

Facultad 5

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



Título: Sistema generador de encuestas.

Autor: Yadira Tasé Torres

Tutor: Ing. Leandro Raúl Ramírez Batista

Habana, 2012



" Si lo puedo pensar e imaginar, es que lo puedo hacer."

Albert Einstein

Declaración de autoría

En el presente proyecto de tesis declaro que cada una de sus observaciones, análisis, evaluaciones, conclusiones y recomendaciones emitidas, es de absoluta responsabilidad de la autora Yadira Tasé Torres y autorizo a la Universidad de las Ciencias Informáticas, a hacer uso del mismo en su beneficio.

Además, es necesario indicar que la información de otros autores empleada en el presente trabajo está debidamente especificada en fuentes de referencia y apartados bibliográficos.

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

Autor:

Tutor:

Yadira Tasé Torres _____

Leandro Raúl Ramírez Batista _____

Datos de contactos

Tutor: Leandro Raúl Ramírez Batista Ingeniero en Ciencias Informáticas graduado en el 2010 en la Universidad de las Ciencias Informáticas (UCI).

Nombre y Apellidos: Leandro Raúl Ramírez Batista

Institución: Universidad de las Ciencias Informáticas (UCI).

E-mail: lrramirez@uci.cu

Agradecimientos

Un sueño se hace realidad cuando se lucha día a día por él, cuando se pone todo el empeño y dedicación para alcanzarlo. Hoy se hizo realidad mi sueño profesional, hoy después de muchos años de sacrificio he alcanzado la meta que una vez me propuse y realmente se hace difícil escribir unos agradecimientos genéricos y plasmar en este papel el nombre de todas las personas de las cuales uno se siente agradecida, pues todas no tiene el mismo papel, ni actitud ante las vicisitudes de la vida para con uno y no quiero ser injusta dejando de incluir a alguna de estas personas que están ligadas indisolublemente a mi vida, por cuestiones de tiempo u otro factor. Y más que toda obra, si se quiere ser realmente sincero, es el resultado de esfuerzo de muchos, aunque el mérito se le atribuya a uno solo. Es el saber aunar voluntades y esfuerzo de aquellos que confluyen en la búsqueda de un objetivo común, o que al menos han participado, de forma activa o pasiva, consciente o no en la misma. Por lo que agradezco a todos los que me han ayudado a pasar estos años. Pero hay personas excepcionales que no puedo dejar de mencionar debido a la importancia de su participación desde el punto de vista espiritual, sentimental, en mi formación y en la realización de este trabajo:

A mi abuelita Migdalia por quererme tanto y estar orgullosa de mí, a ella que ha abierto sus brazos cuando necesito un abrazo y que sé que para ella este será uno de los momentos más felices de su vida.

A mis padres Dulce María y Guillermo por darme tanto de su amor, por estar siempre apoyándome, aconsejándome y alentándome en todo tiempo. Gracias por ser como son, pues la ilusión de su vida ha sido convertirme en persona de provecho y nunca podré pagar todos sus desvelos ni aún con las riquezas más grandes del mundo. Se me hace difícil hablar de tanto en tan pocas palabras pero les estaré agradecida toda mi vida. Los amo.

A mi hermano William, gracias por estar siempre presente en mi vida y por velar por mí desde pequeña. Gracias por tu ayuda y por ese amor que siempre me has dado, gracias por las cosas lindas que hemos compartido juntos.

A mi familia por su amor y su preocupación.

Gracias a Guillermo Pérez Sánchez por estar a mi lado en todo momento de esta carrera, por darme fuerzas cuando creía que todo estaba perdido, por regañarme cuando lloraba innecesariamente, por cambiar mis lágrimas por sonrisas, por darme ese infinito amor que alegró cada día de mi vida, por todos los momentos lindos que vivimos juntos, por enseñarme todas las cosas bellas que tiene el mundo, por ser una de las personas más especiales que he conocido y por querernos tanto.

Gracias Guille, I ♥ you.

A mis **padres de La Habana, María Eugenia Sánchez** por sus consejos y por su amor infinito y **Guillermo Pérez Román** por quererme como una hija más.

A **Yadira O'Reilly Llanez** por soportarme y saber guiarme todos estos años, gracias por permitirme ser tu hermanita querida, y aunque estemos lejos nunca te olvidaré.

A **Claudia Beatriz Larramendi Ferrás** por su preocupación, su apoyo, amor y dedicación, gracias por ser mi amiga.

A **Yadiel y Leanys** gracias por su amor, amistad y compañía durante estos 5 años, por soportarme en mis momentos difíciles y estar siempre presentes cuando los necesité. Los quiero muchísimo y doy Gracias a Dios por permitirme compartir todo este tiempo con ustedes.

A **Raudi** por ser ese chico incondicional que lo convirtió en mi ayudante perfecto en esta etapa.

A **Luis Ernesto** por ser mi Robinson y yo ser su viernes, gracias Teniente.

A **Leandro**, mi tutor, por su apoyo y ayuda cuando más lo necesité.

A **Orestes** por no cansarse de las molestias, por los materiales que me facilitó y por todo lo demás.

A mis **amigos del grupo**, a todos, sin obviar ninguno, por los will be, los buenos momentos, la unión en las reuniones, los escándalos, los correos chistosos, la complicidad, por tenernos presentes y por lograr que estos 5 años sean inolvidables.

A los **profesores** por la ayuda brindada durante el desarrollo de este trabajo y a todos los que han contribuido a mi formación profesional, en especial a **Francisca Ferrer** mi maestra de primaria, a **Reinaldo Porro** mi profesor de Matemáticas de la secundaria y a **Yusmila Videaux** por ser Profe y amiga incondicional en mi vida universitaria.

A mi **antiguo grupo 5101** aunque ya quedemos pocos gracias por todo su apoyo, paciencia y por compartir tantas cosas lindas y decir que es el mejor grupo en que he estado en mi vida universitaria.

En especial a las muchachitas con las que he vivido que me han aguantado mi carácter durante estos 5 cursos, que sé que eso es una dura tarea, a **Yanelis, Anaís, Danieska, Rosa María (La China), Laura, Claudia Morales, Elizabeth, Yamila, Liz Anayanci, Beatriz y Edith**. A pesar de todo las considero excelentes amigas, juntas hemos compartidos grandiosos momentos y no las olvidaré never, ever por eso se han convertido en la familia de la UCI.

A la **UCI** por ser mi hogar, por darme la posibilidad de crecerme como persona, por educarme, por darme tantos amigos.

Dedicatoria

A mis esfuerzos.

A mi abuela por creer en mí.

A mi hermano William por existir.

A toda mi familia por ser especial y excepcional.

A las personas que me dieron su ayuda sin pedir nada a cambio y creyeron en mí y en que podía ser capaz de realizar mis sueños.

A mi madre y a mi padre, por su cariño infinito, por su preocupación constante y por su confianza. Por ustedes y para ustedes. Todo mi esfuerzo en estos 5 años, todo mi empeño por no defraudarlos, este triunfo también es de ustedes.

Resumen

Los sistemas de gestión de la información le han servido al hombre para tomar decisiones importantes en sus vidas. Al pasar los años estos han perfeccionado el proceso y la obtención de datos, algo fundamental para las instituciones; ya que de esta forma pueden conocer el estado de sus trabajadores y pueden llegar a optimizar procesos laborales con las pesquisas realizadas.

Por tanto, el presente trabajo tiene como objetivo desarrollar una aplicación web dinámica que de soporte a la gestión de encuestas y se pueda integrar al portal del “Centro de Consultoría y Desarrollo de Arquitecturas Empresariales”, logrando mejorar el funcionamiento de los diferentes servicios y la obtención de información de una muestra de la población del centro, este proceso se realizará mediante el uso de cuestionarios estructurados que se utilizan para obtener datos difíciles de controlar manualmente y de esta forma poder generar reportes estadísticos con esta información.

En el documento se plasman los resultados del trabajo investigativo realizado. Se hace un análisis sobre las tecnologías existentes seleccionando las más apropiadas y se examinan los diferentes frameworks pues estos constituyen una guía muy completa para mejorar considerablemente la construcción de la solución. Se muestran los resultados del diseño de la propuesta del sistema y se le aplican pruebas para comprobar el correcto funcionamiento de la misma. Finalmente se dejan algunas recomendaciones para el mejoramiento futuro de la aplicación.

Palabras claves: encuestas; gestión; información; sistemas.

Índice

| | |
|--|------|
| DECLARACIÓN DE AUTORÍA | III |
| DATOS DE CONTACTOS | IV |
| AGRADECIMIENTOS..... | V |
| DEDICATORIA | VII |
| RESUMEN | VIII |
| ÍNDICE | IX |
| ÍNDICE DE TABLAS | XI |
| ÍNDICE DE FIGURAS | XIII |
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA. | 5 |
| 1.1 INTRODUCCIÓN | 5 |
| 1.2 ENCUESTAS..... | 5 |
| 1.2.1 Preguntas de la encuesta | 5 |
| 1.2.2 Las preguntas de la encuesta y los objetivos de la investigación deben coincidir. | 6 |
| 1.2.3 Comprender los participantes en la investigación | 6 |
| 1.2.4 Use un lenguaje natural y familiar..... | 6 |
| 1.2.5 Tipos de preguntas de la encuesta | 6 |
| 1.3 WEB | 7 |
| 1.3.1 Aplicaciones WEB | 8 |
| 1.3.2 Aplicaciones RIA | 9 |
| 1.4 PORTLETS..... | 9 |
| 1.5 FRAMEWORKS | 10 |
| 1.5.1 Framework Vaadin | 10 |
| 1.5.2 Framework Spring | 11 |
| 1.5.3 Framework Hibernate..... | 12 |
| 1.6 LIBRERÍAS PARA LA GENERACIÓN DE GRÁFICOS ESTADÍSTICOS | 13 |
| 1.6.1 JFreeChart | 13 |
| 1.6.2 JS Charts..... | 14 |
| 1.6.3 Highcharts..... | 15 |
| 1.7 SELECCIÓN DE LAS TECNOLOGÍAS, LENGUAJES Y HERRAMIENTAS A UTILIZAR..... | 16 |
| 1.7.1 Metodología..... | 16 |
| 1.7.2 Herramienta CASE | 16 |
| 1.7.3 Lenguaje de programación | 16 |
| 1.7.4 Entorno de desarrollo integrado..... | 16 |
| 1.7.5 Sistema gestor de base de datos | 17 |
| 1.7.6 Librerías para la generación de gráficos estadísticos. | 17 |

| | |
|--|-----------|
| 1.7.7 Herramienta de pruebas | 17 |
| 1.8 CONCLUSIONES PARCIALES | 18 |
| CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN | 19 |
| 2.1 INTRODUCCIÓN | 19 |
| 2.2 PATRÓN ARQUITECTÓNICO MODELO-VISTA-CONTROLADOR (MVC)..... | 19 |
| 2.3 PATRONES DE DISEÑO EMPLEADOS | 20 |
| 2.4 PROPUESTA DEL SISTEMA | 22 |
| 2.4.1 Requisitos funcionales..... | 22 |
| 2.4.2 Requisitos no funcionales..... | 22 |
| 2.5 METODOLOGÍA EXTREME PROGRAMMING (XP)..... | 24 |
| 2.6 FASE I: EXPLORACIÓN | 24 |
| 2.6.1 Flujo de procesos del sistema propuesto..... | 24 |
| 2.6.2 Tarjetas de historias de usuarios | 24 |
| 2.7 FASE II: PLANIFICACIÓN DE LA ENTREGA | 28 |
| 2.7.1 Estimación de esfuerzo por historias de usuarios..... | 28 |
| 2.8 FASE III: ITERACIONES | 29 |
| 2.8.1 Plan de duración de iteraciones..... | 29 |
| 2.8.2 Plan de entregas..... | 29 |
| 2.8.3 Plan de tareas..... | 30 |
| 2.9 FASE IV: PRODUCCIÓN. | 31 |
| 2.9.1 Diagrama de clases..... | 31 |
| 2.9.2 Descripción de las clases..... | 31 |
| 2.10 DISEÑO DE LA BASE DE DATOS | 33 |
| 2.10.1 Diagrama de las clases persistentes | 33 |
| 2.11 MODELO DE DESPLIEGUE. | 34 |
| 2.12 CONCLUSIONES PARCIALES..... | 35 |
| CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA. | 36 |
| 3.1 INTRODUCCIÓN | 36 |
| 3.2 TARJETA DE TAREAS DE PROGRAMACIÓN | 36 |
| 3.3 IMPLEMENTACIÓN. | 37 |
| 3.3.1 Iteración 1 | 37 |
| 3.3.2 Iteración 2 | 40 |
| 3.3.3 Iteración 3 | 41 |
| 3.4 PRUEBAS..... | 43 |
| 3.4.1 Pruebas de aceptación..... | 43 |
| 3.4.2 Pruebas unitarias..... | 46 |
| 3.4.3 Pruebas de rendimiento | 47 |
| 3.5 CONCLUSIONES PARCIALES | 48 |
| CONCLUSIONES GENERALES | 49 |
| RECOMENDACIONES | 50 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 51 |
| BIBLIOGRAFÍA..... | 53 |
| GLOSARIO DE TÉRMINOS | 54 |

