Universidad de Cienfuegos
Facultad de informática

Título: Sistema para la gestión y control de planteamientos en el Poder popular de Cienfuegos

Trabajo de Diploma para optar por el título de

Ingeniero Informático

Autor(es): Giordanis Vázquez Cabrera

Tutor(es): Yamilé Mesa

Enero 2010

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la facultad de informática de la Universidad de Cienfuegos a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los __5__ días del mes de ___Enero_____ del año ____2010____.

Giordanis Vázquez Cabrera

_____

Datos de Contacto

"[insertar breve curriculum e información de contacto del tutor]"
"[insertar breve curriculum e información de contacto del asesor]"
"[insertar breve curriculum e información de contacto del consultante]"

Agradecimientos

"[Insertar agradecimientos (opcional)]"

Dedicatoria

"[Insertar dedicatoria (opcional)]"

RESUMEN

La presente investigación titulada: **"Sistema para la gestión y control de planteamientos en el Poder popular de Cienfuegos"** se realiza en esta ciudad como alternativa de solución a las deficiencias en la gestión de la información en dicho centro.

El almacenamiento de datos en el Poder Popular se realiza en modelos de papel y los informes son confeccionados manualmente, lo que da margen a errores de cálculo y redundancia de la información. Hay que emplear tiempo en buscar los registros con los datos de los planteamientos de la población, por lo que este período de consulta es prolongado y se realiza indistintamente cada vez que un directivo solicita información. Todo lo planteado anteriormente sería más fácil utilizando las TICs.

Se emplean los lenguajes HTML, JavaScript y PHP. Para almacenar información se utiliza el gestor de bases de datos MySQL y para modelar el análisis, diseño e implementación del sistema, la metodología RUP.

PALABRAS CLAVE

PP (Poder Popular).

SGC (Sistema de gestión y control).

SCP (Sistema de control de planteamientos).

ÍNDICE

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 4

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA 41

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA 71

CONCLUSIONES 87

RECOMENDACIONES 88

REFERENCIAS BIBLIOGRÁFICAS 89

BIBLIOGRAFÍA 91

GLOSARIO 92

ANEXOS 93

INTRODUCCIÓN

Los servicios de información y documentación accesibles a través de internet, específicamente mediante servidores web, aumentan a una alta velocidad. La evolución de la web desde hace más de una década ha ido produciendo la sustitución de páginas y documentos estáticos por documentos generados dinámicamente, dependiendo la interacción del usuario con la lógica de procesos y flujos de trabajo definida por los creadores del servicio y a la disponibilidad de repositorios de información cada vez mayores. Evidentemente, se ha ido pasando progresivamente de un concepto de publicación de páginas web, muy simple en un principio, a esquemas más complejos y diferenciados, fundamentados en procedimientos y técnicas basados en la gestión de información. La creciente complejidad de los servicios y de los sistemas que los soportan ha hecho necesaria la formulación de un corpus teórico y práctico en el que se combinen las técnicas clásicas de gestión de información en las organizaciones con las características propias del medio digital. El Poder Popular en el municipio Cienfuegos necesita automatizar el área de planteamientos, actualmente labora en él un grupo de funcionarios y la secretaria general que es la encargada de recibir, clasificar y almacenar la información que se recibe (planteamientos).Estos datos solo pueden ser modificados por la misma, la cual supervisa y gestiona el flujo de información acarreada. A estos trabajadores se les hace muy engorroso el trabajo puesto que no cuentan con un software que gestione y controle los numerosos datos con que tienen que lidiar en cada jornada de trabajo. Requieren de una herramienta que les simplifique el almacenamiento y el manejo del volumen de datos específicos derivados de los planteamientos que allí se reciben, un ejemplo esclarecedor es que si el presidente de la asamblea o cualquier otro trabajador del poder popular requiere consultar algunos de los muchos datos con que cuenta esta área, tenga que trasladarse desde su despacho hasta dicho departamento para entonces de forma verbal o por documentos escritos tener acceso a la información, lo cual representa una perdida de tiempo y decrece la calidad del trabajo.

Teniendo en cuenta lo descrito anteriormente se define como **Problema Científico:** ¿Cómo automatizar el flujo de datos derivados de los planteamientos entre la secretaria de la Asamblea y

los trabajadores del Poder Popular Municipal de Cienfuegos, para mejorar la gestión de la información?

El **Objeto de Estudio** está enfocado en la gestión de la información en el Poder Popular, teniendo como **Campo de Acción** los sistemas de gestión de la información en el área de planteamientos del Poder Popular.

Para darle solución al problema planteado se define como **Objetivo general** desarrollar un sistema que facilite el flujo de información entre la secretaria de la Asamblea y los trabajadores del Poder Popular Municipal de Cienfuegos para mejorar la gestión de los planteamientos.

Para dar cumplimiento al objetivo general se han trazado los siguientes **Objetivos específicos:**

- Sistematizar los antecedentes y estado actual de los conceptos y herramientas existentes para procesamiento de la información.
- Realizar el modelo de análisis correspondiente a los conceptos y definiciones asociado al procesamiento de la información.
- Realizar el diseño y la implementación para el sistema de procesamiento de la información.

Para que se lleve a cabo un exitoso cumplimiento de los objetivos planteados se proponen las siguientes **Tareas de investigación:**

- Estudio de las metodologías de desarrollo de software, de las principales Herramientas CASE y lenguajes de modelado.
- Estudio de los principales lenguajes de programación Web y de los Sistemas Gestores de Bases de Datos.
- Estudio del estado del arte referente a los sistemas de gestión de información.

Con el objetivo de desarrollar las tareas propuestas, se utilizaron los siguientes **métodos científicos de la investigación:**

Histórico-lógico

Este método fue empleado para el estudio del surgimiento y la evolución de la gestión de la información en los diferentes períodos de la historia, poniendo de manifiesto la lógica interna de su desarrollo.

Analítico – sintético:

La utilización de este método permitió el estudio de los rasgos que caracterizan y distinguen a los principales sistemas existentes que se relacionan con el objeto de estudio, permitiendo la extracción de los elementos más importantes

Los empíricos:

Método de entrevista:

Este método se utilizó para lograr un mayor acercamiento con los clientes con el objetivo de tener mayor dominio del negocio en cuestión, así como para obtener información relacionada con el tema de algunos de los especialistas del organismo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1. Sistemas de Información.

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. A partir de este reconocimiento los sistemas de información comienzan a formar parte de las estrategias de las organizaciones, por lo que tienen una alta prioridad y son la base de la eficiencia y eficacia de las organizaciones que así los consideran.

Una organización puede adquirir nuevas máquinas computadoras, instalar nuevos productos de telecomunicaciones, elaborar una página web, realizar comercio electrónico, pero ello no implica, que exista en su organización un sistema de información. Junto a la información, los otros componentes básicos que constituyen a un sistema de información son: los usuarios, los equipos informáticos (software, hardware y tecnologías de almacenamiento de la información y de las telecomunicaciones) y los especialistas. Para comprender los Sistemas de Información hay que conocer que existen necesidades en las organizaciones y comunidades que deben ser satisfechas, hay que dominar las complejidades de cómo se maneja la información y cuáles son las potencialidades de los medios que se emplean para organizarla y recuperarla.

El manejo de datos e información constituye uno de los aspectos más importantes para cualquier organización, abarcando diferentes actividades como la recolección, almacenamiento, recuperación, difusión hacia lugares y personas indicadas, así como el uso que de ellos se hace para varias actividades dentro de una organización. Por lo general los Sistemas de Información son grandes y complejos debido a la diversidad de sus componentes, procesos y las relaciones que existen entre ellos, caracterizándose por la capacidad que tienen de adaptarse al medio. Al mencionar los procesos que ocurren en los Sistemas de Información, se consideran aquellos procesos que están asociados a los recursos de información y a cómo estos aumentan la probabilidad de que los usuarios en un ambiente dado encuentren mensajes útiles en las salidas de estos sistemas. Las facilidades que pudiera llegar a dar este tipo de servicio informático dependen en gran medida de la manera en que se conciba la aplicación, por lo que se considera

de carácter obligatorio llevar a cabo un estudio de las tendencia tecnológicas actuales, para que de esta manera se pueda hacer una selección lo mas aceptada posible de las herramientas y otros elementos influyentes. **[1]**

2. Tendencias y Tecnologías Actuales.

En el presente epígrafe se hará un análisis y estudio de las tecnologías actuales con el objetivo de contar con los elementos necesarios para dar una propuesta lo más eficiente posible de las tecnologías destinadas a ser usadas en el desarrollo de la aplicación.

2.1 ¿Qué es un sitio Web?

Un sitio Web es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada home page, con un nombre de dominio y dirección en Internet específicos. Empleados por las instituciones públicas y privadas, organizaciones e individuos para comunicarse con el mundo entero. Ventajas principales de los sitios Web:

• Son sistemas multiplataforma, ya que se puede migrar de sistema y cambiar el hardware libremente y no afecta el funcionamiento de las aplicaciones del servidor.

• Se necesita solo una computadora con un navegador Web para el uso de estos ya que no se requiere de altas complicaciones hardware/software.

• El usuario no se da cuenta de las actualizaciones del software ya que estas se realizan automáticamente.

• Proporciona facilidad en el trabajo a distancia ya que el acceso es a través de la red.

Desventajas principales de los sitios Web:

• Difícil implementación de la seguridad.

• Es necesaria una conexión rápida y permanente.

• Existen limitaciones en los documentos estándar HTML.

Este tipo de aplicación puede ser desarrollada en diversos lenguajes de los cuales se hará un estudio con el fin de lograr explotar al máximo las bondades de las aplicaciones Web. **[2]**

Lenguajes de programación Web.

Un lenguaje de programación es un lenguaje que se utiliza para controlar el comportamiento de una computadora. Es un grupo de reglas sintácticas y semánticas que definen su estructura y significado. Existen muchos lenguajes de programación, y cada uno tiene características que definen los campos en los que pueden ser implementados. A continuación se verán algunos de ellos, específicamente aquellos que se utilizan para implementar aplicaciones Web. **[3]**

Personal Home Page (PHP)

PHP (Personal Home Page) es el acrónimo recursivo de PHP: Hypertext Preprocessor. PHP es un lenguaje de programación usado generalmente para la creación de contenidos para sitios Web. Este lenguaje se acopla al HTML (páginas web) para definir procedimientos que ha de realizar el servidor de Web, por ejemplo procesar un formulario, enviar o extraer datos de una base de datos (acoplándose también con un lenguaje de tipo SQL), enviar una u otra página Web según determinadas condiciones prefijadas por el programador, etc. Últimamente también se usa para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+. PHP fue originalmente diseñado en Perl, seguidos por la escritura de un grupo de CGI binarios escritos en el lenguaje C por programador Danés – Canadiense Rasmus Lerdorf en el año 1994, pero como PHP está desarrollado en política de código abierto muchos desarrolladores han hecho sus contribuciones, siendo la PHP 5.2.3 la versión más reciente. Esta versión incluye PDO (Objetos de Información de PHP o PHP Data Objects) y mejoras utilizando las ventajas que provee el nuevo Zend Engine 2. PHP es gratuito e independiente de la plataforma, pues existe un módulo de PHP para casi cualquier servidor Web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje, lo que significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún

trabajo. Además, es seguro, con gran librería de funciones y mucha documentación. Es un lenguaje multiplataforma que puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux. La característica más potente de PHP es su soporte para una gran cantidad de bases de datos que se utilizan actualmente, destacando su conectividad con MySQL. Agrupa en su última versión una serie de ventajas tales como:

• Soporte sólido para Programación Orientada a Objetos con PHP Data Objects.
• Es un software totalmente libre.
• Mejoras de rendimiento.
• Mejor soporte para MySQL con extensión de rewrite completa.
• Iteradores de datos.
• Excepciones de errores.

[4]

PERL

PERL, Lenguaje Práctico para la Extracción e Informe, es un lenguaje especializado en el procesamiento de textos, particularmente, extraer y validar las respuestas a cuestionarios incluidos en páginas Web. Perl es un lenguaje interpretado que tiene muchas utilidades, pero está orientado principalmente a la búsqueda, extracción y formateo de ficheros de tipo texto. También es muy usado para el manejo y gestión de procesos (estado de procesos, conteo y extracción de parámetros característicos, etc.). Aunque desarrollado originalmente en un entorno UNIX, actualmente hay versiones para casi todos los sistemas operativos: Windows, Mc OS, de manera que es un lenguaje multiplataforma. Es gratuito, libre y uno de los lenguajes más utilizados en la programación de CGI script para el intercambio de información entre aplicaciones externas y servicios de información. Tiene la desventaja de ser un lenguaje que consume muchos recursos de la máquina, lo que significa que no es muy ligero. Si no se escribe con cuidado puede llegar a ser un código ilegible y no se pueden compilar programas con Perl.

ASP.

ASP (Active Server Pages - Página Activa en el Servidor) es una tecnología del lado servidor desarrollado por Microsoft para páginas Web generadas dinámicamente. La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida ya que programar en ASP es como programar en VisualBasic, pero con muchas limitaciones. ASP no es en sí mismo un lenguaje de programación, si no más bien un marco sobre el que se construyen aplicaciones basadas en Internet, apoyándose para ello en el lenguaje HTML, en lenguajes de script, en motores de bases de datos y en el lenguaje de consulta SQL. ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0), ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). Las versiones PRE-.NET se denominan actualmente como ASP clásico. Evidentemente una de las desventajas es que su empleo se realiza solo sobre plataformas funcionando bajo sistema Windows NT. **[5]**

ASP.NET

Microsoft ASP.NET es una tecnología libre que permite a los programadores crear aplicaciones web dinámicas. ASP.NET puede utilizarse para crear cualquier cosa, desde pequeñas aplicaciones personales, hasta grandes sitios de empresas. Puede ejecutarse de formas distintas las cuales son:

Aplicaciones cliente/servidor: Estas aplicaciones están típicamente en formato de ejecutables compilados. Estos pueden integrar toda la riqueza de una interfaz de usuario, tal es el caso de las aplicaciones de desempeño y productividad, pero no se reúne la lógica de negocio como un recurso que se pueda reutilizar. Además acostumbran ser menos gestionables y escalables que las demás aplicaciones.

Aplicaciones que utilizan el navegador: Dichas aplicaciones están caracterizadas por contar con una interfaz de web rica y muy útil. La interfaz gráfica integra varias tecnologías, las cuales son el HTML, XHTML, scripting, etc; siempre y cuando el navegador que se esté utilizando soporte estas tecnologías.

JSP.

Java Server Pages (JSP – Páginas de Servidor Java) es la tecnología para generar páginas Web de forma dinámica en el servidor, desarrollado por Sun Microsystems basado en script que utilizan una variante del lenguaje Java. La tecnología JSP es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página Web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático. En las JSP se escribe el texto que va a ser devuelto en la salida (normalmente en código HTML) incluyendo código Java dentro de él para poder modificar o generar contenido dinámicamente. **[6]**

Java Script.

JavaScript es un lenguaje de programación interpretado del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Únicamente depende del cliente y no del sistema operativo, sólo necesita un browser capaz de interpretarlo .Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Cualquier persona puede desarrollar aplicaciones escritas en JavaScript del mismo modo que realiza páginas HTML. Por otro lado, JavaScript no es un lenguaje válido para desarrollar aplicaciones concurrentes o de acceso compartido. Está diseñado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web. Este se integra directamente en páginas HTML (como programas propiamente dichos, combinando funciones y sentencias, e introduciendo manejadores de eventos JavaScript en etiquetas HTML) y la ventaja

que presenta sobre el HTML es que permite crear páginas Web más dinámicas, lo que las hace más atractivas para el usuario. **[7]**

HTML

El **HTML** (lenguaje de marcas hipertextuales), es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender. El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato. Uno de los principales problemas es el tiempo que se tarda en escribir un código eficiente HTML a mano. Además los elementos estándar HTML tienen también algunas limitaciones. Dependiendo del navegador que se use, pueden aparecer diferentes para varios usuarios. Esto puede causar alguna confusión si los usuarios se mueven de una estación de trabajo a otra diferente. HTML no es propiamente un lenguaje de programación como PHP, sino un sistema de etiquetas. No presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda. A modo de resumen podemos decir que los lenguajes son solo conocimiento que puede ser adquirido de forma libre. Estos no servirían de nada de no ser por los IDE a los que se hace referencia en el siguiente epígrafe.

Entornos de Programación

Los entornos integrados de desarrollo o IDE (acrónimo de Integrated Development Environment), son aplicaciones compuestas por un conjunto de herramientas útiles para un programador. Estos pueden ser exclusivos para un lenguaje de programación o bien para varios. Estos suelen estar formados por un editor de código, un compilador, un depurador y un constructor de interfaz grafica (GUI).

Zend Studio.

Un programa orientado a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más. La parte del programa que nos permite escribir los scripts es bastante útil para la programación en PHP. La interfaz está compuesta por varias partes, en las que encontramos un explorador de archivos, una ventana de depuración, los menús y otra para mostrar el código de las páginas. Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que vayamos creando nosotros, incluso en páginas que tengamos incluidas con la función include(). Otras ayudas que ofrece a la hora de escribir son las típicas en editores avanzados, como permitir editar varios archivos, y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función, identificación automática del código, etc. Zend Studio dispone de una herramienta muy interesante de debug o depuración. Gracias a ella podemos ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la sesión. Podemos colocar puntos de parada de los scripts y realizar las acciones típicas de depuración. Además de la ventana para visualizar el contenido de las variables, dispone de otras donde muestra la salida del script según se va generando, y otra donde se pueden ver las alertas y errores. Las posibilidades se completan con distintos tipos de depuración, en local, en remoto o a partir de una URL. **[8]**

Quanta.

Quanta es un editor web desarrollado para el escritorio KDE. Soporta una multitud de lenguajes como HTML, JavaScript, CSS, PHP, SQL, XML, ColdFusion, Perl, DTML, Zope o C++, lo que hace que hoy por hoy sea uno de los editores más utilizados en el desarrollo de páginas web. Este editor es un híbrido entre un editor web de texto y uno WYSIWYG ya que nos permite tanto hacer la página viendo el resultado final como directamente desde código. Quanta ofrece multitud de funcionalidades, entre ellas, coloreado de sintaxis para todos los lenguajes soportados, cliente FTP integrado para colgar nuestra página de una manera fácil y rápida, cajas de diálogos contextuales, donde podremos elegir las etiquetas HTML que queremos utilizar, excelente navegador de directorios integrado, con el que accederemos a nuestros ficheros fácilmente, completo panel de previsualización en el que podremos ver nuestro resultado final, validador HTML integrado en la propia aplicación, soporte de extensiones/plugins para añadir funcionalidades extra, y gran integración con el escritorio KDE, al estar desarrollado para él, lo que nos permitirá trabajar más a gusto con la aplicación si este es el escritorio que utilizamos.**[9]**

Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto independiente de una plataforma para desarrollar opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Además es una plataforma de programación, desarrollo y compilación de aplicaciones Java. También posee una atractiva interfaz que lo hace fácil y agradable de usar. El usuario puede si lo desea añadir varios plugins que le permitirán usar Eclipse para otros lenguajes de programación como php. Entre las características con que cuanta están:

• Editor de texto
• Resaltado de sintaxis
• Compilación en tiempo real
• Pruebas unitarias con JUnit
• Control de versiones con Concurrent Versions System (CVS)

• Asistentes (wizards): para creación de proyectos, clases, tests, etc.
• Refactorización

Módulos para Eclipse

Los módulos (plugins) son precisamente una de las ventajas del IDE Eclipse, de algunos de los propuestos para el desarrollo de nuestra aplicación hablaremos a continuación.

PDT

PDT (PHP Development Tools), este proyecto ha tenido una gran respuesta entre los desarrolladores de PHP y que ha sido descargado más de 300.000 veces, dato este que nos da una idea de la aceptación que ha tenido a nivel internacional. Entre las características en la versión actual se encuentran:

• Editor sensible al contexto, el cual provee de resaltamiento de código, asistente de código y autocompletado de código.
• Integración con el modelo del proyecto Eclipse, que permite para inspeccionar el uso de las vistas del contorno del fichero y del proyecto, así como la nueva vista PHP Explorer.
• Soporte para el debug incremental del código de PHP
• Extensos frameworks y APIs que permiten a los desarrolladores e ISVs (vendedores de software independientes) fácilmente extender PDT para crear nuevas e interesantes herramientas orientadas al desarrollo de PHP.

Mylyn

Mylyn es un grupo centrado en la interfaz para Eclipse que reduce la sobrecarga de información y realiza varias tareas fáciles. Para ello, hacer tarea de primera clase de Eclipse, y la integración de los ricos y la edición fuera de línea para los repositorios, como Bugzilla, Trac, y Jira. Una vez que sus tareas se integran, Mylyn supervisa su actividad para identificar la información pertinente para la tarea-a-mano, y utiliza este contexto para centrar la interfaz de usuario de Eclipse en información interesante, ocultar la poco interesantes, y de forma automática encontrar lo que está

relacionado. Esto sitúa a la información que necesita para realizar su trabajo al alcance de su mano y mejora la productividad mediante la reducción de búsqueda, desplazamiento y la navegación. Mylyn también facilita la multitarea, planificación, la reutilización de los esfuerzos anteriores, y el intercambio de conocimientos. **[10]**

Aptana

Aptana es una robusta y avanzada interface de desarrollo Web, enfocado a Java script para el desarrollo de aplicaciones dinámicas. Entre sus principales características se tiene un asistente en código Java script, HTML y CSS; incluyendo sus propias funciones en Java script. **[11]**

Sistemas Gestores de Bases de Datos. (SGBD)

En la actualidad la información es uno de los aspectos más importantes para el conocimiento, y la mayor parte de la información mundial se encuentra almacenada en bases de datos, las cuales poseen características que hacen que se pueda acceder a ellas de forma inmediata para adquirir la información necesaria.

Los Sistemas de Gestión de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Los SGBD tienen a su vez un propósito general que es el de manejar de manera clara, sencilla y ordenada un conjunto de información; además, poseen un objetivo definido que es el de brindar a los usuarios una herramienta que les permita de forma general, manejar los datos de una base de

datos sin conocer la forma en que se almacenan o se accede a los mismos. Los SGBD poseen determinados objetivos específicos dentro de los que se pueden señalar:

• Independencia.
• Consistencia.
• Seguridad.
• Integridad.
• Tiempo de respuesta

Existen numerosos programas que se usan para diseñar, crear, definir y utilizar los ficheros y formularios de una base de datos. Entre ellos se pueden encontrar algunos como: SQL Server, MySQL, ACCESS entre otros como los que a continuación se describen. **[12]**

MySQL

MySQL es un sistema de gestión de base de datos, multi-hilo y multiusuario con más de seis millones de instalaciones. Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes además de ser un sistema de administración de Base de Datos. MySQL es muy rápido, confiable y fácil de usar, es multiplataforma, multiusuario y permite elaborar consultas con el robusto SQL, además no tiene valor monetario, es un software cuya licencia es libre. MySQL es muy utilizado en aplicaciones Web. Su popularidad como aplicación Web está muy ligada al lenguaje de programación Web PHP, que a menudo aparece en combinación con MySQL. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

MySQL es un software de “Fuente Abierta” .Significa que cualquier usuario puede usarlo e incluso modificarlo. Puede además bajar el código fuente de MySQL y usarlo sin pagar. Entre las características disponibles en las últimas versiones y que además son implementadas

únicamente por MySQL se puede destacar: Múltiples motores de almacenamiento permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos. Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo. A modo de resumen se muestran a continuación las principales características de este gestor de bases de datos son las siguientes:

• Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
• Soporta gran cantidad de tipos de datos para las columnas.
• Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
• Gran portabilidad entre sistemas.
• Soporta hasta 32 índices por tabla.
• Gestión de usuarios y contraseña, manteniendo un muy buen nivel de seguridad en los datos.

PostgreSql

Es un potente sistema que maneja bases de datos, diseñado para administrar enormes cantidades de datos, es probadamente confiable, mantiene la integridad de los datos. Corre en los sistemas operativos mas utilizados en el mundo como Linux, Windows y en distintas versiones de Unix. Posee tipos de datos como INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, TIMESTAMP y algunos más. Puede almacenar objetos de gran tamaño incluyendo imágenes, videos y sonido, además tiene una amplia documentación. Mantiene una eficiente consulta de datos y resuelve múltiples problemas empresariales en el mundo actual. PostgreSql es escalable en cuanto a la cantidad de datos que maneja y al número de usuarios concurrentes, puede manejar hasta más de 4 Terabytes de datos. Ha sido reconocido por las industrias y premiado en varias ocasiones. Algunas de sus características más importantes son:

• Escritura adelantada de registros, evitando que se pierdan datos si hay alguna falla de energía, fallos del Sistema Operativo, y fallas de hardware.

