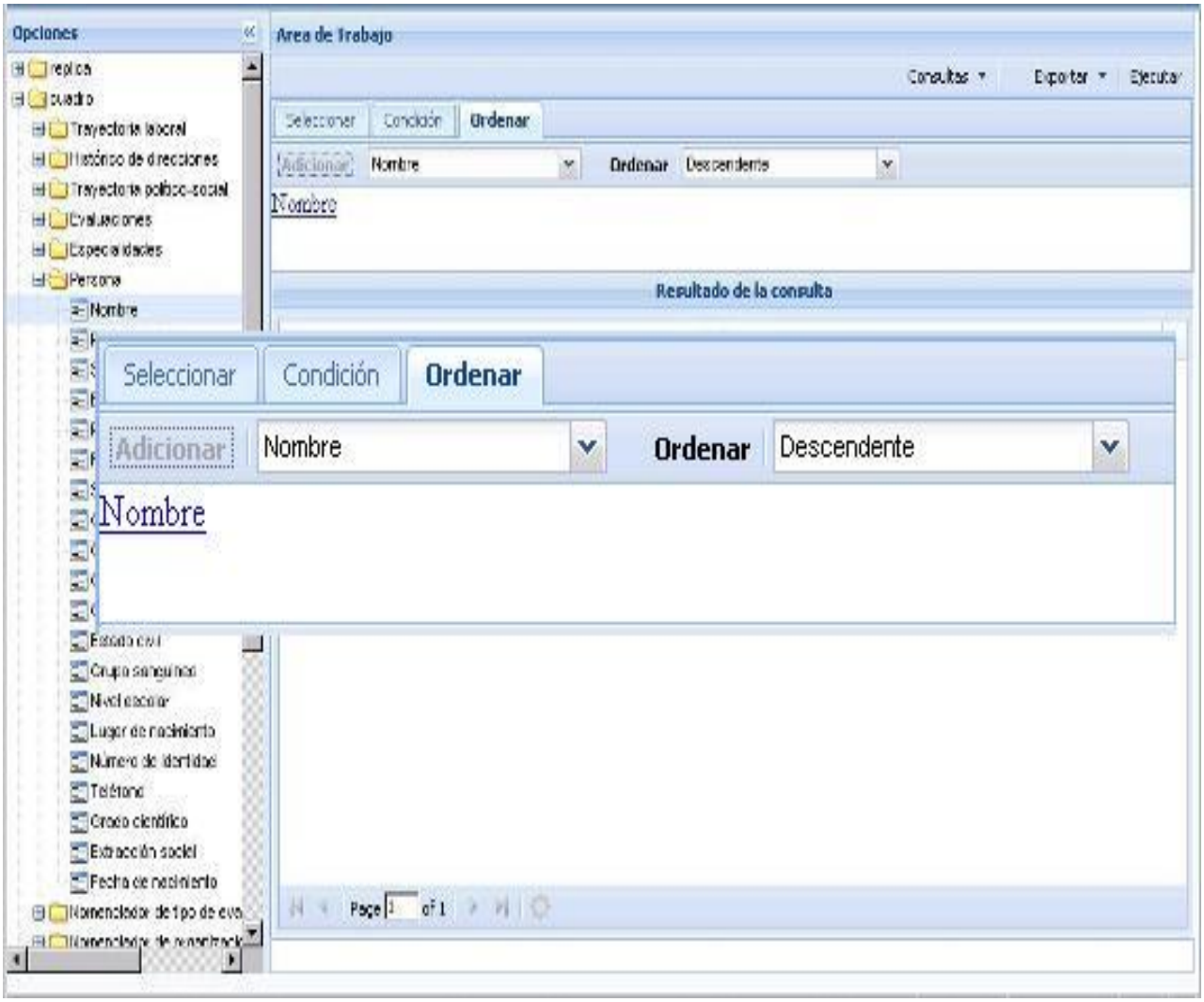
Anexo 3


Opciones
Area de Trabajo
Consultas
Exportar
Ejecutar
Seleccionar
Condición
Ordenar
Nombre , Fecha de inicio como cuadro, Categoria docente
Resultado de la consulta
Nombre
Fecha de inicio como cuadro
Categoria docente
Page 1 of 1
replica
cuadro
Trayectoria laboral
Histórico de direcciones
Trayectoria político-social
Evaluaciones
Especialidades
Persona
Nombre
Primer apellido
Segundo apellido
Estatura(cm)
Peso(kg)
Sexo
Categoria docente
Color de los ojos
Color del pelo
Color de la piel
Estado civil
Grupo sanguíneo
Nivel escolar
Lugar de nacimiento
Número de identidad
Teléfono
Grado científico
Extracción social
Fecha de nacimiento

