

Universidad de las Ciencias Informáticas Facultad 3



Título: Sistema para la gestión de eventos y notificaciones.

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

Autor: Yanier Alba Reyes

Tutor: Yulier Matías León

La Habana, 2013



Seamos realistas y hagamos lo imposible.

Ernesto Guevara de la Serna

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio.

Para que así conste firmo el presente a los ____ días del mes de _____ del año

Yanier Alba Reyes

Yulier Matías León

DATOS DE CONTACTO

Nombre: Yulier Matías León

Correo electrónico: [ymatias@uci.cu](mailto:yurias@uci.cu)

Graduado en la Universidad de las Ciencias Informáticas. Título de oro. Seis años de experiencia. Analista y diseñador del sistema Quarxo. Categoría docente Instructor. Profesor de la asignatura P3 y P4.

AGRADECIMIENTOS

Agradecer en primer lugar a mi madre, a mi padre, a mi padrino Reynaldo, por darme todo su apoyo en mi carrera, en los momentos buenos y más aún en los malos, a mi familia en general, a mis tías, Ana Ivis, Yasmina, Elena, Barbarita, a mis abuelos: Mario y chicha, a mis abuelos habaneros: Manuel y Bárbara que me acogieron como si fuera otro miembro de su familia y siempre estaban preocupado por todo lo que me pasaba. A mis segundos padres: Regla Ileana y Julio, a ustedes les debo mucho, no tengo palabras para expresar lo que siento por ustedes, siempre los llevaré en mi corazón. A Ariadna que en tan poco tiempo significa mucho para mí. A todos mis compañeros de aula en estos 6 años. A todos lo que de una forma u otra me ayudaron en esta difícil contienda, al musulmán de Jorge, a la pelusa de Raúl, al negro de Evelio, a miguel y areanne, gracia por todo, a todos mis compañeros del mejor proyecto de la uci: Banco, a mi tutor Yulier Matías, un agradecimiento enorme, gracias por todo, por estar en las buenas y en las malas, aun cuando las cosas estaban feas y casi no tenía esperanzas. A mi primo Yunier que es como si fuera mi hermano, a Charito, a Kukita, a Surima, que es como la hermanita mayor que nunca tuve. Agradecer especialmente a Saily, gracias por todo, por todas las cosas que me enseñaste, gracias a ti puedo decir que hoy por hoy soy una mejor persona, siempre estarás en mi corazón. A todos muchas gracias.

DEDICATORIA

Quisiera dedicar este trabajo diploma a todos los que han confiado en mí, en especial a mi madre Estervina, a mi padre Juan Daniel, a mi hermanito Yassiel, a mis dos viejitos lindos: mi abuelo Mario y mi abuela Chicha, a mi segunda madre Regla Ileana y mi segundo padre Julio.

RESUMEN

El presente trabajo de diploma se centra en la necesidad de desarrollar una aplicación web que permita mejorar la centralización y notificación de los eventos científicos, culturales, deportivos en el Centro de Informatización de la Gestión de Entidades (CEIGE), de manera que los usuarios puedan estar mejor informados sobre estos eventos. Para dar cumplimiento al objetivo de la investigación se realizó una fundamentación teórica, donde se analizaron los conceptos fundamentales relacionados con la investigación. Además de realizar un estudio de las tecnologías, herramientas y lenguajes utilizados en el desarrollo de aplicaciones web, definiéndose como lenguaje de programación Java, el marco de trabajo Spring en su versión 3.0 para la lógica de negocio, como gestor de base de datos PostgreSQL en su versión 9.1, el proceso de desarrollo de software estuvo guiado por la metodología SXP.

PALABRAS CLAVE

Desarrollo, Centralización, Eventos, Gestión, Notificación.

ÍNDICE

AGRADECIMIENTOS	II
DEDICATORIA.....	III
RESUMEN.....	IV
INTRODUCCIÓN`	7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	11
1.1 Introducción.....	11
1.2 Sistemas existentes	11
1.2.1 Sistemas existentes a nivel internacional.....	11
1.2.2 Sistemas existentes en la Universidad de Ciencias Informáticas.	14
1.3 Tecnologías y herramientas utilizadas en el desarrollo.....	16
1.3.1 Metodología de desarrollo.....	16
1.4 Lenguaje de modelado y Herramientas	21
1.4.1 Lenguaje Unificado de Modelado 2.0 (UML)	22
1.4.2 Herramientas CASE	22
1.4.3 Ambiente de desarrollo	24
1.4.4 Sistemas Gestores de Base de Datos.	26
1.4.5 Lenguajes de Programación	29
1.4.6 Plataforma de programación a emplear	32
1.4.7 Lenguajes en el lado del cliente	32
1.4.8 Servidor Web	33
1.4.9 Marco de trabajo	35
1.5 Arquitectura definida para el sistema.....	37
1.5.1 Arquitectura Multicapa	38
1.6 Patrones	38
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	42
2.1 Introducción.....	42
2.2 Planificación por roles para la realización del sistema.....	42
2.3 Actores del Sistema	43

2.4	Diagrama de Historia de Usuario	44
2.5	Lista de Reserva del Producto (LRP).....	44
2.6	Requerimientos no funcionales	45
2.7	Historias de Usuarios y Tareas de Ingeniería	46
2.8	Plan de lanzamiento	55
2.9	Patrones empleados.....	55
2.10	Diagrama de paquetes	56
2.11	Diagrama de clases	58
2.12	Interfaz de usuario.....	58
2.13	Conclusiones del capítulo	58
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA		60
3.1	Introducción.....	60
3.2	Estándares de Codificación.....	60
3.2.1	Convenciones de nomenclatura	60
3.3	Pruebas	61
3.3.1	Pruebas de aceptación	61
3.3.2	Casos de prueba de aceptación	62
3.3.3	Realización de las pruebas de aceptación.....	67
3.4	Diagrama de despliegue	68
3.5	Conclusiones del capítulo	69
CONCLUSIONES GENERALES.....		70
RECOMENDACIONES.....		71
BIBLIOGRAFÍA CONSULTADA		72
ANEXOS.....		75
GLOSARIO.....		77

INTRODUCCIÓN

En la actualidad el uso de las Tecnologías de la Informática y las Comunicaciones (TIC) constituye un papel importante en el desarrollo de cada país. Como bien se define, son el conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro. Abarcan un abanico de soluciones muy amplio. Incluyen las tecnologías para almacenar, recuperar, enviar y recibir información de un sitio a otro o procesar información para poder calcular resultados y elaborar informes (1).

Estas incluyen todos los sistemas informáticos no solamente la computadora, este es solo un medio más, el más versátil, pero no el único; también los teléfonos celulares, la televisión, la radio, los periódicos digitales.

Por esa razón hoy en día el mundo se ha enmarcado en el desarrollo de las TIC. Las empresas e instituciones en el mundo automatizan y perfeccionan los procesos y actividades que realizan. Esta revolución científica y tecnológica se está aplicando en todos los campos de la sociedad, se presentan cada vez más como una necesidad, es una realidad y uno de los motores principales de la sociedad actual en que vivimos. Gracias al creciente y rápido avance de las tecnologías existen muchos medios, sistemas y aplicaciones que permiten o tratan el tema relacionado con las notificaciones, avisos, calendarios y eventos, referentes a las noticias, al deporte, a la cultura, entre otras esferas de la sociedad donde las TIC han tenido un gran impacto.

Cuba no se ha quedado atrás en cuanto al uso de las TIC, sino que ha mantenido un desarrollo sostenido en la aplicación de las nuevas tecnologías en las diferentes ramas de la sociedad, de forma tal que podamos estar a la par del mundo desarrollado en algunas de ellas (2). Con el objetivo de aprovechar las ventajas que estas tienen, Cuba ha venido creando toda una infraestructura científica y técnica que permite contar con un conjunto de centros de investigación, entre los cuales se puede mencionar a la Universidad de las Ciencias Informáticas (UCI). La UCI entre sus principales objetivos contempla la informatización de la sociedad cubana. En la misma existen varios centros de desarrollo de software teniendo en cuenta las distintas aristas de la sociedad, entre los que se pueden citar: Centro de Identificación y Seguridad Digital, Centro de Informatización de la Gestión de Entidades, Centro de Gobierno Electrónico.

En el Centro de Informatización de la Gestión de Entidades (CEIGE) circula un alto volumen de información referente a eventos científicos, culturales, deportivos. El centro pertenece a la UCI que también planifica eventos y ambas planificaciones son de interés para los usuarios del CEIGE. En la actualidad los eventos circulan por diferentes vías, como son: los portales de las distintas facultades, la Intranet, por el cliente de correo web Zimbra, por la televisión, la radio universitaria. Esta variedad de medios de divulgación existentes conlleva a que el usuario emplee demasiado tiempo consultándolos para mantenerse informado, además de que tenga que recordar todo lo referente a los diversos eventos y tener que replicar la información en su agenda personal, aparte de que se desconozcan una serie de eventos que ocurren en la universidad. Los sistemas existentes en la universidad no tienen centralizada la información de los eventos, razón por la cual no permiten la correcta gestión de los mismos por parte de los usuarios. No presentan la opción de que los usuarios creen sus propios canales y que a su vez otros se suscriban a los mismos y puedan obtener las informaciones de los eventos pertenecientes al canal del cual están suscritos.

Una vez analizada la situación anterior se formula el siguiente **problema a resolver**: ¿Cómo notificar eventos, de manera que contribuya a la centralización, organización y distribución de la información en el CEIGE?

Para dar solución al problema planteado anteriormente se define como **objeto de estudio**: Los procesos de notificación de eventos.

Según lo planteado anteriormente se establece como **campo de acción**: Los procesos de notificación de eventos en el CEIGE.

Además, se define como **objetivo general**: Desarrollar una aplicación web que contribuya a la mejora del funcionamiento de los procesos de gestión de eventos y notificación en el CEIGE.

Con el desarrollo e implementación de este sistema se espera como **resultado**: una aplicación web para la gestión de los eventos y una aplicación de escritorio para la notificación de los mismos en las máquinas clientes. Con dicha aplicación se pretende mejorar la centralización, organización, distribución y notificación de los eventos, de manera que los usuarios puedan estar informados de forma automática sin necesidad de realizar búsquedas por los diferentes medios.

Para cumplir con el objetivo de la problemática especificada, se plantean los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Justificar el uso de la metodología de desarrollo, tecnologías, lenguajes y herramientas propuestas para el desarrollo de la aplicación.
- Identificar los requisitos funcionales para la gestión de eventos y notificación.
- Definir una arquitectura para el sistema.
- Realizar el diseño del sistema bajo la arquitectura seleccionada.
- Implementar el sistema bajo la tecnología propuesta.
- Validar la solución propuesta.

En el desarrollo de este trabajo son utilizados algunos métodos científicos de la investigación que ayudan a guiar todo el desarrollo de la solución, entre los que se encuentran:

Métodos Teóricos:

- **Histórico-lógico:** Con el objetivo de realizar un estudio del estado del arte del tema a investigar. De esta manera, se puede conocer acerca de la existencia y características de sistemas de notificación de eventos.
- **Hipotético-deductivo:** La investigación se desarrolla a partir de un problema concreto del cual se deriva el objetivo y los objetivos específicos, que serán cumplidos durante el desarrollo de la investigación utilizándose métodos científicamente bien fundamentados (3).

Métodos Empíricos:

- **Entrevistas:** Se hicieron con el objetivo de obtener los elementos necesarios para definir qué es lo que se necesita y los problemas existentes que se resolverían.

El presente documento se encuentra compuesto por tres capítulos:

Capítulo 1:

Se presenta la fundamentación teórica del tema mediante; la descripción de algunos sistemas de gestión de eventos relacionados con el objeto de estudio, la valoración del estado del arte, el estudio de la

Arquitectura Base definida para el sistema y la justificación de las herramientas y tecnologías a utilizar para el diseño e implementación del sistema.

Capítulo 2:

En este capítulo se realiza el desarrollo del sistema. Se explica la dinámica del sistema a desarrollar en forma de historias de usuarios, prototipos de interfaz de usuario, diagrama de historia de usuarios, entre otros modelos auxiliares, además del plan de lanzamiento para las entregas intermedias y la lista de reserva del producto.

Capítulo 3:

Estuvo centrado principalmente en la implementación del sistema, se muestra y explica en cada caso los estándares de codificación empleados. Por otra parte, se realiza la validación del sistema a partir de la realización de las pruebas de aceptación, brindando una explicación de los casos de pruebas de aceptación utilizados en la realización de las mismas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se aborda los elementos teóricos necesarios para el desarrollo de la investigación, además de algunas aplicaciones tanto del ámbito nacional como internacional que abordan dicho tema, para a partir de todos estos elementos llevar a cabo el desarrollo del sistema, haciendo más sencilla la comprensión de los temas que serán abordados en lo adelante. También se abordará lo referente al uso de la metodología de desarrollo, herramientas que se utilizarán en el desarrollo de la aplicación, así como la arquitectura definida para el mismo.

1.2 Sistemas existentes

A continuación se muestran algunas aplicaciones que entre las funcionalidades que poseen permiten la gestión, notificación de eventos y tareas.

1.2.1 Sistemas existentes a nivel internacional

Google Calendar

Es un espacio personal gratuito que ofrece Google a todos aquellos que disponen de una cuenta de usuario para que puedan crear, gestionar y compartir eventos dentro de sus agendas. Las interfaces de este servicio (realizadas en Ajax) permiten una fácil introducción de datos, tarea que se puede realizar también a través de importación de datos externos, provenientes de aplicaciones externas como Microsoft Outlook Calendar o Yahoo Calendar, o incluso desde agendas de otros usuarios de Google Calendar, puesto que se utiliza un estándar de transmisión de datos basado en XML. Los usuarios pueden además crear múltiples calendarios (cada uno de ellos sobre una temática diferente o creados para compartirlos con amigos) y enviar a sus contactos un determinado evento para que lo añadan a sus agendas, además dispone de una integración con Gmail (servicio de correo de Google que reconoce eventos mencionados dentro de los mensajes y permite añadirlos a la agenda). Existe la posibilidad de recibir notificaciones vía SMS en algunos países, y también imprimir los calendarios en formato PDF (4).

Desventajas con respecto a la investigación:

- No da la oportunidad de descargar la aplicación y utilizarla.

- Todas las consultas de los diferentes eventos, calendarios y otras opciones que brinda se deben realizar teniendo acceso a internet.

Rainlendar

Rainlendar es una completa aplicación de calendarios. Puede almacenar eventos y tareas diarias y mantenerlas visibles en el escritorio todo el tiempo. La apariencia del calendario puede cambiarse con temas para hacer que coincidan con el aspecto visual de tu escritorio.

Características

- Independiente de la Plataforma Rainlendar puede ejecutarse en Windows, Mac OS X y Linux.
- Rainlendar utiliza el formato estándar iCalendar para sus eventos y tareas para maximizar la compatibilidad con otras aplicaciones. Puedes transferir fácilmente los eventos del calendario que estás usando actualmente a Rainlendar.
- Permite importar y exportar eventos y tareas, pueden ser transferidos de y hacia otra aplicación.
- Permite ser notificado de antemano antes de que el evento culmine. La alarma también funciona con tareas.
- Permite que los eventos y tareas pueden repetirse automáticamente para no tener que introducirlos múltiples veces.
- Permite realizar búsqueda de los eventos y las tareas (5).

Jevent

La calendarización de actividades y eventos en Joomla se puede realizar de varias maneras, pero no son muchas las extensiones de peso que permiten esta gestión de forma atractiva y fácil de entender para el usuario que debe gestionarlas. Una de estas extensiones es **Jevent**.