Índice de tablas

| | |
|---|----|
| Tabla 1 Historias de Usuarios..... | 25 |
| Tabla 2 Historia de usuario: "Crear encuestas" | 26 |
| Tabla 3 Historia de usuario: "Modificar encuestas" | 26 |
| Tabla 4 Historia de usuario: "Eliminar encuestas" | 26 |
| Tabla 5 Historia de usuario: "Buscar encuestas" | 27 |
| Tabla 6 Historia de usuario: "Realizar encuestas" | 27 |
| Tabla 7 Historia de usuario: "Generar estadísticas" | 28 |
| Tabla 8 Estimación de esfuerzo por HU | 28 |
| Tabla 9 Plan de duración de las iteraciones | 29 |
| Tabla 10 Plan de entregas | 29 |
| Tabla 11 Plan de tareas | 30 |
| Tabla 12 Clase estadísticas..... | 32 |
| Tabla 13 Clase encuesta..... | 32 |
| Tabla 14 Clase pregunta | 32 |
| Tabla 15 Clase respuesta..... | 32 |
| Tabla 16 Tarjeta de tareas de programación | 36 |
| Tabla 17 HU implementadas en la primera iteración | 37 |
| Tabla 18 Tarea #1 de la HU: "Crear encuesta" | 37 |
| Tabla 19 Tarea #2 de la HU: "Crear encuesta" | 38 |
| Tabla 20 Tarea #1 de la HU: "Modificar encuesta" | 38 |
| Tabla 21 Tarea #2 de la HU: "Modificar encuesta" | 38 |
| Tabla 22 Tarea #1 de la HU: "Eliminar encuesta" | 39 |

| | |
|--|----|
| Tabla 23 Tarea #1 de la HU: "Buscar encuesta" | 39 |
| Tabla 24 Tarea #2 de la HU: "Buscar encuesta" | 40 |
| Tabla 25 Tarea #8 Aplicar pruebas..... | 40 |
| Tabla 26 HU implementadas en la segunda iteración | 40 |
| Tabla 27 Tarea #1 de la HU: "Realizar encuesta" | 40 |
| Tabla 28 Tarea #2 de la HU: "Realizar encuesta" | 41 |
| Tabla 29 Tarea #3 de la HU: "Realizar encuesta" | 41 |
| Tabla 30 HU implementadas en la tercera iteración | 42 |
| Tabla 31 Tarea #1 de la HU: "Generar estadísticas" | 42 |
| Tabla 32 Tarea #2 de la HU: "Generar estadísticas" | 42 |
| Tabla 33 Tarea #3 de la HU: "Generar estadísticas" | 43 |
| Tabla 34 Prueba de aceptación para HU: #1 | 44 |
| Tabla 35 Prueba de aceptación para HU: #2 | 44 |
| Tabla 36 Prueba de aceptación para HU: #3 | 45 |
| Tabla 37 Prueba de aceptación para HU: #4 | 45 |
| Tabla 38 Prueba de aceptación para HU: #5 | 45 |
| Tabla 39 Características del ambiente de prueba..... | 47 |
| Tabla 40 Características del ambiente de prueba..... | 47 |
| Tabla 41 Cantidad de pruebas unitarias realizadas | 46 |

Índice de figuras

| | |
|---|----|
| Figura 1 Representación del patrón MVC | 20 |
| Figura 2 Diagrama de clases..... | 31 |
| Figura 3 Modelo de datos | 34 |
| Figura 4 Modelo de despliegue | 34 |
| Figura 5 Cantidad de peticiones atendidas | 48 |

Introducción

La humanidad a lo largo de la historia se ha caracterizado por su afán de conocer su entorno, su curiosidad la ha llevado a cuestionar, indagar y profundizar sobre diversos temas que la rodean. Con el objetivo de poder adquirir conocimientos que le permita evolucionar y trascender, ha buscado varias vías de investigación que sean capaces de aportarle información. La búsqueda de nuevas vías y soluciones, capaces de guiar una revolución de cambios favorables a la humanidad, hacen que en el cada día, el hombre se plantee nuevas interrogantes que procuren resolver problemas existentes y sustituir los viejos estándares por modelos óptimos y novedosos. Entre los métodos de recolección de información a los que el hombre ha recurrido se encuentran el censo, los registros y las encuestas con el fin de reunir evidencias, siendo esta última una de las más utilizadas.

Las encuestas son manipuladas en distintas esferas sociales como la economía, la salud y la educación; utilizada considerablemente en ésta última esfera en el proceso de enseñanza-aprendizaje, con el objetivo de obtener información sobre las actitudes, conocimientos, habilidades, preferencias, intereses y hábitos de los estudiantes. (1)

En Cuba, diversas entidades educativas obtienen información estadística sobre el tema que se investiga con el uso de herramientas que proveen distintas funcionalidades para realizar encuestas

Actualmente son varias las organizaciones que utilizan las encuestas con el objetivo de obtener información más detallada en las diferentes investigaciones realizadas, donde la mayoría de las veces el tiempo requerido para los resultados es muy breve. Estas han proporcionado que los temas de investigación que no sean del agrado de las personas encuestadas puedan ser cambiados o mejorados. La necesidad de acceder a los datos de las encuestas en el menor tiempo posible, el lograr un procesamiento más rápido y efectivo, hizo que se implementase la automatización en los sistemas de encuestas.

En Cuba, la necesidad de un enfoque integral de consolidación de información sobre el desempeño organizacional y el apoyo para la toma de decisiones, motivan e impulsan el desarrollo de sistemas de información, en aras de lograr una mejor gestión institucional. (2)

Por esta razón múltiples proyectos de sistemas de gestión de información que se llevan a cabo en Cuba son desarrollados en la UCI (Universidad de las Ciencias Informáticas), centro educacional que forma profesionales en la rama de informática y se considera una de las mayores industrias de software del país.

La universidad y sus centros productivos presentan dificultades con la gestión de la información para la toma de decisiones. Uno de estos centros productivos es el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) que cuenta con un portal Web dinámico que favorece la divulgación de la información relacionada con el personal vinculado sin que exista un espacio donde

sus trabajadores y estudiantes emitan sus opiniones sobre temas favorables al proceso de toma de decisiones realizado por los directivos del centro.

Las dificultades antes expuestas provocan gastos innecesarios en recursos humanos y materiales para la recogida y procesamiento de la información utilizada para tomar decisiones referentes a:

- Selección de las herramientas para los proyectos.
- Captación de estudiantes para ingresar al CDAE.
- Asignación de roles en los proyectos y temas de tesis.
- Lograr consenso en los requisitos funcionales y no funcionales de los trabajos o cualquier otro aspecto de interés.

Las causas de estas dificultades radican en que el sistema de encuestas existente no contiene un mecanismo con posibilidades de archivar las estadísticas extraídas de las encuestas y por ende, los datos tienden a perderse con el tiempo.

Después de hacer un análisis de la **situación problemática** existente, se plantea como **problema de investigación**:

¿Cómo agilizar el proceso de gestión de información y estadísticas en el centro CDAE para la toma de decisiones?

Por lo que el **objeto de estudio** se circunscribe a: las aplicaciones web dinámicas que gestionan información. Mientras que el **campo de acción** se enmarca en: las aplicaciones web dinámicas que gestionan información para la toma de decisiones a partir de las encuestas.

Con el fin de resolver el problema planteado, se formula el siguiente **objetivo general**:

Desarrollar una aplicación web dinámica de generación de encuestas para el portal del centro CDAE que permita automatizar la gestión de información y sus estadísticas.

A partir de un análisis del objetivo general se derivan los siguientes **objetivos específicos**:

- Estudiar las tendencias actuales de las formas y mecanismos de encuestas para el desarrollo de la aplicación.
- Seleccionar metodologías, tecnologías y herramientas de desarrollo para el diseño e implementación del sistema.
- Definir los requerimientos funcionales y no funcionales para el desarrollo de la aplicación.
- Diseñar una base de datos para gestionar la información de las encuestas.
- Implementar una aplicación RIA que sustente un proceso de generación de encuestas.
- Validar los resultados obtenidos del sistema desarrollado.

Este trabajo está dirigido además, **a defender la idea** de que la propuesta desarrollada facilitará la obtención de información y estadística en el centro CDAE.

Para darle cumplimiento a los objetivos propuestos, se desarrollan las siguientes **tareas investigativas**:

- Estudio y análisis de los conceptos básicos y técnicos implicados en el desarrollo de una aplicación de gestión de la información.
- Evaluación de las tendencias, técnicas, tecnologías y metodologías relacionadas con las técnicas de programación actuales, así como de las plataformas de desarrollo que las soportan.
- Estudio y análisis de los frameworks a utilizar en el desarrollo de la aplicación.
- Estudio de librerías de renderizado de gráficos para el proceso de estadísticas.
- Diseño de una base de datos que soporte las funcionalidades del sistema.
- Diseño de las pruebas funcionales para la validación de la aplicación.
- Documentación y corrección de las no conformidades identificadas en la ejecución de las pruebas.

Para darle cumplimiento a las tareas se guiará la investigación en los marcos de los **métodos científicos** de investigación **teóricos** y **empíricos**.

Métodos teóricos:

Analítico-sintético: Analiza la documentación relacionada con los contenidos de los sistemas de gestión de encuestas existentes para lograr mayor comprensión y síntesis de los aspectos involucrados en la construcción de la aplicación.

Histórico-lógico: Brinda la posibilidad de analizar toda la evolución del problema que se estará estudiando.

Modelación: Representa toda la información obtenida hasta el momento, lo que permite reflejar las relaciones y cualidades de la aplicación que se quiere desarrollar.

Métodos empíricos:

Observación: Permite adquirir información necesaria y puede utilizarse en cualquiera de las fases de la investigación, además ofrece acercamiento a la realidad y permite ver la posible solución del problema desde diferentes ángulos.

El presente documento consta de 3 capítulos que a su vez se dividen en epígrafes y estos en sub-epígrafes de acuerdo al nivel de detalle que requiere el contenido abordado en cada uno de ellos. A continuación se explica brevemente el contenido de cada capítulo:

Capítulo I. Fundamentación teórica. Este capítulo recoge el análisis de la información existente acerca del tema a tratar y las tendencias actuales que existen en el mundo. También incluye una descripción que da un sustento teórico a la selección de las herramientas y la metodología a utilizar para resolver el problema planteado.

Capítulo II. Análisis y diseño de la solución. Descripción de los requisitos y análisis de la solución propuesta. Se describen las historias de usuarios que modelan la solución y sus principales funcionalidades.

Capítulo III. Implementación y validación de la solución propuesta. Se ofrecen los detalles relacionados con la implementación de las funcionalidades definidas en el capítulo anterior y la validación de la solución propuesta, refiriéndose al diseño de las pruebas.

Capítulo 1 Fundamentación teórica.

1.1 Introducción

En el presente capítulo se describirán las tendencias existentes para el desarrollo de aplicaciones RIA (Aplicaciones Enriquecidas de Internet). Además se realizará un estudio sobre las características, técnicas y tecnologías a utilizar para el desarrollo del trabajo propuesto. Se justificarán las herramientas, tecnologías y metodologías, así como los frameworks, el lenguaje, la plataforma y el servidor de aplicación que se utilizará en el desarrollo de éste trabajo y darle así solución al problema de esta investigación.

1.2 Encuestas

La encuesta es uno de los métodos más utilizados en la investigación pues permite obtener amplia información de fuentes primarias. Definición de Encuesta según varios autores:

- Una encuesta consiste en reunir datos entrevistando a la gente. (3)
- Las encuestas obtienen información sistemáticamente de los encuestados a través de preguntas, ya sea personales, telefónicas o por correo. (4)
- Las encuestas son entrevistas a numerosas personas utilizando un cuestionario diseñado en forma previa. (5)

En síntesis, la definición de encuesta enfoca a la misma como un método que consiste en obtener información de las personas encuestadas mediante el uso de cuestionarios diseñados. Complementando esta definición, el método de encuesta es un cuestionario estructurado que se da a una muestra de la población y está diseñado para obtener información específica de los entrevistados. Por otra parte, en el Diccionario de Marketing de Cultural S.A. se define el término encuestación como el método de recogida de información cuantitativa, que consiste en interrogar a los miembros de una muestra sobre la base de un cuestionario perfectamente estructurado. (6)

1.2.1 Preguntas de la encuesta

¿Qué tipo de preguntas se deben incluir en las encuestas?

El tipo de pregunta y su elaboración correcta es guiado por la aplicación de varios principios básicos con el fin de medir de manera eficaz los atributos deseados.

Hay varios principios básicos a tener en cuenta en el desarrollo de sus preguntas. A continuación se presentan algunos elementos claves a tener en cuenta al escribir las preguntas de la encuesta.

1.2.2 Las preguntas de la encuesta y los objetivos de la investigación deben coincidir.

Siempre se debe declarar la intensión de la investigación antes de escribir las preguntas de la encuesta. Por ejemplo, al efectuar una encuesta de investigación de carácter exploratorio, por lo general, esta no necesita ser detallada como una encuesta de investigación confirmatoria.

1.2.3 Comprender los participantes en la investigación

Es necesario tener presente que la población es el principal objetivo, por tanto hay que desarrollar una capacidad de pensar como posibles encuestados, considerándose como participantes en la investigación con el objetivo de lograr una interpretación exacta de cada pregunta en la encuesta. Es probable que los datos resultantes proporcionen información útil para la investigación de diversos temas.

1.2.4 Use un lenguaje natural y familiar.

Usar un lenguaje que sea comprensible para el tipo de los encuestados. Tener en cuenta su edad, nivel educativo, y cualquier otra característica relevante cultural de los mismos al momento de decidir cómo formular las preguntas.

1.2.5 Tipos de preguntas de la encuesta

Las preguntas de la encuesta varían de acuerdo al tipo de información que se desea conocer de los encuestados y cómo esta información se aplicará a los objetivos de la encuesta.

Hay dos tipos básicos de preguntas de las encuestas: Abiertas y Cerradas.

Abiertas

Este tipo de preguntas permite a los participantes responder de cualquier manera que elijan. Las preguntas abiertas proporcionan datos principalmente cualitativos y se utilizan con frecuencia en la investigación exploratoria.

¿Pregunta solo comentario? (Emita su opinión)

Cerradas

En contraste con las preguntas abiertas, las preguntas cerradas requieren que los participantes elijan entre un número finito de respuestas predeterminadas por el investigador. Hay 3 tipos básicos de preguntas cerradas: opción múltiple, opción única y diatómica. Las preguntas cerradas proporcionan datos sobre todo cuantitativos y se utilizan con frecuencia en la investigación confirmatoria.

Opción múltiple

Utilizar una pregunta de selección múltiple cuando se pretende que los encuestados elijan de un enunciado y un grupo de alternativas las que deseen (sin límite de opciones).

¿Pregunta de opción múltiple? (Marque al menos una.)

- ☐ Respuesta 1
- ☐ Respuesta 2
- ☐ Respuesta 3
- ☐ Respuesta 4
- ☐ Respuesta 5

Opción única

La respuesta debe ser única y tomada de un listado de alternativas donde se le aclara al encuestado que solo debe escoger una (para determinar prevaecía, importancia, entre otros).

