



Universidad de las Ciencias
Informáticas

Facultad 2

Título: Portal Comercial de Contenidos y Servicios de la
Plataforma de Gestión de Contenidos para Dispositivos
Móviles (PGCDM).

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

Autores: Dainelvis Martínez Basulto.

Miguel Ángel Torres Pérez.

Tutor: Ing. Rosa María Renté Labrada.

Ciudad de la Habana, Mayo del 2010.

Dedicatoria

A Nuestros Padres...

Resumen

La comunicación es un proceso vital para el desarrollo y en la actualidad se han multiplicado en gran número las posibilidades de comunicación a distancia; uno de estos medios más populares y con más difusión en el mundo es el teléfono celular. La telefonía móvil o celular se conoce desde mediados del siglo XX, aunque no fue hasta la década del '70 que comenzó a popularizarse. Desde entonces, la permanente investigación y adelantos tecnológicos permiten enmarcar etapas bien definidas dentro de las comunicaciones móviles, denominadas generaciones, de manera que un teléfono móvil de última generación permite no solo realizar llamadas, sino reproducir música y videos, tomar fotos y pequeñas películas, navegar por internet, enviar y recibir mensajes de texto y gráficos, acceder a prestaciones como juegos electrónicos e incluso ya se puede ver por ellos televisión digital.

Con el transcurso de los años la utilidad de este dispositivo ha aumentado producto a la gran variedad de servicios que puede prestar. Entre ellos se encuentran los Servicios de Valor Agregado, por ejemplo: titulares de noticias, partes climatológicos, busca en directorios telefónicos, entre muchos otros. Además inmersos en esta revolución económica, los sistemas de gestión de contenidos para dispositivos móviles aportan a los operadores móviles mayor capacidad al nivel de aplicación y de servicio, cubriendo las necesidades requeridas para fortalecerse en un mercado tan transformador.

La telefonía móvil cubana también se enfrenta a los retos de la revolución tecnológica. La Empresa de Telecomunicaciones de Cuba (ETECSA), único operador de telefonía móvil del país, se propone ofrecer servicios y contenidos para los móviles, llevando a cabo acciones como el desarrollo de una Plataforma de Gestión de Contenidos para Dispositivos Móviles.

Este trabajo tiene como objetivo el desarrollo de un Portal Comercial de Contenidos y Servicios para la Plataforma de manera que entre otros aspectos se amplíe el mercado al que está destinada.

Índice

<i>Introducción</i>	8
<i>Capítulo 1: Fundamentación Teórica</i>	11
Introducción	11
Comercio Electrónico	11
Modelos de Comercio Electrónico	12
Comercio a Consumidor (B2C – Business to Consumer).....	12
Marketing Digital.	13
Contenidos Digitales para la Telefonía Móvil	13
Telefonía Móvil.....	13
SMS (Short Message Service) y los Contenidos Digitales	14
Portales Comerciales de Contenidos para la Telefonía Móvil.....	15
Estado del Arte.....	15
Actualidad del Trabajo.	15
Metodología de Desarrollo y Lenguaje de Modelado	18
RUP.....	18
UML.....	19
Herramienta CASE	20
Visual Paradigm.....	20
Plataforma de Desarrollo	20
Java.....	20
J2EE (Java 2 Enterprise Edition)	22
Entorno de Desarrollo	22
Eclipse Galileo.....	22
Framework	22
Spring	22

Spring Web Services (Spring-WS).....	23
Spring Security.....	23
GWT.....	24
Apache Tomcat.....	25
Herramienta de Control de Versiones: SVN.....	26
Conclusiones.....	27
<i>Capítulo 2: Características del Sistema.....</i>	<i>28</i>
Introducción.....	28
Propuesta del Sistema.....	28
Modelo de Dominio.....	28
Relación de los Requerimientos.....	30
Requerimientos Funcionales (RF).....	30
Requerimientos No Funcionales.....	33
Modelo de Casos de Uso del Sistema.....	34
Definición de los Actores.....	34
Listado de Casos de Uso del Sistema.....	34
Diagrama de Casos de Uso del Sistema.....	35
Conclusiones.....	38
<i>Capítulo 3: Diseño.....</i>	<i>39</i>
Introducción.....	39
Diseño del Sistema.....	39
Definición del Modelo del Diseño.....	39
Diagramas de Clases del Diseño.....	39
Diagramas de Interacción (Secuencia).....	39
Descripción de la Arquitectura.....	39
Patrones Utilizados.....	39
Patrón Arquitectónico.....	40
Patrones de Diseño.....	42

Conclusiones.....	45
<i>Capítulo 4: Implementación.</i>	46
Introducción.....	46
Modelo de Implementación.....	46
Diagrama de Despliegue.....	46
Diagrama de Componentes.....	47
Conclusiones.....	49
<i>Capítulo 5: Estudio de Factibilidad.</i>	50
Introducción.....	50
Método de Estimación “Puntos de Casos de Uso”.....	50
Definición y Conceptos fundamentales.....	50
Cálculo de Puntos de Casos de Uso.....	50
Estimación del Esfuerzo.....	54
Análisis y Estimación del Costo.....	56
Cálculo del Tiempo de Desarrollo de todo el Proyecto.....	56
Conclusiones.....	57
<i>Conclusiones.</i>	58
<i>Recomendaciones.</i>	59
<i>Referencias Bibliográficas.</i>	60
<i>Bibliografía.</i>	62
<i>Glosario de Términos.</i>	65

Tabla de Figuras.

Figura 1. Diagrama de Clases del Modelo del Dominio.....	29
Figura 2. Diagrama de Paquetes de Casos de Uso del Sistema	36
Figura 3. Diagrama de CU del Sistema Módulo Portal.....	36
Figura 4. Diagrama de CU del Sistema Módulo Contenidos	37
Figura 5. Diagrama de CU del Sistema Módulo Servicios	37
Figura 6: Diagrama de Paquetes. Vista Arquitectónica.	41
Figura 7: Diagrama de Arquitectura del Sistema.....	42
Figura 8: Diagrama de Despliegue.....	46
Figura 9: Diagrama de Componentes General.	47
Figura 10: Diagrama de Componentes. Capa Presentación.	48
Figura 11: Diagrama de Componentes. Capa Negocio.	49

Introducción

Los avances de la tecnología móvil durante los últimos años, el aumento de las capacidades tecnológicas en hardware y el software, unido al surgimiento del nuevo paradigma en cualquier lugar y en cualquier momento, ha propiciado una transición de servicios electrónicos a servicios móviles, de comercio electrónico a comercio móvil. Hoy en día los servicios de telefonía móvil van más allá de la comunicación basada en voz y mensajes de texto. Abarcan además, prestación de servicios y comercialización de contenido digital orientados al ocio y mejoramiento de la calidad de vida de sus usuarios. Las plataformas móviles están evolucionando y rápidamente convirtiéndose en una plataforma de la entrega de servicios y contenido digital.

El progresivo crecimiento del mercado de la telefonía móvil propiciado por la disminución de sus costos , el creciente interés de sus usuarios por los contenidos digitales y las oportunidades comerciales que estos generan, demanda la convergencia de sectores de la industria como las telecomunicaciones inalámbricas, medios de comunicación, de entretenimiento y la informática.

Con este objetivo la Empresa de Telecomunicaciones de Cuba S.A, encargada de la prestación de estos servicios en el país, en convenio con la Universidad de Ciencias Informáticas (UCI), propone el desarrollo de una plataforma para gestionar contenidos para dispositivos móviles (PGCDM). Como parte de esta solución se incluye el desarrollo de un portal wap para la comercialización de los contenidos de dicha aplicación.

La pequeña pantalla de los teléfonos celulares y su menor potencia de procesamiento, brindan poca libertad para el diseño de las páginas para portales wap, las imágenes que se pueden incluir en ellas se reducen a gráficos estáticos muy sencillos y se debe incluir únicamente información significativa y eliminar cualquier contenido irrelevante con el objeto de mejorar su rendimiento. Esto hace que los portales WAP sean poco atractivos para los usuarios, tenga limitaciones a la hora de brindar información detallada a cerca de los contenidos y atraer la atención de los clientes hacia estos. Además, no se aplican técnicas de marketing digital, de afiliación a clientes y el servicio de descarga de los contenidos brindado por el Portal Wap solo puede ser consumido desde un teléfono móvil, por lo que los usuarios de la Internet no podrán

descargarlos desde un explorador web en una computadora convencional, a pesar de que el portal wap pueda ser visualizado en este.

A partir de estas necesidades surge el **problema científico**: ¿Cómo lograr una presentación atractiva de los contenidos, brindar información detallada de estos, la afiliación de nuevos clientes a la PGCDM y la prestación de un servicio de descarga de contenidos disponible desde un explorador web?

Se define como **objeto de estudio** de la investigación:

- Los Portales Comerciales.

Dicho objeto enmarca el **campo de acción** de esta investigación como:

- Los Portales Comerciales de Contenidos para Dispositivos Móviles.

El **objetivo principal** de esta investigación es:

- Desarrollar un portal web comercial para lograr una presentación atractiva de los contenidos, brindar información detallada de estos, la afiliación de nuevos clientes a la PGCDM y la prestación de un servicio de descarga de contenidos disponible desde un explorador web.

Para lograr el objetivo propuesto se han trazado una serie de **tareas de investigación** mencionadas a continuación:

- Realizar un estudio del estado del arte de Portales Comerciales para Contenidos Móviles
- Definir las técnicas de marketing electrónico y afiliación a clientes a utilizar.
- Definir las herramientas y tecnologías a utilizar para el desarrollo del portal.
- Definir los requisitos del diseño gráfico del portal.
- Definir herramientas y métodos de pruebas de rendimiento y seguridad del portal.
- Realizar pruebas y analizar resultados.

Como **resultado** de lo anteriormente planteado se espera obtener:

- El desarrollo de un portal web atractivo, práctico y amigable, que permita explorar, buscar, obtener información detallada y vistas previas de los contenidos gestionados por la PGCDM, la promoción

de servicios dirigidos al cliente y en función de los intereses de la empresa. Además, que incluya técnicas de afiliación y marketing digital dirigido al cliente.

El contenido de esta investigación está desglosado en 5 capítulos, las conclusiones generales, recomendaciones, referencias bibliográficas y bibliografía utilizada, un glosario de términos y por último los anexos que complementan el cuerpo del trabajo y que son necesarios para su entendimiento.

Este trabajo ha sido estructurado de la siguiente forma:

El **Capítulo 1 Fundamentación Teórica** contiene un breve estudio de los principales conceptos tratados en el trabajo. Cuenta con una reseña de otros sistemas utilizados en el mundo para iguales propósitos y un estudio de las tecnologías y herramientas actuales, haciendo énfasis en las utilizadas en el desarrollo del Portal. El **Capítulo 2 Características del Sistema** contiene la descripción de los tipos de objetos más importantes que existen en el dominio. Se definen como se interrelacionan estos objetos. Contiene además la captura de los requisitos funcionales y no funcionales, el diagrama de casos de usos del sistema, así como las descripciones de los casos de usos.

En el **Capítulo 3 Análisis y Diseño del Sistema** se define el modelo de clases del análisis y el diseño, los diagramas de interacción, así como la Arquitectura utilizada y la seguridad a tener en cuenta. Este capítulo está enfocado hacia como construir técnicamente el software. El **Capítulo 4 Implementación** contiene una descripción y representación gráfica del modelo de implementación y despliegue del sistema. Y el **Capítulo 5 Estudio de Factibilidad** contiene una estimación del tiempo de duración del proyecto de software y una estimación del costo del mismo. Además cuenta con un breve análisis de los beneficios tangibles e intangibles que se esperan del mismo.

Capítulo 1: Fundamentación Teórica.

Introducción

Este capítulo contiene un estudio valorativo sobre los Portales Comerciales de Contenidos y Servicios para Dispositivos Móviles. Incluye el estado actual de este proceso, así como las tendencias tecnológicas y metodológicas más utilizadas por los desarrolladores en la esfera del Comercio Electrónico. Se definen además las herramientas y tecnologías que se utilizarán en el desarrollo del sistema propuesto.

Comercio Electrónico

El Comercio Electrónico, en inglés Electronic Commerce, E-Commerce, se puede definir en un sentido amplio, como las actividades comerciales realizadas con el uso de tecnologías de transmisión electrónica de datos tales como las empleadas en Internet y la World Wide Web.

El Comercio Electrónico se ha convertido en los últimos años en una nueva vía para el desarrollo de nuevas oportunidades de negocio para las empresas, que les permite dar a conocer sus marcas, establecer vínculos comerciales de empresa a consumidor y de empresa a empresa en forma automatizada y adquirir y ofertar productos y/o servicios de manera eficiente a través de redes privadas o de la Internet. Además, posibilita que sus consumidores, principales “accionistas” de estas, puedan hacer, de una manera más sencilla, sin costo de desplazamiento ni de tiempo, la compra de los artículos que consideren más convenientes a sus necesidades.

Oportunidades y Beneficios:

➤ Globalización

Los suministradores pueden disfrutar de una presencia global a un coste impensable a través de otros medios. Los consumidores tienen una capacidad de elección dentro de una oferta extensa que no viene limitada por distancias geográficas ni restricciones al comercio.

