Universidad de las Ciencias Informáticas
Facultad 9

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Título

Sistema para la Gestión y Estandarización de los Nomencladores en la Oficina Nacional de Recursos Minerales.

Autores

Sureya Pérez Gé
Lianne Yazmín García Vázquez

Tutores

Msc. Alexeis Companioni Guerra
Ing. Neysis Hernández Díaz

UCI

Ciudad de La Habana, Julio 2008
"Año 50 de la Revolución""

"Por este mundo pasaré solamente una vez, si hay una buena obra que pueda hacer, si hay una buena palabra que pueda decir; haré esa buena obra y diré esa buena palabra, pues ya nunca volveré a pasar por aquí."

Edmundo D'Amicis "Corazón"

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

A mis padres Yazmín y Rafael, porque son la luz que me guía, mi inspiración, el orgullo de mi vida...

A mi abuela Nela, por haberme colmado de tanto amor desde que nací, porque es como una madre para mí y porque la quiero con todo mi corazón...

A mi abuela Carmen, porque a pesar de ya no estar entre nosotros, sé que hubiese sido muy feliz por este gran logro de su nieta (la princesa patas largas), por eso nunca dudé en dedicarle este trabajo...

A mis hermanos Nely y Claudio, porque son lo más grande que tengo en la vida, por el cariño tan grande que me han brindado, por hacer que me sienta cada día más orgullosa de ellos...

...de Lia

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

A mi papá por ser siempre mi inspiración y el mejor de mis recuerdos.

A mami y a Julito por haber confiado y compartido junto a mí este sueño.

...de Sure

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Quisiera agradecerle a tantas personas por ayudarme a realizar este gran sueño que no sabría por donde empezar, pero no puedo dejar de mencionar a todos los que siempre confiaron en mí y supieron tolerar mi carácter (el que por momentos se volvía completamente insoportable), y a todos los que con esfuerzo y dedicación me enseñaron a ser una mejor persona, a los que no les importó la hora ni el lugar para transmitirme y brindarme sus conocimientos, a los que de una manera u otra ayudaron para que lo que un día parecía una lejana utopía, hoy se haya convertido en una cercana realidad...

- A mis padres, porque sin su amor y comprensión hoy no fuera la persona que soy, no me alcanzaría esta vida para agradecerles todo el esfuerzo y sacrificio que han hecho para que hoy este sueño se haya podido realizar...gracias por haber confiado siempre en mí. Este logro no es solo mío, es de ustedes también...

- A mi abuela Mabel que tanto me ha apoyado y tanto le ha pedido a Dios por mí y a la que quiero con todas las fuerzas de mi ser, por ser quien es y como es, gracias abuela...

- A mi tía Mabelita, por haber estado ahí cada vez que la necesité, por soportar mis cantaletas cada vez que estaba deprimida porque algo me salía mal, por darme aliento en cada uno de esos momentos de desesperación...

- A Yadian por haber sido mi pareja, mi amigo, mi todo durante 4 años de mi universidad, por haberme mostrado que la vida tiene otros colores y porque hizo derroche de paciencia para poder lidiar conmigo por tanto tiempo...y a sus padres porque supieron aceptarme y quererme como a una hija más ...

- A Faviola y Enrique porque siempre han estado ahí para mí...

- A mis amigos, los viejos y los nuevos, a los de aquí y a los de allá, a Yoilén y a Gretel, que más que amigas, han sido como hermanas para mí...

- A mi compañera de tesis, por tolerarme todos estos meses...

- A mi amigo Armando Ortiz (Mandy), porque nunca tuvo un NO para mí, por haber sido incondicional en todo momento, porque demostró ser un verdadero amigo...

- A mis tutores Alexeis y Neysis por todos sus conocimientos transmitidos, por la largas horas de paciencia y dedicación...

- A la Revolución por concederme mi preparación y formación como profesional...

... de Lia

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Quiero agradecer a cada una de las personas que me han ayudado a ser la persona que hoy soy y con temor a olvidar a alguien quiero agradecer especialmente a:

- Mami y papi, ustedes que me han dado fuerzas para levantarme en los peores momentos de mi vida, que me han enseñado todo lo que me ha ayudado a cumplir este sueño, gracias por esforzase tanto, por confiar en que yo lo lograría.

- Nene, te voy a agradecer siempre por aguantarme, quererme, ayudarme y apoyarme en un momento tan importante de mi vida, gracias por darme ánimo para ir siempre adelante y estar cerquita de mí siempre que te necesité.

- Julito, gracias por quererme y ayudarme siempre.

- Toda mi familia por su apoyo incondicional, en especial a mi tía Eladia por sus consejos y su ayuda en estos cinco años de carrera.

- Mi compañera de tesis por soportarme todo este tiempo.

- Sandy por tener para mí, siempre que lo necesité, un buen consejo; pero sobre todo, gracias por ser mi amigo.

- Dos personas especiales por su infinita paciencia, por la dedicación, por el afán que pusieron todo este tiempo en que fuéramos personas mas preparadas cada día: Alexis y Mandy.

- Neysis por su apoyo y asesoramiento en la realización de este trabajo.

- Mis amigos, mis compañeros de aula y mis profesores de estos cinco años de universidad.

- A la Revolución por darme la oportunidad de superarme.

... de Sure

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Resumen

La Oficina Nacional de Recursos Minerales (ONRM) tiene a su cargo el proceso de almacenamiento y control de la documentación generada durante años sobre la geología de nuestro país, labor que desempeña sustentada en el empleo de listas código conformadas por elementos y sus sinónimos y para las cuales se establecen normas de empleo y modificación. Este trabajo, de modo particular, se plantea la propuesta de diseño de un sistema que permita informatizar estos procesos, dada la existencia actual de un gran número de documentos en copia dura el cual genera un flujo de información en la oficina demasiado lento e ineficiente. Además, el sistema servirá de soporte para otros subsistemas, posibilitando así elevar la eficiencia y rapidez en todas las actividades colaterales que se desarrollan en la citada entidad.

De manera adicional, el documento refiere un conjunto de resultados obtenidos durante el desarrollo de la investigación y brinda argumentos que justifican la adecuación de la solución propuesta, a la situación problémica existente. Además, se realiza un análisis comparativo entre diferentes tecnologías y herramientas hoy disponibles con el objetivo de seleccionar las más idóneas; y se concluye con el empleo del Proceso Unificado de Desarrollo de software como instrumento para el desarrollo y control de la propuesta realizada. Por último, se muestra el diseño completo de la aplicación así como un grupo de conclusiones y recomendaciones de enorme valía para la continuidad del trabajo.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Índice

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Índice de Figuras

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Introducción

Nuestro país se encuentra inmerso en un proceso de informatización, con el propósito de extender el desarrollo informático hacia todos los sectores de la sociedad. No cabe duda de que un elevado número de las empresas cubanas van camino a la adaptación de los modelos de negocio basados en las tecnologías de la información y las comunicaciones (TIC); sin embargo, el problema actual radica en que no todas disponen de la capacidad suficiente para adoptar aquellos modelos de negocio que las hagan más abiertas y competitivas. La base tecnológica de la gran mayoría de las empresas de nuestro país enfrenta una etapa de revolución y por otra parte, es necesario un cambio de mentalidad hacia posiciones más flexibles respecto al empleo de las TIC en el sector.

Ante esta realidad, actualmente constituye un reto la necesidad de demostrar a las empresas las ventajas de los distintos modelos disponibles de manera que sirva de detonador hacia un pensamiento más revolucionario sobre el cómo cada una de ellas puede beneficiarse a través de las nuevas tecnologías y los nuevos enfoques. La implantación de soluciones de este tipo siempre será paulatina y en ocasiones con determinada cuota de inercia pero sin dudas inevitable, conduciendo a la empresa y a la sociedad a una verdadera revolución tecnológica que abrirá un enorme campo de posibilidades a todos por igual.

Para una empresa u organización actual la implementación de las tecnologías de la información se ha vuelto una necesidad y una cuestión de subsistencia dentro de un mundo tan competitivo; además, un paso que lleva a una actividad dinámica eficaz, no importa el sector en el que esté operando dentro del sistema económico. No existe organización alguna que no pueda obtener un beneficio mediante la introducción de las TIC en sus procesos de producción, gestión y/o control.

El desarrollo de las TICs está provocando una migración en el funcionamiento interno de las organizaciones hacia sistemas electrónicos y digitales. Esto se logra mediante la instalación de redes informáticas y la implementación de aplicaciones y sistemas que mejoran la rapidez de las operaciones inherentes que a su vez están protegidas por mecanismos de mayor seguridad que los tradicionales. La aplicación de sistemas electrónicos modernos de esta naturaleza (i.e. aplicaciones básicas y herramientas avanzadas de gestión de los recursos de la empresa así como de la relación con los clientes, etc.), posibilita un control permanente y mucho más efectivo de todas las secciones de la organización así como una evaluación más integral de todo el sistema.

Entre las ventajas principales que se obtienen por la aplicación de las TIC's en las diferentes organizaciones y en las empresas de modo particular, podemos citar la actividad eficiente, la cual

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

garantiza que se logre un mejor funcionamiento y óptimos resultados, reduciendo factores cruciales como son los costes que con la disminución de los mismos, se logra un proceso mucho más barato y más seguro que con las soluciones clásicas. Las nuevas aplicaciones de gestión logran una disminución de los costes tanto para organizaciones que ofrecen productos como para las que ofrecen servicios; posibilitan además ahorro en el tiempo de transferencia de información y documentación, disminuyendo sustancialmente las posibilidades de error humano y gran cantidad de los errores de cálculo, lo que contribuye a la mejora de la seguridad e impide la fuga de información y es esta precisamente una de la ventajas que nos brindan las TIC's, que podemos preservar la seguridad y la información al usar una red segura que proporcionaría un acceso a los archivos y bases de datos a personas específicas. Está también el control que se establece sobre los sistemas ya que se pueden supervisar por modernas tecnologías, así como proporcionar una base de datos compleja de todo el funcionamiento dentro de una empresa con información de quién realizó alguna operación, cuánto tiempo duró la misma y con qué costes y ganancias. Por último es muy importante destacar el papel que representa el desarrollo de una imagen moderna para la empresa, este es un elemento muy importante para la promoción de la misma y de sus productos. Dicha imagen se puede conseguir por medio de diversos caminos como por ejemplo, un sitio web de presentación de la empresa, de sus productos y servicios.

Adentrándonos en un ámbito más reducido, Cuba ha tratado de mantenerse, a pesar de su estatus de país subdesarrollado y de todas las barreras que le impone el bloqueo, en contacto directo con el desarrollo de las nuevas tecnologías y los diferentes procesos por los que ha transitado la informática; es por ello que prepara a su población en el estudio de esta ciencia, al mismo tiempo que extiende la utilización de estos medios a otras esferas de la vida como: la salud y la cultura.

El período especial en Cuba se ha constituido de más de 12 años de crisis económica que comenzó con el colapso de la Unión Soviética en 1991. La depresión económica que supuso el Periodo Especial fue particularmente severa durante la primera mitad de la década de los '90 y muchos sectores de nuestra economía sufrieron su impacto directo. El Ministerio de la Industria Básica (MINBAS), considerado el tercer enclave ministerial más poderoso en la isla después de los ministerios de las Fuerzas Armadas y del Interior, no estuvo exento de las carencias de recursos por la que pasaba el país en esos años, por lo que el desarrollo óptimo de sus labores se dificultó extraordinariamente.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

En la actualidad se desarrolla en la Universidad de las Ciencias Informáticas (UCI) conjuntamente con el MINBAS un proyecto de informatización que constituye un paso más hacia el aprovechamiento más racional del conocimiento de la geología, la minería, el petróleo y hacia la solución de muchos de los problemas existentes como consecuencia de la situación antes descrita. Este proyecto lleva por nombre "Programa Nacional de Informatización del Conocimiento Geológico" (PNICG) y el mismo pretende incrementar la disponibilidad de los datos geológicos para la entidades y personal de la rama. La iniciativa tiene el objetivo de abarcar, sistematizar y poner a disposición de la sociedad el conocimiento que el país ha ido acumulando a lo largo de décadas de estudio y trabajo geológico. El proyecto a su vez contribuiría a una mejor ubicación de redes eléctricas ya que lleva implícito la georeferenciación de la información [1].

Cuando el MINBAS sufrió el impacto de la crisis económica, la Oficina Nacional de Recursos Minerales (ONRM), decide tomar como estrategia reorientar su ocupación laboral hacia la recuperación y almacenamiento de datos en un esfuerzo por salvar toda la información que se había acumulando hasta ese momento y que representaba el fruto del esfuerzo de varias generaciones y que corría el riesgo de perderse por la gran escasés de recursos para su mantenimiento y conservación. A la hora de llevar a cabo esta nueva labor se introdujo una cantidad de errores realmente significativos, principalmente en las nomenclaturas, las cuales eran utilizadas de manera diferente por cada equipo de trabajo. Debido a esta situación nos planteamos como problema científico: "La inexistencia de un sistema automatizado que posibilite la estandarización y gestión de nomencladores en la ONRM", para cuya solución se pretende "Estudiar el flujo de datos que se genera a partir del uso de los sistemas informáticos que existen en la ONRM y proponer una solución informática".

Idea a defender

Con el diseño de una aplicación informática que permita la gestión y estandarización de los nomencladores en la ONRM, se sentarán las bases para una futura implementación que impida la pérdida de información sensible por recuperación incompleta y la generación de recursos no normalizados.

Objeto de estudio

Flujo de datos que se genera a partir del empleo de los sistemas informáticos que existen en la ONRM.

Campo de acción

Proceso de gestión de nomencladores en la ONRM.

Objetivo General

Diseño de una aplicación informática que posibilite la gestión y estandarización de los nomencladores en la ONRM.

Tareas de la investigación:

1. Estudiar el flujo de datos que se genera a partir del uso de los sistemas informáticos que existen en la ONRM.
2. Realizar un estudio detallado de las tecnologías más adecuadas a emplear.
3. Analizar y diseñar un sistema informático que permita la gestión de la información referente a los nomencladores en la ONRM.

Resultados esperados

Diseñar un sistema que brinde las posibilidades siguientes:

- Lograr eficiencia en los proceso de gestión y estandarización de los nomencladores en la ONRM.
- Lograr satisfacción en los clientes con el diseño de la aplicación.

El presente documento está estructurado por cuatro capítulos, además de la introducción así como las conclusiones y recomendaciones finales. De modo particular, cada capítulo expone un grupo importante de elementos que se describen a continuación.

Capítulo 1: Fundamentación Teórica.

Describe los procesos actuales que se realizan para la gestión de las listas código, elementos y sinónimos en la ONRM; para ello se muestra el problema actual que existe en la entidad así como sistemas anteriores que aportaron soluciones inmediatas pero no garantizaron la eficiencia y control del proceso en su conjunto.

Capítulo 2: Tendencias y tecnologías actuales a considerar.
Se abordan temas relacionados con las herramientas, lenguaje y gestor de base de datos a utilizar, así como el lenguaje de modelado y la metodología a emplear. Además se describen temas asociados con la seguridad en aplicaciones Web, así como con la conveniencia e idoneidad del sistema operativo a utilizar.

Capítulo 3: Descripción de la solución propuesta.
Este capítulo contiene los primeros pasos del desarrollo de la solución, modelado del negocio y sistema; incluye también la descripción de la información que se maneja, de la aplicación que se propone así como de todos los requisitos funcionales (RF) y no funcionales (RNF) que forman parte de la propuesta.

Capítulo 4: Construcción de la solución propuesta.
Incluye la definición del modelo de análisis del sistema y del modelo de clases. Además, describe los diagramas de secuencia del modelo del diseño para cada realización de los casos de uso y muestra el diagrama de clases del diseño, modelo de datos y el modelo de despliegue

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Capítulo 1

Fundamentación Teórica

1.1 Introducción

En el presente capítulo se analizan y clasifican los conceptos necesarios para lograr una mejor investigación del tema, se describen los procesos actuales de la ONRM y además se identifican los principales problemas que motivan la investigación. De manera adicional, se analizan las soluciones existentes hasta el momento y que pudieran aportar una variante de solución más integral del problema.

1.2 Conceptos básicos

1.2.1 Listas código [2]

Una lista código no es más que una tabla que contiene los atributos: código, nombre, y comentario. Adicionalmente estas listas brindan la posibilidad de:

1. Trabajar con los códigos y no con sus nombres lo que produce un ahorro sustancial de espacio en disco.

2. Asegurar la uniformidad de los datos al no tener que usar el nombre en el procesamiento garantizando la uniformidad en todo el servicio geológico.

3. Utilizar sinónimos, garantizando la integridad del dato original y la recuperación de datos que son iguales pero con nombres diferentes (algo muy común en las Geociencias).

1.2.2. Nomenclador [2]

Caso particular de una Lista de código cuyas definiciones están aprobadas a un nivel determinado.

1.2.3. Identificador [2]

Valor numérico que identifica al tipo de rasgo dentro de la base de datos.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

1.2.4. Rasgo [2]

Es la entidad que representa un grupo de elementos georeferenciados del mundo real que contiene información geológica y tienen una representación geométrica dada, contiene los atributos de ubicación de acuerdo con el tipo geométrico establecidos en la ISO 19107.

1.2.5. Listas de unidades de medidas [2]

Las listas de Unidades de Medidas es una tabla que contiene los datos necesarios para la conversión de la unidad de medida original del dato a otro sistema de unidad diferente. Resuelve el problema de la recuperación del dato en una unidad de medida distinta a la utilizada en el momento de la captación del mismo.

Ejemplo:
En los años anteriores a 1950 se utilizaba como unidad de medida de longitud el pie (sistema inglés de unidades) posteriormente se utilizó el metro (sistema internacional de unidades). Estas listas permiten recuperar todas longitudes en Metros.

1.2.6 Aplicación Informática

Son los programas con los cuales el usuario final interactúa, es decir, son aquellos programas que permiten la interacción entre el usuario y la computadora. Esta comunicación se lleva a cabo cuando el usuario elige entre las diferentes opciones o realiza actividades que le ofrece el programa.

Juan Salvador Castejón Garrido, Secretario del CIIRM[1] ; define aplicación informática mediante la figura que se muestra a continuación.[3]

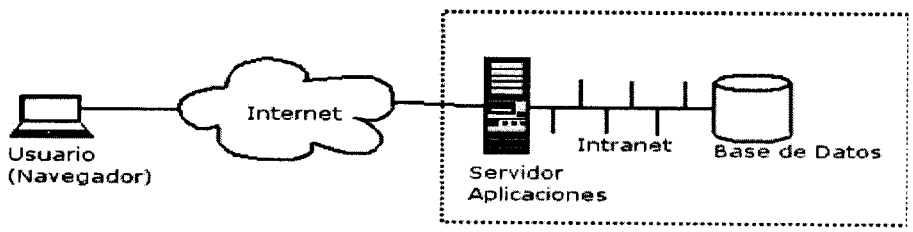


Figura # 1: Descripción gráfica de aplicación informática

[1] Colegio de Ingenieros en Informática de la Región de Murcia

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Es un programa de ordenador que se compra ya realizado y listo para usar. Las hay de muy diversos tipos, según el propósito con que se hayan diseñado: procesadores de texto, bases de datos, programas de contabilidad, de facturación, entre otras.[4] Las aplicaciones por su uso se pueden dividir en dos grandes tipos:

- Aplicaciones de Escritorio
- Aplicaciones Web

1.2.7 Aplicación Web

Con el surgimiento de Internet se abrieron nuevos horizontes para la publicación de información de toda índole y diversos son los métodos que hoy día existen con este fin. Las aplicaciones Web, por ejemplo, son las encargadas de manejar el estado del negocio y de gestionar datos almacenados con ayuda de algún Sistema Gestor de Base de Datos (SGBD) con el fin de poder ser utilizadas por una amplia gama de usuarios.

Una aplicación Web es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Es un sitio que se encuentra en la Red, al cual los usuarios acceden con el fin de beneficiarse con el uso de la información que posea. Aborda un determinado tema. Se dice que los usuarios se benefician con la información que este les brinda porque una vez que acceden a él, pueden interactuar con la aplicación, actualizarla o modificarla según los privilegios de seguridad de la misma que estos tengan.

1.2.8. Estructura de las aplicaciones Web

Las aplicaciones Web pueden ser estructuradas de diversas formas, todo depende de los desarrolladores de la misma. En la actualidad las que más se desarrollan son las aplicaciones por capas, y de este conjunto las más difundidas son aquellas desarrolladas en tres capas lógicas (Modelo-Vista-Controlador):

- Presentación de la aplicación web (vista)
- Lógica de negocio (controlador)
- Lógica de servidor (modelo)

Entre las capas que conforman la aplicación web existe una gran relación y comunicación. Su funcionamiento se sustenta en efectuar peticiones desde la capa interfaz hacia la capa de acceso a

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

datos a través de la capa Lógica o de Negocio, pues la capa de Presentación no puede comunicarse directamente con la de almacenamiento ya que esta arquitectura consiste en aislar la lógica de la aplicación y en convertirla en una capa intermedia bien definida y lógica del software. En la capa de presentación se realiza relativamente poco procesamiento de la aplicación; las ventanas envían a la capa intermedia peticiones de trabajo y éste a su vez se comunica con la capa de almacenamiento del extremo posterior.[5]

1.3 Descripción del dominio del problema

En Cuba existe un organismo nombrado (MINBAS), el cual fue creado en 1980 y es el encargado de dirigir, ejecutar y controlar la política del Estado y el Gobierno, en cuanto a las actividades de:

- Generación, transmisión, distribución y comercialización de la energía eléctrica
- Producción de la industria del combustible y los lubricantes, y su distribución
- Búsqueda, exploración y extracción de petróleo y gas
- Búsqueda, exploración, extracción y beneficio de minerales sólidos
- Producción de fertilizantes, fibras químicas, gases industriales, artículos de plásticos, plaguicidas, pinturas, colorantes, barnices y otros productos químicos
- Producción de neumáticos, cámaras y productos del caucho
- Producción y conversión de papel, cartón y pulpa celulosa
- Producción de vidrio y sus artículos

Este ministerio tiene una estructura de ocho uniones que comprenden las ramas de generación y distribución de electricidad; extracción, refinación y distribución de petróleo; extracción y procesamiento de Níquel; producción de sal, extracción y procesamiento de minerales; producción de pinturas, fertilizantes, gases industriales y medicamentos; industria de neumáticos, de papel, cemento y vidrio. En dicho organismo no existen delegaciones territoriales, solo existe en cada provincia un Consejo de Cooperación formado por las empresas de ese territorio, con un presidente que es el director de una de estas empresas y que su función es de coordinación e información.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Adicionalmente, existen cuatro entidades independientes que realizan las funciones siguientes:

- La ONRM: es la rectora nacional para garantizar la racional explotación y utilización de los recursos minerales e implementar el marco jurídico para el desarrollo y control de la geología, la minería y el petróleo.
- Escuela Superior: es la encargada del sistema de superación e información de jefes y trabajadores sobre temas de carácter técnico, político y económico.
- Sanatorio Nacional Obrero: brinda servicios médicos a los trabajadores de la Industria Básica.
- Empresa de Computación: ejerce un doble papel, al asumir la función estatal rectora de la Computación, Automatización y Comunicaciones en el MINBAS, y a la vez, una función empresarial, ambas de carácter nacional dentro del Ministerio.

Nuestro trabajo esta dirigido específicamente al análisis y solución de los problemas existente en la ONRM que como bien se explicó anteriormente es la autoridad minero petrolera de la República de Cuba; es además la encargada de administrar el conocimiento geológico y toda la información geológica, minera y petrolera de la nación, con el fin de que todos lo datos se encuentren almacenados de manera organizada y tener conocimiento de cada unas de las actividades que se realicen en el sector de la geología y de la minería a nivel nacional.

En esta oficina en el año 2001 el Comité Ejecutivo del Consejo de Ministros de Cuba aprueba las funciones que debe realizar la ORNM (acuerdo 3985), adscripta al Ministerio de la Industria Básica y son:

- Fiscalizar y controlar la actividad minera y el uso racional de los recursos minerales, según lo dispuesto en la legislación vigente.
- Fiscalizar y controlar las actividades de exploración-producción de hidrocarburos líquidos y gaseosos, según las disposiciones legales vigentes.
- Asesorar al Ministerio de la Industria Básica y demás organismos de la administración central del estado sobre las actividades mineras de exploración producción de hidrocarburos, sin perjuicio de sus debidas competencias.
- Responder por el registro minero y el registro petrolero y mantener actualizadas las anotaciones sobre las concesiones mineras, áreas mineras reservadas, los contratos de exploración-

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

producción de petróleo, yacimientos, manifestaciones minerales y de hidrocarburos, áreas en investigación, minas y pozos de hidrocarburos en explotación o abandonados.