¿Pregunta de opción única? (Seleccione una.)

- ☐ Respuesta 1
- ☐ Respuesta 2
- ☐ Respuesta 3
- ☐ Respuesta 4
- ☐ Respuesta 5

Diatónica

Preguntas que establecen sólo 2 alternativas de respuesta, "Si o No", solo se puede seleccionar una respuesta

¿Pregunta?

- ☐ Sí
- ☐ No

1.3 Web

Con respecto a la web las tecnologías presentan un avance acelerado con relación a otros tipos de tecnología en la rama de la informática, son renovadas o mejoradas en breves espacios de tiempo. La web en sus inicios era un entorno estático que hacía uso de páginas HTML e imágenes GIF para

lograr cierta animación en sus contenidos y presentaba muy poca actualización de sus datos e interacción con los usuarios, esta es la llamada Web 1.0.

La Web 2.0 es la representación de la evolución tomando como punto de partida la Web 1.0, donde las aplicaciones web son enfocadas al usuario final, dándole a este el control de las mismas; haciendo uso de AJAX, DHMTL, XML, SOAP, entre otras que han ido surgiendo con el transcurrir de los años. (7)

1.3.1 Aplicaciones WEB

Una aplicación Web no es un sistema en tiempo real, es decir, es una aplicación de software que se basa en el modelo cliente-servidor y mayormente corren en los servidores. Solo se corre en el navegador lo que se programa en el cliente que está diseñado para automatizar procesos cuyo resultado se entrega a través de internet o una intranet. (8)

Las aplicaciones Web son notorias debido a la facilidad para actualizarlas y mantenerlas sin tener que distribuir e instalar software a miles de usuarios. En la actualidad y debido al auge de este tipo de aplicaciones, se reconocen diferentes tipos de categorías en dependencia del tipo de negocio que modelen:

- Informacionales: periódicos, catálogos, manuales, libros electrónicos.
- Interactivas: Formularios de registros, presentación de información personalizada.
- Transaccionales: tienda electrónica, bancos online.
- Workflow: planificación online, monitoreo, gerencia de inventario.
- Comunitarias: Chat, mercados, subastas online.
- Portales: tiendas electrónicas.

Ventajas de las aplicaciones WEB

Las aplicaciones Web han tenido un auge en los últimos años, colocándose en primer lugar ante las aplicaciones de escritorio que en ocasiones requieren de instalaciones previas y dispositivos auxiliares para la perfecta visualización de las mismas.

Las páginas Web brindan la posibilidad del acceso de navegación y de interacción pues no necesitan ninguna configuración especial o cambios necesarios en la computadora del usuario. Los datos se encuentran centralizados garantizando la integridad y la copia de seguridad y las actualizaciones pueden hacerse de forma rápida y fácilmente. (9)

1.3.2 Aplicaciones RIA

Durante el uso de las aplicaciones Web se necesita recargar la página para realizar tareas que son simples o que requieren un cambio mínimo. Para darle solución a este inconveniente se decidió estudiar otras vías de crear aplicaciones sobre la plataforma Web.

Una aplicación RIA es ejecutada en el interior de un navegador y se caracteriza y diferencia de las aplicaciones web corrientes por presentar un comportamiento similar al de las de escritorio. El aspecto fundamental es que no se precisa una actualización de la página completa, que actúa como interfaz de usuario cada vez que interactúa con la funcionalidad que ofrece el programa.

RIA introduce un nuevo modelo de programación de aplicaciones que combina las ventajas de los dos modelos predominantes hasta el momento: el de las aplicaciones cliente-servidor y el del modelo multicapa utilizado por las aplicaciones web, con un claro objetivo: mejorar la experiencia del usuario. Consisten en el aprovechamiento de la experiencia del usuario en herramientas y funciones de escritorio tan naturales como copiar, cortar y pegar, redimensionar columnas y ordenar, con el alcance y la flexibilidad de presentación y despliegue que ofrecen las aplicaciones o páginas Web junto con lo mejor de la multimedia (voz, vídeo, etc.).

Ventajas:

1. Agilidad en la respuesta.
2. Cálculos rápidos, controles prediseñados y funciones gráficas, interactivas y multimedia avanzadas.
3. En muchos casos no requieren de instalación en el equipo del usuario (es suficiente con disponer de un navegador web), por lo que no es necesario pensar en distribuciones de software.
4. Uso desde cualquier ordenador con acceso a internet.

1.4 Portlets

Son componentes web gestionados por un contenedor que tras la petición de un usuario generan y presentan contenidos dinámicos de forma identificable en la interfaz de usuario del portal como componentes de contenido. El portlet permite la personalización, la presentación y la gestión de la seguridad. Típicamente, siguiendo la metáfora de escritorio, una página de un portal se visualiza como una colección de ventanas de portlet que no se solapan, donde cada una de estas muestra un portlet. Por lo tanto un portlet (o colección de portlets) se asemeja a una aplicación web que está hospedada en un portal.

Los portlets son similares a los servlets en que:

- Los portlets son manejados por un contenedor especializado.
- Los portlets generan contenido dinámicamente.
- El ciclo de vida de los portlets es controlado por el contenedor.
- Los portlets interactúan con el cliente web mediante el uso del paradigma request/response.

Los portlets son diferentes a los servlets en que:

- Los portlets son únicamente generados como fragmento de etiquetado y no como documentos completos.
- Los portlets no están asociados directamente a una URL.
- Los portlets no pueden generar contenido arbitrario, ya que el contenido de los portlets va a estar incluido en la página del portal. Si un servidor de un portal está solicitando text/html, entonces todos los portlets deben ser generados en text/html. Por otro lado si el servidor del portal está solicitando por WML, entonces cada portlet deberá ser generado en contenido WML. (10)

Los portlets son elementos dependientes de las tecnologías empleadas para el desarrollo de los contenedores, por tanto para su integración debe usarse el mismo lenguaje de programación y framework de desarrollo.

1.5 Frameworks

Un framework no es más que una estructura de soporte definida que permite la organización y desarrollo de un software. Puede incluir soportes a programas, APIs y lenguajes scripting entre otros software para ayudar a desarrollar y a unir los diferentes componentes de un proyecto.

1.5.1 Framework Vaadin

Vaadin es un framework web creado para desarrollar aplicaciones RIA sin plugins y con una arquitectura server-driven, utilizando el modelo orientado a componentes que para su utilización no es necesario conocer HTML, XML y tampoco Javascript

Este framework trabaja sin tener en cuenta estar en el lado del cliente o del servidor y se puede extender usando Google Web Toolkit (GWT). Los componentes son similares a los widgets de GWT pero su estado se almacena en el servidor. La sincronización entre el servidor y los componentes es manejada automáticamente por el framework. (11)

Ventajas:

1. La realización de las UI (Interfaz de Usuario) enteramente en código Java es importante pues evita al desarrollador tener que aprender otros frameworks de desarrollo para la creación de interfaz de usuario.
2. En la comunidad Vaadin el equipo de trabajo ha desarrollado componentes visuales que se pueden ver en acción y consultar el código de la implementación de los mismos, en la página Vaadin demo samplers.
3. Vaadin tiene un sistema de add-ons, que son componentes desarrollados por el equipo de desarrollo de Vaadin o por colaboradores que se añaden a los componentes core básicos. Lo bueno de este sistema no es que se tenga acceso a esos add-ons sino que se puede extender el core de Vaadin para realizar componentes que se adapten a las necesidades de cada proyecto.
4. Vaadin está diseñado para poder utilizar los componentes visuales realizados con GWT, de manera que es capaz de gestionar en la parte servidora los eventos generados por los componentes visuales de GWT en la parte cliente.
5. Se puede crear Portlets a partir de Vaadin. Permite llevar la aplicación desarrollada directamente como portlet a Liferay.
6. Integración con Eclipse, STS y Netbeans. Vaadin se integra perfectamente con los IDEs de desarrollo más populares que existen actualmente.

1.5.2 Framework Spring

El framework Spring facilita al desarrollador la implementación del software, promoviendo buenas prácticas de programación, maneja objetos del negocio y además se integra fácilmente con otros frameworks. Tiene el objetivo de facilitar la construcción de aplicaciones en Java, se puede utilizar en cualquier tipo de aplicaciones, no solamente es utilizado en aplicaciones web y es ligero debido al poco impacto que tiene en las aplicaciones, lo que trae varios beneficios arquitectónicos como son:

- Puede organizar eficazmente la capa intermedia de cualquier aplicación, tanto de escritorio como web.
- Las aplicaciones con Spring son fáciles de testear.
- Facilita el desarrollo de buenas prácticas, reduce el costo de programación en una aplicación.

Spring está basado en el principio de inyección de dependencias, esta técnica hace externo el manejo y la creación de las dependencias de los componentes, logrando así obtener una mayor claridad y limpieza en el código.

Características:

- Soporta: JTA, Hibernate, JDO, JDBC, ODBC.
- Usa implementaciones ORM de terceros como Hibernate, Ibatis, JDO, OJB.
- Provee servicios de aplicación en forma declarativa.
- Permite usar archivos XML de configuración, programación mediante la API y mediante un estándar JSR.
- Aporta integración con la solución de código abierto Acegi (soporta seguridad declarativa basada en el uso de IoC y AOP).
- Permite el uso de cualquier servicio usando un archivo XML de configuración.
- Todos los componentes pueden ser testeados fuera del contenedor.

Ventajas:

1. Reduce potencialmente el código de enlace entre los diferentes componentes de la aplicación.
2. Permite la reconfiguración de las dependencias sin necesidad de compilar todo el código.
3. Disminuye el margen de errores.
4. Fomenta un buen diseño de la aplicación.
5. Spring recomienda el uso de Interfaces, si se decide no utilizar Spring, basta con volver a implementar las interfaces. (12)

1.5.3 Framework Hibernate.

Hibernate es un motor de persistencia de código abierto. Es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Trabajar con software orientado a objetos y bases de datos relacionales, puede hacer invertir mucho tiempo en los entornos actuales. Realiza el mapping entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa. Hibernate no solo realiza esta transformación sino que proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que reducen el tiempo de desarrollo.

Ventajas:

1. Soporta el paradigma de orientación a objetos de una manera natural, como en caso de utilizar herencia, composición, polimorfismo y las librerías de colecciones de Java.
2. Posee lenguaje de consultas de datos llamado HQL (Hibernate Query Language), lo que proporciona una independencia del SQL de cada base de datos ya sea para la recuperación como para el almacenamiento de objetos.
3. Permite una gran variedad de mapeos para colecciones y objetos dependientes.
4. Posee una cache de dos niveles, un alto rendimiento y puede ser utilizado en un clúster.
5. Compuesto de muchos proyectos como: core (proyecto principal), Annotations, entitymanager, shards, validator.
6. Tiene varios plugins para eclipse y tareas de Ant para ayudar a la utilización y automatización de la persistencia. Están dentro del proyecto Hibernate-tools.
7. Se pueden crear filtros definidos por el usuario.
8. Gracias al proyecto Hibernate-Annotations se pueden utilizar anotaciones de JDK 5.0 junto con los ficheros XML para el mapeo de objetos. (13)

Hibernate presenta un mecanismo de persistencia transparente lo que trae consigo que los objetos desconocen su capacidad de persistencia, no necesita un contenedor de aplicaciones, son fáciles de manejar y pueden ser usados para transportar los datos a cualquier capa de la aplicación. Estas son algunas de las razones por las que ha ganado en preferencia y popularidad entre los grandes desarrolladores de aplicaciones, además de su integración con el framework Spring lo cual trae numerosas ventajas de seguridad, del manejo de las transacciones y de la inyección de dependencias, entre otras.

1.6 Librerías para la generación de gráficos estadísticos

Cuando se desarrolla una aplicación informática los reportes estadísticos ya sea solo de datos o de datos y gráficos son una de las partes más importantes a tener en cuenta para facilitar el análisis y toma de decisiones. Los gráficos estadísticos presentan los datos en forma de dibujo de tal modo que se pueda percibir fácilmente los hechos esenciales y compararlos con otros. A continuación se describen algunas librerías en java que posibilitan la generación de gráficos estadísticos.

1.6.1 JFreeChart

Es un marco de software open source para el lenguaje de programación Java, el cual permite la creación de gráficos complejos de forma simple.

JFreeChart también trabaja con GNU Classpath, una implementación en software libre de la norma estándar de biblioteca de clases para el lenguaje de programación Java.

JFreeChart es compatible con una serie de gráficas diferentes, incluyendo cuadros combinados.

Tipos de gráficos compatibles:

- Gráficos XY (línea, spline y dispersión). Es posible usar un eje del tiempo.
- Gráfico circular.
- Diagrama de Gantt.
- Gráficos de barras (horizontales y verticales, apiladas e independientes). También tiene incorporado un dibujador de histogramas.
- Valor único (termómetro, brújula, indicador de velocidad) que luego se pueden colocar sobre el mapa.
- Varias gráficas específicas (tabla de viento, gráfica polar, burbujas de diferentes tamaños, etc.).

Además de los gráficos, es posible colocar varios marcadores en el área de gráfica.

JFreeChart dibuja automáticamente las escalas de los ejes y leyendas. Con el ratón informático se puede hacer zoom en la interfaz de la gráfica automáticamente y cambiar algunos ajustes a través del menú local. (14)

1.6.2 JS Charts

JS Charts es un generador basado en JavaScript que requiere poca o ninguna codificación. Con JS Charts dibujar gráficos es una tarea simple y fácil, ya que sólo se tienen que utilizar secuencias de comandos del lado del cliente (es decir, realizada por el navegador web). No hay plugins adicionales o módulos de servidor. Sólo se incluyen los guiones, preparar los datos del gráfico en la matriz de XML, JSON o JavaScript y su gráfica está lista.

JS Charts permite crear gráficos en diferentes plantillas, como gráficos de barras, gráficos pasteles o gráficos de líneas simples.

Características. ¿Por qué utilizar gráficos JS?

- No se tiene que escribir el código.
- Es fácil de integrar.
- Es personalizable.
- Puede configurar los gráficos en línea.
- Puede crear los tres tipos más comunes de gráficos: barras, pasteles y la línea.