➤ **Reducciones de costes y precios**

El acortamiento del ciclo de compra conlleva disminución de costes, no solo por requerir un número menor de intermediarios -de búsqueda, promoción, entrega, cobro, etcétera- sino que un gran número de estos procesos pueden ser automatizados, reduciéndose los costes en órdenes de magnitud, lo que inevitablemente quedara reflejado en los precios de las mercancías.

➤ **Mejora de calidad de servicios**

La “proximidad” al usuario y el acortamiento de ciclos de respuesta son elementos fundamentales que posibilitan una mejora de la calidad de servicio, como servicios de consulta o de postventa, y de una mejora de la competitividad.

Modelos de Comercio Electrónico

En el comercio electrónico participan como actores principales los mismos que en cualquier transacción comercial, es decir: las empresas, los consumidores y las administraciones públicas como entes reguladores de dichas transacciones. De este modo, se distinguen normalmente tres modelos o formas básicas de realizar el comercio electrónico: comercio a comercio (B2B), comercio a consumidor (B2C) y comercio a gobierno (B2G). Recientemente se han comenzado a utilizar otros dos modelos de comercio, uno que agrupa el comercio a comercio con comercio al consumidor (B2B2C) y el otro que es empresa a empleados (B2E). El modelo seleccionado a aplicar en la solución fue el de Comercio a Consumidor (B2C), ya que es el más aplicable al modelo comercial del cliente. A continuación se explica.

Comercio a Consumidor (B2C – Business to Consumer).

El comerciante ofrece sus bienes y servicios al consumidor que visita su dirección electrónica o página Web con el objetivo de satisfacer alguna necesidad de consumo o bien de entretenimiento. Esta categoría incluye todos aquellos sitios de Internet que vendan cualquier tipo de producto u ofrezcan servicios al público en general.

Las principales ventajas del B2C:

- Comunicación directa con los clientes sin intermediarios.
- No es necesario poseer grandes cantidades de inventario físico para poder vender en línea, lo

importante es contar con rápidas soluciones de distribución.

- Mejoramiento de la imagen de la empresa.
- Grandes y pequeñas empresas pueden mostrar sus catálogos en línea y vender sus productos sin incurrir a altas inversiones de publicidad.
- La compra suele ser más rápida y más cómoda. Las ofertas y los precios están siempre actualizados.
- Las empresas intervienen como usuarios (compradoras o vendedoras) y a la vez, como proveedoras de herramientas o servicios de soporte para el comercio electrónico.

Marketing Digital.

El Marketing Digital es un sistema interactivo dentro de un conjunto de acciones de marketing de la empresa, que utilizan los sistemas de comunicación telemáticos para conseguir una respuesta medible ante un producto, una transacción comercial y que sus clientes o clientes potenciales puedan tanto efectuar consultas al producto como seleccionar y adquirir la oferta existente al momento de un determinado producto.

Estrategias de Marketing a aplicar:

- **Marketing de retención:** se trata de crear motivos para que tanto visitantes como clientes del portal permanezcan en él.
- **Marketing de recomendación:** tiene como máxima finalidad conseguir que el cliente satisfecho actúe como prescriptor, y de comienzo a un nuevo ciclo. Suele estar apoyada en la técnica del marketing viral, es decir, en formas estudiadas de generación de red por parte de los usuarios y/o clientes, en las que estos generan un número determinado de recomendaciones, dependiendo del círculo de amistades o "contactos" que tenga estos.

Contenidos Digitales para la Telefonía Móvil

Telefonía Móvil

La telefonía móvil es un sistema de comunicación telefónica. Consiste en la combinación de una red de estaciones transmisoras-receptoras de radio y una serie de centrales telefónicas de conmutación que

posibilita la comunicación entre terminales telefónicos portátiles (teléfonos móviles) o entre terminales portátiles y teléfonos de la red fija tradicional.

Tanto los desarrollos de estándares tecnológicos como la fuerte implantación social en relación al uso cotidiano, la portabilidad y la identidad individual han hecho del teléfono móvil el dispositivo idóneo para aglutinar buena parte de los usos que caracterizan a la Sociedad de la Información, convirtiéndolo en un dispositivo abarcador de dispositivos digitales y sentando las bases para una verdadera integración de opciones y servicios que constituye la base del proceso de mediatización de la telefonía móvil.

SMS (Short Message Service) y los Contenidos Digitales

El servicio de mensajes cortos o SMS es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. Los mensajes cortos hacen un uso extremadamente eficaz de la red de radio, y además pueden ser enviados y recibidos en cualquier momento, incluso durante una llamada.

Desde su implantación, el SMS ha alcanzado una alta penetración entre los usuarios jóvenes. La implantación SMS constituye un hito sociológico y comercial, dadas sus condiciones de ahorro en los costes de la comunicación (limita el tiempo de conexión al momento del envío), seguridad de destino y discreción (el nivel de interferencia en la vida cotidiana es menor que el de la conversación). Pero, sobre todo, la importancia del SMS en la evolución de la telefonía móvil reside en que implica una ruptura en la concepción de la tecnología (paso de la transmisión de voz a la transmisión de datos), abriendo la puerta al desarrollo del teléfono móvil como dispositivo de acceso a contenidos y publicidad (paso de la conectividad interindividual a la conectividad medio/usuario), donde posterior el incremento y diversificación de formatos de datos demanda un desarrollo coherente de estándares tecnológicos de intercambio: el EMS (Enhanced Messaging Service) añade al envío de texto la posibilidad de transmitir imágenes, animaciones y sonidos en soporte SMS para GSM (Global System for Mobile Communications) y GPRS (General Packet Radio Services); y, a partir de 2002, el MMS (Multimedia Messaging Service) hace posible el envío de datos en forma de imágenes (JPEG), vídeo (MPEG), sonido (MP3, MIDI) y aplicaciones (JAVA) para GPRS y 3G (Tercera Generación). Finalmente, a partir de los estándares 3G, 3.5 G e i-Mode (mobile internet), el IMS (IP Multimedia Subsystem) incorpora la

transmisión de datos en formato IP, planteando el horizonte de una creciente compatibilidad con los estándares de Internet.

Portales Comerciales de Contenidos para la Telefonía Móvil

Un portal es un sitio en el Internet que proporciona un punto de entrada general para un arsenal enorme de recursos y servicios. Los portales suelen contener noticias, servicios gratuitos de correo electrónico, los motores de búsqueda, compras en línea, salas de chat, foros de discusión y enlaces a otros sitios. Yahoo, MSN y AOL son los portales típicos para entrar en la Internet. La idea de un portal es recopilar información de diferentes fuentes y crear un único punto de acceso a la información - una biblioteca de categorías y contenido personalizado. Por lo tanto, los portales deben obtener información de uno o más servidores, así como de la Internet, y entregar esa información a través de una única interfaz coherente. Dicho esto y con lo tratado anteriormente podemos concretar que los Portales Comerciales de Contenidos para Telefonía Celular son aquellos portales mediante los cuales una empresa dedicada a la comercialización de contenido digital para la telefonía móvil, aplica su modelo de comercio electrónico, ofreciendo sus bienes y servicios, con el objetivo de satisfacer necesidad de consumo y/o entretenimiento de clientes de la telefonía móvil.

Estado del Arte.

A continuación se hará un análisis de alguna de las funcionalidades presentes en portales que se dedican a la comercialización de contenidos para dispositivos móviles para de esta manera poder definir las características que debe presentar el portal que se va a desarrollar.

Actualidad del Trabajo.

❖ **GSMspain** (<http://www.gsmspain.com/>)

Este portal no cuenta con un diseño atractivo y poco intuitivo, sin soporte para varios idiomas, a pesar de contar con elementos que faciliten la navegación y ubicación de los usuarios en el sitio, con un apartado de ayuda al cliente. Cuenta con mecanismos de afiliación al cliente, pero las opciones de personalización son muy limitadas y cuenta con un sistema para la búsqueda de contenidos por compatibilidad con móviles. No hace uso de estrategias de marketing digital. En cuanto a los contenidos que comercializa son numerosos, están agrupados por categorías y subcategorías, cuenta con un buscador avanzado,

muestra información relevante a estos, permite su vista preliminar en línea, con vistas de lo más descargado y no es obligatorio el registro del usuario para realizar la descarga. La variante de descarga de contenidos usada es el envío de un SMS y no cuenta con mecanismos de seguridad. Posee varios servicios adicionales como venta de celulares, noticias, envío de SMS, foros, chat, de tarifas y especiales. Su rendimiento es bajo y cada acción requiere la recarga completa del sitio, siendo en ocasiones muy lento.

❖ **Sony Ericsson Play Now Arena** (<http://www.playnow-arena.com/>).

Cuenta con un diseño atractivo, ligero, fresco, intuitivo, con un buen uso de elementos de navegación, soporte para varios idiomas y una sección de ayuda al cliente. Cuenta con afiliación a clientes, un selector de dispositivo móvil y no usa estrategias de marketing digital. La disponibilidad de contenidos es muy amplia, con mecanismos de exploración, búsqueda personalizada, donde estos están bien organizados en categorías, sub-categorías, brindando información detallada y vistas previas de los contenidos. Cuenta con la variante de descarga directa al ordenador y además el envío del enlace al móvil vía SMS, donde en cualquiera de los dos casos es necesario estar registrado y autenticado en el sitio. No cuenta con servicios adicionales. El rendimiento del sitio es muy bueno, con el uso de técnicas Ajax y animaciones flash interactivas con lo que se logra un buen balance entre atractivo y eficiencia, a pesar de que estas hacen que el sitio sea poco ligero a la hora de la carga inicial.

❖ **Mobango** (<http://www.mobango.com/>).

Cuenta con un diseño colorido, atractivo, refrescante e intuitivo, aunque un poco cargado, con una excelente navegación, soporta varios idiomas y cuenta con ayuda al usuario en línea. Cuenta un filtro de compatibilidad de los contenidos mediante la selección del dispositivo móvil, un apartado para la afiliación de los clientes y hace uso de estrategias de marketing digital. Cuenta con un amplio número de categorías de contenidos, pero no sub-categorías, con detallada información acerca de los contenidos y vistas preliminares. Están disponibles las variantes de descarga directa al PC del usuario o por vías SMS al móvil de este, siendo requerida la selección del dispositivo para ambas vías. Cuenta con una comunidad de usuarios y grupos de usuarios entre los cuales se recomiendan contenidos, pueden seguir los más visitados por sus amigos y enviarse mensajes entre ellos. El rendimiento del sitio es aceptable, a pesar de no utilizar Ajax, pues hace un uso racional de las imágenes y animaciones.

❖ **Movistar en Venezuela** (<http://emocion.movistar.com.ve/>).

Cuenta con un diseño refrescante, atractivo e intuitivo, con uso amplio de animaciones, con navegación sencilla, sin soporte para varios idiomas ni soporte para el usuario. No cuenta con selector de móvil para el filtrado por compatibilidad, o mecanismos de afiliación a usuarios y tampoco hace uso de estrategias de marketing digital. Los contenidos se dividen en categorías y sub-categorías, con información básica a cerca de estos y vínculos a vistas preliminares. Los contenidos pueden ser descargados mediante envíos SMS y no tienen requerimiento alguno. No cuenta con servicios adicionales. El sitio por su sencillez es de rendimiento aceptable, sin obviar las demoras en las cargas de las animaciones flash e imágenes notables.

❖ **Movilnet** (<http://venezuela.movilnet.com.ve/descargas/>).

Es un portal atractivo, colorido, poco intuitivo, un tanto cargado, con buena navegabilidad, sin soporte para diferentes idiomas y con soporte en línea. No cuenta con selector de móvil, ni estrategias de marketing digital y si sistema para el registro de clientes. Los contenidos están agrupados en categorías y sub-categorías, con muy poca información a cerca de estos y con vista preliminar disponible. La descarga de contenidos es vía SMS y exige el registro previo. Cuenta con varios servicios extras como pronóstico del tiempo, noticias, servicios especiales, de información nacional y el gobierno en línea. Cuenta con un rendimiento aceptable, aunque el cargado de las animaciones e imágenes utilizadas, hacen que este disminuya inevitablemente.

Después de este estudio podemos concretar que estos portales de empresas competidoras cuentan con diseños atractivos, coloridos, agradables e intuitivos, cargados de animaciones e imágenes, en diferentes idiomas, con mecanismos de navegación sencillos, orientados a la visita del usuario, y con un apartado de ayuda en línea. Incluye también mecanismos de afiliación a clientes, de selección de dispositivos móviles para soporte de compatibilidad. En cuanto a los contenidos son clasificados en categorías y sub-categorías, tanto con exploradores como buscadores avanzados de los contenidos disponibles, brindando información detalla y vistas previas de estos, aunque son pocos los que utilizan estrategias de marketing digital identificadas anteriormente. La variante de descarga más utilizada es el envío de SMS, aunque varios cuentan con la variante de descarga a la PC, donde en cualquiera de los casos por lo general se exige el registro y autenticación del usuario. Algunos prestan servicios

adicionales para brindar valor agregado al usuario, haciendo que su visita sea más agradable y provechosa. El rendimiento por lo general no es el mejor, pues cada acción del usuario genera una recarga de la página cargada de imágenes y animaciones.