Características de la extensión Jevent:

- Permite crear de una sola vez eventos y complejos patrones de repetición de eventos y visualizarlos en un calendario con una gran variedad de formatos.
- Posee múltiples diseños de plantilla.
- Permite importación/exportación de iCal.
- Manejo repetitivo de excepciones.

- Los eventos pueden ser categorizados y las vistas del calendario se pueden personalizar para mostrar todas o algunas de estas categorías.
- Permite añadir a la web un calendario en el que se puede publicar eventos y clasificarlos en distintas categorías. Además de permitir a los usuarios crear y publicar eventos, realizar búsquedas entre los eventos publicados, o exportar todos los elementos del calendario en formato iCal (6).

Desventajas con respecto a la investigación:

- No permite múltiples notificaciones por períodos.
- No permite realizar búsquedas por varias categorías al mismo tiempo.
- No permite a un usuario registrar su asistencia a un determinado evento.
- No permite la notificación de estos eventos un tiempo antes de que este ocurra.

RSS

RSS (del inglés Really Simple Syndication, en español Sindicación Realmente Simple) es un sencillo formato de datos que es utilizado para re difundir (sindicar) contenidos a suscriptores de un web. RSS es utilizado en los sitios que se actualizan con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas desde la computadora del usuario (7).

Los archivos RSS comúnmente se llaman feeds RSS o canales RSS y contienen un resumen de lo publicado en el sitio web de origen. Se estructura en uno o más ítems. Cada ítem consta de un título, un resumen de texto y un enlace a la fuente original en la web donde se encuentra el texto completo. Además, puede incluir información adicional como el nombre del autor o la fecha y la hora de publicación del contenido. Por tanto, cualquier fuente de información susceptible puede ser dividida en ítems (los mensajes de un foro, por ejemplo) y pueden distribuirse utilizando RSS.

Desventajas de las tecnologías RSS con respecto a la investigación:

- No permiten la notificación de estos eventos un tiempo antes de que ocurra el evento en sí.
- Estas tecnologías no permiten realizar búsquedas por categorías sobre los diferentes canales y eventos existentes.
- Permiten la divulgación de noticias, pero el estándar actual no tiene campos para la fecha de comienzo de un evento y el lugar.

1.2.2 Sistemas existentes en la Universidad de Ciencias Informáticas.

Zimbra Collaboration Suite

Zimbra Collaboration Suite (ZCS) es una suite de colaboración que combina herramientas de correo electrónico, calendarios, libreta de direcciones, bloc de notas. Todas estas herramientas se encuentran al utilizar el cliente de web en una misma dirección, un único login y desde cualquier lugar donde se encuentre, lo que demuestra la potencia de esta solución sin ni siquiera haber entrado en detalle sobre las distintas herramientas. En muy poco tiempo, Zimbra se ha convertido en la solución de código abierto líder a nivel mundial tanto para empresas como para proveedores de servicio, centros educativos y administraciones públicas (8).

Características sobre las que se asienta Zimbra:

Flexibilidad: personaliza Zimbra fácilmente según las necesidades de la organización.

Libertad: utiliza el cliente web de Zimbra junto con otros programas tradicionales, como plataforma mixta.

Bajo mantenimiento: gestión completamente sencilla.

Desventajas de Zimbra con respecto a la investigación:

- A pesar de que permite compartir agendas entre usuarios, el calendario o agenda tiene un propietario que es el dueño de la agenda, por lo que el calendario o agenda dependen de la existencia de una cuenta de correo específica.
- No es posible realizar una búsqueda avanzada por categorías sobre estas agendas o calendarios creados públicos.
- No es posible saber cuántas personas están suscritas a una agenda pública.
- No permite definir una fecha de conclusión para una agenda, lo que conlleva a que el usuario deba revisar constantemente sus agendas suscritas y ver cuáles están activas y cuáles no.
- Presenta una vista vertical de eventos que provoca que los eventos concurrentes se solapen unos encima de otros, cosa que no es conveniente cuando pueden existir muchos eventos en un mismo horario.

Portales de la UCI e intranet

Actualmente la Universidad de las Ciencias Informáticas cuenta con una amplia gama de sitios web, entre los que se encuentran: el portal de la Facultad 3 (<http://dragones.uci.cu/>), el portal de la Facultad 4

(<http://octavitos.uci.cu/>), el portal de la Facultad 1 (<http://facultad1.uci.cu/>) y la Intranet. Tienen entre sus principales objetivos divulgar todas las actividades que se realizan en la misma, ya sea deporte, cultura, noticias, eventos. Además de contar con información referente a los distintos centros de desarrollo existentes en las mismas, así como aspectos importantes referentes a la extensión y formación universitaria.

Desventajas de los portales de la uci e intranet:

Estos portales e intranet de la UCI no brindan la posibilidad de una verdadera notificación de los diferentes eventos que se realizan en la misma. No permiten la opción de divulgar eventos por diferentes canales, sino que se le muestran todos los eventos al usuario. No se muestran todos los eventos que ocurren dentro de la Universidad a los usuarios, ya que estos portales no se comunican entre sí, razón por la cual no se puede consultar toda la información de los eventos desde un solo lugar. Además de no brindar la opción al usuario de una notificación automática de los eventos un tiempo antes de que estos ocurran.

GesPro

Es un gestor de proyectos muy potente y maduro, flexible y bastante estable. Es una aplicación de software libre con libertad de uso, modificación y distribución. Está actualmente en desarrollo, por lo que continuamente surgen nuevas versiones. Es multiplataforma y multilenguaje, incluyendo el español. Se pueden definir los diferentes hitos del proyecto y las tareas a realizar para cada uno de estos hitos. Cada tarea se puede asignar a uno de los desarrolladores. Los desarrolladores tienen en su página de entrada una lista de las tareas que tienen asignadas. Según van trabajando en las mismas, pueden ir marcando el tiempo que estiman que les llevará la tarea, el tiempo que han trabajado en ella y/o el porcentaje que creen que tienen realizado. Actualmente esta herramienta está siendo utilizada en la Universidad de las Ciencias Informáticas, para facilitar el monitoreo de los diferentes proyectos. Uno de los hitos que se gestionan de cada proyecto es el vinculado con el soporte a software. En este espacio se le da el tratamiento adecuado a las incidencias recibidas en dependencia del tipo que sean. Cada responsable de las actividades de soporte tiene asignado en su página, las tareas que debe realizar. De esta manera, se agiliza y se facilita el proceso (9).

Desventajas de este sistema con respecto a la investigación:

- No permite la creación de canales o grupos libremente por parte de los usuarios.
- No está pensado para que los usuarios se suscriban libremente a los grupos.

- No se tienen en cuenta para las tareas la hora de comienzo ni el tiempo de recordatorio.

1.3 Tecnologías y herramientas utilizadas en el desarrollo

Es importante conocer el contexto donde se aplicará el software que se desarrolla. En este epígrafe se hace un análisis de las principales tecnologías a considerar para ser empleadas en el desarrollo del sistema que dará solución a la presente investigación. También se describen los lenguajes, las metodologías, las tecnologías y herramientas a considerar para su posterior utilización en el desarrollo, analizando sus características, y seleccionando las mejores propuestas con el objetivo de dar cumplimiento con la mayor eficiencia y calidad posible al objetivo general de la presente investigación.

1.3.1 Metodología de desarrollo

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software. A continuación se mencionan algunas metodologías, entre las que se encuentran RUP (Proceso Unificado de Software), MSF (Microsoft Solutions Framework), XP (Programación Extrema), AUP (Proceso Ágil Unificado), FDD (Desarrollo basado en funcionalidades) y Scrum.

Se podrían clasificar en dos grandes grupos: Metodologías Pesadas o Robustas y Metodologías Ligeras o Ágiles.

1.3.1.1 Metodologías ligeras o ágiles

Las metodologías ágiles han despertado gran interés en el desarrollo de software ya que proponen simplicidad y velocidad para crear sistemas. Plantean varios aspectos como son:

- Los individuos y la interacción por encima de los procesos y herramientas.
- El software que funciona por encima de la documentación abarcadora y detallada.
- La colaboración con el cliente por encima de la negociación contractual.
- La respuesta al cambio por encima del seguimiento de un plan.
- Flexibilidad para que el cliente pueda cambiar los requisitos cuando quiera y que el código funcione bien.

- Iteraciones cortas, de esta forma, se puede realizar un mejor control de lo que se está realizando y si cumple con las necesidades del cliente.
- Incremental. Entregas pequeñas de software, con ciclos rápidos

Programación Extrema (XP)

La metodología XP, es la primera metodología ágil y la que le dio conciencia al movimiento actual de metodologías ágiles. XP es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define especialmente para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. El objetivo de la XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio (10).

XP es un proceso muy orientado a la programación, debido al bajo número de documentación a generar, además la programación se hace en pareja que es un punto a favor de XP, pero el código pertenece al equipo completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento para el final.

Una de las herramientas más importantes en la metodología XP es el desarrollo orientado a pruebas, que utiliza las pruebas unitarias como eje de todo desarrollo. Las interacciones suelen ser muy cortas y se promueve a los programadores a buscar soluciones y experiencia con ellas, programar sin miedo a descomponer el sistema.

Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Scrum, más que una metodología de desarrollo software, es una forma de auto-gestión de los equipos de programadores, estos equipos pueden contar entre 2 y 8 personas (11). Un grupo de programadores deciden cómo hacer sus tareas y cuánto van a tardar en ello. Scrum ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro.

Scrum permite además seguir de forma clara el avance de las tareas a realizar, de forma que los "jefes" puedan ver día a día cómo progresa el trabajo. En Scrum se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo del producto. Esta metodología tiene algunas desventajas, como el hecho de que genera muy poca documentación en comparación con otras metodologías, no es apto para todos los proyectos y en muchas ocasiones es necesario completarlo con otros procesos de XP (12).

SXP

Esta metodología surgió en Cuba, en la UCI, con el objetivo de lograr un buen desarrollo de software. Es el resultado de la unión de las metodologías ágiles Scrum y XP, estas metodologías se acoplan y retroalimentan perfectamente una dentro de la otra. Scrum modela la interfaz entre el equipo y los clientes y XP entra a funcionar en cómo el equipo debe hacer su trabajo. De esta manera, se integran efectivamente, surgiendo así la metodología de procedimiento ágil a seguir SXP, la cual cuenta para su desarrollo con cuatro fases, estas son: Planificación-Definición, Desarrollo, Entrega y Mantenimiento. Cada una de estas fases está compuesta por una serie de actividades que son las que generan los artefactos que quedan incluidos en el nuevo expediente de proyecto que se concibe.

Planificación–Definición: En esta fase se generan todos los documentos que se encuentran relacionados con la concepción inicial del sistema, así como la definición del mismo. También se incluyen algunos que están vinculados a la primera parte de los procesos de ingeniería de software tales como los relacionados con el negocio, los requisitos y el diseño.

Desarrollo: En la primera parte de esta fase se generan todos los documentos relacionados con la planificación de las iteraciones, y además se recogen las principales definiciones que se manejan en la metodología y otros términos de difícil entendimiento para los clientes, así como de las tareas a realizar durante la implementación. Además, se genera el código fuente en la etapa de implementación, los documentos relacionados con las pruebas, como última parte de esta etapa.

Entrega: En esta fase se realiza la entrega del software y su documentación, generándose aquellos documentos que son imprescindibles para el entrenamiento y entendimiento del producto.

Mantenimiento: Se realizan las actividades relacionadas con el soporte del software y se generan los documentos relacionados con los cambios que puedan ocurrir en el mismo. Para la definición de los artefactos que se generan en cada una de las fases, se tiene en cuenta como elemento fundamental las características de las metodologías ágiles, las cuales tienen como premisa la no duplicación de esfuerzos. Así como la integración del cliente en el equipo de desarrollo, esto garantiza que no haya necesidad de documentaciones extensas, solo se documenta lo necesario para una futura reutilización.

De cada una de las fases anteriormente mencionadas se despliegan 7 flujos de trabajo: concepción inicial, captura de requisitos, diseño con metáforas, implementación, prueba, entrega de la documentación, soporte e investigación, el cual se utiliza por el equipo de desarrollo cuando sea necesario, es decir, es un flujo que se puede mover y utilizarlo en cualquier parte del ciclo de vida del proyecto. De estos flujos se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, planificación de las iteraciones y las actividades que se van a realizar para lograr el producto, pruebas, además de las tareas necesarias para realizar las investigaciones para documentar todo el proceso (13).

1.3.1.2 Metodologías robustas o pesadas

Son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida.

Proceso Unificado Racional (RUP)

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. El Proceso Unificado es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes niveles de aptitud y diferentes tamaños de proyecto.

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida del proyecto. Cada ciclo está dividido en cuatro fases: Inicio, Elaboración, Construcción y Transición y durante cada fase tienen lugar un grupo de flujos de trabajos que son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, Instalación entre otras (14).

Características generales de RUP:

- Es un proceso pesado.
- Dirigido por casos de uso, donde se refleja lo que los usuarios futuros necesitan y desean representándose a través de los requerimientos.
- Centrado en la arquitectura, pues esta muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo.
- Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo aunque no sean completamente necesarios productivamente, haciendo el proceso menos ágil.
- Está pensado para equipos de desarrollo grandes en los que están muy bien definidos los roles, artefactos que generan y el nivel de detalle con que lo hacen en cada fase.

- Es necesario incluir a más personas en el equipo de desarrollo: Especialistas en los diseños y evolución de casos de uso, de los modelos de análisis y diseño, de los modelos de implementación.
- Está basado mucho en la documentación. Por solo citar un ejemplo: Existen diferentes elementos de planificación (plan de desarrollo, plan de iteración, plan de calidad, entre otros).
- Si el conjunto de documentos y artefactos no son concebidos tal y como se plantea, dicha documentación solo servirá para ser archivada lo cual no genera valor respecto a la calidad del desarrollo, y evoluciona en problemas más complejos.
- Lo más importante en el desarrollo de un producto informático es el propio desarrollo, en RUP se gasta posiblemente demasiado tiempo para pasar a la fase de desarrollo (15).

Selección de la metodología de desarrollo

Después de haber realizado un análisis de las metodologías anteriormente expuestas atendiendo a sus principales características y los aspectos preponderantes de cada una de ellas, se optó por la metodología de desarrollo SXP. Con la utilización de SCRUM para la gestión, se logra una buena planificación y organización; mientras que XP respalda con sus prácticas todo el proceso de desarrollo. Por otro parte está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. RUP no es recomendable usarla en soluciones como la que se pretende desarrollar en esta investigación. Entre sus desventajas se pueden citar: el exceso de documentación que no aporta valores a la solución y retrasa la entrega del producto, su utilización en proyectos de gran envergadura y que a su vez cuentan con grandes equipos de desarrollo.

1.4 Lenguaje de modelado y Herramientas

El progreso de software ha tenido un gran avance en los últimos tiempos y las herramientas de modelado forman un componente muy significativo en el entorno de desarrollo, puesto que son esenciales para el análisis de sistemas. Las herramientas mejoran la forma en que ocurre el desarrollo y tiene influencia sobre la calidad del resultado final. En la actualidad se han creado una serie de herramientas con este fin, las cuales han sido mejoradas por parte de sus desarrolladores con el fin de encontrar en ellas fiabilidad, eficiencia, entre estas se encuentra: Visual Paradigm y Rational Rose Enterprise Edition, dichas

herramientas usan el Lenguaje de Modelado UML para la especificación, la visualización, la construcción y la documentación de los artefactos de los sistemas de software y también para otros tipos de sistemas.