- Es compatible con la mayoría de los navegadores.
- No es necesario ningún plugins del lado del servidor y módulos.
- Se trata de un 100% de componentes JavaScript.
- Es gratuito.

¿Qué hay de nuevo?

- Los pasteles en 3D y nuevos bares en 3D.
- Nuevas leyendas.
- Varias series en la barra de gráficos.
- Los gráficos de barras horizontales.
- Animación de línea / barra de nueva planta de recogida.
- Los nuevos métodos de personalización. (15)

1.6.3 Highcharts

Highcharts es una biblioteca de gráficos escrito en JavaScript puro, que ofrece una manera fácil de agregar gráficos interactivos a su sitio web o aplicación web. Highcharts actualmente soporta la línea, ranura, área, columna, barra, pastel y los tipos de gráfico de dispersión. Se basa únicamente en las tecnologías de navegador nativo y no requiere plugins del lado del cliente como Flash o Java. Además no es necesario instalar nada en su servidor. No PHP o ASP.NET. Highcharts sólo necesita dos archivos JS para funcionar: el núcleo highcharts.js y, o bien el jQuery, MooTools o marco prototipo.

Características:

- **Numerosos tipos de gráficos:** Highcharts apoya la línea, ranura, área, columna, barra, pastel y los tipos de gráfico de dispersión. Cualquiera de estos se pueden combinar en un gráfico.
- **Sintaxis de configuración simple:** Las opciones se presentan en una estructura de JavaScript Object Notation, que es básicamente un conjunto de claves y valores conectados por dos puntos, separados por comas y se agrupan por llaves.
- **Exportar e imprimir:** Con el módulo de exportación habilitado, los usuarios pueden exportar el gráfico a PNG, JPG, PDF o SVG en el tecleo de un botón, o imprimir el gráfico directamente desde la página web.
- **Contenedores:** Highcharts expone las características de sus principales funcionalidades a través de contenedores (wrappers) a los programadores mediante una API del lado del servidor, en lenguajes como PHP, ASP.NET, Java, entre otros, las más relevantes son HighRoller, Invient Chart y Highcharts.Net. (16)

1.7 Selección de las tecnologías, lenguajes y herramientas a utilizar

Una de las principales pautas a seguir para la construcción o desarrollo de un software es la selección de las tecnologías, herramientas y lenguajes a emplear de acuerdo con las tendencias actuales y considerando las que aporten mayor beneficio a su uso. Por lo que se desarrolló un estudio de las mismas, mencionando sus principales características y ventajas. A continuación se mencionan las seleccionadas:

1.7.1 Metodología

Se escogió XP debido a que es una metodología ágil, cuyo objetivo fundamental es la satisfacción del cliente. Disminuye el riesgo de cometer errores; pues existe un contacto directo entre el desarrollador y el cliente y este último puede formar parte del desarrollo y a la vez puede ir realizando pruebas al software para asegurar su conformidad con el mismo. Además como el ciclo de desarrollo de la aplicación no es extenso, con las fases que plantea se simplifica y organiza el trabajo logrando disminuir el tiempo en el proceso de desarrollo del sistema.

1.7.2 Herramienta CASE

Para el modelado del sistema se empleó la herramienta CASE Visual Paradigm por su amplio uso vinculado a la plataforma y al lenguaje de programación. Este soporta el ciclo de vida completo del desarrollo del software, es fácil de instalar y actualizar, exporta e importa los diagramas con estándar XML y como imágenes, permite generar la documentación automáticamente a partir de los diagramas en diferentes formatos PDF, HTML y Microsoft Word, ahorra tiempo al desarrollador y minimiza las posibilidades de cometer errores.

1.7.3 Lenguaje de programación

Para el desarrollo del sistema de gestión de encuesta se escogió Java debido a que el portal del centro CDAE está desarrollado sobre este lenguaje y se puede lograr integrar con facilidad el sistema generador de encuesta al mismo. Además se logra la independencia de plataforma lo que posibilita que no sea necesario compilar el código fuente cada vez que se cambie de sistema operativo por lo que es capaz de producir código totalmente transportable. El modelo de objeto en Java es simple y elimina herramientas de bajo nivel como punteros.

1.7.4 Entorno de desarrollo integrado

El IDE escogido para la implementación de este proyecto fue Eclipse porque es un entorno de desarrollo muy completo, el mismo incorpora un sin fin de utilidades para simplificar la labor del programador; para él se han desarrollado varias extensiones (plugins).

1.7.5 Sistema gestor de base de datos

Como sistema gestor de base de datos se escogió PostgreSQL por ser rápido, confiable y fácil de usar, ofrece hoy una rica variedad de funciones, su conectividad, velocidad y seguridad lo hacen altamente satisfactorio para acceder a bases de datos. PostgreSQL está considerado como el gestor de base de datos de código abierto más avanzada del mundo.

1.7.6 Librerías para la generación de gráficos estadísticos.

Para la selección de la librería de generación de gráficos estadísticos se eligieron 6 aspectos comparativos:

- Facilidad de uso, ergonomía.
- Adaptabilidad.
- Documentación.
- Compatible con framework Vaadin.
- Actividad de las versiones.
- Popularidad.

Luego se tomó cada librería y se le dio una evaluación siguiendo las reglas de notación siguiente:

- 0 si la funcionalidad no está cubierto.
- 1 si la función está parcialmente cubierto.
- 2 si la función está completamente cubierto.

Obteniéndose como resultados:

- JS Chart (7 puntos).
- JFreeChart (9 puntos).
- Highcharts (12 puntos).

Destacando como aspecto más relevante la compatibilidad con el framework Vaadin. Por tanto la librería de generación de gráficos estadísticos escogida fue Highcharts.

1.7.7 Herramienta de pruebas

JMeter es una herramienta java de uso gratuito, que permite realizar pruebas de rendimiento y funcionales sobre aplicaciones web. Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Permite además activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas

partes iniciales que sean muy pesadas o largas. Tiene la forma de generar un caso de prueba a través de una navegación de usuario.

1.8 Conclusiones parciales.

En este capítulo se describieron las principales características de las aplicaciones RIA así como sus ventajas y el uso e importancia de los portlets en este tipo de software.

Hasta este momento solo se plantean ideas bien fundamentadas que constituyen la base para el posterior desarrollo de este trabajo.

Se realizó además un análisis de las tecnologías y herramientas utilizadas a lo largo del desarrollo del sistema propuesto, profundizándose en el conocimiento de conceptos necesarios para la comprensión del trabajo. Se mencionaron las características que se tuvieron en cuenta al escoger el lenguaje de programación, los frameworks y la metodología a utilizar. Estas herramientas apoyan el desarrollo general del sistema propuesto, brindando facilidades al desarrollador en cuanto a rapidez, facilidad de uso y variedad de funciones.

Capítulo 2 Análisis y diseño de la solución.

2.1 Introducción

En este capítulo se obtiene la modelación del sistema mediante la descripción de los procesos del negocio, teniendo en cuenta que antes de realizar la automatización de un proceso del negocio es necesario entender su funcionamiento y estructura correctamente, pues un entendimiento apropiado de los mismos permite la comprensión y modelación de estos, así como la vinculación del desarrollador con los términos utilizados por el cliente.

A partir de comprender el negocio, se obtienen los requisitos funcionales y no funcionales. La traducción de los mismos a una especificación que describe cómo implementar la aplicación.

2.2 Patrón arquitectónico modelo-vista-controlador (MVC)

Teniendo en cuenta que Spring está basado en un patrón clásico conocido como arquitectura MVC, que está formado por 3 niveles es el más óptimo para la aplicación porque:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página Web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. (17; **Error! No se encuentra el origen de la referencia.**)

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de por ejemplo: el tipo de gestor de bases de datos utilizado por la aplicación. (18)



Figura 1 Representación del patrón MVC

Se selecciona el patrón de arquitectura Modelo Vista Controlador por:

- Brindar la posibilidad de desarrollar la aplicación de forma organizada.
- Facilitar agregar nuevos tipos de datos según sea requerido por la aplicación, pues son independientes del funcionamiento de las otras capas.
- Facilitar el mantenimiento de la aplicación.

2.3 Patrones de diseño empleados.

Constituyen la base para la búsqueda de soluciones a problemas comunes en el desarrollo de un software y otros ámbitos referentes al diseño de interfaces. (19)

En el desarrollo del software se emplearon los patrones GRASP (Responsibility Assignment Software Patterns) los que describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

GRASP se destaca por presentar 5 patrones principales:

- Experto.
- Creador.
- Alta cohesión.
- Bajo acoplamiento.
- Controlador.

Experto: Asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. En la aplicación se evidencia este patrón con el uso de clases que poseen responsabilidades específicas a cumplir, por ejemplo las clases de acceso a datos poseen la responsabilidad de modificar los datos persistentes del sistema.

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. Indica que la clase incluyente del contenedor o registro es idónea para asumir la responsabilidad de crear algo contenido o registrado. Una demostración clara del uso de este patrón en la aplicación es correspondiente a la historia de usuario, "Crear Encuesta".

Alta Cohesión: Asignar una responsabilidad de modo que la cohesión siga siendo alta. Es un principio que se debe tener presente en todas las decisiones de diseño: es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. El uso de este patrón se manifiesta fundamentalmente en las clases de acceso a datos, teniendo estas el mínimo de atributos y funciones a utilizar, para facilitar la reutilización y el mantenimiento de dichas clases al realizar cambios que puedan afectar la integridad del código.

Bajo Acoplamiento: Estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienten la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta Cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades. El acoplamiento tal vez no sea tan importante, si no se busca la reutilización. En la aplicación se muestra un ejemplo del buen uso de este patrón, al mostrarse las mínimas relaciones existentes entre las clases seleccionadas.

GoF

Los Patrones GoF son soluciones basadas en la experiencia que corrigen problemas comunes que se presentan en el diseño. Estos describen 23 patrones comúnmente utilizados y de gran aplicabilidad en problemas de diseño usando modelamiento UML. Se agrupan en tres grandes categorías: creacionales, estructurales y de comportamiento. Dentro de esta primera categoría se va a enmarcar el desarrollo de la aplicación utilizando el patrón Singleton.

Singleton (Instancia única): El patrón Singleton en la aplicación asegura que sólo exista una instancia de una clase dentro del sistema proporcionando un punto de acceso común, también conocido como “Clase de instancia única” o “Instancia única”. La utilización de este patrón posibilitará tener solamente un punto de acceso a la clase que se encarga de establecer la conexión a la base de datos. (20)

2.4 Propuesta del sistema

La solución que se plantea es proporcionar una mejor confección de las encuestas, debido a las nuevas modalidades de preguntas añadidas que permiten recopilar mayor información, y la generación de estadísticas que posibilitan la realización de un estudio más profundo de los datos obtenidos. Esta solución facilita una mejor interpretación y visualización de los resultados alcanzados en las encuestas a partir de tipos de gráficas.

2.4.1 Requisitos funcionales.

R1 Gestionar encuestas.

R2 Mostrar encuestas a responder.

R3 Realizar encuesta.

R4 Buscar encuesta.

R5 Mostrar estadísticas de las encuestas.

2.4.2 Requisitos no funcionales.

- **Apariencia o interfaz de usuario:** La aplicación está diseñada con una interfaz amigable donde predominan los colores gris y rojo pues está alineada con los colores del portal CDAE, fácil de usar por el usuario de manera que agilice y facilite el trabajo con el software pues el sistema brindará servicios tanto a usuarios familiarizados con ambientes informáticos como a otros no familiarizados.

- **Usabilidad:** La aplicación tiene que ser capaz de ofrecer facilidades de uso para un buen entendimiento y aceptación del producto por los usuarios finales. Debe ser sencillo a la vista teniendo en cuenta que lo pueden utilizar estudiantes de nuevo ingreso que no tienen habilidades con herramientas automatizadas.
- **Rendimiento:** El sistema para lograr un buen rendimiento, debe tener un rápido procesamiento de los datos pues los usuarios estarán en constante intercambio de información. Al estar concebida para un ambiente cliente - servidor, debe garantizar la rapidez de respuesta del sistema ante las solicitudes de los usuarios, al igual que la velocidad de procesamiento de la información. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos.
- **Soporte:** Se necesita un servidor que contenga soporte para grandes volúmenes de datos y velocidad media de procesamiento, con menor tiempo de respuesta posible, para la aplicación y la base de datos. Este debe ser compatible con los demás sistemas automatizados que intervengan en el funcionamiento del portal CDAE.
- **Portabilidad:** El sistema será multiplataforma debido a que será ejecutado sobre una máquina virtual que interpreta el código implementado en el lenguaje de programación Java convirtiéndolo a código particular del sistema operativo sin importar sus versiones, y sin necesidad de modificar su código fuente. El sistema se podrá acceder a través de cualquier navegador.
- **Legales:** El sistema estará desarrollado en base a las políticas del software libre, que fueron ajustadas al sistema como: uso de servidores, gestor de base de datos PostgreSQL, lenguaje del lado del servidor Java, que son herramientas libres además del uso de Visual Paradigm que no es libre pero la Universidad de las Ciencias Informáticas posee licencia para su uso.
- **Confiabilidad:** El sistema debe ser confiable y preciso en la información que le suministra al usuario para evitar cualquier tipo de error. La información puede ser manejada solamente por usuarios autenticados en el portal que le da los permisos de acceso al sistema. Este debe ser estable, fiable y la velocidad de respuesta debe ser rápida durante la utilización del mismo. La información almacenada debe ser confiable en cuanto a su veracidad e integridad desde su recopilación y durante toda su manipulación.
- **Software:**

Sistema Gestor de Base de Datos PostgreSQL, versión 8.4.

En los clientes Mozilla Firefox 3.4o superior.

➤ **Hardware**

Requerimientos mínimos para el servidor:

Computador Pentium a 333 MHz o superior.

512 MB RAM o superior.

4 GB de espacio libre en Disco Duro como mínimo.

Requerimientos mínimos para la conexión del cliente:

Computador Pentium a 233 MHz o superior.

128 MB RAM o superior.

MODEM o red con TCP-IP para conexión al servidor.

2.5 Metodología Extreme Programming (XP)

A continuación se presentan las características esenciales de XP organizando el ciclo de vida en fases. La misma permite mantener un mínimo nivel de documentación, lo cual a su vez se traduce en una gran velocidad en el desarrollo del sistema.