• Juegos de caracteres internacionales, codificación de caracteres multi-byte y Unicode;
• Características para la integridad de los datos como: claves primarias, llaves foráneas con capacidad de actualizar en cascada o restringir la acción en caso que existan hijos, restricción check, restricción de unicidad y restricción not null; todas las restricciones se pueden postergar hasta el momento de terminar la transacción.
• Multi-Version Concurrency Control, que sirve para lograr un control de concurrencia tan eficiente que en la gran mayoría de los casos no se requiere de bloqueos.
• Un sofisticado optimizador de consultas, que es capaz de resolver consultas complejas.
• Herencia de tablas (aunque la orientación a objetos no esta completa esta característica se puede utilizar para lograr el particionamiento de tablas en varios discos mediante una técnica llamada "restricciones excluyentes"). PostgreSql puede ser modificado y distribuido en la forma en que se quiera, puesto que su código fuente está disponible bajo la licencia BSD, licencia de código abierto.

SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Es un servidor de bases de datos pensado para gestionar tantos clientes simultáneos como admita la potencia del hardware del equipo en el que esté instalado. Posee una gran ventaja que es la de poder manejar bases de datos con millones de registros y lo hace funcionando con miles de conexiones simultáneas a los datos. Entre sus desventajas se encuentran que está diseñado solamente para Windows o sistemas operativos de Microsoft por lo que se considera que no es multiplataforma, y de ahí su mayor desventaja. Otras desventajas son los altos precios de las licencias y el gran consumo de recursos como la memoria RAM que necesita para ejecutarse correctamente.

Oracle

Es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR) que dispone de potentes herramientas para la gestión y seguridad de los datos como son el motor de la base de datos Oracle y la herramienta Oracle Forms que permite diseñar pantallas de introducción y consulta de datos. Se basa en la tecnología cliente/servidor. Es un gestor de base de datos muy usado y algunas de sus características son:

• Seguridad en el acceso a los datos mediante la gestión de privilegios.

• Copias de seguridad. Oracle proporciona mecanismos para realizar copias de seguridad de los datos y su recuperación.

• Conectividad. Se puede acceder a datos de Oracle desde software de otro fabricante.

Es un producto vendido a nivel internacional, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales. **[13]**

Estilos arquitectónicos.

¿Qué es un estilo arquitectónico?

Cuando se habla del término de estilo arquitectónico en la arquitectura de software no queda más remedio que establecer una comparación entre estilos arquitectónicos de la arquitectura civil, por ejemplo: estilos eclécticos, barrocos, clásicos, etc. Los estilos de la arquitectura de software definen un conjunto de concepciones arquitectónicas comunes que identifican un momento en el desarrollo de la arquitectura. Algunos de los principales estilos arquitectónicos que se usan en la actualidad están divididos por Clases de Estilos las que engloban una serie de estilos arquitectónicos específicos. Entre estos estilos numeran:

• Estilos de Flujo de datos

• Estilos centrados en datos

• Estilos de Llamada y Retorno

• Estilo de Código Móvil

• Estilos Peer – To – Peer

Cada uno de estos estilos contiene una serie de especializaciones, las que a continuación se hablan pertenecen al grupo de las de llamada y retorno.

Modelo – Vista – Controlador (MVC)

Se utiliza principalmente cuando es necesario modularizar la interfaz de usuario, las reglas de negocio y el control de eventos. Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas que tendrán funcionalidades propias como se podrá ver a continuación.

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista y responde a instrucciones de cambiar el estado habitualmente desde el controlador). Mantiene el conocimiento del sistema. No depende de ninguna vista y de ningún controlador.

Vista: Maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

Controlador: Proporciona significado a las ordenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

Ventajas:

• Se puede mostrar distintas variantes de interfaz gráfica simultáneamente.

• La interfaz tiende a cambiar más rápido que las reglas del negocio. Agregar nuevos tipos de vista no afecta el modelo.

• Evita poner el código indebido en la capa impropia. Facilita despliegue en caso de modificaciones en el modelo de datos.

Desventajas:

• Puede aumentar un poco la complejidad de la solución. Como está guiado por eventos puede ser algo más difícil de depurar.
• Si hay demasiados cambios en el modelo la vista puede provocar un constante refrescamiento de las vistas, a menos que se prevea programáticamente.

Arquitectura en Capas

Resulta útil cuando se pueden identificar distintas clases de servicios que pueden ser articulados jerárquicamente. Los componentes de cada capa consisten en conjuntos de clases. Las interacciones entre las capas ocurren generalmente por invocación de métodos, por definición los niveles de abajo no deben poder utilizar funcionalidad ofrecida por los de niveles superiores.

Ventajas:

• Modularidad del sistema.
• Facilita la localización de errores.
• Mejora soporte del sistema

Desventajas:

• Puede ser difícil definir que componentes ubicar en cada una de las capas.

Estos patrones rigen muchas de las aplicaciones actuales, ejemplo del uso de estos se puede apreciar en los Framework, de los que se hará un análisis de los mas usados hoy en día con el objetivo de seleccionar uno para la futura implementación del sistema.

Framework

Un 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. Entre otras cosas un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework

facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. En la actualidad existen diferentes framework tales como Symfony, CakePHP, entre otros. De las características particulares de cada una de estos se hará un estudio con el propósito de determinar cual es el más apropiado para la aplicación. **[14]**

CakePHP

CakePHP es un framework de desarrollo rápido de aplicaciones de código abierto en PHP. Inspirado en Rails, un framework para la construcción de sitios web que utilizan una base de datos como fuente de recursos, posee una infraestructura que tiene como finalidad, permitir el desarrollo de aplicaciones web de manera ágil y estructurada, sin perder flexibilidad. Entre las características más destacables de CakePHP se incluyen:

• Arquitectura basada en el patrón Modelo Vista Controlador (MVC) y orientada a objetos:
define clases modelo, vista y controlador con funcionalidades básicas y de las cuales heredan todas las clases que se ajustan a este patrón y que son usadas en la aplicación construida con el framework.

• Una comunidad activa de usuarios: creada tras la publicación del framework en 2005 y que ha contribuido a mejorar el framework (a través de subproyectos específicos en CakeForge.Org) y difundir su uso.

• Compatible con PHP4 y PHP5: aunque en PHP4 se requiere especificar algunos parámetros de configuración adicionales en las clases a implementar.

Zend Framework

Zend Framework destaca el hecho de que no sólo busca facilitar la programación a través del patrón MVC, sino también automatizar tareas más específicas, como el acceso a base de datos, el filtrado de datos ingresados a la aplicación o la búsqueda en un sitio web ordenando resultados por relevancia. Entre sus metas se encuentran:

• Proveer un repositorio de componentes de alta calidad y que cuenten con soporte activo.

• Proveer un sistema completo para el desarrollo de aplicaciones web elaboradas en PHP5.

• Facilitar el aprendizaje en el uso del framework sin tener que aprender un nuevo lenguaje de programación.

• Organizar la colaboración de la comunidad para una programación avanzada en PHP5.

Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web4.

Symfony está desarrollado completamente con PHP 5. Además es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle entre otros. Es multiplataforma pues se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se mencionan algunas de las principales características de Symfony.

• Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)

• Independiente del sistema gestor de bases de datos.

• Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

• Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.

• Sigue la mayoría de mejores prácticas y patrones de diseño para la web.

• Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

• Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
• Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
[15]

Metodología de desarrollo de software.

Según Piattini (1996), no hay un consenso entre los autores sobre el concepto de metodología, y por lo tanto no existe una definición universalmente aceptada. Sí hay un acuerdo en considerar a la metodología como "*un conjunto de pasos y procedimientos que deben seguirse para el desarrollo del software".*
Maddison (1983) define metodología como un *"conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información".*
Por lo tanto, una metodología es un conjunto de componentes que especifican:

- Cómo se debe dividir un proyecto en etapas.
- Qué tareas se llevan a cabo en cada etapa.
- Qué salidas se producen y cuándo se deben producir.
- Qué restricciones se aplican.
- Qué herramientas se van a utilizar.
- Cómo se gestiona y controla un proyecto.

Generalizando, Piattini llega a la definición de metodología de desarrollo como *"un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software"*. Normalmente consistirá en fases o etapas descompuestas en subfases, módulos, etapas, pasos, etc. Esta descomposición ayuda a los desarrolladores en la

elección de las técnicas a utilizar en cada estado del proyecto, facilitando la planificación, gestión, control y evaluación de los proyectos.
Sintetizando lo anterior, el autor dice que: *"una metodología representa el camino para desarrollar software de una manera sistemática".*

La necesidad de una metodología de desarrollo

Para desarrollar un proyecto de software es necesario establecer un enfoque disciplinado y sistemático. Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaboran a partir del marco definido por uno o más ciclos de vida.
Las metodologías persiguen tres necesidades principales:

- Mejores aplicaciones, tendientes a una mejor calidad, aunque a veces no es suficiente.
- Un proceso de desarrollo controlado, que asegure uso de recursos apropiados y costo adecuado.
- Un proceso estándar en la organización, que no sienta los cambios del personal.

Las metodologías a veces tienen diferentes objetivos, pero los más representativos pueden ser:

- Brindar un método sistemático, de modo de controlar el progreso del desarrollo.
- Especificar los requerimientos de un software en forma apropiada.
- Construir productos bien documentados y de fácil mantenimiento.

- Ayudar a identificar las necesidades de cambio lo más pronto posible.

- Proporcionar un sistema ágil que satisfaga a todas las personas involucradas.

Los procesos se descomponen hasta el nivel de tareas o actividades elementales, donde cada tarea está identificada por un procedimiento que define la forma de llevarla a cabo. Para aplicar un procedimiento se pueden usar una o más técnicas. Estas pueden ser gráficas con apoyos textuales, formales y determinan el formato de los productos resultantes en la tarea.
Para llevar a cabo las tareas se pueden usar herramientas software que automatizan la aplicación en determinado grado. **[16]**

Ventajas de tener una metodología.

Mejora de los procesos de desarrollo.

- Todas las personas del proyecto trabajan bajo un marco común.
- Estandarización de conceptos, actividades y nomenclaturas.
- Actividades de desarrollo apoyadas por procedimientos y guías.
- Resultados del desarrollo predecibles.
- Uso de herramientas de ingeniería software.
- Planificación de actividades en base a un conjunto de tareas definidas y a la experiencia en otros proyectos.
- Recopilación de mejores prácticas para proyectos futuros.

Mejora de los productos.

- Se asegura que los productos cumplen con los objetivos de calidad propuestos.
- Detección temprana de errores.
- Se garantiza la trazabilidad de los productos a lo largo del desarrollo.

Mejora de las relaciones con el cliente.

- El cliente percibe el orden en nuestros procesos.
- Facilita al cliente el seguimiento de la evolución del proyecto.
- Se establecen mecanismos para asegurar que los productos desarrollados cumplen con las expectativas del cliente.

Rational Unified Process (RUP)

Es un proceso de Ingeniería del Software que proporciona una visión disciplinada para la asignación de tareas y responsabilidades en las organizaciones de desarrollo de software. RUP integra un conjunto de "buenas prácticas" para el desarrollo de software en un marco de procesos válido para un rango amplio de tipos de proyectos y organizaciones. Las principales buenas prácticas cubiertas son:

- Desarrollo iterativo.
- Gestión de requisitos.
- Uso de arquitecturas basadas en componentes.
- Uso de técnicas de modelado visual.
- Verificación continua de la calidad.
- Gestión y control de cambios.

El proceso propuesto por RUP posee dos dimensiones:

– la primera, representa el aspecto dinámico del proceso, y está expresado en términos de ciclos, fases, iteraciones e hitos;
– la segunda, representa el aspecto estático, que se describe en términos de componentes, actividades, flujos de trabajo, artefactos, y actores.

Como se mencionó anteriormente en su visión estática, el modelo RUP está compuesto por:

- Roles: analista de sistemas, diseñador, diseñador de pruebas, roles de gestión y roles de administración.
- Actividades: RUP determina el trabajo de cada rol a través de actividades. Cada actividad del proyecto debe tener un propósito claro, y se asigna a un rol específico. Las actividades pueden tener duración de horas o de algunos días; y son elementos base de planificación y progreso.
- Artefactos: Son los elementos de entrada y salida de las actividades. Son productos tangibles del proyecto. Las cosas que el proyecto produce o usa para componer el producto final (modelos, documentos, código, ejecutables…)
- Disciplinas: son "contenedores" empleados para organizar las actividades del proceso. RUP comprende 6 disciplinas técnicas y 3 de soporte:
  Técnicas: modelado del negocio, requisitos, análisis y diseño, implementación, pruebas y desarrollo.
  Soporte: gestión de proyecto, gestión de configuración y cambio, y entorno.
- Flujos de trabajo: son el "pegamento" de los roles, actividades, artefactos y disciplinas, y constituyen la secuencia de actividades que producen resultados visibles.

En su visión dinámica, la visión de la estructura del ciclo de vida RUP se basa en un desarrollo iterativo, jalonado por hitos para revisar el avance y planear la continuidad o los posibles cambios de rumbo.

RUP divide el proceso de desarrollo en ciclos, teniendo una versión del producto al final de cada ciclo.

Cuatro son las fases que dividen el ciclo de vida de un proyecto RUP, las cuales concluyen con un hito bien definido donde deben tomarse ciertas decisiones:

1.- Inicio. Es la fase de la idea, de la visión inicial de producto, su alcance. El esbozo de una arquitectura posible y las primeras estimaciones. Concluye con el "hito de objetivo".

2.- Elaboración. Comprende la planificación de las actividades y del equipo necesario. La especificación de las necesidades y el diseño de la arquitectura. Termina con el “hito de Arquitectura”.

3.- Construcción. Desarrollo del producto hasta que se encuentra disponible para su entrega a los usuarios. Termina con el “hito del inicio de la capacidad operativa”.

4.- Transición. Traspaso del producto a los usuarios. Incluye: manufactura, envío, formación, asistencia y el mantenimiento hasta lograr la satisfacción de los usuarios. Termina con el “hito de entrega del producto”.

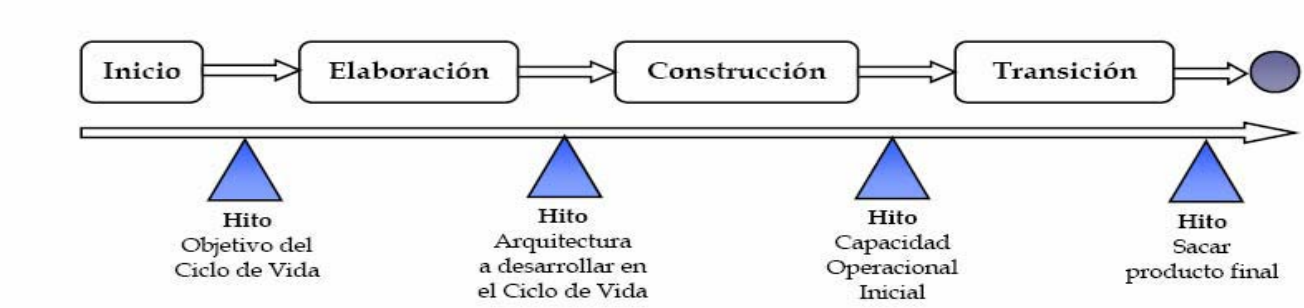


Figura 1: Fases e Hitos de RUP.

Cada fase se subdivide a su vez en iteraciones, siendo el número de iteraciones en cada una de ellas variable.

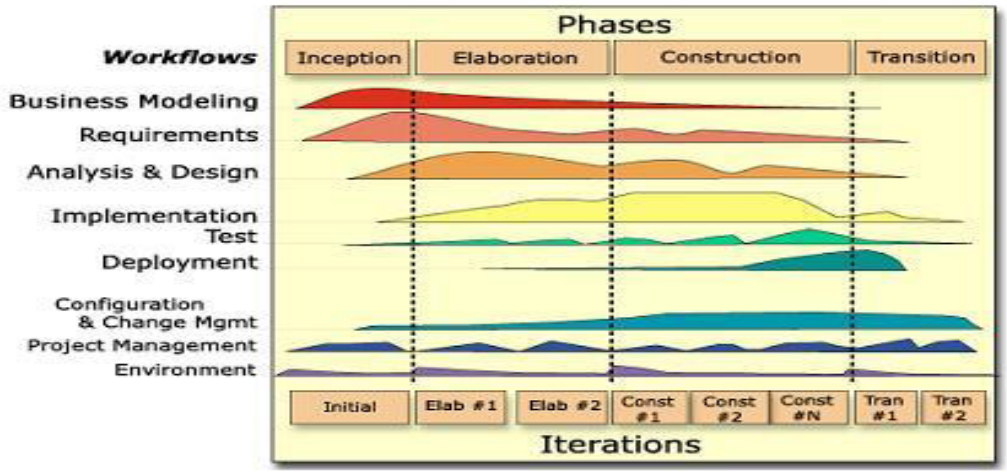


Figura 2: Fases e Iteraciones de la Metodología RUP.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

El Proceso Unificado está dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

Un proceso dirigido por casos de uso.

Los casos de uso han sido adoptados casi universalmente para la captura de requisitos de sistemas software en general, y de sistemas basados en componentes en particular, pero los casos de uso son mucho más que una herramienta para capturar requisitos. Dirigen el proceso de desarrollo en su totalidad.

¿Por qué casos de uso?

Existen varios motivos por los cuales los casos de uso son buenos, se han hecho populares y se han adoptado universalmente. Las razones fundamentales son:

- Proporcionan un medio sistemático e intuitivo de capturar requisitos funcionales centrándose en el valor añadido para el usuario.
- Dirigen todo el proceso de desarrollo debido a que la mayoría de las actividades como el análisis, diseño y prueba se llevan a cabo partiendo de los casos de uso. El diseño y la prueba pueden también planificarse y coordinarse en términos de casos de uso. Esta característica es aun más evidente cuando la arquitectura se ha estabilizado en el proyecto, después del primer conjunto de iteraciones.

Un proceso centrado en la arquitectura.

Los casos de uso solamente no son suficientes. Se necesitan más cosas para conseguir un sistema de trabajo. Esas “cosas” son la arquitectura. Se puede pensar que la arquitectura de un sistema es la visión común en la que todos los empleados (desarrolladores y otros usuarios)

deben estar de acuerdo, o como poco, deben aceptar. La arquitectura da una clara perspectiva del sistema completo, necesaria para controlar el desarrollo.

¿Por qué es necesaria la arquitectura?

Un sistema software grande y complejo requiere de una arquitectura para que los desarrolladores puedan progresar hasta tener una visión común. Se necesita una arquitectura para:

- Comprender el sistema.
- Organizar el desarrollo.
- Fomentar la reutilización.
- Hacer evolucionar el sistema.

Un proceso iterativo e incremental.

Como se indicó anteriormente, dos de las tres claves del Proceso Unificado son que el proceso de desarrollo software debería estar dirigido por los casos de uso y centrado en la arquitectura. Estos aspectos tienen un impacto técnico evidente sobre el producto del proceso. Conseguir el equilibrio correcto entre los casos de uso y la arquitectura es algo muy parecido al equilibrado de la forma y la función en el desarrollo de cualquier producto. Se consigue con el tiempo. Durante el más corto proceso de desarrollo de software, los desarrolladores elaboran conscientemente este equilibrio (entre casos de uso y arquitectura) a lo largo de una serie de iteraciones. Por tanto, la técnica de desarrollo iterativo e incremental constituye el tercer aspecto clave del Proceso Unificado.

La tercera clave proporciona la estrategia para desarrollar un producto software en pasos pequeños manejables:

• Planificar un poco.

• Especificar, diseñar, e implementar un poco.

• Integrar, probar, y ejecutar un poco cada iteración.

¿Por qué un desarrollo iterativo e incremental?

En dos palabras: para obtener un software mejor. Dicho con unas cuantas palabras más, para cumplir los hitos principales y secundarios con los cuales se controla el desarrollo. Y dicho con algunas palabras más:

- Para tomar las riendas de los riesgos críticos y significativos desde el principio.
- Para poner en marcha una arquitectura que guíe el desarrollo del software.
- Para proporcionar un marco de trabajo que gestione de mejor forma los inevitables cambios en los requisitos y en otros aspectos.
- Para construir el sistema a lo largo del tiempo en lugar de hacerlo de una sola vez cerca del final, cuando el cambiar algo se ha vuelto costoso.
- Para proporcionar un proceso de desarrollo a través del cual el personal puede trabajar de manera más eficaz.

Es importante destacar que la Metodología RUP es más adaptable para proyectos de largo plazo. En resumen, la meta de RUP es:

- asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.
- proveer un enfoque disciplinado para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.
- reducir en gran medida el riesgo que representa la construcción de sistemas complejos, porque evoluciona de forma incremental partiendo de sistemas más pequeños en los que ya se tiene confianza.

Programación Extrema (XP).

Es una de las metodologías de desarrollo de software ágil más reconocidas en la actualidad, fue creada por Kent Beck, dirigida a equipos pequeños o medianos (Beck sugiere que el tamaño de los equipos se limite de 3 a 20 como máximo, otros dicen no más de 10 participantes).

Esta metodología consiste en una programación ágil o extrema, utilizada para proyectos de corto plazo. Consiste en desarrollos rápidos e iterativos, cuya particularidad es tener como parte del equipo al usuario final, este es uno de los requerimientos para llegar al triunfo del proyecto.

Está organizada en 4 cuatro fases: Planificación, Diseño, Desarrollo y Pruebas.

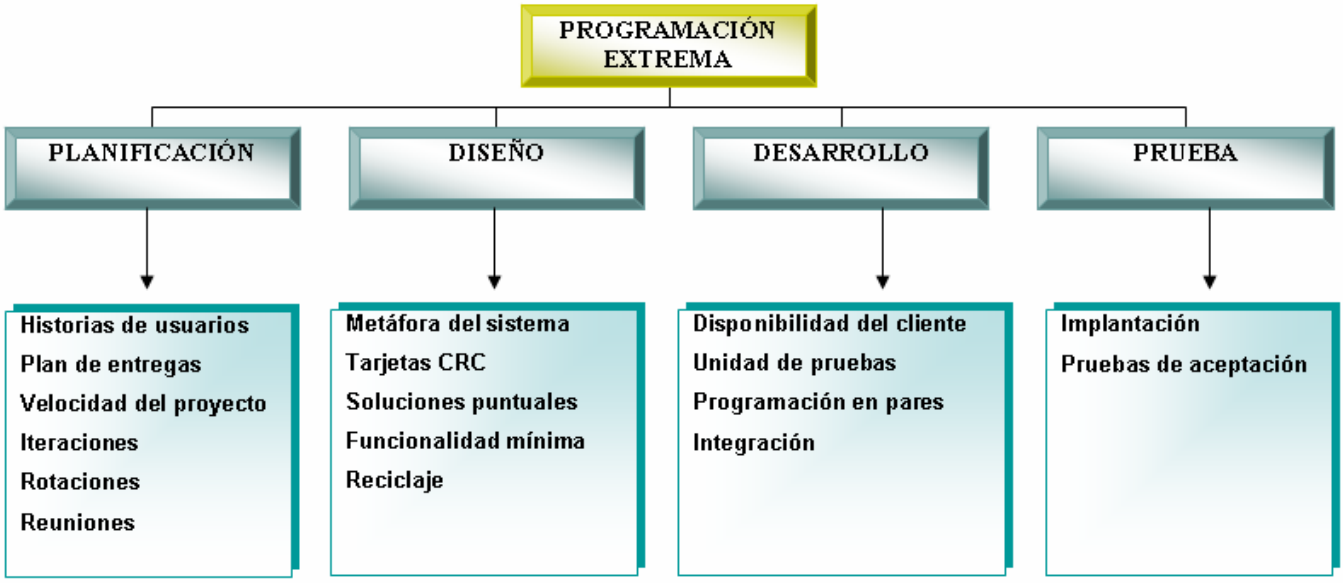


Figura 3: Fases de Programación Extrema.

Posee cuatro variables principales que son:

- Coste: Máquinas, especialistas y oficinas
- Tiempo: Total y de Entregas
- Calidad: Externa e Interna
- Alcance: Intervención del cliente

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en

realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

La programación extrema pone la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando escriben su código de producción. Las pruebas se integran en un proceso continuo de construcción, lo que rinde una plataforma altamente estable para el desarrollo futuro. Constituyendo en la actualidad un proceso de diseño evolutivo que se basa en reorganizar un sistema simple en cada iteración, todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado, combinando la disciplina con la adaptabilidad de una manera que la hace la más desarrollada entre todas las metodologías adaptables.