1.4.1 Lenguaje Unificado de Modelado 2.0 (UML)

Cuando se define UML no se puede hablar de métodos o guías, pues simplemente es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema intensivo de software. Este lenguaje ofrece una forma estándar para escribir un plano del sistema, incluyendo cosas conceptuales tales como procesos de negocios y funciones del sistema, así como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Características de UML

- Estandarización del lenguaje.
- Visualiza, detalla, construye y documenta los artefactos de un sistema que involucra una gran cantidad de software.
- Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas.
- Permite la modelación de sistemas con tecnología orientada a objetos

1.4.2 Herramientas CASE

Roger Pressman plantea que las herramientas CASE ayudan a los gestores y practicantes de la ingeniería de software en todas las actividades asociadas a los proyectos de software, automatizan las actividades de gestión de proyectos, gestionan todos los productos de los trabajos elaborados a través del proceso y ayudan a los ingenieros en el trabajo de análisis, diseño y codificación.

Las herramientas CASE brindan una gran gama de componentes que incluyen todos o la mayoría de los requisitos necesarios para el desarrollo de los sistemas, además son creadas con una gran exactitud en torno a las necesidades de los desarrolladores de software para la automatización de procesos incluyendo el análisis, diseño e implementación. Ofrecen una gran plataforma de seguridad a sistemas que las usan.

Las herramientas CASE se han venido ampliando y desarrollando, existe una gran variedad de estas con características específicas, a continuación describiremos algunas de ellas: Visual Paradigm for UML Enterprise Edition 6.4 y Rational Rose Enterprise Edition (16).

Visual Paradigm 6.4

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Visual Paradigm soporta el ciclo de vida completo del desarrollo de software: Análisis y Diseño orientados a objetos, Construcción, Pruebas y Despliegue. En Visual Paradigm for UML, el diseño está centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Es una herramienta diseñada para una amplia gama de usuarios, incluyendo el analista del sistema, ingenieros de software, entre otros.

Visual Paradigm for UML se caracteriza por lo siguiente:

- Soporta todos los diagramas de SysML y diagramas de entidad-relación.
- Ofrece amplias características de modelado de casos de uso incluyendo la función completa de UML, diagrama de casos de uso y editor de flujo de eventos.
- Genera Java, C#, C++, PHP y lenguaje de definición de datos (DDL) para todas las bases de datos populares.
- Permite diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. El mismo proporciona abundantes tutoriales del Lenguaje de Modelado, demostraciones interactivas de UML y proyectos UML.
- Visual Paradigm for UML se integra con herramientas Java, como son: Eclipse/IBM, NetBeans IDE, entre otras.
- Visual Paradigm for UML es apoyado por un conjunto de lenguajes tanto en la generación del código como en la Ingeniería Inversa, por mencionar algunos ejemplos los cuales tienen la capacidad de soporte, Java, C, C++, CORBA IDL, PHP, XML, Schema, Ada, y Python.
- Proporciona soporte a varios lenguajes en generación de código e Ingeniería Inversa a través de plataforma Java, Ruby, .NET, Python, CORBA IDL, Java, C++, XML Schema, entre otros.

Rational Rose

Rational Rose es una potente herramienta de modelado visual para ayudar en el análisis y diseño de software orientado a objetos. Es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++ y Visual Basic. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Rational Rose posee características adicionales incluidas:

- Rational Rose está basada en modelos que se integra con las bases de datos y los IDE (Entorno de Desarrollo Integrado) de las principales plataformas del sector.
- Rational Rose soporta el análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- Rational Rose Empresarial posee características de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Rational Rose Empresarial tiene soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- Rational Rose Enterprise tiene integración con cualquier sistema de control de versiones compatible con SCC, como IBM Rational Clear Case.
- Rational Rose Empresarial es capaz de crear definiciones de tipo de documento XML para el uso en la aplicación (17).

Selección de la herramienta CASE a utilizar

Después de realizar un análisis entre las herramientas CASE, se escogió Visual Paradigm For UML Enterprise Edition 6.4, la cual soporta las últimas versiones del mismo, (Lenguaje de Modelado Unificado), y la Notación y Modelado de Procesos de Negocio. Visual Paradigm For UML Enterprise Edition, es una herramienta muy completa y fácil de emplear. Permite que varios desarrolladores trabajen en el mismo diagrama y vean los cambios que se hacen en tiempo real, además de ayudar a los desarrolladores a mejorar la construcción del modelo del software, lo que posibilita acelerar la producción y mejorar la calidad del trabajo. En cambio Rational Rose Enterprise Edition es una herramienta muy utilizada pero es software privativo y no es multiplataforma, al contrario de Visual Paradigm For UML Enterprise Edition que es una herramienta multiplataforma y algunas de sus versiones son gratuitas.

1.4.3 Ambiente de desarrollo

Un entorno de desarrollo integrado es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda

para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones. Hoy en día los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y Visual Basic entre otros). Entre los principales IDE's de desarrollo para el lenguaje Java se encuentran Eclipse, NetBeans, e IntelliJ Idea.

Eclipse

Esta herramienta es un entorno de desarrollo integrado multiplataforma (utilizado para los lenguajes C, C++, Python y Java entre otros) de código abierto, utilizado en su mayoría para desarrollar otros entornos de desarrollo. En la actualidad esta herramienta está siendo desarrollada por la Fundación Eclipse, una organización independiente sin ánimo de lucro que intenta fomentar una comunidad de código abierto así como un conjunto de productos complementarios, capacidades y servicios. La versión existente en la actualidad de esta herramienta ofrece las siguientes características: editor de texto, resaltado de sintaxis, compilación en tiempo real, pruebas unitarias con JUnit, control de versiones y asistentes para el inicio en algunos de los elementos soportados (como proyectos, clases y pruebas).

Emplea plug-ins para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. La arquitectura de plug-ins permite integrar diversos lenguajes, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo, tales como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías, entre otros.

NetBeans

El IDE NetBeans es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans es una aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, Java Script y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una amplia selección de complementos de terceros. NetBeans es un entorno multiplataforma para crear aplicaciones Java. Incluye módulos tales como un editor, herramientas para trabajar con el código fuente (Java, C++ y otros), control de versiones. NetBeans puede sincronizar su modelo en Visual Paradigm y el código en NetBeans.

IntelliJ IDEA

Es un IDE centrado en la productividad del código. El editor entiende profundamente el código, hace sugerencias muy adecuadas y ayuda a dar forma a su código. Soporta Java, Groovy, XML con asistencia de código y Subversión. Entre sus principales características se encuentra la asistencia inteligente de codificación, la generación de código, el estilo de código, la documentación que posee, el análisis de código sobre la marcha lo cual usa un método intuitivo para ayudar a escribir el código. Posee una amplia gama de lenguajes de programación permitidos. Trae integrado varias tecnologías y framework como: Java Server Pages (JSP), Java Server Faces (JSF), Spring, Hibernate, Servicios WEB, entre otras. Una característica de importancia que se debe conocer, es que es una aplicación propietaria y por tanto se debe pagar por su licencia o uso (18).

Selección del ambiente de desarrollo a emplear

Todas estas herramientas son muy potentes y facilitan en gran medida el desarrollo con el lenguaje java. La utilización de IntelliJ Idea fue descartada desde el inicio por su condición de software privativo. Entre el NetBeans y el Eclipse se analizó la experiencia de los desarrolladores con los mismos, comprobándose que en su mayoría usaban Eclipse debido a las prestaciones de las estaciones de trabajo y al bajo consumo de recursos que evidencia este frente al NetBeans.

Se hace uso de Eclipse en su versión 3.5

1.4.4 Sistemas Gestores de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.

Entre algunos de los sistemas gestores de base de datos más usados se encuentran: Oracle y PostgreSQL, a continuación se especifican las características de cada uno para luego hacer la elección de cuál SGBD se utilizará en el desarrollo de la aplicación.

Oracle

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos fabricado por Oracle Corporation. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales.

Principales características:

- Entorno cliente/servidor.
- Gestión de grandes bases de datos.
- Usuarios concurrentes.
- Alto rendimiento en transacciones.
- Sistemas de alta disponibilidad.
- Autogestión de la integridad de los datos.
- Opción distribuida.
- Portabilidad.
- Compatibilidad.
- Replicación de entornos.

Ventajas:

- Las entidades complejas del mundo real y la lógica se pueden modelar fácilmente, lo que permite reutilizar objetos para el desarrollo de base de datos de una forma más rápida y con mayor eficiencia.
- Los programadores de aplicaciones pueden acceder directamente a tipos de objetos Oracle, sin necesidad de ninguna capa adicional entre la base de datos y la capa cliente.
- Las aplicaciones que utilizan objetos de Oracle son fáciles de entender y mantener porque soportan las características del paradigma orientado a objetos.
- Tiene buen rendimiento y hace buen uso de los recursos.
- Posee un rico diccionario de datos.
- Brinda soporte a la mayoría de los lenguajes de programación.
- Es un sistema multiplataforma, disponible en Windows, Linux y Unix.
- Permite tener copias de la base de datos productiva en lugares lejanos a la ubicación principal. Las copias de la Base de Datos productiva pueden estar en modo de lectura solamente (19).

Desventajas:

- El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras.
- Necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y conectar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.
- También es elevado el coste de la formación, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.

PostgreSQL

PostgreSQL es uno de los Sistemas Gestores de Base de Datos más utilizados por la comunidad de software libre por las razones siguientes: cumple con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) y soporta el lenguaje común de acceso a los datos: SQL. Es multiplataforma y posee buenas interfaces de instalación y administración. Aproxima los datos a un modelo Objeto-Relacional, y es capaz de manejar completas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones y optimización de consultas.

Está basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre (Open Source) de este proyecto, y utiliza el lenguaje SQL92/SQL99. Fue el pionero en muchos de los conceptos existentes en el Sistema Objeto-Relacional actual, incluido más tarde en otros sistemas de gestión comerciales. Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores e integridad transaccional. Lleva más de una década de desarrollo, siendo hoy en día un sistema bastante avanzado, que tiene soporte nativo para los lenguajes de programación: C, C++, Java, Python, PHP y muchos más. Se encuentra bajo la licencia BSD (Berkeley Software Distribution).

Entre otras características que presenta se encuentran las siguientes:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits.
- Incorpora una estructura de datos arreglo.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, entre otras.

- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Selección del Sistema Gestor de Base de Datos

Después del estudio realizado de los diferentes gestores de base de datos más usados, se determina utilizar PostgreSQL, aunque Oracle se encuentra entre los mejores gestores de base de datos en la actualidad, es un software propietario y con un costo muy elevado. Por el contrario, PostgreSQL es libre, es un gestor de base de datos objeto-relacional altamente extensible, soporta operadores y tipos de datos definidos por el usuario; es capaz de manejar complejas rutinas y reglas de modo que, su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas. Puede manejar un ilimitado número de base de datos y permite una fácil gestión de los usuarios. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. También es preciso destacar que presenta prestaciones muy similares a las ofrecidas por Oracle, pero con un consumo de recursos inferior (espacio en disco y memoria), lo que hace que el rendimiento de la aplicación no se vea afectado en forma considerable en caso de implementar el SGBD y la aplicación en la misma máquina.

Se hace uso del gestor de base de datos PostgreSQL en su versión 9.1

1.4.5 Lenguajes de Programación

Lenguaje de programación en el lado del servidor

En la actualidad existe una fuerte tendencia al desarrollo de aplicaciones web favorecido por el desarrollo de Internet. Un aspecto importante de este tipo de aplicaciones lo constituye el lenguaje que se emplea en el servidor, que es el encargado de ejecutarse en el mismo y del cual los usuarios obtienen el beneficio del procesamiento de la información. En el presente coexisten gran número de lenguajes por lo que un equipo de desarrollo puede optar para realizar esta función, pero hay tres particularmente que marcan la delantera en este sentido, atendiendo a la popularidad que presentan dentro de la comunidad de desarrollo de software, el cúmulo de sistemas que han sido desarrollados sobre ellos, además de los resultados que se han obtenido mediante la explotación de estas aplicaciones, evidenciando su robustez y

dominio sobre los demás, ellos son: C++ desarrollado y estandarizado por Microsoft que soporta el paradigma orientado a objetos, PHP diseñado originalmente para la creación de páginas web dinámicas siendo además multiparadigma y finalmente Java (20).

JAVA

Java es un lenguaje de programación desarrollado por Sun Microsystems⁵. Sirve para crear todo tipo de aplicaciones (locales, intranet o internet). El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Es orientado a objetos, soporta las tres características de la programación orientada a objetos: la encapsulación, la herencia y el polimorfismo. Es un lenguaje independiente de la plataforma, es uno de los escasos programas que puede ser transportado de sistema operativo, computadora o entorno sin la necesidad de cambiar su código, por lo que ha sido concebido desde sus orígenes como un lenguaje capaz de producir código totalmente transportable. Es robusto, ya que realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución.

PHP

PHP es un lenguaje de programación generalmente usado en la programación de sitios web dinámicos y actualmente es casi el lenguaje de desarrollo de sitios más usado en todo el mundo. Fue originalmente creado por Rasmus Lerdorf para presentar su portafolio de trabajo en el año 1994. Originalmente fue desarrollado en Perl. PHP al principio significaba Personal Home Page pero con el tiempo como ya es desarrollado por otro grupo se llama PHP Hypertext Preprocessor.

PHP es un lenguaje interpretado, solo se necesita un navegador web para ejecutarlo. Es un lenguaje del lado del servidor, por lo que los script se ejecutan remotamente y el resultado aparece en la máquina cliente (local). Tiene soporte para muchos tipos de bases de datos, entre las principales están MySQL, PostgreSQL, SQLite, Oracle, entre otras. La sintaxis es parecida a la del lenguaje C (Que también tiene un parecido a Perl). Es embebido en código HTML y presenta soporte de orientación a objetos

C++

El C++ es un lenguaje híbrido, que permite programar tanto en estilo procedimental (como si fuese C), como en estilo orientado a objetos, como en ambos a la vez. Este lenguaje se puede emplear mediante

programación basada en eventos para crear programas que usen interfaz gráfica de usuario (21). C++ es a la vez un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con él. Como lenguaje orientado a objeto se basa en la filosofía completamente diferente, que exige del programador un completo cambio de mentalidad.

Entre las características que C++ posee se encuentran:

- C++ es un lenguaje versátil, potente y general.
- C++ presenta una gran versatilidad es decir, es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- Es un lenguaje que está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- El C++ es uno de los lenguajes más rápidos en cuanto a ejecución.

El C++ una de las desventajas que presenta es que obliga a hacer todo el trabajo manualmente al igual que el lenguaje de programación C, lo que provoca a los desarrolladores dificultades en su aprendizaje. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original.

Selección del lenguaje del lado del servidor a utilizar.

Después de haber analizados estos tres lenguajes de programación se determinó usar JAVA, primeramente por la experiencia y preparación del equipo de desarrollo en dicho lenguaje, además por su integración con la plataforma de programación a emplear JEE, ya que la misma define estándares para desarrollar y ejecutar aplicaciones con este lenguaje de programación, basándose además en los resultados obtenidos por el proyecto SIGEP (Sistema de Gestión Penitenciaria), el cual usó este lenguaje. Además de que es orientado a objetos, soporta las tres características de la programación orientada a objetos: la encapsulación, la herencia y el polimorfismo. Con la comprobación de tipos, Java ayuda a detectar errores lo antes posible, en el ciclo de desarrollo. Además, obliga a la declaración explícita de métodos, reduciendo al mínimo las posibilidades de error. Por otra parte, es dinámico, se adapta con suma facilidad a los cambios en los componentes fundamentales del lenguaje.