2.6 Fase I: Exploración

En la primera fase de la metodología XP, los clientes plantean a grandes rasgos las historias de usuarios que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán durante el desarrollo del sistema.

2.6.1 Flujo de procesos del sistema propuesto.

El proceso comienza con la autenticación de un trabajador en el portal CDAE, el sistema comprueba la corrección de los datos y lo autentica. Una vez autenticado este podrá entrar al portlet sistema generador de encuestas, si es trabajador contará con todos los privilegios dentro de la aplicación; será capaz de crear, modificar y eliminar encuestas, además de poder analizar los datos de todas las encuestas almacenados en la base de datos y publicar sus resultados estadísticos, si es estudiante solo podrá acceder a responder las encuestas y consultar las estadísticas de las mismas.

2.6.2 Tarjetas de historias de usuarios

La metodología XP para desarrollar las tareas de programación, propone el control de la documentación a través de historias de usuario. Se considera que si se hace en tarjetas individuales el manejo de éstas será más fácil, por tanto se propone utilizar el siguiente modelo.

Historias de Usuarios**Número:** **Usuario:****Nombre historia:**

| | |
|---|---|
| Prioridad en negocio: (Alto/Medio/Bajo) | Riesgo en desarrollo: (Alto/Medio/Bajo) |
|---|---|

Puntos estimados: **Iteración asignada:****Programador responsable:****Descripción:****Observación:****Tabla 1 Historias de Usuarios****Campos de la tarjeta historia de usuario**

Número: Índice de la historia de usuario. Es un número único que se le asigna a cada historia de usuario con el fin de lograr una mejor organización de éstas.

Usuario: Rol que algo o alguien juega cuando interactúa con el sistema. Puede ser cualquier individuo, grupo, organización o máquina que interactúa con el sistema.

Nombre historia: Nombre de la historia de usuario. Debe ser descriptivo, en la medida de las posibilidades, de lo que se implementará y no muy extenso.

Prioridad en negocio: Prioridad que se le asigna a la historia de usuario en el negocio. Si es importante y debe ser implementada lo antes posible, entonces la prioridad debe ser alta, si no media o baja.

Riesgo en desarrollo: Riesgo que representa para el desarrollo la historia de usuario. En dependencia del riesgo que represente se le asigna tres valores, alto, medio y bajo.

Alto: historia de usuario que no pueden ser estimadas con suficiente precisión o que implementarla en iteraciones posteriores puede representar cambios sustanciales en el código base del proyecto.

Medio: historia de usuario que es estimada razonablemente bien.

Bajo: historia de usuario que es estimada sin problema alguno.

Puntos estimados: Tiempo estimado que demorará la implementación de la historia de usuario. Tiempo ideal en que se estima se implementará la historia de usuario.

Iteración asignada: Iteración en la que se implementará la historia de usuario.

Programador responsable: Programador responsable de implementar la historia de usuario.

Descripción: Se describe el requerimiento que da origen a la historia de usuario. La descripción debe ser corta, precisa y dejar claro qué es lo que se desea hacer.

Observaciones: Aspectos que se deben tener en cuenta para implementar la historia de usuario. Precondiciones o pos-condiciones que se deben tener presentes al implementar la historia de usuario.

Módulos a implementar en el sistema

Un módulo es una de las partes en las cuales se divide un programa para facilitar su ejecución. De las varias tareas que debe realizar un programa para cumplir su función u objetivo, el módulo realizará una de estas tareas o varias de ellas simultáneamente.

Historia de Módulo

Historias de Usuarios

| | |
|--|--|
| Número: 01 | Usuario: Desarrollador |
| Nombre historia: Crear encuestas | |
| Prioridad en negocio: Alta (Alto/Medio/Bajo) | Riesgo en desarrollo: Alta (Alto/Medio/Bajo) |
| Puntos estimados: 2/5 | Iteración asignada: 1 |
| Programador responsable: Yadira Tasé Torres | |
| Descripción: El trabajador del sistema crea las encuestas introduciendo los criterios para la creación de las mismas. | |
| Observación: Las encuestas deben tener título. | |

Tabla 2 Historia de usuario: "Crear encuestas"

Historia de Módulo

Historias de Usuarios

| | |
|--|---|
| Número: 02 | Usuario: Desarrollador |
| Nombre historia: Modificar encuestas | |
| Prioridad en negocio: Medio (Alto/Medio/Bajo) | Riesgo en desarrollo: Medio (Alto/Medio/Bajo) |
| Puntos estimados: 1/5 | Iteración asignada: 1 |
| Programador responsable: Yadira Tasé Torres | |
| Descripción: El trabajador busca una encuesta por coincidencia textual, el sistema muestra todas las encuestas encontradas, se selecciona la deseada y se muestra un formulario donde se modifican los datos. | |
| Observación: En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco. | |

Tabla 3 Historia de usuario: "Modificar encuestas"

Historia de Módulo**Historias de Usuarios****Número:** 03 **Usuario:** Desarrollador**Nombre historia:** Eliminar encuestas**Prioridad en negocio:** Medio **Riesgo en desarrollo:** Medio
(Alto/Medio/Bajo) (Alto/Medio/Bajo)**Puntos estimados:** 1/5 **Iteración asignada:** 1**Programador responsable:** Yadira Tasé Torres**Descripción:** El trabajador busca una encuesta por coincidencia textual, el sistema muestra un formulario con todas las encuestas encontradas, se selecciona la deseada y se elimina.**Observación:** En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco.**Tabla 4 Historia de usuario: "Eliminar encuestas"****Historia de Módulo****Historias de Usuarios****Número:** 04 **Usuario:** Desarrollador**Nombre historia:** Buscar encuestas**Prioridad en negocio:** Medio **Riesgo en desarrollo:** Medio
(Alto/Medio/Bajo) (Alto/Medio/Bajo)**Puntos estimados:** 4/5 **Iteración asignada:** 1**Programador responsable:** Yadira Tasé Torres**Descripción:** El usuario busca una encuesta por coincidencia textual y el sistema es capaz de mostrar los resultados encontrados en la búsqueda.**Observación:** En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco.**Tabla 5 Historia de usuario: "Buscar encuestas"****Historia de Módulo****Historias de Usuarios****Número:** 05 **Usuario:** Desarrollador**Nombre historia:** Realizar encuestas**Prioridad en negocio:** Alta **Riesgo en desarrollo:** Alta
(Alto/Medio/Bajo) (Alto/Medio/Bajo)**Puntos estimados:** 3 **Iteración asignada:** 2**Programador responsable:** Yadira Tasé Torres

Descripción: Después que se realice la encuesta, el sistema comienza la interacción con la base de datos y almacena los datos de las respuestas en un registro para publicar posteriormente las estadísticas.

Observación: Una vez que se realice el cuestionario se termina el proceso.

Tabla 6 Historia de usuario: "Realizar encuestas"

Historia de Módulo

Historias de Usuarios

Número: 06 **Usuario:** Desarrollador

Nombre historia: Generar estadísticas

Prioridad en negocio: Alta **Riesgo en desarrollo:** Alta
(Alto/Medio/Bajo) (Alto/Medio/Bajo)

Puntos estimados: 2 **Iteración asignada:** 1

Programador responsable: Yadira Tasé Torres

Descripción: El sistema muestra todas las encuestas que están activas, el trabajador o el usuario buscan y seleccionan la encuesta que desean consultar sus resultados y el sistema muestra un gráfico por cada pregunta con dichos resultados.

Observación: En caso de no encontrarse ninguna encuesta de acuerdo a la búsqueda se muestra la lista de encuestas en blanco.

Tabla 7 Historia de usuario: "Generar estadísticas"

2.7 Fase II: Planificación de la entrega

Durante esta fase se priorizan las historias de usuarios y se estima cuánto esfuerzo requiere cada una de ellas. Se utiliza como métrica la medida de puntos. Un punto se considera como una semana ideal de trabajo, donde el desarrollador trabaja sin interrupciones.

2.7.1 Estimación de esfuerzo por historias de usuarios.

Para el desarrollo del sistema propuesto se realizó una estimación del esfuerzo de cada una de las historias de usuarios, obteniéndose como resultados los que se muestran a continuación:

| Historia de Usuario | Puntos de estimación |
|---------------------|----------------------|
| Crear encuestas | 2/5 |
| Modificar encuestas | 1/5 |
| Eliminar encuestas | 1/5 |
| Buscar encuestas | 4/5 |
| Realizar encuestas | 3 |

| | |
|----------------------|---|
| Generar estadísticas | 2 |
|----------------------|---|

Tabla 8 Estimación de esfuerzo por HU

2.8 Fase III: Iteraciones

En esta fase se definieron tres iteraciones para la realización del sistema:

Iteración 1

Las HU que presentan mayor prioridad serán implementadas en esta primera iteración, dando al sistema las primeras funcionalidades, como por ejemplo: crear encuestas, modificar encuestas, eliminar encuestas y buscar encuestas.

Iteración 2

En la segunda iteración se implementará la HU realizar encuestas. Una vez que se finalice se contará con una versión de prueba de la aplicación que estará integrada con el portal web CDAE.

Iteración 3

Luego de haberse desarrollado las funcionalidades de alta prioridad para el sistema se elabora la HU generar estadísticas.

2.8.1 Plan de duración de iteraciones.

| Cantidad de iteraciones. | Orden de la HU a implementar. | Duración total de las iteraciones. |
|--------------------------|-------------------------------|------------------------------------|
| 1 | Crear encuestas | 10 días |
| 1 | Modificar encuestas | 10 días |
| 1 | Eliminar encuestas | 10 días |
| 1 | Buscar encuestas | 10 días |
| 2 | Realizar encuesta | 10 días |
| 3 | Generar estadísticas | 15 días |

Tabla 9 Plan de duración de las iteraciones

2.8.2 Plan de entregas.

| Historia de Usuario | Final 1 iteración (15 de abril del 2012) | Final 2 iteración (15 de mayo del 2012) | Final 3 iteración (10 de junio del 2012) |
|---------------------|---|--|---|
| Crear encuestas | x | | |
| Modificar encuestas | x | | |
| Eliminar encuestas | x | | |
| Buscar encuestas | x | | |

| | | | |
|----------------------|--|---|---|
| Realizar encuestas | | x | |
| Generar estadísticas | | | x |

Tabla 10 Plan de entregas

2.8.3 Plan de tareas

Se descompone cada HU en varias tareas que serán desarrolladas para dar cumplimiento a la misma, de esta forma se facilita el trabajo de implementación de los programadores.

| Historia de Usuario | Tareas |
|----------------------|---|
| Crear encuestas | Definir y crear los componentes visuales necesarios para mostrar la información relacionada con la encuesta y ejecutar el procedimiento de creación de preguntas. |
| Modificar encuestas | Definir y crear los componentes visuales necesarios para modificar la información deseada por el usuario y ejecutar el procedimiento de actualización de datos de las informaciones relacionadas con la encuesta. |
| Eliminar encuestas | Eliminar la información deseada por el usuario y ejecutar el procedimiento de actualización de datos de las informaciones relacionadas con la encuesta. |
| Buscar encuestas | Definir y crear los componentes visuales necesarios para mostrar los resultados arrojados según el criterio de búsqueda del usuario la información deseada por el usuario. |
| Realizar encuestas | Realizar las preguntas y almacenar los datos de las respuestas en un registro para ser publicadas las estadísticas posteriormente. |
| Generar estadísticas | Definir y crear los componentes visuales necesarios para mostrar los resultados estadísticos arrojados en forma de gráficas según el criterio de búsqueda del usuario la información deseada por el usuario. |

Tabla 11 Plan de tareas

2.9 Fase IV: Producción.

En esta fase se definen las clases del sistema, su tipo y la relación entre ellas, con todas sus funciones (atributos y métodos).

Para un mejor entendimiento se muestra a continuación un diagrama de clases con sus funcionalidades y relaciones, además de las tarjetas cargo.

2.9.1 Diagrama de clases.

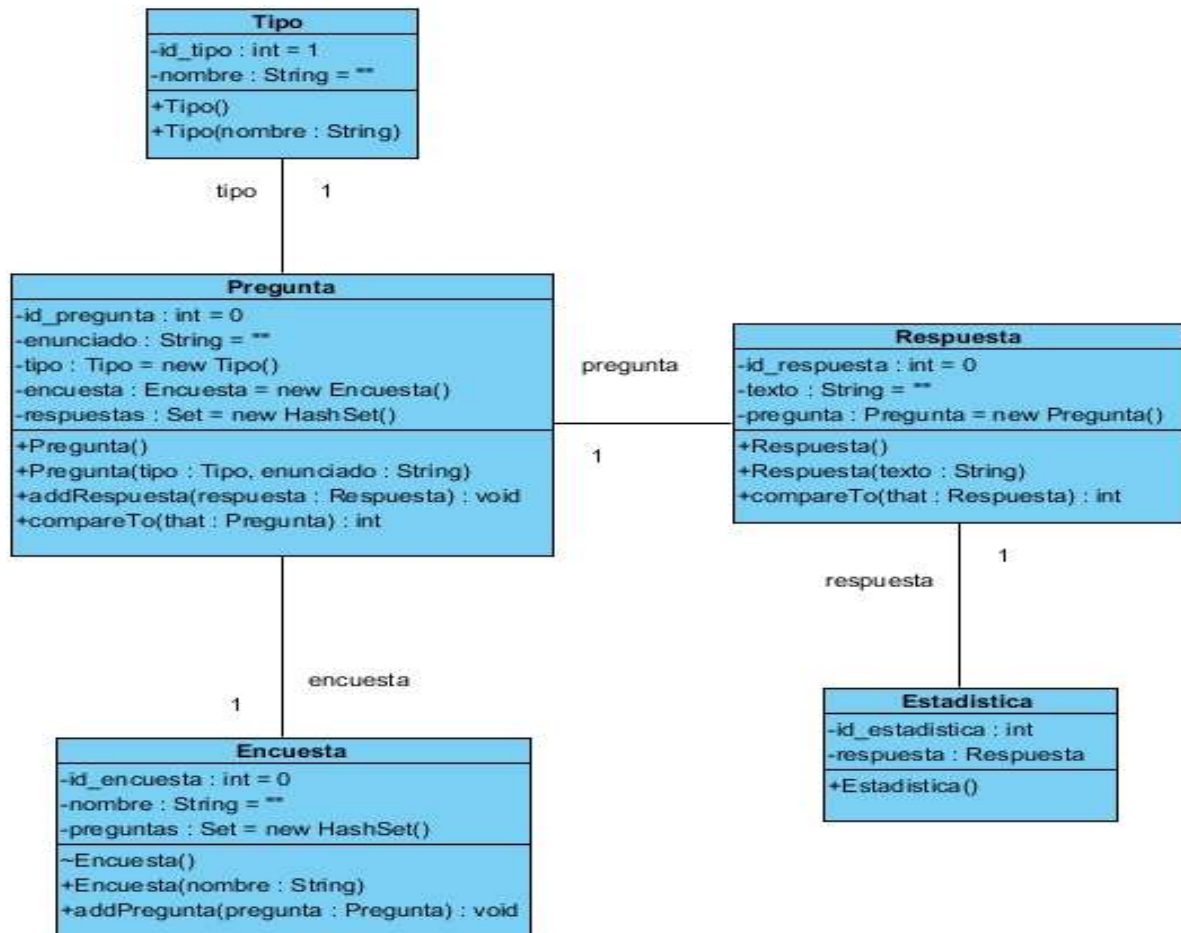


Figura 2 Diagrama de clases

2.9.2 Descripción de las clases

Tarjetas cargo o clase, responsabilidad y colaboración (CRC).