Características generales de XP

Una de las características distintivas de XP (que comparte con otras metodologías ágiles) es que de alguna manera representa la antítesis de lo que es el tradicional proceso de desarrollar software. XP es deliberadamente una metodología “liviana” que pasa por alto la utilización de elaborados casos de uso, la exhaustiva definición de requerimientos y la producción de una extensa documentación. Todo lo anterior puede parecer caótico según el enfoque tradicional de la ingeniería de software, aunque no hay que olvidar que XP tiene asociado un ciclo de vida y es considerado a su vez un proceso. La tendencia de entregar software en lapsos cada vez menores de tiempo y con exigencias de costos reducidos y altos estándares de calidad, hace que XP sea una opción a considerar.

Otra característica que distingue a XP en la práctica es que, típicamente, los que se encargan de introducirla en los ambientes de trabajo son los propios encargados del desarrollo y sus equipos de programación, a diferencia de lo que sucede con otras metodologías, las cuales normalmente

se introducen a un nivel corporativo y gradualmente se van bajando hasta alcanzar a los equipos de desarrollo.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua
- El manejo del cambio se convierte en parte sustantiva del proceso
- El costo del cambio no depende de la fase o etapa
- No introduce funcionalidades antes que sean necesarias
- El cliente o el usuario se convierte en miembro del equipo

Derechos del Cliente

- Decidir que se implementa
- Saber el estado real y el progreso del proyecto
- Añadir, cambiar o quitar requerimientos en cualquier momento
- Obtener lo máximo de cada semana de trabajo
- Obtener un sistema funcionando cada 3 o 4 meses

Derechos del Desarrollador

- Decidir como se implementan los procesos
- Crear el sistema con la mejor calidad posible
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos
- Estimar el esfuerzo para implementar el sistema
- Cambiar los requerimientos en base a nuevos descubrimientos

Prácticas XP.

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo

funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas.

1. La planeación, en la cual la opinión del cliente y del equipo de desarrollo deben fusionarse como un todo coherente.

2. Entregas en iteraciones pequeñas, que permitan al cliente utilizar el sistema con las funcionalidades mínimas lo antes posible, e irlo complementando gradual y continuamente.

3. Manejo de metáforas, donde la metáfora ayuda a que todo el equipo de desarrollo entienda los elementos básicos del sistema y las relaciones entre ellos sin la necesidad de una arquitectura muy elaborada y detallada.

4. Diseño simple, donde siempre se intenta tener el código más simple, menos redundante y con las funcionalidades estrictamente necesarias en el presente.

5. Pruebas continuas, donde es imperativo que los programadores escriban pruebas por cada unidad de código y que el cliente participe en el diseño de pruebas funcionales.

6. Refabricación, donde este concepto se refiere a mantener una depuración y simplificación constante del sistema. Una vez que se ha añadido alguna funcionalidad, es necesario revisar y ser críticos para encontrar puntos de simplificación del código.

7. Programación en pares, donde toda la codificación debe hacerse en parejas de programadores, cada pareja compartiendo un mismo monitor y teclado. El que utiliza el teclado piensa en la mejor manera de implementar alguna funcionalidad, el otro piensa estratégicamente, cuestionando si se puede simplificar, anticipando pruebas, o preguntándose si el enfoque es el adecuado o si debe descartarse código y replantear el problema. Se alienta la rotación continua de programadores en los pares, combinando diferentes niveles de experiencia y de conocimiento del código.

8. Propiedad colectiva del código. Aquí se dice que el código puede ser modificado por cualquier elemento del equipo de programación, esto es, que no hay propiedad individual de algún programador sobre alguna sección de código. Se asume que al seguir las prácticas de XP, después de un tiempo razonable, cualquier programador conoce todo el código, principalmente por el hecho de la programación en pares.

9. Integración continua (mantener una sola revisión para todo el equipo de programación), tan pronto como pequeños “saltos” en la versión actual sean concluidos. Se recomiendan lapsos entre integraciones de pocas horas a no más de un día.
10. Semana de 40 horas. XP afirma que las condiciones de trabajo óptimas para programar “a máxima velocidad” es solo en un turno normal (8 horas). En este sentido las horas extras o fines de semana trabajados solo desgastan al equipo de programación, afectan el rendimiento y generan un ambiente propicio para cometer errores, ser displicentes en el apego a las normas y producir software de mediocre calidad.
11. El cliente debe estar disponible localmente, donde un representante del cliente debe permanecer por turnos completos en el sitio de programación para contestar cualquier pregunta y ayudar en el desarrollo de pruebas funcionales.
12. Mantener estándares de codificación entre los programadores. Esto es esencial para la programación en pares y para la propiedad colectiva del código.

El enfoque de XP respecto a estas doce prácticas es que se debe aprovechar la sinergia de todas ellas cuando se adoptan como un todo. Esto es, que si se elige implementar solamente algunas, todo el enfoque metodológico corre el riesgo de fracasar.
La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo. **[17]**

Herramientas CASE de modelado de UML

Las Herramientas para el modelado visual son también un aspecto importante a tener en cuenta en el desarrollo del presente trabajo. Existen varias y cada una de ellas con una serie de características que le permiten destacarse o no en dependencia de la línea de trabajo en la que se quiera insertar este tipo de herramientas. Dentro del uso de estas se estable como lenguaje de

modelado el UML el cual es el más utilizado y conocido en la actualidad, este no es más que un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

Rational Rose

Es una herramienta propietaria para el modelado visual, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida de desarrollo de software. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Características:

• Mantiene la consistencia de los modelos del sistema software.

• Chequeo de la sintaxis UML.

• Genera la documentación automáticamente.

• Generación de código a partir de los modelos.

• Ingeniería inversa (crear modelo a partir de código).

Visual Paradigm

Es una herramienta CASE que da soporte al modelado visual con UML 2.0, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Visual Paradigm ofrece:

• Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.

• Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

• Capacidades de ingeniería directa (versión profesional) e inversa.

• Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

• Disponibilidad de integrarse en los principales IDEs.
• Disponibilidad en múltiples plataformas.

Decisiones y soluciones técnicas.

En el presente epígrafe se mostrarán las decisiones tomadas respecto a las herramientas que se utilizarán para el futuro desarrollo del sistema y se establecerá una comparación en cada uno de los casos con el objetivo de dejar claro la razón de la decisión tomada.

Metodología de desarrollo de software

RUP es la metodología propuesta para desarrollar el modelado del sistema, ya que posee un alto nivel organizacional, permite la asignación de roles y responsabilidades de forma disciplinada y cada una de sus fases está bien documentada dentro del desarrollo de software. Este proceso de Ingeniería del Software aumenta la productividad de los desarrolladores mediante un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo. Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos.

Además RUP es más adaptable para proyectos de largo plazo, dentro de sus características principales se encuentran que está dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental, las mismas son puntos claves en el desarrollo de software. Es una guía de cómo utilizar de manera efectiva *UML,* que es el lenguaje de modelado seleccionado.

Lenguaje de programación

Se utilizará PHP como lenguaje de programación para la implementación del sistema, debido a que es un lenguaje "open source" interpretado de alto nivel embebido en páginas HTML, y su código es ejecutado en el servidor. Este lenguaje nos permite no solo leer, sino también manipular datos desde diversas fuentes, incluso los datos que pueden ser ingresados por los usuarios desde formularios HTML. El código PHP es mucho más legible que el de PERL, este

último lenguaje consume muchos recursos de la máquina haciéndolo menos ligero. PHP no es el único lenguaje de programación cliente/servidor Web que existe, entre las alternativas más conocidas esta ASP por parte de Microsoft, el cual corre bajo el servidor Web propietario de Microsoft: IIS, debido a esto es de pago, por lo que es muy poco usado y aunque es muy potente su metodología de programación es poco robusta y no logra superar a PHP como lenguaje de programación Web. JSP por parte de Java, es la otra alternativa desarrollada por Sun Microsystems, este se suele usar más para scripts de línea de comando, mientras que PHP se usa más para páginas dinámicas. PHP a la hora de servir páginas Web es más eficaz que JSP, ya que se integra como módulo en Apache, y así se evita la carga del intérprete en memoria cada vez que se ejecuta una página. PHP al ser una herencia positiva de Perl facilita al programador muchas cosas que en JSP son más difíciles de programar. Por ejemplo, el acceso a bases de datos es mucho más sencillo en PHP, ya que el API de MySQL, por ejemplo, viene integrado en el propio lenguaje.

Sistema Gestor de Base de Datos

Se utilizará MySQL como gestor de base de datos, primeramente porque está desarrollada bajo la filosofía de código abierto. A diferencia de PostgreSql su principal objetivo de diseño fue la velocidad, y consume muy pocos recursos, tanto de CPU como de memoria, mientras que PostgreSql es lento, pesado, y consume bastante recursos.

Una de las ventajas que tiene MySQL es que a partir de la versión 3.23.19 está siendo desarrollado bajo la licencia GPL, a diferencia de SQL Server que posee como desventaja altos costos de licencia y al no ser multiplataforma solo esta disponible en Sistemas Operativos de Microsoft.

Conclusiones del capítulo

En el presente capítulo se abordaron una serie de conceptos que fundamentan todo lo referente a la parte teórica de la investigación. Además se han descrito una serie aspectos relacionados con la información y el control de la misma dentro de un sistema web, así como los lenguajes de programación, las tecnologías, los sistemas de gestión de bases de datos y la metodología a utilizar para guiar el desarrollo de la aplicación, lo cual ha permitido realizar una propuesta del sistema que compete a esta investigación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

En este capítulo se hace una descripción de las características del sistema donde se da a conocer cada uno de los elementos que serán utilizados para desarrollar la aplicación, así como el flujo de trabajo con el que se determina la solución al problema planteado. Además se describe el Modelo de Negocio, definiendo conceptos que son muy útiles para entender el contexto del problema. Se enumeran los requisitos no funcionales y funcionales, estos últimos se estructuran mediante los Casos de Uso del sistema, de los cuales se ofrece una descripción textual, para brindar una mayor interpretación de los mismos.

Flujo actual del proceso del negocio.

En el Poder Popular Municipal de Cienfuegos se realiza la tarea de recolección de inquietudes provenientes de la población, para a partir de estas trazar estrategias y metas para el perfeccionamiento del bienestar social. Estos datos son recogidos específicamente en el departamento de planteamientos, aquí labora la compañera encargada del acopio de los datos, ella es la Secretaria general de la Asamblea, quien además maneja y controla el flujo de la información. En cada consejo popular se realizan asambleas donde se identifican los planteamientos provenientes de las áreas y circunscripciones, de aquí estos son enviados por los delegados al área de planteamientos de la Asamblea municipal y una vez aquí la secretaria general se encarga de almacenar la información así como clasificarla, identificar a que entidad administrativa le corresponde el planteamiento , la cual debe responder por su solución en caso de que exista, decir si es con recursos o no, a que nivel de solución pertenece, municipal provincial o nacional, si ya fue solucionado ,esta pendiente o no tiene solución , además debe gestionar una serie de estadísticas e informaciones necesarias para los trabajadores del Poder Popular , los cuales podrán tener acceso a la información gestionada por la secretaria.Teniendo en cuenta lo descrito anteriormente se identifica como proceso del negocio la gestión de la información en el área de planteamiento del Poder Popular en el municipio Cienfuegos.

Reglas del negocio

-Solo tendrán acceso los trabajadores del Poder Popular.

-Si es trabajador del Poder Popular tendrá que registrarse para poder acceder a la información.

-Solo puede modificar la información la secretaria general quien jugará el papel de administrador y gestionará la información.

-Los trabajadores del Poder Popular solo podrán consultar la información que requieran pero sin posibilidades de modificarla.

-La información que va a ser gestionada solo podrá ser enviada por los delegados.

Modelo del negocio.

Un modelo del negocio (también llamado diseño del negocio) es el mecanismo por el cual un negocio trata de de generar ingresos y beneficios. Es un resumen de cómo una compañía planifica servir a sus clientes.

De ahí, que en el campo del software también resulte útiles la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio. Los objetivos de la modelación del negocio son:

• Comprender le estructura y la dinámica de la organización en la cual se va a implantar un sistema.

• Comprender los problemas actuales de la organización e identificar las mejoras potenciales.

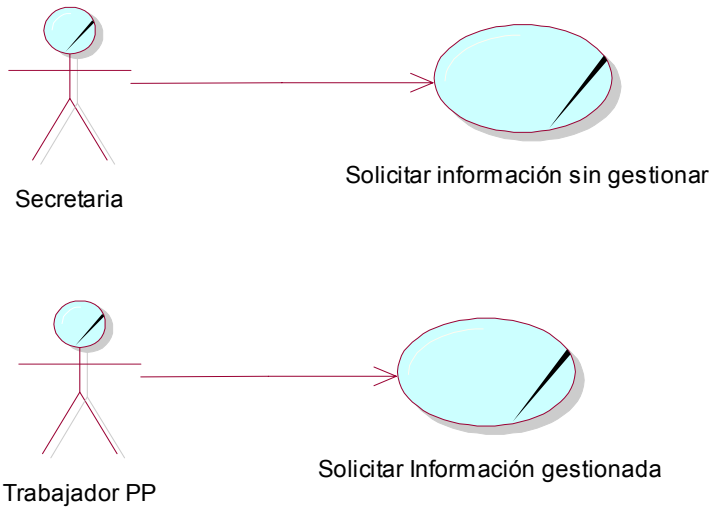
• Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.

• Derivar los requerimientos del sistema que va a soportar la organización.

Diagrama de casos de uso del negocio

Figura 4

Diagrama de Casos de Uso del Negocio



Actores del negocio

Tabla No1.

Actor del negocio	Justificación
Trabajador del Poder Popular	Es el individuo que solicita y se beneficia con la información.
Secretaria General	Es la encargada de solicitar información aún no gestionada a los delegados beneficiándose de la misma.

Trabajador del negocio

Tabla No2.

Trabajador del negocio	Justificación
Secretaria general	Es la encargada de brindar la información solicitada por los trabajadores del Poder Popular.
Delegado	Es el responsable de enviar la información a la secretaria general de la asamblea.

Descripción textual de los casos de uso del negocio.

Tabla No3.

Caso de uso	Solicitar información sin gestionar.
Actores	Secretaria general.
Propósito	Obtener información proporcionada por el delegado para clasificarla.
Resumen El caso de uso comienza cuando la secretaria solicita la información al delegado que se presenta para entregar la información obtenida en los consejos populares, la secretaria obtiene la información, la revisa y clasifica para luego archivarla, terminando así el caso de uso.	
Curso normal de evento	
Acción del actor	**Respuesta del negocio**
1. La secretaria solicita la información.	2. El delegado entrega la información.

	3. La secretaria recepciona y revisa la información.
	4. Clasifica la información.
	5. Archiva la información.
Curso Alternativo de eventos	
Prioridad	Alta
Mejoras	

Tabla No4.

Caso de uso	Solicitar información
Actores	Trabajador del Poder Popular (inicia)
Propósito	Pedir información que va a ser proporcionada por la secretaria.
Resumen El caso de uso comienza cuando el trabajador del Poder Popular solicita una información determinada a la secretaria general. Esta verifica si el trabajador posee permiso para acceder a la información solicitada. Si no lo tiene niega la información; en caso contrario comprueba que la información exista y la entrega, el trabajador revisa la información.De esta forma la secretaria general facilita la información y termina el caso de uso.	
Curso normal de evento	
Acción del actor	**Respuesta del negocio**
1. El trabajador del Poder Popular solicita información relacionada con los planteamientos.	2. La secretaria verifica si tiene acceso el trabajador.

	3. Si tiene acceso, la secretaria revisa si existe la información solicitada.
	4. Si existe la información solicitada la secretaria la entrega.
5. El trabajador recibe la información y la revisa.	
Curso Alternativo de eventos	
Acción 3	Si el trabajador no tiene acceso a la información, se deniega y termina el caso de uso.
Acción 4	Si no existe la información la secretaria lo informa y termina el caso de uso.
Prioridad	Alta
Mejoras	

Diagrama de actividades

Figura 5

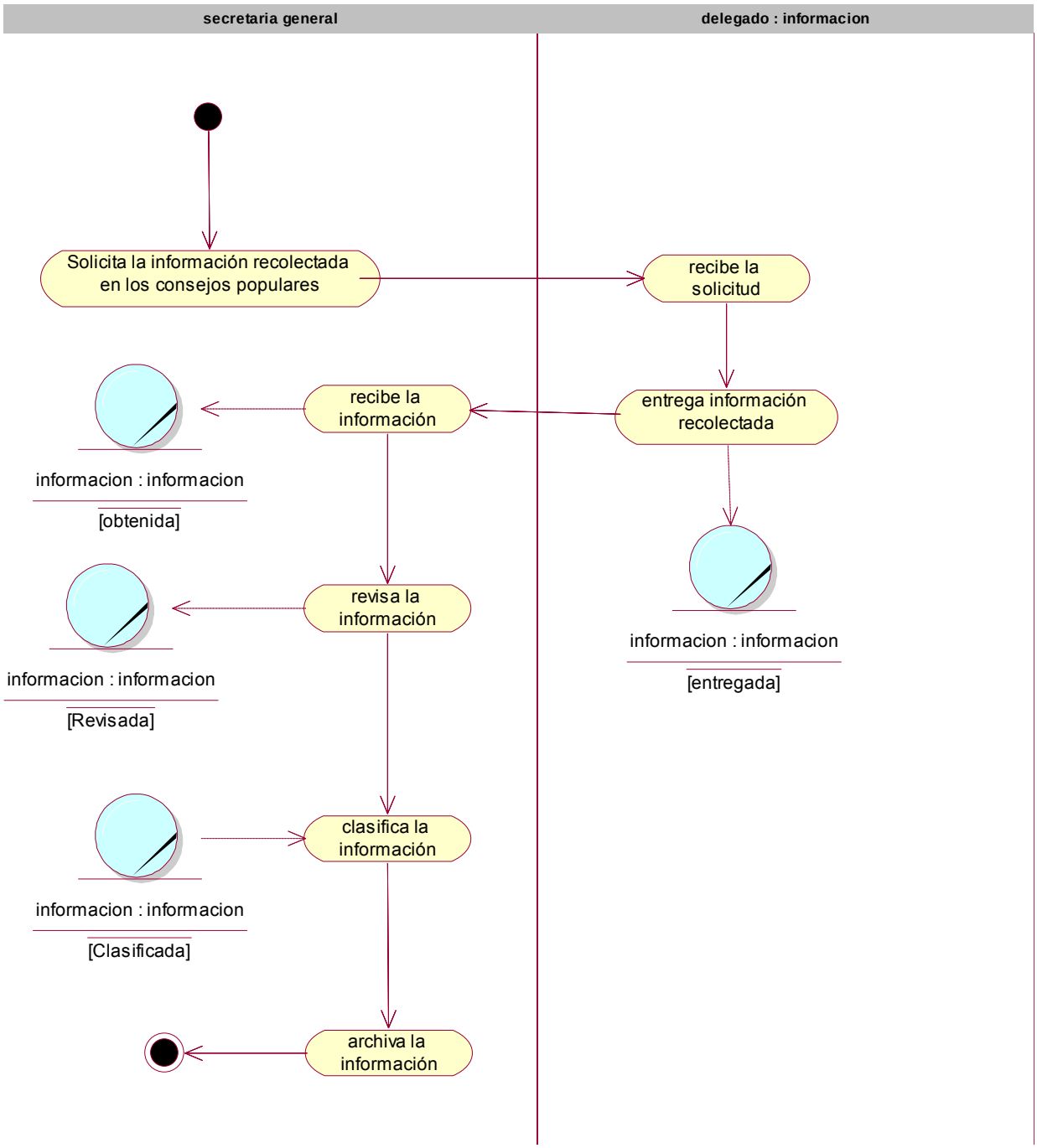

secretaria general
delegado : informacion
Solicita la información recolectada en los consejos populares
recibe la solicitud
recibe la información
entrega información recolectada
informacion : informacion
[obtenida]
revisa la información
informacion : informacion
[entregada]
informacion : informacion
[Revisada]
clasifica la información
informacion : informacion
[Clasificada]
archiva la información

Figura 6

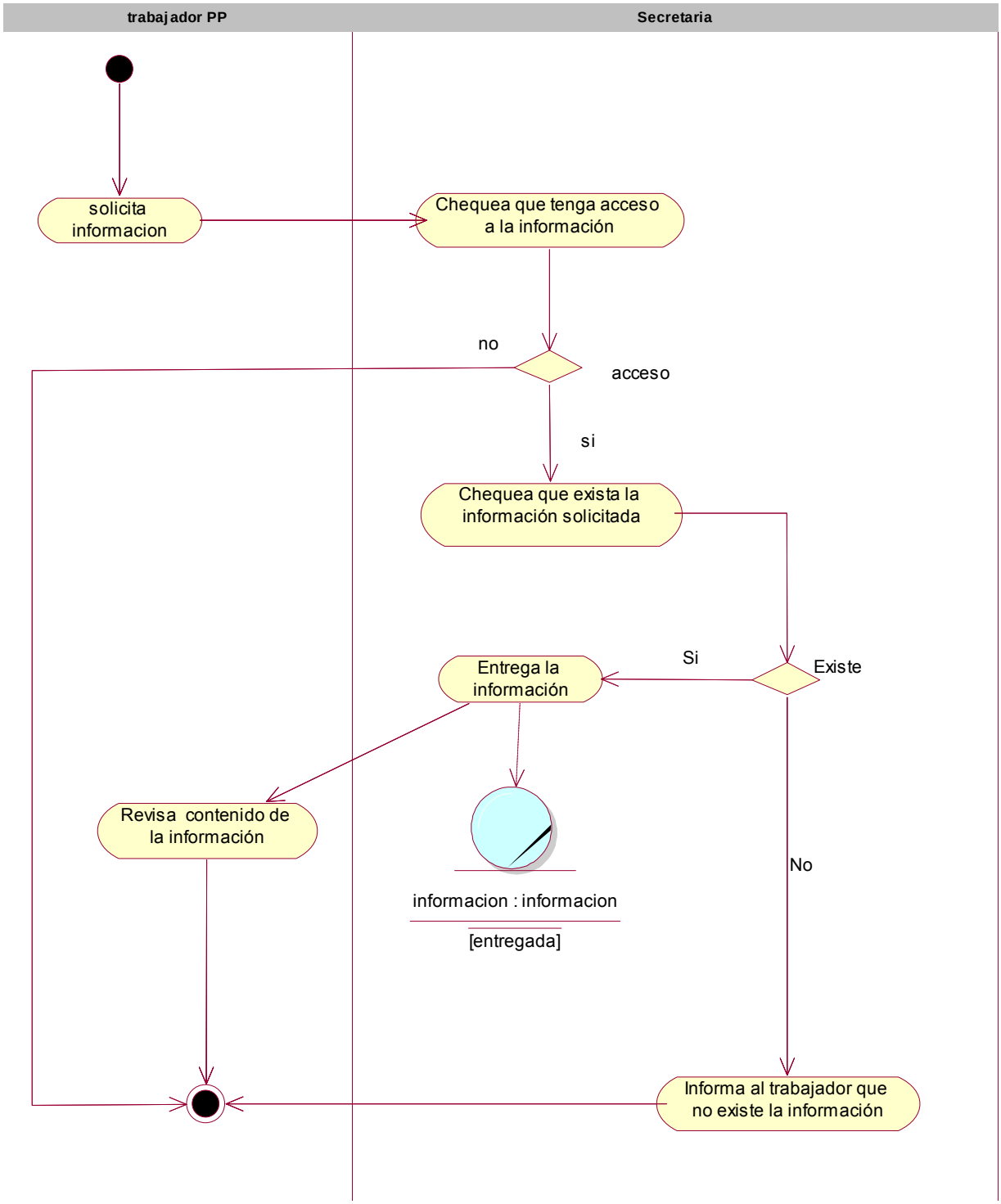
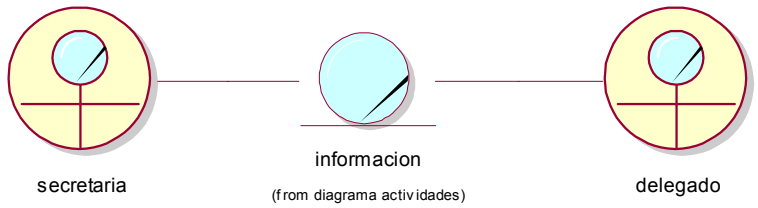

trabajador PP
Secretaria
solicita informacion
Chequea que tenga acceso a la información
no
acceso
si
Chequea que exista la información solicitada
Si
Existe
Entrega la información
Revisa contenido de la información
informacion : informacion
[entregada]
No
Informa al trabajador que no existe la información

Figura 7

Modelo de Objetos



Requisitos funcionales

1. Autentificar.
2. Gestionar usuario.
   2.1 Adicionar usuario.
   2.2 Modificar usuario.
   2.3 Listar usuario.
   2.4 Eliminar usuario.
3. Gestionar reunión.
   3.1 Adicionar reunión.
   3.2 Modificar reunión.
   3.3 Listar reunión
   3.4 Eliminar reunión
4. Gestionar planteamientos.
   4.1 Adicionar planteamientos.
   4.2 Modificar planteamientos.
   4.3 Listar planteamientos.
   4.4 Eliminar planteamientos.
5. Gestionar consejo popular.
   5.1 Adicionar consejo popular.
   5.2 Modificar consejo popular.