1.4.6 Plataforma de programación a emplear

Plataforma JEE

JEE (en español: Java Edición Empresarial) es una plataforma de programación, que define estándares para desarrollar y ejecutar aplicaciones en el lenguaje de programación Java, empleando arquitecturas que definen un modelo multicapa y que se apoyan en componentes de software modulares. JEE incluye tecnologías, tales como Servlets, JSP (en inglés: Java Server Pages) y varias tecnologías de servicios web. Las aplicaciones desarrolladas en esta plataforma tienden a ser portables, escalables, robustas y seguras a la vez que son integrables con tecnologías anteriores (22).

1.4.7 Lenguajes en el lado del cliente

JavaScript

Es un lenguaje interpretado, es decir que no requiere compilación y se utiliza comúnmente para la construcción de páginas web en combinación con el XHTML. JavaScript no es orientado a objetos debido a que no soporta la herencia de objetos, sino que está basado en objetos que incorpora para su funcionalidad, aunque permite la creación de objetos propios. Se caracteriza por ser un lenguaje manejado por eventos por el hecho de responder a eventos generados, ya sea por el usuario o por el navegador, es independiente de la plataforma debido a que solo se necesita un navegador para ejecutar el código, permite un desarrollo rápido y es relativamente fácil de aprender. Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape y Mozilla Firefox.

XHTML:

XHTML (en inglés: Extensible Hypertext Markup Languaje) es el lenguaje de marcado pensado para sustituir a HTML (en inglés: Hypertext Markup Languaje), es una reformulación del mismo que es compatible con XML (en inglés: Extensible Markup Languaje) Se utiliza para generar documentos y contenidos de hipertexto generalmente publicados en la WEB. El documento se escribe en forma de etiquetas, que hasta cierto punto pueden establecer la apariencia del documento, aunque en la actualidad se suele mezclar con lenguajes script como JavaScript, para lograr mayor interacción con los usuarios (23).

1.4.8 Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP sus siglas vienen de Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto en español) o el protocolo HTTPS (la versión cifrada y autenticada). El protocolo HTTP pertenece a la capa de aplicación del modelo OSI (en inglés: Open Systems Interconnection, o Interconexión de Sistemas Abiertos por sus siglas en español) y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música (24).

Servidor Web Apache Tomcat

Es un servidor web multiplataforma que funciona como contenedor de servlets y que se desarrolla bajo el proyecto denominado Jakarta perteneciente a la Apache Software Foundation bajo la licencia Apache 2.0 y que implementa las especificaciones de los servlets y de Java Server Pages o JSP de Sun Microsystems. Dicho servidor es mantenido y desarrollado por miembros de la fundación y voluntarios independientes, los cuales tienen libre acceso al código fuente.

Dado que dicho producto fue desarrollado en Java, éste puede ejecutarse sobre cualquier sistema operativo, previa instalación de la máquina virtual de Java aunque, también se puede usar con MAMPP (Mac OS X), LAMPP (GNU/Linux), WAMPP (Windows) o XAMPP (cualquier sistema operativo). Además, puede funcionar como servidor web por sí mismo, sin embargo en sus inicios se pensaba que dicho servidor era recomendable usarse en entornos de desarrollo con requisitos mínimos de velocidad. En la actualidad no existe esta percepción y por esto, es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Principales características que soporta la última versión estable son: autenticación de acceso básico, negociación de credenciales, HTTPS, alojamiento compartido, interfaz de entrada común, Servlets de Java, consola de administrador (25).

Servidor Web Internet Information Server (IIS)

Internet Information Server (IIS) es el servidor de páginas web avanzado de la plataforma Windows. IIS incluye una serie de servicios para los ordenadores que funcionan con Windows, como son: FTP y HTTP/HTTPS, convirtiendo un ordenador en un servidor de Internet, es decir, que las computadoras que tiene este servicio instalado pueden publicar páginas web, tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que dan la capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. Del mismo modo, pueden ser incluidos los de otros fabricantes, como PHP o Perl. IIS es compatible con un gran número de interfaces que pueden utilizarse para desarrollar otras características un sitio Web.

Internet Information Server incluye los siguientes componentes:

- Servicios de Internet: WWW, FTP.
- Administrador de servicios de Internet, la herramienta para administrar los servicios de Internet.
- Conector de bases de datos de Internet, el componente para enviar consultas a bases de datos.
- Administrador de claves, la herramienta para la instalación de las claves de Secure Sockets Layer (SSL).

Internet Information Server posee la autenticación implícita, que permite a los administradores autenticar a los usuarios de forma segura a través de servidores de seguridad y proxy. Microsoft ha mejorado fundamentalmente su software en el campo de los servicios Web. Los avances vienen motivados sobre todo por la seguridad y el rendimiento pero todavía padece de algunas brechas de seguridad. IIS solo puede operar en plataformas Windows.

Selección del servidor web a utilizar.

A partir de un estudio realizado, se escogió como Servidor Web Apache Tomcat, primeramente por su aceptación universal y gran potencialidad, además de experiencia que se tiene en el trabajo con este servidor web, por otra parte a diferencia del IIS puede ser ejecutado en cualquier plataforma de trabajo. Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, es una tecnología gratuita de código fuente abierta, se selecciona además debido precisamente a que la plataforma de desarrollo que se utiliza es Java y se complementan, debido a que fue desarrollado en dicha plataforma.

Se hace uso del servidor web ApacheTomcat en su versión 7.0.23.

1.4.9 Marco de trabajo

Un marco de trabajo es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto (26).

En el contexto actual de la industria de software se cuenta con grupo considerable de marcos de trabajos que facilitan y agilizan el desarrollo de sistemas. Específicamente para los lenguajes seleccionados con anterioridad existen un conjunto de ellos que se emplean con mucha frecuencia. Para lo referente al trabajo de acceso a datos se pueden mencionar algunos marcos de trabajo de elevado prestigio como Athena Framework, Ebean ORM e Hibernate Framework. Por otro lado están un conjunto de marcos de trabajo que hacen posible la aplicación del patrón MVC; propuesta de la arquitectura base, que poseen una elevada aceptación a nivel mundial, entre ellos: Struts, JBoss Seam y Spring Framework, vale destacar que los dos últimos presentan un marcado uso en la universidad.

De igual manera para la lógica de presentación se utilizan marcos de trabajos, debido a la importancia que esta posee en los sistemas informáticos, teniendo en cuenta su interacción con los usuarios. Diversas son las librerías desarrolladas para JavaScript, que enriquecen la interfaz de usuario y brindan un ambiente de trabajo más limpio en el uso de Ajax, entre ellos destacan Qooxdoo, Ext JS y Dojo Toolkit.

Marco de trabajo Spring

Del lado del servidor existen muchas tecnologías que facilitan el desarrollo de aplicaciones, entre las más populares se encuentran: Struts, JSF y Spring (27).

El marco de trabajo Struts brinda soporte para el desarrollo de aplicaciones web bajo el patrón MVC para la plataforma JEE. Struts permite reducir el tiempo de desarrollo ya que implementa un solo controlador (ActionServlet) que evalúa las peticiones del usuario mediante un archivo configurable (struts-config.xml).

El marco de trabajo JSF es un estándar para aplicaciones web basadas en Java, facilita la construcción de aplicaciones siguiendo el patrón MVC, modelo de componentes orientado a objetos, desarrolladores de componentes, desarrolladores de lógica de aplicación, uso de simples clases java como controladores,

fácil incorporación de potencialidades AJAX, posee un conjunto prefabricado de componentes de interfaz de usuario y modelo de programación orientado a eventos (28).

Spring es un marco de trabajo de código abierto de desarrollo de aplicaciones para la plataforma Java. Por su diseño ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar. Está diseñado como una serie de módulos que pueden trabajar independientemente uno de otro. Además, intenta mantener un mínimo acoplamiento entre la aplicación y el propio framework de forma que podría ser desvinculada de él sin demasiada dificultad.

Entre las tecnologías anteriormente descritas se seleccionó Spring como herramienta de desarrollo debido a que posee una comunidad de gran prestigio a nivel internacional (<http://www.springsource.org/>) que facilita una buena retroalimentación, además por la experiencia de los desarrolladores en el trabajo con dicho marco de trabajo.

Se hace uso del marco de trabajo Spring en su versión 3.0.

Dojo Toolkit

Dojo Toolkit es una colección de scripts estáticos que permiten el desarrollo de aplicaciones web enriquecidas en el cliente, incorpora soporte para el trabajo con la tecnología AJAX (en inglés: Asynchronous JavaScript and XML). Se destaca por permitir desarrollar aplicaciones que pueden funcionar con independencia del navegador web donde se ejecuten, hace transparente el desarrollo para diferentes implementaciones del DOM, ofrece soporte para la internacionalización e incluye un gran cúmulo de componentes visuales basados en XHTML, CSS y JavaScript para enriquecer la interfaz de usuario. Presenta una arquitectura modular donde destacan los módulos: Dojo, Dijit y Dojox (29).

Se hace uso de Dojo Toolkit en su versión 1.8.

Mapeador de Objeto Relacional (ORM)

Estos marcos de trabajo se utilizan para el trabajo con el acceso a datos de una aplicación, entre ellos se encuentran IBATIS, Spring JDBC e Hibernate que son soportados por la plataforma JEE.

IBATIS toma los mejores atributos e ideas de las tecnologías ORM y Data Mapper (Mapeador de datos) encontrando un equilibrio entre ellas, convirtiéndose así en una solución híbrida. Algunas de las ideas que toma son: el soporte para procedimientos almacenados, SQL dinámico, SQL en línea y el Mapeo Objeto Relacional.

Spring JDBC es un módulo perteneciente al framework Spring. El mismo posee algunas librerías de clases para trabajar con base de datos a través del API de Java JDBC. Unas de las características principales de este módulo es el soporte que brinda para invocar procedimientos y funciones almacenadas.

Hibernate es un marco de trabajo Java que permite agilizar el desarrollo en la Capa de Datos en aplicaciones JEE. (30) Hibernate es una solución ORM para el lenguaje de programación Java, concebido bajo la filosofía del código abierto. Busca solucionar el problema de la diferencia entre el modelo de objetos y el modelo relacional, realizando un mapeo entre tablas de la base de datos y los objetos del negocio a través de archivos declarativos, en este caso archivos XML. Soporta la conexión a una gran variedad de servidores de base de datos, como PostgreSQL, Oracle, SQLServer y otros, permite además adaptarse a una base de datos ya existente, así como generar la base de datos a partir de un modelo objetual. Entre sus funciones se encuentran: permitir transacciones, asociaciones, polimorfismo, herencia, carga mediante referencia, persistencia transitiva (31).

Fue seleccionado Hibernate principalmente por la sólida integración con el framework Spring facilitando el trabajo ya que este libera en gran medida al programador de mantener el control sobre las sesiones. Ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria") que permite crear consultas mediante la manipulación de los criterios de los objetos en tiempo de ejecución. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Se hace uso del marco de trabajo Hibernate en su versión 3.5.

1.5 Arquitectura definida para el sistema

Uno de los elementos clave en todo proceso de desarrollo de software es la definición de la Arquitectura. Esta representación eleva el nivel de abstracción, facilitando así la comprensión de sistemas de software complejos, así mismo hace que aumenten las posibilidades de reutilizar tanto la arquitectura como los componentes que aparecen en ella. En la medida que sea concebida la arquitectura basada en los principios de cohesión, utilidad y flexibilidad de los componentes se obtendrá un mejor acabado del producto.

“La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán y los principios que orientan su diseño y evolución” (32).

1.5.1 Arquitectura Multicapa

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica; facilita de una manera muy efectiva la separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada. El estilo de arquitectura basado en capas se identifica por las siguientes características:

- Describe la descomposición de servicios, de forma que la mayoría de la interacción ocurre solamente entre capas vecinas. Las capas de una aplicación pueden residir en la misma máquina física (misma capa) o puede estar distribuido sobre diferentes computadores (n-capas).
- Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas. Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella.

Un ejemplo de la misma lo constituye la Arquitectura de Tres Capas, que se empleará en el desarrollo del sistema. A continuación se presentan los principales componentes que conforman la arquitectura del sistema, que tiene como primer elemento el desarrollo bajo la tecnología Java Enterprise Edition. La arquitectura base está compuesta por tres capas: Presentación, Negocio y Acceso a Datos:

- **Capa de presentación:** la cual estaría centrada en la presentación al usuario que va a usar la aplicación, conformada por una interfaz de usuario desarrollada con el marco de trabajo Dojo.
- **Capa de lógica de negocio:** contendría el núcleo de la aplicación con sus principales módulos y funcionalidades.
- **Capa de Acceso a datos:** en esta capa se encuentran las operaciones que permiten la interacción con el gestor de base de datos desde la aplicación. Desde aquí se ejerce la conexión con el gestor de base de datos, permitiendo así la persistencia y el acceso a la información.

1.6 Patrones

En el mundo del desarrollo del software se ha hecho muy común el uso de los patrones, ya que su empleo brinda a los equipos de desarrollo un arma que agiliza el diseño del sistema, debido a que establecen

soluciones a problemas particulares del diseño, facilitan la reutilización del código y permiten una fácil comprensión debido a la documentación estándar que presentan. Una definición para el término patrón sería:

Un patrón de diseño es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios, una estructura de implementación que logra una finalidad determinada, manera práctica de describir aspectos de la organización de un programa (33).

Según el libro “UML y Patrones” del autor *Craig Larman* publicado por la editorial *Prentice Hall* en el año 1999, se definen dos grupos de patrones:

Patrones GRASP

Los patrones generales para asignar responsabilidades del inglés GRASP (General Responsibility Assignment Software Patterns). Abarca cinco patrones principales: experto, creador, bajo acoplamiento, alta cohesión y controlador. Algunos de estos serán utilizados para contribuir a un mejor desarrollo del sistema.

Patrón Experto: Este patrón consiste en asignar una responsabilidad al experto en información, es decir, a la clase que cuenta con la información necesaria para cumplir con la responsabilidad. Se refuerza el encapsulamiento y se favorece el bajo acoplamiento.

Patrón Creador: El patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Su propósito principal es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Brinda soporte al bajo acoplamiento.

Patrón Bajo Acoplamiento: Pretende asignar una responsabilidad para mantener el bajo acoplamiento, es decir, el diseño de clases más independientes, que no se relacionen con muchas otras, que reducen el impacto de los cambios, que son más reutilizables y acrecientan la oportunidad de una mayor productividad.

Patrón Alta Cohesión: Su objetivo es asignar una responsabilidad, de modo que la cohesión siga siendo alta. Las clases con alta cohesión se caracterizan por tener responsabilidades estrechamente

relacionadas y no realizar un trabajo enorme. Una clase con alta cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño.

Patrón Controlador: Consiste en asignar la responsabilidad a una clase de manejar los mensajes correspondientes a eventos de un sistema. El controlador es un intermediario entre la interfaz de usuario y el núcleo de las clases donde reside la lógica de la aplicación. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a la vez tener un mayor control.

Patrones GoF

Por sus siglas en inglés Gang of Four. Llamados así debido a que fueron cuatro autores los que escribieron el libro "Design Patterns" que ilustra sus funciones.