- Ejercer la inspección estatal sobre las personas naturales y jurídicas que ejecuten actividades mineras y de exploración producción de hidrocarburos, en el territorio nacional o en la zona económica exclusiva, para controlar y comprobar el cumplimiento de las disposiciones legales vigentes que rigen estas actividades.
- Aprobar los cálculos de reservas minerales sólidos, líquidos y gaseosos, registrar y mantener actualizado el inventario de dichas reservas, así como certificar el grado de aprobación de dichas reservas para su asimilación industrial u otros fines.
- Ser el depositario de la información geológica, minera y petrolera de la nación, recibir, organizar y conservar la información, así como brindar servicios de información técnica.
- Emitir dictámenes técnicos sobre el otorgamiento, anulabilidad y extinción de las concesiones mineras.
- Fiscalizar el cumplimiento de las condiciones bajo las cuales se otorgó la concesión.
- Proponer y en su caso dictar las disposiciones de carácter técnico que regulen la investigación, la explotación, el procesamiento y el uso de los recursos minerales.
- Proponer y en su caso dictar las disposiciones técnicas que regulen los trabajos de exploración-producción de petróleo, basadas en el óptimo conocimiento de las características geológicas y el potencial petrolífero de las zonas objeto de exploración, así como en la protección y el uso racional de las reservas de las reservas de hidrocarburos.
- Aprobar los proyectos de explotación y procesamiento, según corresponda, de minas, aguas minerales, fangos medicinales, salinas e hidrocarburos, de acuerdo con las disposiciones legales vigentes.
- Controlar la ejecución de los planes de preservación del medio ambiente y de las medidas para mitigar el impacto ambiental establecidas por los organismos competentes a las entidades que en el territorio nacional o en la zona económica exclusiva de Cuba realicen actividades mineras y petroleras.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

- Mantener actualizadas las estadísticas mineras y de exploración-producción de petróleo y gas del país.
- Controlar el estado del fondo de pozos de petróleo y gas natural del país y dictar las disposiciones que permitan establecer un efectivo control de los pozos, como parte principal del sistema de explotación de los yacimientos.
- Participar en el cierre de minas, yacimientos y pozos de petróleo, certificar dicho cierre y controlar las medidas del programa de cierre que se ejecuten.
- Proponer y elaborar la documentación para la declaración de las áreas mineras reservadas
- Rectorar la organización y división del territorio nacional y la zona económica exclusiva de Cuba en bloques o parcelas para la realización de trabajos de exploración-producción de petróleo y gas, certificando sus dimensiones y límites a los efectos de considerar los mismos como el objeto de los contratos petroleros que se negocien con personas extranjeras.
- Revisar y elaborar dictámenes para el nivel superior correspondiente, sobre los proyectos de contratos petroleros que se negocien con personas extranjeras en los aspectos de su competencia. Asimismo, elaborar dictámenes en los casos de paralización, anulación y extinción de dichos contratos.
- Controlar la ejecución de las disposiciones vigentes sobre la seguridad minera y petrolera en las actividades de exploración-producción y dictar las medidas que correspondan cuando se detecte la posibilidad de ocurrencia de un accidente de trabajo o peligro para la seguridad humana.
- Procesar y generalizar la información especializada conservada en su archivo técnico.
- Brindar servicios de consultoría y asesoría técnica en la esfera de su competencia.
- Asesorar a los organismos y entidades competentes en la evaluación de planes, programas de desarrollo ramales y estudios de factibilidad, de los proyectos de explotación de los recursos minerales y los hidrocarburos.
- Aprobar y controlar los objetivos que se ejecuten financiados por el presupuesto estatal para la actividad geológica de minerales y de hidrocarburos.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

La Oficina Nacional de Recursos Minerales tiene una estructura de un solo nivel, su oficina central, que tiene las direcciones que a continuación se relacionan:

- Dirección General.
- Dirección de Documentación e Informática.
- Dirección de Control Económico.
- Dirección de Registro Minero.
- Dirección de Evaluación Económica.
- Dirección de Minerales.
- Dirección de Níquel.
- Dirección de Hidrocarburos.

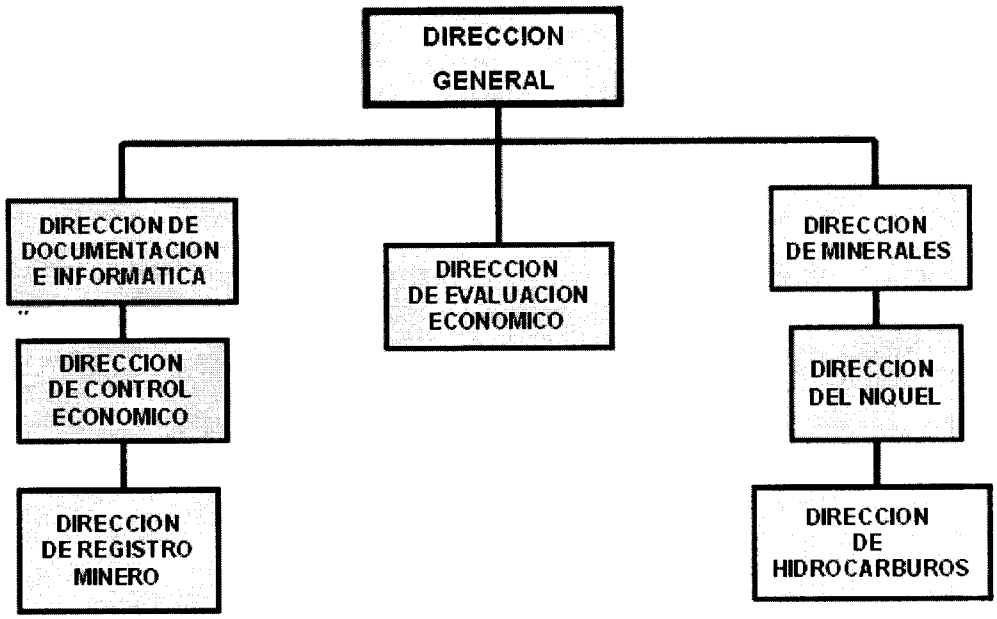


Figura # 2. Estructura de la ONRM.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

1.3.1 Situación Problemática

Actualmente la ONRM se encuentra inmersa en un importante proceso de informatización, la misma está desarrollando un grupo de proyectos que forman parte del programa integral PNICG que se lleva a cabo de conjunto con la Universidad de las Ciencias Informáticas, con el fin de adentrarse en un proceso de informatización de sus diferentes actividades. Entre las principales acciones a emprender en la ONRM se encuentran el control de concesionario, el archivo técnico, el balance de recursos, las reservas del país, los meta datos y la información de las investigaciones geológicas; las que a través de un proceso de informatización del flujo de informaciones posibilitará una mayor eficiencia y rapidez en toda la actividad de servicio, área esta última, que actualmente posee muchos puntos débiles debido fundamentalmente al gran número de información en copia dura (papel) disponible. De manera adicional, esta realidad cuenta con la agravante de que con el paso del tiempo gran parte de esta información se ha perdido o deteriorado, provocando que el flujo de información en la oficina sea mucho más lento e ineficiente, además de permanecer concebidas todas aquellas actividades ya mencionadas de forma manual y por tanto complejas y agotadoras para el personal responsable de llevar a cabo dichas tareas.

En el de proyecto que se desarrolla en la UCI, debido a la situación que tiene la ONRM, planteada anteriormente, surgen un gran número de aplicaciones, las cuales utilizan informaciones en común llamadas listas código que no son más que descripciones de la información que utiliza cada aplicación. Estas listas código se usan por separado y en algunas ocasiones están repetidas en varias aplicaciones o incluso la misma información aparece con nombres diferentes provocando que se dificulte el trabajo al hacer uso de ellas, además de las dificultades que aporta el hecho de que cada una de estas aplicaciones no tiene forma de interactuar entre sí.

Según lo descrito anteriormente la ONRM conjuntamente con la UCI ha decidido crear y brindar a través de una aplicación que sea capaz de ofrecer y mostrar de manera eficiente y confiable toda la información referente a las listas códigos, para que de esta forma el resto de las aplicaciones que se desarrollarán en el proyecto PNICG puedan hacer uso de las mismas, con el fin de tener mejor control de ellas.

- Control, porque toda la información estaría almacenada, segura y disponible para aquellos usuarios que tengan acceso a la misma.

1.3.2 Análisis de otras soluciones existentes

Utilizar las listas código es una actividad que se realiza diariamente no solo en la ONRM sino también en otros centros asociados a la misma. Uno de los propósitos de este trabajo es lograr que este proceso se lleve a cabo con rapidez, confiabilidad y seguridad. En la actualidad existen países que cuentan con mecanismos que permiten asegurar cada una de las funcionalidades que pretendemos hacer pero están soportadas en plataformas que no son libres y que además necesitan máquinas de rápido procesamiento.

Hoy se utiliza en la ONRM un sistema que permite ciertas funcionalidades pero es aplicado solo para un número reducido de provincias como Santiago de Cuba, Pinar del Río y Ciudad de la Habana esto reduce la facilidad de uso para el usuario e impide que el proceso se lleve a cabo de manera rápida y eficiente.

1.4 Conclusiones del capítulo

1- El sistema informático existente en la ONRM no es óptimo.

2- La construcción de un sistema informático en la ONRM resulta beneficioso para utilizar de manera eficiente los nomencladores y listas código en el registro de información.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Capítulo 2

Tendencias y Tecnologías a Utilizar

Ante el desafío que impone el avance de las nuevas TICs, merece destacarse que el desarrollo tecnológico está produciendo cambios significativos en la estructura económica y social, y en el conjunto de las relaciones sociales. La evolución histórica de las TICs, en su corta andadura de menos de 40 años, vive un camino paralelo al de las corrientes organizativas que han sido implantadas en las empresas. La coincidencia de este hecho obedece a una condición de reciprocidad entre los cambios organizativos y los tecnológicos que los posibilitan. Hoy en día, los progresos en las tecnologías de la información, que abarcan los equipos y aplicaciones informáticas y las telecomunicaciones, están teniendo un gran efecto; de hecho, se dice que estamos en un nuevo tipo de sociedad llamada Sociedad de la información o Sociedad de Conocimiento. En este capítulo, de forma particular, se hace un análisis de las tecnologías y tendencias que existen en la actualidad a nivel mundial y que serán útiles en el desarrollo de la propuesta de solución informática que se defiende. Se analizan también un conjunto de herramientas encaminadas a enriquecer el diseño del sistema.

2.1. El Lenguaje Unificado de Modelado (UML) como soporte de lenguaje orientado a objeto para el modelado de aplicaciones Web.

UML[2] es una especificación de notación orientada a objetos. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto. El UML se define como un "lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software..."[6]
El Lenguaje Unificado de Construcción de Modelos introduce nuevos diagramas que representan una visión dinámica del sistema y a partir de estos pueden inferirse las partes del mismo que necesitan ser sincronizadas, así como del estado de cada una de las instancias durante todo momento.

Intenta también solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación

[2] Unified Modeling Language (Lenguaje Unificado de Construcción de Modelos)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

UML es ahora un estándar, no existe otra especificación de diseño orientado a objetos, ya que es el resultado de las tres opciones existentes en el mercado. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

UML permite la modificación de todos sus miembros mediante estereotipos y restricciones. Un estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. Una restricción identifica un comportamiento forzado de una clase o relación, es decir mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

Con el objetivo de lograr un buen entendimiento se decidió utilizar UML que es un lenguaje gráfico que consiste en visualizar, especificar y construir documentos y artefactos que permiten la buena documentación que será la cantera para versiones futuras.

2.2. Metodología de Desarrollo de Software

Todo proyecto por sencillo que sea necesita un nivel de organización determinado, una guía que evite las pérdidas millonarias que ocurren a nivel mundial como consecuencia de no aplicar una metodología que respalde y organice el proceso que se desarrolla, de ahí que como parte de la organización e investigación de este trabajo se propone el estudio de tres metodologías que son no solo usadas en la UCI sino también a nivel internacional.

2.2.1. Proceso Unificado de Desarrollo de Software (RUP)

Mucho se discute actualmente sobre las ventajas o desventajas del uso de RUP como metodología de desarrollo de software, algunos autores plantean que el hecho de escribir a priori todo lo que se desea en el, de forma que aseguremos que el cliente y los desarrolladores estén de acuerdo en lo que se va a hacer, de alguna forma, elimina parte del encanto de la programación. Los programadores, en vez de dedicarse a lo que se supone que les gusta, programar, debían dedicar bastante de su tiempo a hacer y leer documentos y reuniones; pero precisamente RUP asegura que luego de escribir lo que se va a hacer se logra que los muchos programadores, en primer lugar trabajen más o menos con un objetivo común y organizados y en segundo lugar que respondan a las exigencias del cliente

El proceso unificado de desarrollo, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar el máximo provecho de los conceptos

asociados a la orientación a objetos y al modelado visual. Cuenta con las mejoras prácticas del modelo de desarrollo de un software en particular. Se define además entre sus principales características un:

1) Desarrollo de software de forma iterativa.

2) Manejo de requerimientos.

3) Utiliza arquitectura basada en componentes.

4) Modela el software de forma visual, usando UML.

5) Verifica la calidad del software.

6) Controla los cambios.

7) Dirige las tareas de cada desarrollador por separado y del equipo como un todo.

8) Especifica los artefactos que deben desarrollarse en cada fase de desarrollo del software [7].

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- Inicio
- Elaboración
- Construcción
- Transición

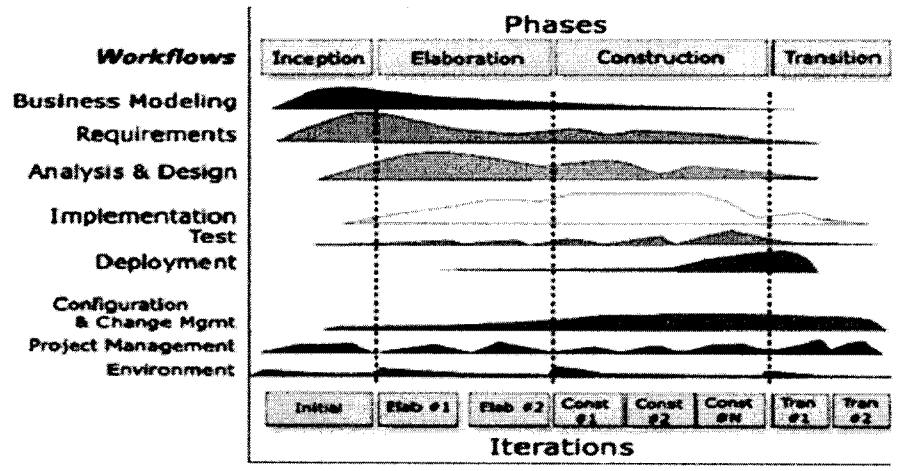


Figura # 3: Proceso Unificado de Desarrollo de Software

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Y como se ilustra en la figura se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo:

- Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

- Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

- Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

- Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

- Prueba (Testeo): Busca los defectos a los largo del ciclo de vida.

- Instalación: Produce una liberación (reléase) del producto y realiza actividades (empaque, instalación y asistencia a usuarios) para entregar el software a los usuarios finales.

- Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

- Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

- Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

En el caso específico de las aplicaciones Web, las actividades establecidas por dicho proceso son suficientes para garantizar cubrir todos los aspectos de los entornos de este tipo de aplicación. Él ciclo de vida de RUP se caracteriza por ser:

1. Dirigido por Casos de Uso
2. Centrado en la Arquitectura
3. Iterativo e Incremental

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.
La metodología RUP es la más adaptable para proyectos de largo plazo. Es una metodología completa y extensa que intenta abarcar todo el mundo del desarrollo de software, tanto para pequeños proyectos, como proyectos más ambiciosos de varios años de duración. Por lo que existe una gran cantidad de documentación sobre el mismo, tanto en libros como en la red [8].

2.2.2. Programación Extrema (XP)

En la programación extrema se da por supuesto que es imposible prever todo antes de empezar a codificar. Es imposible capturar todos los requisitos del sistema, saber qué es todo lo que tiene que hacer ni hacer un diseño correcto al principio [9].
Básicamente la idea de la programación extrema consiste en trabajar estrechamente con el cliente, haciéndole mini-versiones de producto con mucha frecuencia (cada dos semanas [9]). En cada mini-versión se debe hacer el mínimo de código y lo más simple posible para que funcione correctamente. El diseño se hace sobre la marcha, haciendo un mini-diseño para la primera mini-versión y luego modificándolo en las siguientes mini-versiones. Además, no hay que hacer una documentación para el diseño, no hay mejor documentación que el mismo código. El código, por tanto, también se modifica continuamente de mini-versión en mini-versión, añadiéndole funcionalidad y extrayendo sus partes comunes.
Después de analizar las características fundamentales de la metodología XP se aprecia como desventaja esencial que la misma no permite que una vez terminado el producto, exista una documentación que facilite el desarrollo de futuras versiones del sistema.

2.2.3. Desarrollo rápido de aplicaciones (RAD)

RAD permite el desarrollo ágil de software. Una de las ideas centrales de esta metodología es que el desarrollo empieza lo antes posible para que el cliente pueda revisar un prototipo que funciona y pueda indicar el camino a seguir. A partir de ahí, la aplicación se desarrolla de forma iterativa, en la que cada nueva versión incorpora nuevas funcionalidades y se desarrolla en un breve espacio de tiempo. Las consecuencias de estas metodologías para el desarrollador son numerosas. El programador no debe pensar acerca de las versiones futuras al incluir una nueva funcionalidad. Los métodos utilizados deben ser lo más sencillos y directos posibles. Estas ideas se resumen en el principio denominado KISS. [3]

La desventaja fundamental radica en que cuando se añade una nueva funcionalidad, se debe reescribir parte del código existente. Este proceso se llama refactorización y sucede a menudo durante el desarrollo de una aplicación web. El código suele moverse a otros lugares en función de su naturaleza. Los bloques de código repetidos se refactorizan en un único lugar, aplicando el principio DRY[4].Para asegurar que la aplicación sigue funcionando correctamente a pesar de los cambios constantes, se necesita una serie de pruebas unitarias que puedan ser automatizadas. Si están bien escritas, las pruebas unitarias permiten asegurar que nada ha dejado de funcionar después de haber refactorizado parte del código de la aplicación.

2.2.4. Selección de la Metodología a utilizar

Después de haber realizado un estudio sobre las principales ventajas, desventajas y características de las metodologías más difundidas internacionalmente, se decide escoger la metodología RUP como instrumento para el desarrollo y control de la propuesta central de este trabajo por las razones siguientes:

- Implementa buenas prácticas para el desarrollo de software.
- Permite reducir los riesgos y hace que el proyecto en cuestión se torne más predecible.
- Posibilita la realización de pruebas continuas e iterativas que promueven una mejor evaluación del estado del proyecto.
- Permite a los patrocinadores recibir evidencia concreta del avance de los proyectos.
- Es un proceso guiado por requerimientos, iterativo e incremental y centrado en la arquitectura.

[3] Del inglés "Keep It Simple, Stupid" o lo que es lo mismo ¡Haz las cosas sencillas, tonto!

[4] Del inglés "Don't Repeat Yourself" o lo que es lo mismo ¡No te repitas!

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

- Se posee un mayor dominio de la misma.

2.3. Herramienta CASE de Desarrollo de Software

La tecnología CASE[5] supone la automatización del desarrollo del software contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información y se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

Sin lugar a dudas las herramientas CASE han venido a revolucionar la forma de automatizar los aspectos clave en el desarrollo de los sistemas de información, debido a la gran plataforma de seguridad que ofrecen a los sistemas que las usan y es que éstas, brindan toda una gama de componentes que incluyen todas o la mayoría de los requisitos necesarios para el desarrollo de los sistemas, han sido creadas con una gran exactitud en torno a las necesidades de los desarrolladores de sistemas para la automatización de procesos incluyendo el análisis, diseño e implementación.

Desde que se crearon éstas herramientas (1984) hasta la actualidad, las mismas cuentan con una credibilidad y exactitud de un reconocimiento universal, siendo usadas por cualquier desarrollador y / o programador que busca un resultado óptimo y eficiente, pero sobre todo que busca esa minuciosidad necesaria de los procesos y entre los procesos [10].

Una vez presentada la finalidad de las herramientas CASE así como sus características fundamentales se hace necesario profundizar un tanto en las dos más representativas de las que se encuentran hoy

[5] Siglas en inglés de "Ingeniería del Software Asistida por Computadoras".

disponibles con independencia de los tipos de licencias que las soportan: Visual Paradigm y Rational Rose.

2.3.1. Rational Rose

Dentro de los tres tipos de herramientas case existentes, Rational Rose pertenece a al tipo Semántico, el mismo tiene las características siguientes:

- Modelo correctamente expresado en diagramas coherentes entre sí.
- Las únicas que con propiedad pueden llamarse herramientas CASE para UML.
- Puntuaciones intermedias y extremas (Argo UML, Magic Draw).

De forma general Rational Rose no es más que una herramienta CASE que da soporte al modelado visual con UML ofreciendo distintas perspectivas del sistema. Da soporte al Proceso Unificado de Rational (RUP),modelado de Negocio, Captura de Requisitos (parcial), Análisis y Diseño (Completo), Implementación (como ayuda), Control de Cambio y gestión de configuración). Ofrece además un diseño centrado en casos de uso, enfocado al negocio que genera un software de mayor calidad y dirigido por modelos que redundan en una mayor productividad de los desarrolladores. Hace uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación. Tiene capacidades de ingeniería inversa. Posee modelo y código que permanece sincronizado en todo el ciclo de desarrollo, tiene además disponibilidad en múltiples plataformas [11].

2.3.2. Visual Paradigm

Existen tres tipos de herramientas case, Visual Paradigm pertenece al tipo Sintáctico, el mismo tiene las características siguientes:

- Diagramas correctos pero no construyen internamente un modelo
- Los diagramas quedan desconectados, sin significado o referente común (mensajes y operaciones)
- No se puede comprobar la coherencia entre diagramas

Visual Paradigm es una de las principales compañías de herramientas CASE. Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community (las cuales son gratuitas)

El visual paradigm es una herramienta que se diseña para los usuarios que están interesados en

sistemas de software de la escala grande del edificio confiablemente con el uso del acercamiento orientado al objeto. La herramienta apoya los estándares más recientes de las notaciones de Java y de UML, además de ser una herramienta multiplataforma.

De forma general, Visual Paradigm no es más que:

Una herramienta Case para UML fácil de usar que soporta la última notación UML 2.1, ingeniería inversa, generación de código, importación desde Rational Rose, exportación/importación XMI, generador de informes, editor de figuras, integración con MS Visio, plug-in, integración IDE con Visual Studio, IntelliJ IDEA, Eclipse, NetBeans y otros. Entre sus nuevas características se incluyen el modelado colaborativo con CVS y Subversión, interoperabilidad con modelos UML2 a través de XMI [12].

2.3.3. Selección de la herramienta CASE a utilizar.

Después de haber analizado las características de ambas herramientas CASE de modelado y haciendo un balance de las mismas, se considera que la más adecuada y efectiva para utilizar en el desarrollo de este sistema informático es Visual Paradigm, ya que ofrece navegación intuitiva entre el modelo visual y el código, además de permitir la sincronización entre el código fuente y el modelo en tiempo real o bajo demanda. Brinda soporte para toda la notación UML, sofisticados y automáticos diagramas de capas y el análisis de textos además es importante destacar que es una herramienta multiplataforma.

2.4. El sistema operativo a usar

El software propietario era una clara limitante para la gran comunidad de informáticos existentes en el mundo, no poder innovar o usar por la carencia de recursos un determinado software era uno de los problemas palpable de la realidad pasada, hoy en día existen muchas comunidades que promueven el código libre, de ahí la importancia de analizar que sistema operativo escogeremos para la realización de nuestro trabajo que responda siempre a las necesidades de nuestro usuario. Como parte de nuestra investigación analizaremos a Linux que es uno de los mas usado en nuestra universidad y en el mundo, específicamente dos de sus distribuciones mas conocidas: Debian y Ubuntu.

2.4.1. Debian

Debian o Proyecto Debian es una comunidad conformada por desarrolladores y usuarios, que pretende crear y mantener un sistema operativo GNU basado en software libre precompilado y empaquetado, en un formato sencillo en múltiples arquitecturas de computador y en varios núcleos.

Debian nace como una apuesta por separar en sus versiones el software libre del software no libre. El modelo de desarrollo del proyecto es ajeno a motivos empresariales o comerciales, siendo llevado adelante por los propios usuarios, aunque cuenta con el apoyo de varias empresas en forma de infraestructuras. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

2.4.2. Ubuntu

Ubuntu es una distribución GNU/Linux de tipo escritorio, basada en Debian siendo a su vez una de las distribuciones más respetadas, tecnológicamente avanzadas y mejor soportadas. Ubuntu pretende crear una distribución que proporcione un sistema GNU/Linux actualizado y coherente para la informática de escritorio y servidores, incluye además una cuidadosa selección de los paquetes de Debian y mantiene su poderoso sistema de gestión de paquetes que permite instalar y desinstalar programas de una forma fácil y limpia. A diferencia de la mayoría de las distribuciones, que vienen con una enorme cantidad de software que puede o no ser de utilidad, la lista de paquetes de Ubuntu se ha reducido para incluir solo aplicaciones importantes y de alta calidad. El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar procesos latentes por omisión al momento de instalarse. Por lo que no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema. Para labores/tareas administrativas incluye una herramienta llamada sudo, con la que se evita el uso del usuario root [13].