Metodología de Desarrollo y Lenguaje de Modelado

Una metodología de desarrollo es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un software. En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

RUP es una metodología para proyectos de largo alcance, pero es estructurable y puede ser modificado en dependencia de las necesidades del proyecto y de la experiencia del desarrollador en el uso de esta metodología, logrando que se ajuste también a proyectos con un corto tiempo de desarrollo. Es fácil de entender tanto por usuarios avanzados como por clientes con pocos conocimientos técnicos. Está basado en normas provenientes de estándares seguidos por el entorno de desarrollo. Tiene procesos mucho más controlados, con numerosas políticas o normas. La arquitectura del software es esencial y se expresa mediante modelos. Por todo lo expuesto se usa RUP como metodología de desarrollo [1].

RUP

El Proceso Unificado de desarrollo de Software (RUP) es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del lenguaje UML (del inglés Unified Modeling Language), y el trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado se creó en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0, de ahí las siglas con las que se identifica a este proceso de desarrollo.

RUP está conformado por fases y flujos de trabajo. Las fases son inicio, elaboración, construcción y transición, los flujos de trabajo están comprendidos en 2 grupos, 6 de ingeniería: modelación del negocio, requerimientos, análisis y diseño, implementación, prueba e instalación y 3 de soporte: administración del proyecto, administración de configuración y cambios y por último ambiente.

Características de RUP:

-
1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo debido a que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
 2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
 3. **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

RUP establece actividades y criterios que conducen a un sistema desde su máximo nivel de abstracción hasta su nivel más concreto. UML (Unified Modeling Language) ofrece la notación gráfica necesaria a RUP para representar los sucesivos modelos que se obtienen en este proceso [2] [3].

UML

Para el modelado se usa UML. Este lenguaje de modelado aparece en 1995 como el Método Unificado, que posteriormente se reorienta para dar lugar al Unified Modeling Language (UML), la notación orientada a objetos más popular en la actualidad. UML es un lenguaje de modelado gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de estos es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. Ha sido adoptado por muchos proveedores de herramientas CASE y es el lenguaje de modelado soportado por RUP, metodología escogida para el desarrollo del sistema [4].

Herramienta CASE

Visual Paradigm

Es una herramienta CASE, utilizada en un ambiente de software libre, debido a la posibilidad de ejecutarse sobre cualquier sistema operativo, por lo que se convierte en una herramienta multiplataforma, permite crear diferentes tipos de diagramas en un ambiente totalmente visual.

Soporta además el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite graficar todos los tipos de diagramas de clases, generar documentación, generar código desde diagramas y código inverso. También proporciona tutoriales, demostraciones interactivas y proyectos UML.

Como principales características se encuentran: [5].

Beneficios:

- Navegación intuitiva entre el modelo visual y el código.
- Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- Sincronización entre el código fuente y el modelo en tiempo real o bajo demanda.
- Entorno visual de modelado superior.
- Soporte para toda la notación UML.
- Análisis de textos.
- Sofisticados y automáticos diagramas de capas.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

Plataforma de Desarrollo

Java

La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Java es un lenguaje de alto nivel, orientado a objetos, multiplataforma, es decir, que permite la ejecución de

un mismo programa en múltiples sistemas operativos, como dice el axioma de Java, “write once, run everywhere”. Es distribuido ya que contiene un conjunto de clases que permiten la comunicación a través de la red facilitando así la creación de aplicaciones distribuidas.

Tiene una plataforma, que a diferencia de las demás que son una combinación de hardware y software, está basada en software, que corre sobre cualquier hardware. Consta de 2 componentes: la máquina virtual de java (JVM) y la interfaz de programación de aplicaciones (API). Este lenguaje permite a los desarrolladores implementar software en una plataforma y ejecutarlo prácticamente en cualquier otra y combinar aplicaciones o servicios basados en la tecnología Java para crear servicios o aplicaciones totalmente personalizados. Es libre y les proporciona a los desarrolladores una mayor flexibilidad, ahorro en los costes de desarrollo, independencia, compatibilidad con otros sistemas, entre otros beneficios [6] [7].

La plataforma Java es originaria de Sun Microsystems. Dentro de la plataforma Java se pueden destacar tres sub-estándares [8]:

- JSE (Java Standard Edition)
- JME (Java Micro Edition)
- JEE (Java Enterprise Edition)

El desarrollo de aplicaciones en Java nunca empieza desde cero. Java implementa consigo un gran número de clases, incluidas gratuitamente en su entorno de desarrollo, para realizar variadas tareas que permiten al desarrollador centrarse en el negocio y no en la implementación. Permite además, la incorporación de un gran número de componentes reutilizables. El lenguaje de programación Java es el único soportado por la inmensa mayoría de dispositivos móviles [9].

Los programas de Java son mucho más seguros que los de C o C++; es decir, existen menos posibilidades de que generen errores graves. Java no tiene punteros. La Máquina Virtual de Java se ocupa además de administrar la memoria, por lo que no existen pérdidas de memoria, uno de los problemas más difíciles de detectar. En su diseño, se incorporaron los últimos resultados en Ingeniería de Software orientado a objetos. Las bibliotecas estándar de Java también recurren con frecuencia a los patrones de diseño, que incrementan su flexibilidad. Java es Open Source (Software libre o Código abierto). Tiene licencia de software libre GPL.

Java ha sido probado, mejorado, ampliado y probado por una comunidad especializada de más de 6,5 millones de desarrolladores, la mayor y más activa del mundo. Gracias a su versatilidad, eficiencia y portabilidad, Java se ha convertido en un recurso inestimable [10].

J2EE (Java 2 Enterprise Edition)

Plataforma del lenguaje Java que provee una especificación de cómo debe construirse una aplicación empresarial. Esta especificación describe cómo los servidores de aplicación deben proporcionar seguridad, escalabilidad, portabilidad, consistencia, manejo transaccional robusto e independencia de la plataforma tanto de hardware como de sistemas operativos, y al mismo tiempo para que el desarrollador final pueda desplegar aplicaciones empresariales con menor esfuerzo.

Entorno de Desarrollo

Eclipse Galileo

Eclipse es un entorno de desarrollo integrado de código abierto y multiplataforma. Es desarrollado por la Fundación Eclipse, una organización independiente que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal, un IDE abierto y extensible para todo y nada en particular" [11].

Tiene además una comunidad de desarrolladores, realmente activa. Continuamente se están desarrollando nuevos plug-ins y revisando los anteriores. Al igual que en cualquier otro software, Eclipse permite la instalación de plug-ins destinados a mejorar las funcionalidades del propio IDE y a extenderse en cada vez más tecnologías. Eclipse Galileo es la versión 3.5 de este IDE presentada en el año 2009 [12].

Framework

Spring

Spring es un framework de aplicaciones Java/JEE desarrollado usando licencia de OpenSource. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable. Aspectos

fundamentales de Spring: Está diseñado desde el principio para ayudar a escribir código fácil de testear. Se integra con Hibernate, JDO e iBatis SQL Maps en términos de soporte a implementaciones DAO. Presenta una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control (IoC). El uso de un contenedor de Inversión de Control reduce grandemente la complejidad del código a interfaces, más que a clases. El objetivo central de Spring es permitir que objetos de negocio y de acceso a datos sean reutilizables, no atados a servicios JEE específicos. Spring proporciona un marco coherente para el acceso a datos, ya sea utilizando JDBC o una O/R de productos de mapeo como TopLink, Hibernate o JPA o aplicación JDO. Spring además puede eliminar la proliferación de Singletons visto en muchos proyectos. Spring proporciona una respuesta coherente, un modelo de programación simple en muchas áreas [13].

Spring Web Services (Spring-WS).

Spring Web Services (Spring-WS) es un producto de la comunidad de Spring centrada en la creación de documentos manejadores de servicios web. Tiene por objeto facilitar el contrato primero del servicio, lo que permite la creación flexible de servicios web. Esto constituye una buena práctica a la hora de desarrollar servicios web ya que permite un bajo acoplamiento entre el contrato y la implementación del servicio. El framework tiene soporte para varias tecnologías y APIs de manejo de contenidos de los mensajes XML (**E**xtensible **M**arkup **L**anguage). Brinda soporte al estándar Web Service Security permitiendo firmar mensajes SOAP (**S**imple **O**bject **A**ccess **P**rotocol), encriptar y desencriptar estos mensajes, así como la autenticación a través de los mismos. El producto se basa en Spring, lo que significa que pueden usarse los conceptos de Spring tales como la Inyección de Dependencia como parte integrante del servicio web. Spring-WS usa los contextos de aplicaciones de Spring para toda la configuración, lo que ayuda a los desarrolladores de Spring a desarrollar rápidamente. Además la arquitectura de Spring-WS se asemeja a la de Spring-MVC [14].

Spring Security.

Spring Security es la versión nueva de Acegi Security, la cual es una librería de seguridad para JEE. Proporciona una solución completa a la seguridad al implementar servicios de autenticación y autorización a recursos, tanto a nivel de peticiones Web como a nivel de invocación de métodos. Sobre

la base de Spring, Spring Security aprovecha al máximo las técnicas de Inyección de Dependencia y orientación a aspectos. Posee seguridad basada en roles y en listas de control de acceso [15].

Spring Security cuenta con una serie de ventajas: Es capaz de gestionar seguridad en varios niveles: URLs que se solicitan al servidor, acceso a métodos y clases Java, y acceso a instancias concretas de las clases. Permite separar la lógica de las aplicaciones del control de la seguridad, utilizando filtros para las peticiones al servidor de aplicaciones o aspectos para la seguridad en clases y métodos. La configuración de la seguridad es portable de un servidor a otro, ya que se encuentra dentro del WAR o el EAR de las aplicaciones. Soporta muchos modelos de identificación de los usuarios (HTTP BASIC, HTTP Digest, basada en formulario, LDAP, OpenID, JAAS y muchos más). Además se pueden ampliar estos mecanismos implementando clases que extiendan el modelo de Spring Security [15].

GWT

Google Web Toolkit (GWT) hace que sea aún más fácil diseñar una aplicación AJAX utilizando sólo el lenguaje de programación Java. Es un marco abierto de desarrollo de Java y su mejor característica es que no hay que preocuparse acerca de las incompatibilidades entre los navegadores Web y las plataformas. En GWT, se escribe el código en Java y luego el compilador lo traducirá en Java Script y HTML. Proporciona un marco de programación que es similar a la utilizada por los desarrolladores para crear aplicaciones Java, utilizando uno de los kits de herramientas GUI como el Swing, AWT, o SWT [16].

Por otra parte, hace que la reutilización de fragmentos de programa sea fácil. Esto reduce en gran medida el número de tecnologías diferentes que se necesita para el desarrollo puede usarse cualquier IDE de Java para escribir y depurar una aplicación AJAX GWT. GWT no es sólo una interfaz de programación; proporciona un conjunto de herramientas que permiten desarrollar funcionalidades Java Script de alto rendimiento en el navegador del cliente. Eso significa por ejemplo que se pueden poner los puntos de interrupción en el código y depurar a la perfección desde el lado del cliente para el servidor. Se puede desplegar las aplicaciones en cualquier contenedor de servlets, así crear y ejecutar pruebas de unidad [16].

GWT es utilizado por muchos productos de Google, como Google Wave y la nueva versión de AdWords. Es de código abierto, completamente libre, y utilizado por miles de desarrolladores de todo el mundo [17].

Como apéndice de esta herramienta surge Ext GWT, la cual es “una librería para la creación de aplicaciones de internet ricas con Google Web Toolkit (GWT). Incluye [18]:

- Alto rendimiento y personalización de la Interfaz de Usuario (UI) a través de Widgets.
- Un soporte completo de temas con Cascading Style Sheets (CSS) estándar.
- Bien diseñado, consistente y el código fuente completamente documentado.
- Solución de GWT Nativo, sin Java Script externo.
- Soporte completo a procedimientos remotos utilizando GWT RPC, JSON y XML.
- Soporte para características de Java 1.5, inclusiones genéricas, enumerativos y variables como argumentos.
- Licencias Comercial y de Código Abierto disponibles.

Apache Tomcat

Apache es un servidor de páginas web de código abierto multiplataforma y modular, se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Tomcat es un contenedor de Servlets con un entorno JSP. Puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad [19].

Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. (Ciberaula, 2006)

Características principales:

- Trabaja sobre múltiples plataformas (Unix, Linux, MacOSX, Vms, Win32, OS2, etc.).
- Incluye módulos que se cargan de forma dinámica.
- Soporta CGI, Perl, PHP.

-
- Soporte para Bases de datos.
 - Soporte SSL para transacciones seguras.
 - Incluye soporte para host virtuales.
 - Soporta HTTP 1.1.
 - Código Abierto.
 - Rápido.
 - Eficiente.

Ventajas:

- Ayuda en la mejora del posicionamiento.
- Este servidor junto con el módulo mod_rewrite puede convertirse en una herramienta muy jugosa para crear páginas con enlaces amigables para los buscadores.
- Es un software libre.
- Multiplataforma.
- Open Source.
- Modular.
- Extensible.
- Presenta mensajes de error altamente configurables.

Herramienta de Control de Versiones: SVN

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

Subversion maneja ficheros y directorios a través del tiempo. Permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A

cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Subversion es un sistema general que puede ser usado para administrar cualquier conjunto de ficheros [20].