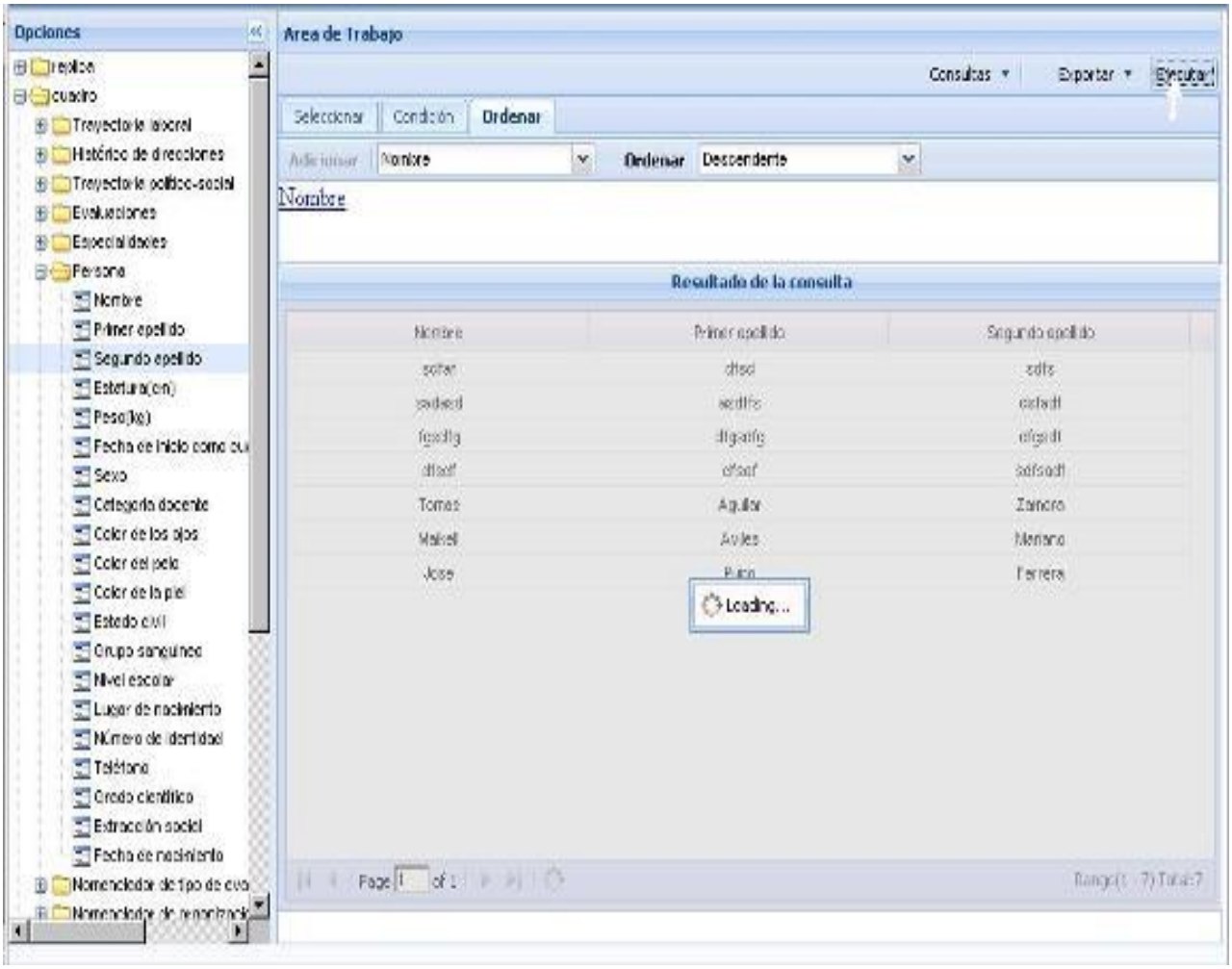
Anexo 4


Opciones
Area de Trabajo
Condición
Resultado de la consulta
Nombre
Fecha de inicio como cuadro
Categoria docente
Condiciones..
Color de los ojos:
Verdes
Aceptar
Cancelar