La metodología XP no recomienda la utilización de diagramas de clases, debido a la gran cantidad de documentación que requiere, aunque tampoco excluye su elaboración, en su lugar orienta la realización de Tarjetas de Cargo o CRC. Estas tarjetas se utilizan para el diseño de software orientado a objetos. Son un puente de comunicación entre diferentes participantes.

| Clase: Estadística. | |
|---------------------------------|-----------------------------------|
| Responsabilidad | Colaboración |
| Mostrar estadística de encuesta | Encuesta Pregunta Respuesta |

Tabla 12 Clase estadísticas

| Clase: Encuesta. | |
|---|--------------|
| Responsabilidad | Colaboración |
| Adicionar una encuesta. Modificar los datos de una encuesta. Eliminar una encuesta. Buscar una encuesta. | Encuesta |

Tabla 13 Clase encuesta

| Clase: Pregunta. | |
|---|--------------|
| Responsabilidad | Colaboración |
| Insertar una pregunta. Modificar una pregunta. Eliminar una pregunta. Buscar una pregunta. | Pregunta |

Tabla 14 Clase pregunta

| Clase: Respuesta. | |
|---|--------------|
| Responsabilidad | Colaboración |
| Insertar una respuesta. Modificar una respuesta. Eliminar una respuesta. Buscar una respuesta. | Respuesta |

Tabla 15 Clase respuesta

2.10 Diseño de la Base de Datos

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas entre otros. Por lo general se hace a través de un modelo, pues permite describir los tipos de datos y la forma en que se relacionan, así como las restricciones de integridad y las operaciones de manipulación de los mismos. Este modelo describe la representación lógica y física de los datos persistentes. (21)

2.10.1 Diagrama de las clases persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Algunas de las reglas que se recomiendan aplicar son:

- Cuando una clase está formada por otras clases es persistente, automáticamente las que la componen también son persistentes. Lo contrario no se cumple necesariamente.
- Cuando una clase hija de una jerarquía es persistente, automáticamente son persistentes sus ancestros en el árbol de jerarquía. Lo contrario no se cumple necesariamente.
- Cuando se define como persistente a una clase que agrupa a objetos de un mismo tipo de la clase base (listas, colecciones, registros), entonces son persistentes todas las clases hijas a partir de la clase base, incluyéndose la misma.
- Cuando hay herencia múltiple, esta debe ser resuelta antes si el medio de almacenamiento no soporta este concepto.
- El tratamiento a los datos es algo primordial en el sistema, pues permite que los resultados de las encuestas perduren.

Para el almacenamiento de las encuestas se crea un conjunto de tablas en correspondencia con el diseño de la misma. Este modelo facilita el posterior procesamiento de los datos que serán guardados luego que se publica la encuesta. (22)

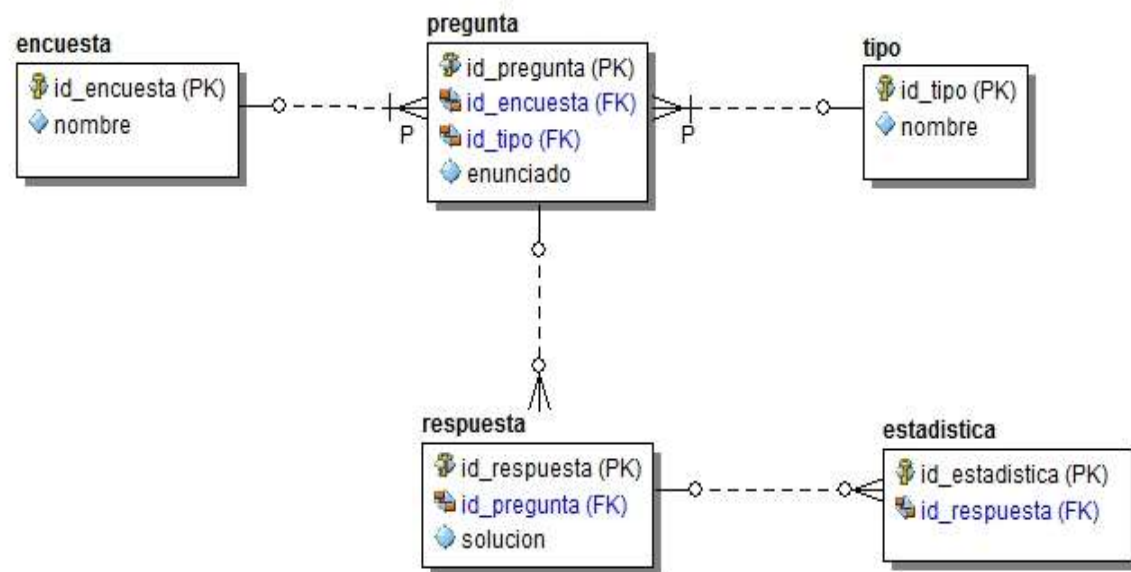


Figura 3 Modelo de datos

2.11 Modelo de despliegue.

El modelo de despliegue especifica la distribución física de la aplicación, mostrando cómo se encuentran repartidas las capas arquitectónicas en los niveles correspondientes. Indicando además, los enlaces de comunicación que existen entre cada instancias de componentes en ejecución.

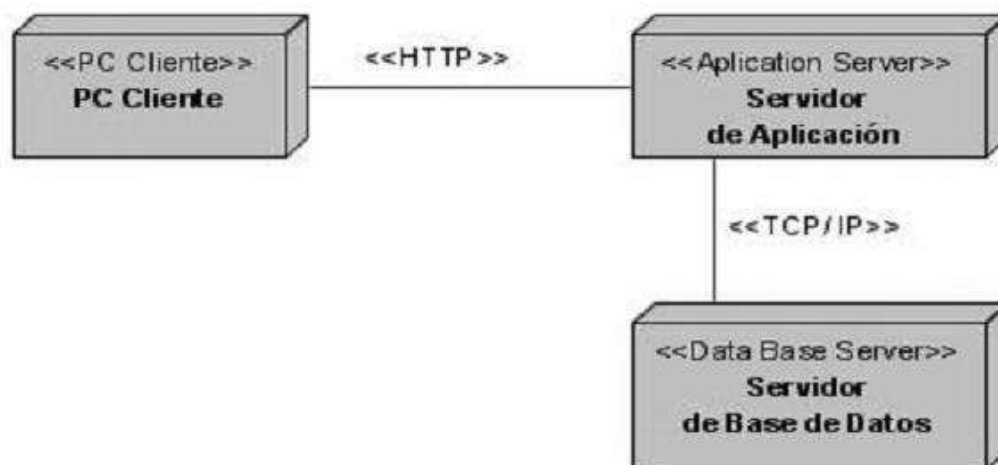


Figura 4 Modelo de despliegue

2.12 Conclusiones Parciales.

En este capítulo se pasó a la definición de los distintos módulos. Para ello se realizó un profundo análisis de la solución propuesta utilizando los artefactos que propone la metodología Extreme Programming (XP). Es decir, en esta fase, una vez definidos los módulos que se implementarían, se establecieron las Historias de Usuarios correspondientes a cada uno de ellos. Por último se crearon las Tareas de programación necesarias para dar cumplimiento a las necesidades del cliente expuestas en dichas Historias de Usuarios.

Capítulo 3 Implementación y validación de la solución propuesta.

3.1 Introducción

XP plantea que se requiere prueba y comprobación extra del funcionamiento del sistema antes de que este se pueda liberar al cliente. En esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de incluirlos o no en el release actual.

3.2 Tarjeta de tareas de programación

Las tareas de programación corresponden a una historia de usuario. Pero a una historia de usuario pudieran corresponderle varias tarjetas de tareas de programación, para dar respuestas a las necesidades del cliente. El tipo de tarjetas que se propone adoptan el siguiente modelo:

| Tarea | |
|---------------------------------|------------------------------------|
| Número tarea: | Número historia de usuario: |
| Nombre tarea: | |
| Tipo tarea: | Puntos estimados: |
| Fecha inicio: | Fecha fin: |
| Programador responsable: | |
| Descripción: | |

Tabla 16 Tarjeta de tareas de programación

Campos de la tarjeta de tarea de programación

Número tarea: Índice de la tarea de programación. Es un número único que se le asigna a cada tarea de programación que pertenece a una historia de usuario determinada con el fin de lograr una mejor organización de éstas.

Número historia de usuario: Número de la historia de usuario a la que corresponde. Índice de la historia de usuario a la que se corresponde esta tarea de programación.

Nombre tarea: Nombre de la tarea de programación. Debe ser descriptivo, en la medida de las posibilidades, de lo que se realizará y no muy extenso.

Tipo de tarea: Informa el tipo de tarea a realizar. Las tareas pueden ser.

Puntos estimados: Tiempo estimado que demorará la implementación de la tarea de programación. Tiempo ideal en que se estima se implementará la tarea de programación.

Fecha inicio: Fecha en que se inicia la implementación de la tarea de programación.

Fecha fin: Fecha en que concluye la implementación de la tarea de programación.

Programador responsable: Programador responsable de implementar la tarea de programación.

Descripción: Se describe qué es lo que se desea realizar. La descripción debe ser corta y precisa.

3.3 Implementación.

En la implementación se lleva a cabo la producción del software. El desarrollador se encarga de implementar la funcionalidad especificada en las historias de usuarios mediante el lenguaje de programación elegido. En paralelo a la construcción de la aplicación se realiza la revisión de la misma mediante las pruebas; la metodología define diversas pruebas que podrán ser realizadas en las iteraciones. Las mismas incluyen: Pruebas unitarias y Pruebas de aceptación.

3.3.1 Iteración 1.

En esta iteración se desarrollan las HU que presentan mayor prioridad, dando al sistema las primeras funcionalidades, estas son: crear encuestas, modificar encuestas y eliminar encuestas.

Historias de usuario implementadas en la primera iteración.

| Historias de Usuarios | Estimación | Real |
|-----------------------|------------|------|
| Crear encuestas | 2/5 | 2/5 |
| Modificar encuestas | 1/5 | 1/5 |
| Eliminar encuestas | 1/5 | 1/5 |
| Buscar encuestas | 2/5 | 4/5 |

Tabla 17 HU implementadas en la primera iteración

Tareas de las historias de usuario implementadas en la primera iteración.

Tareas implementadas de la HU: "Crear encuestas"

Tareas del Módulo

Tarea para el Módulo de Gestionar encuestas

Número tarea: 01

Número historia de usuario: 01

Nombre tarea: Diseñar interfaz de crear encuestas.

Tipo tarea: Desarrollo

Puntos estimados: 2/5

Fecha inicio: 7/03/2012

Fecha fin: 9/03/2012

Programador responsable: Yadira Tasé Torres

Descripción: Se diseña la interfaz de la funcionalidad Crear Encuesta, permitiendo que se introduzcan los datos para la creación de la encuesta.

Tabla 18 Tarea #1 de la HU: "Crear encuesta"

Tareas del Módulo**Tarea para el Módulo de Gestionar encuestas****Número tarea:** 02**Número historia de usuario:** 01

Nombre tarea: Implementar las funcionalidades que permitan la creación de las encuestas.

Tipo tarea: Desarrollo**Puntos estimados:** 2/5**Fecha inicio:** 9/03/2012**Fecha fin:** 14/03/2012**Programador responsable:** Yadira Tasé Torres

Descripción: Se implementa la funcionalidad Crear Encuesta, permitiendo la confección de la encuesta con los datos introducidos (nombre de la encuesta, texto de la pregunta, texto de las respuestas), estos son guardados en sus tablas correspondientes en la base de datos.

Tabla 19 Tarea #2 de la HU: "Crear encuesta"**Tareas implementadas de la HU: "Modificar encuestas"****Tareas del Módulo****Tarea para el Módulo de Gestionar encuestas****Número tarea:** 01**Número historia de usuario:** 02

Nombre tarea: Diseñar interfaz de modificar encuestas

Tipo tarea: Desarrollo**Puntos estimados:** 1/5**Fecha inicio:** 14/03/2012**Fecha fin:** 16/03/2012**Programador responsable:** Yadira Tasé Torres

Descripción: Se diseña la interfaz de la funcionalidad Modificar Encuesta, permitiendo que se modifiquen los campos deseados.

Tabla 20 Tarea #1 de la HU: "Modificar encuesta"**Tareas del Módulo****Tarea para el Módulo de Gestionar encuestas****Número tarea:** 02**Número historia de usuario:** 02

Nombre tarea: Implementar las funcionalidades que permitan la modificación de las encuestas.

Tipo tarea: Desarrollo**Puntos estimados:** 1/5**Fecha inicio:** 16/03/2012**Fecha fin:** 21/03/2012**Programador responsable:** Yadira Tasé Torres

Descripción: Se implementa la funcionalidad Modificar Encuesta, permitiendo la actualización de los datos que se deseen de la encuesta, estos son modificados en sus tablas correspondientes en la base de datos.