Proporciona:

- Versionado de directorios.
- Verdadero historial de versiones.
- Envíos atómicos.
- Versionado de metadatos.
- Elección de las capas de red.
- Manipulación consistente de datos.
- Ramificación y etiquetado eficientes.
- Hackability.

Conclusiones

En este capítulo se detallaron las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que envuelven al mismo. Se seleccionó el lenguaje de programación, así como las tecnologías y herramientas informáticas de mayor efectividad para llevar a cabo el presente trabajo.

Capítulo 2: Características del Sistema.

Introducción.

En este capítulo permitirá conocer el contexto en el que se emplaza la solución propuesta. Se describirán los procesos existentes u observados con el objetivo de comprenderlos y se especificarán aquellos que soportará el sistema. Además guiará el desarrollo del software hacia el sistema adecuado, mediante la captura y descripción de los requerimientos funcionales y no funcionales del sistema.

Propuesta del Sistema.

La solución que se propone es el desarrollo de un Portal Web, el cual se espera que cuente con las siguientes características:

- Diseño atractivo, agradable, sugerente e intuitivo, con uso de animaciones e imágenes que atraigan la atención del usuario.
- Uso de elementos de navegación orientados al usuario y soporte para diferentes idiomas.
- Un sistema de afiliación a clientes.
- Sistema de filtrado de contenidos por compatibilidad con dispositivo móvil.
- Explorador y buscador avanzado de los contenidos disponibles organizados en categorías y sub-categorías, mostrando información detallada de los contenidos.
- Uso de técnicas de marketing digital a los contenidos.
- Mecanismo de descarga de contenidos por envío de SMS, con mecanismo de autorización para la seguridad en el envío de la petición.
- Prestación de servicios adicionales para brindar valor agregado al usuario.
- Utilización de la tecnología Ajax, para lograr un perfecto balance entre el rendimiento y el uso de elementos del diseño atractivos y atrayentes para el usuario.

Modelo de Dominio.

Se propone un modelo del dominio, ya que en el sistema los procesos del negocio no están bien definidos, no son visibles y las fronteras no están bien establecidas. Además permite de manera visual mostrar al

usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes, desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Ver figura 1.

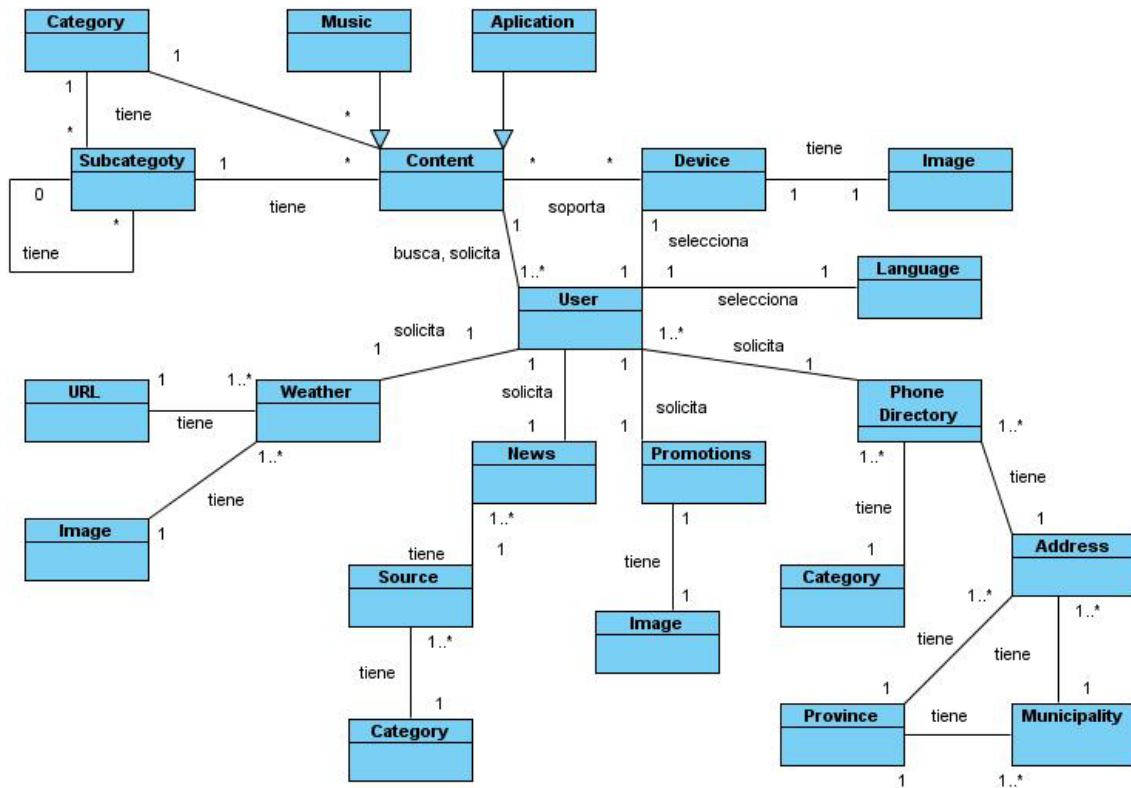


Figura 1. Diagrama de Clases del Modelo del Dominio

User: Representa al usuario final de Portal Web.

News: Representa los datos referentes a las Noticias.

Source: Representa los datos de una Fuente de Noticias de un Canal RSS.

News_Category: Representa la Categoría de una Noticia.

Promotions: Representa los datos referentes a las Promociones.

Promotions_Image: Representa la Imagen asociada a una Promoción determinada.

Weather: Representa los datos referentes al Estado del Tiempo.

URL: Representa la URL Fuente que proporciona el Estado del Tiempo de un Canal RSS.

Weather_Image: Representa la Imagen asociada a un Estado del Tiempo determinado.

Phone Directory: Representa los datos referentes a la dirección y número de teléfono de una entidad en el Directorio Telefónico de Páginas Amarillas.

Phone Directory_Category: Representa la Categoría de una Entrada Telefónica.

Address: Representa la Dirección de una Entrada Telefónica determinada.

Municipality: Representa el Municipio de la Dirección de una Entrada Telefónica determinada.

Province: Representa la Provincia de la Dirección de una Entrada Telefónica determinada.

Conten: Representa los Contenido que se gestionan.

Music: Representa los Contenidos de tipo Música.

Aplication: Representa los Contenidos de tipo aplicaciones.

Device: Representa el dispositivo móvil del usuario.

Device_Image: Representa la Imagen asociada a un Móvil determinado.

Language: Representa la agrupación de los textos de la aplicación en un lenguaje determinado.

Subcategory: Representa la Subcategoría de los Contenidos.

Content_Category: Representa la Categoría de los Contenidos.

Relación de los Requerimientos.

Los requisitos se pueden clasificar en: funcionales y no funcionales.

Requerimientos Funcionales (RF).

Los requisitos que se identificaron la solución del sistema fueron:

RF1. Registrar Usuario.

Permitirá a un visitante insertar su información de registro como un nuevo usuario.

RF2. Modificar Usuario.

Permitirá a un usuario modificar su información de registro.

RF3. Desactivar Usuario.

Permitirá a un usuario desactivar su registro.

RF4. Autenticar Usuario.

Permitirá a los usuarios registrados del portal autenticarse.

RF5. Des-autenticar Usuario.

Permitirá a un usuario autenticado en el portal des autenticarse.

RF6. Recuperar Contraseña.

Permitirá a un usuario recuperar su contraseña mediante el reenvío por correo electrónico.

RF7. Obtener Modelos de Móviles

Permitirá mostrar un listado con los modelos de móviles almacenados en la plataforma.

RF8. Obtener Modelos de Móviles por Marca

Permitirá mostrar un listado con los modelos de móviles de una marca determinada almacenados en la plataforma.

RF9. Seleccionar Móvil.

Permitirá a un usuario seleccionar un modelo de móvil para filtrar los contenidos compatibles con su móvil.

RF10. Buscar Directorio Telefónico por Nombre de la Entidad

Permitirá buscar el directorio telefónico dado el nombre de una entidad.

RF11. Buscar Directorio Telefónico por ID de Categoría.

Permitirá buscar el directorio telefónico dado un identificador de la categoría.

RF12. Mostrar Estado del Tiempo

Permitirá mostrar un resumen de la información referente al estado de tiempo.

RF13. Mostrar Detalles del Estado del Tiempo

Permitirá mostrar con detalles la información referente al estado de tiempo.

RF14. Mostrar Listado de las Noticias.

Permitirá mostrar un listado con los títulos de las noticias más recientes disponibles en la plataforma.

RF15. Mostrar Resumen de la Noticia por ID

Permitirá mostrar información resumida de una noticia según el identificador de la misma.

RF16. Mostrar Fuente de la Noticia

Permitirá ver la fuente original donde está publicada la noticia dada su URL.

RF17. Mostrar Promoción por ID.

Permitirá mostrar los datos de una promoción según el identificador de la misma.

RF18. Mostrar Siguiente Promoción.

Permitirá mostrar la siguiente promoción en la cola automáticamente.

RF19. Mostrar Contenidos por ID de Categoría.

Permitirá mostrar la lista de los contenidos según el identificador de una categoría.

RF20. Buscar Contenido por su Nombre e ID de Categoría.

Permitirá buscar contenidos dado el nombre o parte del nombre y el identificador de su categoría.

RF21. Mostrar Detalles del Contenido por ID.

Permitirá mostrar todos los datos de un contenido dado el identificador del mismo.

RF22. Descargar Contenido por ID.

Permitirá mostrar la información necesaria para realizar la descarga de un contenido según el identificador del mismo.

RF23. Cambiar Lenguaje.

Permitirá al usuario cambiar el lenguaje del portal a inglés o español según su preferencia.

RF24. Mostrar Contenidos por ID de Sub-categoría.

Permitirá mostrar la lista de los contenidos según el identificador de una sub-categoría.

RF25. Mostrar los Contenidos más Descargados.

Permitirá mostrar la lista de los contenidos más descargados por los usuarios.

RF26. Mostrar los Contenidos más Descargados por ID de Categoría.

Permitirá mostrar la lista de los contenidos más descargados por los usuarios según el identificador de una categoría.

RF27. Mostrar Siguiente Página de Últimos Contenidos por ID de Categoría.

Permitirá mostrar automáticamente la siguiente sub-lista de los últimos contenidos agregados a la plataforma según el identificador de una categoría.

RF28. Mostrar Siguiente Página de Últimos Contenidos.

Permitirá mostrar automáticamente la siguiente sub-lista de los últimos contenidos agregados a la plataforma.

RF29. Seleccionar Página de Últimos Contenidos.

Permitirá mostrar una sub-lista de los últimos contenidos incorporados a la plataforma.

RF30. Seleccionar Página de Últimos Contenidos por ID de Categoría.

Permitirá mostrar una sub-lista de los últimos contenidos incorporados a la plataforma según el identificador de una categoría.

Requerimientos No Funcionales.

Software:

En la PC destina para ser el servidor:

- Se necesita tener instalada la máquina virtual de Java (JVM).
- Se necesita tener instalado el servidor web Apache Tomcat 6.0.

Hardware:

- Se necesita como requerimientos mínimos una PC con procesador Intel Pentium IV que servirá como servidor.
- Requiere como mínimo que la PC destinada a ser servidor posea una memoria RAM de 1 GB.

Portabilidad:

- El sistema deberá ser portable en cualquier sistema operativo.

Fiabilidad:

- Disponibilidad: El sistema debe estar disponible a tiempo completo durante su vida útil.
- Confiabilidad: El sistema debe ser capaz de recuperarse ante la ocurrencia de un fallo, de no ser posible, emitir alertas al personal encargado de la administración del mismo, así como proteger la información y contenidos.

Usabilidad:

- Se deben tener conocimientos previos del manejo de portales web. La aplicación podrá ser manejada por cualquier usuario con conocimientos básicos del tema.
- Mensajes y Textos en la Interfaz: Tanto la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en el lenguaje seleccionado por el usuario para la aplicación y tener una apariencia estándar. Los mensajes de error deberán ser lo suficientemente informativos para dar a conocer la severidad del error. Estos mensajes no deben revelar información interna.
- Uso de elementos de navegación orientados al usuario y soporte para diferentes idiomas.
- Utilización de la tecnología Ajax, para lograr un perfecto balance entre el rendimiento y el uso de elementos del diseño atractivos y atrayentes para el usuario.

Apariencia o Interfaz Externa:

- Diseño atractivo, agradable, sugerente e intuitivo, con uso de animaciones e imágenes que atraigan la atención del usuario.

Interfaz Interna:

- La implementación de las funcionalidades debe seguir los lineamientos de código establecidos por la dirección del proyecto, para garantizar la comprensión del código por desarrolladores de futuras versiones.

Seguridad:

- La información de carácter confidencial manejada por la Plataforma de Contenidos y de sus usuarios deberá ser protegida de acceso no autorizado haciendo uso de mecanismos de autenticación y autorización que garanticen la protección de los datos que la requieran. La autenticación se realizará mediante el nombre de usuario y la contraseña: Las contraseñas deberán tener más de siete caracteres de longitud y tener una fortaleza media.

Modelo de Casos de Uso del Sistema.

Definición de los Actores.

- Visitante: representa al usuario final de la aplicación.
- Usuario: representa al Usuario autenticado.
- Reloj: representa el encargado de las funciones automatizadas.

Listado de Casos de Uso del Sistema.