5.3 Listar consejo popular.

5.4Modificar consejo popular.

6. Gestionar clasificación.
   6.1 Adicionar clasificación.
   6.2 Modificar clasificación.
   6.3 Listar clasificación.
   6.4 Eliminar clasificación.
7. Gestionar proceso.
   7.1 Adicionar proceso.
   7.2 Modificar proceso.
   7.3 Listar proceso.
   7.4 Eliminar proceso.
8. Gestionar mandato.
   8.1 Adicionar mandato.
   8.2 Modificar mandato.
   8.3 Listar mandato.
   8.4 Eliminar mandato.
9. Gestionar solución.
   9.1 Adicionar solución.
   9.2 Modificar solución.
   9.3 Listar solución.
   9.4 Eliminar solución.
10. Gestionar nivel de solución.
    10.1 Adicionar nivel de solución.
    10.2 Modificar nivel de solución.
    10.3 Listar nivel de solución.
    10.4 Eliminar nivel de solución.

11. Gestionar entidad.
    11.1 Adicionar entidad.
    11.2 Modificar entidad.
    11.3 Listar entidad.
    11.4 Eliminar entidad.
12. Gestionar delegado.
    12.1 Adicionar delegado.
    12.2 Modificar delegado.
    12.3 Listar delegado.
    12.4 Eliminar delegado.
13. Visualizar datos generales de los planteamientos por mandato.
14. Visualizar datos generales de los planteamientos por entidad administrativa.
15. Visualizar datos específicos de los planteamientos según el campo.
16. Visualizar cantidad de planteamientos según el campo.

Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto usable, rápido o confiable. Los requisitos no funcionales aunque no aportan funcionalidades propiamente dichas dentro de la aplicación, son de vital importancia para una puesta en marcha exitosa del software, y para lograr que este responda a las expectativas del usuario.

Requerimientos de apariencia o interfaz externa:

El sistema esta concebido para que sea utilizado por personas con conocimientos mínimos de informática, no obstante, la interfaz a de ser amigable y de fácil entendimiento, para facilitar el trabajo de los usuarios en la misma.

Requerimientos de Usabilidad:

El sistema estará controlado por un administrador con conocimientos de informática, los cuales, con el apoyo de materiales de ayuda que les serán proporcionados y teniendo acceso pleno al sistema, será capaz de dar solución a cualquier situación que se pueda presentar.

Requerimientos de Rendimiento:

Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a sus páginas. La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas en la Base de Datos. La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el cliente requiere la respuesta a su petición.

Requerimientos de Portabilidad:

La aplicación propuesta debe ser multiplataforma (podrá ser usada bajo cualquier sistema operativo).

Requerimientos de Seguridad:

Confiabilidad: la información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación, dándole a casa usuario solo los privilegios indispensables para este.
Integridad: la información con la que se cuenta debe ser protegida, con el objetivo de mantenerla íntegra, es decir que esta no sea transformada.
Disponibilidad: La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

Requerimientos de Software:

En la computadora de los usuarios sólo se requiere un navegador Web. Como servidor Web se debe tener el Apache en su versión 2.0 en adelante y como Gestor de Base de Datos MySQL.

Requerimientos de Hardware:

En el cliente se requiere de una máquina con 128 MB de RAM como mínimo, el servidor Web deberá tener 256 MB de RAM y 20 GB de capacidad de almacenamiento en disco como mínimo, y el servidor de base de datos 256 MB de RAM como mínimo, todas las máquinas implicadas en la funcionalidad de la aplicación deben estar conectadas a la red.

Modelo caso de uso del sistema

Se representan los requisitos funcionales del sistema mediante un diagrama de casos de uso utilizando UML como lenguaje de modelado visual, donde se definen los actores y los casos de uso del mismo.

Actores del sistema

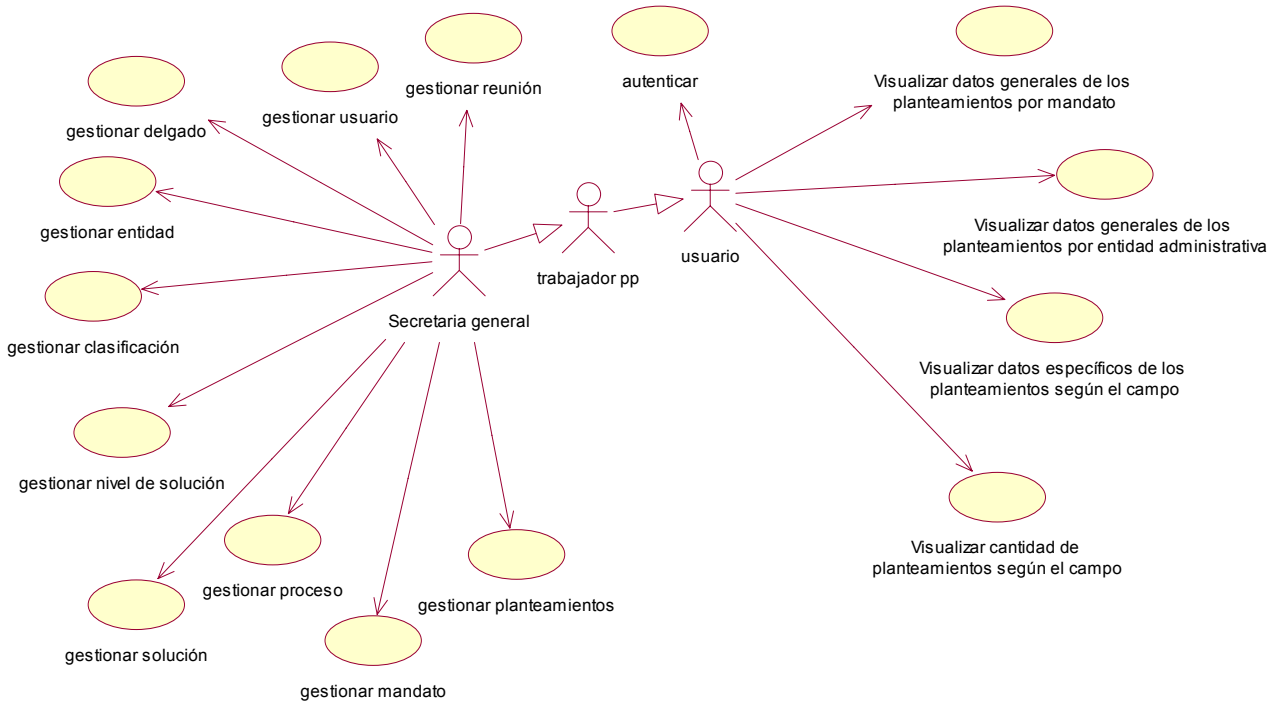
Tabla No4.

Actor del sistema	Justificación
Secretaria general	Es la encargada de gestionar la información(los planteamientos).
Usuario	Es un usuario general, ya sea la secretaria o el trabajador, ambos se van a autenticar y van a acceder a los reportes.

Trabajador PP	Es quien solicitará los reportes que debe brindarle el sistema.

Figura 8

Diagrama de casos de uso del sistema



Descripción de los casos de uso.

Tabla No5.

Caso de uso : Autentificar
Actores: Usuario
Propósito: Permitir a un usuario autenticarse y acceder al sistema.
Resumen: El caso de uso inicia cuando el usuario decide acceder al sistema, este introduce los datos que le solicitan para acceder al mismo y finaliza cuando el sistema verifica los datos y le habilita la entrada.
Referencias: R1
Precondiciones El trabajador del poder popular debe estar registrado.
Poscondiciones El trabajador del poder popular puede tener acceso a la información permitida.
Prototipo: ANEXO 1

Tabla No6.

Caso de uso: Gestionar usuario
Actores: Secretaria general
Propósito: Permite a la secretaria general de la asamblea realizar cualquier operación como agregar, modificar o eliminar usuarios del sistema.
Resumen: El caso de uso se inicia cuando la secretaria general desea modificar los datos

referida a los usuarios.Si lo que quiere es agregar un trabajador el sistema le presentará un formulario para que llene los datos. Si lo que desea es modificar entonces se muestra un formulario que permite buscar y modificar los datos del trabajador. Si quiere eliminar a un trabajador el sistema muestra un formulario para que marque el identificador del trabajador y lo elimine. El caso de uso termina con la actualización de los datos.
Referencias: R2.
Precondiciones Para eliminar o modificar un trabajador, este debe de estar registrado en la base de datos. Si se desea agregar un nuevo trabajador este debe pertenecer al centro de trabajo.
Postcondiciones Los datos de los usuarios del sistema son modificados.
Prototipo: ANEXO 2

Tabla No7.

Caso de uso : Gestionar reunión.
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referente a las reuniones, puede adicionar, listar, modificar o eliminar.
Resumen: El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a las reuniones en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el

sistema le mostrara el listado de reuniones con sus identificadores y los datos pertenecientes a estas. En caso que quiera modificar alguna reunión se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador de la reunión a eliminar.
Referencias: R3.
Precondiciones Si se quiere listar, modificar o eliminar alguna reunión esta debe de estar en la base de datos del sistema en caso de que se quiera adicionar una reunión, se podrá agregar solo si es una reunión de un consejo popular.
Poscondiciones Los datos referentes a las reuniones quedan modificados.
Prototipo: ANEXO 3

Tabla No8.

Caso de uso : Gestionar planteamientos
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referente a los planteamientos, puede adicionar, listar, modificar o eliminar.
Resumen:

El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a los planteamientos en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de planteamientos con sus identificadores y los datos pertenecientes a estos. En caso que quiera modificar algún planteamiento se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador del planteamiento a eliminar.
Referencias: R4.
Precondiciones Si se quiere listar , modificar o eliminar algún planteamiento este debe de estar en la base de datos del sistema en caso de que se quiera adicionar un planteamiento , se podrá agregar solo si este lo envia un delegado del consejo popular .
Poscondiciones Los datos referentes a los planteamientos quedan modificados.
Prototipo: ANEXO 4

Tabla No9.

Caso de uso: Gestionar consejo popular.
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referente a los consejos populares, puede adicionar, listar, modificar o eliminar.
Resumen:

El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a los consejos populares, en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de los consejos populares con sus identificadores y los datos pertenecientes estos. En caso que quiera modificar algún consejo popular se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador del consejo popular a eliminar.
Referencias: R5.
Precondiciones Si se quiere listar, modificar o eliminar algún consejo popular este debe de estar en la base de datos del sistema, en caso de que se quiera adicionar un consejo popular, se podrá agregar solo si es un consejo popular perteneciente al municipio Cienfuegos.
Poscondiciones Los datos referentes a los consejos populares quedan modificados.
Prototipo: ANEXO 5

Tabla No10.

Caso de uso: Gestionar clasificación.
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referente a las clasificaciones, puede adicionar, listar, modificar o eliminar.
Resumen:

El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a las clasificaciones, en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de las clasificaciones con sus identificadores y los datos pertenecientes a estas. En caso que quiera modificar alguna clasificación se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador de la clasificación a eliminar.
Referencias: R6.
Precondiciones Si se quiere listar, modificar o eliminar alguna clasificación esta debe de estar en la base de datos del sistema, en caso de que se quiera adicionar una clasificación, se podrá agregar solo si es una clasificación aprobada por los máximos dirigentes del poder popular municipal de la ciudad de Cienfuegos.
Poscondiciones Los datos referentes a las clasificaciones quedan modificados.
Prototipo: ANEXO 6

Tabla No11.

Caso de uso: Gestionar proceso.
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referente a los procesos, puede adicionar, listar, modificar o eliminar.
Resumen:

El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a los procesos, en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de los procesos con sus identificadores y los datos pertenecientes estos. En caso que quiera modificar algún proceso se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador del proceso a eliminar.
Referencias: R7.
Precondiciones Si se quiere listar, modificar o eliminar algún proceso este debe de estar en la base de datos del sistema, en caso de que se quiera adicionar un proceso, se podrá agregar solo si es un proceso aprobado por los máximos dirigentes del poder popular municipal de la ciudad de Cienfuegos.
Poscondiciones Los datos referentes a los procesos quedan modificados.
Prototipo: ANEXO 7

Tabla No12.

Caso de uso: Gestionar mandatos.
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referente a los mandatos, puede adicionar, listar, modificar o eliminar.
Resumen: El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o

eliminar información referida a los mandatos, en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de los mandatos con sus identificadores y los datos pertenecientes estos. En caso que quiera modificar algún mandato se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador del mandato a eliminar.
Referencias: R8.
Precondiciones Si se quiere listar, modificar o eliminar algún mandato este debe de estar en la base de datos del sistema, en caso de que se quiera adicionar un mandato, se podrá agregar solo si es un mandato aprobado por los máximos dirigentes del poder popular municipal de la ciudad de Cienfuegos.
Poscondiciones Los datos referidos a los mandatos quedan modificados.
Prototipo: ANEXO 8

Tabla No13.

Caso de uso: Gestionar solución.
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referida a las soluciones, puede adicionar, listar, modificar o eliminar.
Resumen: El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a las soluciones, en caso de que desee adicionar

el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de las soluciones con sus identificadores y los datos pertenecientes estos. En caso que quiera modificar alguna solución se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador de la solución a eliminar.
Referencias: R9.
Precondiciones Si se quiere listar, modificar o eliminar alguna solución esta debe estar en la base de datos del sistema, en caso de que se quiera adicionar una solución, se podrá agregar solo si la solución es aprobada por los máximos dirigentes del poder popular municipal de la ciudad de Cienfuegos.
Poscondiciones Los datos referidos a las soluciones quedan modificados.
Prototipo: ANEXO 9

Tabla No14.

Caso de uso:	Gestionar nivel de solución.
Actores: Secretaria general	
Propósito: Permite a la secretaria general controlar y hacer cualquier acción referida a los niveles de solución, puede adicionar, listar, modificar o eliminar.	
Resumen: El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o	

eliminar información referida a los niveles de solución, en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de los niveles de solución con sus identificadores y los datos pertenecientes a estos. En caso que quiera modificar algún nivel de solución se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador del nivel de solución a eliminar.
Referencias: R10.
Precondiciones Si se quiere listar, modificar o eliminar algún nivel de solución esta debe estar en la base de datos del sistema, en caso de que se quiera adicionar un nivel de solución, se podrá agregar solo si el nivel de solución es aprobado por los máximos dirigentes del poder popular municipal de la ciudad de Cienfuegos.
Poscondiciones Los datos referidos a los niveles de solución quedan modificados.
Prototipo: ANEXO 10

Tabla No15.

Caso de uso: Gestionar entidad.
Actores: Secretaria general
Propósito: Permite a la secretaria general controlar y hacer cualquier acción

referida a las entidades, puede adicionar, listar, modificar o eliminar.
Resumen: El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a las entidades, en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de las entidades con sus identificadores y los datos pertenecientes a estas. En caso que quiera modificar alguna entidad se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador de la entidad a eliminar.
Referencias: R11.
Precondiciones Si se quiere listar, modificar o eliminar alguna entidad esta debe estar en la base de datos del sistema, en caso de que se quiera adicionar una entidad, se podrá agregar solo si la entidad pertenece al municipio al municipio Cienfuegos.
Poscondiciones Los datos referidos a las entidades quedan modificados.
Prototipo: ANEXO 11

Tabla No16.

Caso de uso: Gestionar delegado**.**
Actores: Secretaria general

Propósito: Permite a la secretaria general controlar y hacer cualquier acción referida a los delegados, puede adicionar, listar, modificar o eliminar.
Resumen: El caso de uso inicia cuando la secretaria quiere adicionar, listar, modificar o eliminar información referida a los delegados, en caso de que desee adicionar el sistema le mostrará un formulario con los campos a llenar. Si decide listar el sistema le mostrara el listado de los delegados con sus identificadores y los datos pertenecientes a estos. En caso que quiera modificar algún delegado se le brinda un formulario con los campos a modificar y si lo que desea es eliminar debe marcar el identificador del delegado a eliminar.
Referencias: R12.
Precondiciones Si se quiere listar, modificar o eliminar algún delegado este debe estar en la base de datos del sistema, en caso de que se quiera adicionar un delegado, se podrá agregar solo si es un delegado de un consejo popular perteneciente al municipio Cienfuegos.
Poscondiciones Los datos referidos a los delegados quedan modificados.
Prototipo: ANEXO 12

Tabla No17.

Caso de uso: Visualizar los datos generales de los planteamientos por mandato.
Actores: usuario.
Propósito: Permite al trabajador del poder popular visualizar la información referida a los datos generales de los planteamientos por mandatos electorales.
Resumen: El caso de uso inicia cuando el trabajador del poder popular entra al sitio y necesita consultar los datos generales de los planteamientos hechos por cada mandato electoral.
Referencias: R13.
Precondiciones El trabajador debe escoger la opción mandato en el menú.
Poscondiciones El trabajador consulta la información requerida.
Prototipo: ANEXO 13

Tabla No18.

Caso de uso: Visualizar los datos generales de los planteamientos por entidad.
Actores: usuario.
Propósito: Permite al trabajador del poder popular visualizar la información referida a los datos generales de los planteamientos por entidades

administrativas.
Resumen: El caso de uso inicia cuando el trabajador del poder popular entra al sitio y necesita consultar los datos generales de los planteamientos pertenecientes a cada entidad administrativa.
Referencias: 14.
Precondiciones El trabajador debe escoger la opción entidad en el menú.
Poscondiciones El trabajador consulta la información requerida.
Prototipo: ANEXO 14

Tabla No19.

Caso de uso: Visualizar los datos específicos de los planteamientos por campos.
Actores: usuario.
Propósito: Permite al trabajador del poder popular visualizar la información referida a los datos específicos de los planteamientos por campos, los campos a escoger son: reunión, mandato, proceso, clasificación, solución, nivel de solución, entidad, también puede combinar los campos y así restringir la búsqueda de los datos que desea.
Resumen: El caso de uso inicia cuando el usuario (trabajador del poder popular,

secretaria) entra al sitio y necesita consultar los datos específicos de los planteamientos según el campo.
Referencias: R15.
Precondiciones El trabajador debe escoger el campo o los campos en el menú.
Poscondiciones El trabajador consulta la información requerida.
Prototipo: ANEXO 15

Tabla No20.

Caso de uso: Visualizar la cantidad de planteamientos por campos.
Actores: usuario.
Propósito: Permite al trabajador del poder popular visualizar la cantidad de planteamientos por campos, los campos a escoger son: reunión, mandato, proceso, clasificación, solución, nivel de solución, entidad, también puede combinar los campos y así restringir la búsqueda que desea.
Resumen: El caso de uso inicia cuando el usuario (trabajador del poder popular, secretaria) entra al sitio y necesita consultar la cantidad de planteamientos según el campo.
Referencias: R16.
Precondiciones

El trabajador debe escoger el campo o los campos en el menú.
Poscondiciones El trabajador consulta la información requerida.
Prototipo: ANEXO 16

Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución a partir de un listado con las propiedades que debe cumplir el sistema y las funcionalidades que debe realizar, representándolas mediante el modelado de los procesos del negocio y finalmente describiendo paso a paso los requisitos funcionales que intervienen en el sistema. Ahora se puede comenzar a construir la aplicación, tratando de que se cumplan todos los requisitos y las funciones que se han considerado necesarias para este capítulo.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

El presente capítulo se relacionara con la construcción de la solución propuesta el estudio de la factibilidad del producto de software y la validación del sistema, por lo que se analizarán los diagramas de clases de diseño, diagramas del modelo lógico de datos, diagramas del modelo físico de datos, el diagrama de implementación y el diagrama de componentes los cuales nos posibilitan una mayor comprensión para la construcción del sistema.

Diagrama de clases del Diseño

El diagrama de clases del diseño es un modelo que permite representar las relaciones existentes entre las diferentes páginas de la aplicación, donde cada página lógica puede ser representada con una clase, además visualiza cómo se quiere que funcione el sistema informático y la relación con los restantes artefactos de UML.
El Diagrama de clases Web fue definido a partir de lo Casos de Uso del Sistema y se muestra en la siguiente tabla:

Tabla No21.

Caso de Uso	**Escenario**	**Diagrama de clases Web**
Gestionar reunión	Adicionar reunión	Anexo 16
	Modificar reunión	

	Eliminar reunión	
Gestionar usuario	Adicionar usuario	Anexo 17
	Modificar usuario	
	Eliminar usuario	
Gestionar planteamientos	Adicionar planteamientos	Anexo 18
	Modificar planteamientos	
	Eliminar planteamientos	
Gestionar consejo popular	Adicionar consejo popular	Anexo 19
	Modificar consejo popular	
	Eliminar consejo popular	
Gestionar clasificación	Adicionar clasificación	Anexo 20
	Modificar clasificación	
	Eliminar clasificación	
Gestionar proceso	Adicionar	Anexo 21
	Modificar	
	Eliminar	
Gestionar mandato	Adicionar	Anexo 22
	Modificar	

	Eliminar	
Gestionar solución	Adicionar	Anexo 23
	Modificar	
	Eliminar	
Gestionar nivel de solución	Adicionar	Anexo 24
	Modificar	
	Eliminar	
Gestionar delegados	Adicionar	Anexo 25
	Modificar	
	Eliminar	
Autenticarse	Autenticarse	Anexo 26
Gestionar entidad	Adicionar	Anexo 27
	Modificar	
	Eliminar	
Visualizar datos generales de los planteamientos por mandato	Visualizar datos generales de los planteamientos por mandato	Anexo 28
Visualizar datos generales de los planteamientos por entidad administrativa	Visualizar datos generales de los planteamientos por entidad administrativa	Anexo 29

Visualizar datos específicos de los planteamientos según el campo	Visualizar datos específicos de los planteamientos según el campo	Anexo 30
Visualizar cantidad de planteamientos según el campo	Visualizar cantidad de planteamientos según el campo	Anexo 31

Diagramas del modelo lógico de datos

El modelo lógico de datos representa un diagrama de clases persistentes muestra la capacidad de un objeto de mantener su valor en el espacio y en el tiempo y son utilizadas para interactuar, desde la capa del negocio, con la capa de datos. **ANEXO 32**

Diagramas del modelo físico de datos

El modelo físico de datos se obtiene del modelo lógico de datos y es una representación gráfica de las tablas físicas almacenadas en la base de datos con sus relaciones. Quedan señalados los campos llaves en cada tabla así como los que son llaves extranjeras que provienen de tablas relacionadas. **ANEXO 33**

Diagramas de implementación

El modelo de implementación muestra la implementación del sistema en términos de componentes y subsistemas de implementación. Describe como se organizan los componentes de acuerdo a los mecanismos de estructuración. Los diagramas de implementación muestran los aspectos físicos del sistema. Incluyen la estructura del código fuente y la implementación, en tiempo de implementación. **ANEXO 34**

Tratamiento de errores

A menudo los usuarios cometen errores que pueden afectar el funcionamiento de los sistemas por lo que es necesario siempre realizar un tratamiento responsable de los posibles errores que puedan cometer los usuarios como entrada de datos incorrectos o en un formato no soportado por el sistema. El sistema propuesto al detectar errores informa al usuario mediante un mensaje de color rojo donde, de forma explicativa y sencilla, se dan detalles generales del error cometido. Los formularios del sistema propuesto contienen una validación de sus elementos para evitar los errores más frecuentes cometidos por los usuarios. Algunos de estos errores son: campos requeridos sin valor, números donde sólo pueden haber letras y viceversa, longitud de campo incorrecta, etc.…

Concepción del sistema de seguridad y protección.

El sistema presenta un adecuado sistema de seguridad y protección ya que para lograr acceder a la información que puede ser cambiada en la base de datos debe seguir la política de usuario que

tiene el sistema, pues si no se esta registrado como administrador no se pude realizar gestión de información con el objetivo de que la incurrencia de errores sea mínima.

