Universidad de las Ciencias Informáticas

Facultad 4

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

Módulo para la exportación de reportes del Juez en Línea Caribeño

Autor: Jorge Gabriel Díaz Reinoso

Tutor: Ing. Angel Alberto Vazquez Sánchez

La Habana, Cuba, junio del 2015

Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Jorge Gabriel Díaz Reinoso

_____

Firma del Autor

Ing. Angel Alberto Vazquez Sánchez

_____

Firma del Tutor

Agradecimientos

Agradecer a todos los que de una forma u otra me han ayudado a alcanzar este logro personal. A los que me brindaron su mano amiga cuando pensaba que todo estaba perdido desde lo más profundo de mi corazón gracias.

Agradecer a mi tutor Angel Alberto Vazquez por todo el apoyo y dedicación que me brindó, por toda su ayuda, por batallar siempre por mí, por ser un gran profesor y un excelente amigo.

Dar gracias a Andy Cabrera por todo su esfuerzo, consagración y ayuda cuando estaba desesperado y sin esperanzas.

Agradecer a Orlando Grabiel, a Eddy Roberto, a Lizandra Hernández, a Renides y a Ocleidy por su ayuda y recomendaciones.

A Carlos Gonzales por todo, por su ayuda, apoyo incondicional y por ser un gran compañero.

A mis compañeros de aula los que vienen arrastrando conmigo desde 1er año y los que se han ido incorporando durante estos 5 años. En especial a mis hermanos Juan Carlos, Dariel, Armando, Kelvis el muslete y Claudia de León, al Sifre, a Jesús el titi, Yannier, Wilson, Roberto, Osiel, Leonardo el gordo y Leonardo el que más permite.

A las poquitas niñas del aula Yanet, Yaritza, Nayara, Lianne, Nory y Anadelys.

A mis amigos de los Palacios, Daniel el gago, a Yasmani, a Ricardito, al Vledo, al Valdi, al Piku, al Cachero, Osmani y al Cala.

A Claudia Anisel y a su familia por brindarme todo su apoyo y amistad en todo momento; por preocuparse siempre, por eso y por mucho más estarán presentes eternamente para mí.

A mis padres Lumi y Fermín por siempre estar en los momentos buenos y malos a mi lado aconsejándome y amándome.

A mi abuela mima, a mi abuela Ada, a mi tía Bárbara, a Carlitos, a Teo, a mi tío Rene, a mi hermana Heidy, a Román, a toda mi familia. A ellos mi mayor agradecimiento.

Dedicatoria

A mi mamá Lumi, lo que más quiero en la vida, por ser fuente de inspiración y amor, por depositar toda su confianza en mí y siempre creer que yo sí puedo, por brindarme su apoyo incondicional, por ser mi ejemplo y mi guía.

A mi abuela mima que es mi segunda mamá y mi viejita linda. Te quiero mucho.

A mi hermana Heidy y a mi sobrino Rodney mi relevo. Los quiero tata y rodny.

"La informática se convertirá en una poderosísima fuerza científica, económica y política para Cuba."

Fidel Castro Ruz.

Resumen

El Juez en Línea Caribeño (COJ) es un sistema desarrollado en la Universidad de las Ciencias Informáticas (UCI) con el objetivo de probar, mejorar y compartir habilidades en la resolución de problemas de diversos lenguajes de programación y motivar al trabajo en equipo. Actualmente el sistema permite exportar reportes pero esta funcionalidad no satisface las necesidades de los usuarios y del sistema, esto se debe principalmente a la ausencia de un mecanismo que permita a los usuarios obtener, de forma rápida, la mayor cantidad de información en formato portable sobre los procesos que se realizan en el COJ. Por todo lo expuesto anteriormente el presente trabajo de diploma está orientado al desarrollo del módulo para la exportación de reportes del Juez en Línea Caribeño con el fin de mejorar la calidad que requiere este proceso y que sus administradores necesitan, pues solo se enmarca en la exportación de reportes del archivo de problemas del módulo 24 Horas. Con el propósito de lograr lo planteado, se hizo un estudio del estado del arte para la reutilización de funcionalidades en algunos jueces en línea de los diferentes ámbitos. Para la implementación de la solución se empleó la metodología de desarrollo XP, Java como lenguaje de parte del servidor y Javascript del lado del cliente, utilizando como apoyo los marcos de trabajo jsPDF, JQuery y Spring. Para el modelado se utilizó Visual Paradigm 8.0, como IDE NetBeans 8.0 y como servidor de aplicaciones web Apache Tomcat 8.0.3.