Los casos de uso son artefactos narrativos que describen el comportamiento del sistema desde el punto de vista del usuario. Establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. A continuación se muestran los casos de uso del sistema identificados teniendo en cuenta los requisitos funcionales.

- Registrar usuario
- Gestionar Usuario
- Autenticar Usuario
- Des-autenticar Usuario

-
- Recuperar Contraseña
 - Obtener Modelos de Móviles
 - Obtener Modelos de Móviles por Marca
 - Seleccionar Móvil
 - Buscar Directorio Telefónico
 - Mostrar Estado del Tiempo
 - Mostrar Detalles del Estado del Tiempo
 - Mostrar Noticias
 - Mostrar Resumen de la Noticia
 - Mostrar Fuente de la Noticia
 - Mostrar Promoción
 - Mostrar Siguiente Promoción
 - Mostrar Contenidos por Categoría
 - Buscar Contenidos por Categoría
 - Mostrar Detalles de un Contenido
 - Descargar Contenido
 - Seleccionar Lenguaje
 - Mostrar Contenidos por Sub-categoría
 - Mostrar Contenidos más Descargados
 - Mostrar los Contenidos más Descargados por Categoría
 - Mostrar Siguiente Página de Últimos Contenidos por Categoría
 - Mostrar Siguiente Página de Últimos Contenidos
 - Seleccionar Página de Últimos Contenidos
 - Seleccionar Página de Últimos Contenidos por Categoría

La descripción detallada de los Casos de Uso se encuentra en el Anexo 1: Descripción Textual de los Casos de Uso

Diagrama de Casos de Uso del Sistema.

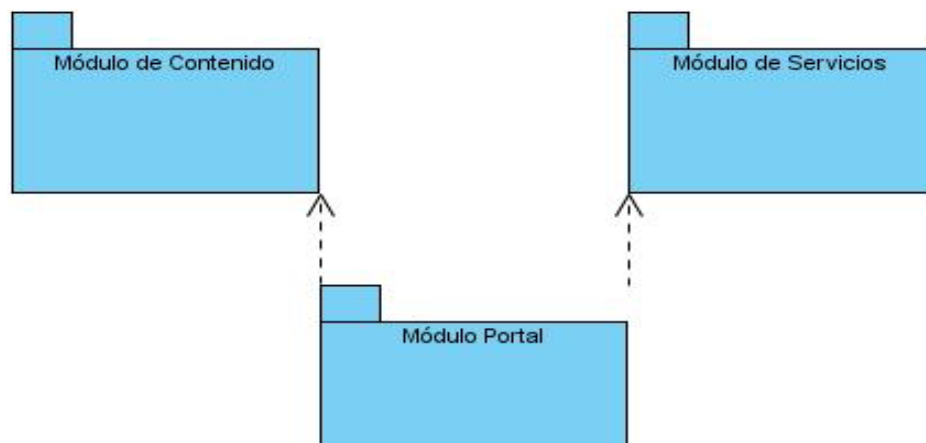


Figura 2. Diagrama de Paquetes de Casos de Uso del Sistema

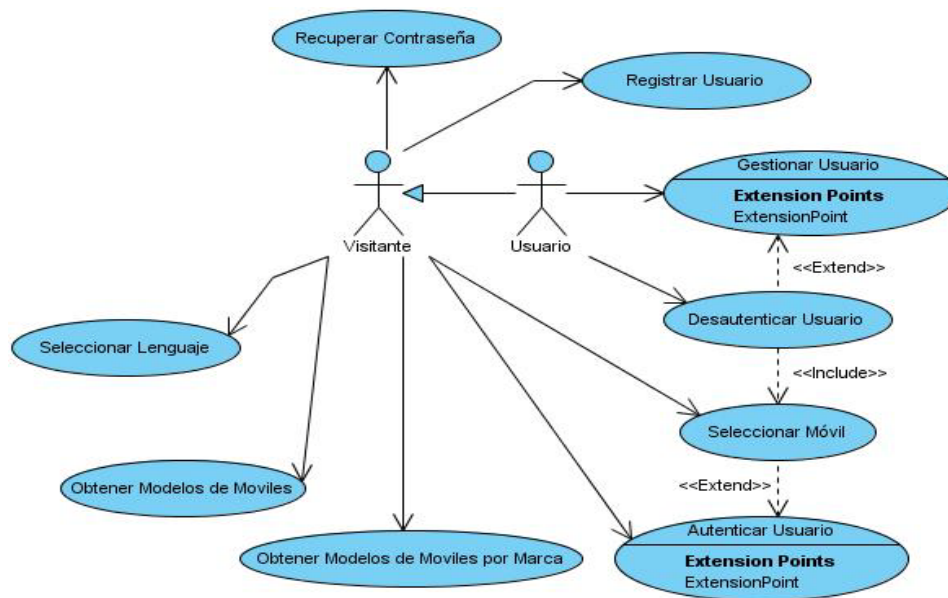


Figura 3. Diagrama de CU del Sistema Módulo Portal

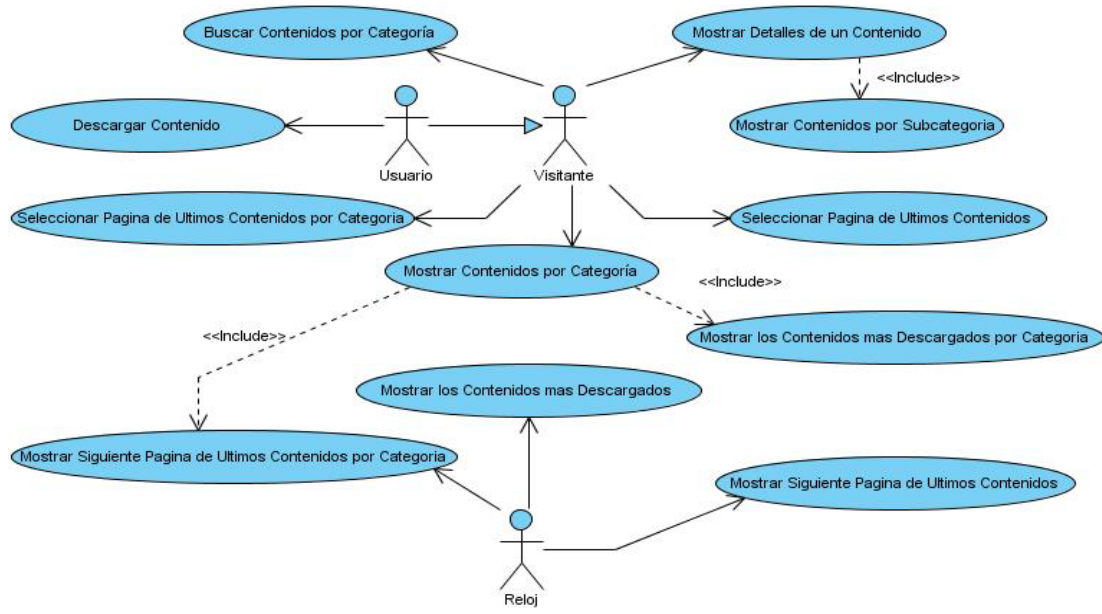


Figura 4. Diagrama de CU del Sistema Módulo Contenidos

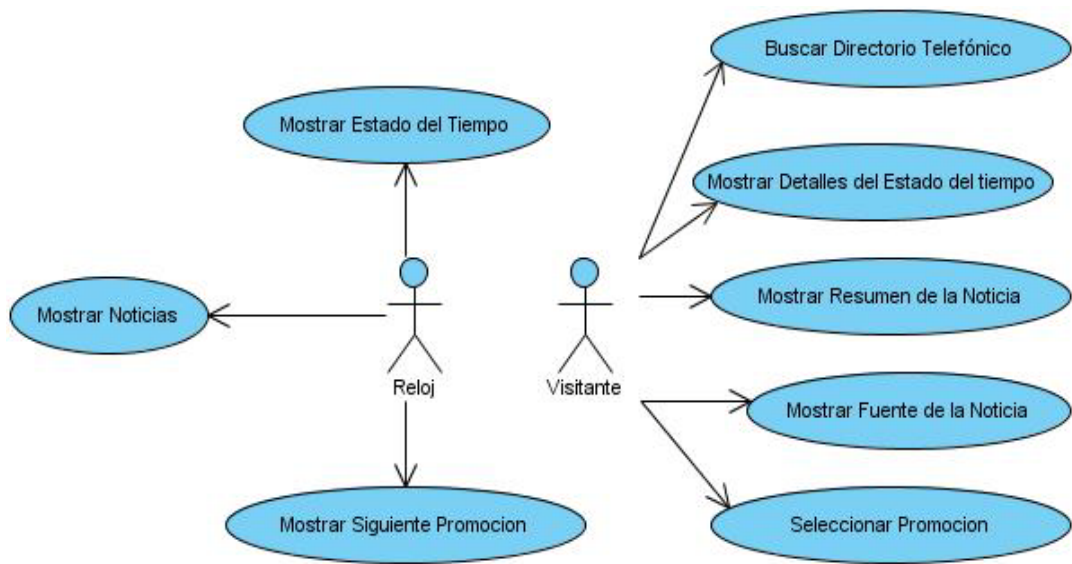


Figura 5. Diagrama de CU del Sistema Módulo Servicios

Conclusiones.

En este capítulo se realizó una descripción de la solución propuesta, se representó el modelo de dominio con los principales objetos del sistema, así como los requisitos funcionales y no funcionales. Se realizó el diagrama de casos de uso del sistema, así como la descripción detallada de estos. Los casos de usos identificados dirigirán el trabajo a lo largo del resto de los flujos de trabajos propuestos por la metodología de desarrollo de software utilizada, por lo tanto serán los encargados de enlazar directamente los diferentes flujos de trabajo.

Capítulo 3: Diseño.

Introducción.

En este capítulo se describen los procesos que se llevan a cabo en el flujo de trabajo de diseño. Se organizan los diagramas que describen el análisis y diseño de las clases, así como los que representan la interacción entre objetos, siguiendo el criterio de distribución por caso de uso.

Diseño del Sistema.

Definición del Modelo del Diseño.

En el diseño se modela el sistema de manera que se cumplan los requisitos funcionales y no funcionales. Dentro de los propósitos del diseño están: adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. [21]

Diagramas de Clases del Diseño.

Los Diagramas de Diseño se encuentran en el Anexo 3: Diagramas de las Clases del Diseño.

Diagramas de Interacción (Secuencia).

Los Diagramas de Secuencia se encuentran en el Anexo 4: Diagramas de Secuencia por Caso de Usos.

Descripción de la Arquitectura.

Patrones Utilizados.

Un patrón define una solución a un problema en un contexto determinado e indica cómo puede ser utilizado en diferentes circunstancias. Pueden organizarse por categorías según la escala o nivel de abstracción entre las que pueden mencionarse:

-
- **Patrones arquitecturales:** Aquellos que expresan un esquema organizativo estructural fundamental para sistemas software.
 - **Patrones de diseño:** Aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software.

Patrón Arquitectónico.

Patrón Modelo-Vista-Controlador

Modelo Vista Controlador (MVC), en la mayoría de las aplicaciones web la lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Con el modelo vista controlador se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Elementos del patrón:

- Modelo: Datos y reglas de negocio
- Vista: Muestra la información del modelo al usuario
- Controlador: Gestiona las entradas del usuario

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador.

1. El modelo es el responsable de:
 - Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
 - Definir las reglas de negocio (la funcionalidad del sistema).
 - Llevar un registro de las vistas y controladores del sistema.
2. El controlador es responsable de:
 - Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
 - Contener reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar.
3. Las vistas son responsables de:

-
- Recibir datos del modelo y mostrarlos al usuario.
 - Tener un registro de su controlador asociado.
 - Poder dar el servicio de actualizar, para que sea invocado por el controlador o por el modelo.
- [22]

Este patrón se utilizó para desacoplar la lógica de la vista con respecto a la de modelo y minimizar el impacto de los cambios entre la interfaz y el negocio; donde las clases vistas son las View, las controladoras son los Controller y las clases principales del modelo son los Manager.

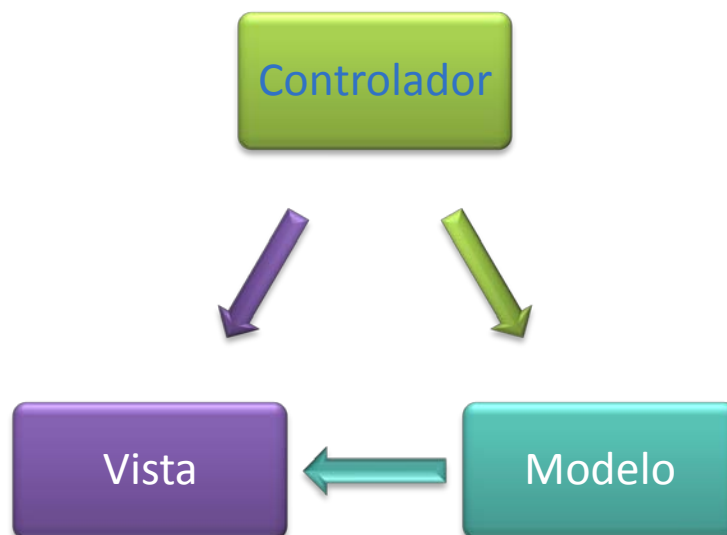


Figura 6: Diagrama de Paquetes. Vista Arquitectónica.