Análisis de la factibilidad
Planificación

En el desarrollo de este capítulo utilizamos el método de Puntos de Función para la estimación del esfuerzo, el tiempo de desarrollo y el costo del proyecto. Para realizar el cálculo de los costos de desarrollo del sistema se deben obtener primero las instrucciones fuentes. Analizándose para esto las cantidades de entradas, salidas, peticiones, archivos lógicos e interfaces externas preliminares que tiene el sistema.

Tabla No22. Entradas Externas

Nombre de la entrada externa	**Cantidad de ficheros**	**Cantidad de elementos de datos**	**Clasificación (alto, medio, bajo)**
Agregar usuario	1	4	B
Modificar usuario	1	3	B
Eliminar usuario	1	1	B
Agregar reunión	1	7	B
Eliminar reunión	1	1	B
Modificar reunión	1	7	B
Agregar planteamientos	1	11	B
Eliminar planteamientos	1	1	B
Modificar planteamientos	1	11	B
Agregar consejo popular.	1	3	B
Modificar consejo popular.	1	3	B
Eliminar consejo popular.	1	1	B
Agregar clasificación	1	2	B
Modificar clasificación	1	2	B
Eliminar clasificación	1	1	B
Agregar proceso	1	2	B
Modificar proceso	1	2	B
Eliminar proceso	1	1	B
Agregar mandato	1	2	B
Modificar mandato	1	2	B
Eliminar mandato	1	1	B
Agregar solución	1	2	B
Eliminar solución	1	1	B
Modificar solución	1	2	B
Agregar nivel de solución	1	2	B

Modificar nivel de solución	1	2	B
Eliminar nivel de solución	1	1	B
Agregar entidad	1	6	B
Modificar entidad	1	6	B
Eliminar entidad	1	1	B
Agregar delegado	1	7	B
Modificar delegado	1	7	B
Eliminar delegado	1	1	B

Tabla No23. Salidas externas

Nombre de la Salida externa	**Cantidad de ficheros**	**Cantidad de elementos de datos**	**Clasificación (alto, medio, bajo)**
Visualizar datos generales de planteamientos por mandato.	2	13	B
Visualizar datos generales de los planteamientos por entidad administrativa	2	19	M
Visualizar datos específicos de planteamientos según el campo.	1	7	B
Visualizar cantidad de planteamientos el campo.	2	3	A

Tabla No24. Peticiones

Nombre de la Salida externa	**Cantidad de ficheros**	**Cantidad de elementos de datos**	**Clasificación (alto, medio, bajo)**
Buscar reunión	1	7	B
Buscar planteamiento	1	6	B
Buscar consejo popular	1	5	B
Buscar clasificación	1	4	B
Buscar proceso	1	4	A
Buscar mandato	2	4	M
Buscar solución	2	4	B
Buscar nivel de solución	2	4	A
Buscar entidad	1	8	B
Buscar delegado	1	7	B

Tabla No25. Ficheros lógicos

Ficheros lógicos

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
reunión	1	7	B

planteamiento	1	6	B
consejo popular	1	5	B
clasificación	1	4	B
proceso	1	4	B
mandato	1	4	B
solución	1	4	B
nivel de solución	1	4	B
entidad	1	8	B
delegado	1	7	B

Tabla No26.

Puntos de función

Elementos	Bajos	X Peso	Medios	X Peso	Altos	X Peso	Subtotal puntos función
Ficheros lógicos internos	12	7	0	10	0	15	84
Entradas externas	27	3	1	4	2	6	97
Salidas externas	7	4	4	5	4	7	68

Peticiones	12	3	4	4	4	16	116
Total							365

Tabla No27.

Características		**Valor**	
Puntos de función desajustados		365	
Lenguaje	MYSQL	PHP	Java Script
Instrucciones fuentes por puntos de función	37	44	58
Por ciento de la aplicación en cuanto a requerimientos funcionales	35%	50%	15%
Instrucciones fuentes	2025.75	8030	7409.5
Total de Instrucciones fuentes			**17465.25**

Determinación de los costos

Tabla No28

Cálculo de:	Valor	Justificación
RCPX	1,00	BD moderada, no se requiere amplia documentación. La aplicación Web tiene una moderada complejidad. (Nominal)
RUSE	1,00	Se implementa código reutilizable para el aprovechamiento de este en toda la aplicación.(Nominal)
PDIF	1,00	No tiene grandes restricciones en cuanto al tiempo de ejecución ya que el software podrá estar trabajando varias horas. EL Software no tiene limitación de memoria impuesta. La plataforma de aplicación tiene gran estabilidad. (Nominal)
PERS	0,70	Hay poco movimiento del personal. (Alto)
PREX	0,84	El equipo tiene buen dominio y posee conocimiento del lenguaje de programación. Con una experiencia de aproximadamente un año. (Alto)
FCIL	0,87	Se utilizan herramientas de programación como: Macromedia Dreamweaver 2004, Zend Develoment Studio, así como la herramienta CASE Rational Rose

		para la documentación, empleando como notación UML. (Alto)
SCED	1,00	La planificación se hace con moderada frecuencia. (Nominal)
PREC	3,72	El equipo de desarrollo posee una comprensión considerable de los objetivos del producto, no tiene experiencia en la realización de software de este tipo. (Nominal)
FLEX	3,04	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa. (Nominal)
TEAM	1,10	El equipo que va a desarrollar el software es
		altamente cooperativo.
RESL	4,24	Teniendo en cuenta la alta experiencia que existe en
		el país acerca de este tipo de estudios existen
		algunos factores de riesgo. (Nominal)
PMAT	6,24	Nivel I Alto porque se encuentra en su primera etapa
		un poco avanzada. (Bajo)

Multiplicador de esfuerzos

EM = Π Emi = RCPX*RUSE*PDIF*PERS*PREX*FCIL*SCED
i=1

EM = Π Emi = 1,00 * 1,00 * 1,00 * 0,70 * 0.84 * 0,87 * 1,00 = 0,51156 ≈ **0,51**

i=1

Factores de escala

SF = Σ SFi = PREC + FLEX + RESL + TEAM +PMAT

SF = Σ SFi = 3, 72 + 3, 04 + 4, 24 + 1, 10 + 6, 24 = **18, 34**

Valores de los coeficientes

A = 2,94; B = 0,91; C = 3,67; D = 0,24

E = B + 0,01 * SF F = D + 0,2 * (E - B)

E = 0,91 + 0,01 * 18,34 F = 0,24 + 0,2 *(1.0934 – 0.91)

E = 1,0934 F = 0,27668

Esfuerzo

PM = A * (MF) E * EM

PM = 2,94 * (17,465) 1,0934 * 0,51

PM = 34.21 (personas meses)

Cálculo del tiempo de desarrollo

TDEV = C * *PMF*

TDEV = 3,67 * (34.21) 0,27668

TDEV = 9.75

Cálculo de la cantidad de hombres

CH = PM / TDEV

CH = 34.21 / 9.75

CH = 3.5

Costo

Se asume como salario promedio mensual 275$

CHM = 2 * Salario Promedio

CHM = 2 * 275

CHM = 550 $/mes

Costo = CHM * PM

Costo = $550 * 17

Costo = $9350

Los costos en los que se incurriría de desarrollarse el sistema serían:

Tabla No29.

Cálculo de:	**Valor**
Esfuerzo(PM)	34.21≈34
Tiempo de desarrollo	17 meses
Cantidad de hombres	2
Costo	$9350
Salario medio	$275,0
RCPX	1,00
RUSE	1,00
PDIF	1,00
PREX	0,84
FCIL	0,87
SCED	1,00

Conclusiones del capítulo

La herramienta propuesta trae consigo una serie de beneficios para la organización, porque va a contribuir a mejorar su funcionamiento, lo que indica que es factible implementar la herramienta propuesta.

CONCLUSIONES

Como resultados de la investigación se puede concluir lo siguiente:

• La gestión de la información en el Poder Popular del municipio Cienfuegos resulta compleja.

• La utilización de sistemas informáticos capaces de gestionar la información generada en las distintas etapas tiene influencia positiva.

• La base de datos desarrollada resulta apropiada para organizar y almacenar eficientemente datos relacionados con el proceso de gestión de la información en el PP, además permite establecer las políticas de seguridad necesarias y los niveles de acceso para cada tipo de usuario.

• El sistema informático obtenido como resultado de esta investigación, agiliza la confección y entrega de información.

RECOMENDACIONES

Una vez concluido el desarrollo del sistema para la gestión y control de la información en el Poder Popular, se recomienda:

• Implantar este sistema informático en dicho centro a partir de Enero del 2009.

• Identificar nuevas funcionalidades para el sistema a partir de su puesta en marcha, tomando en cuenta los criterios de los diferentes usuarios.

• Validar la efectividad del sistema, una vez implantado, mediante pruebas comparativas que midan la eficiencia del proceso de gestión de la información en el organismo en cuestión, a partir de la utilización de la aplicación.

• Desarrollar una versión mejorada del sistema a partir de los resultados obtenidos de las recomendaciones anteriores.

• Es de gran utilidad para los usuarios, la incorporación al sistema de una ayuda.

• Generalizar el contenido de esta investigación, utilizando el sistema para gestionar la información en otros organismos de este tipo.

REFERENCIAS BIBLIOGRÁFICAS

[1]Peralta, Manuel. Sistema de Información. [En línea] 1997. http://www.monografias.com/trabajos7/sisinf/sisinf.shtml#esi.

[2]Anónimo. Milenium. [En línea] 2008. http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm.

[3]Anónimo. ¿Qué es un lenguaje de programación? [En línea] 2009. http://www.alegsaonline.com/art/11.php.

[4]Hinostroza, Raúl Rodas. LinuxCentro.Net. [En línea] 2007. http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP.

[5]Anónimo. ASP Fácil. [En línea] 2009. http://www.aspfacil.com/articulos/278001.asp.

[6]Arambillete, Wilman. Introducción a las Java Server Pages. [En línea] 2002. http://www.webexperto.com/articulos/art/57/introduccion-a-las-java-server-pages-/.

[7]Valdés, Damián Pérez. ¿Qué es Javascript? [En línea] 2007. http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/.

[8]Anónimo. Zend Studio. [En línea] 2003. http://www.desarrolloweb.com/articulos/1178.php.

[9]Anónimo. Softonic. [En línea] 2008. http://quanta-plus.softonic.com/linux.

[10]Anónimo. Eclipse. [En línea] 2008.
http://www.eclipse.org/mylyn/

[11]Anónimo. Eclipse. [En línea] 2008.
http://www.eclipse.org/pdt/.

[12]Anónimo. Comparación entre SGBD. [En línea] 2009.
http://www.ilustrados.com/documentos/sgbd.pdf.

[13]Daniel, Pecos. PostGreSQL vs. MySQL. [En línea] 2009.
http://www.netpecos.org/docs/mysql_postgres/index.html.

[14]Anónimo. Wikipedia. [En línea] 2008.
http://es.wikipedia.org/wiki/Framework.

[15]Zaninotto, François. Symfony, la guía definitiva. [En línea] 2009.
http://www.librosweb.es/symfony/index.html.

[16] (Cataldi, 2000)
http://www.fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf.

[17]Patricio Letelier, Mª Carmen Penadés. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] http://www.willydev.net/descargas/masyxp.pdf.

BIBLIOGRAFÍA

Anónimo. El blog de INEW. [En línea] 2009.
http://elblogdeinwe.com/blogviejo/raiz/index.php?op=ver_noticia&noticia=56.

Anónimo. Gamarod. [En línea] 2004.
http://www.gamarod.com.ar/articulos/introduccion_a_aspnet.asp.

Anónimo. Milenium. [En línea] 2008.
http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm.

Anónimo. Wikipedia. [En línea] 2008.
http://es.wikipedia.org/wiki/Eclipse_(software).

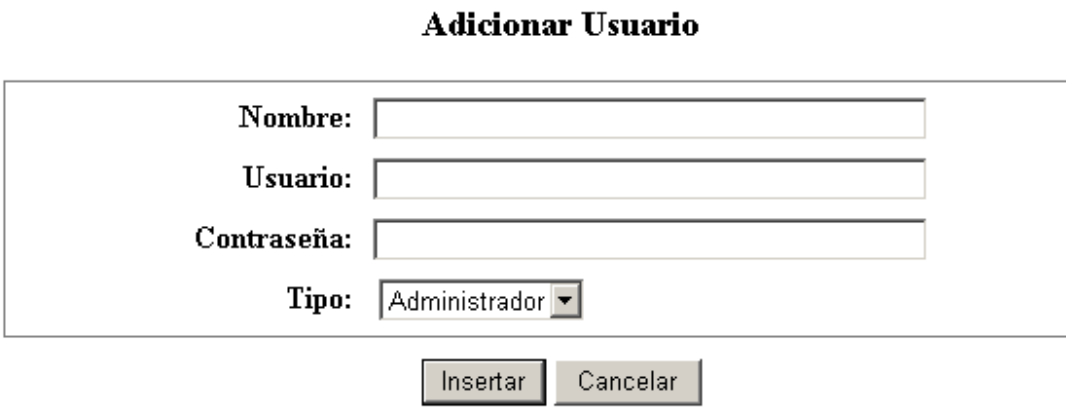
(González, D. (2008). Base de Datos PostgreSQL, SQL avanzado y PHP.
http://www.usabilidadweb.com.ar/postgre.php)

GLOSARIO

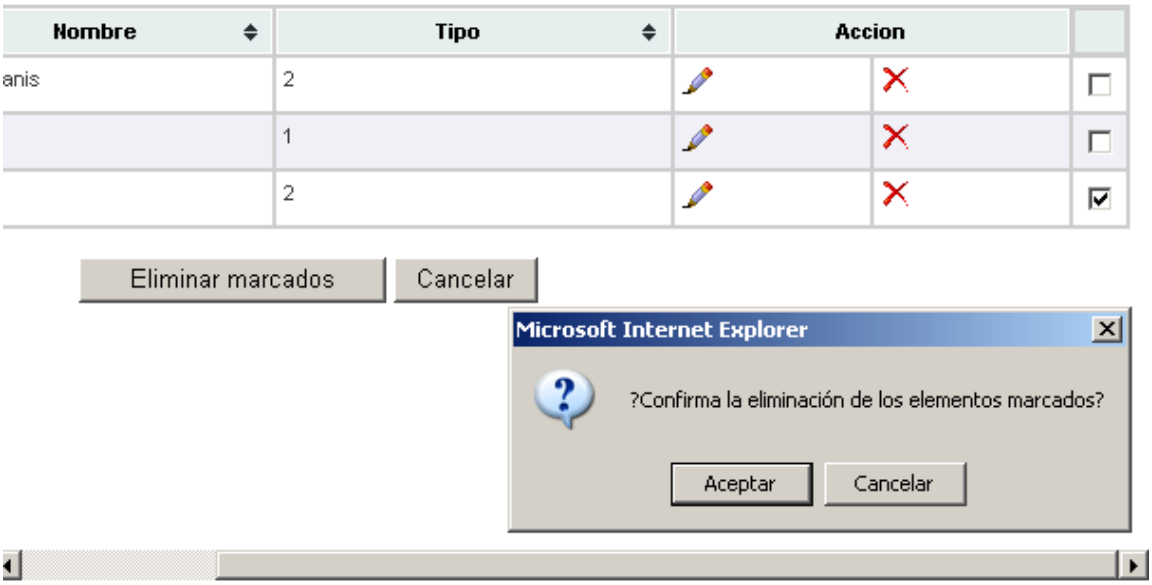
ANEXOS

ANEXO 1

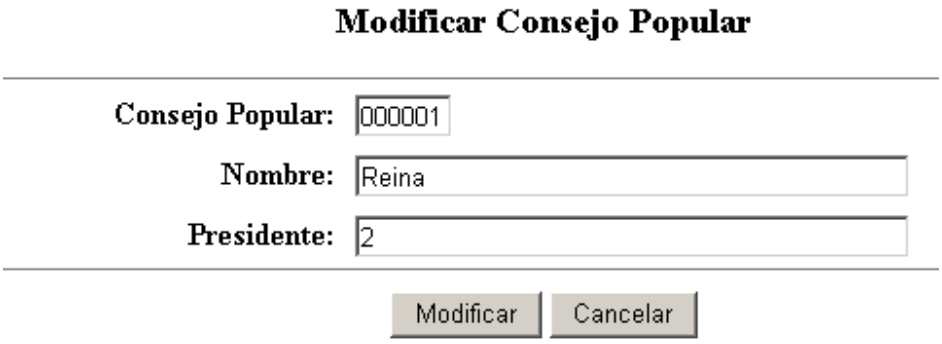
ANEXO 2



ANEXO 3

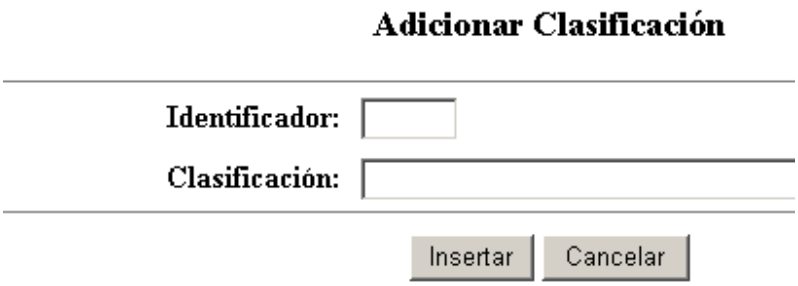

Nombre
Tipo
Accion
anis
2
1
2
Eliminar marcados
Cancelar
Microsoft Internet Explorer
?Confirma la eliminación de los elementos marcados?
Aceptar
Cancelar

ANEXO 4


Modificar Consejo Popular
Consejo Popular:
000001
Nombre:
Reina
Presidente:
2
Modificar
Cancelar

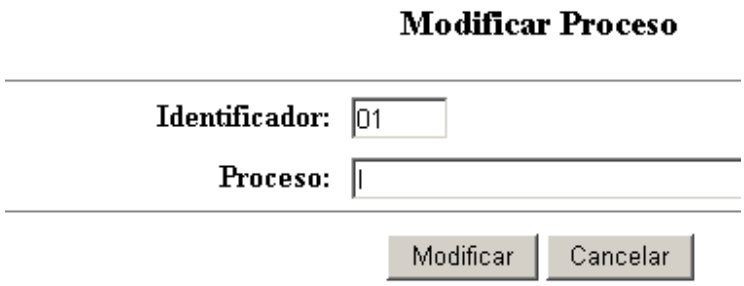
Adicionar Consejo Popular
Consejo Popular:
Nombre:
Presidente:
Insertar
Cancelar

ANEXO 5


Adicionar Clasificación
Identificador:
Clasificación:
Insertar
Cancelar

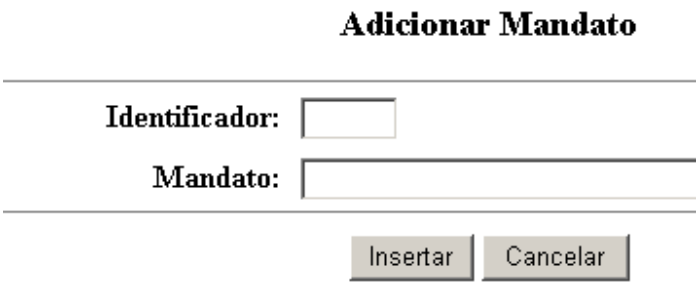
Listado de Clasificaciones
1/1
10
Identificador Clasificación Acción
1 Con Recursos
2 Con las Masas
3 Con Medidas
Eliminar marcados
Cancelar
Modificar Clasificación
Identificador: 1
Clasificación: Con Recursos
Modificar
Cancelar

ANEXO 6

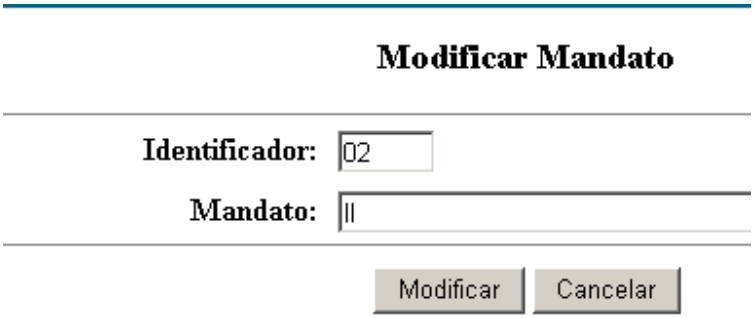

Modificar Proceso
Identificador: 01
Proceso: I
Modificar
Cancelar

Adicionar proceso
Identificador:
Proceso:
Insertar
Cancelar
Listado de Procesos
1/1
10
Identificador Proceso Acción
1 01 I
2 02 II
3 03 III
4 04 IV
Eliminar marcados
Cancelar

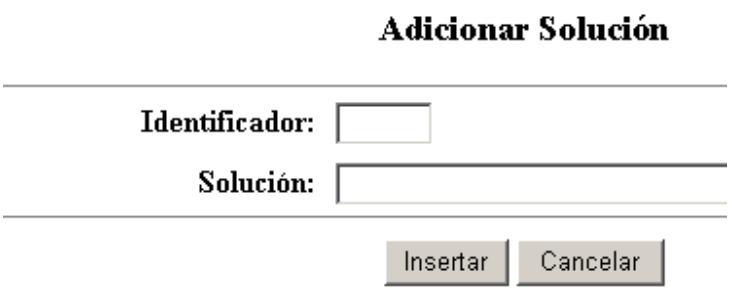
ANEXO 7


Adicionar Mandato
Identificador:
Mandato:
Insertar
Cancelar

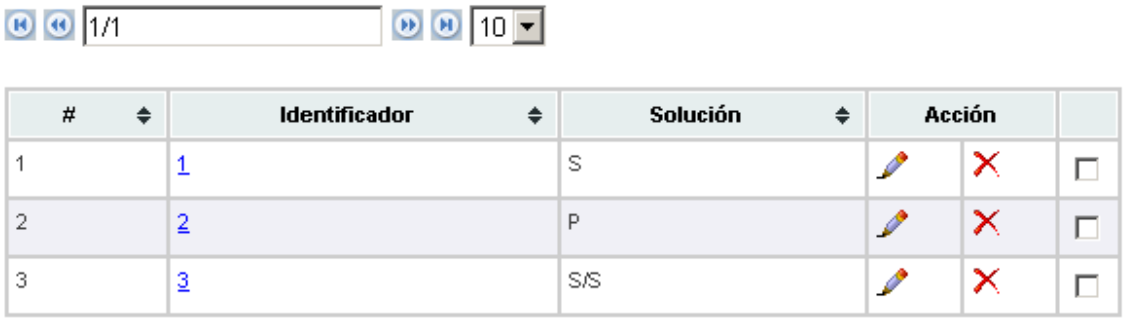
Listado de Mandatos
1/2
10
Identificador Mandato Acción
01 I
02 II
03 III


Modificar Mandato
Identificador: 02
Mandato: II
Modificar
Cancelar

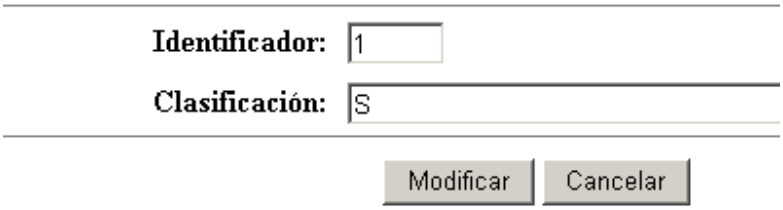
ANEXO 8


Adicionar Solución
Identificador:
Solución:
Insertar
Cancelar

Listado de Soluciones

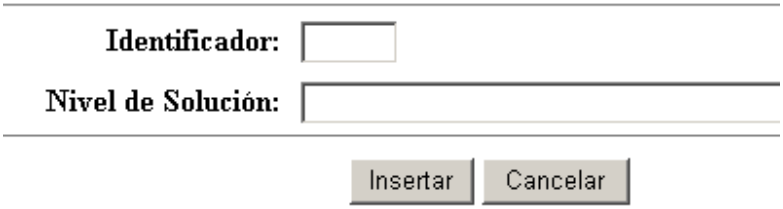


Modificar Solución



ANEXO 9

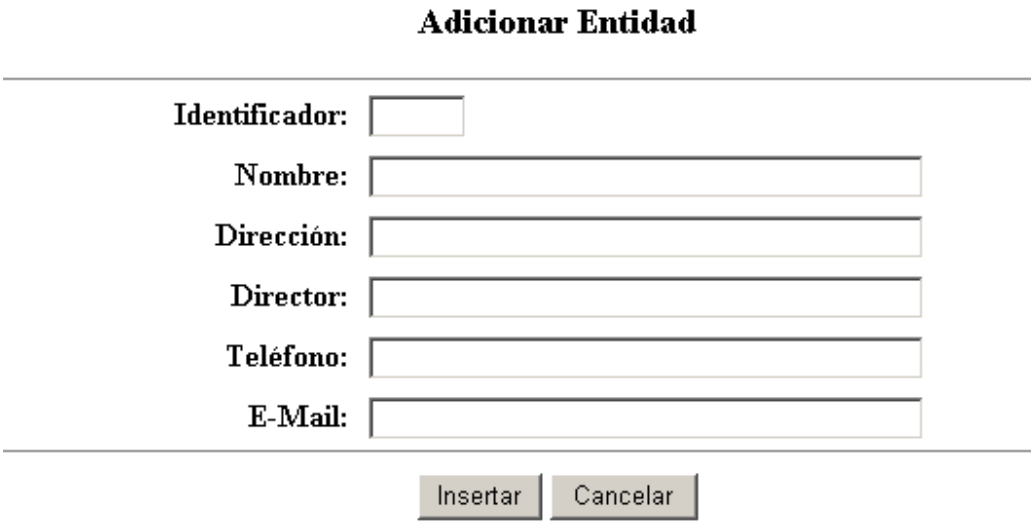
Adicionar Nivel de Solución



Listado de Niveles de Solución

Modificar Nivel de Solución

ANEXO 10



Anexo 11

Anexo 12

Resultados

Mandato y Proceso	Con Recursos			Con las Masas			
	Planteamientos	Solucionados	%	Planteamientos	Solucionados	%	Planteamie
I - I	0	0	0%	0	0	0%	0
I - II	0	0	0%	0	0	0%	0
I - III	0	0	0%	0	0	0%	0
I - IV	0	0	0%	0	0	0%	0