Patrón Fachada: Patrón estructural que trata de simplificar la interfaz entre dos sistemas o componentes de software ocultando un sistema complejo detrás de una clase que funciona como pantalla o fachada. La idea principal es la de ocultar todo lo posible la complejidad de un sistema, el conjunto de clases o componentes que lo definen, de forma que solo se ofrezca un punto de entrada al sistema cubierto por la fachada. Su uso aísla los posibles cambios que se puedan producir en alguna de las partes.

Patrón Mediador: Patrón de comportamiento que coordina las relaciones entre sus asociados. Permite la interacción de varios objetos, sin generar acoples fuertes en esas relaciones.

Patrón Cadena de Responsabilidad: La cadena de responsabilidad se encarga de evitar el acoplamiento del remitente de una petición a su receptor, dando a más de un objeto la posibilidad de manejar la petición.

Patrón Singleton: Patrón creacional diseñado para restringir la creación de objetos pertenecientes a una clase. Su objetivo es garantizar que una determinada clase solo tenga una instancia y proporcionar un punto de acceso global a esta. Es un patrón muy sencillo de diseñar y a menudo es implementado por otros patrones.

Conclusiones del capítulo

En el presente capítulo se valoró que los sistemas estudiados no son capaces de gestionar eventos y notificaciones de manera centralizada, organizada y distribuida, siendo esta una deficiencia que pretende solucionar este trabajo, con el desarrollo de las aplicaciones propuestas, a pesar de ello, los mismos sirvieron de guía para aplicar sus principales ventajas en el desarrollo del sistema. Igualmente se hace una descripción de las herramientas de desarrollo usadas y el porqué de su elección.

Como conclusión parcial se destaca el uso de SXP como Metodología Ágil de desarrollo de Software, ya que la misma agilizará el proceso de desarrollo de la aplicación, permitirá ajustar el proceso productivo al tiempo disponible evitando la generación documental innecesaria y propone que el equipo que llevará adelante la tarea sea pequeño, concordando con la realidad. PostgreSQL 9.1 como Gestor de Base de Datos, ya que permite manejar grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la misma vez al sistema.

La utilización correcta de los marcos de trabajo, lenguajes y herramientas de desarrollo anteriormente definidos, facilitarán entre otros aspectos el desarrollo del sistema, con el menor costo posible y la obtención de un producto como resultado final con la calidad requerida.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se realiza el desarrollo del sistema utilizando la metodología ágil SXP. Se definen los requisitos del mismo, las descripciones de las historias de usuarios, el diagrama de historia de usuario, el diagrama de clases, el diagrama de paquetes, así como otros modelos auxiliares a utilizar durante el desarrollo del sistema.

2.2 Planificación por roles para la realización del sistema.

Rol	Responsabilidad del Rol	Nombre y Apellidos
Gerente	Es el encargado de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el desarrollo del sistema	Yanier Alba Reyes
Cliente	El cliente participa en las tareas que involucran la lista de reserva del producto.	Yulier Matías León
Programador	Es el encargado de producir el código y escribir las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.	Yanier Alba Reyes
Analista	Es el encargado de escribir las historias de usuario y las pruebas funcionales para validar su implementación.	Yanier Alba Reyes Yulier Matías León
Diseñador	Encargados del diseño del sistema; así como el de los prototipos de interfaces, máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción.	Yanier Alba Reyes Andy José Rivas

Probador	Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo.	Yanier Alba Reyes
Arquitecto	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas.	Yanier Alba Reyes

2.3 Actores del Sistema

Actores	Descripción
Usuario	Usuarios que acceden al sistema, que tienen la posibilidad de gestionar canales, eventos y suscripciones, definir asistencia a evento y obtener notificación de los mismos.

2.4 Diagrama de Historia de Usuario

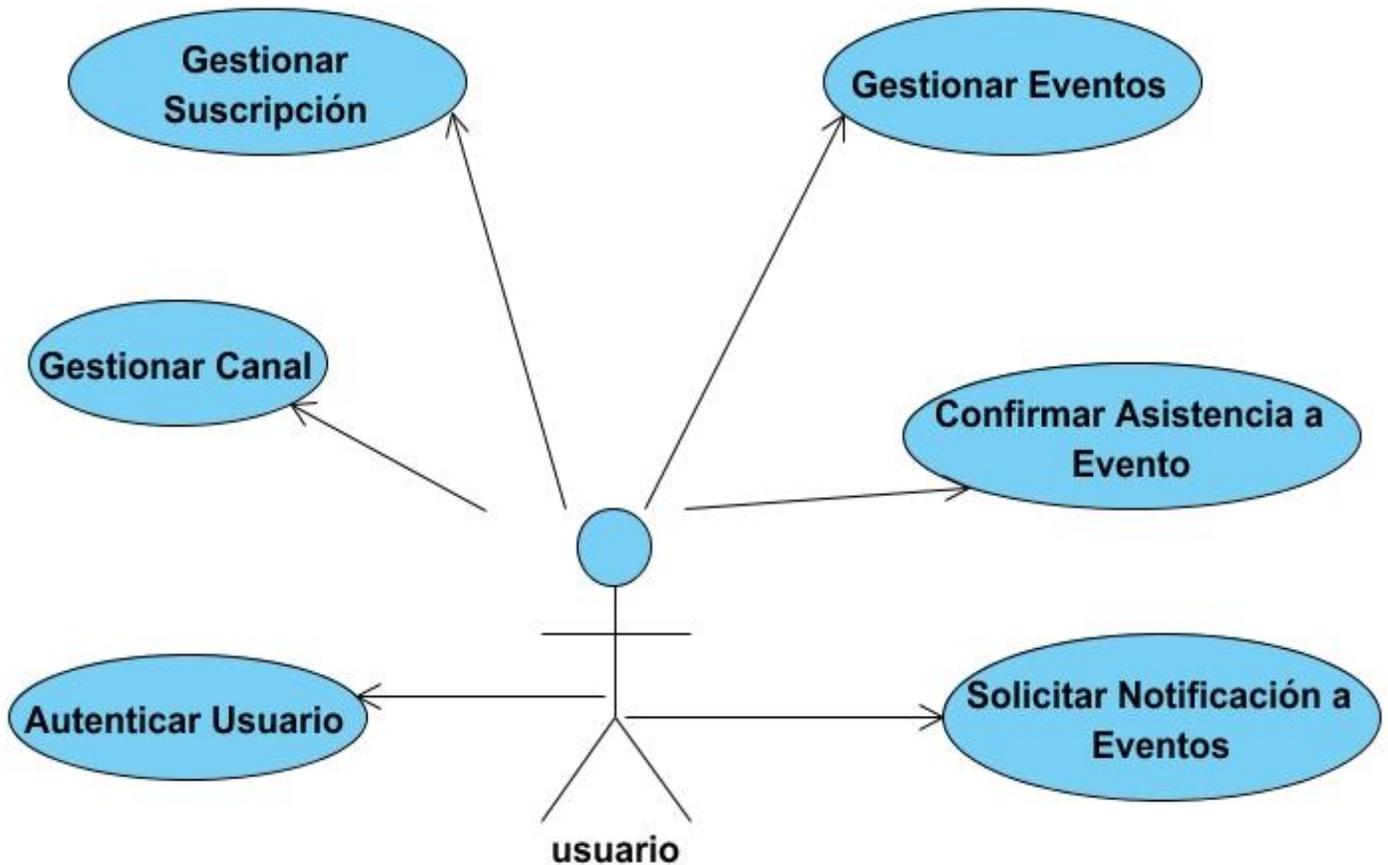


Imagen 1: Diagrama de historia de usuario.

2.5 Lista de Reserva del Producto (LRP)

Unos de los artefactos más importantes definidos en la metodología SXP es la Lista de Reserva del Producto (LRP), en la cual se recoge en una lista priorizada todo el trabajo a desarrollar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para una iteración. Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente, con la restricción de que sólo puede cambiarse entre iteraciones.

Prioridad	Ítem	Descripción	Estimación	Estimado por
Muy Alta	1	Caracterización de las herramientas y metodología de desarrollo a utilizar para la realización del sistema a desarrollar.	2 semana	Yanier
Media	2	Autenticar Usuario.	1 semana	Yanier
Alta	3	Gestionar Canales.	3 semana	Yanier
Alta	4	Gestionar Eventos.	3 semana	Yanier
Media	5	Gestionar Suscripción.	1 semana	Yanier
Alta	6	Solicitar Notificación a Evento.	3 semana	Yanier
Baja	7	Definir Asistencia a Evento	1 semana	Yanier

2.6 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad (34).

Usabilidad

- El sistema puede ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

Soporte

- Realizar pruebas, mantenimiento e instalaciones necesarias para lograr el mejoramiento y evolución con el transcurso del tiempo.

Seguridad

- El sistema garantizará la autenticación como primera acción para el acceso a la aplicación; esta consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.

2.7 Historias de Usuarios y Tareas de Ingeniería

Las historias de usuario son la técnica utilizada en SXP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son utilizadas como el único documento de requisitos que se genera. Son escritas en lenguaje natural, sin un formato predeterminado, no excediendo su tamaño de unas pocas líneas de texto (13).

Las tareas de ingeniería son un conjunto de acciones a desarrollar para resolver las historias de usuario. Permiten organizar el proceso de implementación además de posibilitar que sea conocido el grado de complejidad de cada historia de usuario teniendo en cuenta la cantidad de tareas asociadas a ella.

A continuación se dan a conocer las distintas historias de usuarios que están presentes en el desarrollo del sistema y las tareas de ingeniería asociadas a cada historia de usuario.

Tabla 1 HU Autenticar usuario

Historia de Usuario	
Número:01	Nombre Historia de Usuario: Autenticar usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yanier Alba Reyes	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1 semana.
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El usuario inserta sus datos (usuario y contraseña), el sistema verifica y otorga el acceso al sistema con los permisos correspondientes a dicho usuario.	
Observaciones:	

Tabla 2 HU Solicitar notificación a evento

Historia de Usuario	
Número:02	Nombre Historia de Usuario: Solicitar Notificación a Evento.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yanier Alba Reyes	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3 semanas.
Riesgo en Desarrollo: Alto	Puntos Reales: 3
Descripción: El sistema debe ofrecer la opción de mostrar al usuario un recordatorio de los eventos a los cuales está suscrito, mediante un sistema externo.	
Observaciones:	

Tabla 3 HU Gestionar eventos

Historia de Usuario	
Número:03	Nombre Historia de Usuario: Gestionar Eventos
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yanier Alba Reyes	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3 semanas.
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: El sistema debe ofrecer la opción de crear un nuevo evento; así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar eventos.	
Observaciones:	

Tabla 4 HU Gestionar canales

Historia de Usuario	
Número:04	Nombre Historia de Usuario: Gestionar Canales
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yanier Alba Reyes	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3 semanas.
Riesgo en Desarrollo: Alto	Puntos Reales: 3
Descripción: El sistema debe ofrecer la opción de crear un nuevo canal; así como brindar la posibilidad de poder modificar, buscar, bloquear y consultar canales.	
Observaciones:	

Tabla 5 HU Gestionar suscripción

Historia de Usuario	
Número:05	Nombre Historia de Usuario: Gestionar Suscripción
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yanier Alba Reyes	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1 semana.
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de suscribirse a un determinado canal para obtener toda la información de los eventos asociados al mismo, así como desuscribirse.	
Observaciones:	

Tabla 6 HU Definir asistencia a evento

Historia de Usuario	
Número:06	Nombre Historia de Usuario: Definir Asistencia a Evento
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Yanier Alba Reyes	Iteración Asignada: 3
Prioridad en Negocio: Baja	Puntos Estimados: 1 semana.
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
Descripción: El sistema debe ofrecer al usuario la opción de asistir o no a un determinado evento.	
Observaciones:	

Tabla 7 Tarea de ingeniería 1

Tarea de Ingeniería	
Número Tarea: 01	Número Historia de Usuario: 01
Nombre Tarea: Implementar código del Autenticar usuario.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23/03/2013	Fecha Fin: 28/03/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: El usuario accede al sistema, a través de la web entra al sistema mediante un formulario que posee un campo para usuario y otro para la contraseña	

Tabla 8 Tarea de ingeniería 2

Tarea de Ingeniería	
Número Tarea: 02	Número Historia de Usuario: 03
Nombre Tarea: Implementar código del Adicionar evento.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 13/01/2013	Fecha Fin: 17/01/2013
Programador Responsable: Yanier Alba Reyes	
<p>Descripción: En el menú de la aplicación aparece la opción Crear Eventos. Cuando se hace “clic” en el botón se carga un formulario con los datos del evento. Luego se procede a llenar los datos correspondientes, se da aceptar y se registra el evento, luego de realizar esta opción se regresa a la página inicio de la aplicación.</p>	

Tabla 9 Tarea de ingeniería 3

Tarea de Ingeniería	
Número Tarea: 03	Número Historia de Usuario: 03
Nombre Tarea: Implementar código del Eliminar evento.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18/01/2013	Fecha Fin: 25/01/2013
Programador Responsable: Yanier Alba Reyes	
<p>Descripción: Se muestra el listado de eventos con la opción “eliminar evento”, en el caso que se elimine el sistema permite registrar esta operación en la base de datos.</p>	

Tabla 10 Tarea de ingeniería 4

Tarea de Ingeniería	
Número Tarea: 04	Número Historia de Usuario: 03
Nombre Tarea: Implementar código del Editar evento.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 26/01/2013	Fecha Fin: 1/02/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: Se muestra el listado de eventos con la opción “editar evento”, en el caso que se modifiquen los datos el sistema permite registrar esta operación en la base de datos.	

Tabla 11 Tarea de ingeniería 5

Tarea de Ingeniería	
Número Tarea: 05	Número Historia de Usuario: 03
Nombre Tarea: Implementar código del Buscar evento.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 02/02/2013	Fecha Fin: 06/02/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: En el menú de la aplicación aparece la opción buscar eventos. Cuando se hace “clic” en el botón se carga un formulario con los criterios de búsqueda de los eventos. Luego se da clic en el botón buscar y en la parte derecha se muestran los resultados de la búsqueda.	

Tabla 12 Tarea de ingeniería 6

Tarea de Ingeniería	
Número Tarea: 06	Número Historia de Usuario: 04
Nombre Tarea: Implementar código del Adicionar canal.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 11/02/2013	Fecha Fin: 15/02/2013
Programador Responsable: Yanier Alba Reyes	
<p>Descripción: En el menú de la aplicación aparece la opción Crear Canal. Cuando se hace “clic” en el botón se carga un formulario con los datos del canal. Luego se procede a llenar los datos correspondientes, se da aceptar y se registra el canal, luego de realizar esta operación se regresa a la página inicio de la aplicación.</p>	

Tabla 13 Tarea de ingeniería 7

Tarea de Ingeniería	
Número Tarea: 07	Número Historia de Usuario: 04
Nombre Tarea: Implementar código del Editar canal.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/02/2013	Fecha Fin: 27/02/2013
Programador Responsable: Yanier Alba Reyes	
<p>Descripción: Se muestra el listado de los canales con la opción “editar canal”, en el caso que se modifiquen los datos, el sistema permite registrar esta operación en la base de datos.</p>	

Tabla 14 Tarea de ingeniería 8

Tarea de Ingeniería	
Número Tarea: 08	Número Historia de Usuario: 04
Nombre Tarea: Implementar código del Buscar canal.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 1/03/2013	Fecha Fin: 8/03/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: En el menú de la aplicación aparece la opción buscar canal. Cuando se hace “clic” en el botón se carga un formulario con los criterios de búsqueda de los canales. Luego se da clic en el botón buscar y en la parte derecha se muestran los resultados de la búsqueda.	