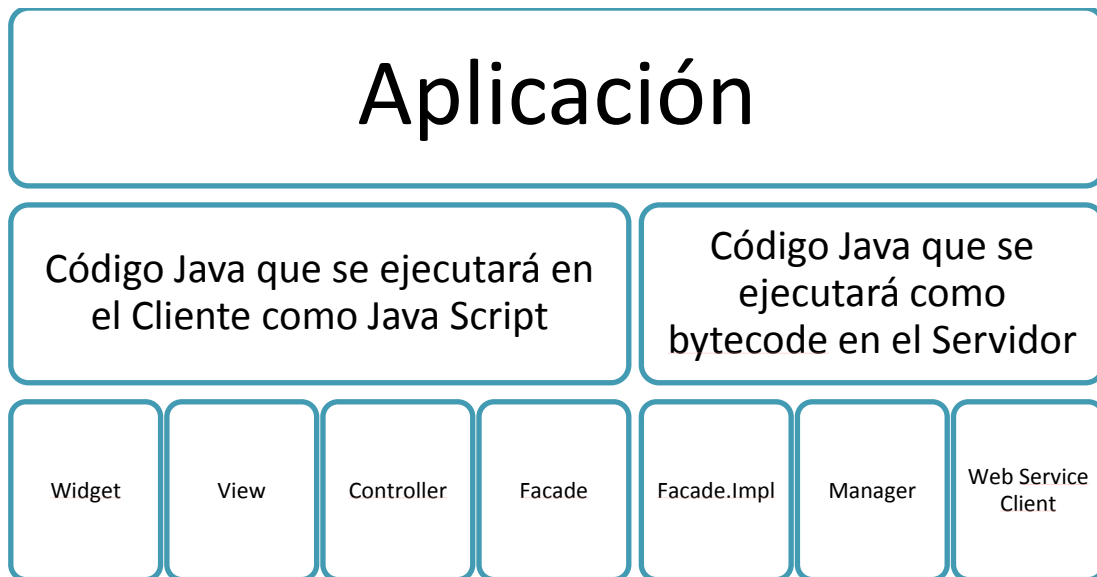


Figura 7: Diagrama de Arquitectura del Sistema.

Patrones de Diseño.

Son directrices y principios estructurados que describen un problema común y entregan una buena solución ya probada a la que le dan un nombre. Estos ayudan a diseñar correctamente en menos tiempo, ayudan a construir problemas reutilizables y facilitan la documentación. [23]

➤ Patrones GRASP (General Responsibility Assignment Software Patterns)

Los patrones GRASP permiten describir los principios fundamentales de asignación de responsabilidades a objetos. Estos patrones son esenciales en el diseño eficaz de un software. Por las características de estos patrones, se consideró necesaria su utilización en este trabajo para obtener un producto de mayor calidad.

❖ Controlador

Este patrón GRASP establece la asignación de la responsabilidad de manejar los eventos a una clase que será la encargada de controlarlos. Permite la reutilización de componentes al delegar las responsabilidades en una clase, lo que facilita que aunque existan cambios desde el punto de vista de la

interfaz del sistema las funcionalidades se mantengan intactas y puedan ser reutilizadas. Aplicado en el sistema, en la aplicación web este patrón se ve reflejado en las clases controladoras, las cuales son las encargadas de manejar las peticiones del cliente, en el caso de los servicios web los controladores serían los Controller, clases encargadas de manejar los eventos lanzados en la aplicación, obtener los datos del modelo y entregarlos a las clases View que los mostrarán al usuario final.

❖ **Experto**

Este patrón de diseño define asignar la responsabilidad a la clase que cuente con la información necesaria para manejar esta responsabilidad. Es necesario que la asignación de responsabilidades a las clases se lleve a cabo de una forma adecuada, lo que permitirá que el sistema sea más fácil de entender así como que se puedan reutilizar componentes. Con su aplicación se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. El comportamiento se distribuye entre las clases que cuentan con la información requerida permitiendo la creación de clases sencillas y más cohesivas. Este patrón se ve reflejado en las clases del negocio como los Manager y WebServiceClient, donde cada una realiza funciones específicas.

❖ **Alta Cohesión**

Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un gran trabajo. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. En el sistema se tiene bien definido las responsabilidades de cada clase, por ejemplo, se definen clases del negocio por módulo como los Manager, para evitar la sobrecarga de funcionalidades en estas clases.

❖ **Bajo Acoplamiento**

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Un alto acoplamiento tendría como consecuencia que las clases sean más difíciles de entender cuando estén aisladas, más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen. Por lo que un bajo acoplamiento plantea que debe haber pocas dependencias entre las clases. Spring plantea el uso de interfaces para evitar un alto acoplamiento, por lo que en el sistema este patrón se ve claramente reflejado, ya que tanto la capa de negocio como la capa

de acceso a datos tienen interfaces que son implementadas por otras clases. [23] En el sistema en cuestión las clases Facade contienen interfaces que son implementadas por los Manager y a su vez estos contienen interfaces implementadas por los WebServiceClient.

❖ Patrón Inversión de Control (IoC) / Inyección de Dependencia (DI)

Inversión de Control es un principio que generalmente siguen los framework, en lugar de tener las clases de la aplicación el control sobre la creación de objetos, le asigna esta responsabilidad a un agente externo como un contenedor. Spring proporciona un contenedor que maneja el ciclo de vida de los objetos y resuelve las dependencias entre dichos objetos. Se suelen confundir los términos de Inversión de Control e Inyección de Dependencia, pero no son lo mismo. Mientras que la Inversión de Control es un principio de diseño general, la Inyección de Dependencia es un patrón de diseño concreto que encarna este principio. La Inyección de Dependencia es la más típica realización de la Inversión de Control.

El patrón Inyección de Dependencia, es una técnica que busca facilitar la resolución de dependencias entre objetos. Consiste en inyectar objetos a una clase en lugar de ser la propia clase quien cree el objeto. La forma habitual de implementar este patrón es mediante un contenedor. El contenedor inyecta a cada objeto los objetos necesarios según las relaciones plasmadas en un fichero de configuración. Típicamente este contenedor es implementado por un framework externo a la aplicación (como Spring). El uso de este patrón permite construir aplicaciones mucho más flexibles y robustas. [24]

Se hace uso de este patrón utilizando el framework externo Spring a la hora de inyectar las clases que implementan las interfaces incluidas en los Facade y Manager.

❖ Patrón Fachada (Facade Pattern)

Este patrón de diseño tiene como objetivo proporcionar una interfaz unificada para un conjunto de interfaces en un subsistema, haciéndolo más fácil de usar. Además oculta a los clientes de la complejidad del subsistema y lo hace más fácil de usar, favorece un acoplamiento débil entre el subsistema y sus clientes, consiguiendo que los cambios de las clases del sistema sean transparentes a los clientes, facilita la división en capas y reduce dependencias de compilación, y no impide el acceso a las clases del sistema.

Este patrón de refleja en el uso de las clases Facade que proporcionan una interfaz entre los Controller y los Manager facilitando la comunicación entre la capa de Presentación y la de Negocio.

Conclusiones.

En este capítulo se expusieron los principales artefactos del diseño del sistema propuesto como solución, en específico los diagramas de clases del diseño y los diagramas de interacción. Además se describieron los patrones utilizados.

Capítulo 4: Implementación.

Introducción.

En este capítulo se lleva a cabo la descripción de las principales tareas del flujo de implementación y de prueba, además de los artefactos que se generan en estos flujos.

También se describirá cómo se relacionan y organizan los componentes de acuerdo con los mecanismos de estructuración y modularización en el entorno de implementación y en el lenguaje de programación utilizado. Se indicará la situación física de los componentes lógicos desarrollados en el Diagrama de Despliegue.

Modelo de Implementación.

Diagrama de Despliegue.

El modelo de despliegue se crea durante las últimas actividades del flujo de trabajo de diseño y provee la descripción de la distribución física del sistema. Se utiliza como entrada fundamental en las actividades de diseño e implementación. El diagrama de componentes, por su parte, estructura el modelo de implementación y muestra las relaciones entre los elementos de implementación. Ambos diagramas conforman el modelo de implementación, que describe como los elementos del diseño se implementan en términos de componentes, ficheros de código fuente, ejecutables, etc.

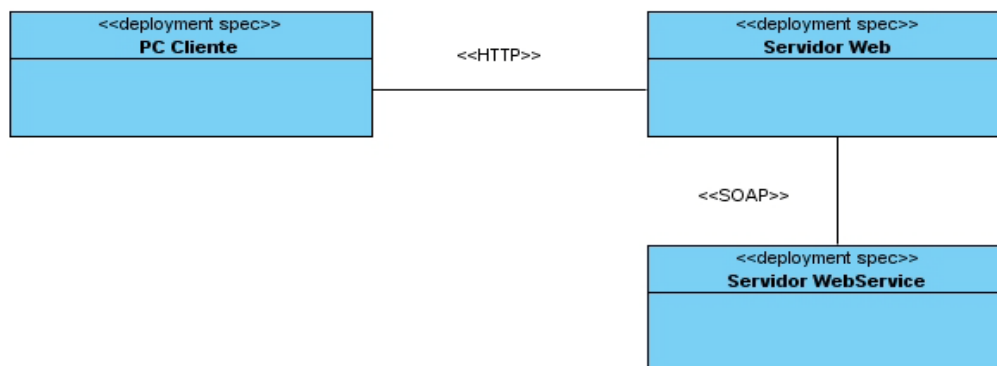


Figura 8: Diagrama de Despliegue.

Diagrama de Componentes.

Los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema. Muestran tanto los componentes de software (código fuente, binario y ejecutable) como las relaciones lógicas entre ellos en un sistema. Y como todos los diagramas, también pueden contener paquetes utilizados para agrupar elementos del modelo.

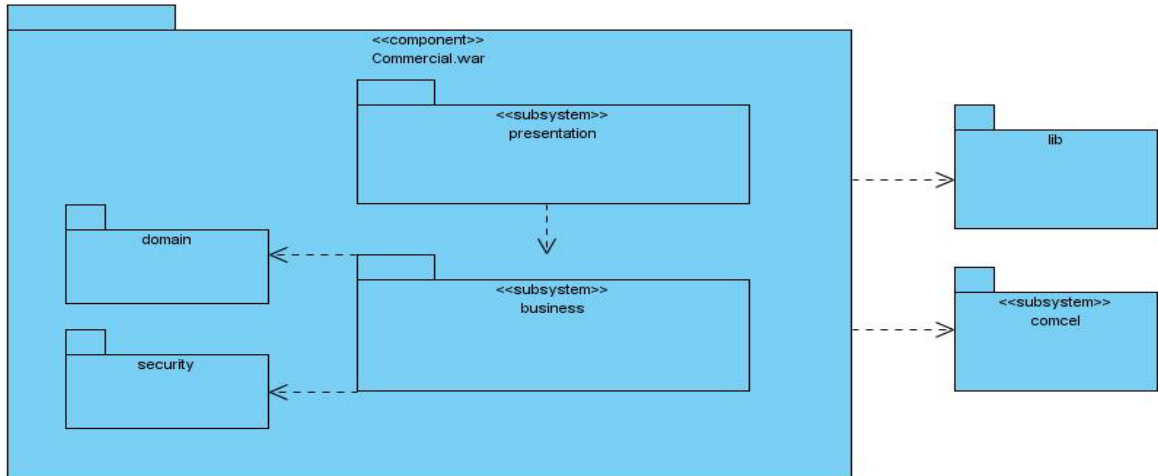


Figura 9: Diagrama de Componentes General.

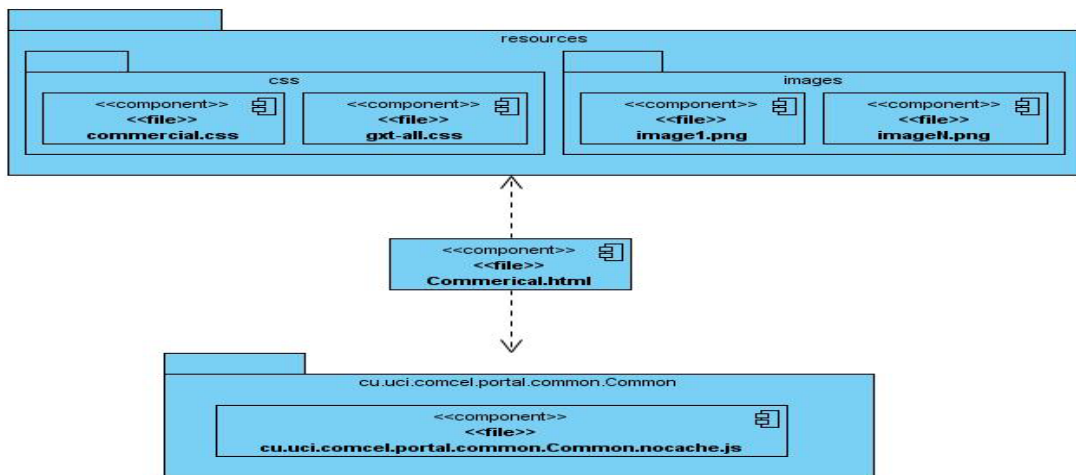


Figura 10: Diagrama de Componentes. Capa Presentación.