2.4.3. Selección del sistema operativo a utilizar

Después de hacer un análisis de las principales necesidades, se descartó a Windows como sistema operativo a usar teniendo en cuenta que es un SO propietario y que como tal incurre en limitaciones y costes para los clientes. Teniendo en cuenta la necesidad de un entorno robusto y funcional, adecuado tanto para uso doméstico como profesional y las múltiples facilidades que proporciona Ubuntu, las que contribuyen al uso eficiente de las tecnologías a usar para el desarrollo de la aplicación se decide por el equipo de desarrollo utilizar Ubuntu 7.10 (última versión estable) como sistema operativo a usar.

2.5. Arquitectura Modelo-Vista-Controlador

La razón fundamental por la que se escoge la arquitectura Modelo-Vista-Controlador es porque el framework que se va a utilizar, el cual se definirá y explicará en epígrafes posteriores, está basado en este patrón clásico de diseño web.

La arquitectura MVC (*Modelo-Vista-Controlador*) fue diseñada 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; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

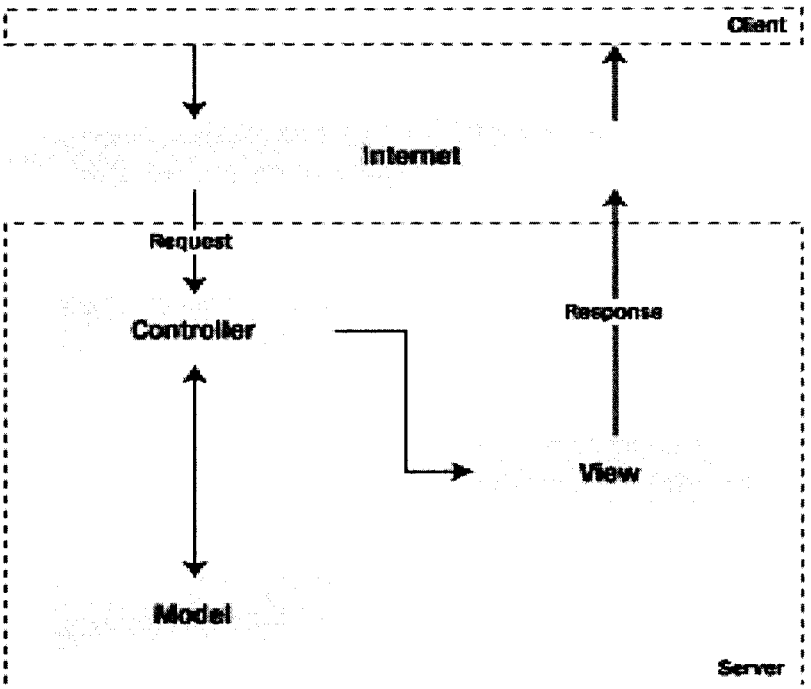


Figura # 4: El patrón MVC [14]

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlas en tiempo de ejecución. Si uno de los Componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas. Ante esta perspectiva, hacer un cambio aquí no es nada trivial.

Es importante resaltar que Symfony simplifica el MVC con funcionalidad de que trae por defecto lo que contribuye a facilitar el trabajo y elevar la calidad.

2.6. Aplicación Web como vía alternativa de solución

Actualmente con el desarrollo de las TIC y la introducción de Internet, se facilita la forma de acceso a la información desde cualquier parte del mundo pero también se marca una meta para las aplicaciones y sus desarrolladores los que a su vez tienen la obligación de hacer aplicaciones más rápidas, ligeras y robustas, que permitan ser utilizadas por cualquier persona que las necesite sin importar el desarrollo tecnológico con que cuente.

La Oficina Nacional de Recursos Minerales tiene una estructura de un solo nivel, su oficina central tiene ocho direcciones que están distribuidas por todo el país por tanto tener una aplicación web no solo facilitará el acceso a la información sino también que posibilitará el uso de la misma desde cualquier máquina con el solo uso de un navegador y como parte de las exigencias del cliente podrá ser administrada desde cualquier parte del mundo.

2.6.1. Seguridad en aplicaciones Web

En los últimos años hemos sido testigos de como los servidores web se han establecido como el primer vector de ataque para intentar infectar a víctimas vulnerables, utilizándolas como plataforma para ataques más sofisticados, de ahí que la seguridad de los sitios web es una cuestión de importancia fundamental, además de compleja, para los programadores. Garantizar la protección de un sitio requiere la elaboración cuidadosa de un plan de seguridad.

Los ataques de piratas contra las tecnologías de la información son cada vez más frecuentes y sus efectos más devastadores como consecuencia de que internet es un entorno abierto al acceso de todo tipo de usuarios y donde se deben extremar las medidas de seguridad. Por estos motivos resulta de crucial importancia poseer la capacidad de detectar y bloquear ataques en tiempo real contra los propios sistemas de información así como familiarizarse con las posibilidades de seguridad que ofrece el sistema operativo a explotar.

La mejor inversión en seguridad consiste en formar adecuadamente al personal implicado en el desarrollo de aplicaciones web. A largo plazo resulta más viable desarrollar aplicaciones seguras que intentar proteger aplicaciones inseguras por eso abordamos también en este epígrafe cuestiones fundamentales que se deberían adoptar para proteger la aplicación web a implementar como respuesta al objetivo central del presente trabajo.

2.6.2. Principios básicos de seguridad

- Validación de la entrada y salida de información: siempre debe verificarse que cualquier dato entrante o saliente es apropiado y en el formato que se espera. Las características de estos datos deben estar predefinidas y debe verificarse en todas las ocasiones.
- Diseños simples: los mecanismos de seguridad deben diseñarse para que sean los más sencillos posibles, huyendo de sofisticaciones que compliquen excesivamente la vida a los usuarios. Si los pasos necesarios para proteger de forma adecuada una función o módulo son muy complejos, la probabilidad de que estos pasos no se ejecuten de forma adecuada es muy elevada.
- Utilización y reutilización de componentes de confianza debe evitarse reinventar la rueda constantemente: por tanto, cuando exista un componente que resuelva un problema de forma correcta, lo más inteligente es utilizarlo.
- Defensa en profundidad: nunca confiar en que un componente realizará su función de forma permanente y ante cualquier situación. Hemos de disponer de los mecanismos de seguridad suficientes para que cuando un componente del sistema fallen ante un determinado evento, otros sean capaces de detectarlo.
- Tan seguros como el eslabón más débil: no debemos fiarnos únicamente de los mecanismos de seguridad "exteriores", sino que es preciso identificar cuales son los puntos precisos en los que deben establecerse las medidas de seguridad. Si nosotros no hacemos este trabajo, seguro que los atacantes si lo harán.
- La "seguridad gracias al desconocimiento" no funciona: el simple hecho de ocultar algo no impide que, a medio o largo plazo, llegue a ser descubierto. Tampoco es ninguna garantía de que tampoco será descubierto a corto plazo.

- Verificación de privilegios: los sistemas deben diseñarse para que funcionen con los menos privilegios posibles. Igualmente, es importante que los procesos únicamente dispongan de los privilegios necesarios para desarrollar su función, de forma que queden compartimentados.
- Ofrecer la mínima información: Ante una situación de error o una validación negativa, los mecanismos de seguridad deben diseñarse para que faciliten la mínima información posible. De la misma forma, estos mecanismos deben estar diseñados para que una vez denegada una operación, cualquier operación posterior sea igualmente denegada.

Luego de analizar cada uno de los principios expuestos se presentan a continuación una serie de recomendaciones asociadas al diseño de cualquier aplicación web.

2.6.3. Recomendaciones generales de seguridad para la aplicación Web a implementar.

- Ejecutar aplicaciones con privilegios mínimos.
- Conocer a los usuarios.
- Crear mensajes de error seguros.
- Usar cookies de forma segura.
- Realice copias de seguridad con asiduidad y guárdelas en lugar seguro.
- Mantenga el equipo del servidor en un lugar físico seguro, de forma que los usuarios no autorizados no puedan tener acceso a él, apagarlo, llevárselo.
- Cierre los puertos que no se utilicen y desactive los servicios no usados.
- Ejecute un programa antivirus que supervise el tráfico.
- Establezca y haga respetar una política que prohíba a los usuarios tener sus contraseñas escritas en una ubicación fácil de localizar.
- Use un servidor de seguridad (firewall).
- Manténgase informado sobre las últimas revisiones de seguridad del sistema operativo con que cuenta.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

2.7. Sistema Gestor de Base de Datos (SGBD)

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.
Los Sistemas Gestores de Bases de Datos: proporcionan una interfaz entre aplicaciones y sistema operativo, consiguiendo entre otras cosas, que el acceso a los datos se realice de una forma más eficiente, más fácil de implementar y, sobre todo, más segura.

Existen distintos objetivos que deben cumplir los SGBD, entre ellos pueden enumerarse:

1. Abstracción de la información.
2. Redundancia mínima.
3. Consistencia.
4. Seguridad.
5. Integridad.
6. Respaldo y recuperación.
7. Control de la concurrencia.
8. Tiempo de respuesta.

El objetivo de este epígrafe será introducir las características de los tres sistemas de gestión de bases de datos, mas usados en nuestra Universidad (MySQL, PostgreSQL, SQL Server) haciendo una pequeña comparación entre ellos, y concluir con la elección de aquel más adecuado para nuestra situación.

2.7.1. MySQL

MySQL es un sistema de gestión de bases de datos relacional, bajo la licencia de GNU/GPL. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar

este gestor en un software propietario, ya que de no ser así, se infringiría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.

1. Soporta gran cantidad de tipos de datos para las columnas.
2. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
3. Gran portabilidad entre sistemas.
4. Soporta hasta 32 índices por tabla.
5. Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

2.7.2. Microsoft 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. Así como un grupo de bondades que a continuación se listan:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accedan a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos y gratuito pero orientado a proyectos más pequeños. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase ASE.

2.7.3. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. El director de este proyecto es el profesor Michael Stonebraker. PostgreSQL es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99, así como otras características que comentaremos más adelante.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

A continuación se enumeran las principales características de este gestor de bases de datos:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de arreglos.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.

- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

2.7.4. Selección del Sistema Gestor de Base de Datos a utilizar

Después de analizar las principales características de los tres sistemas gestores de base de datos más difundidos se decide que sea PostgreSQL la herramienta a utilizar porque dentro de sus principales ventajas está: ser multiplataforma y contar con características de la orientación a objetos, como son: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

2.8. Fundamentación del lenguaje de programación a utilizar (PHP)

PHP es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios web con los cuales se puede programar las páginas HTML y los códigos de fuente. PHP es un acrónimo recursivo que significa "**P**HP **H**ypertext **P**re-processor" (inicialmente PHP Tools, o, *P*ersonal *H*ome *P*age Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando las librerías Qt o GTK+.

2.8.1. Características principales del lenguaje de programación PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Posee una amplia documentación de ayuda para un mayor entendimiento del lenguaje.
- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.

La versión más reciente de PHP es la 5.2.5 (8 de noviembre de 2007), que incluye todas las ventajas que provee el nuevo Zend Engine 2, entre ellas podemos citar:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario, con PHP Data Objects.
- Mejoras de rendimiento.

- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML (XPath, DOM, etc.).
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Manejo de excepciones.

2.9 Fundamentación del framework a utilizar.

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 de ahí la importancia de escoger, estudiar y seleccionar el adecuado para resolver el problema existente en el negocio. Como parte de esta investigación, teniendo en cuenta su uso en la UCI y su respaldo a nivel internacional se decidió encaminar el estudio hacia tres de ellos: CakePHP, Symfony y Zend.

2.9.1. CakePHP

CakePHP es un framework de desarrollo rápido de aplicaciones de código abierto en PHP. Posee una infraestructura que tiene como finalidad permitir el desarrollo de aplicaciones web de manera ágil y estructurada, sin perder flexibilidad [15].

2.9.2. 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. Zend Framework, a diferencia de los otros entornos de desarrollo presentados, es un intento dirigido por la compañía responsable del desarrollo del lenguaje PHP y mantenido por una comunidad de voluntarios. A pesar de ello hasta la fecha aún no consigue niveles de eficacia y adopción similares a los frameworks en PHP ya existentes [16].

2.9.3. Symfony

Symfony es un framework diseñado para optimizar el desarrollo de aplicaciones web a través de diversas características clave. Separa las reglas de negocio de la aplicación, la lógica del servidor y las vistas de presentación. Contiene una gran variedad de herramientas y clases para conseguir acortar el tiempo de desarrollo de aplicaciones web complejas. Adicionalmente, automatiza tareas comunes para que el programador pueda enfocarse por completo en las especificaciones.

Symfony cumple los requisitos siguientes:

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de 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 solo 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.

Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software [17].

2.9.4. Selección del framework a utilizar:

Características	Cake PHP	Symfony	Zend FW
Arquitectura de aplicaciones			
Incorporación del patrón Modelo Vista Controlador orientado a objetos.	X	X	X
Mapeado de objetos a bases de datos relacionales (ORM).	X	X	
Implementación de código HTML			
Uso de plantillas en PHP.	X	X	X
Seguridad			
Verificación de la salida generada en HTML por procesamiento de peticiones (Data Sanitization).	X	X	
Usabilidad y acceso rápido			
Almacenamiento en caché de las vistas.	X	X	
Almacenamiento en caché de configuración de las aplicaciones.		X	
Documentación para su uso			
Manual de referencia.	X	X	X
Herramientas de programación			
Generación de código PHP.	X	X	
Herramientas de prueba y depuración.		X	X
Interfaz de línea de comandos para la creación y mantenimiento de aplicaciones.	X	X	
Almacenamiento de logs de funcionamiento del framework.		X	
Extensibilidad y opciones adicionales			
Integración con otras herramientas a través de plugins.	X	X	
Soporte para envío de correo electrónico.		X	X
Generación de archivos PDF.		X	X
Soporte para internacionalización y localización de contenidos.		X	X
Soporte PHP			
Soporte para PHP5.	X	X	X
Características adicionales			
Comunidad activa de usuarios.	X	X	X

Después de analizar la comparación establecida entre los framework estudiados en este epígrafe se decide que por ser Symfony el framework mas completo, diseñado para optimizar el desarrollo de las aplicaciones web y automatizar las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación, por la extensibilidad, por las opciones adicionales y las ventajas que este proporciona.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

2.10. Conclusiones del capítulo

Después de realizar un estudio de factibilidad de las tecnológicas más usadas no solo en Cuba sino también a nivel internacional se escogió un conjunto o KIT de herramientas y metodologías que no solo son las más óptimas sino también las que garantizarán la calidad y eficiencia de la aplicación informática a desarrollar.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Capítulo 3

Descripción de la Solución Propuesta

En el presente capítulo se aborda el tema correspondiente al estado actual del negocio, con el objetivo de entender el contexto en que se ubica el sistema, se define en un modelo de dominio los principales conceptos que nos ayuden a comprenderlo, además se especifican los Requisitos Funcionales y Requisitos No Funcionales que debe tener el sistema, los que posibilitarán proponer una solución asociada a las necesidades y facilidades de uso para el cliente, se pretende además identificar las relaciones y secuencias de acciones existentes entre los actores que interactúan con el sistema mediante un Diagrama de Caso de Uso del Sistema.

3.1. Estado actual del Negocio

Para describir los procesos del negocio que se relacionan con el campo de acción de este trabajo, es necesario centrar la atención en el proceso de almacenamiento, gestión y control de los nomencladores, en la ONRM.

Actualmente en la aplicación existente se puede adicionar información referente a las listas código, los elementos y los nomencladores pero se introducen errores con facilidad por la carencia de un sistema que permita establecer normas a la hora de interactuar con la aplicación. Por otra parte resalta, como una de las principales dificultades, la imposibilidad de ofrecer un servicio de búsqueda para otras aplicaciones, un servicio en línea y fuera de línea que brinde las últimas actualizaciones y que permita no solo informar al administrador en caso de detectar un error sino proponer una solución. La solución existente además impide tener un control sobre las acciones diarias de la ONRM referentes a los nomencladores. Además es imposible la integración de este sistemas con otros lo cual dificulta el trabajo debido a inmensa cantidad de oficinas que están distribuidas por todo el país.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

3.2. Modelo del dominio

Teniendo en cuenta el bajo nivel de estructuración que presenta el negocio que se está estudiando, se propone un modelo del dominio, ya que de manera visual permite mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

Para identificar todos los conceptos que se emplearán en el diagrama se utilizará un glosario de términos sobre los nombres:

- Se le denominará **administrador** a todo aquel usuario que tenga permiso para acceder a todas las opciones que brinda el sistema.
- Se le denomina **aplicación** al sistema actual que permite realizar todos los cambios solicitados por los usuarios a través del administrador a la información contenida en la base de datos referente a las listas código, elementos, nomencladores o sinónimos.
- Se le denominará **solicitudes de búsqueda** a todas aquellas solicitudes que se realicen por un usuario al acceder al sistema.
- Se le denominará **usuario** a cualquier persona que solicite el servicio de realizar búsquedas en el sistema o consultar los reportes predefinidos del sistema.
- Se le denominará **log de fallas** al reporte que se genera a partir buscar información y no encontrarla.
- Se le denominará **reporte de búsqueda** a toda la información que se genera a partir de una solicitud al sistema.
- Se le denominará **controlador de búsqueda** a la funcionalidad encargada de realizar búsquedas y devolver como resultado un "reporte de búsqueda" de la información contenida en la base de datos.

- Se ele denominara **información** a todo lo que está almacenado en la base de datos, referente a las listas código, elementos, nomencladores o sinónimos.
- Se le denominará **log de cambios** al reporte que se genera a partir de que alguna información contenida en la base de datos sea modificada en cualquiera de sus extensiones.
- Se le denominará **controlador de información** a la funcionalidad encargada de gestionar la información referente a las listas código, elementos, nomencladores o sinónimos y devolver como resultado un "log de cambios" de la información contenida en la base de datos.

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

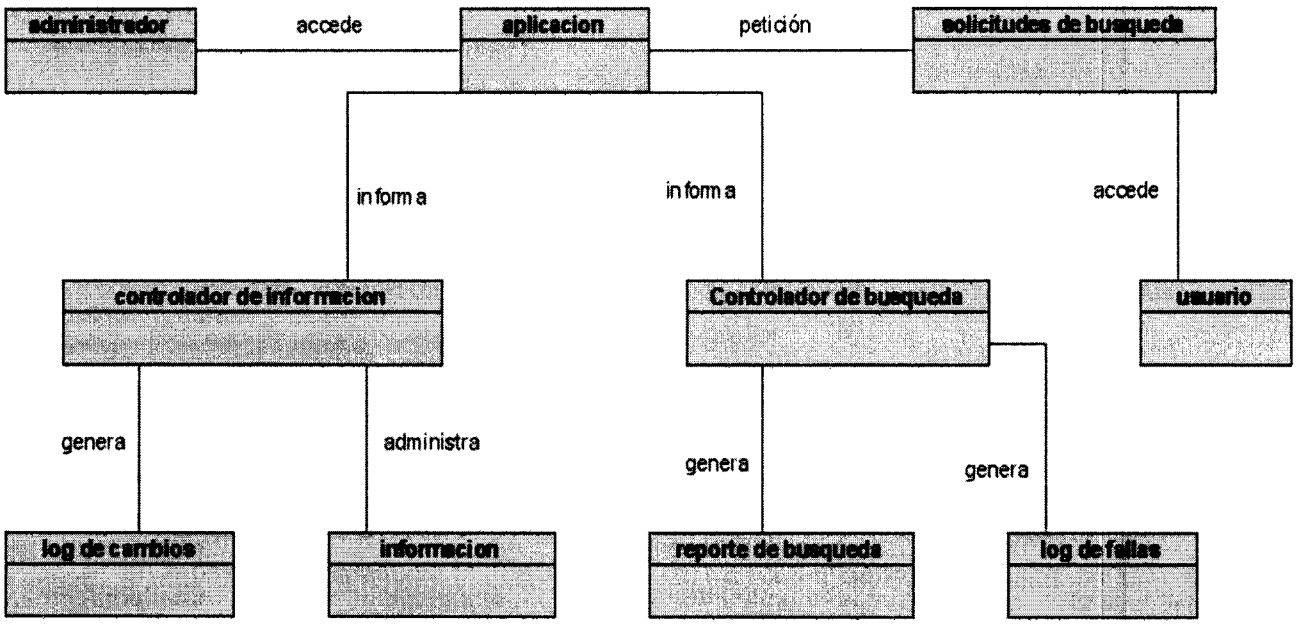


Figura # 5: Modelo de dominio.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

3.2.1. Reglas generales del Negocio

Se identificaron varias reglas que debe seguir el sistema que se desarrolle para respetar y garantizar las restricciones existentes en el negocio:

1. Si no es administrador no puede insertar o modificar ninguna información referente a las listas código, elementos o sinónimos.
2. Los cambios referentes a los nomencladores, serán previamente aprobados por un equipo calificado en el tema.
3. Los cambios que se realicen sobre la información deben almacenarse en una pequeña tabla.
4. No se puede eliminar la información contenida en la base de datos.

3.3. Requisitos del Sistema

Una vez conocidos los conceptos que rodean al objeto de estudio, se puede comenzar a analizar cuales serán las funcionalidades que debe tener el sistema para que cumpla con los objetivos planteados al inicio de este trabajo. Por lo tanto se definen Requerimientos funcionales (RF) y Requerimientos no funcionales (RNF), los primeros se basan en los casos de usos, describen de qué forma el usuario va a utilizar el sistema y permiten a los analistas proponer cómo será la interfaz del sistema (Jacobson); los segundos especifican las propiedades del sistema que tienen que ver con rendimiento, velocidad, uso de memoria, plataforma, fiabilidad, tiempo de respuesta media, defectos por miles de líneas de código es decir imponen condiciones a requisitos funcionales.

3.3.1. Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, por tal motivo a continuación se presenta un conjunto de ellos:

R.1 Gestionar información.

R.1.1 Insertar una nueva lista de código.

R.1.2 Insertar un nuevo elemento.

R.1.3 Insertar un nuevo sinónimo.

R.1.4 Modificar una lista código.

R.1.5 Modificar un elemento.

R.1.6 Modificar un sinónimo.

R.1.7 Convertir una lista código en nomenclador.

R.2 Buscar información.
R.3 Exportar
R.4 Realizar solicitud de cambio.
R.5 Realizar solicitud de inserción.
R.6 Gestionar solicitud.
  R.6.1 Visualizar solicitud de cambio.
  R.6.2 Visualizar solicitud de inserción.

3.3.2. Requerimientos no funcionales

Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

Apariencia o interfaz externa:

Diseño sencillo y fácil de usar, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

Usabilidad:

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora en sentido general.

Soporte:

El sistema debe de tener una garantía de instalación y prueba del mismo, además de un breve entrenamiento a los futuros usuarios.

Seguridad:

1. Identificar al usuario antes de que pueda realizar cualquier acción sobre la configuración del sistema.
2. Garantizar que la información sea vista únicamente por quien tiene derecho a verla.
3. Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
4. Verificación sobre acciones irreversibles (eliminaciones).
5. Garantía de que el sistema funcione correctamente aun cuando no haya conectividad.

Confiabilidad:

El sistema de gestión de bases de datos debe tener soporte para recuperación ante fallos y errores.

Requisitos de Hardware:

Servidor

1. Correrá en una configuración mínima
   1.1. Procesador Pentium III, 1,8 GHz.
   1.2. Memoria RAM 512 Mgb.
   1.3. Capacidades disco no menor a 40 GB.
2. Correrá en una configuración óptima
   2.1. Procesador Pentium IV, 3.0 GHz, Cache L2 Mb.
   2.2. Memoria RAM 2 Gb.
   2.3. Capacidades disco no menor a 120 GB.

Requisitos de Software:

El sistema debe de permitir que los requerimientos en el lado del cliente para la utilización de la aplicación solo se limiten a tener disponible un navegador Web. Para la implantación de la herramienta debe de requerirse:

1. Lenguaje de programación orientado a objeto (versión PHP 5.2.5).
2. Framework Symfony (versión 1.0.12).
3. Gestor de bases de datos PostgreSQL (versión 8.2.7).
4. Sistema Operativo Linux (versión Ubuntu 7.10.

3.4. Propuesta de Solución

Con el apoyo y las facilidades que brinda la notación UML y el Diagrama de Casos de Uso se representan las funcionalidades del futuro sistema establecidas en los requisitos funcionales. Para ello se definen los actores que van a interactuar con el sistema y las funcionales con las que se relacionan. Un Caso de Uso es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. Es valido resaltar además que un Actor no es como tal parte del sistema, sino que es un Rol de un usuario, donde este puede intercambiar información y representa a un ser humano, software o a una máquina que interactúa con el sistema. En nuestro caso interactúan 3 actores.

Descripción de los Actores del sistema a automatizar

usuario	Es un usuario simple que interactúa con la aplicación sin tener privilegios de administración y que realiza acciones que son propias del y no de ningún otro actor.
cliente	Es el usuario simple que interactúa con la aplicación sin tener privilegios de administración pero que a su vez realiza operaciones que pueden ser ejecutadas por el administrador.
administrador	Es el encargado de administrar todos los cambios que se efectúan en la aplicación y realiza además acciones propias del usuario simple.