Tabla 15 Tarea de ingeniería 9

Tarea de Ingeniería	
Número Tarea: 09	Número Historia de Usuario: 04
Nombre Tarea: Implementar código del Bloquear canal.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 1/03/2013	Fecha Fin: 8/03/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: Después de realizar la búsqueda de un canal, este debe mostrar un checkbox con la opción de Bloquear Canal, en caso de seleccionar la opción el sistema registra la operación realizada.	

Tabla 16 Tarea de ingeniería 10

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: 01
Nombre Tarea: Estudio del Protocolo Ligerero de Acceso a Directorios (LDAP).	

Tipo de Tarea : Estudio	Puntos Estimados: 1
Fecha Inicio: 16/03/2013	Fecha Fin: 22/03/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: Se profundizó en el estudio de este protocolo para realizar la autenticación de la aplicación.	

Tabla 17 Tarea de ingeniería 11

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: 05
Nombre Tarea: Implementar código del Gestionar suscripción.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10/03/2013	Fecha Fin: 15/03/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: Después de realizar la búsqueda de una canal, este debe mostrar un checkbox con la opción de suscribirse, en caso de seleccionar la opción el sistema registra la operación realizada.	

Tabla 18 Tarea de ingeniería 12

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 06
Nombre Tarea: Implementar código del Definir asistencia a evento.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10/03/2013	Fecha Fin: 15/03/2013
Programador Responsable: Yanier Alba Reyes	
Descripción: Después de realizar la búsqueda de los eventos, este debe mostrar un checkbox con la opción de “asistiré”, en caso de seleccionar la	

opción el sistema registra la operación realizada.

2.8 Plan de lanzamiento

En este paso se define el plan de lanzamiento e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha historia. Como resultado de la priorización de historias se llegó a la siguiente planificación:

Tabla 19 Plan de lanzamiento

Lanzamiento	Descripción de la Iteración	Orden de las HU a implementar	Duración Total
Iteración 1	En esta iteración se desarrollarán las historias de usuario que tienen prioridad alta.	HU_2, HU_3, HU_5	9 semanas
Iteración 2	En esta iteración se desarrollarán las historias de usuario que tienen prioridad media y baja.	HU_1, HU_4, HU_6	3 semanas

2.9 Patrones empleados

El desarrollo de un sistema conlleva, en la mayoría de las ocasiones, a darle solución a problemas muy complejos que ya alguien ha resuelto con anterioridad. Por esta razón uno de los pasos a tener en cuenta cuando se decide desarrollar un proyecto de software es identificar qué patrones pueden ser utilizados. Entiéndase por patrón como una solución estándar para un problema común de programación. A continuación se muestran los patrones aplicados en el diseño del sistema.

Patrón Arquitectónico Modelo-Vista-Controlador (MVC)

Este patrón propone dividir la aplicación en tres capas el Modelo, la Vista y el Controlador, el modelo es la representación del dominio o datos del sistema, la vista se encarga de presentar la interfaz al usuario, en sistemas web, esto es típicamente HTML, aunque pueden existir otro tipo de formatos. En la vista sólo se

deben de hacer operaciones simples y el controlador es el encargado de escuchar los cambios en la vista y enviarlos al modelo, el cual le regresa los datos a la vista como un ciclo. Cuando se aplica este patrón los accesos a la base de datos se hacen en el modelo, la vista y el controlador no deben de saber si se usa o no una base de datos. El controlador es el que decide qué vista se debe de imprimir y qué información es la que se envía (35).

Patrón Objeto de Acceso a Datos

El patrón Objeto de acceso a datos (DAO por sus siglas en inglés **Data Access Object**) resuelve el problema de contar con diversas fuentes de datos, además de que encapsula y oculta la forma de acceder a la base de datos, eliminando así la complejidad del uso de JDBC a la capa de presentación o de negocio. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los datos o de cuál es la fuente de almacenamiento (36).

Patrón Controlador: La clase controladora CanalController, constituye un ejemplo de la aplicación de este patrón, la misma tendrá en cuenta la responsabilidad de manejar los eventos que consisten en gestionar los canales en el sistema.

Patrón Experto: Se evidencia en la definición de las clases de acuerdo con las funcionalidades que deben realizar a partir de la información que manejan. Específicamente: la clase EventoDAO será la responsable de efectuar las operaciones que corresponden a la gestión de los eventos.

2.10 Diagrama de paquetes

Durante el desarrollo de software resulta muy conveniente agrupar clases y ficheros por diferentes criterios que ayudarán a una fácil comprensión de la aplicación, resultando conveniente el desarrollo de los diagramas de paquetes, los cuales muestran cómo está dividido el sistema en agrupaciones lógicas mostrando las dependencias entre estas. A continuación, en la siguiente figura se muestra la estructura y dependencia de paquetes y una explicación de la composición de cada uno. **(Ver imagen 2)**

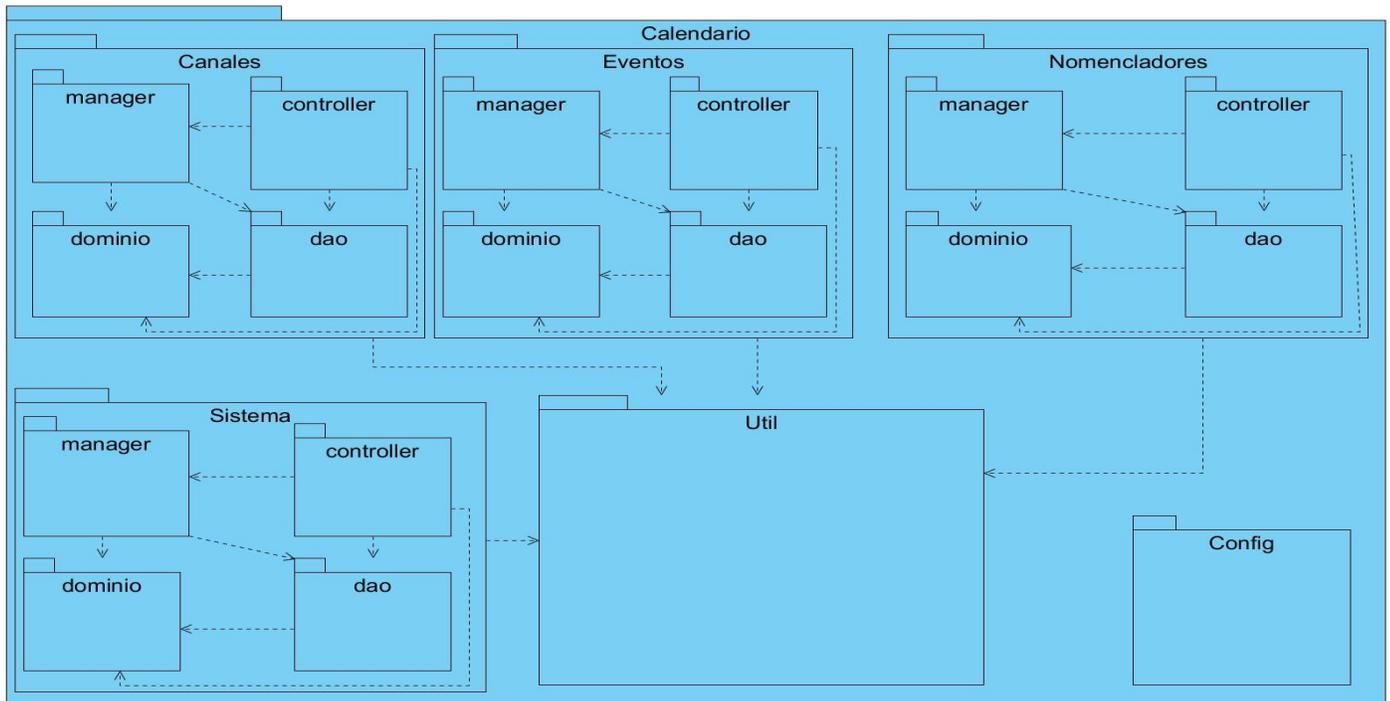


Imagen 2: Diagrama de paquetes.

Paquete config: contiene el fichero de configuración del marco de trabajo Spring en formato XML donde se añadirán todos los objetos que deberán existir en la aplicación al inicializarse la misma, contiene además la configuración para el acceso a la base de datos, así como la configuración del Protocolo Ligero de Acceso a Directorios (LDAP) para la autenticación de la aplicación.

Paquete dao: contiene las clases y las implementaciones encargadas de brindar los servicios de comunicación con el gestor de base de datos.

Paquete domain: contiene las clases relacionadas con el dominio de la aplicación, que se mapean con la base de datos usando anotaciones JPA.

Paquete controller: contiene las clases encargadas de responder a las peticiones realizadas por el cliente usando las anotaciones de Spring para mapear las url con los métodos.

Paquete Útil: contiene las clases que facilitan la comunicación entre el cliente y el servidor y el acceso a datos.

2.11 Diagrama de clases

Los diagrama de clases especifican que clases intervienen en el desarrollo del sistema y como se relacionan. Un diagrama de clases está compuesto por los siguientes elementos: Clase: atributos, métodos y visibilidad, Relaciones: Herencia, Composición, Agregación, Asociación (37).

A continuación se representa el diagrama de clases de la aplicación: **(Ver imagen 3)**

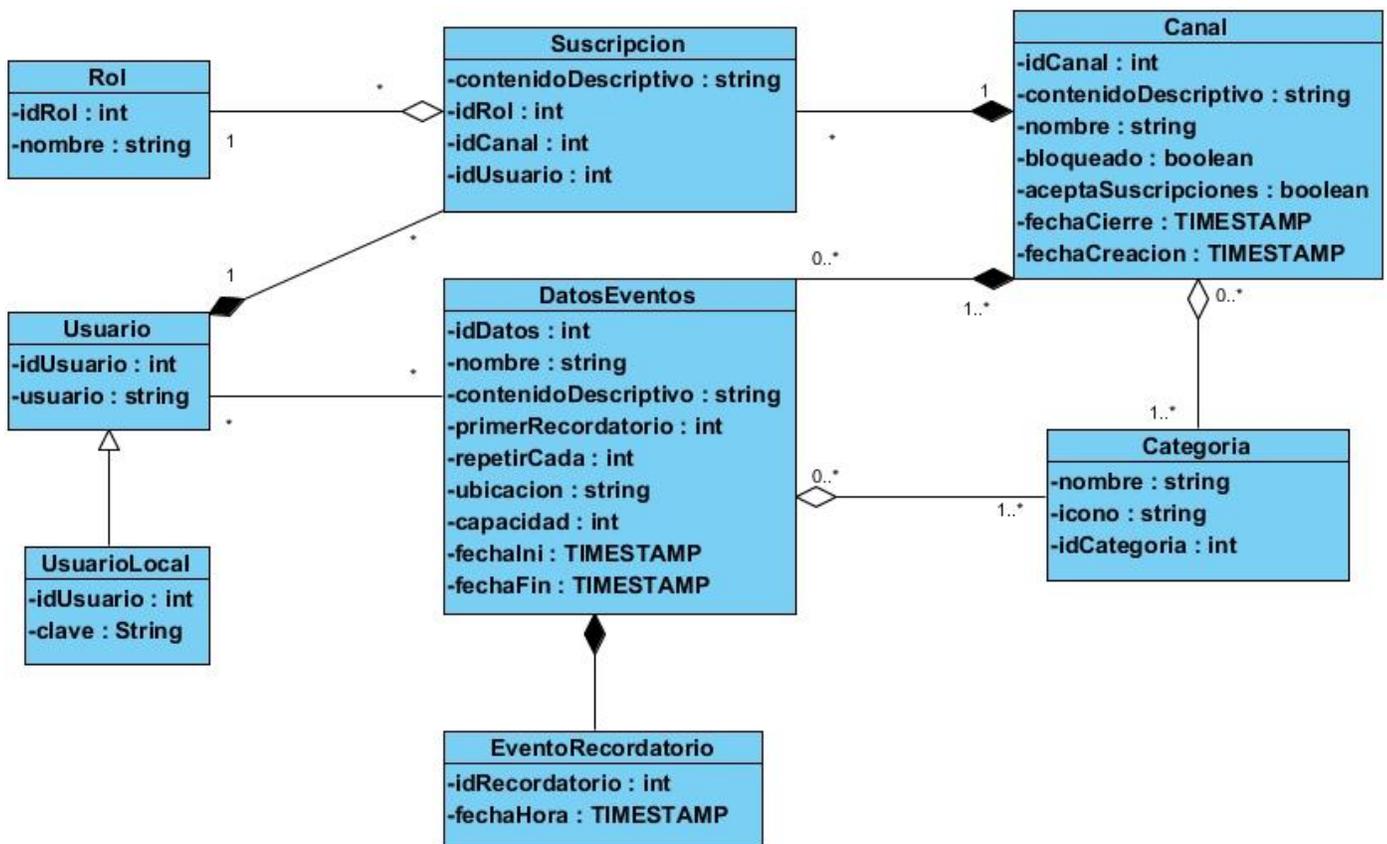


Imagen 3: Diagrama de clases.

2.12 Interfaz de usuario

En este epígrafe serán mostradas las diferentes interfaces de usuario pertenecientes a la aplicación que será desarrollada. Ver ([Anexo # 1](#), [Anexo # 2](#), [Anexo # 3](#), [Anexo # 4](#))

2.13 Conclusiones del capítulo

En este capítulo se describieron las principales características del sistema a través de los artefactos definidos por la metodología. Se definieron los requisitos funcionales y no funcionales del software a

través de la Lista de Reserva del Producto, aportándole al desarrollador una visión más clara del sistema que se desea desarrollar. Fueron definidas las distintas historias de usuarios, las mismas mantienen una relación cercana con el cliente y al ser escritas por él, en un lenguaje natural y sencillo permitirá llevar a cabo su rápida implementación, además de permitir estimar fácilmente el esfuerzo de desarrollo de las mismas. A partir de todos los artefactos descritos se puede dar paso a la construcción e implementación del sistema.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1 Introducción

En el presente capítulo se exponen las pruebas de aceptación que se le realizaron al sistema; estas fueron realizadas en cada una de las iteraciones, las cuales fueron necesarias para avanzar hacia la próxima iteración ya que para lograr un producto con calidad es necesario implementar un plan de pruebas desde el principio, y así darle seguimiento a los cambios y desarrollar iterativamente. Se especifican los estándares de codificación que serán empleados durante la implementación del sistema.

3.2 Estándares de Codificación

Los estándares de codificación se definen por el equipo de desarrollo para lograr estandarización en la programación del software. Estos se basan en la estructura y apariencia física de un programa con el fin de facilitar la lectura, comprensión, mantenimiento del código, reutilización a lo largo del proceso de desarrollo de un software y no en la lógica del programa. La generalización de aspectos tan simples como el trato de las mayúsculas, ayuda a eliminar conflictos de funcionalidades implementadas con nombres iguales y guían de forma clara el proceso de desarrollo. Los estándares de codificación por lo general son reglas que se siguen para la escritura del código fuente, de tal manera que a otros programadores se les facilite entender el código.

3.2.1 Convenciones de nomenclatura

La nomenclatura de las clases está definida por la utilización de la notación Pascal Casing, la cual define que los nombre e identificadores pueden estar compuestos por múltiples palabras juntas y la primera letra de cada palabra irá siempre en mayúsculas además se obvia el uso de artículos.