Tabla 21 Tarea #2 de la HU: "Modificar encuesta"

Tareas implementadas de la HU: "Eliminar encuestas"

Tareas del Módulo

Tarea para el Módulo de Gestionar encuestas

Número tarea: 01

Número historia de usuario: 03

Nombre tarea: Implementar las funcionalidades que permitan la eliminación de encuestas.

Tipo tarea: Desarrollo

Puntos estimados: 1/5

Fecha inicio: 21/03/2012

Fecha fin: 27/03/2012

Programador responsable: Yadira Tasé Torres

Descripción: Se implementa la funcionalidad Eliminar Encuesta, que permite eliminar la encuesta seleccionada. Los datos de la encuesta son borrados de su tabla correspondiente en la base de datos.

Tabla 22 Tarea #1 de la HU: "Eliminar encuesta"

Tareas implementadas de la HU: "Buscar encuestas"

Tareas del Módulo

Tarea para el Módulo de Buscar encuestas

Número tarea: 01

Número historia de usuario: 04

Nombre tarea: Implementar las funcionalidades que permitan la búsqueda de las encuestas.

Tipo tarea: Desarrollo

Puntos estimados: 4/5

Fecha inicio: 27/03/2012

Fecha fin: 7/04/2012

Programador responsable: Yadira Tasé Torres

Descripción: Se implementa la funcionalidad Buscar Encuesta, permitiendo listar todas las encuestas encontradas después de buscar por coincidencia textual con los valores introducidos.

Tabla 23 Tarea #1 de la HU: "Buscar encuesta"

Tareas del Módulo**Tarea para el Módulo de Buscar encuestas****Número tarea:** 02**Número historia de usuario:** 04**Nombre tarea:** Conexión con la base de datos.**Tipo tarea:** Desarrollo**Puntos estimados:** 1/5**Fecha inicio:** 7/04/2012**Fecha fin:** 11/04/2012**Programador responsable:** Yadira Tasé Torres**Descripción:** Se realiza la conexión con la base de datos definida para la aplicación, haciendo uso de una clase HibernateUtil.**Tabla 24 Tarea #2 de la HU: "Buscar encuesta"****Tareas del Módulo****Tarea para el Módulo de Gestionar encuestas****Número tarea:** 08**Número historia de usuario:** 01, 02, 03, 04**Nombre tarea:** Prueba de la aplicación gestionar encuestas.**Tipo tarea:** Desarrollo**Puntos estimados:** 10 horas**Fecha inicio:** 11/04/2012**Fecha fin:** 15/04/2012**Programador responsable:** Yadira Tasé Torres**Descripción:** Se realizan una serie de pruebas funcionales a la aplicación, con la finalidad de obtener posibles fallas al realizar estas pruebas.**Tabla 25 Tarea #8 Aplicar pruebas****3.3.2 Iteración 2.**

En la segunda iteración se implementará la HU realizar las encuestas con una alta prioridad.

Historias de usuario implementadas en la segunda iteración.

| Historias de Usuarios | Estimación | Real |
|-----------------------|------------|------|
| Realizar encuestas | 3 | 3 |

Tabla 26 HU implementadas en la segunda iteración**Tareas de las historias de usuario implementadas en la segunda iteración.****Tareas implementadas de la HU: "Realizar encuestas"****Tareas del Módulo****Tarea para el Módulo de Realizar encuestas****Número tarea:** 01**Número historia de usuario:** 05

Nombre tarea: Realizar encuesta

Tipo tarea: Desarrollo **Puntos estimados:** 3

Fecha inicio: 17/04/2012 **Fecha fin:** 28/04/2012

Programador responsable: Yadira Tasé Torres

Descripción: Se implementa la funcionalidad Realizar Encuesta, permitiendo llenar un cuestionario con los datos necesarios para la creación de un registro con los datos recopilados (id de la encuesta y la respuesta seleccionada) después de finalizada la encuesta.

Tabla 27 Tarea #1 de la HU: "Realizar encuesta"

Tareas del Módulo

Tarea para el Módulo de Realizar encuestas

Número tarea: 02 **Número historia de usuario:** 05

Nombre tarea: Conexión con la base de datos

Tipo tarea: Desarrollo **Puntos estimados:** 3

Fecha inicio: 28/04/2012 **Fecha fin:** 4/05/2012

Programador responsable: Yadira Tasé Torres

Descripción: Se realiza la conexión con la base de datos definida para la aplicación, haciendo uso de una clase HibernateUtil.

Tabla 28 Tarea #2 de la HU: "Realizar encuesta"

Tareas del Módulo

Tarea para el Módulo de Realizar encuestas

Número tarea: 03 **Número historia de usuario:** 05

Nombre tarea: Prueba de la aplicación gestionar encuestas.

Tipo tarea: Desarrollo **Puntos estimados:** 3

Fecha inicio: 4/05/2012 **Fecha fin:** 14/05/2012

Programador responsable: Yadira Tasé Torres

Descripción: Se realizan una serie de pruebas funcionales a la aplicación, con la finalidad de obtener posibles fallas al realizar estas pruebas.

Tabla 29 Tarea #3 de la HU: "Realizar encuesta"

3.3.3 Iteración 3.

Luego de haberse desarrollado las funcionalidades de alta prioridad para el sistema se confeccionara en este capítulo la HU generar estadísticas siendo la última iteración a realizar sobre el sistema.

Historias de usuario implementadas en la tercera iteración.

| Historias de Usuarios | Estimación | Real |
|-----------------------|------------|------|
| Generar estadísticas | 2 | 2 |

Tabla 30 HU implementadas en la tercera iteración**Tareas de las historias de usuario implementadas en la tercera iteración.****Tareas implementadas de la HU: "Generar estadísticas"****Tareas del Módulo****Tarea para el Módulo de Generar estadísticas.**

| | |
|-------------------------|---------------------------------------|
| Número tarea: 01 | Número historia de usuario: 06 |
|-------------------------|---------------------------------------|

Nombre tarea: Implementar las funcionalidades que permitan la generación de estadísticas.

| | |
|-------------------------------|----------------------------|
| Tipo tarea: Desarrollo | Puntos estimados: 2 |
|-------------------------------|----------------------------|

| | |
|---------------------------------|------------------------------|
| Fecha inicio: 15/05/2012 | Fecha fin: 25/05/2012 |
|---------------------------------|------------------------------|

Programador responsable: Yadira Tasé Torres

Descripción: Se implementa la funcionalidad Generar estadísticas, permitiendo llenar un cuestionario con los datos necesarios para la creación de un registro con los datos recopilados y la muestra de gráficas con los resultados de las encuestas.

Tabla 31 Tarea #1 de la HU: "Generar estadísticas"**Tareas del Módulo****Tarea para el Módulo de Generar estadísticas.**

| | |
|-------------------------|---------------------------------------|
| Número tarea: 02 | Número historia de usuario: 06 |
|-------------------------|---------------------------------------|

Nombre tarea: Conexión con la base de datos

| | |
|-------------------------------|----------------------------|
| Tipo tarea: Desarrollo | Puntos estimados: 2 |
|-------------------------------|----------------------------|

| | |
|---------------------------------|-----------------------------|
| Fecha inicio: 25/05/2012 | Fecha fin: 2/06/2012 |
|---------------------------------|-----------------------------|

Programador responsable: Yadira Tasé Torres

Descripción: Se realiza la conexión con la base de datos definida para la aplicación, haciendo uso de una clase HibernateUtil.

Tabla 32 Tarea #2 de la HU: "Generar estadísticas"

Tareas del Módulo**Tarea para el Módulo de Generar estadísticas.****Número tarea:** 03**Número historia de usuario:** 06**Nombre tarea:** Prueba de la aplicación gestionar encuestas.**Tipo tarea:** Desarrollo**Puntos estimados:** 2**Fecha inicio:** 2/06/2012**Fecha fin:** 10/06/2012**Programador responsable:** Yadira Tasé Torres

Descripción: Se realizan una serie de pruebas funcionales a la aplicación, con la finalidad de obtener posibles fallas al realizar estas pruebas.

Tabla 33 Tarea #3 de la HU: "Generar estadísticas"**3.4 Pruebas.**

El desarrollo de software implica la realización de una serie de actividades entre las que se encuentran las pruebas, siendo esta una actividad que garantiza la calidad del producto desarrollado.

La metodología XP anima a probar constantemente tanto como sea posible, siendo las pruebas uno de sus pilares fundamentales. Esto permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados. También permite aumentar la seguridad de evitar efectos colaterales no deseados al realizar modificaciones y refactorizaciones. (23)

El sistema de programación extrema divide las pruebas en dos grupos: pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final y pruebas unitarias, encargadas de verificar el código y diseñada por el programador. Además se realizarán pruebas de rendimiento para validar y verificar diferentes aspectos de la calidad de software respecto a la respuesta de las peticiones realizadas a la aplicación. (24)

3.4.1 Pruebas de aceptación.

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. Se deben especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Son consideradas como “pruebas de caja negra” (“Black box system tests”). Se definen para cada historia de usuario. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de solución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Estas pruebas tienen una importancia crítica para el éxito de una iteración..

Pruebas de aceptación del sistema.

A continuación se muestran algunos casos de pruebas relacionados con las historias de usuarios crear, modificar, eliminar, buscar y realizar encuestas.

| Prueba de aceptación | |
|--|----------------|
| Código: HU01-P01 | HU: #01 |
| Nombre: Crear encuestas | |
| Descripción: Esta funcionalidad será la encargada de crear una encuesta en el sistema. Todos los datos asociados al tema de la encuesta serán importados en una lista de encuestas. | |
| Condiciones de ejecución: El usuario debe de estar autenticado y tener privilegios de trabajador para tener acceso a este servicio. | |
| Entrada/Pasos de ejecución: Se selecciona el botón generar encuestas y se comienza el procedimiento de crear una encuesta. | |
| Resultado esperado: Mostar una encuesta. | |
| Resultado obtenido: Se muestra una encuesta. | |
| Evaluación de la prueba: Satisfactoria | |

Tabla 34 Prueba de aceptación para HU: #1

| Prueba de aceptación | |
|---|----------------|
| Código: HU02-P01 | HU: #02 |
| Nombre: Modificar encuestas | |
| Descripción: Esta funcionalidad será la encargada de agregar o cambiar algunos datos de interés de las encuestas vinculadas al centro que se encuentren registrados en el sistema. | |
| Condiciones de ejecución: El usuario debe de estar autenticado y tener privilegios de trabajador para tener acceso a este servicio. | |
| Entrada/Pasos de ejecución: Se entran los datos a modificar de la encuesta correctamente. | |
| Resultado esperado: Mostrar la encuesta con los datos modificados. | |
| Resultado obtenido: Se muestra la encuesta con los datos modificados. | |
| Evaluación de la prueba: Satisfactoria | |

Tabla 35 Prueba de aceptación para HU: #2

| Prueba de aceptación | |
|--|----------------|
| Código: HU03-P01 | HU: #03 |
| Nombre: Eliminar encuestas | |
| Descripción: Esta funcionalidad será la encargada de eliminar encuestas vinculadas al centro que se encuentren registrados en el sistema. | |
| Condiciones de ejecución: El usuario debe de estar autenticado y tener privilegios de trabajador para tener acceso a este servicio. | |
| Entrada/Pasos de ejecución: Se selecciona la encuesta que se desea eliminar. | |
| Resultado esperado: Eliminar correctamente la encuesta seleccionada. | |
| Resultado obtenido: Se elimina correctamente la encuesta seleccionada. | |
| Evaluación de la prueba: Satisfactoria | |

Tabla 36 Prueba de aceptación para HU: #3

| Prueba de aceptación | |
|--|----------------|
| Código: HU04-P01 | HU: #04 |
| Nombre: Buscar encuestas | |
| Descripción: Esta funcionalidad será la encargada de filtrar según el criterio que desee el usuario las encuestas vinculados al centro que se encuentren registrados en el sistema. | |
| Condiciones de ejecución: El usuario debe de estar autenticado y tener privilegios de trabajador para tener acceso a este servicio. | |
| Entrada/Pasos de ejecución: Se selecciona la encuesta que se desea buscar de la lista de encuestas que aparece en el sistema. | |
| Resultado esperado: Obtener la respuesta de forma inmediata. | |
| Resultado obtenido: Se obtiene la respuesta de forma inmediata. | |
| Evaluación de la prueba: Satisfactoria | |

Tabla 37 Prueba de aceptación para HU: #4

| Prueba de aceptación | |
|--|----------------|
| Código: HU05-P01 | HU: #05 |
| Nombre: Realizar encuestas | |
| Descripción: Esta funcionalidad será la encargada de realizar las encuestas | |

| |
|--|
| vinculados al centro que se encuentren registrados en el sistema. |
| Condiciones de ejecución: El usuario debe de estar autenticado para tener acceso a este servicio. |
| Entrada/Pasos de ejecución: Se selecciona la encuesta que se desea realizar de la lista de encuestas que aparece en el sistema. |
| Resultado esperado: Realizar una encuesta de forma correcta. |
| Resultado obtenido: Se realizó una encuesta de forma correcta. |
| Evaluación de la prueba: Satisfactoria |

Tabla 38 Prueba de aceptación para HU: #5

3.4.2 Pruebas unitarias

Las pruebas unitarias o pruebas de unidad consisten en comprobaciones (manuales o automatizadas) desarrolladas por los programadores. Las cuales se realizan para verificar que el código correspondiente a un módulo concreto se comporta de manera esperada. Las pruebas unitarias proporcionan beneficios tales como:

El programador puede realizar cambios de forma segura respaldada por efectivos casos de prueba.

Permite saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo.

Brindan al programador una inmediata retroalimentación de cómo está realizando su trabajo.