Diagrama de Caso de Uso de Sistema

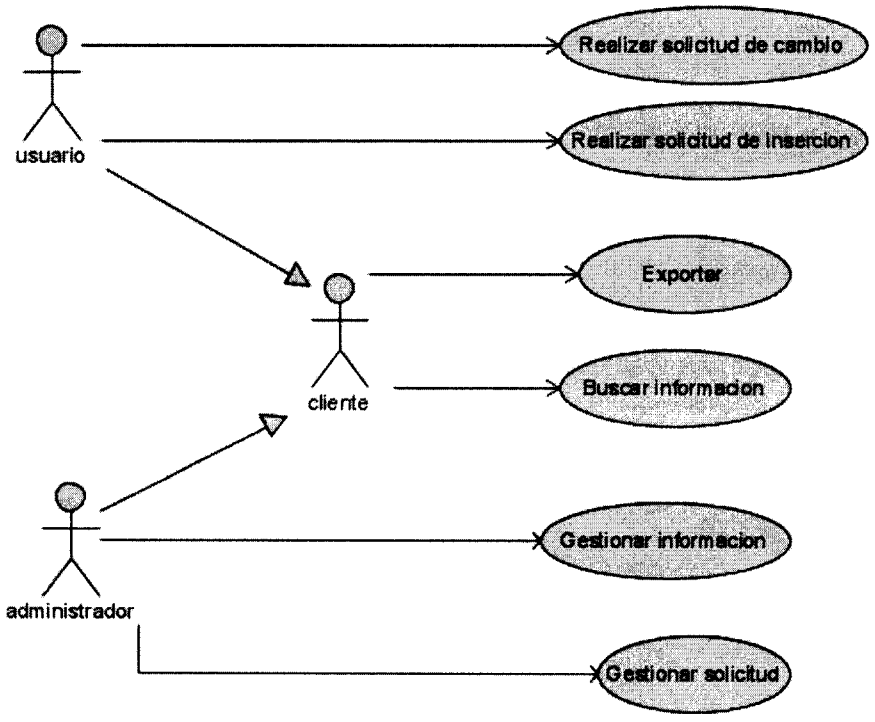


Figura # 6: Diagrama de Casos de Uso del Sistema

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

3.5 Diagrama de paquetes

Los paquetes de casos de uso son la forma de agrupar a estos últimos respondiendo a algún criterio, y se representan a través de los diagramas que evidencian gráficamente la relación entre los actores y los casos de uso. Se definieron cuatro paquetes atendiendo a la funcionalidad: Paquete de Solicitudes Paquete de Exportar, Paquete de Seguridad y Paquete de Información

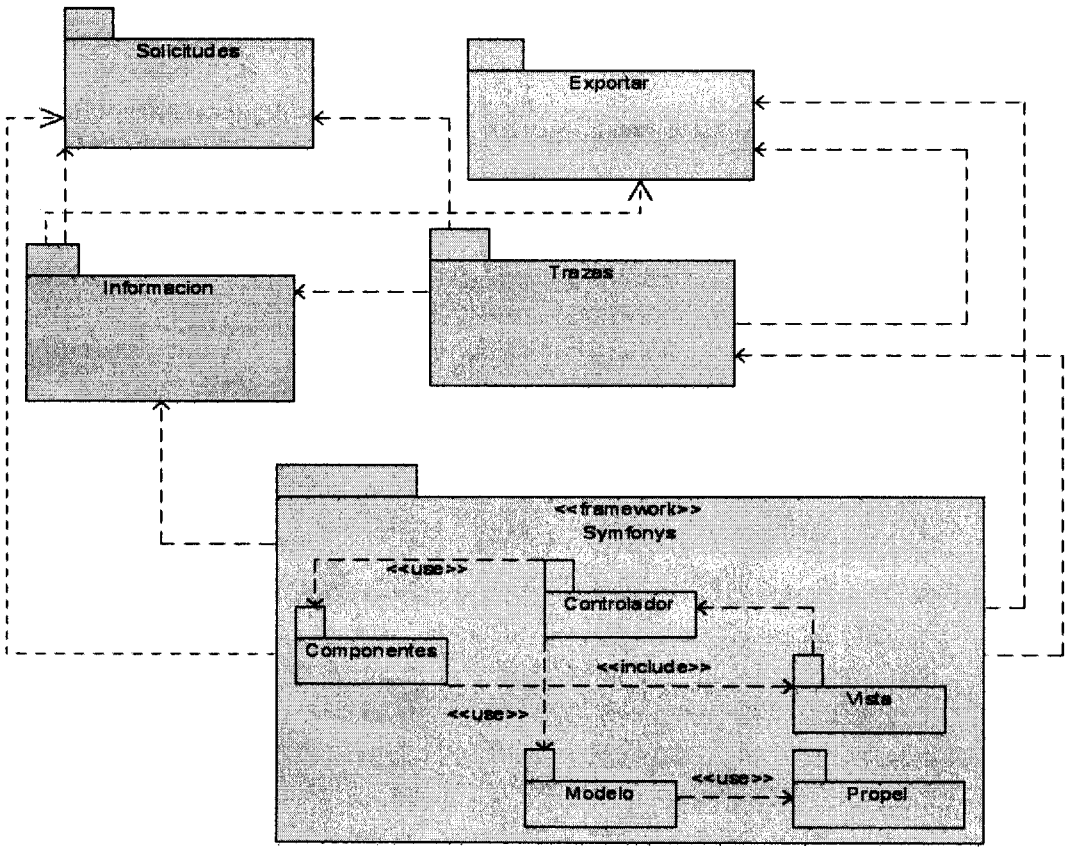


Figura # 7: Relación entre paquetes

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

3.6. Descripción de los Casos de Uso del Sistema

Realizar solicitud de Cambio.

Caso de Uso:	Realizar solicitud Cambio
Actores:	cliente
Resumen:	El cliente realiza una solicitud de cambio en caso de que la información buscada contenga error en la base de datos. Registra los datos en un formulario de solicitud de cambio, se le validan los datos y se almacena en una lista de solicitudes pendientes. Termina el caso de uso.
Precondiciones:	Que se ejecute el CU Buscar Información.
Referencias	R4
Prioridad	Secundario

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
El cliente selecciona la opción Realizar solicitud de cambio.	El sistema le muestra un menú con el tipo de cambio. - Lista de Código. - Elementos. - Sinónimos.

Prototipo de Interfaz

Solicitud de Cambio

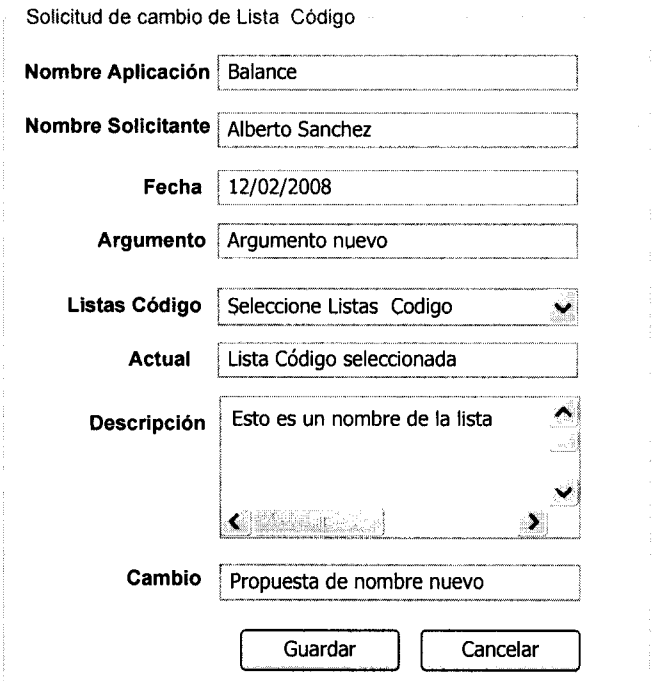
Listas Código
Elementos
Sinónimos

Flujos Normal de Eventos	
Sección "Cambiar Listas de Código"	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra un formulario de entrada de datos. - Nombre Aplicación. - Nombre Solicitante. - Fecha. - Argumento. - Listado de Listas Código Almacenado. - Nombre Actual de la Lista de Código. - Descripción de Lista de Código.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

	- Propuesta de Cambio de Nombre.
2. El actor cliente introduce los datos en formulario de solicitud.	3. El sistema verifica que los datos entrados por el cliente sean correctos. 4. El sistema Almacena la solicitud en una lista de solicitudes pendientes. Termina el CU.
Flujo Alterno	
3.1.a El sistema muestra un mensaje de alerta al cliente "Los datos introducción son incorrectos"	

Prototipo de Interfaz



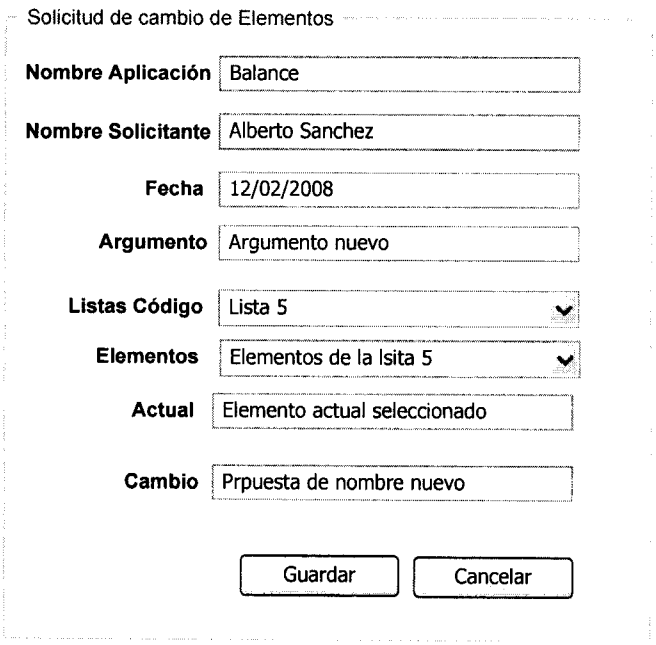
Flujos Normal de Eventos	
Sección "Cambiar Elementos"	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra un formulario de entrada de datos. - Nombre Aplicación. - Nombre Solicitante. - Fecha. - Argumento. - Listado de Listas Código Almacenado.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

	- Listado de Elementos de la lista - Nombre Actual del elemento. - Propuesta de Cambio de Nombre.
2. El actor cliente introduce los datos en formulario de solicitud.	3. El sistema verifica que los datos entrados por el cliente sean correctos. 4. El sistema Almacena la solicitud en una lista de solicitudes pendientes. Termina el CU.
Flujo Alterno	
3.1. a El sistema muestra un mensaje de alerta al cliente "Los datos introducción son incorrectos"	

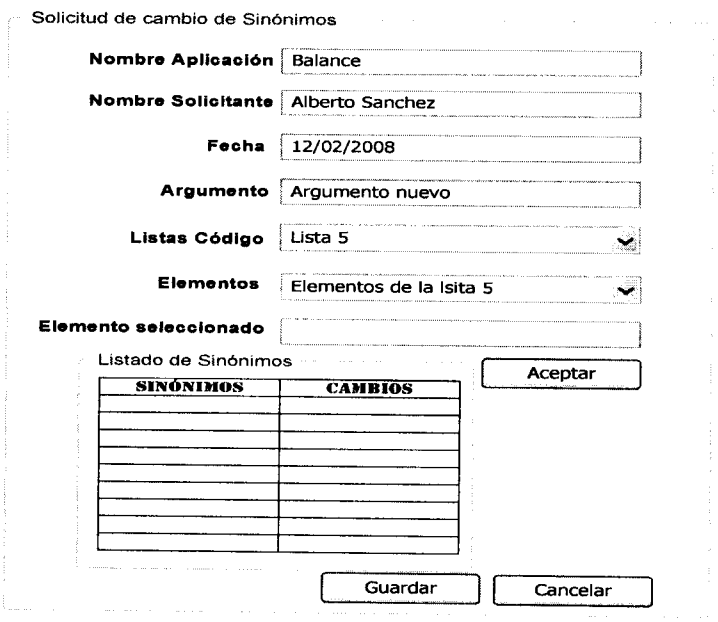
Prototipo de Interfaz



Flujos Normal de Eventos	
Sección "Cambiar Sinónimos"	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra un formulario de entrada de datos. - Nombre Aplicación. - Nombre Solicitante. - Fecha. - Argumento.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

	- Listado de Listas Código Almacenado. - Listado de Elementos de la lista. - Elemento seleccionado.
2. El actor cliente introduce los datos en el formulario de solicitud de sinónimos. 3. El actor cliente selecciona un elemento de la lista de elementos para visualizar los sinónimos. 5. El actor cliente introduce los nuevos cambios de los sinónimos que están asociado al elemento seleccionado.	4. El sistema muestra un listado de los sinónimos del elemento seleccionado. - Nombre del Sinónimo - Cambio 6. El sistema verifica que los datos introducidos por el cliente sean correctos. 7. El sistema Almacena la solicitud en una lista de solicitudes pendientes. Termina el CU.
Flujo Alterno	
6.1.a El sistema muestra un mensaje de alerta al cliente "Los datos introducción son incorrectos"	
Prototipo de Interfaz  	
Pos condiciones	Se tiene una versión actualizada del registro de trazas, que en caso de ser aprobada formara parte de la actualización del sistema.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Realizar solicitud de Inserción.

Caso de Uso:	Realizar solicitud Inserción
Actores:	cliente
Resumen:	El cliente realiza una solicitud de inserción en caso de que la información buscada no esté contenida en la base de datos. Registra los datos en un formulario de solicitud de inserción, se le validan los datos y se almacena en una lista de solicitudes pendientes. Termina el caso de uso.
Precondiciones:	Que se ejecute el CU Buscar Información.
Referencias	R5
Prioridad	Secundario

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
El cliente selecciona la opción Realizar solicitud de Inserción.	El sistema le muestra un menú con el tipo de Inserción. - Insertar Lista de Código. - Insertar Elementos. - Insertar Sinónimos.

⊟ **Solicitud de Inserción**

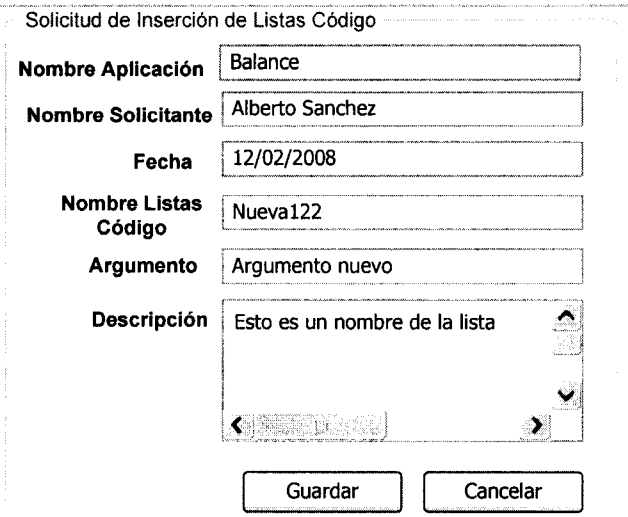
Insertar Listas Código
Insertar Elementos
Insertar Sinónimos

Flujos Normal de Eventos	
Sección "Insertar Lista de Código"	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra un formulario de entrada de datos. - Nombre Aplicación. - Nombre Solicitante. - Fecha. - Argumento. - Nombre de la nueva Lista Código - Descripción de Lista de Código.
2. El actor cliente introduce los datos en formulario	

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

de solicitud inserción. 3. El actor cliente selecciona el botón de guardar.	4. El sistema verifica que los datos entrados por el cliente sean correctos. 5. El sistema Almacena la solicitud de inserción de la lista código en una lista de solicitudes pendientes. Termina el CU.
Flujo Alterno	
4.1.a El sistema muestra un mensaje de alerta al cliente "Los datos introducidos son incorrectos"	



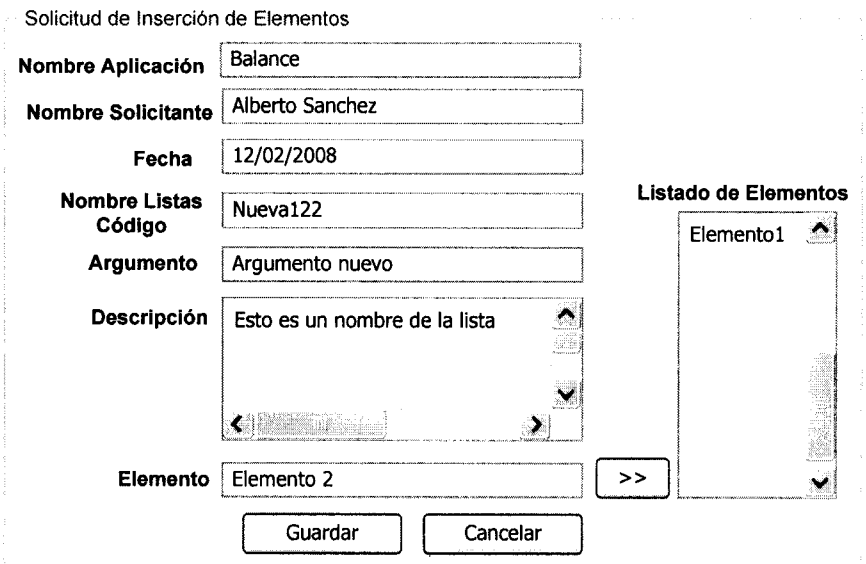
Flujos Normal de Eventos	
Sección "Insertar Elementos"	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra un formulario de entrada de datos. - Nombre Aplicación. - Nombre Solicitante. - Fecha. - Argumento. - Listado con todas las Listas Código - Listado de Elementos de la lista - Nombre Actual del elemento.
2. El actor cliente introduce los datos en formulario	

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

de solicitud.de inserción de elementos	3. El sistema verifica que los datos entrados por el cliente sean correctos. 4. El sistema Almacena la solicitud en una lista de solicitudes pendientes. Termina el CU.
Flujo Alterno	
3.1. a El sistema muestra un mensaje de alerta al cliente "Los datos introducidos son incorrectos"	

Prototipo de Interfaz



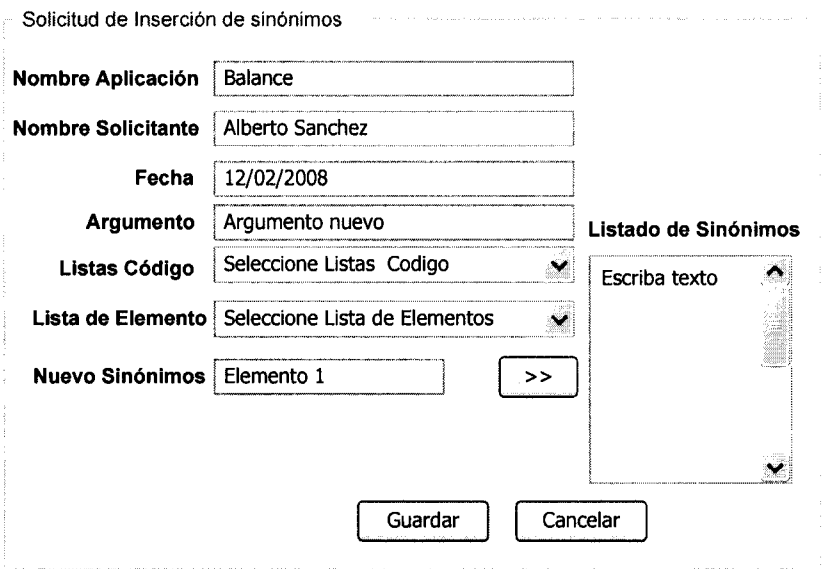
Flujos Normal de Eventos	
Sección "Insertar Sinónimos"	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra un formulario de entrada de datos. - Nombre Aplicación. - Nombre Solicitante. - Fecha. - Argumento. - Listado con todas las Listas Código - Listado de Elementos de la lista. - Nombre del nuevo sinónimo.
2. El actor cliente introduce los datos en el formulario de solicitud de sinónimos.	

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

	3. El sistema verifica que los datos entrados por el cliente sean correctos. 4. El sistema Almacena la solicitud en una lista de solicitudes pendientes. Termina el CU.
Flujo Alterno	
3.1.a El sistema muestra un mensaje de alerta al cliente "Los datos introducción son incorrectos"	

Prototipo de Interfaz



Pos condiciones	Se tiene una versión actualizada del registro de trazas, que en caso de ser aprobada formara parte de la actualización del sistema.

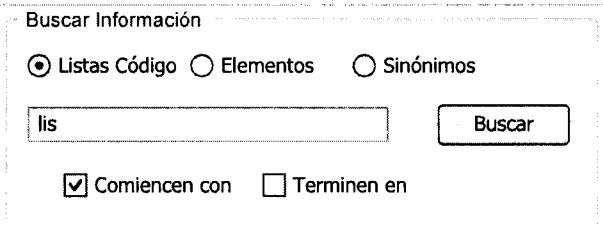
Buscar información.

Caso de Uso:	Buscar información
Actores:	usuario
Resumen:	El usuario selecciona la opción de Buscar Información, el sistema le muestra una interfaz de búsqueda múltiple, donde el usuario puede realizar la búsqueda de los elementos necesarios. Selecciona las opciones de búsqueda que brinda el sistema como, Listas Código, Elementos o Sinónimos. El sistema brinda como resultado un listado con toda la información encontrada y en caso de no haber muestra un mensaje "No existe resultado para esa búsqueda". Termina el CU.
Precondiciones:	Que se encuentre almacenada la información en el sistema.
Referencias	R2

Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción de buscar información.	2. El sistema muestra una interfaz de búsqueda con el siguiente formulario. Tipo de Búsqueda • Listas código. • Elementos • Sinónimos
3. El usuario introduce los parámetros de búsqueda necesarios, Presiona el botón buscar.	4. El sistema verifica que los parámetros de búsqueda sean los correctos. 5. El sistema muestra una interfaz con el resultado de la búsqueda. 6. Termina el CU.



Resultados de la búsqueda

No	Nombre
1	lista1
2	lista2
3	lista3
4	lista4
5	lista5
6	lista6
7	Lista7
8	lista8
9	lista9

Flujos Alternos

4.1. a El sistema muestra un mensaje "No existe resultado para esa búsqueda" y termina el CU.

Pos condiciones	Que el sistema muestre un listado con todos los elementos que coinciden con los parámetros de búsqueda.

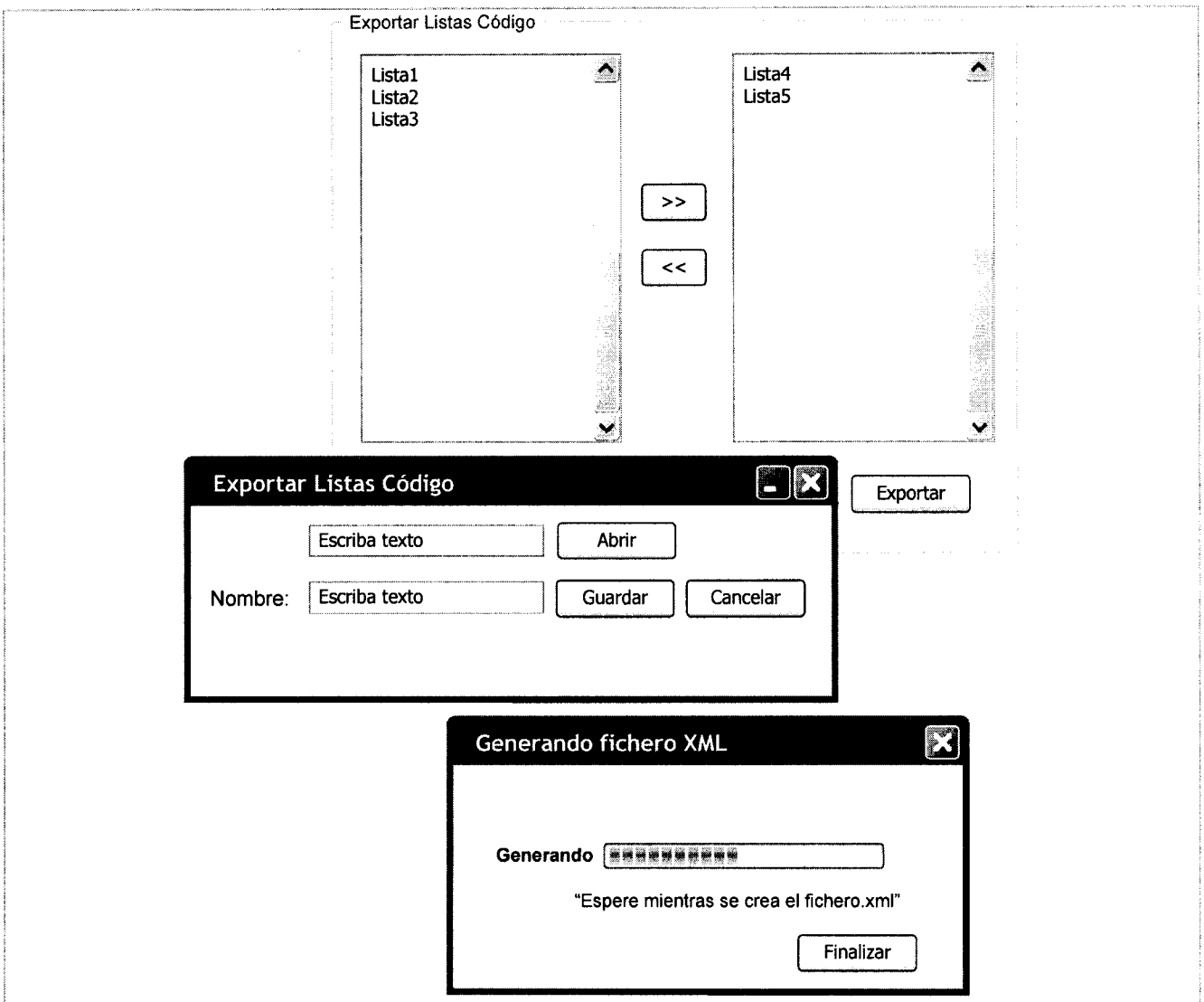
Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Exportar

Caso de Uso:	Exportar
Actores:	usuario
Resumen:	El usuario selecciona la opción de exportar listas código, el sistema brinda la posibilidad de escoger un conjunto de listas código. Selecciona el directorio donde desea almacenar el archivo y el sistema procede a crearlo.
Precondiciones:	Que este almacenadas las listas de código en el sistema.
Referencias	R3
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El usuario selecciona del menú la opción Exportar.	2. El sistema muestra una interfaz con todas las listas códigos existentes en la BD.
3. Selecciona la(s) listas código que desea exportar y presiona el botón de exportar.	4. El sistema muestra una interfaz donde el usuario selecciona el directorio donde desea guardar el archivo.
5. El usuario selecciona el directorio donde se almacenará y presiona el botón de guardar.	6. El sistema verifica la acción del botón "Guardar", carga las listas código y genera el fichero XML.
	7. El sistema muestra una interfaz donde visualiza el porciento de generación del
8. El usuario presiona el botón de finalizar la exportación	fichero, una vez que termine de generar se activa el botón de finalizar.
	9. Terminar el CU.