Ejemplo: TipoFrecuenciaSerie. En este caso el nombre de clase está compuesto por 3 palabras iniciadas cada una con letra mayúscula.

También se tomó en cuenta para el nombrado de las clases el tipo que esta posee, entiéndase como tipo el rol que ellas desempeñan en el sistema. A continuación se presentan las nomenclaturas organizadas por los paquetes a los que pertenecen las clases.

Controller: las clases incluidas en este paquete, después del nombre llevan el nombre del controlador de Spring del cual hereda. Ejemplo: CanalController.

Manager: las clases incluidas en este paquete, después del nombre llevan la palabra Manager. Ejemplo: EventosManager

Dao: las clases que se encuentran dentro del paquete dao se nombran adicionándoles como sufijo del nombre las siglas Dao. Ejemplo: CanalDAO

De manera general el nombre de los métodos y los atributos de las clases se escriben con la inicial del identificador en minúscula, en caso de que sea un nombre compuesto se empleará notación lowerCamelCasing, que es muy similar a Pascal Casing con la excepción de que la letra inicial comienza con minúscula. Lo mismo se aplica a los nombres de ficheros de código JavaScript y sus funciones y variables internas. Los comentarios deben ser lo más claros y precisos posibles de forma tal que se entienda el propósito de lo que se está desarrollando. En caso de ser una función complicada se debe comentar para lograr una mejor comprensión del código.

3.3 Pruebas

En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos definidos en el análisis y diseño. Las pruebas no pueden asegurar la ausencia de defectos, sólo pueden demostrar que existen errores.

En la metodología SXP las pruebas de funcionalidad se realizan a través de los casos de prueba. La metodología realiza las pruebas de aceptación que son las destinadas a comprobar que la funcionalidad implementada sea la esperada por el cliente. Las pruebas funcionales se escriben para cada historia de usuario que deba validarse.

3.3.1 Pruebas de aceptación

El objetivo de estas pruebas es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Las pruebas de aceptación son las realizadas por el cliente y usuarios finales de la aplicación. En estas serán probadas las funcionalidades exigidas por el cliente y descritas en las historias de usuario, además

de los aspectos de seguridad requeridos. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para su uso y despliegue.

Los casos de pruebas de aceptación se caracterizan por:

- Ser escritos conjuntamente con el cliente.
- Ejecutados por el equipo de desarrollo.

3.3.2 Casos de prueba de aceptación

La evaluación del caso de prueba va a estar dada en satisfactoria o insatisfactoria, para que una prueba sea satisfactoria tiene que cumplir con todas las condiciones que define el caso de prueba, que se cumpla con todo lo descrito en la prueba, que se verifiquen las condiciones de ejecución y los pasos de la misma, además que se obtengan todos los resultados esperados. En caso de que alguno de los anteriores aspectos no se cumpla el caso de prueba sería evaluado de insatisfactorio.

3.3.2.1 Plantillas de casos de prueba de aceptación

Tabla 20 Caso de Prueba de Aceptación HU1_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU1_P1	Nombre Historia de Usuario: HU1 autenticar usuario.
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario pueda acceder a las funcionalidades de la aplicación.	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución: Se inserta usuario y contraseña.	
Resultado Esperado: Se autentica correctamente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 21 Caso de Prueba de Aceptación HU3_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P2	Nombre Historia de Usuario: HU3 Gestionar Eventos.
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda crear un nuevo evento.	
Condiciones de Ejecución: Que el usuario autenticado tenga permiso para crear evento.	
Entrada / Pasos de ejecución: El usuario debe seleccionar en el menú la opción de Crear Evento Llenar los datos del formulario del crear evento.	
Resultado Esperado: Se crea un nuevo evento y el mismo se registra en la base de datos.	
Evaluación de la Prueba: Satisfactoria	

Tabla 22 Caso de Prueba de Aceptación HU3_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P3	Nombre Historia de Usuario: HU3 Gestionar Eventos
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda buscar eventos.	
Condiciones de Ejecución: Debe de existir algún evento creado.	
Entrada / Pasos de ejecución: El usuario debe seleccionar en el menú la opción Buscar Evento. Seleccionar los criterios por lo cual realizará la búsqueda	
Resultado Esperado: Se deben mostrar todos los eventos encontrados durante la realización de la búsqueda.	

Evaluación de la Prueba: Satisfactoria

Tabla 23 Caso de Prueba de Aceptación HU3_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P4	Nombre Historia de Usuario: HU3 Gestionar Eventos
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda eliminar un evento.	
Condiciones de Ejecución: Debe de existir algún evento registrado en la base de datos. El usuario debe tener permisos para realizar esta operación.	
Entrada / Pasos de ejecución: El usuario debe realizar la búsqueda de los eventos y seleccionar el evento que desea eliminar.	
Resultado Esperado: Se elimina el evento.	
Evaluación de la Prueba: Satisfactoria	

Tabla 24 Caso de Prueba de Aceptación HU3_P5

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P5	Nombre Historia de Usuario: HU3 Gestionar Eventos
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda modificar un evento.	
Condiciones de Ejecución: Debe de existir algún evento registrado en la base de datos. El usuario autenticado debe tener permisos para realizar esta operación.	

Entrada / Pasos de ejecución: El usuario debe realizar la búsqueda de los eventos y seleccionar el evento que desea modificar.
Resultado Esperado: Se actualizan los nuevos datos del evento modificado.
Evaluación de la Prueba: Satisfactoria

Tabla 25 Caso de Prueba de Aceptación HU4_P6

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P6	Nombre Historia de Usuario: HU4 Gestionar Canales.
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda crear un nuevo canal.	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución: El usuario debe seleccionar en el menú la opción Crear Canal El usuario debe de entrar los datos en el formulario del crear un nuevo canal.	
Resultado Esperado: Se crea un nuevo canal y el mismo se registra en la base de datos.	
Evaluación de la Prueba: Satisfactoria	

Tabla 26 Caso de Prueba de Aceptación HU2_P7

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P7	Nombre Historia de Usuario: HU4 Gestionar Canales
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda realizar la búsqueda de los canales creados.	
Condiciones de Ejecución: Deben de existir canales creados en la aplicación.	

Entrada / Pasos de ejecución: Deben de existir canales creados en la aplicación
Resultado Esperado: Se muestran los canales creados después de realizar la búsqueda.
Evaluación de la Prueba: Satisfactoria

Tabla 27 Caso de Prueba de Aceptación HU2_P8

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P8	Nombre Historia de Usuario: HU4 Gestionar Canales
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda modificar un canal.	
Condiciones de Ejecución: Debe de existir algún canal creado. El usuario autenticado debe tener los permisos necesarios para realizar esta operación.	
Entrada / Pasos de ejecución: El usuario debe seleccionar en el menú la opción Buscar Canal. Seleccionar los criterios por lo cual realizará la búsqueda. Seleccionar en el menú la opción Editar Canal.	
Resultado Esperado: Se actualizan los nuevos datos del canal modificado.	
Evaluación de la Prueba: Satisfactoria	

Tabla 28 Caso de Prueba de Aceptación HU3_P10

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P10	Nombre Historia de Usuario: HU3 Gestionar Eventos.
Nombre de la persona que realiza la prueba: Yulier Matías León.	

Descripción de la Prueba: Probar que el usuario autenticado pueda consultar eventos.
Condiciones de Ejecución: Debe de existir algún evento registrado en la base de datos.
Entrada / Pasos de ejecución: El usuario debe realizar la búsqueda de los eventos. Seleccionar el evento que desea consultar.
Resultado Esperado: Se muestra toda la información del evento consultado.
Evaluación de la Prueba: Satisfactoria

Tabla 29 Caso de Prueba de Aceptación HU2_P11

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P11	Nombre Historia de Usuario: HU4 Gestionar Canales
Nombre de la persona que realiza la prueba: Yulier Matías León.	
Descripción de la Prueba: Probar que el usuario autenticado pueda consultar canales.	
Condiciones de Ejecución: Debe existir algún canal creado en la base de datos.	
Entrada / Pasos de ejecución: El usuario debe realizar la búsqueda de los canales. Seleccionar el canal que desea consultar.	
Resultado Esperado: Se muestra toda la información del canal consultado.	
Evaluación de la Prueba: Satisfactoria	

3.3.3 Realización de las pruebas de aceptación

En la realización de pruebas de aceptación se llevaron a cabo dos iteraciones. En la primera iteración se evaluó un total de 17 casos de pruebas, de ellos, 13 fueron satisfactorios pues en estos no se detectaron

no conformidades, 3 fueron insatisfactorios debido a que presentaban no conformidades y 1 no fue evaluado puesto que no se encontraba en funcionamiento. En la segunda iteración, todos los requisitos fueron evaluados de satisfactorios, quedando culminada las pruebas de aceptación. **(Ver imagen 4)**

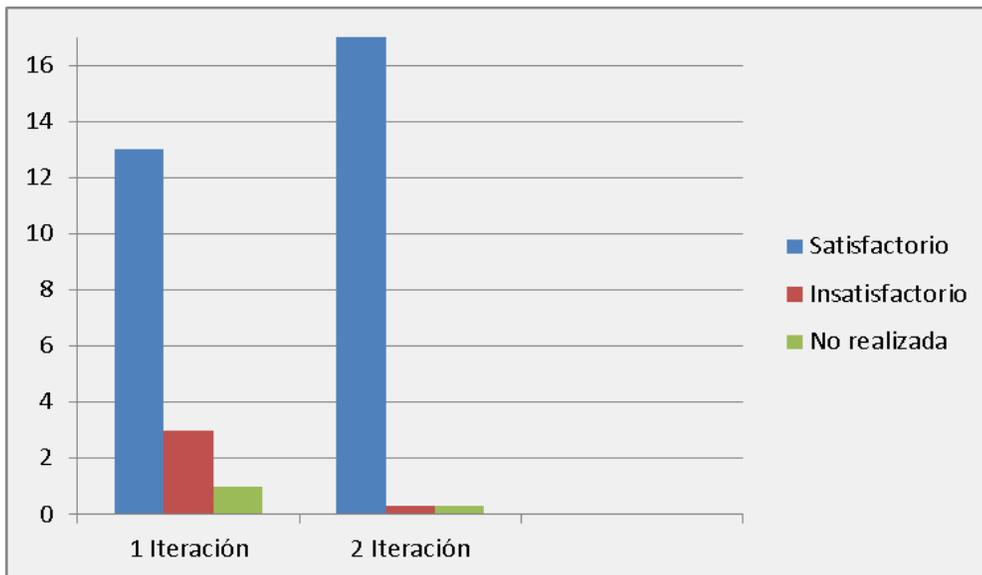


Imagen 4: Pruebas de Aceptación.

3.4 Diagrama de despliegue

El diagrama de despliegue es el artefacto generado en UML para mostrar las relaciones físicas entre los componentes hardware y software de un sistema informático. Este diagrama está compuesto por los diferentes nodos que pueden ser representaciones de estaciones clientes, servidores de aplicaciones, servidores de bases de datos y los protocolos de comunicación. A continuación se muestra el diagrama de despliegue correspondiente al sistema. **(Ver imagen 5)**

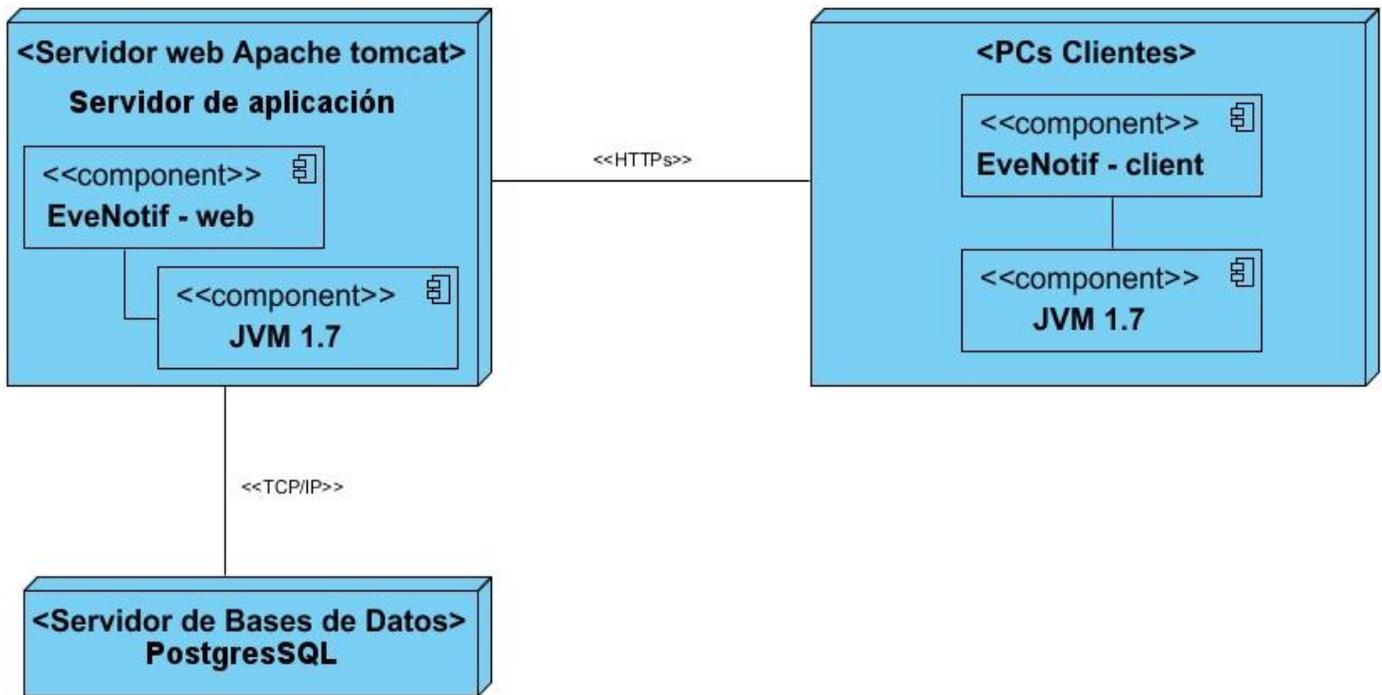


Imagen 5: Diagrama de despliegue.

3.5 Conclusiones del capítulo

En el presente capítulo se mencionaron todos los estándares de codificación por los que se rigió la implementación del sistema asegurando así, un mejor entendimiento del código fuente de la aplicación. Se valida el sistema desarrollado mediante las pruebas de aceptación, realizando casos de pruebas a todas las funcionalidades del mismo en 2 iteraciones. Este proceso permitió detectar la mayor cantidad de errores que presentaba el sistema, para darle así solución a los mismos. Las pruebas realizadas de forma general arrojaron resultados satisfactorios, siendo corregidos en tiempo y forma todos los errores encontrados.

CONCLUSIONES GENERALES

- El estudio del estado del arte mostró como resultado que ninguno de los sistemas estudiados daría solución al problema planteado. A pesar de ello, los mismos sirvieron de guía para aplicar sus principales ventajas en el desarrollo del sistema.
- Una vez realizado el levantamiento de los requisitos funcionales y no funcionales, a través de la Lista de Reserva del Producto y la confección de las Historias de Usuario, se pudo diseñar el Sistema EveNotif, permitiendo obtener los artefactos necesarios para comenzar con su construcción.
- Los casos de prueba de aceptación es unos de los artefactos generados por la metodología SXP, permitió comprobar que los requisitos implementados cumple con las especificaciones definidas por el cliente en las Historia de Usuario, mediante el diseño y la aplicación de los mismos se pudo comprobar que la aplicación desarrollada cumple con las necesidades del cliente, garantizando así resultados satisfactorios.