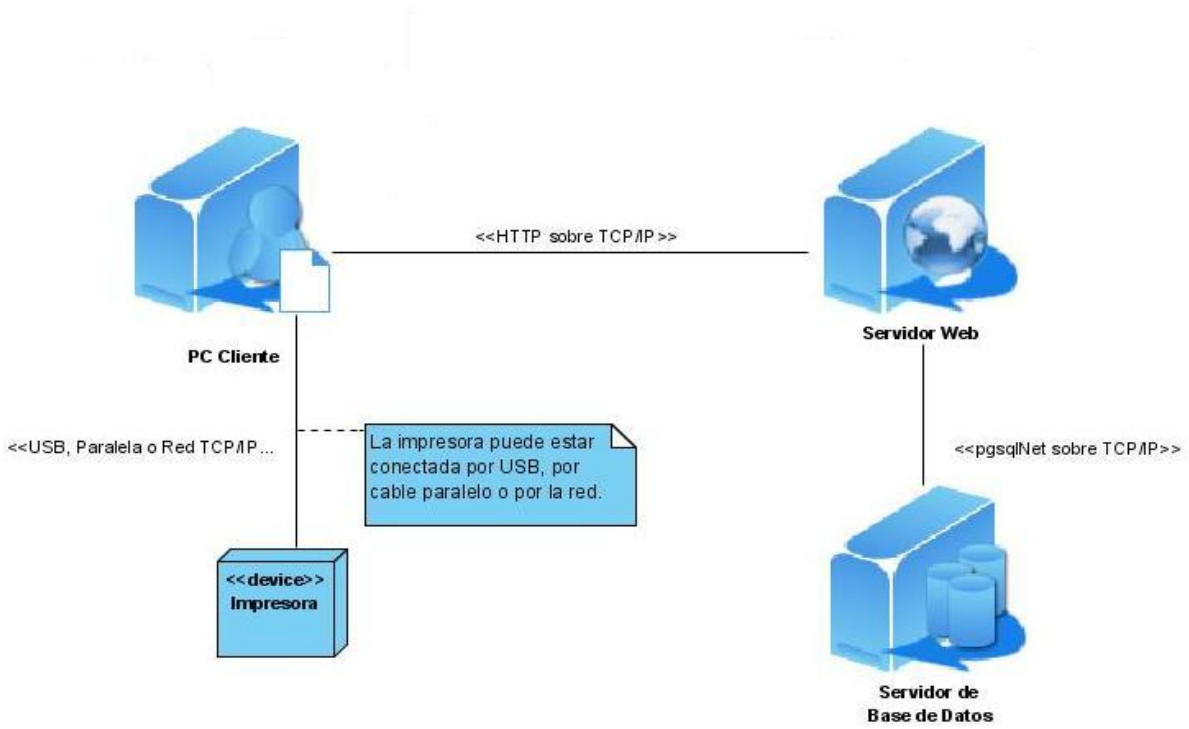
Anexo 5


Opciones
Area de Trabajo
Consultas
Exportar
Ejecutar
Seleccionar
Condición
Ordenar
Adicionar
Nombre
Ordenar
Descendente
Nombre
Resultado de la consulta
Trayectoria laboral
Histórico de direcciones
Trayectoria político-social
Evaluaciones
Especialidades
Persona
Nombre
Nivel escolar
Lugar de nacimiento
Número de identidad
Teléfono
Centro científico
Extracción social
Fecha de nacimiento
Page 1 of 1

Anexo 6


Opciones
Area de Trabajo
Consultas
Exportar
Ejecutar
Seleccionar
Condición
Ordenar
Nombre
Ordenar
Descendente
Nombre
Resultado de la consulta
Nombre
Primer apellido
Segundo apellido
Tomas
Aguilar
Zamora
Maikell
Aviles
Moreno
Jose
Loading...
Page 1 of 1

Anexo 7



Anexo 8: Diagrama de despliegue.

Bibliografía

Lockhart, Thomas. *The PostgreSQL Programmer's Guide* . 1998.

Data Structures, Algorithms and Object-Oriented Programming. Gregory L. Heileman,University of New Mexico

Relational Database Index Design and the Optimizers.Tapio Lahdenmaki & Michael Leach, New Jersey, John Wiley, 2005

Recuperaciones Dinámicas .Universidad de Las Ciencias Informáticas .2006.Raidel Muñoz Vidal y Michel López

Recuperaciones Dinámicas para aplicaciones de Gestión.2006.Elisabeth Pérez Pérez y Dalkis Mok Vásquez

Sant'Anna, Mauro. Reportes en .NET con Cristal Reports. MSDN, 2007 http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art11.asp

KRONOS, J.T.Concepto de recuperación de información, 2007. http://www.tramullas.com/documatica/3-1.html

Jhon, R.Agata Report:un reportador libre para las BD más comunes,2006. http://alts.homelinux.net/libreapp.php?id=408

Microsoft Office Excel 2003. Qué es Microsoft Query?,2007. http://office.microsoft.com/es-es/excel/HP052747513082.aspx?pid=CH062528423082