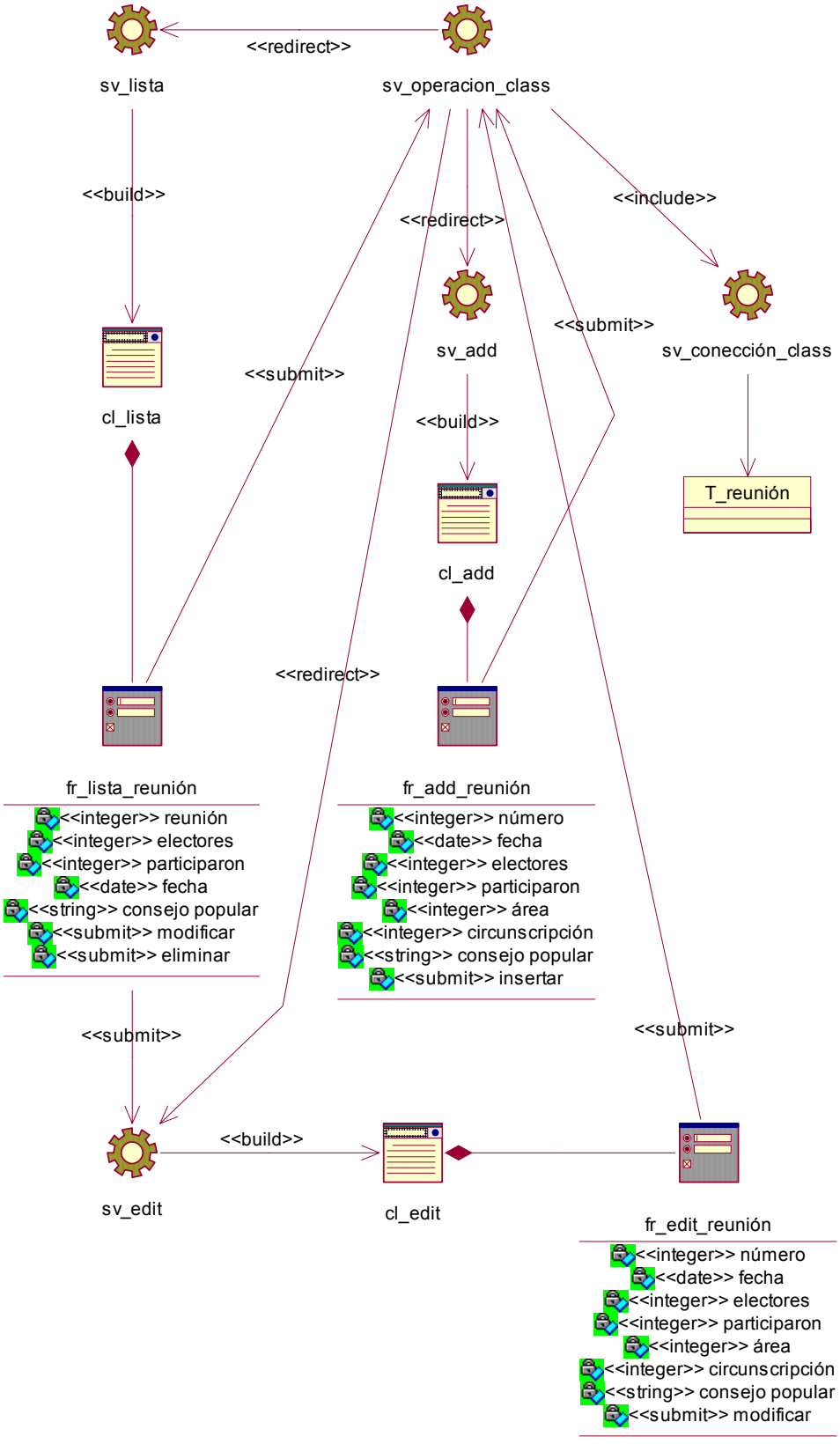
Anexo 13

Entidad	Con Recursos					C	
	Planteamientos	S	P	S/S	%	Planteamientos	S
CMP VC	0	0	0	0	0%	0	0
Acueducto y Alcantarillado	1	0	1	0	0%	0	0
OBE	0	0	0	0	0%	0	0

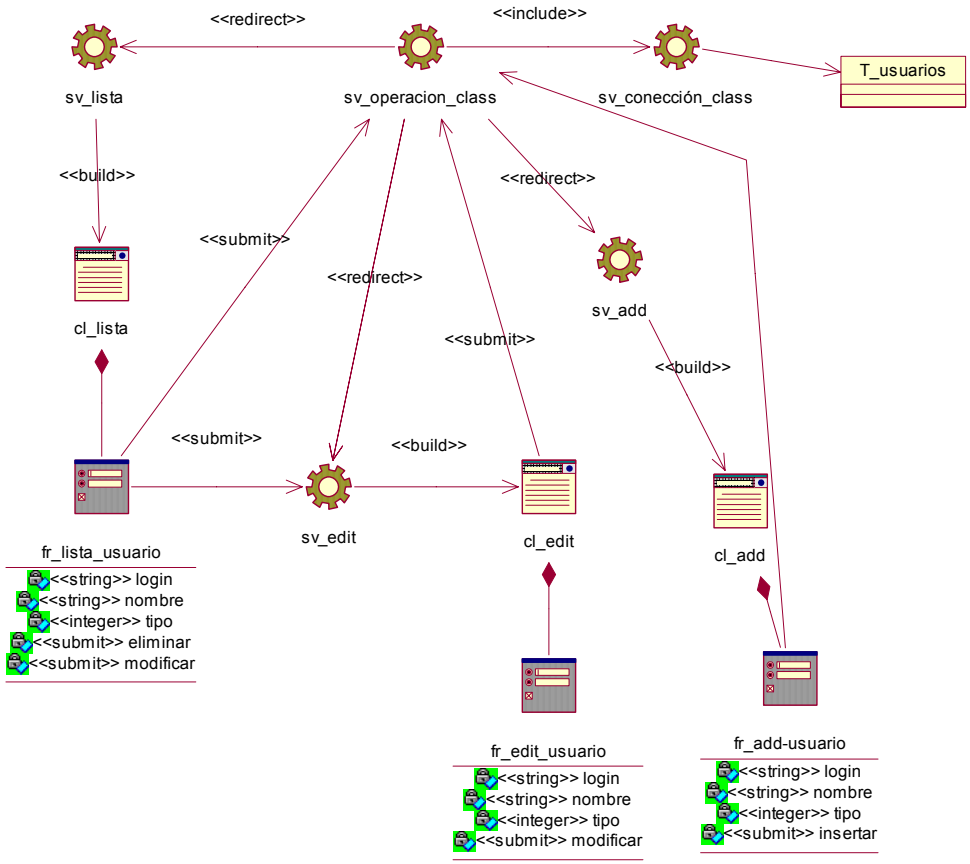
Anexo 14

Anexo 15

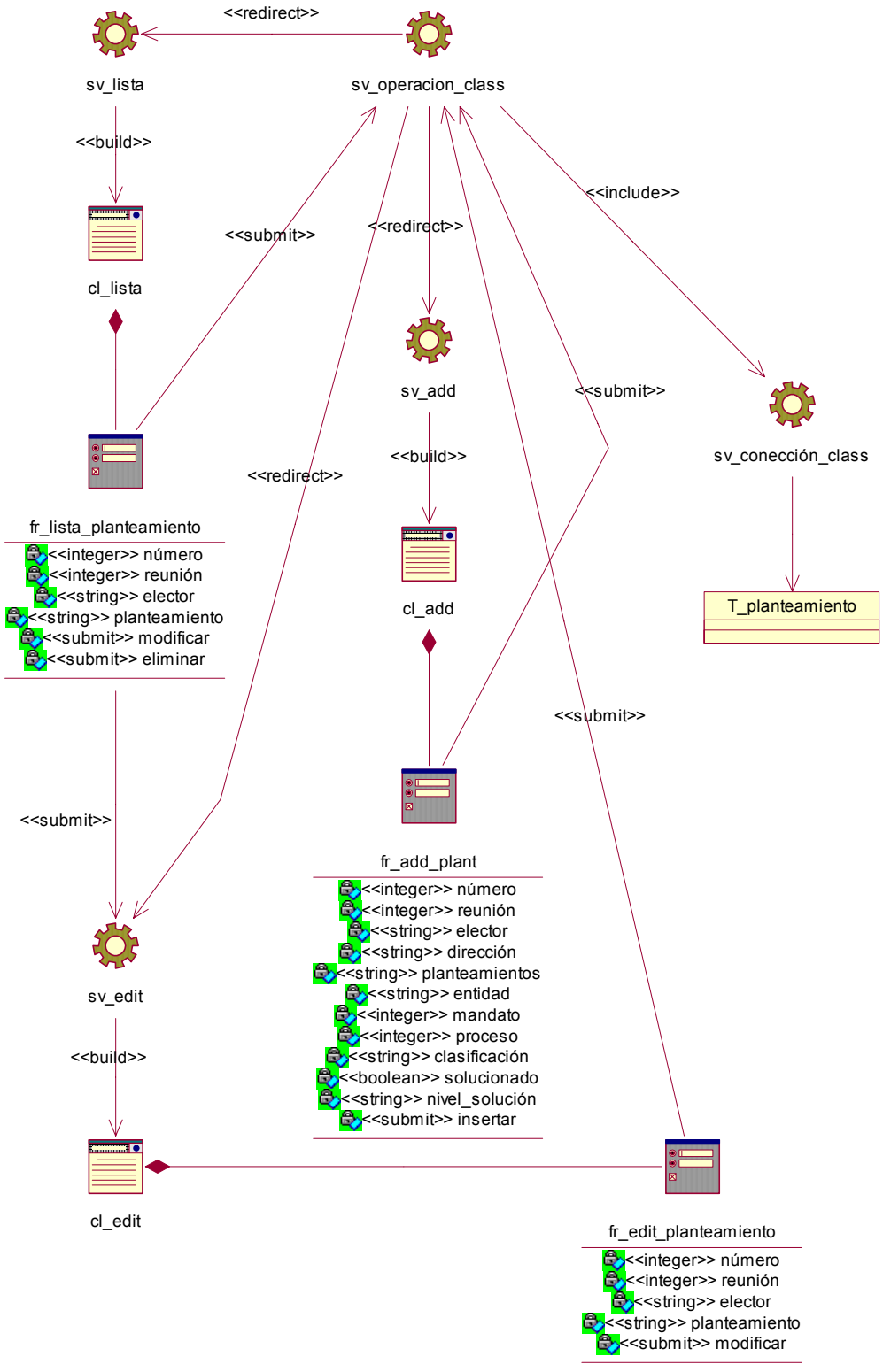
Anexo 16


<<redirect>>
sv_lista
sv_operacion_class
<<build>>
<<redirect>>
<<include>>
sv_add
<<submit>>
sv_conección_class
cl_lista
<<submit>>
<<build>>
T_reunión
cl_add
<<redirect>>
fr_lista_reunión
<<integer>> reunión
<<integer>> electores
<<integer>> participaron
<<date>> fecha
<<string>> consejo popular
<<submit>> modificar
<<submit>> eliminar
fr_add_reunión
<<integer>> número
<<date>> fecha
<<integer>> electores
<<integer>> participaron
<<integer>> área
<<integer>> circunscripción
<<string>> consejo popular
<<submit>> insertar
<<submit>>
<<submit>>
<<build>>
sv_edit
cl_edit
fr_edit_reunión
<<integer>> número
<<date>> fecha
<<integer>> electores
<<integer>> participaron
<<integer>> área
<<integer>> circunscripción
<<string>> consejo popular
<<submit>> modificar

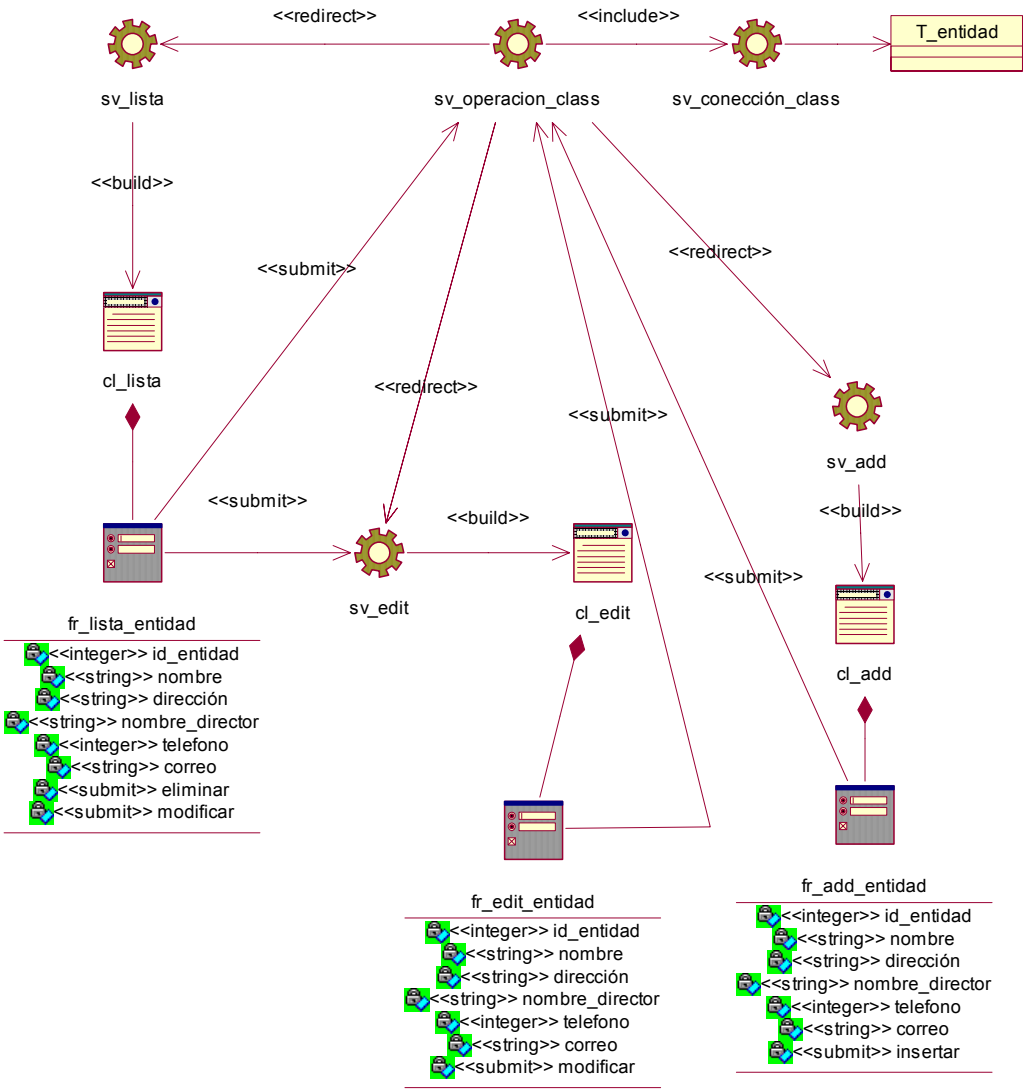
Anexo 17



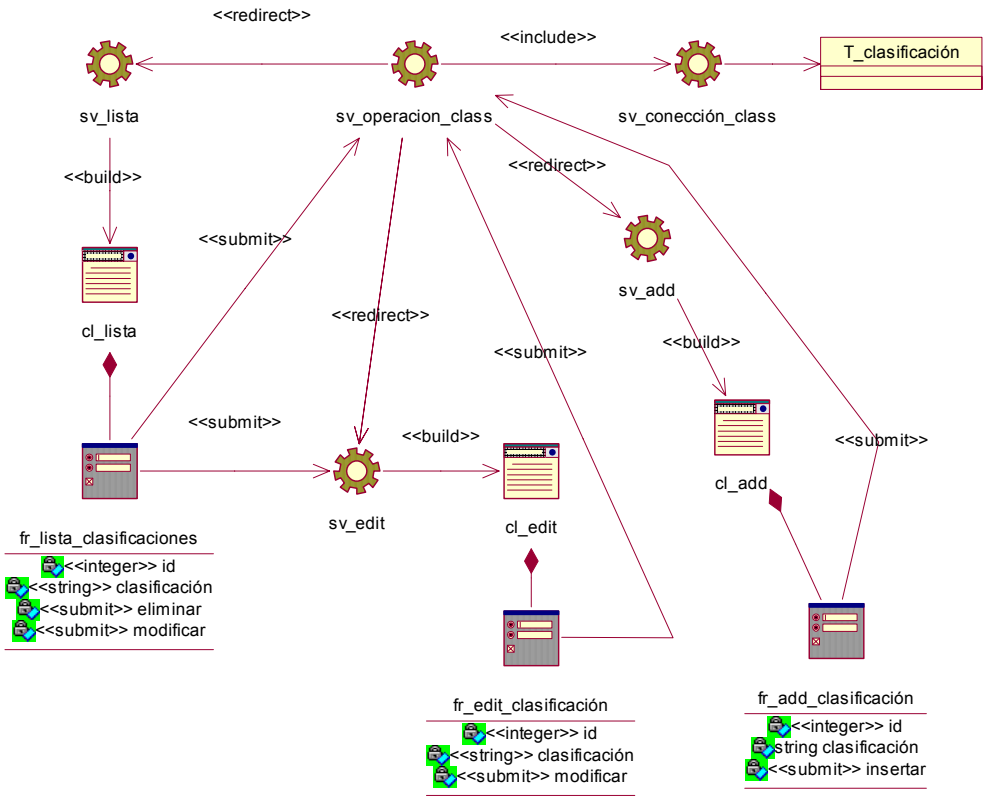
Anexo 18


<<redirect>>
sv_lista
sv_operacion_class
<<build>>
<<include>>
<<submit>>
<<redirect>>
cl_lista
sv_add
<<submit>>
sv_conección_class
<<redirect>>
<<build>>
fr_lista_planteamiento
<<integer>> número
<<integer>> reunión
<<string>> elector
<<string>> planteamiento
<<submit>> modificar
<<submit>> eliminar
cl_add
T_planteamiento
<<submit>>
<<submit>>
fr_add_plant
<<integer>> número
<<integer>> reunión
<<string>> elector
<<string>> dirección
<<string>> planteamientos
<<string>> entidad
<<integer>> mandato
<<integer>> proceso
<<string>> clasificación
<<boolean>> solucionado
<<string>> nivel_solución
<<submit>> insertar
sv_edit
<<build>>
cl_edit
fr_edit_planteamiento
<<integer>> número
<<integer>> reunión
<<string>> elector
<<string>> planteamiento
<<submit>> modificar

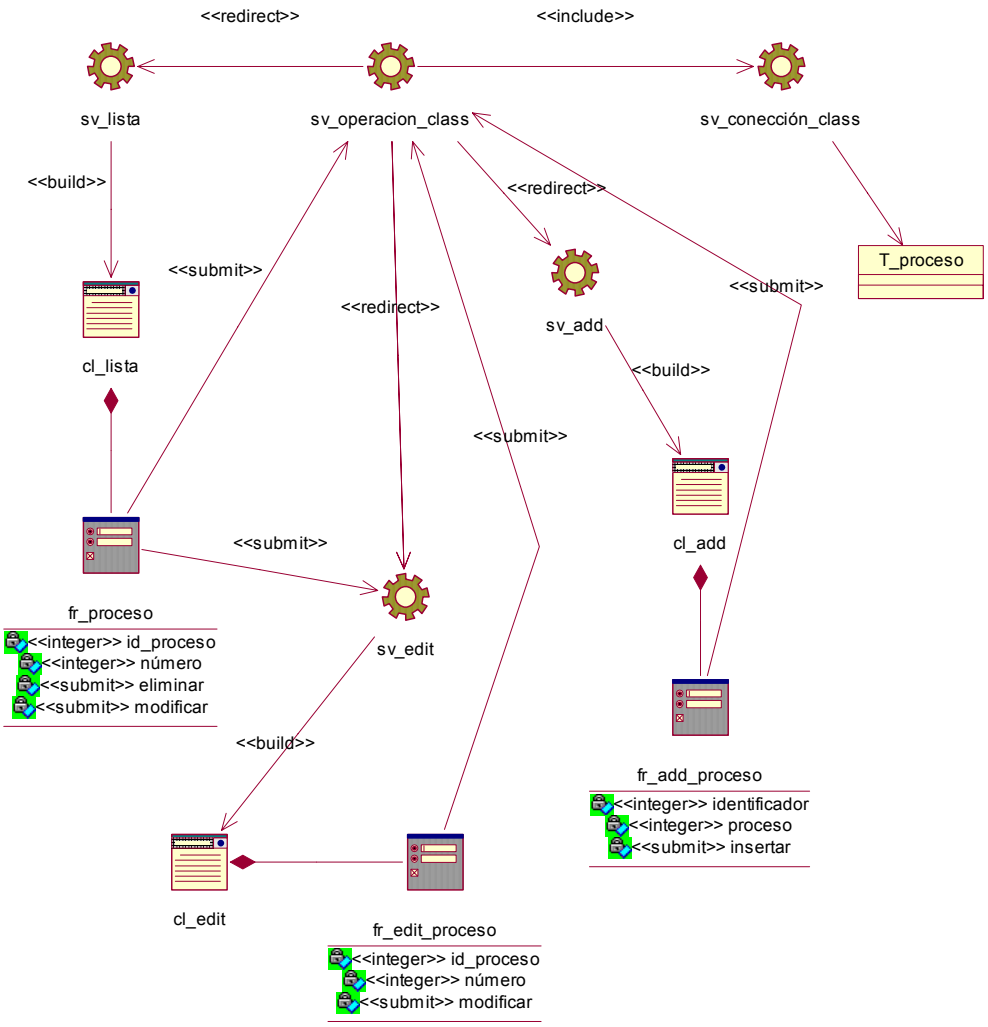
Anexo 19


<<redirect>>
<<include>>
T_entidad
sv_lista
sv_operacion_class
sv_conección_class
<<build>>
<<submit>>
<<redirect>>
cl_lista
<<redirect>>
<<submit>>
sv_add
<<submit>>
<<build>>
<<build>>
<<submit>>
sv_edit
cl_edit
cl_add
fr_lista_entidad
<<integer>> id_entidad
<<string>> nombre
<<string>> dirección
<<string>> nombre_director
<<integer>> telefono
<<string>> correo
<<submit>> eliminar
<<submit>> modificar
fr_edit_entidad
<<integer>> id_entidad
<<string>> nombre
<<string>> dirección
<<string>> nombre_director
<<integer>> telefono
<<string>> correo
<<submit>> modificar
fr_add_entidad
<<integer>> id_entidad
<<string>> nombre
<<string>> dirección
<<string>> nombre_director
<<integer>> telefono
<<string>> correo
<<submit>> insertar