Flujos Alternos	
6.1.a Si se presiona el botón de cancelar el sistema aborta la exportación del fichero y termina el CU.	
Pos condiciones	El fichero XML creado y almacenado en directorio elegido por el usuario.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

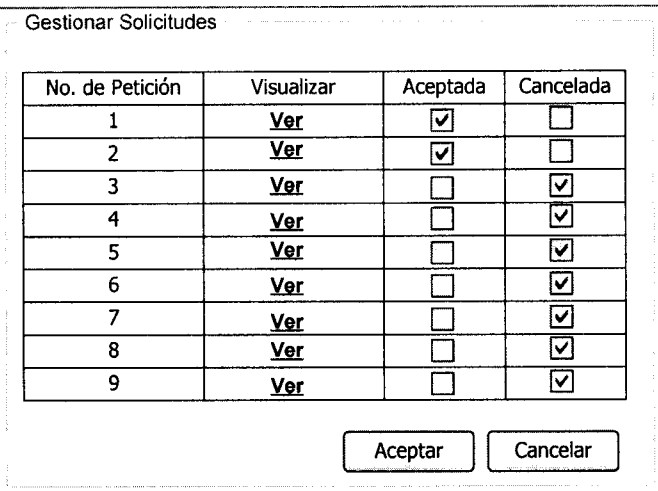
Gestionar solicitud

Caso de Uso:	Gestionar solicitud
Actores:	administrador
Resumen:	El sistema le muestra al administrador todas las solicitudes que están almacenadas en el sistema. Revisa y evalúa la(s) solicitud (es) presionando el vínculo [Ver], donde visualizará toda la información sobre la solicitud. Una vez terminada la evaluación, el administrador presiona la opción de aceptar o cancelar dicha solicitud. Si esta solicitud es aceptada se realizan los cambios correspondientes y se elimina de lista de solicitudes, en caso contrario se cancela la solicitud y se elimina de la lista de solicitudes archivas en el sistema.
Precondiciones:	Que existan solicitudes almacenadas en el sistema.
Referencias	R6
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El administrador selecciona la opción de gestionar solicitudes.	2. El sistema muestra una interfaz con los siguientes elementos. (Tabla) a. No. De petición b. Visualizar. c. Aceptada. d. Cancelada.
3. El usuario selecciona la opción de Visualizar (Ver).	4. El sistema muestra una Interfaz del CU realizar solicitud (Cambio o Inserción).
5. El administrador selecciona la opción de Aceptar o Cancelar si la solicitud fue aprobada o no. Presiona (Botón Aceptada).	6. El sistema realiza los cambios correspondientes y elimina la solicitud de la lista de solicitudes pendientes almacenadas en el sistema. Termina el CU.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



Flujos Alternos	
6.1. a Si la opción es cancelar el sistema elimina la solicitud de la lista de solicitudes pendientes almacenadas en el sistema. Termina CU	
Pos condiciones	Las solicitudes pueden ser o no aprobadas por el administrador.

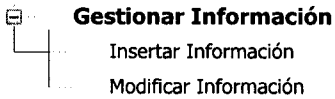
Gestionar información

Caso de Uso:	Gestionar Información
Actores:	administrador
Resumen:	El administrador realiza una acción de insertar o modificar información siempre que no esté contenida en la base de datos o que presente errores. Registra los datos en un formulario llamado Insertar información o Modificar información, ya sea de Listas Código, Elementos o Sinónimos, en dependencia de lo que desee insertar o modificar, se le validan los datos y se almacena en la base de datos. Termina el caso de uso.
Precondiciones:	Que se ejecute el CU Buscar Información.
Referencias	R1
Prioridad	Crítico

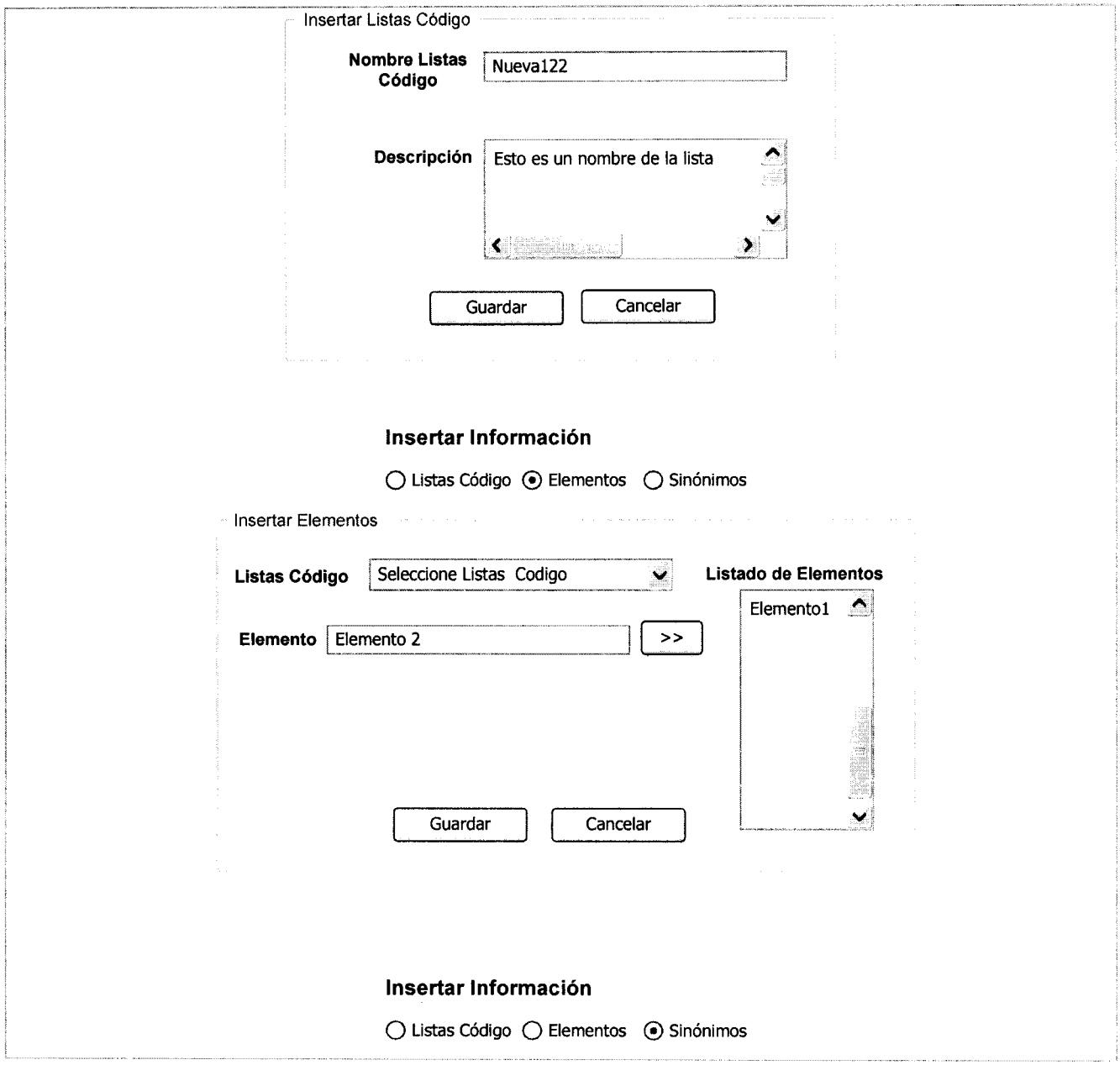
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1- El administrador selecciona la opción gestionar Información. 2- El administrador selecciona la opción de: Ir a la sección [Insertar Información]	2- El sistema le muestra un menú con los tipos de operaciones que puede realizar. - Insertar Información - Modificar Información

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

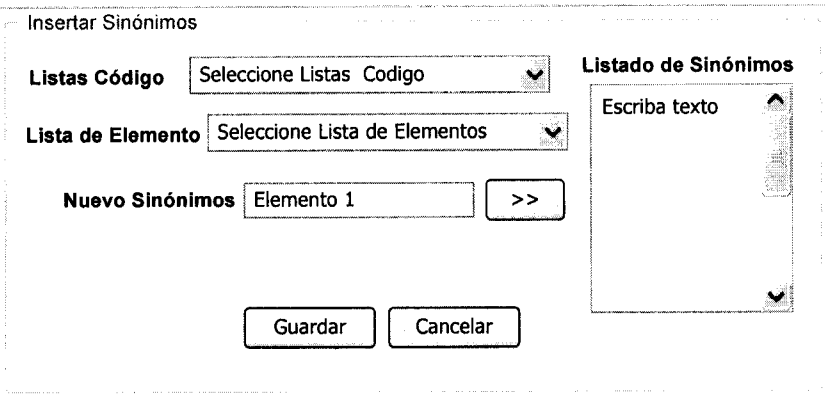
Ir a la sección [Modificar Información]	
Prototipo de Interfaz  **Gestionar Información** Insertar Información Modificar Información	
Flujos Normal de Eventos **Sección "Insertar Información"**	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra una interfaz donde el administrador puede seleccionar la opción de inserción que desee.
2. El administrador selecciona la opción de: • Listas Código • Elementos • Sinónimos	3. El sistema verifica la opción seleccionada y muestra un formulario de inserción.
4. El administrador introduce los datos correspondientes y presiona el botón de "Guardar"	5. El sistema verifica la información introducida por el administrador. 6. El sistema procesa la información y la almacena en la base de datos. 7. Termina el CU.
Flujos Alternos	
5.1.a En caso de que la información contenga errores, el sistema muestra un mensaje al administrador donde cometió el error. "La información presenta errores"	
Prototipo de Interfaz **Insertar Información** ◉ Listas Código ○ Elementos ○ Sinónimos	

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz



Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



Flujos Normal de Eventos

Sección "Modificar Información"

Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una interfaz donde el administrador puede seleccionar la opción de modificar la información que desee.
2. El administrador selecciona la opción de: a. Listas Código b. Elementos c. Sinónimos	3. El sistema verifica la opción seleccionada y muestra listado con todos los elementos. a. Nombre. b. Apellidos c. Opción [Modificar]

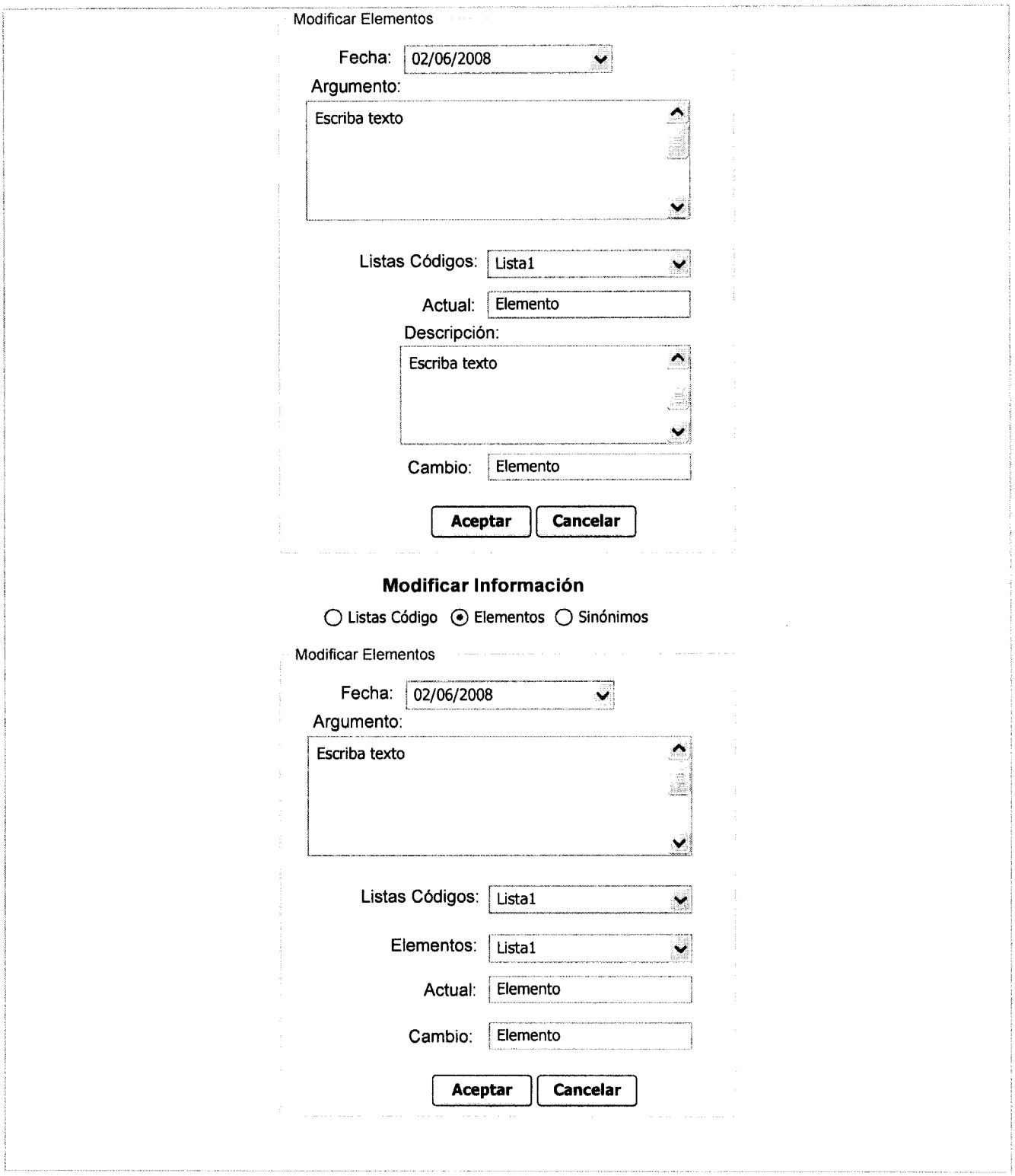
Prototipo de Interfaz

Modificar Información

◉ Listas Código ○ Elementos ○ Sinónimos

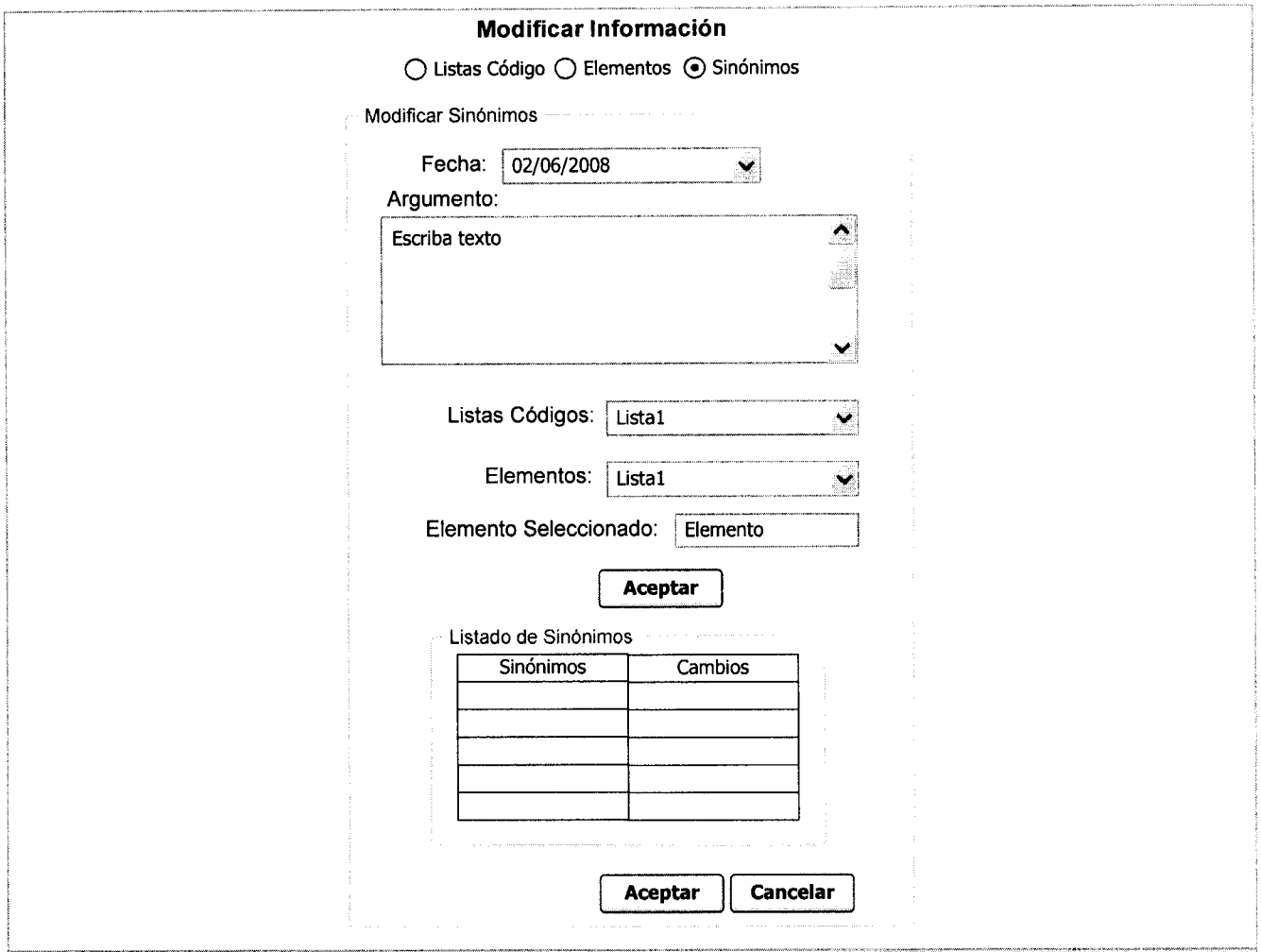
Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



Flujos Alternos	
Pos condiciones	La solicitud de modificar o insertar una información puede ser o no aprobada.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

3.7. Conclusiones del capítulo

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio y del levantamiento de requisitos funcionales y no funcionales los cuales definen las propiedades que debe cumplir el sistema y las funciones que este debe de realizar, representándolas mediante un Diagrama de Casos de Uso del Sistema. Además a partir de aquí se tiene una visión más acertada para el desarrollo del siguiente flujo de trabajo que nos propone RUP, Análisis y Diseño.

Capítulo 4

Análisis y Diseño de la Solución Propuesta

En este capítulo se propone modelar y analizar los principales artefactos que ayudan a la construcción de una aplicación web. Se definen además los diagramas de clases del análisis así como los diagramas de interacción que permitirán analizar los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos, como parte del diseño se modela el sistema y se analiza: cómo el sistema satisface los requerimientos funcionales, requerimientos de calidad y las restricciones, es decir, en esta parte del proceso de desarrollo del software se decide como se va a llevar a cabo el mismo. Se obtiene además el diagrama de clases persistentes y su modelo de datos, para construir la base de datos que almacenara los datos de la aplicación, se aborda brevemente los patrones de diseño y arquitectura que se utilizaron. Al cierre del mismo se propone el modelo de despliegue donde se representan los nodos en los que se distribuye la aplicación y el de componentes para una mejor descripción de la solución propuesta

4.1. Diagrama de Clases del Análisis

4.1.1. Modelo de Análisis

Un modelo de análisis ofrece una especificación mas precisa de los requisitos que la que se obtiene como resultado de la captura de requisitos, contribuye a razonar sobre los aspectos internos del sistema, nos proporciona una estructura centrada en el mantenimiento y en aspectos tales como la flexibilidad ante los cambios y la reutilización. Se describe además utilizando el lenguaje de los desarrolladores, y puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre funcionamientos internos del sistema. Puede considerarse además como una primera aproximación al modelo de diseño y es por tanto una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación [8].

4.1.2. Clases del Análisis

Una clase del análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema y dentro de las principales características se puede resaltar que se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales, siempre encaja en uno de tres estereotipos básicos: de interfaz, de control o de entidad [8] .Ver Anexo

4.2. Diagrama de Interacción

El diagrama de interacción, representa la forma en como un Cliente (Actor) u Objetos (Clases) se comunican entre si en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: diagramas de colaboración y diagramas de secuencia [18].

4.2.1. Diagrama de Colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos. A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia [19].

4.2.2. Diagrama de Secuencia

En los diagramas de secuencia se muestra la interacción entre objetos mediante transferencias de mensajes entre objetos o subsistemas. Además forma parte del modelado dinámico del sistema. Se modelan las llamadas entre clases desde un punto concreto del sistema. Es útil para observar la vida de los objetos en sistema, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de llamadas entre los componentes del sistema [20].

4.3. Diagrama de Clases de Diseño

4.3.1. Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones tienen impacto en el sistema. Sirve también de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación [8]. Ver Anexo

4.3.2 Patrones de diseño

Los patrones son parejas de problema/solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Asignar correctamente las responsabilidades es muy importante en el diseño orientado a objetos [6].

En el diseño de la aplicación se tuvieron en cuenta los patrones GRASP[6] los que se explican a continuación:

Creador: En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

Experto: Este es uno de los más utilizados, puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

Alta Cohesión: Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

Controlador: Todas las peticiones Web son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador

[6] General Responsibility Asignment Software Patterns

frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Bajo Acoplamiento: La clase Actions hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

Además de los ya mencionado Symfony implementa también patrones GOF. Se utilizará el framework Symfony para el desarrollo del sistema informático, este framework utiliza una serie de patrones GOF como son:

Patrones Creacionales:

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el controlador frontal hay una llamada a sfContext::getInstance(). En una acción, el método getContext(), un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony

Abstract Factory (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

Patrones Estructurales:

Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de diseño llamado:

Composite (Objeto compuesto): Permite tratar objetos compuestos como si de uno simple se tratase. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol .Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera [21].

4.3.3 Patrón Modelo Vista Controlador

La arquitectura MVC divide una aplicación interactiva en tres áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

Modelo (Model): modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

Vista (View): Transforma el modelo en una página web que permite al usuario interactuar con ella.

Controlador (Controller): se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista [14].

Se han desarrollado a lo largo de los años, desde la presentación de este patrón a la comunidad científica tres variantes fundamentales, que se presentan brevemente a continuación.

Variante I: Variante en la cual no existe ninguna comunicación entre el Modelo y la Vista y esta última recibe los datos a mostrar a través del Controlador.

Variante II: Variante en la cual se desarrolla una comunicación entre el Modelo y la Vista, donde esta última al mostrar los datos los busca directamente en el Modelo, dada una indicación del Controlador, disminuyendo el conjunto de responsabilidades de este último.

Variante III: Variante en la cual se diversifica las funcionalidades del Modelo teniendo en cuanta las características de las aplicaciones multimedia, donde tienen un gran peso las medias utilizadas en estas.

Para la realización de la arquitectura del software se empleó la Variante I del patrón MVC lo que se provoca un mantenimiento más sencillo de las aplicaciones.