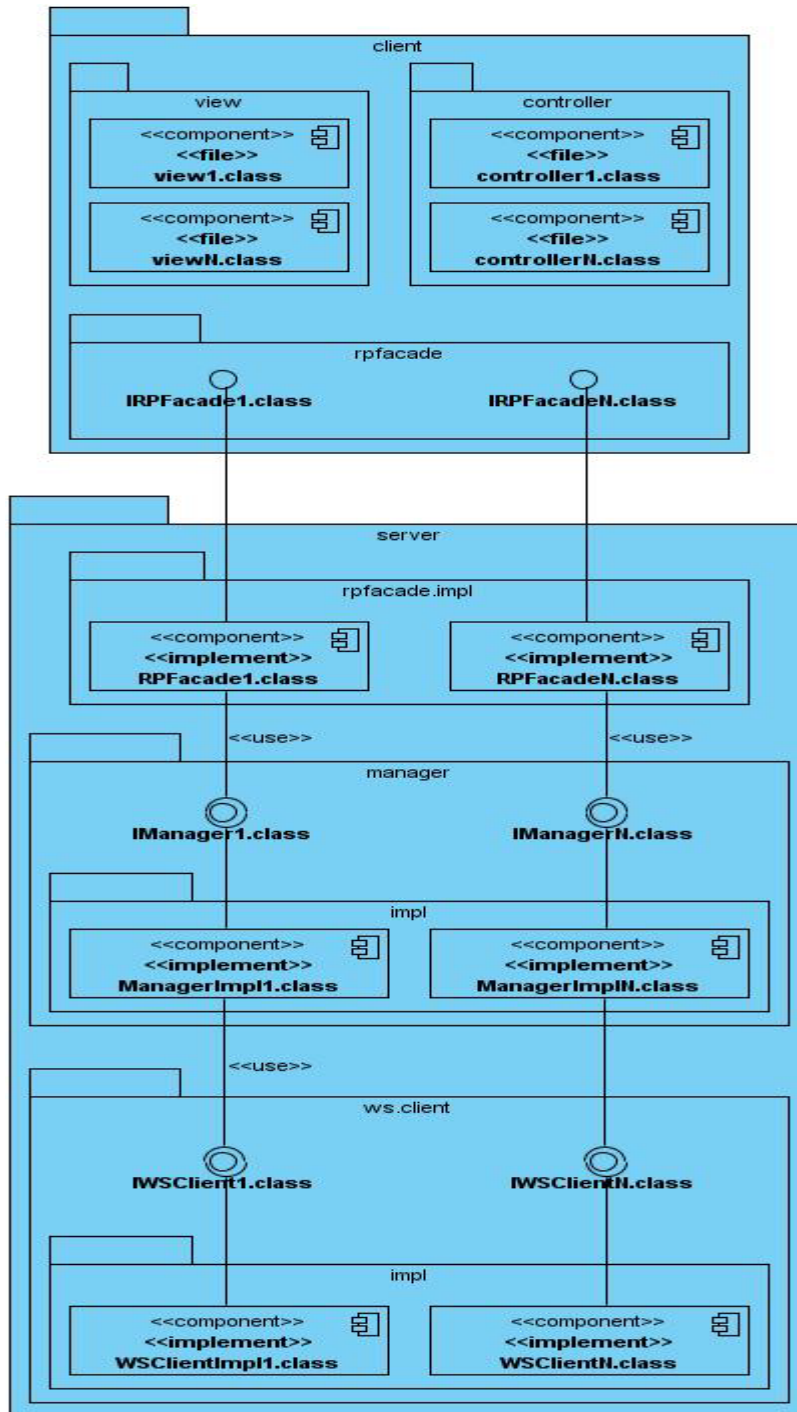


Figura 11: Diagrama de Componentes. Capa Negocio.

Conclusiones.

Se finaliza la etapa de Implementación del sistema, haciéndose los modelos de despliegue y los diagramas de componentes. También se concluye así con la etapa de prueba del sistema.

El propósito fundamental de la implementación es desarrollar la arquitectura del sistema capturada durante el diseño. En esta fase no solo se distribuyeron los componentes y se asignaron a nodos en el diagrama de despliegue, también se implementaron las clases y subsistemas encontrados durante el diseño.

Capítulo 5: Estudio de Factibilidad.

Introducción

Para el estudio de la factibilidad existen varios métodos: COCOMO, COCOMO II, Análisis de Punto de función, Análisis de Puntos de Casos de Uso, entre otros.

La estimación por Puntos de Caso de Uso resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros Casos de Uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational. En éste tipo de aproximación, los primeros Casos de Uso a desarrollar son los que ejercitan la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos. Además permite estimar el proyecto antes de empezar la etapa de implementación. [25]

En este capítulo se realizará mediante el método de Puntos de Caso de Uso una estimación del esfuerzo, costo y tiempo de desarrollo del proyecto. Además se hará un estudio de los costos y beneficios del mismo para llegar a la conclusión si el sistema es factible o no.

Método de Estimación “Puntos de Casos de Uso”.

Definición y Conceptos fundamentales.

La estimación mediante el análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. El método utiliza los actores y casos de uso identificados para calcular el esfuerzo que costará desarrollarlos. A los casos de uso se les asigna una complejidad basada en transacciones, de los escenarios de los casos de uso. A los actores se les asigna una complejidad basada en el tipo de actor, es decir, si son interfaces con usuarios o si son interfaces con otros sistemas. También se utilizan factores de entorno y de complejidad técnica para afinar el resultado.

Cálculo de Puntos de Casos de Uso.

El cálculo de los Puntos de Caso de Uso sin ajustar constituye el primer paso y se calcula a partir de la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Donde:

- **UUCP:** Puntos de Casos de Uso sin Ajustar.
- **UAW:** Factor de Peso de los Actores sin Ajustar.
- **UUCW:** Factor de Peso de los Casos de Uso sin Ajustar.

El Factor de Peso de los Actores sin Ajustar (**UAW**) se calcula analizando la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. Los criterios para evaluar la complejidad pueden ser los siguientes:

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0	0
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2	6
Total				6

Para obtener el valor del Factor de Peso de los Casos de Uso sin Ajustar (**UUCW**) se debe tener en cuenta la cantidad de casos de uso y la complejidad de cada uno de ellos. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Peso	Cantidad	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	23	115

Medio	El caso de uso tiene de 4 a 7 transacciones.	10	4	40
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
Total				155

Sustituyendo las ecuaciones 2 y 3 en la ecuación 1:

$$\mathbf{UUCP = UAW + UUCW}$$

$$\mathbf{UUCP = 6 + 155}$$

$$\mathbf{UUCP = 161}$$

Luego de calcular los Puntos de Casos de Uso sin Ajustar (**UUCP**), se hace necesario ajustar este valor multiplicándolo por el Factor de Complejidad Técnica (**TCF**) y el Factor de Ambiente (**EF**), obteniendo como resultado los Puntos de Casos de Uso Ajustados (**UCP**) .

$$\mathbf{UCP = UUCP * TCF * EF [4]}$$

El Factor de Complejidad Técnica (**TCF**) es un coeficiente que se calcula cuantificando un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de estos factores se cuantifica con un valor de 0 a 5, donde 0 significa irrelevante y 5 un aporte muy importante.

- 0: No presente o sin influencia.
- 1: Influencia incidental o presencia incidental.
- 2: Influencia moderada o presencia moderada.
- 3: Influencia media o presencia media.
- 4: Influencia significativa o presencia significativa.
- 5: Fuerte influencia o fuerte presencia.

Factor	Descripción	Peso	Valor Asignado	Total
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	3	3
T4	Funcionamiento Interno complejo	1	2	2
T5	El código debe ser reutilizable	1	0	0

T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	2	2
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	1	1
T12	Provee acceso directo a terceras partes	1	4	4
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
Total				41

El Factor de Complejidad Técnica, se calcula utilizando la ecuación:

$$\text{TCF} = 0.6 + 0.01 * \Sigma(\text{pesoi} * \text{valor asignadoi}) [5]$$

Donde pesoi representa el peso del factor i y valor asignadoi el valor que se le asignó según el aporte del mismo al sistema.

$$\text{TCF} = 0.6 + 0.01 * \Sigma(\text{pesoi} * \text{valor asignadoi})$$

$$\text{TCF} = 0.6 + 0.01 * 41$$

$$\text{TCF} = 1.01$$

El Factor de Ambiente (EF) se calcula similar al Factor de Complejidad Técnica. Los factores que se cuantifican están relacionados con las habilidades y el entrenamiento del grupo involucrado en el desarrollo del sistema.

La siguiente tabla muestra los factores cuantificados y el valor asignado a cada uno.

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	5	7.5
E2	Experiencia en la aplicación.	0.5	3	1.5
E3	Experiencia en la orientación a objeto.	1	5	5
E4	Capacidad del analista líder.	0.5	2	3
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos.	2	3	6
E7	Personal Part-Time.	-1	3	-3
E8	Dificultad del lenguaje de programación.	-1	5	-5

Total	18
--------------	-----------

- Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).
- Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

El Factor de Ambiente (**EF**) se calcula mediante la siguiente ecuación:

$$\mathbf{EF = 1.4 - 0.03 \times \Sigma (\text{pesoi x valor asignadoi}) [6]}$$

Donde pesoi representa el peso del factor i y valor asignadoi el valor que se le asignó según el aporte del mismo al sistema.

$$\mathbf{EF = 1.4 - 0.03 * \Sigma (\text{pesoi x valor asignadoi})}$$

$$\mathbf{EF = 1.4 - 0.03 * 18}$$

$$\mathbf{EF = 0.86}$$

Sustituyendo las ecuaciones:

$$\mathbf{UCP = UUCP * TCF * EF}$$

$$\mathbf{UCP = 161 * 1.01 * 0.86}$$

$$\mathbf{UCP = 139.85}$$

Estimación del Esfuerzo.

El esfuerzo en horas - hombre viene dado por:

$$\mathbf{E = UCP * CF}$$

Donde:

- E: Esfuerzo estimado en horas-hombre.
- UCP: Puntos de Casos de Uso ajustados.
- CF: Factor de conversión.

CF = 20 horas-hombre (si Total EF \leq 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Total EF = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Como: Total EF = 1+1

Total EF = 2

Por tanto:

CF = 20 horas-hombre (porque Total EF \leq)

Luego E = 139.85 * 20 horas-hombre

E = 2797 horas-hombre

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se puede tener en cuenta el siguiente criterio:

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	699.25
Diseño	20	1398.5
Implementación	40	2797
Pruebas	15	1048.875
Sobrecarga (otras actividades)	15	1048.875
Total	100	6992.5

Tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

El Esfuerzo Total sería 6992.5 horas-hombre, si estimamos teniendo en cuenta que un mes tiene 192 horas laborables, pues se trabajan 8 horas diarias 24 días al mes, entonces el Esfuerzo Total en mes-hombre sería 36.42 mes-hombre.

Análisis y Estimación del Costo.

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH}$$

Donde:

- CH: Cantidad de hombres.
- CHM: Costo Hombre – Mes.
- ET: Esfuerzo Total.

Si la Cantidad de hombres es 3 y se tiene un Salario Promedio mensual igual a \$100.00. Entonces:

$$\text{CHM} = \text{CH} * \text{Salario Promedio}$$

$$\text{CHM} = 3 * 100$$

$$\text{CHM} = 300.00$$

Luego:

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH}$$

$$\text{Costo} = 300.00 * 36.42 / 3$$

$$\text{Costo} = \$ 3642$$

Cálculo del Tiempo de Desarrollo de todo el Proyecto.

A partir de los datos conocidos anteriormente se puede decir que una persona puede realizar el trabajo en un tiempo aproximado de 36.42 meses. Pero teniendo en cuenta que la cantidad de hombres para la realización del Portal Comercial de Contenidos y Servicios es 3, el tiempo va a ser:

- Tiempo = ET / CH
- Tiempo = 36.42 / 3
- Tiempo = 12.14 ≈ 12 meses

De los resultados obtenidos se interpreta que con 3 hombres trabajando en el proyecto, el mismo se desarrolla en 12 meses y su costo total se estima que sea \$ 3642.

Conclusiones.

En este capítulo se llevó a cabo un estudio de la factibilidad del proyecto teniendo en cuenta el tiempo de duración, el posible costo y los posibles beneficios que podrá brindar. La estimación permite determinar si el producto o el proyecto son viables. Con esta información se puede estimar no sólo los recursos humanos, materiales, etc. sino también las fechas de entrega y culminación del proyecto. Se pueden planificar además las actividades de los desarrolladores.

Conclusiones.

Con el objetivo de darle cumplimiento a los objetivos generales y específicos de este trabajo se han llevado a cabo una serie de tareas que permitieron el buen desarrollo del sistema. Se llevó a cabo una previa investigación sobre el comercio electrónico y los Contenidos Digitales para la Telefonía Móvil, reflejándose la situación actual de estas tecnologías en el estado del arte. Se realizaron varios estudios para elegir las tecnologías, lenguaje de programación y metodología de desarrollo a utilizar donde se escogieron las que resultaron más adecuadas para el desarrollo del sistema. Se utilizó el lenguaje Java y como metodología RUP, para documentar el proceso de desarrollo del software y una arquitectura en capas utilizando el patrón modelo-vista-controlador. Se utilizaron diferentes patrones de diseño que permitieron desarrollar un producto con calidad.