RECOMENDACIONES

- Agregar la funcionalidad de repetir eventos en serie, con el objetivo de que los eventos puedan repetirse automáticamente en una fecha determinada y no tener que introducirlos múltiples veces.

BIBLIOGRAFÍA CONSULTADA

1. ServiciosTic. *BeiT*. [En línea] 2013. [Citado el: 15 de enero de 2013.] <http://www.serviciostic.com/las-tic/definicion-de-tic.html>.
2. Edutec. *Revista Electrónica de Tecnología Educativa*. [En línea] julio de 2008. [Citado el: 6 de diciembre de 2011.] <http://edutec.rediris.es/Revelec2/revelec21/jmontero.htm>.
3. **León, Rolando Alfredo Hernández y Coello González, Sayda.** *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. Ciudad de la Habana : Editorial Universitaria, 2011. pág. 59. ISBN 978-959-16-3.
4. Google.dirson.com. [En línea] 2010. [Citado el: 16 de enero de 2012.] <http://google.dirson.com/o.a/google-calendar>.
5. Rainlendar. [En línea] 2011. [Citado el: 20 de enero de 2012.] <http://www.rainlendar.net/download/SkinTutorial-Spanish.pdf>.
6. Webempresa.com. *Servicio Profesionales de Joomla*. [En línea] 2013. [Citado el: 2 de marzo de 2013.] <http://www.webempresa.com/blog/item/488-crear-eventos-y-actividades-en-joomla-con-jevents-i.html>.
7. Messenger.com.es. *Todo sobre el mundo messenger*. [En línea] 10 de mayo de 2008. [Citado el: 23 de enero de 2012.] http://messenger.com.es/el-rss_441.
8. [En línea] [Citado el: 23 de enero de 2012.] http://www.badajoz.org/web/codigo/zimbra_manual_apnt.pdf.
9. **Rodríguez, Daniuska Fresneda Cruzata y Ernesto Rivero.** *Solución para el Despliegue, Implantación y Soporte del Sistema GeForza*. Ciudad de la Habana : s.n., 2010.
10. **Valverde Rebaza, Carlos Jorge y Amaro Calderón, Sarah Dámaris.** *Metodologías Ágiles*. Universidad Nacional de Trujillo. Facultad de Ciencias Físicas y Matemáticas : s.n., 2008.
11. Navegapolis.net. [En línea] 8 de abril de 2010. [Citado el: 17 de junio de 2013.] <http://www.navegapolis.net/content/view/968/49>.
12. Ecured. [En línea] 2013. [Citado el: 21 de 06 de 2013.] http://www.ecured.cu/index.php/Metodolog%C3%ADas_de_desarrollo_de_software.
13. **Romero, Gladys Marsi Peñalver.** *MA-GMPR-UR2: Metodología ágil para proyectos de software libre*. Ciudad de la Habana : s.n., 2008.
14. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Pearson Educación, 2000. 84-7829-036-2.
15. **Sosa, Yanay Hernández.** *Subsistema para la gestión de la seguridad en sistemas distribuidos*. Ciudad de la Habana : s.n., 2009.

16. **Morales, Jackeline Ortiz Fidalgo y Fredy Héctor Ríos.** *Desarrollo de un componente de software para importar reportes desde documentos Excel a Sistemas Gestores de Bases de Datos en el Centro de Investigaciones del Petróleo (CEINPET).* Ciudad de la Habana : s.n., 2010.
17. Rational Rose Enterprise. *IBM.* [En línea] 2010. [Citado el: 9 de febrero de 2012.] [http://www-142.ibm.com/software/products/es/es/enterprise/..](http://www-142.ibm.com/software/products/es/es/enterprise/)
18. ProgrammerWorld.NET. *Best Java IDE.* [En línea] 2011. [Citado el: 9 de febrero de 2012.] <http://faq.programmerworld.net/programming/best-java-ide-review.html>.
19. Oracle. *Oracle.* [En línea] 2009. [Citado el: 4 de febrero de 2012.] <http://www.oracle.com/us/index.html>.
20. **Boroto, Manuel Alejandro.** *Diseño e implementación de los subsistemas Créditos y Depósitos del sistema Quarxo.* Ciudad de la Habana : s.n., 2011.
21. **Mora, Sergio Luján.** *C++ Paso a Paso.* Alicante: *Publicaciones de la Universidad de Alicante*, 84-7908-888-5. 2006.
22. Sun Microsystems. *Sun Microsystems.* [En línea] 2010. [Citado el: 28 de febrero de 2012.] <http://java.sun.com/javaee/5/docs/tutorial/doc/>.
23. Manual de XHTML. [En línea] [Citado el: 26 de febrero de 2012.] <http://manual-xhtml.blogspot.com/2006/05/primer-documento-xhtml.html>.
24. Manuales relacionado con el desarrollo de aplicaciones web. [En línea] 2012. [Citado el: 22 de febrero de 2012.] <http://www.cibernetia.com/manuales/>.
25. Apache Tomcat. *APACHE TOMCAT.* [En línea] 2013. [Citado el: 22 de junio de 2013.] <http://apachefoundation.wikispaces.com/Apache+Tomcat>.
26. Codebox. Glosario. [En línea] 2010. [Citado el: 1 de marzo de 2012.] <http://www.codebox.es/glosario>.
27. Chapter 15. Integrating with other web frameworks. [En línea] 2010. [Citado el: 11 de diciembre de 2012.] <http://static.springsource.org/spring/docs/2.5.x/reference/web-integration.html>.
28. *Core JavaServer™ Faces.* Santa Clara, California : s.n., 2008. 978-0-13-173886-7.
29. **A.Russell, Matthew.** *Dojo the Definitive Guide.* 2008. 978-0-596-51648-2.
30. Linked in. [En línea] 2009. [Citado el: 4 de marzo de 2012.] <http://www.linkedin.com/company/global-mentoring/curso-hibernate-framework-263767/product>.
31. **Patrick Peak, Nick Heudecker.** *Hibernate Quickly.* s.l. : Manning Publications Co, 2008.
32. Desarrollo Web. Usabilidad y arquitectura del software. [En línea] 2010. [Citado el: 15 de diciembre de 2011.] <http://www.desarrolloweb.com/articulos/1622.php>.
33. **Larman, Craig.** *UML y Patrones.* 1999. ISBN 970-1 7-0261-1.

34. **IEEE Standard Glossary of Software Engineering Terminology.** Institute of Electrical and Electronics Engineers. New York: IEEE Standards Board, 1990.
35. **Lara, Rafael Bello.** *Diseño e Implementación del Subsistema Cartas de Créditos Del Proyecto SAGEB.* Ciudad de la Habana : s.n., 2010.
36. **Torrijos, Ricard Lou.** Programación en Castellano. *Catálogo de Patrones de Diseño J2EE.* [En línea] 2012. [Citado el: 11 de diciembre de 2012.] http://www.programacion.com/articulo/catalogo_de_patrones_de_diseno_j2ee_y_ii:_capas_de_negocio_y_de_integracion_243/8.
37. **Vergara, Juan Raúl.** Slideshare. [En línea] 2009. [Citado el: 20 de junio de 2013.] <http://www.slideshare.net/jrvv2009/diagramas-de-clase>.

ANEXOS

Anexo # 1: Prototipo interfaz usuario Crear Canal.

Gestionar canal

Nombre

Fecha de cierre

Acepta suscripciones
 Bloqueado

Contenido descriptivo

▼ Categorías

- campismo
- fiesta
- viajes
- tesis
- conferencias
- deportes

Anexo # 2: Prototipo interfaz usuario Crear Evento

Gestionar eventos

Difundir por Canal

► Categorías

Nombre

Fecha de Inicio Hora

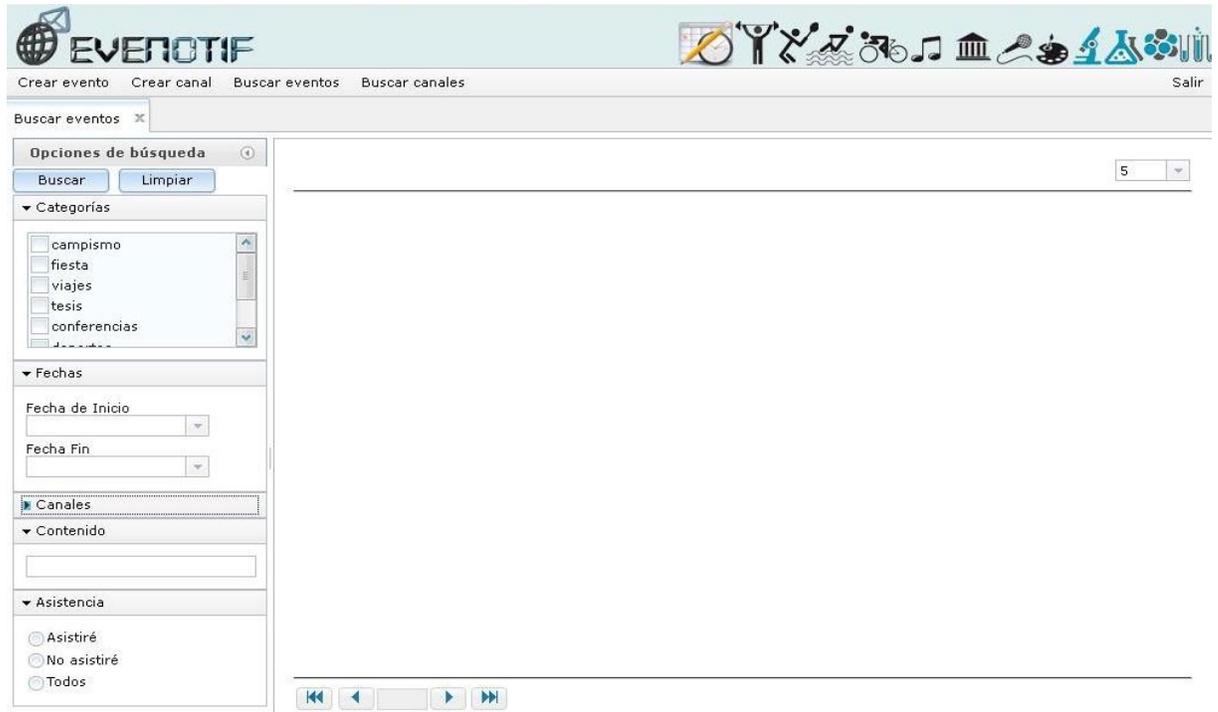
Fecha Fin Hora

Ubicación

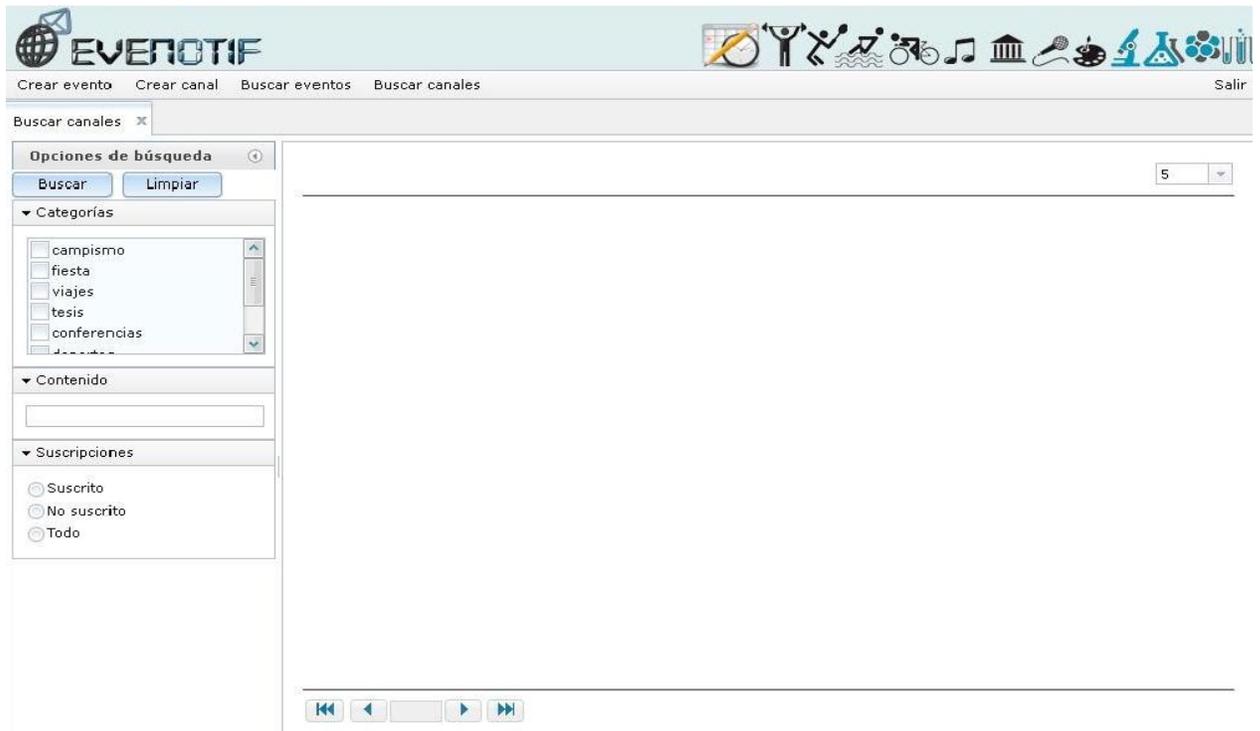
Contenido Descriptivo

Recordar antes de: Medida Repetir Cada Medida

Anexo # 3: Prototipo interfaz usuario Buscar Evento



Anexo # 4: Prototipo de interfaz de usuario Buscar Canal



GLOSARIO

API: es la abreviatura de **Application Programming Interface**. Un API no es más que una serie de servicios o funciones que se le ofrece al programador.

Plugin: es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

HQL: **Hibernate Query Language** (Lenguajes de consultas de Hibernate), es un lenguaje consultor potente y muy parecido al SQL, completamente orientado a objetos y comprende nociones como herencia, polimorfismo y asociación.

Joomla: es un Sistema de gestión de contenidos (**CMS**) que permite desarrollar sitios web dinámicos e interactivos. Permite crear, modificar o eliminar contenido de un sitio web de manera sencilla a través de un Panel de Administración.

ICal: es un estándar para el intercambio de información de calendarios.

AJAX: **Asynchronous JavaScript And XML** (JavaScript Asíncrono y XML). Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones hacen posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

CASE: es la abreviatura de **Computer Aided Software Engineering** (Ingeniería de Software Asistida por Computadora) es la aplicación de métodos y técnicas a través de las cuales las personas pueden modelar o diseñar sistemas por medio de programas, procedimientos y su respectiva documentación.

Plugin: módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

DOM: el Modelo de Objetos del Documento es una plataforma que proporciona un conjunto estándar de objetos a través de la cual se pueden crear documentos HTML y XML, navegar por su estructura y, modificar, añadir y borrar tanto elementos como contenidos.

JPA: es la abreviatura de Java Persistence API, proporciona un modelo de persistencia para mapear bases de datos relacionales en Java.

Servlets: son objetos que están presentes dentro del contexto de un contenedor de servlets como el Tomcat, es comúnmente utilizado para generar páginas web de forma dinámica a partir de parámetros de una petición de un navegador web.

JSP: es una tecnología Java para crear contenido dinámico para páginas web en forma de documentos HTML o XML.