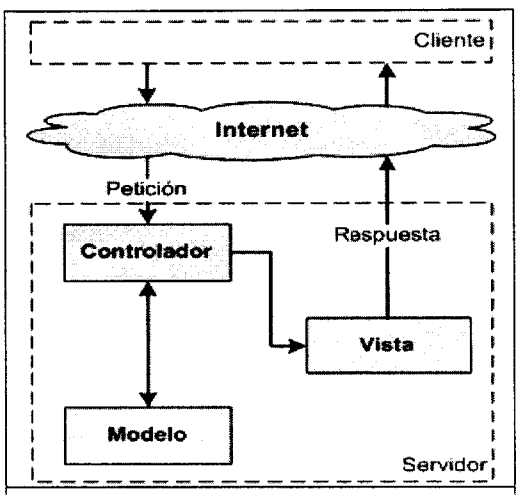


Figura # 8: Variante I [14]

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

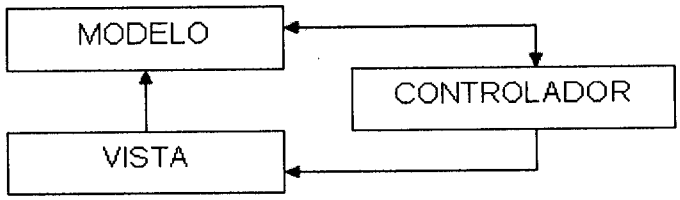


Figura # 9: Variante II

4.3.4. Diagrama de Clases Persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo por lo que se ha elaborado el siguiente diagrama de clases persistentes, para poder obtener el Modelo de Datos de la aplicación. A continuación se muestra el diagrama de clases persistentes del sistema, donde aparecen todas las entidades que se manejan en él [22].

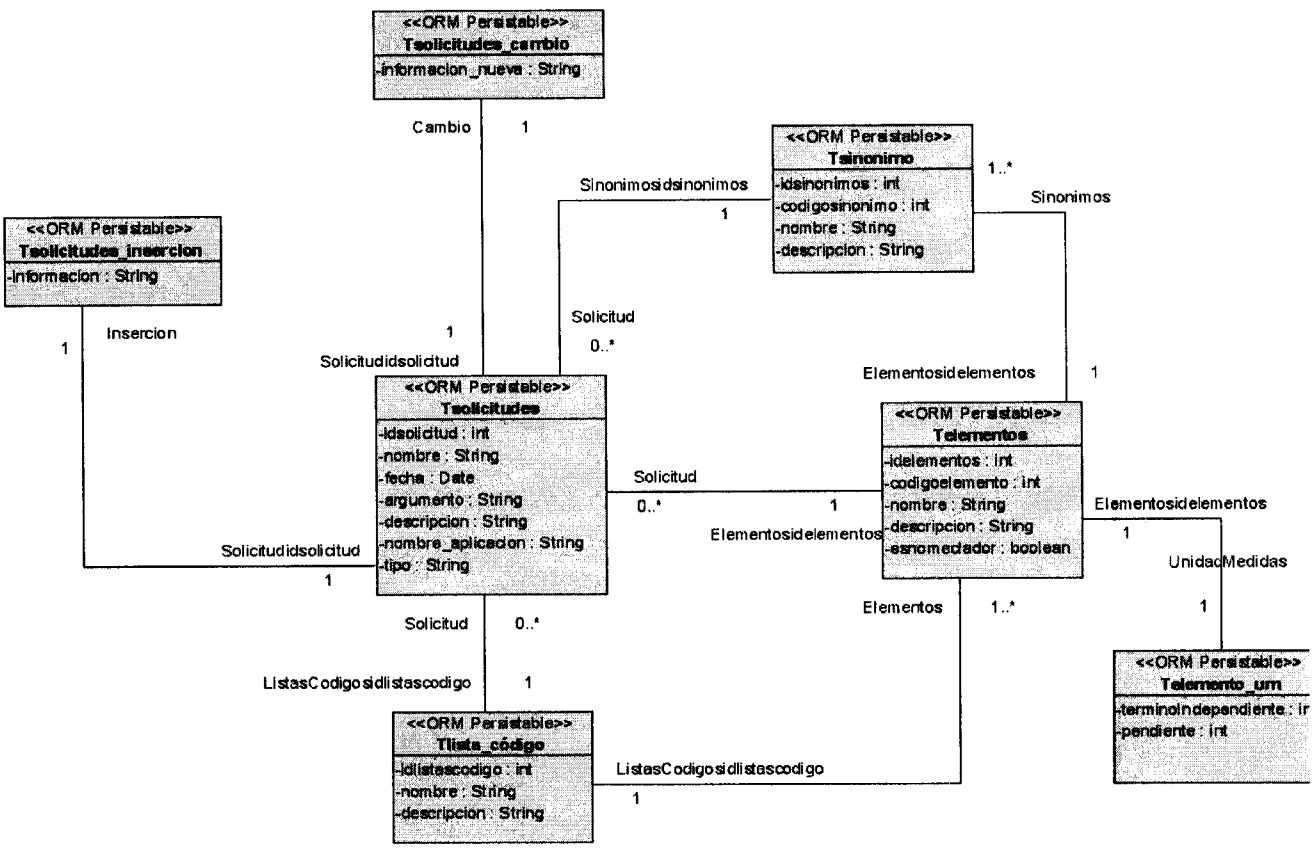


Figura # 10: Diagrama de Clases Persistentes

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

4.4. Diseño de Base de Datos

Para el diseño de la base de datos del sistema se utilizó el diagrama de clases persistentes y el modelo de datos, representando las clases que simbolizan los datos que se obtienen y almacenan durante los procesos de la aplicación, siendo estos los que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará el modelamiento de la lógica del negocio.

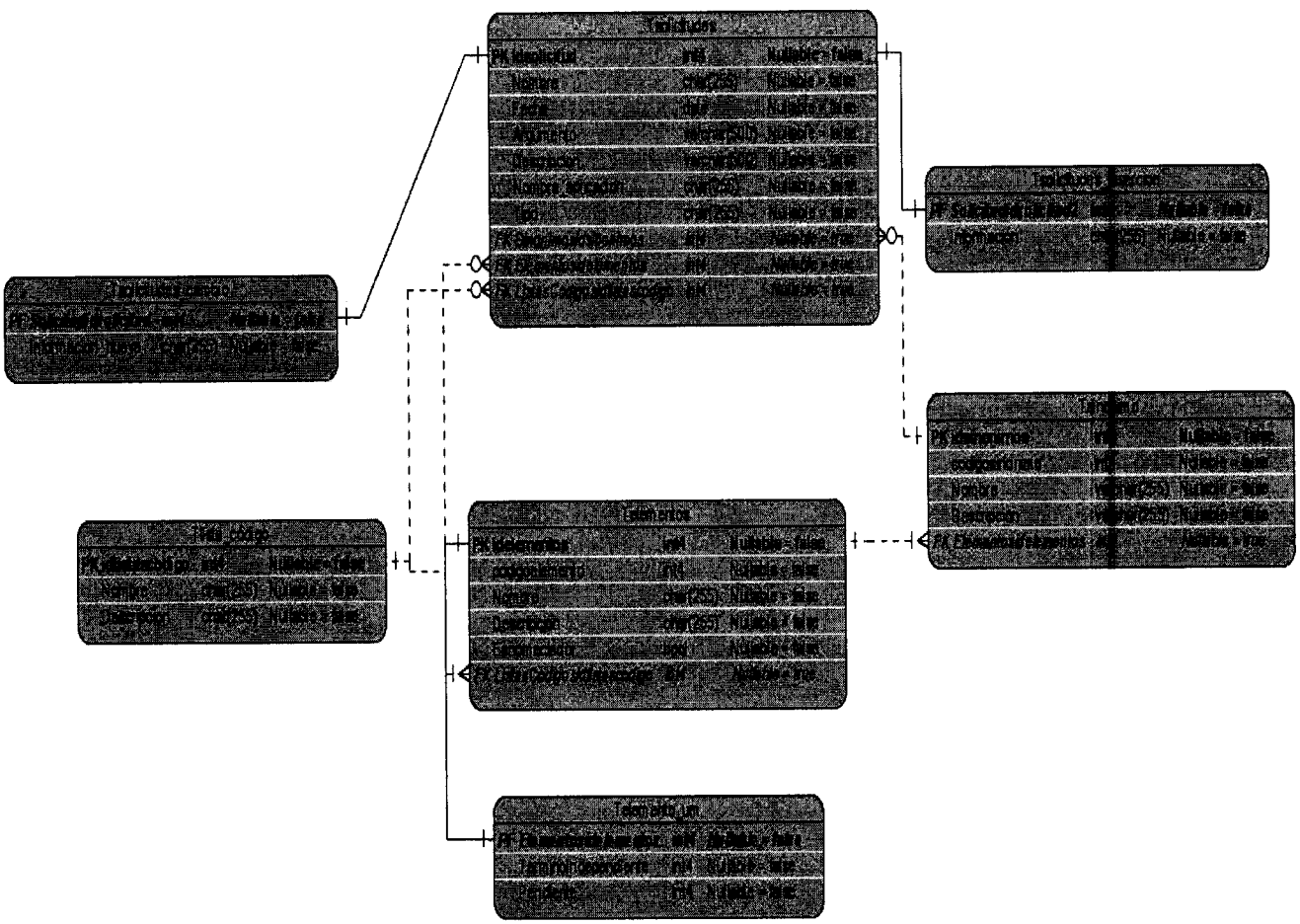


Figura # 11: Modelo de datos (Modelo físico)

Para mejor comprensión de las tablas, Ver anexo

4.4.1. Diagrama de Clase (Extensión Web)

Cuando se habla de modelar aplicaciones Web con UML se puede apreciar que difiere un poco del resto de las aplicaciones que estamos acostumbrados a construir, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación.
Modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente sí es muy importante pues estos son los artefactos que se necesitan modelar para que el desarrollador los implemente luego y obtener así un producto final.

4.4.2. Principios de Diseño

El diseño tiene que estar encaminando al usuario, usuario que en la mayoría de los casos no presenta una buena preparación en la rama de la informática. Por eso del diseño depende que la información contenida en el sistema sea útil y que los servicios se puedan usar con la calidad requerida, es decir el diseño no es solo apariencia, combinación de colores o logos acertados, es la fórmula para que el usuario se sienta atraído y la forma en que disfruta la usabilidad de la aplicación. Por ellos se establecen varios principios de diseño que de forma general garantizan la usabilidad en los diseños para aplicaciones Web.

1) Requerir de un mínimo transcurso de instrucción, permitiendo su uso desde el primer momento, por cualquier persona que posea un inapreciable dominio de la computación.
2) Permitir que la privacidad, garantía y seguridad estén igualmente disponibles para todos los usuarios, y que el diseño sea atractivo para todos los usuarios.
3) Garantizar la legibilidad, el color de los textos debe contrastar con el del fondo, y el tamaño de fuente debe ser suficientemente grande.
4) Evitar elementos invisibles de navegación que han de ser inferidos por los usuarios, menús desplegables, indicaciones ocultas, entre otras.
5) Requerir de los usuarios un mínimo esfuerzo para alcanzar sus objetivos.
6) Evitar las caídas inesperadas de la aplicación y los enlaces rotos.
7) Limitar el número de acciones que puede realizar el usuario sobre la aplicación, mostrando sugerencias (opciones) para cada posible acción, evitando así al máximo los errores de usuario.
8) Mostrar al usuario solamente aquellas opciones a las que, dado su rol en el negocio, tiene derecho a acceder.

9) Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse de que es correcta la opción seleccionada.
10) Mostrar la mayor cantidad de información acerca de las opciones brindadas en un momento dado, de modo que el usuario siempre sepa cuáles son las operaciones a las que puede acceder y en qué consiste exactamente cada una.

4.5. Estándares en la interfaz de la aplicación

Diseñar una interfaz amigable para el usuario es de vital importancia para lograr que en la interacción con el sistema el usuario se sienta identificado con el mismo, es válido resaltar además que la calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso. Para el diseño de la misma se tuvieron en cuenta aspectos necesarios que garantizaran la comodidad para el usuario, la facilidad del framework utilizado permitió la organización de la información que se muestra y su distribución en la pantalla, así como la colocación de los elementos que se repiten ubicándolos en el mismo lugar adaptando al usuario a acostumbrarse al entorno de trabajo, solamente se cuenta en la interfaz con elementos necesarios que no sobrecarguen el entorno. La aplicación contiene imágenes representativas de la entidad y los colores responden a las peticiones del usuario. Para garantizar todo lo descrito anteriormente se utilizo en todas las pagina el esquema Cabecera-Navegador-Contenido. La cabecera contiene el nombre de la aplicación en el centro superior y muestra una imagen relaciona con el medio para el que se realiza el sistema. En el navegador se incluyen los enlaces a las distintas secciones. En el área del contenido se muestran los formularios de entrada, los reportes, etc.

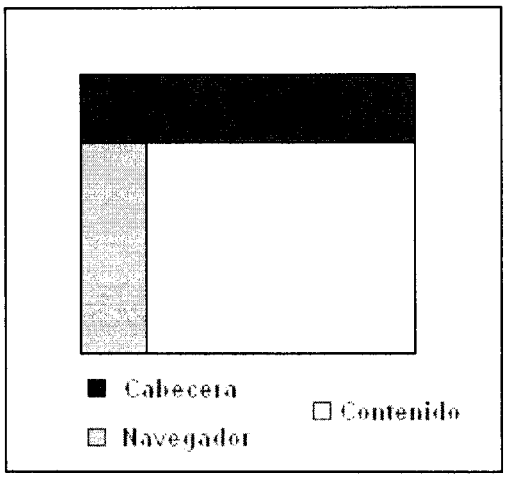


Figura # 12: Esquema de página

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

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

4.6. Tratamiento de excepciones

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias [23]. Debido a esto el tratamiento de excepciones posibilita el buen funcionamiento del sistema, validando todas las entradas de datos del cliente a nivel de interfaz. Cuando se produce un error por la entrada incorrecta de un valor suministrado por el usuario se le señaliza en la pantalla donde se encuentra para que sea rectificado.

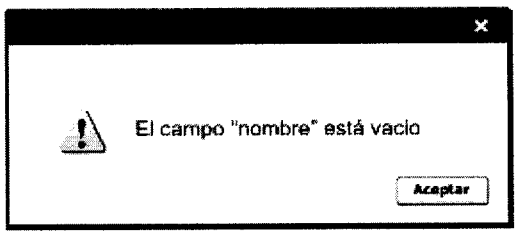


Figura # 13: Excepción del sistema

Una de las ventajas que ofrece para el programador el uso de Symfony es que cuando se produce una excepción se muestra un mensaje de error muy útil que contiene toda la información necesaria para descubrir la causa del problema. Los mensajes que produce la excepción están escritos de forma clara y hacen referencia a la causa más probable del problema. Normalmente ofrecen posibles soluciones para arreglar el error y para la mayoría de problemas comunes, incluso se muestra un enlace a la página del sitio web de Symfony que contiene más información sobre la excepción. La página con el mensaje de la excepción muestra en qué parte del código PHP se ha producido el error y la lista completa de los métodos que se han invocado [14].

4.7. Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes *software* que representan manifestaciones del código en tiempo de

ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes) [24].

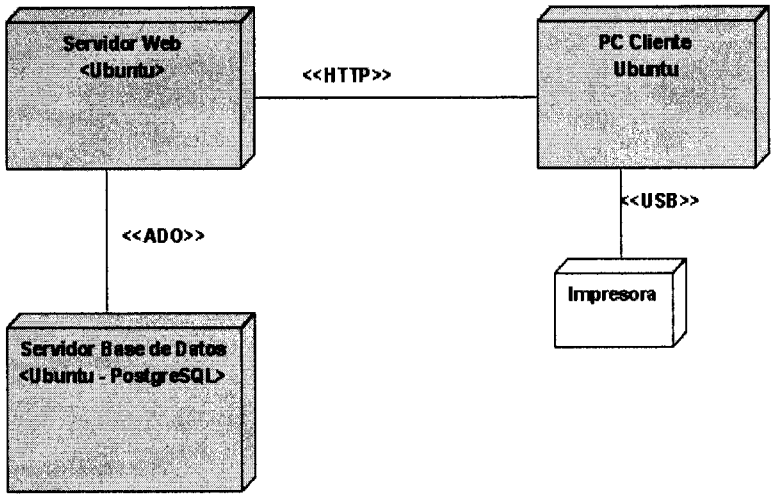


Figura # 14: Diagrama de Despliegue.

4.7.1 Descripción de los componentes del Diagrama de Despliegue.

Componentes	Descripción
DB Server	Servidor donde se encuentra ubicada la base de dato de donde nuestros servicios se nutren de información.
PC Cliente	Ordenador desde donde los usuarios o las aplicaciones consumen los servicios.
Server	Servidor Web que contiene las aplicaciones web y los servicios web.
HTTP	Protocolo utilizado para la conexión entre las PC Cliente y el servidor de aplicaciones y servicios web.
ADO	Librería utilizada para establecer la conexión entre el servidor de aplicaciones y servicios web y el servidor de base de dato.

Tabla 8: Descripción de los componentes del Diagrama de Despliegue.

4.8. Conclusiones del capítulo

1. Se obtiene un diseño sólido de la propuesta de aplicación que se formuló para resolver los problemas existentes con el manejo de la información referente a listas código, elementos y sinónimos existentes en la ONRM.
2. Se definen los nuevos reportes que posibilitarán un uso eficiente de la aplicación y buen control de la información.
3. Con la propuesta de diseño que se brinda en este capitulo se obtendrá un acercamiento exacto al modelo de implementación.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Conclusiones Generales

Es la obra fecunda cuando no se claudica en el empeño de lograr las metas que se trazan al comienzo, cuando existe constancia en el propósito y firmeza en las ideas, cuando hay aspiraciones, cuando quedan esperanzas, cuando existen sueños. La realización de éste trabajo, fruto de la labor mancomunada de sus autores y tutores, satisfizo en su totalidad los objetivos trazados, pudiéndose destacar de manera general las conclusiones siguientes:

1. Se desarrolló una propuesta de análisis y diseño de la aplicación "Sistema para la gestión y estandarización de nomencladores en la ONRM" que permite sentar las bases para la futura implementación del sistema.

2. Se realizó un estudio completo del flujo de datos que se genera a partir del uso de los sistemas informáticos hoy existentes en la ONRM.

3. Se cumplimentó un estudio detallado de las tecnologías más convenientes y adaptables a la solución propuesta.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Recomendaciones

A partir de los resultados alcanzados y los beneficios perspectivos del presente trabajo se recomienda:

1. Se lleve a cabo la implementación de la aplicación propuesta.
2. Que el documento sea puesto a disposición del grupo de desarrollo para su continua revisión, enriquecimiento y adaptación durante la fase de implementación de la aplicación.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Referencias Bibliográficas

1. Geociencias *Desarrollan en Cuba proyecto de Informatización de Geología.*
2. *Geodato IC. Interfaz de Captación. Documentación Técnica.*
3. Castejón Garrido, J.S. *Arquitectura y diseño de sistemas web modernos.* 2004 [cited; Available from: http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_disenyo_de_sistemas_web_modernos.html.
4. Ignacio Cabanes, J. *Diccionario Básico de Informática.* [cited; Available from: http://usuarios.lycos.es/Resve/diccioninform.htm.
5. Larman, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 1999 [cited 2007; Available from: http://bibliodoc.uci.cu/pdf/reg00061.pdf.
6. Larman, C., *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 1999.
7. Pressman, R.S., *Ingeniería de Software. Un enfoque práctico. .* 2005.
8. James Rumbaugh, I.J., Grady Booch., *El lenguaje Unificado de Modelado.Manual de Referencia.* 1998.
9. *Programación Extrema(XP).* 2008 [cited; Available from: http://www.chuidiang.com/ood/metodologia/extrema.php.
10. *Herramientas Case.* [cited; Available from: http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm.
11. *Rational Rose.* [cited; Available from: www.rational.com.
12. *Visual Paradigm.* [cited; Available from: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_14720_p/.
13. *Ubuntu.* [cited; Available from: http://ubuntu.com.es/.
14. *Symfony, Guía Definitiva.*
15. *CakePHP* 2008 2008 [cited; Available from: http://www.cakephp.org/.
16. *Zend Framework.* 2008 [cited; Available from: http://framework.zend.com/.
17. *Symfony.* 2008 [cited; Available from: http://www.symfony-project.org/
18. *Diagrama de Interacción* [cited; Available from: http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html
19. *Diagrama de Colaboración* [cited; Available from: http://www.clikear.com/manuales/uml/diagramasinteraccion.aspx

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

20. *Diagrama de Secuencia*. [cited; Available from: http://www.programacion.net/tutorial/uml/8/
21. *Patrones de diseño*. [cited; Available from: http://groups.google.com/group/symfony-users/msg/cd94d2ddb2057355
22. *Clases Persistentes* [cited; Available from: http://www.cic.ipn.mx/revistas/pages/vol07-04/art1.pdf
23. *Tratamiento de excepciones* [cited; Available from: http://www.usuarios.lycos.es/manualesjava/manuales/excepciones/excepciones.pdf
24. *Diagrama de Despliegue*. [cited; Available from: http://www-gris.det.uvigo.es/~avilas/UML/node50.html.

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Glosario de Términos

Apache: Proyecto nacido para crear un servidor Web estable, fiable y veloz para plataformas Unix.

API: Application Programming Interface (Interfaz de Programación)

BD: Base de Datos.

C++: Es un lenguaje híbrido, que se puede compilar y resulta más sencillo de aprender para los programadores que ya conocen C. Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Las principales características son abstracción (encapsulación), el soporte para programación orientada a objetos (polimorfismo) y el soporte de plantillas o programación genérica (templates). Es un lenguaje que abarca tres paradigmas de la programación: La programación estructurada, la programación genérica y la programación orientada a objetos.

CASE: Ingeniería del software asistida por computadora.

COCOMO: Modelo Constructivo de Coste, permite realizar estimaciones y planificaciones de proyectos de sistemas de información.

CU: Caso de Uso.

CUS: Caso de Uso del Sistema.

DDL: Lenguaje de creación de datos.

Debian: Asociación o comunidad conformada por desarrolladores y usuarios que pretenden crear y mantener un Sistema operative GNU basado en software libre precompilado y empaquetado en un formato sencillo en múltiples arquitecturas y en varios núcleos.

Distribución: Es un conjunto de aplicaciones reunidas que permiten brindar mejoras para instalar fácilmente un sistema Linux.

DML: Lenguaje de manipulación de datos.

Eclipse: Caso especifico de un IDE.

Firewall: Cortafuegos.

GNU/LINUX: Es un sistema operativo, es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Es multitarea, multiusuario, multiplataforma y multiprocesador.

GNU: Conjunto de programas desarrollados por miembros de la Fundación por el Software Libre, son de uso gratuito (FSF- Free Software Foundation).

GPL: Es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre (General Public License).

HTML: Es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el estándar de las páginas Web. (Hyper Text Markup).

HTTP: Protocolo para transferir archivos o documentos hipertexto a través de la red. (Hyper Text Transmition Protocol).

IDE: Entorno de desarrollo integrado.

Java: Lenguaje de programación orientado a objetos con el que se puede realizar cualquier tipo de programa, es un lenguaje muy extendido, es un lenguaje independiente de la plataforma, es compilado en un bytecode que es interpretado desarrollado por la compañía Sun Microsystems a principios de los 90.

MINBAS: Ministerio de la Industria Básica.

MS Visio: Herramienta del paquete de Office que permite construir interfaces.

MVC: Modelo-Vista-Controlador

MySQL: Sistema de administración de base de datos, es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

NetBeans: Caso especifico de un IDE.

Office: Es la suite ofimática por Microsoft y la más usada en la actualidad. Funciona bajo los sistemas operativos Microsoft Windows y Apple Mac OS.

ONRM: Oficina Nacional de Recursos Minerales.

OOP: Programación Orientada a Objetos.

PHP: Es acrónimo de Hipertext Pre-processor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación (Personal Home Page).

PINCG: Proceso de Informatización Nacional del Conocimiento Geológico.

Plug-in: Funcionalidades que permiten la actualización y mejoras de un software.

PostgreSQL: Servidor de Base de Datos relacional libre, liberado bajo la licencia BSD, es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

RAD: Desarrollo rápido de aplicaciones.

Rational Rose: Herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto.

RF: Requisitos Funcionales.

RNF: Requisitos no Funcionales.

RUP: Proceso Unificado de Desarrollo.

SGBD: Sistema Gestor de Base de Datos

SGBD: Sistema Gestor de Bases de Datos.

SQL: Lenguaje declarativo de acceso a bases de datos relacionales que permiten especificar diversos tipos de operaciones sobre las mismas (Structured Quero Language).

Sybase ASE: Gestor de base de datos.

TICs: Tecnologías de la Información y las Comunicaciones

UCI: Universidad de las Ciencias Informáticas

UML: Lenguaje Unificado de Modelado.

Unix: Sistema operativo portable, flexible, potente, con entorno programable, multiusuario y multitarea, muy difundido.