Como resultado de este trabajo se desarrolló una aplicación web cumpliendo con los requerimientos establecidos inicialmente. Por lo anteriormente dicho se concluye que se vencieron con éxitos los objetivos propuestos.

Recomendaciones.

Referencias Bibliográficas.

1. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Addison Wesley 2000. Capítulo 1.
2. **Ayuda de la herramienta Case Rational Rose.** 2008.
3. **Molpeceres, Alberto.** *Procesos de Desarrollo RUP, SP y FDD.* Alemania: AT javahispano DOT org, 2002.
4. **Larman, Craig.** *UML y Patrones.* La Habana: Editorial Felix Varela, 2004.
5. **Colectivo de Autores.** *Visual Paradigm for UML User's Guide.* 2007.
6. **Java.** *Breve historia de la tecnología Java.* [En línea] <http://www.java.com/es/about/>.
7. **Cruz Martínez-Lacaci, Fernando.** *El código abierto en el ámbito empresarial.* [En línea] 2007. <http://www.mkm-pi.com/mkmpi.php?article56>.
8. **Qualitrain.** *¿Por qué seleccionar java como tecnología de desarrollo? Qualitrain México.* [En línea] Empresa especializada en capacitación., 2009. <http://www.qualitrain.com.mx/Blog.html>
9. **Iborra Baeza, Ignacio y Lozano Ortega, Miguel Angel.** *Presentación del curso "Java y Dispositivos Móviles".* Universidad de Alicante. [En línea] 2009. <http://www.jtech.ua.es/jdm/present.htm>.
10. **Java.** [En línea] 2010. <http://www.java.com/es/about/>
11. **Foundation, E.** Eclipse. [En línea] 2009. <http://www.eclipse.org/>.
12. **Colectivo de Autores.** Help - Eclipse SDK. *eclipse.* [En línea] 2008. <http://help.eclipse.org/galileo/index.jsp>.
13. **Jonhson, Rod.** *Introduction to the Spring Framework.* [En línea] 2007. <http://www.theserverside.com/tt/articles/article.tss?!=IntrotoSpring25>
14. **Poutsma, A.E., Rick y Abed, Tareq.** *Spring Web Services – Reference.* 2005-2007.
15. **Spring Security.** [En línea] <http://www.acegisecurity.org>.
16. **Chaganti, Prabhakar.** *Birmingham, UK: Packt Publishing Ltd.* *Google Web Toolkit GWT Java AJAX Programming.*[En línea] 2007. <http://assets.devx.com>
17. **GWT.** [En línea]. <http://code.google.com/webtoolkit>

-
18. **Ext GWT - Java Component Library**'. [En línea]. 2010. <http://www.extjs.com/products/gxt/>
 19. **Apache.** *Tomcat* – *Introducción.* [En línea]. http://www.programacion.com/tutorial/tomcatintro/1/#1_intro.
 20. **Collins-Sussman, Ben, W. Fitzpatrick, Brian y Pilato, C. Michael.** *Control de versiones con Subversion.* [En línea]. <http://svnbook.red-bean.com/nightly/es/index.html>
 21. **Booch Grady, Jacobson Ivar, Rumbaugh James.** El Proceso Unificado de Desarrollo de Software. Addison Wesley 2000. Capítulo 9 Páginas 205-210.
 22. **Burbeck, Steve.** Application programming in Smalltalk-80: How to use Model-View-Controller (MVC). 2007.
 23. **Gamma, J.M.V.R.H.R.J.E.** Design Patterns: Elements of Reusable Object-Oriented Software. 1995.
 24. **Breidenbach, C.W.w.R.** Spring in Action.
 25. **Thomas Fihlman.** *Estimation in software development projects.*

Bibliografía.

Fleming, Paul y Alberdi, Maria Jose . Hablemos de Marketing Interactivo, Reflexiones sobre marketing digital y comercio electrónico. [En línea]. 2000. http://www.google.com/books?hl=es&lr=&id=Fj-I5ZXro_sC&oi=fnd&pg=PA17&dq=Marketing+Digital&ots=pQx2puTmpO&sig=qxCh1z2Q51So5pkP_xv0CWeOYIq#v=onepage&q&f=false

Aguado, Juan Miguel y Martínez, Inmaculada J. El proceso de mediatización de la telefonía móvil: de la interacción al consumo cultural.[En línea] 2006. http://ehu.es/zer/zer20/zer20_19_aguado.pdf

Fernández Acebal, Cesar. Universidad de Oviedo. Tema 3, Modelos de negocio, Comercio Electrónico. [En línea] 2006. http://eva.uci.cu/file.php/447/Tema_II/Lectura_Tema_II/Modelos_de_Negocio_Electronico.pdf

Improven Consultores. Negocios en Internet y el comercio electrónico. [En línea]. http://eva.uci.cu/file.php/447/Tema_II/Lectura_Tema_II/Negocios_en_Internet.pdf

IBM Global Services. Fomentar la fidelización del cliente en el mercado electrónico. [En línea] 2006. http://eva.uci.cu/file.php/447/Tema_III/Lecturas_Complementarias_del_Tema_III/Fidelizacion.pdf

¿Qué es la Telemática? [En línea]. <http://schubert.upc.es/upc/epsc/WEPSCCas.nsf/67895027ae2f5cd8c12569210041e599/aac69b0802c351cec12569b9003b40e3?OpenDocument>

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.* Addison Wesley 2000. Capítulo 1.

Ayuda de la herramienta Case Rational Rose. 2008.

Molpeceres, Alberto. *Procesos de Desarrollo RUP, SP y FDD.* Alemania: AT javahispano DOT org, 2002.

Larman, Craig. *UML y Patrones.* La Habana: Editorial Felix Varela, 2004.

Foundation, E. Eclipse. [En línea] 2009. <http://www.eclipse.org/>.

Java. [En línea]. 2010. <http://www.java.com/>

Spring Source. [En línea] 2010. <http://www.springsource.org/>

Jonhson, Rod. *Introduction to the Spring Framework.* [En línea] 2007. <http://www.theserverside.com/tt/articles/article.tss?!=IntrotoSpring25>

Spring Security. [En línea] <http://www.acegisecurity.org>.

Universidad de Sevilla. [En línea] 2010. <http://www.lsi.us.es/docencia/get.php?id=1310>

Apache Software Foundation. The Apache Software Foundation. [En línea] 2009. <http://tomcat.apache.org/index.html>.

Apache. Tomcat – Introducción. [En línea]. http://www.programacion.com/tutorial/tomcatintro/1/#1_intro.

Colectivo de Autores. Help - Eclipse SDK. *eclipse.* [En línea] 2008. <http://help.eclipse.org/ganymede/index.jsp>.

Colectivo de Autores. Visual Paradigm for UML User's Guide. 2007.

GWT. [En línea]. <http://code.google.com/webtoolkit>

Iborra Baeza, Ignacio y Lozano Ortega, Miguel Angel. Presentación del curso "Java y Dispositivos Móviles". *Universidad de Alicante.* [En línea] 2009. <http://www.jtech.ua.es/jdm/present.htm>.

Institute of Electrical and Electronics Engineers (IEEE). *IEEE Recommended Practice for Software Requirements Specifications.* ANSI 830-1993.

Küng, Stefan, Onken, Lübbe y Large, Simon. *TortoiseSVN: Un cliente de Subversion para Windows: Versión 1.4.1.* Alemania : s.n., 2006.

Qualitrain. ¿Por qué seleccionar java como tecnología de desarrollo? *Qualitrain México.* [En línea] Empresa especializada en capacitación., 2010. <http://www.qualitrain.com.mx/Blog.html>

Collins-Sussman, Ben, W. Fitzpatrick, Brian y Pilato, C. Michael. *Control de versiones con Subversion.* [En línea]. <http://svnbook.red-bean.com/nightly/es/index.html>

Gamma, J.M.V.R.H.R.J.E. *Design Patterns: Elements of Reusable Object-Oriented Software.* 1995.

Burbeck, Steve. *Application programming in Smalltalk-80: How to use Model-View-Controller (MVC).* 2007.

Thomas Fihlman. *Estimation in software development projects.*

Glosario de Términos.

Ajax : Acrónimo de *Asynchronous Java Script And XML* (Java Script asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

Arquitectura: Descripción de la organización y estructura de un sistema.

API: (Application Programming Interface, en español Interfaz de Programación de Aplicaciones). Es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Caso de Uso: Descripción narrativa textual de la secuencia de eventos y acciones que ocurren cuando un usuario parte o divide en un diálogo con un sistema durante un proceso significativo.

CGI: (en inglés *Common Gateway Interface*, abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa.

CVS: (Concurrent Version System, en español Sistema concurrente de versiones). El CVS es un modo muy elegante de administrar versiones de archivos que permite trabajar fácilmente a más de un desarrollador en el mismo proyecto.

DAO:(Data Access Object, en español Objeto de Acceso a Datos). Es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología comprendidos por los practicantes de ese dominio.

Framework: Esquema, esqueleto o patrón para el desarrollo y/o la implementación de una aplicación. Conjunto de clases que cooperan y forman un diseño reutilizable para un tipo específico de software ofreciendo una guía arquitectónica partiendo el diseño en clases abstractas y definiendo sus responsabilidades y sus colaboraciones.

GPL: en inglés *GNU General Public License*, es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

GUI: (Graphical User Interface, en español Interfaz Gráfica de Usuario). Es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

IDE: (Integrated Development Environment, en español Entorno de Desarrollo Integrado): Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Inyección de Dependencia: Propone que las dependencias entre clases no sean tan unidas, sino atadas a clases interfaces, es decir, crear objetos, sin importar quién provea la implementación. Solo se sabe que quién provea la construcción, va a implementar una clase interfaz determinada.

IoC: (Inversion of Control, en español Inversión de Control): Es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones.

JAAS: (Java Authentication and Authorization Service), es una Interfaz de Programación de Aplicaciones que permite a las aplicaciones Java acceder a servicios de control de autenticación y acceso.

JavaBeans: Es un componente hecho en software que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje Java.

JDBC: (Java Database Connectivity, en español Conexión a Bases de Datos en Java). Es un API para trabajar con bases de datos desde Java, independientemente de la base de datos a la que accedemos.

JDO: Java Data Objects, de la API estándar de Java es un modelo de abstracción basada en la interfaz de persistencia, desarrollado bajo los auspicios de la Comunidad Proceso de Java.

JEE: Java Platform Enterprise Edition, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura de N niveles. La plataforma Java EE está definida por una *especificación*. Java EE es también considerada informalmente como un estándar debido a que los suministradores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son *conformes a Java EE*.

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

JPA: Java Persistence API, es la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB3. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares.

JSP: (del inglés *Java Server Pages*). Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

JVM: (Java Virtual Machine, en español Máquina Virtual de Java). Es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

LDAP: (*Lightweight Directory Access Protocol*, Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

MMS: (Multimedia Messaging System, en español Sistema de Mensajería Multimedia). Es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia, incorporando sonido, video, fotos o cualquier otro contenido disponible en el futuro.

PHP: es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o' GTK+.

RPC: (del inglés Remote Procedure Call, Llamada a Procedimiento Remoto) es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

RUP: (Rational Unified Process, en español Proceso Unificado de Desarrollo): Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Servlets: Aplicación sin interfaz gráfica que se ejecuta en un servidor de Internet, procesando información HTML previamente recogida por un navegador.

Singleton: El patrón de diseño Singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. *URL* (del inglés *Uniform Resource Locator*): Es un localizador uniforme de recurso y se refiere a la dirección única que identifica a una página web en Internet.

SMS : (del inglés *Short Message Service*). El servicio de mensajes cortos o SMS es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto, o más coloquialmente, textos o mensajitos) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano.

SOAP: (Simple Object Access Protocol): Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

SVN: El Sistema de Control de Versiones es un sistema centralizado para compartir información. En su núcleo está un repositorio, que es un almacén central de datos.

Telemática: es la unión de las telecomunicaciones y la informática. Esto se puede ver en su doble vertiente:

- Las telecomunicaciones al servicio de la informática, es decir, los medios de transmisión, las redes y los servicios de comunicaciones, permitiendo y facilitando el diálogo y el uso compartido de recursos entre ordenadores, lo que se hace patente en la realidad en las redes de área local de ordenadores, tanto para aplicaciones ofimáticas como industriales, intranets, Internet, etc.
- La informática al servicio de las comunicaciones, entendida como computadoras y programas que desarrollan tareas de comunicaciones como, por ejemplo, centrales digitales de telefonía, de transmisión de datos, redes digitales de servicios integrados (RDSI), Internet, conmutadores, routers, etc.

UML (Unified Modeling Language, en español Lenguaje Unificado de Modelado): Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

WAP: (del inglés *Wireless Application Protocol*). Es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, p.ej. acceso a servicios de Internet desde un teléfono móvil.

Web Service: Conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Widget: es cualquier elemento gráfico que podamos usar en el diseño de una interfaz gráfica. Estos elementos definidos según cada API de programación nos permiten con su combinación generar cualquier pantalla.

XML (Extensible Markup Language): Lenguaje de marcado utilizado para definir y utilizar las etiquetas necesarias para describir datos jerárquicos.