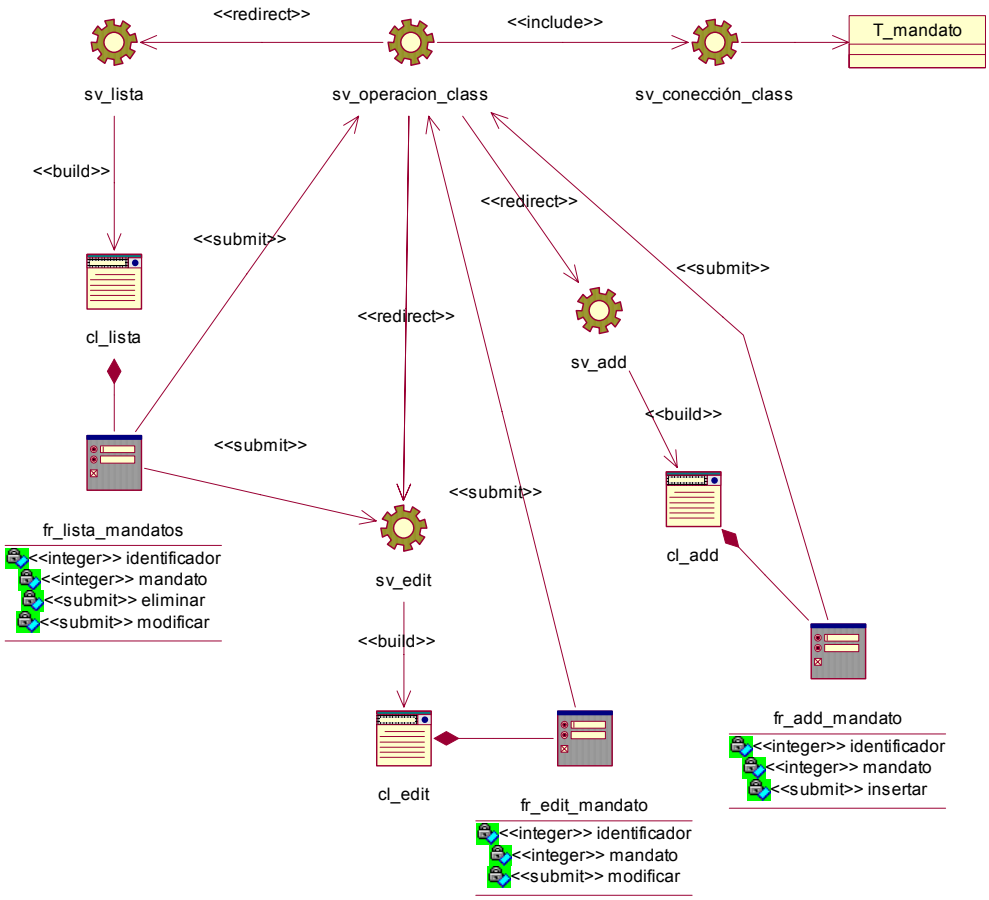
Anexo 20


<<redirect>>
<<include>>
T_clasificación
sv_lista
sv_operacion_class
sv_conección_class
<<build>>
<<redirect>>
<<submit>>
sv_add
cl_lista
<<redirect>>
<<submit>>
<<build>>
<<submit>>
<<submit>>
<<build>>
<<submit>>
cl_add
sv_edit
cl_edit
fr_lista_clasificaciones
<<integer>> id
<<string>> clasificación
<<submit>> eliminar
<<submit>> modificar
fr_edit_clasificación
<<integer>> id
<<string>> clasificación
<<submit>> modificar
fr_add_clasificación
<<integer>> id
string clasificación
<<submit>> insertar

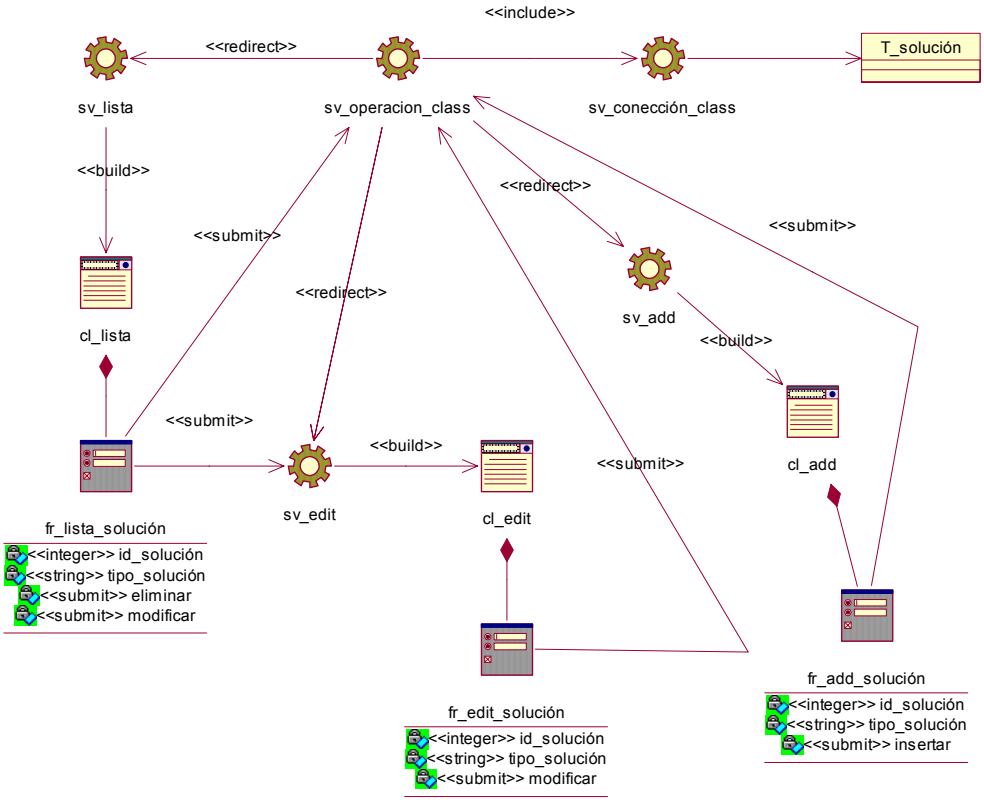
Anexo 21


<<redirect>>
<<include>>
sv_lista
sv_operacion_class
sv_conección_class
<<build>>
<<redirect>>
T_proceso
<<submit>>
<<redirect>>
sv_add
<<submit>>
cl_lista
<<build>>
<<submit>>
cl_add
<<submit>>
fr_proceso
<<integer>> id_proceso
<<integer>> número
<<submit>> eliminar
<<submit>> modificar
sv_edit
<<build>>
fr_add_proceso
<<integer>> identificador
<<integer>> proceso
<<submit>> insertar
cl_edit
fr_edit_proceso
<<integer>> id_proceso
<<integer>> número
<<submit>> modificar

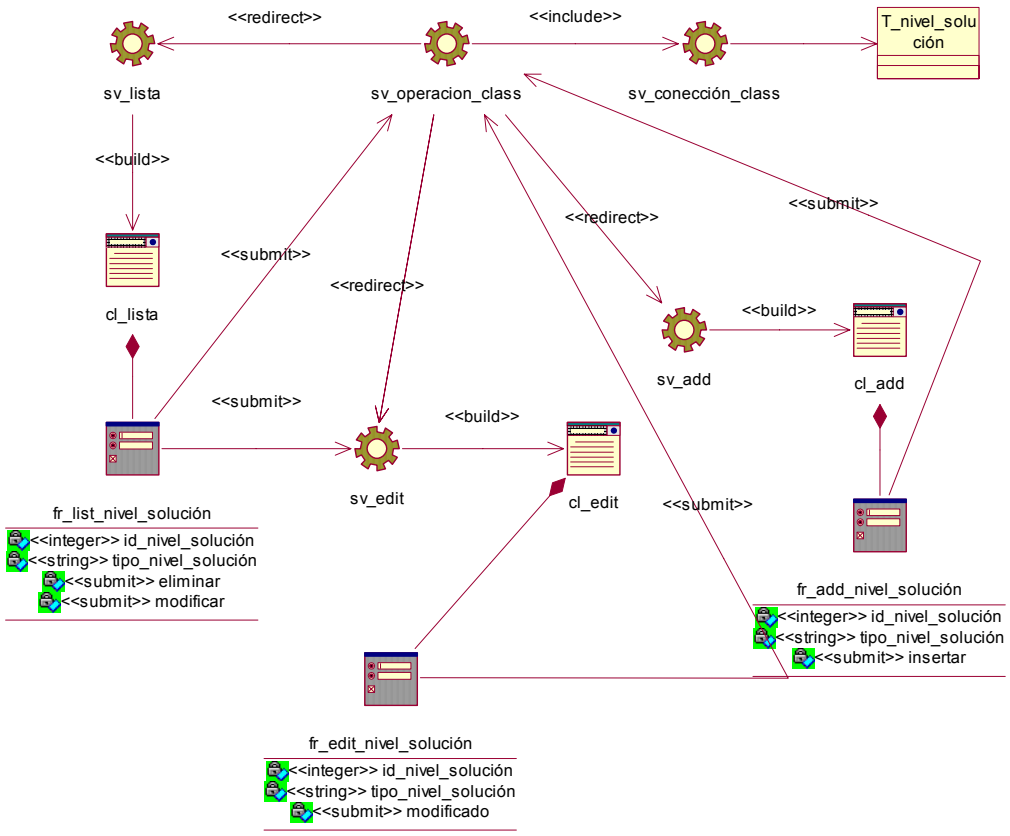
Anexo 22


<<redirect>>
<<include>>
T_mandato
sv_lista
sv_operacion_class
sv_conección_class
<<build>>
<<submit>>
<<redirect>>
<<submit>>
cl_lista
<<redirect>>
sv_add
<<build>>
<<submit>>
<<submit>>
cl_add
fr_lista_mandatos
<<integer>> identificador
<<integer>> mandato
<<submit>> eliminar
<<submit>> modificar
sv_edit
<<build>>
fr_add_mandato
<<integer>> identificador
<<integer>> mandato
<<submit>> insertar
cl_edit
fr_edit_mandato
<<integer>> identificador
<<integer>> mandato
<<submit>> modificar

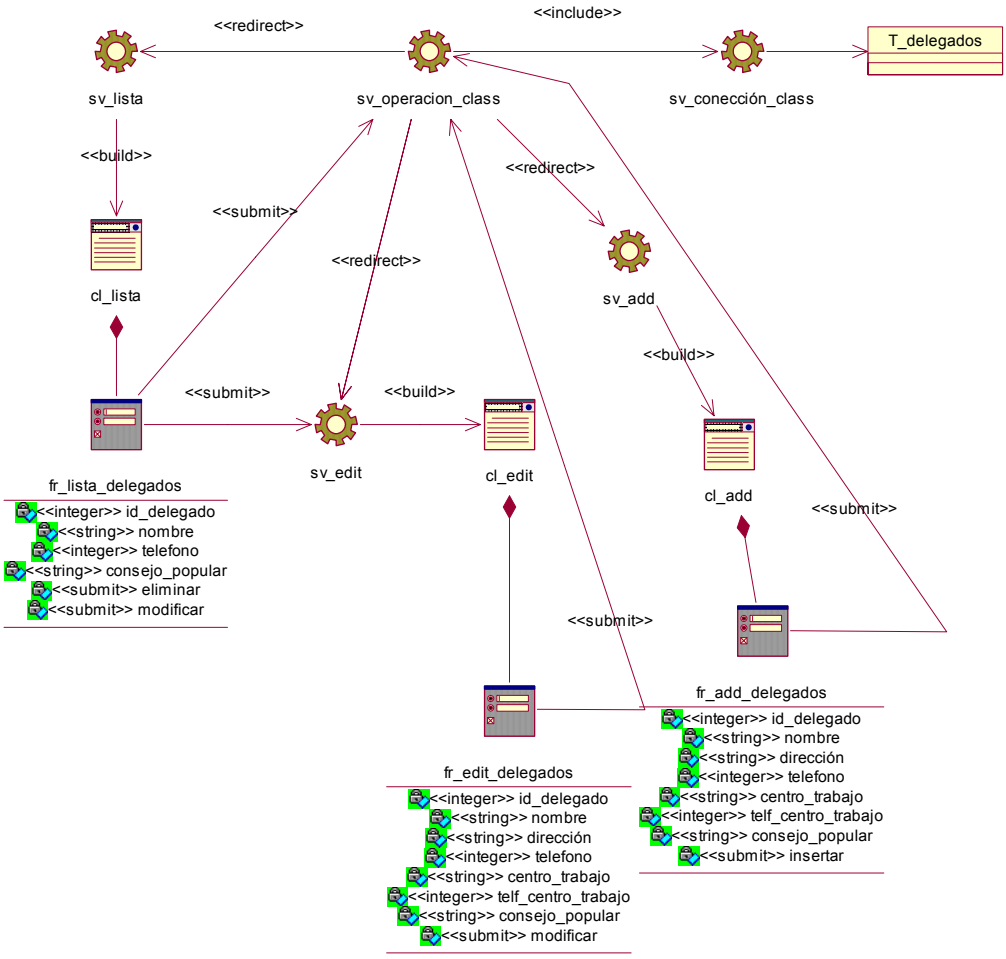
Anexo 23


<<include>>
<<redirect>>
T_solución
sv_lista
sv_operacion_class
sv_conección_class
<<build>>
<<redirect>>
<<submit>>
<<submit>>
<<redirect>>
sv_add
cl_lista
<<build>>
<<submit>>
<<build>>
<<submit>>
cl_add
sv_edit
cl_edit
fr_lista_solución
<<integer>> id_solución
<<string>> tipo_solución
<<submit>> eliminar
<<submit>> modificar
fr_add_solución
<<integer>> id_solución
<<string>> tipo_solución
<<submit>> insertar
fr_edit_solución
<<integer>> id_solución
<<string>> tipo_solución
<<submit>> modificar

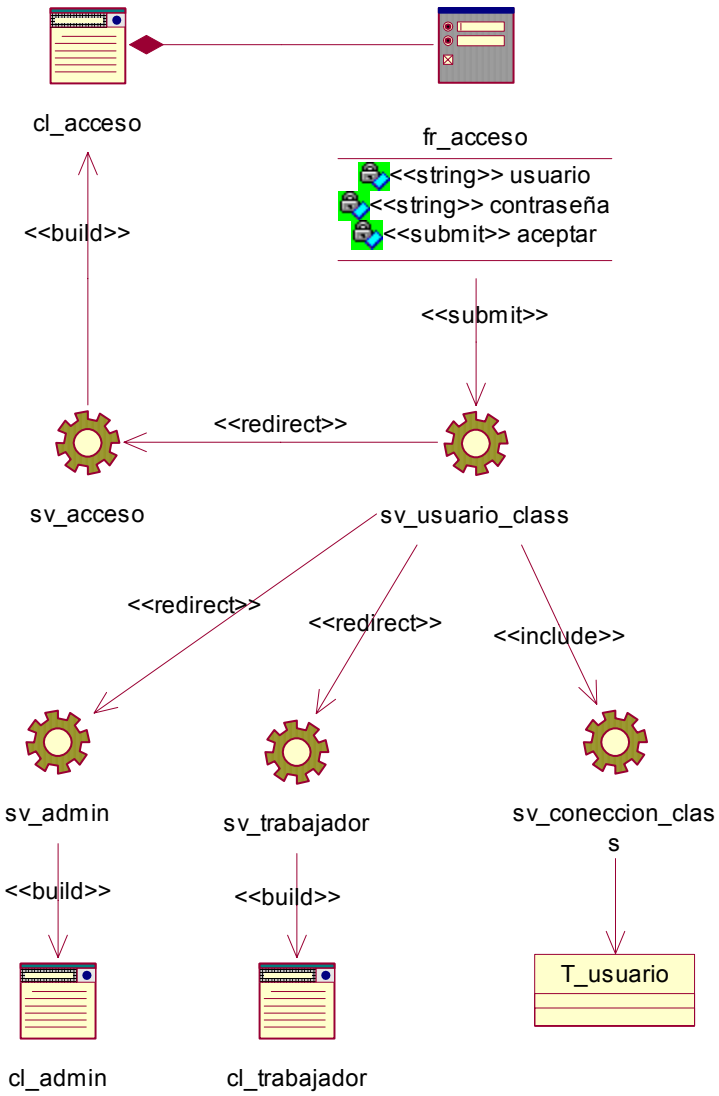
Anexo 24


<<redirect>>
<<include>>
T_nivel_solución
sv_lista
sv_operacion_class
sv_conección_class
<<build>>
<<submit>>
<<redirect>>
<<submit>>
cl_lista
<<redirect>>
<<build>>
sv_add
cl_add
<<submit>>
<<build>>
sv_edit
cl_edit
<<submit>>
fr_list_nivel_solución
<<integer>> id_nivel_solución
<<string>> tipo_nivel_solución
<<submit>> eliminar
<<submit>> modificar
fr_add_nivel_solución
<<integer>> id_nivel_solución
<<string>> tipo_nivel_solución
<<submit>> insertar
fr_edit_nivel_solución
<<integer>> id_nivel_solución
<<string>> tipo_nivel_solución
<<submit>> modificado

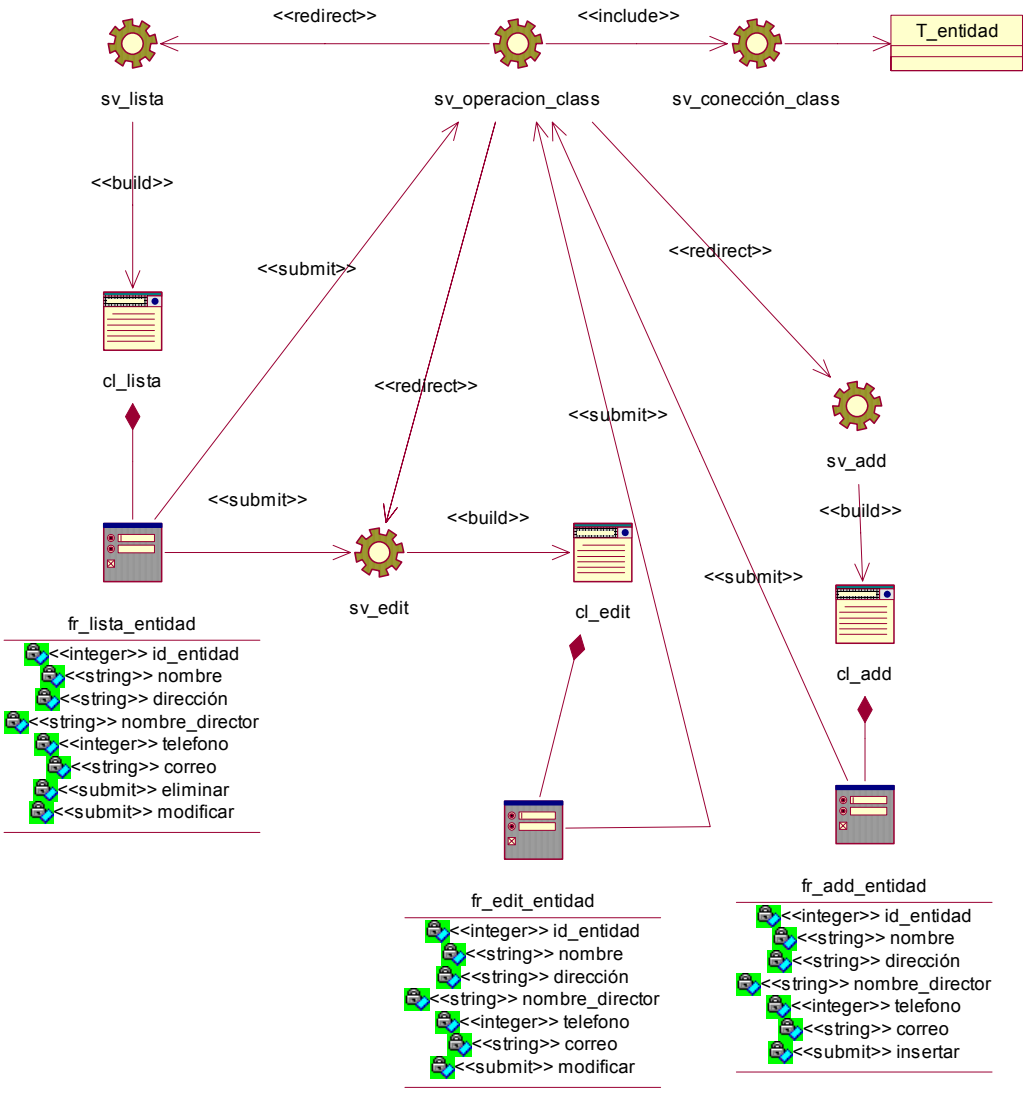
Anexo 25


<<redirect>>
<<include>>
T_delegados
sv_lista
sv_operacion_class
sv_conección_class
<<build>>
<<submit>>
<<redirect>>
<<redirect>>
cl_lista
sv_add
<<build>>
<<submit>>
<<build>>
sv_edit
cl_edit
cl_add
<<submit>>
fr_lista_delegados
<<integer>> id_delegado
<<string>> nombre
<<integer>> telefono
<<string>> consejo_popular
<<submit>> eliminar
<<submit>> modificar
<<submit>>
fr_add_delegados
<<integer>> id_delegado
<<string>> nombre
<<string>> dirección
<<integer>> telefono
<<string>> centro_trabajo
<<integer>> telf_centro_trabajo
<<string>> consejo_popular
<<submit>> insertar
fr_edit_delegados
<<integer>> id_delegado
<<string>> nombre
<<string>> dirección
<<integer>> telefono
<<string>> centro_trabajo
<<integer>> telf_centro_trabajo
<<string>> consejo_popular
<<submit>> modificar

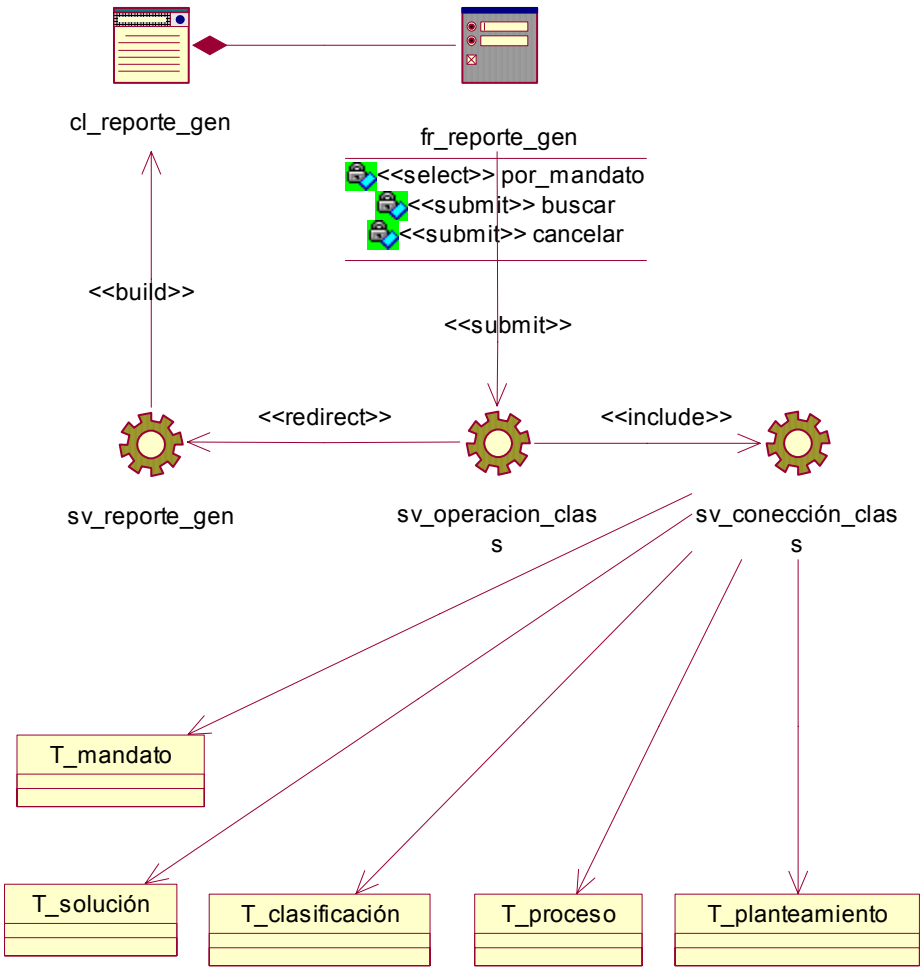
Anexo 26


cl_acceso
fr_acceso
<<string>> usuario
<<string>> contraseña
<<submit>> aceptar
<<build>>
<<submit>>
<<redirect>>
sv_acceso
sv_usuario_class
<<redirect>>
<<redirect>>
<<include>>
sv_admin
sv_trabajador
sv_coneccion_clas
s
<<build>>
<<build>>
T_usuario
cl_admin
cl_trabajador