En el transcurso del desarrollo de la aplicación se aplicaron una serie de este tipo de pruebas, que ayudaron a corregir errores encontrados dentro del código fuente del sistema. Un ejemplo lo constituye el siguiente fragmento:

| Cantidad de pruebas realizadas. | Cantidad de pruebas satisfactorias. | Cantidad de errores encontrados | Cantidad de errores corregidos. |
|---------------------------------|-------------------------------------|---------------------------------|---------------------------------|
| 56 | 50 | 30 | 30 |

Tabla 39 Cantidad de pruebas unitarias realizadas

3.4.3 Pruebas de rendimiento.

Considerando que la metodología propone las pruebas de aceptación y las unitarias, se hace necesario realizar las pruebas de rendimiento debido al tipo de software que se desarrolla. Las pruebas de rendimiento son aquellas que son realizadas para determinar qué tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. (25)

Estas pruebas son utilizadas con diferentes propósitos; como demostrar que el sistema cumple los criterios de rendimiento establecidas, medir qué partes del sistema muestran un bajo rendimiento son ejemplos del uso que se les puede dar a este tipo de pruebas. El uso de herramientas que permitan el monitoreo y diagnóstico del producto es de gran importancia para realizar tareas de este tipo. (26)

Pruebas de rendimiento del sistema.

| Hardware | |
|---------------------|-----------|
| Tipo de Procesador | Core Duo |
| Memoria del Sistema | 4 gigas R |
| Tipo de Red | LAN |

Tabla 40 Características del ambiente de prueba

| Software | |
|------------------------------|------------------|
| Tipo de Servidor web | Apache Tomcat |
| Memoria máxima | 4 g RAM |
| Máximo de hilos concurrentes | 50/páginas |
| Plataforma | Linux Centos 5.2 |
| Servidor de base de datos | PostgreSQL 8.4 |

Tabla 41 Características del ambiente de prueba

Resultados obtenidos.

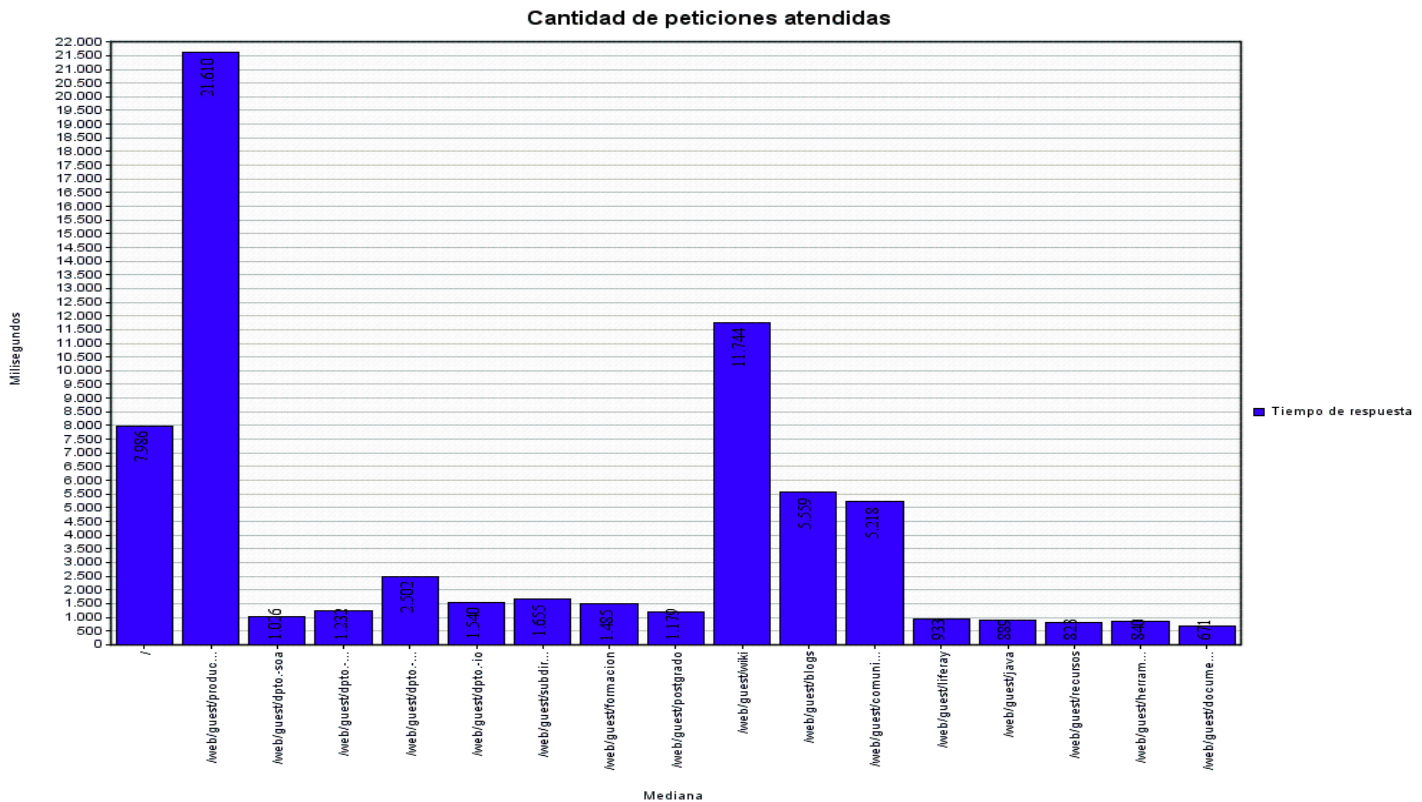


Figura 5 Cantidad de peticiones atendidas

3.5 Conclusiones parciales.

Como resultado de este capítulo se detallaron las historias de usuario de cada una de las iteraciones definidas, haciendo uso de las tareas de ingeniería identificadas por historia de usuario, elaborando a su vez las pruebas de aceptación para confirmar que las historias de usuarios han sido implementadas correctamente. Se obtuvo la implementación del sistema, proporcionando solución a los requisitos especificados en el capítulo anterior. Para ellos se representaron las clases persistentes y la descripción de la estructura de tablas sobre las que se establece el modelo de datos del sistema, así como sus atributos y relaciones. Una vez concebida la estructura y la implementación del sistema, se realizaron los casos de pruebas para validar la correcta construcción de los requerimientos, obteniéndose resultados satisfactorios.

Conclusiones generales

Una vez culminado el trabajo es posible afirmar que se les dio cumplimiento a los objetivos trazados para el mismo:

- La metodología XP, la herramienta CASE Visual Paradigm y el lenguaje de modelado UML utilizados en el desarrollo del software permitieron la realización del análisis con la calidad requerida.
- Se diseñó e implementó una base de datos para gestionar la información de las encuestas.
- Las pruebas realizadas para validar las funcionalidades que fueron implementadas, demostraron que los indicadores de calidad cumplieron satisfactoriamente con los requisitos, garantizando su correcto funcionamiento.

Recomendaciones

El objetivo general de este trabajo ha sido logrado, pero a lo largo de su desarrollo, han surgido ideas que podrían implementarse en un futuro, de forma que se logre una aplicación más útil y efectiva, para lo cual se recomienda:

- Probar las encuestas creadas en una sub-muestra de entre 100 y 150 personas para garantizar que cumpla con los objetivos trazados por el encuestador de la misma.
- Fortalecer el sistema con la posibilidad de realizar una salva local de los datos en caso de fallo en la comunicación entre el cliente y el servidor.
- Realizar un módulo de gestión de usuarios en caso de ser necesario.

Referencias Bibliográficas

1. Campbell y Stanley. "Tipos de encuesta y diseños de investigación". [En línea]. http://www.unavarra.es/personal/vidaldiaz/pdf/tipos_encuestas.PDF
2. **Meléndez, Jorge**. "El humanismo frente al desarrollo científico y tecnológico en la educación del ser humano. Sitio Web: Observatorio Ciudadano de la Educación". [\[http://www.observatorio.org/colaboraciones/melendrez4.html\]](http://www.observatorio.org/colaboraciones/melendrez4.html).
3. **Stanton, Etzel y Walker, Mc Graw Hill**. Fundamentos de Marketing.
4. **Sandhusen, L. Richard**. Mercadotecnia. Primera. s.l.: Compañía Editorial.
5. **Malhotra Naresh, Prentice Hall**. Investigación de Mercados Un Enfoque Práctico.
6. Diccionario de Marketing Cultural S.A. 1999. pág. 113.
7. **O'Reilly, Tim**. "Que es web 2.0. Patrones del diseño y modelos del negocio para la siguiente generación del software". [En línea] 2006. <http://sociadaddelainformacion.telefonica.es/jsp/articulo/detalle.jsp?elem=2146>
8. **Über, Denken**. "Los 10 principios de una aplicación web exitosa". [En línea] 2010. <http://www.uberbin.net/archivos/tips/10-principios-de-una-aplicacion-web-exitosa.php>
9. **Vegas, Jesús**. "Introducción a las Aplicaciones Web". [En línea] 2002. <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>
10. **WebOpedia**. "Portlets". [En línea] 2012. <http://www.webopedia.com/TERM/P/portlet.html>
11. **Java Enterprise**. "Introducción a Vaadin, un framework para RIA". [En línea] 2010. http://www.javamexico.org/blogs/ezamudio/introduccion_vaadin_un_framework_para_ria.html
12. **Lijin**. "12 benefits of Spring MVC over Struts". [En línea] 2006. <http://lijinoseji.wordpress.com/2006/11/10/12-benefits-of-spring-mvc-over-struts>
13. **Hibernate Community**. "Persistencia relacional para Java idiomático". [En línea] 2012. <http://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html>
14. **JFreeChart Official Site**. [En línea] 2012. <http://www.jfree.org/jfreechart>
15. **JS Chart Web Site**. [En línea] 2012. <http://www.jscharts.com>
16. **Highcharts Web Site**. "What is Highcharts?" [En línea] 2012. <http://www.highcharts.com/products/Highcharts>

17. **PARC, XEROX.** "MVC". [En línea] 2003. <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
18. **Burbeck, Steve Ph.D.** "How to use Model-View-Controller (MVC)". [En línea] 1997. <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>
19. **Universidad de Alcalá.** "Patrones de diseño: Grasp". [En línea] 2009. <ftp://www.cc.uah.es/pub/Alumnos/Informática/Patrones.Software/GRASP/PatDisGRASP.pdf>
20. **Gracia, Joaquín.** "Diseño de Software Orientado a Objetos". [En línea] 2005. <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
21. **Costa, D.C.** Introducción al diseño de bases de datos.
22. **González, A.H.** Un método para el diseño de la base de datos a partir del modelo orientado a objeto. Vol. 7. 2004.
23. **J. J. Gutiérrez, M.J.E., M. Mejías, J. Torres,** PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA.
24. **Juan Pablo Cassinelli, R.S., Dayvis Malfará, Diego Cukerman, Fernando Cócaro,** Testing en eXtreme Programming.
25. **Pruebas de Rendimiento.,** <http://www.corporacionsybven.com/Sybven>
26. **CS, Corporación.** "Pruebas de Rendimiento". [En línea] 2012. http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246

Bibliografía

1. **O'Reilly, Tim**. "Qué es web 2.0. Patrones del diseno y modelos del negocio para la siguiente generacion del software".[en línea] 2006.
<http://sociedaddelainformacion.telefonica.es/jsp/articulo/detalle.jsp?elem=2146>
2. **Letelier, Patricio**. Metodologías Ágiles en el Desarrollo de Software. [En línea] 2008.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.4553&rep=rep1&type=pdf#page=9>.
3. **Letelier, Patricio y Penadés , Carmen**. "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)".2008. Universidad Politécnica de Valencia.España.
4. **Quiñones, A.** Introducción a PostgreSQL. [En línea] 2007.
http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf.
5. **García, Aitor**. "Guía del autoestopista a Hibernate".2007

GLOSARIO DE TÉRMINOS

AJAX: (Asynchronous JavaScript And XML), JavaScript asíncrono y XML, es una técnica de desarrollo web para crear aplicaciones interactivas.

Apache: Es un software libre, servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

API: (Application Programming Interface). Una interfaz de programación de aplicaciones es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Aplicación: Programa informático creado para facilitar al usuario un determinado tipo de trabajo

BD: Base de Datos.

CASE: (Computer Aided Software Engineering). Ingeniería de Software Asistida por Computación.

GRASP: (General Responsibility Assignment Software Patterns), son patrones generales de software para asignación de responsabilidades. Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

Hardware: Componentes físicos que constituyen las computadoras y demás dispositivos periféricos.

HTML: (HyperText Markup Language). Lenguaje de Marcado de Hipertexto, es el lenguaje de marcado predominante para la elaboración de páginas web.

HTTP: (HyperText Transfer Protocol) .Protocolo de transferencia de hipertexto usado en cada transacción de la World Wide Web.

Lenguaje de Programación: Es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

LINUX: Es un núcleo de sistema operativo libre tipo Unix. Es uno de los principales ejemplos de software libre.

MVC: Modelo Vista Controlador, es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Open Source: Código abierto, es el término con el que se conoce al software distribuido y desarrollado libremente.

Open UP: Es un método y un proceso de desarrollo de software propuesto por un conjunto de empresas de tecnología.

ORM: (Object-Relational mapping), Mapeo objeto-Relacional, técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

PgAdmin: Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source.

Plugin: Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como complemento.

PostgreSQL: Es un sistema de gestión de base de datos relacional orientada a objetos y libre.

Software: Equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP). El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos.

UNIX: Es un sistema operativo de tiempo compartido, controla los recursos de una computadora y los asigna entre los usuarios. Permite a los usuarios correr sus programas. Controla los dispositivos de periféricos conectados a la máquina.

URL: (Uniform Resource Locator). Localizador de Recurso Uniforme, dirección global de documentos y de otros recursos en la World Wide Web.

USB: Universal Serial Bus, puerto que sirve para conectar periféricos a una computadora.

WEB: World Wide Web también conocida como la web, el sistema de documentos o páginas web interconectados por enlaces de hipertexto, disponibles en Internet.

Windows: Es un sistema operativo con interfaz gráfica para computadoras personales propiedad de la empresa Microsoft. Windows es el sistema operativo más utilizado en el mundo.

Word: (Portable Document Format) Formato de Documento Portátil), es un formato de almacenamiento de documentos.

XP: (eXtreme Programming) La programación extrema es un enfoque de la ingeniería de software. Se puede considerar como la adopción de las mejores metodologías de desarrollo de acuerdo a lo

que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.