Web: se aplica este término para identificar una serie de aplicaciones y páginas de Internet que utilizan la inteligencia colectiva para proporcionar servicios interactivos en red dando al usuario el control de sus datos.

www: Es el universo de información accesible a través de Internet, una fuente inagotable del conocimiento humano. Es un sistema de información global, interactiva, dinámica, distribuida, gráfica, basada en hipertexto, con plataforma de enlaces cruzados, que se ejecutan en Internet (Word Wide Web).

XML: Extensible Markup Language (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.

XP: Programación Externa.

Anexos

Descripciones del modelo de datos.

Nombre: **Tlista_código**		
Descripción: En esta tabla se almacenan los datos de cada una de las listas código que están definida hasta la actualidad en la rama de la geología.		
Atributo	**Tipo**	**Descripción**
idlistascodigo	int	Identificador de las listas código
nombre	String	Nombre de la lista código
descripción	String	Breve descripción de cada una de las listas código.

Tabla: Tlista_código

Nombre: Telemento		
Descripción: En esta tabla se almacenan los datos de cada uno de los elemento por los que están compuestas las listas código que están definida hasta la actualidad en la rama de la geología.		
Atributo	**Tipo**	**Descripción**
codigoelemento	int	Identificador que permite establecer un formato para identificar la lista código a la que pertenece el elemento.
idelementos	int	Identificador de los elementos de las listas código.
nombre	String	Nombre de los elementos por los que esta compuesto las listas código.
descripcion	String	Breve descripción de cada uno de los elementos de las listas código.
esnomeclador	boolean	Información de si un elemento es un nomenclador o no.

Tabla: Telemento

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz

Nombre: Telemento_um		
Descripción: En esta tabla se almacenan los datos de cada uno de los elemento que poseen unidad de medida por los que están compuestas las listas código que están definida hasta la actualidad en la rama de la geología.		
Atributo	**Tipo**	**Descripción**
terminoindependiente (n)	int	Valor fijo en la base de datos que permite hacer la conversión de una unidad de medida a otra. Utilizando para ello la ecuación y = m x + n siendo y el valor a obtener.
pendiente (m)	int	Valor fijo en la base de datos que permite hacer la conversión de una unidad de medida a otra. Utilizando para ello la ecuación y = m x + n, siendo y el valor a obtener.

Tabla: Telemento_um

Nombre: Tsinonimo		
Descripción: En esta tabla se almacenan todos los sinónimos de los elementos que tiene cada lista código		
Atributo	**Tipo**	**Descripción**
idsinonimos	int	Identificador de los sinónimos.
codigosinonimo	int	Identificador que nos permite establecer un formato para identificar la lista código y el elemento al que pertenece el sinónimo.
nombre	String	Nombre del elemento al cual pertenece el sinónimo.
descripcion	String	Breve descripción del elemento al que le corresponde el sinónimo.

Tabla: Tsinonimo

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Nombre: Tsolicitudes		
Descripción: En esta tabla se almacenan los datos de cada una de las solicitudes que se realizan ya sea por otra aplicación o por un usuario.		
Atributo	**Tipo**	**Descripción**
idsolicitudes	int	Identificador de las solicitudes,
nombre_aplicacion	String	Nombre de la aplicación que solicita una inserción o una modificación.
argumento	String	Breve explicación que justifica el por qué de la solicitud.
nombre	String	Nombre de la persona que realiza la solicitud.
fecha	Date	Fecha en que solicita una inserción o una modificación.
tipo	String	Identificador para saber el tipo de solicitud realizada (Puede ser de cambio o de inserción).
descripcion	String	Breve descripción de cada uno de los elementos de las listas código.

Tabla: Tsolicitudes

Nombre: Tsolicitudes_cambio		
Descripción: En esta tabla se almacenan los datos de cada una de las solicitudes que se realizan ya sea por otra aplicación o por un usuario.		
Atributo	**Tipo**	**Descripción**
informacion	String	Este campo contiene la información referente al cambio que se va a guardar en la base de datos a solicitud del usuario.

Tabla: Tsolicitudes_cambio

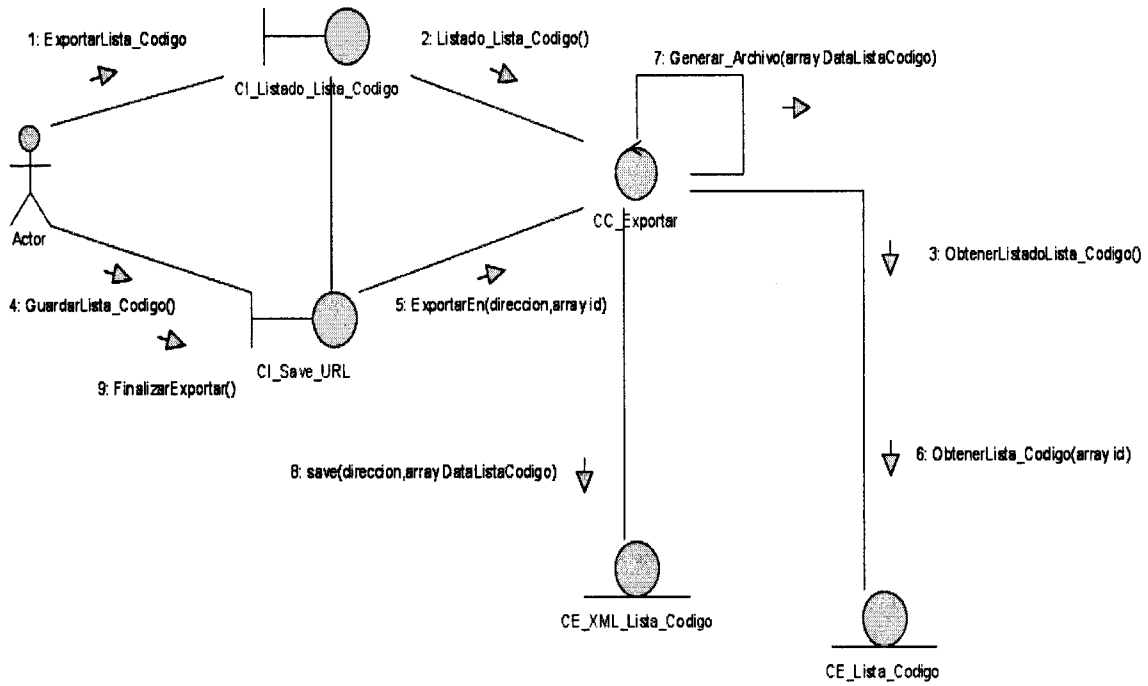
Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

Nombre: Tsolicitudes_insercion		
Descripción: En esta tabla se almacenan los datos de cada una de las solicitudes que se realizan ya sea por otra aplicación o por un usuario.		
Atributo	**Tipo**	**Descripción**
informacion_nueva	String	Este campo contiene la información nueva que se va a guardar en la base de datos a solicitud del usuario.

Tabla: Tsolicitudes_insercion

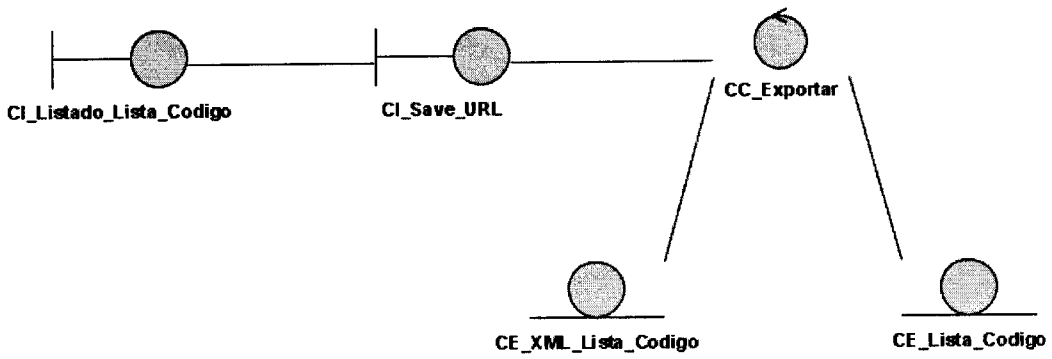
Modelos de análisis



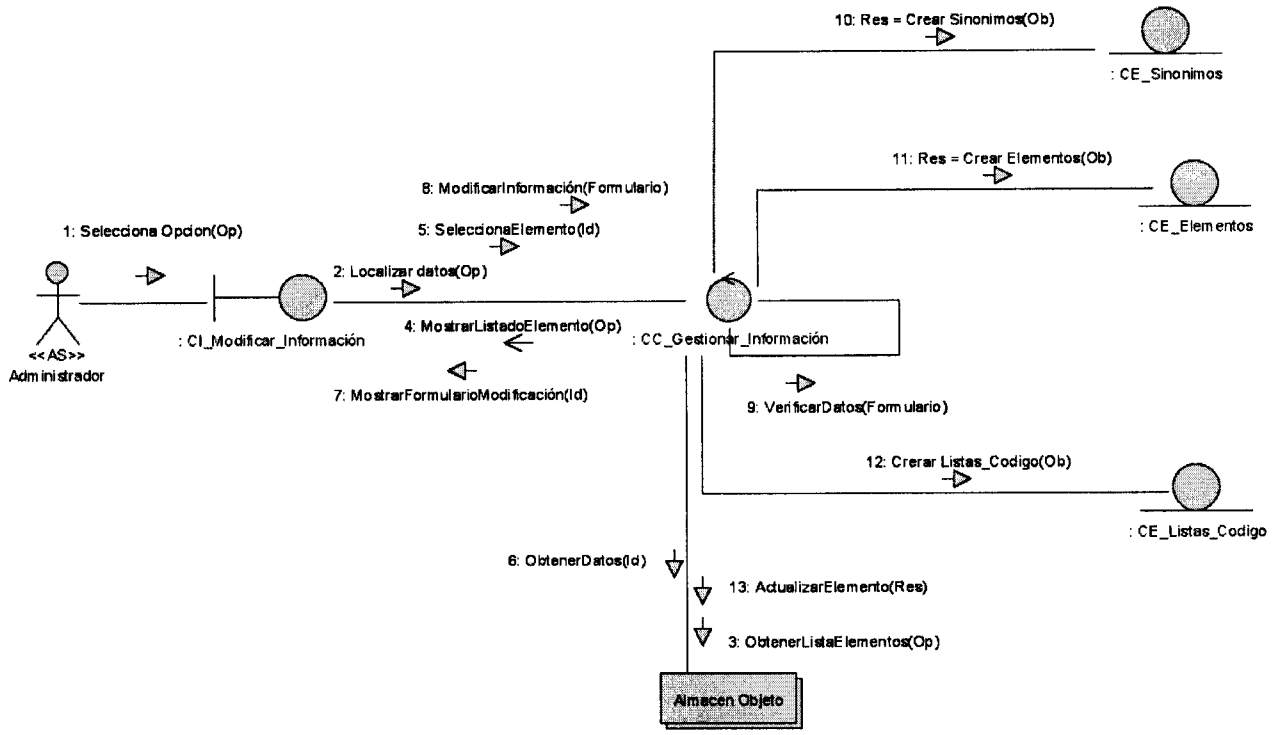
CU Exportar(Diagrama de colaboración)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



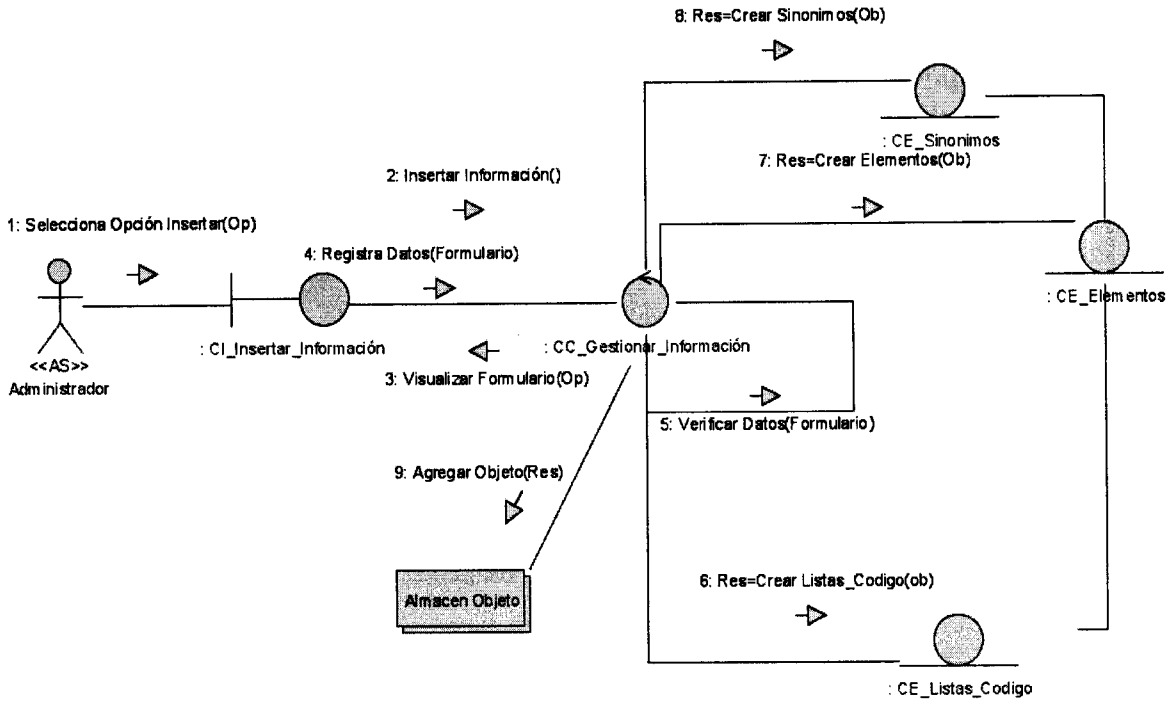
CU Exportar(Diagrama de clases del análisis)



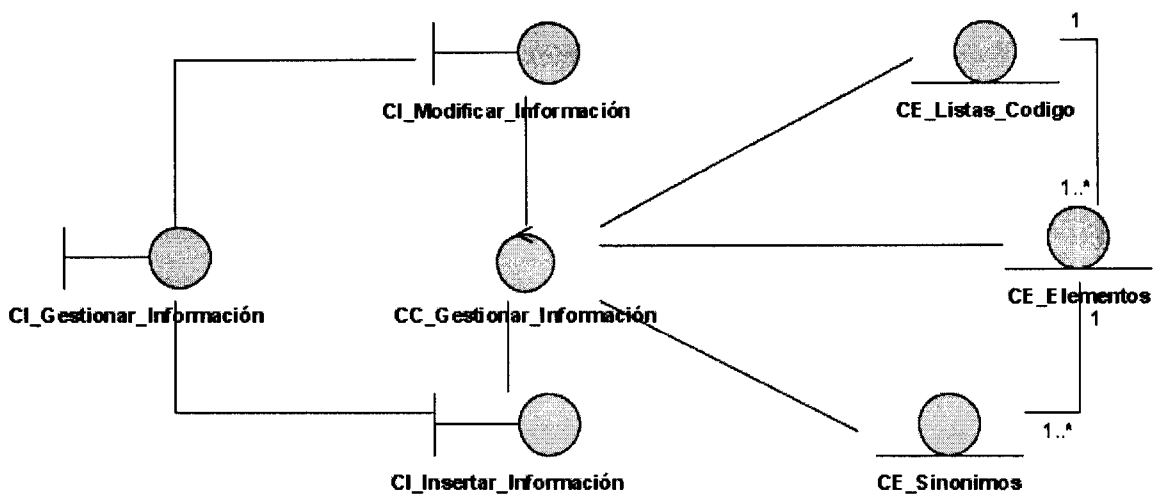
CU Gestionar Información (Sección Modificar_Diagrama de colaboración)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



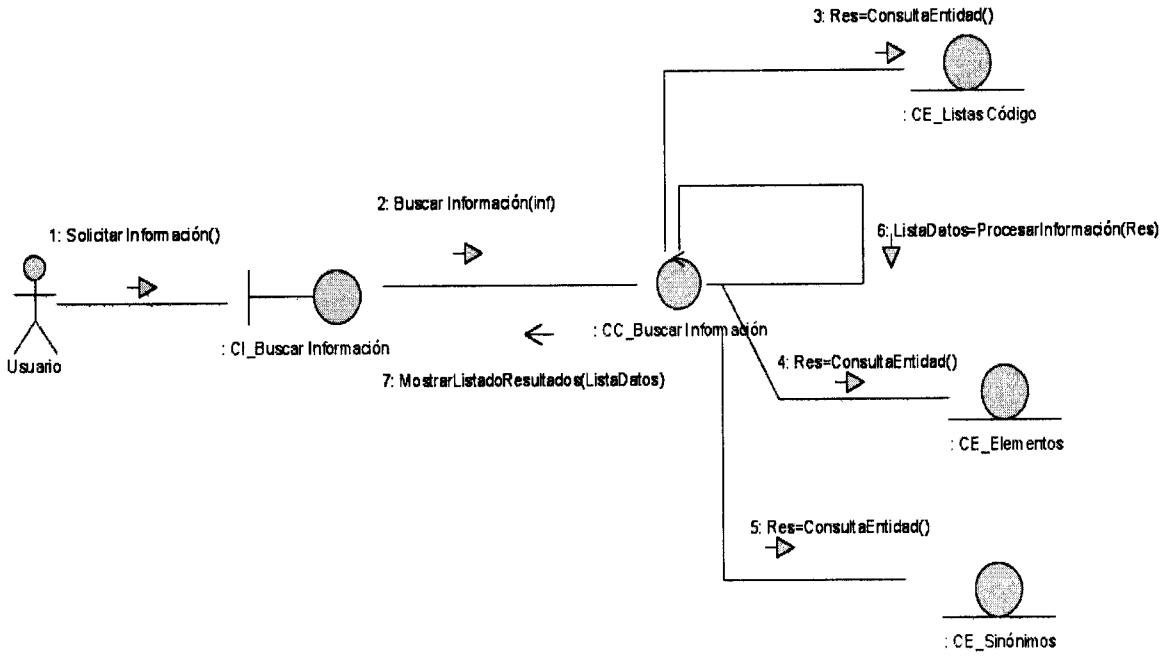
CU Gestionar Información (Sección Insertar_Diagrama de colaboración)



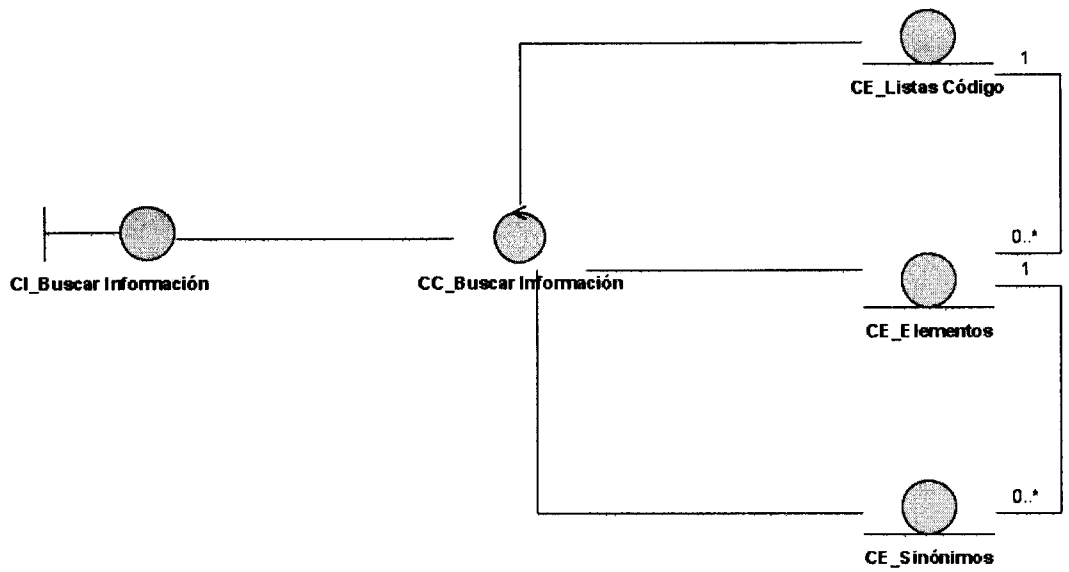
CU Gestionar Información (Diagrama de clases del análisis)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



CU Buscar información (Diagrama de colaboración)

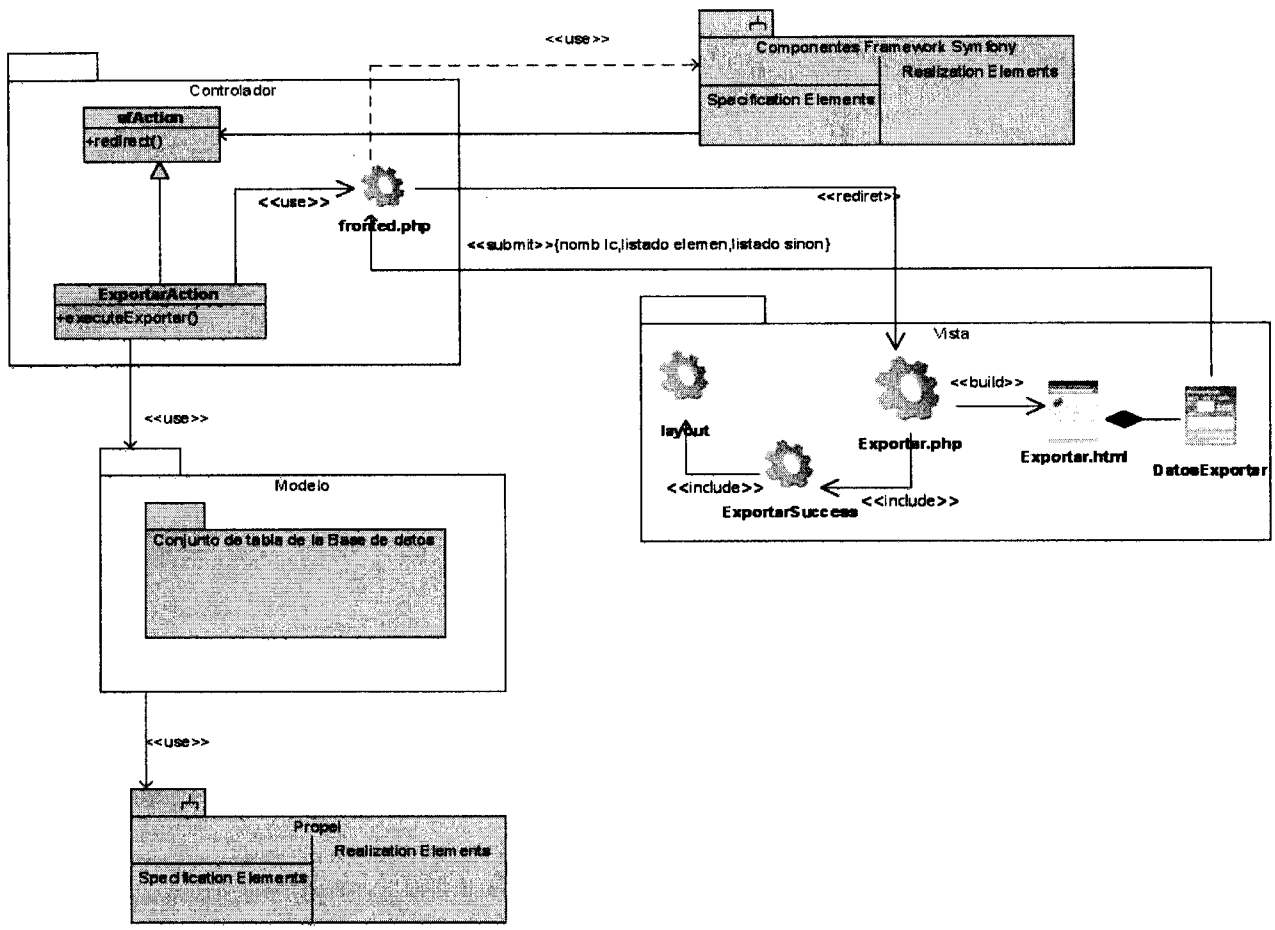


CU Buscar información (Diagrama de clases del análisis)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz

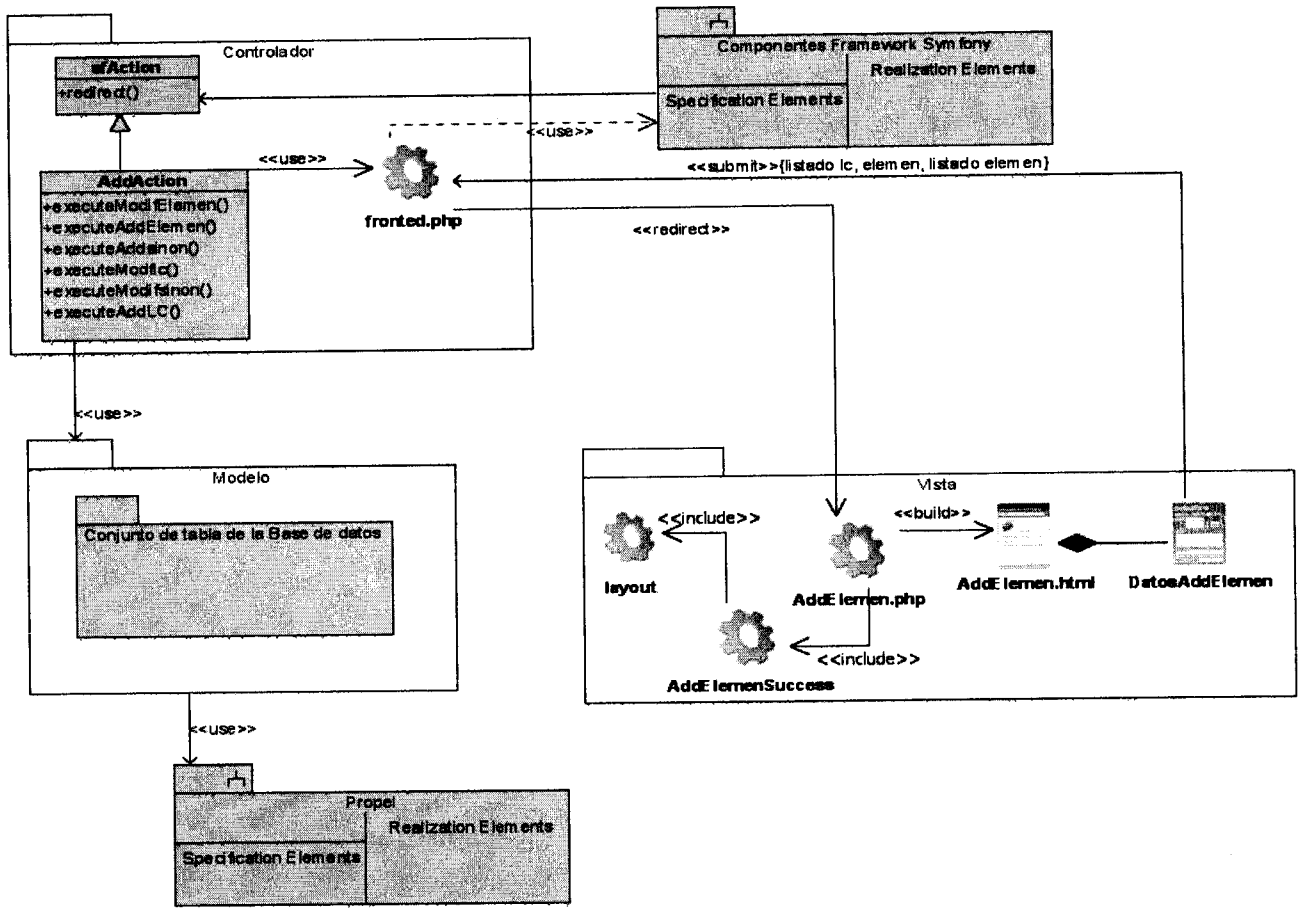
Modelos de diseño



CU Exportar

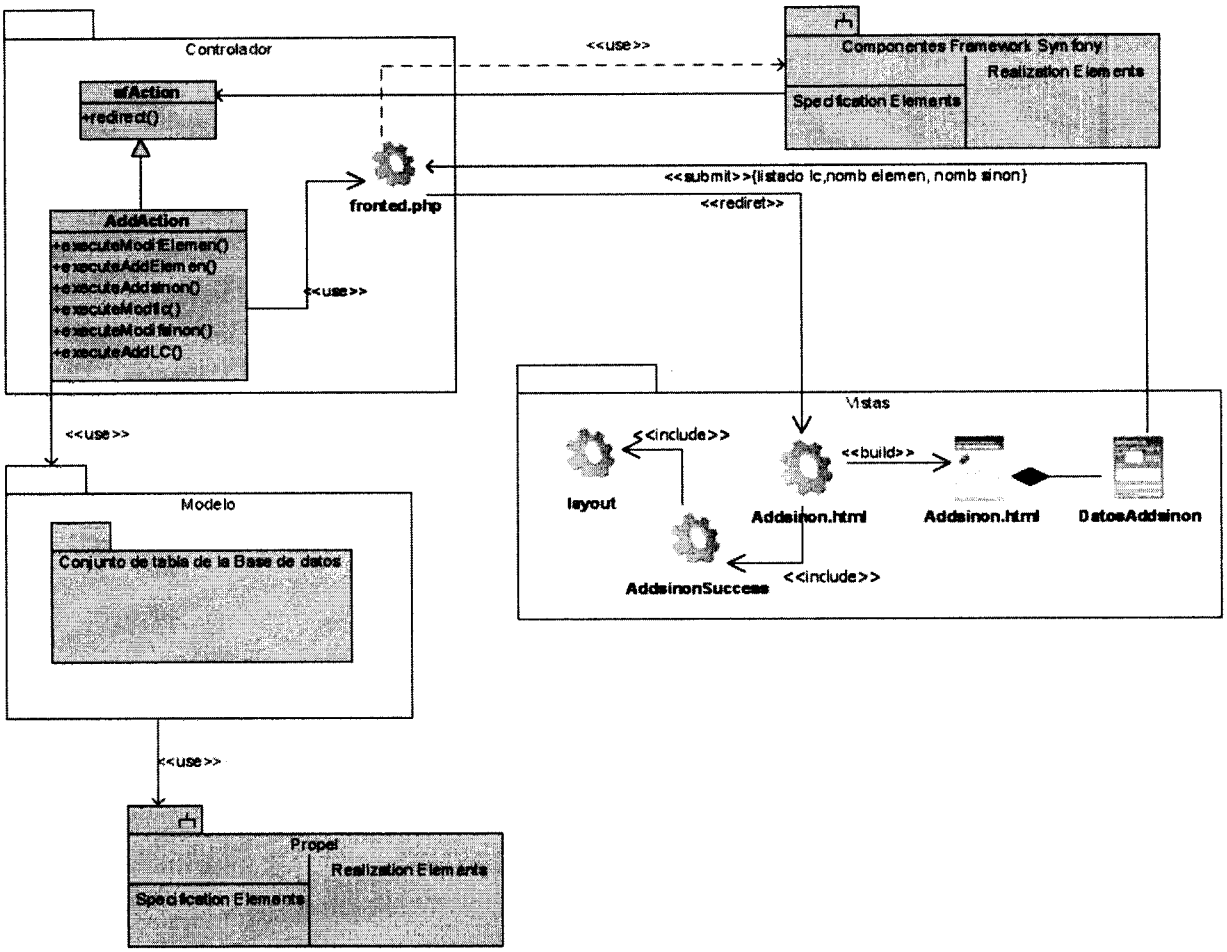
Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



CU Gestionar Información (Sección Insertar_Elementos)

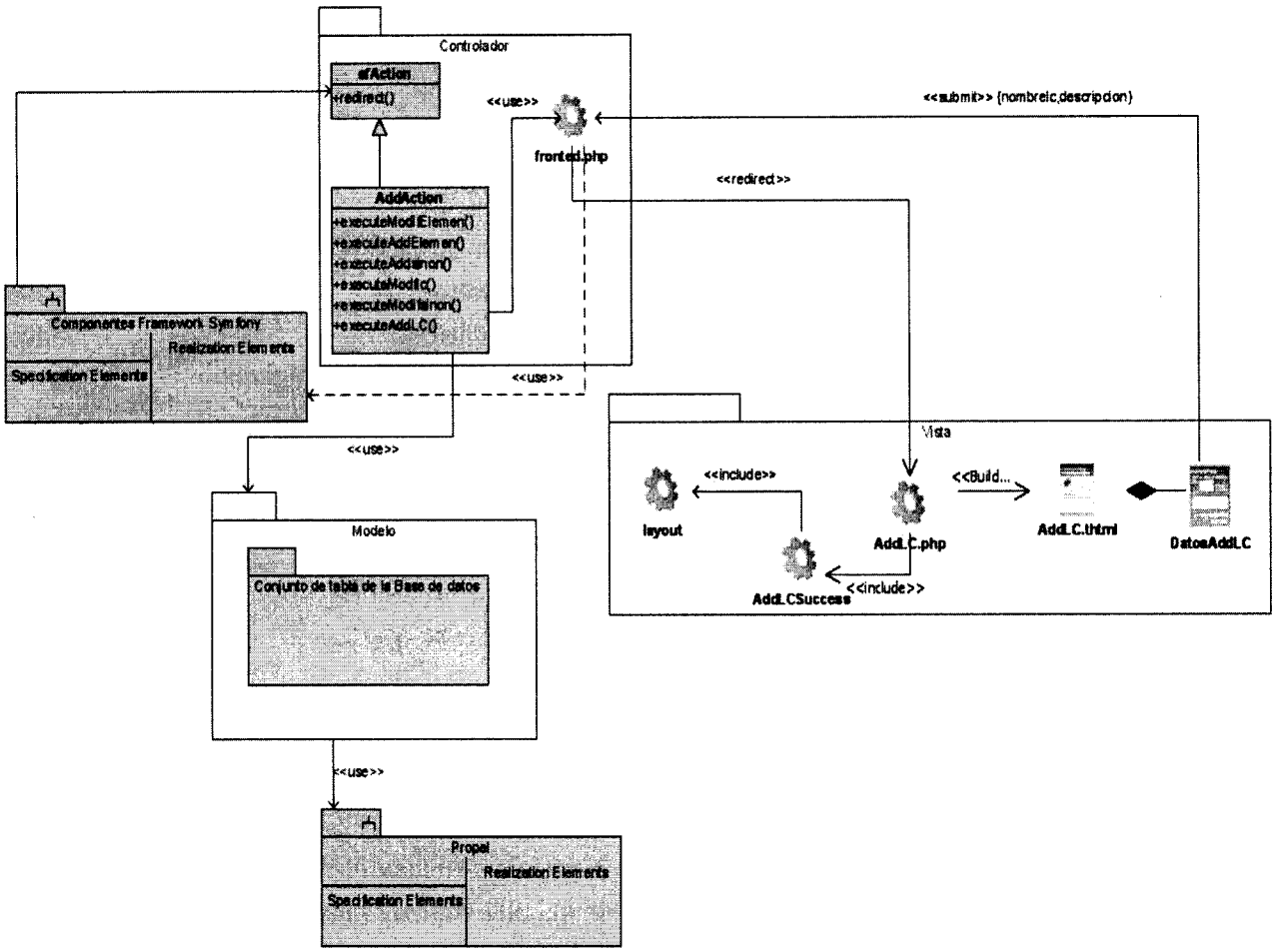
Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.
Fidel Castro Ruz



CU Gestionar Información (Sección Insertar_Sinónimos)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

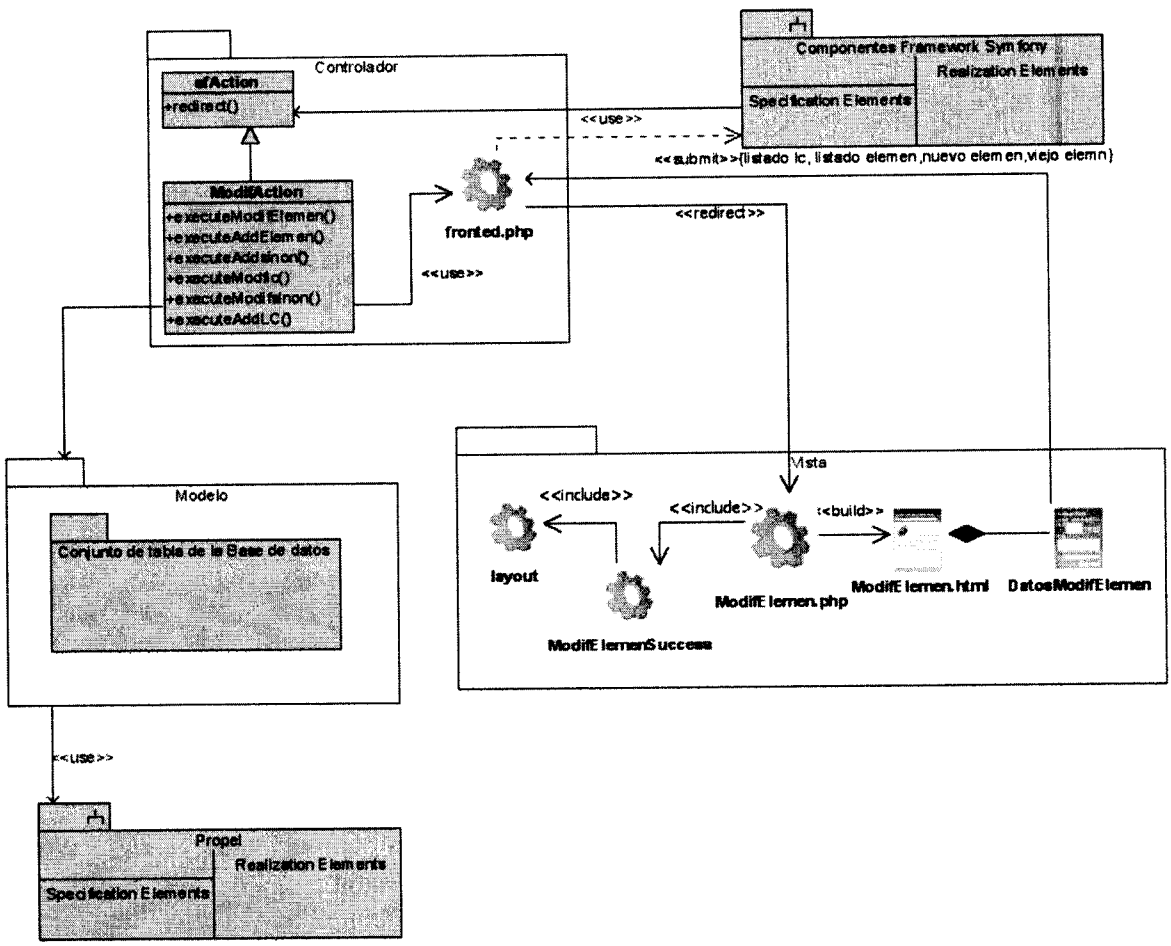
Fidel Castro Ruz



CU Gestionar Información (Sección Insertar_ListasCódigo)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

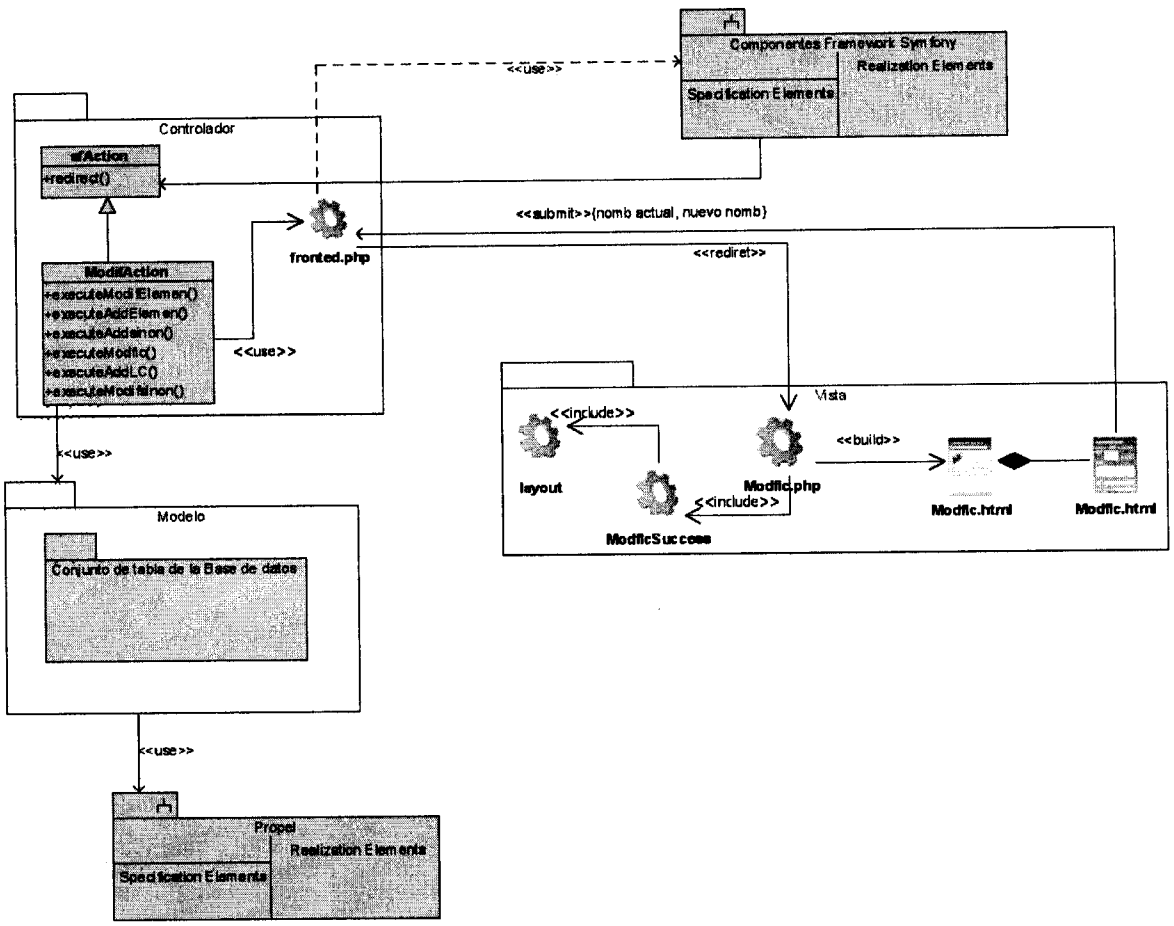
Fidel Castro Ruz



CU Gestionar Información (Sección Modificar_Elementos)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

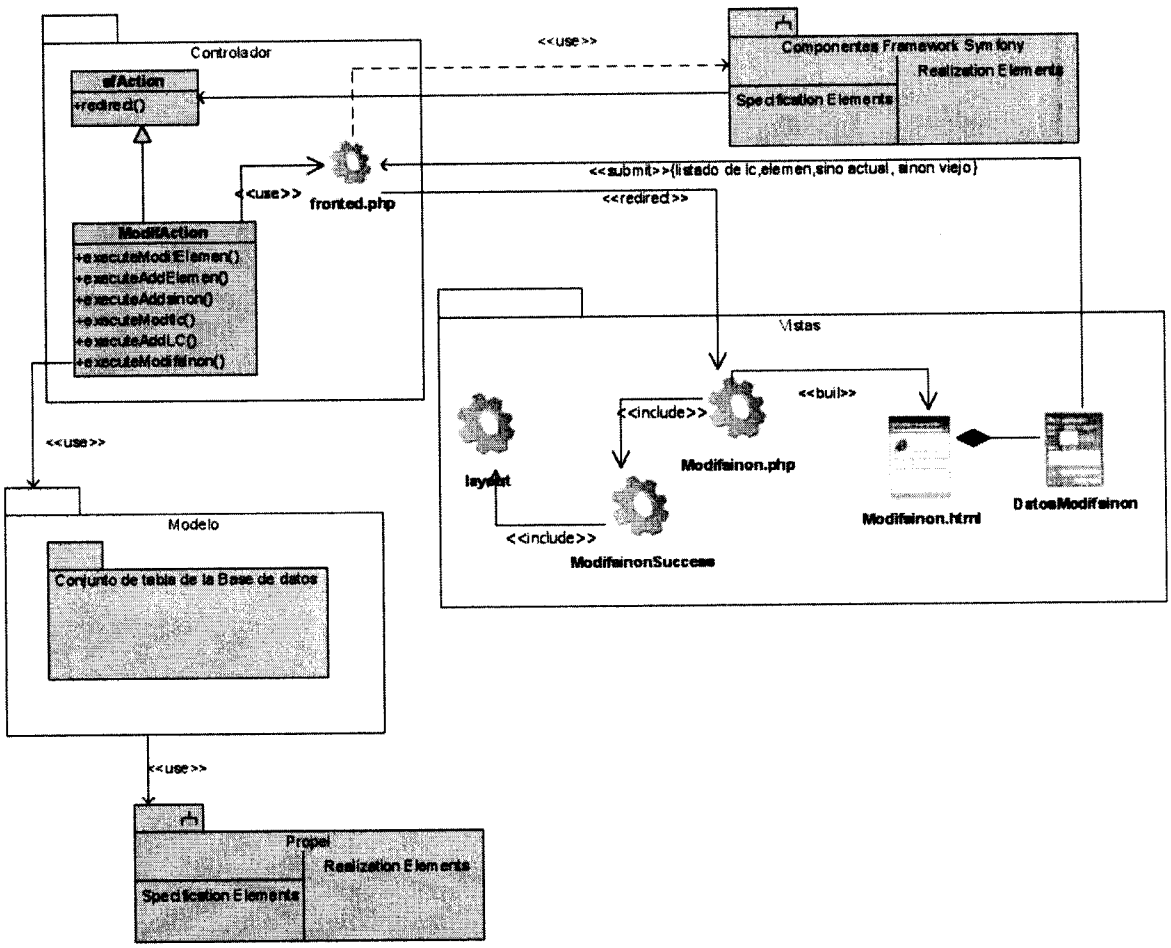
Fidel Castro Ruz



CU Gestionar Información (Sección Modificar_ListasCódigo)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

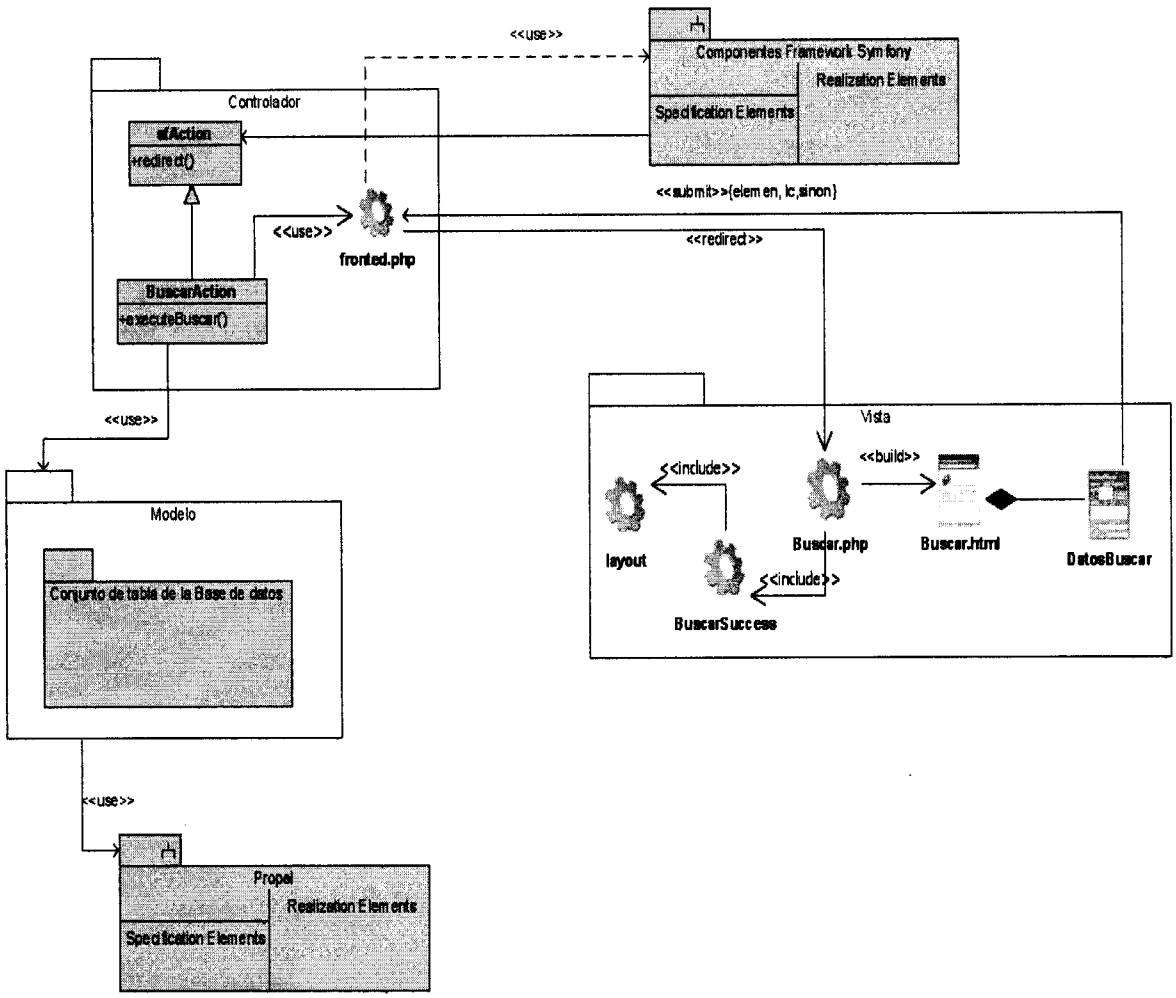
Fidel Castro Ruz



CU Gestionar Información (Sección Modificar_Sinónimos)

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz



CU Buscar Información

Tan solo hace falta una pequeña idea, para hacer un gran sueño realidad.

Fidel Castro Ruz