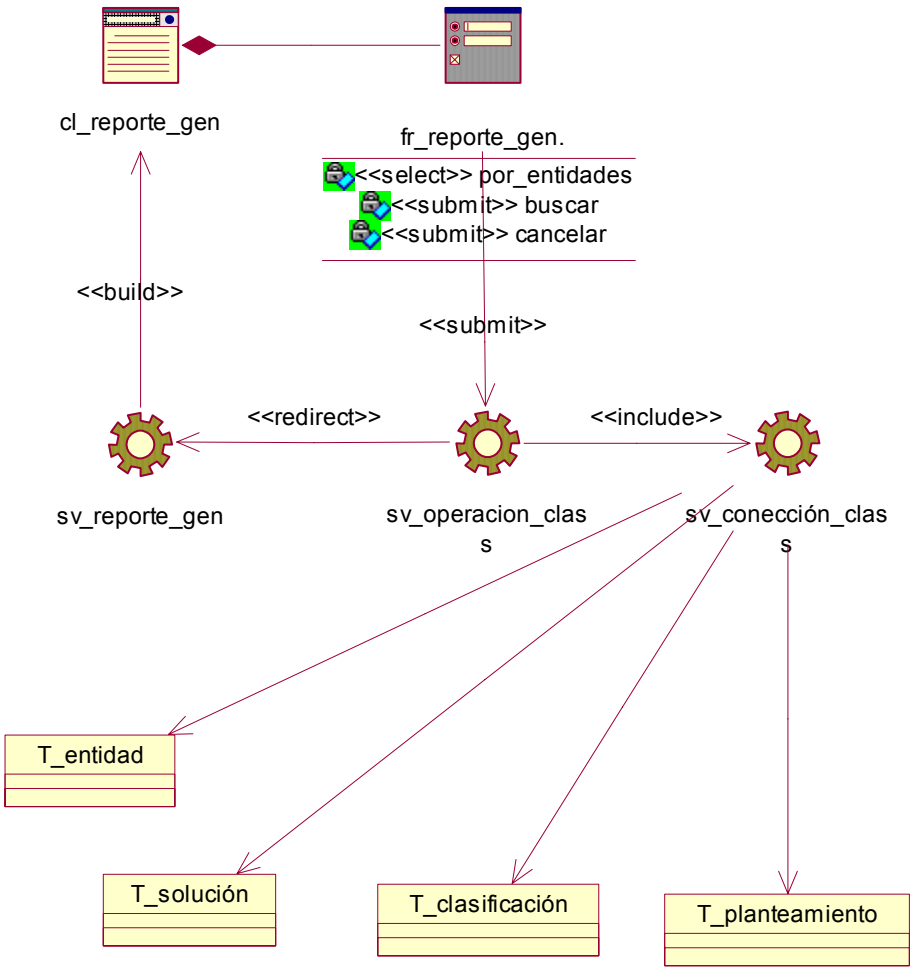
Anexo 27



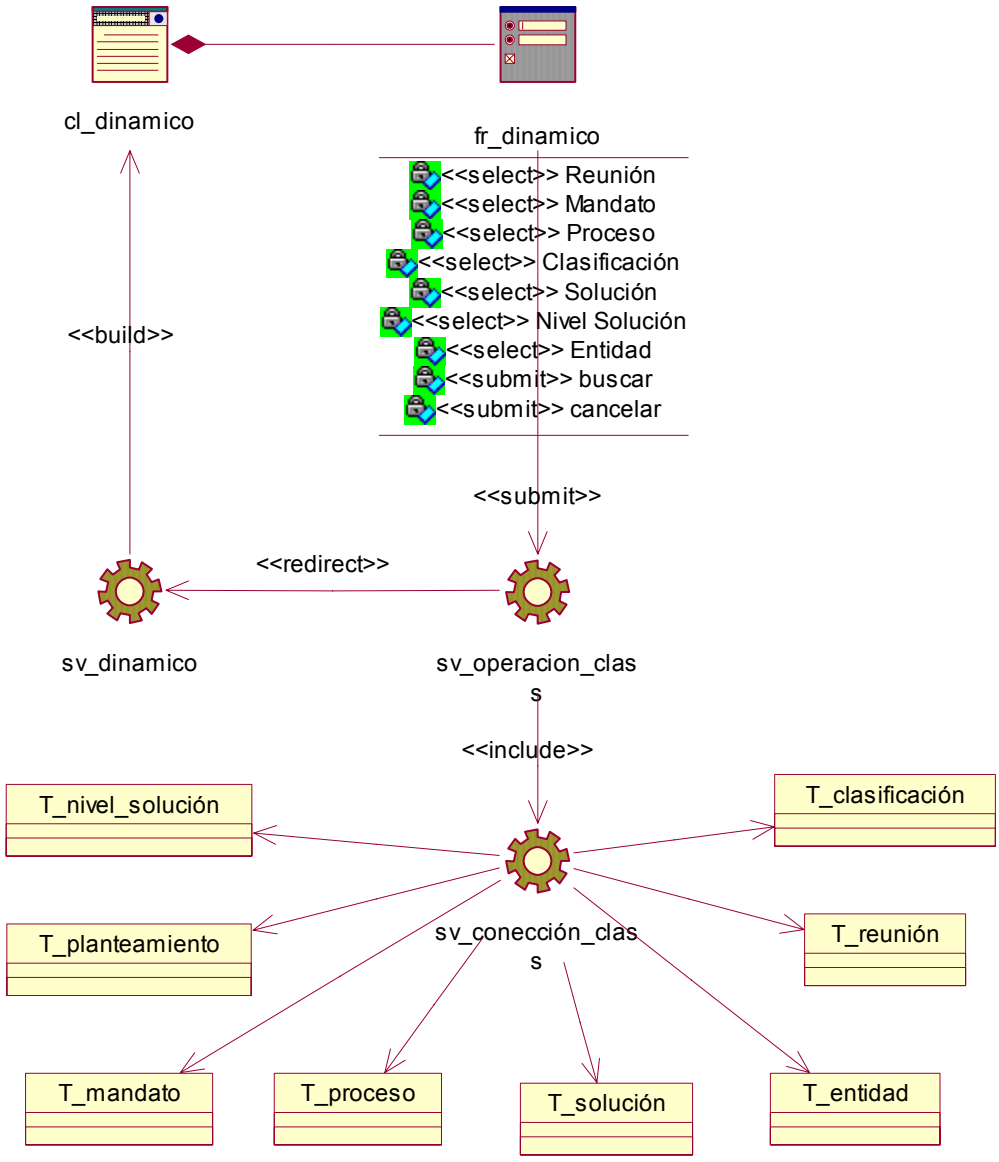
Anexo 28


cl_reporte_gen
fr_reporte_gen
<<select>> por_mandato
<<submit>> buscar
<<submit>> cancelar
<<build>>
<<submit>>
<<redirect>>
<<include>>
sv_reporte_gen
sv_operacion_clas
s
sv_conección_clas
s
T_mandato
T_solución
T_clasificación
T_proceso
T_planteamiento

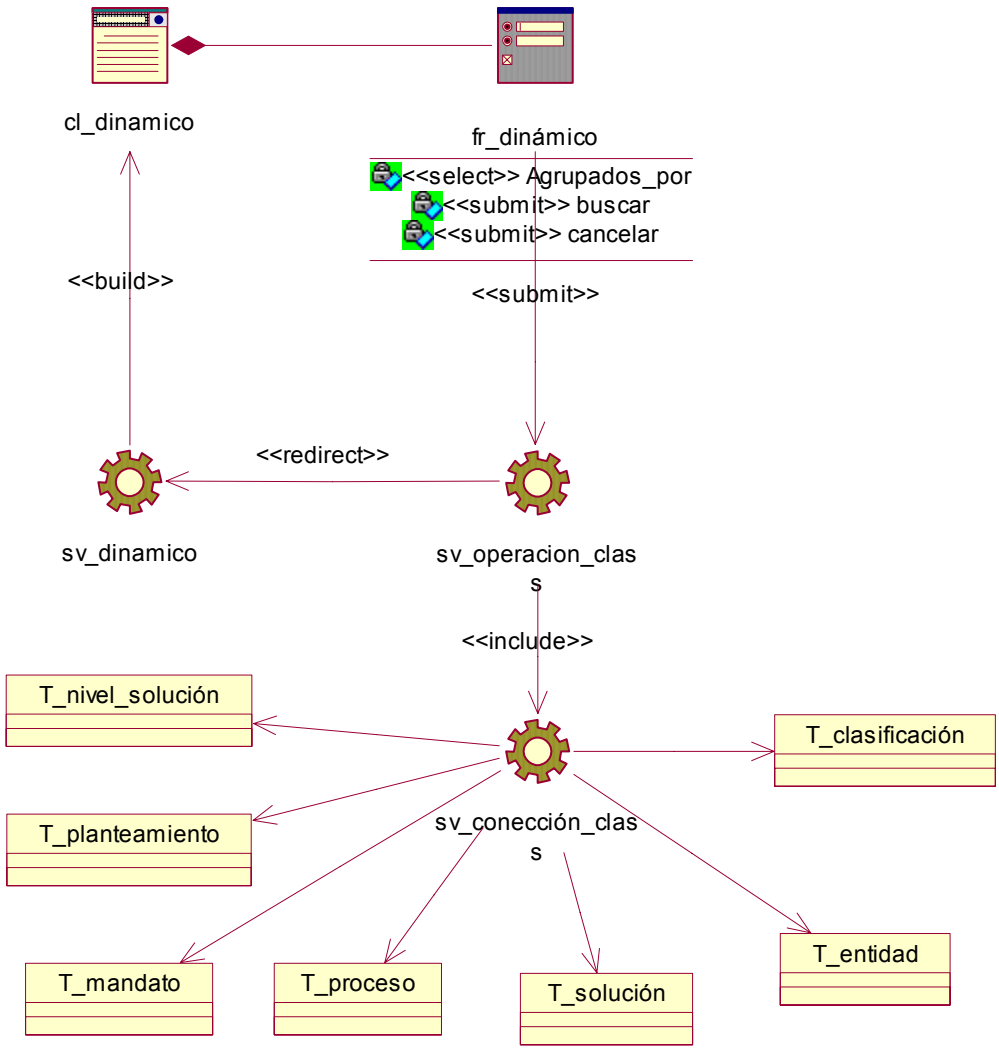
Anexo 29


cl_reporte_gen
fr_reporte_gen.
<<select>> por_entidades
<<submit>> buscar
<<submit>> cancelar
<<build>>
<<submit>>
<<redirect>>
<<include>>
sv_reporte_gen
sv_operacion_clas
s
sv_conección_clas
s
T_entidad
T_solución
T_clasificación
T_planteamiento

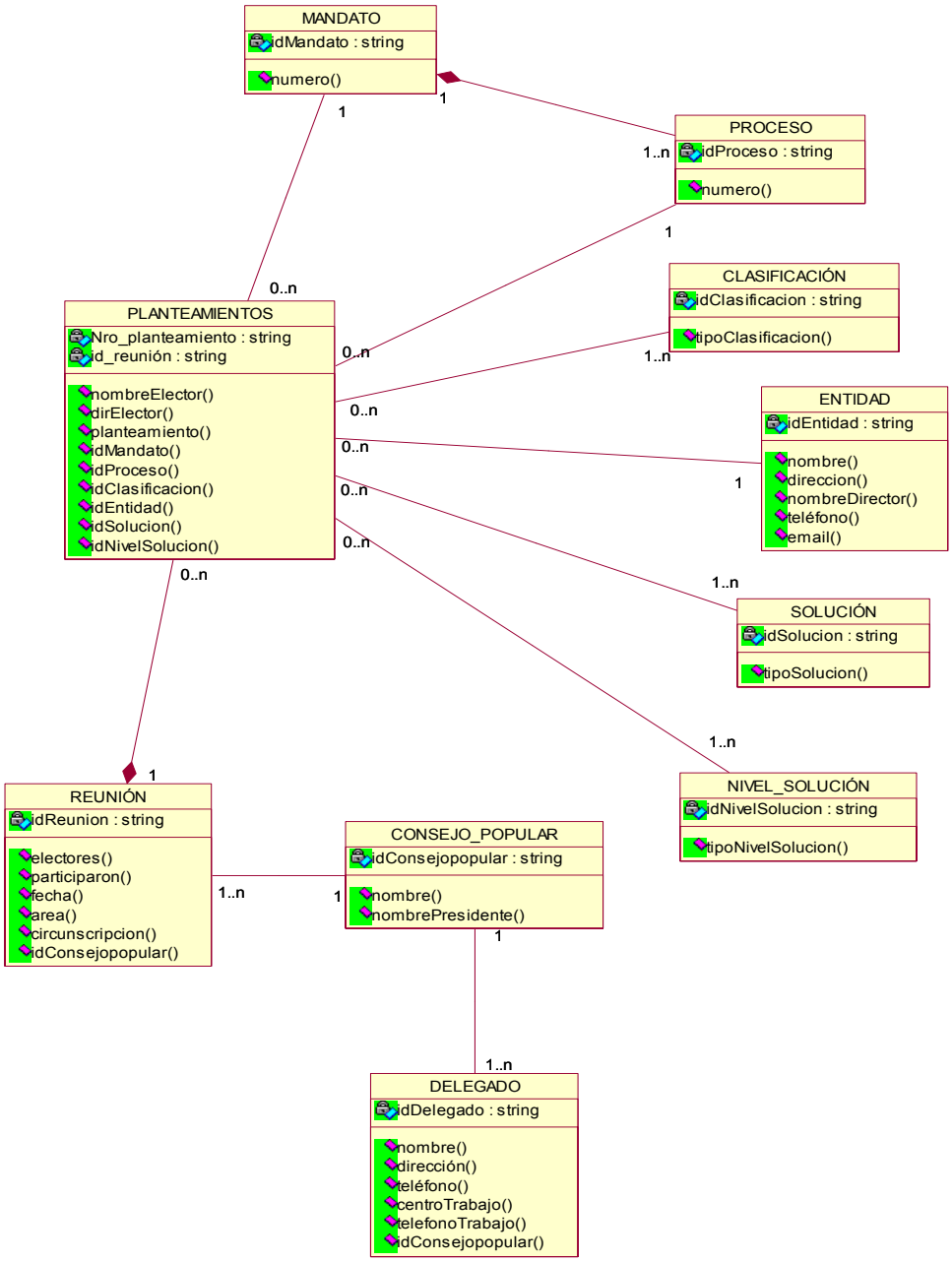
Anexo 30


cl_dinamico
fr_dinamico
<<select>> Reunión
<<select>> Mandato
<<select>> Proceso
<<select>> Clasificación
<<select>> Solución
<<select>> Nivel Solución
<<select>> Entidad
<<submit>> buscar
<<submit>> cancelar
<<build>>
<<submit>>
<<redirect>>
sv_dinamico
sv_operacion_clas
s
<<include>>
T_nivel_solución
T_clasificación
T_planteamiento
T_reunión
sv_conección_clas
s
T_mandato
T_proceso
T_solución
T_entidad

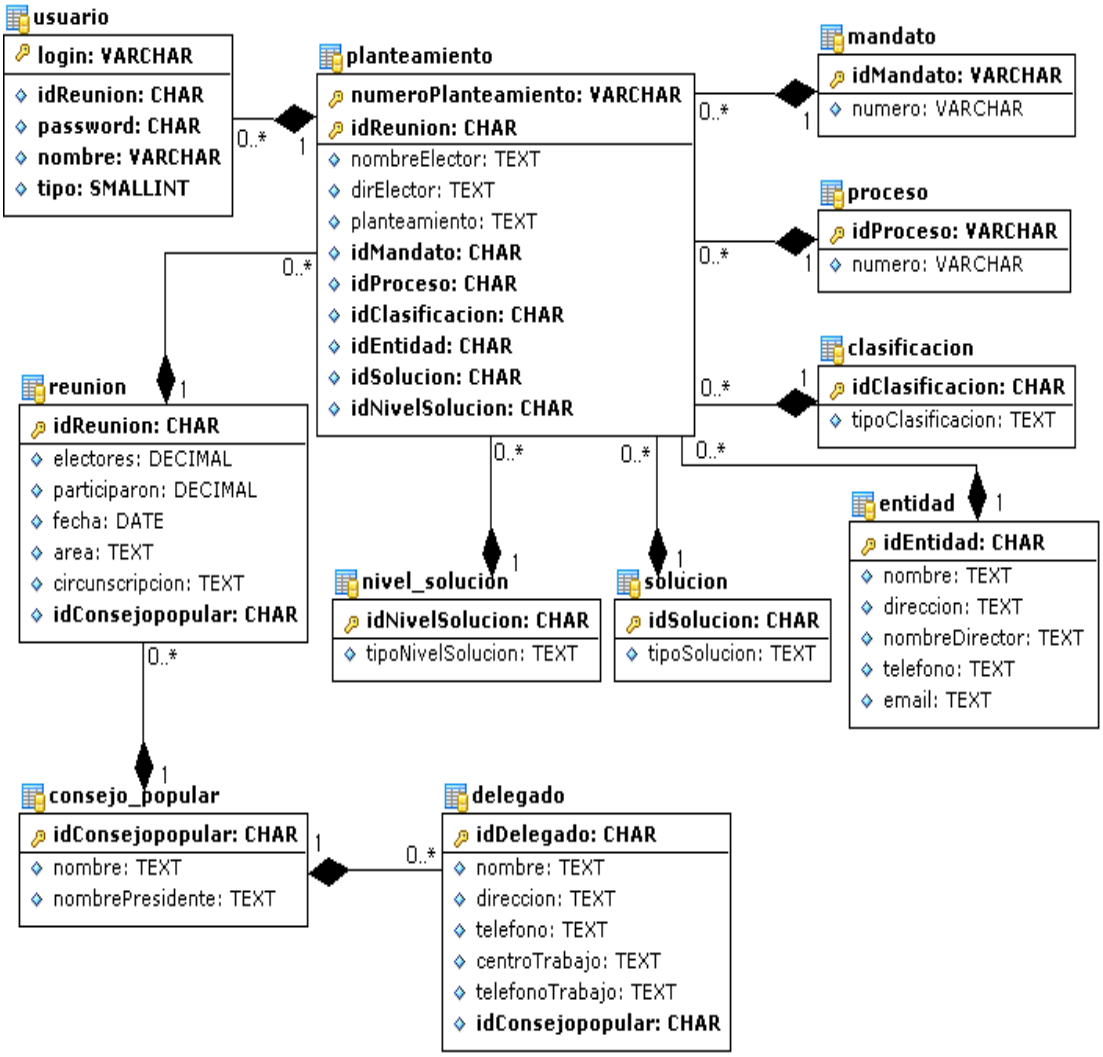
Anexo 31


cl_dinamico
fr_dinámico
<<select>> Agrupados_por
<<submit>> buscar
<<submit>> cancelar
<<build>>
<<submit>>
<<redirect>>
sv_dinamico
sv_operacion_clas
s
<<include>>
T_nivel_solución
T_clasificación
T_planteamiento
sv_conección_clas
s
T_mandato
T_proceso
T_solución
T_entidad

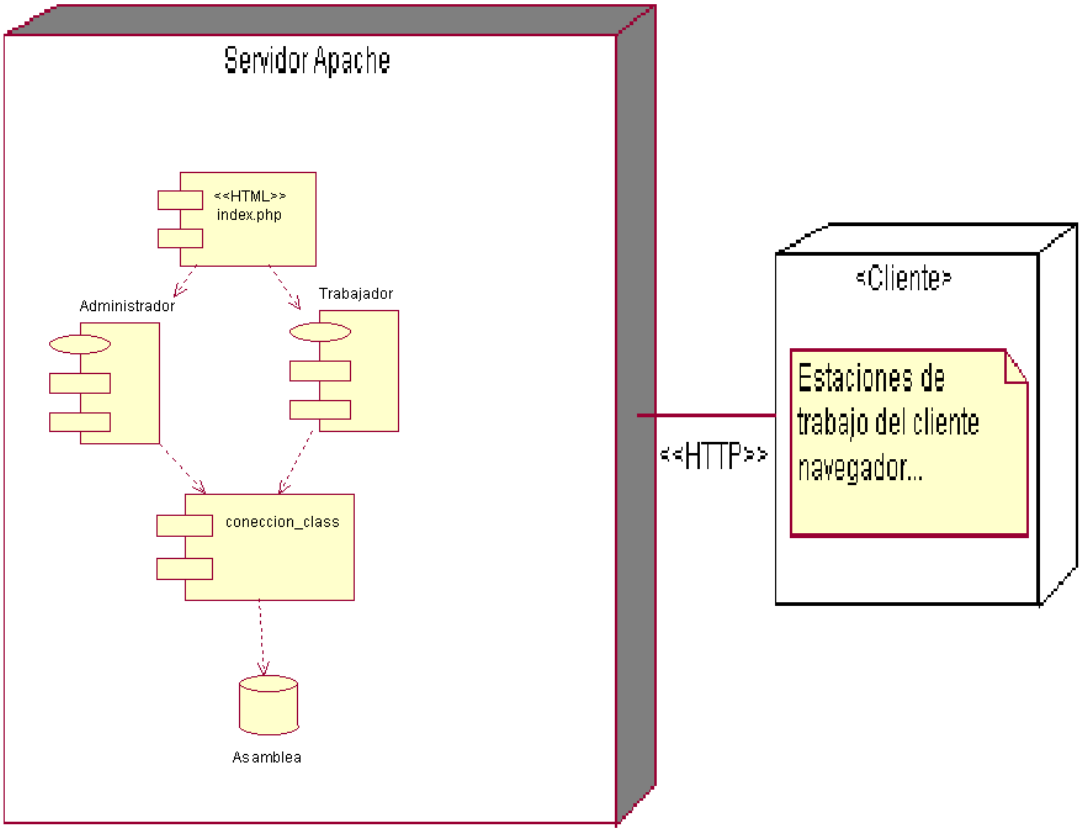
ANEXO 32


MANDATO
idMandato : string
numero()
PROCESO
idProceso : string
numero()
CLASIFICACIÓN
idClasificacion : string
tipoClasificacion()
PLANTEAMIENTOS
Nro_planteamiento : string
id_reunión : string
nombreElector()
dirElector()
planteamiento()
idMandato()
idProceso()
idClasificacion()
idEntidad()
idSolucion()
idNivelSolucion()
ENTIDAD
idEntidad : string
nombre()
direccion()
nombreDirector()
teléfono()
email()
SOLUCIÓN
idSolucion : string
tipoSolucion()
NIVEL_SOLUCIÓN
idNivelSolucion : string
tipoNivelSolucion()
REUNIÓN
idReunion : string
electores()
participaron()
fecha()
area()
circunscripcion()
idConsejopopular()
CONSEJO_POPULAR
idConsejopopular : string
nombre()
nombrePresidente()
DELEGADO
idDelegado : string
nombre()
dirección()
teléfono()
centroTrabajo()
telefonoTrabajo()
idConsejopopular()
1
1..n
0..n

Anexo 33


usuario
login: VARCHAR
idReunion: CHAR
password: CHAR
nombre: VARCHAR
tipo: SMALLINT
planteamiento
numeroPlanteamiento: VARCHAR
idReunion: CHAR
nombreElector: TEXT
dirElector: TEXT
planteamiento: TEXT
idMandato: CHAR
idProceso: CHAR
idClasificacion: CHAR
idEntidad: CHAR
idSolucion: CHAR
idNivelSolucion: CHAR
mandato
idMandato: VARCHAR
numero: VARCHAR
proceso
idProceso: VARCHAR
numero: VARCHAR
clasificacion
idClasificacion: CHAR
tipoClasificacion: TEXT
reunion
idReunion: CHAR
electores: DECIMAL
participaron: DECIMAL
fecha: DATE
area: TEXT
circunscripcion: TEXT
idConsejopopular: CHAR
entidad
idEntidad: CHAR
nombre: TEXT
direccion: TEXT
nombreDirector: TEXT
telefono: TEXT
email: TEXT
nivel_solucion
idNivelSolucion: CHAR
tipoNivelSolucion: TEXT
solucion
idSolucion: CHAR
tipoSolucion: TEXT
consejo_popular
idConsejopopular: CHAR
nombre: TEXT
nombrePresidente: TEXT
delegado
idDelegado: CHAR
nombre: TEXT
direccion: TEXT
telefono: TEXT
centroTrabajo: TEXT
telefonoTrabajo: TEXT
idConsejopopular: CHAR

Anexo34


Servidor Apache
<<HTML>>
index.php
Administrador
Trabajador
coneccion_class
Asamblea
<<HTTP>>
<Cliente>
Estaciones de trabajo del cliente navegador...

1