Palabras claves: exportar, información, Juez en Línea Caribeño, reportes.

Índice

Introducción 1
Capítulo 1: Fundamentación teórica 5
1.1 Introducción 5
1.2 Conceptos fundamentales 5
1.3 Jueces en línea 6
1.3.1 Selección de los sistemas similares útiles para el desarrollo de la solución 9
1.4 Tecnologías para el desarrollo de la solución 9
1.5 Herramientas para el desarrollo de la solución 11
1.5.1 Herramientas Case de Modelado de Software 11
1.5.2 Entorno Integrado de Desarrollo (IDE) 12
1.5.3 Control de Versiones 13
1.5.4 Servidor de Aplicaciones Web 15
1.5.5 Sistemas Gestor de Base de Datos (SGBD) 16
1.5.6 Marcos de trabajo 17
1.6 Metodología de desarrollo de software 19
1.7 Conclusiones del Capítulo 21
Capítulo 2: Propuesta de Solución 23
2.1 Introducción 23
2.2 Propuesta de solución 23
2.3 Personas relacionadas con el sistema 23
2.4 Levantamiento de requisitos 24
2.4.1 Lista de reserva del producto 24
2.4.2 Aspectos no funcionales del sistema 25
2.5 Exploración 26
2.5.1 Historias de usuario 27
2.6 Estimación y Planificación 30
2.6.1 Estimación de esfuerzos por HU 30
2.6.2 Planificación de iteraciones 32
2.6.3 Plan de duración de iteraciones 33

2.6.4 Plan de entregas 35
2.7 Diseño 36
2.7.1 Patrones de diseño 36
2.7.2 Tarjetas CRC (Clase-Responsabilidad-Colaboración) 38
2.8 Modelo de Datos 39
2.9 Conclusiones del capítulo. 40
Capítulo 3: Implementación y Prueba 42
Introducción 42
3.1 Propuesta de arquitectura 42
3.1.1 Estilo arquitectónico 44
3.2 Fase de implementación 47
3.2.1 Tareas de ingeniería 47
3.3 Pruebas 53
3.3.1 Pruebas unitarias 53
3.3.2 Pruebas de integración 55
3.3.3 Pruebas de carga 56
3.3.4 Pruebas de aceptación 58
Conclusiones del capítulo 60
Conclusiones Generales 61
Recomendaciones 62
Referencias bibliográficas 63
Anexos 67
Anexo I 67
Anexo II 75
Anexo III 76
Tareas de Ingeniería 76
Primera Iteración 76
Segunda Iteración 81
Tercera Iteración 88
Anexo IV 97

Prueba unitaria con JUnit..... 97
Prueba unitaria con QUnit..... 100
Pruebas de carga 104
Pruebas de aceptación 106

Introducción

La utilización de las Tecnologías de la Informática y las Comunicaciones (TIC) representan un importante logro en el enriquecimiento de espacios donde las personas puedan intercambiar experiencias y conocimientos. Disminuyendo paulatinamente las distancias de la brecha digital y aumentando la accesibilidad para la comunidad de profesionales y estudiantes.

El Concurso Internacional Universitario ACM de Programación (en inglés ACM International Collegiate Programming Contest (ACM-ICPC[1])), es una competencia de programación de algoritmos de diferentes lenguajes. En sus inicios no contaba con muchos participantes pues solo competían principalmente equipos de Estados Unidos y Canadá, lo que posibilitaba la gestión de la información de manera manual. Con el paso del tiempo aumentó el número de participantes y de países convirtiéndose en una competición mundial con más de 5606 equipos de 2534 universidades de 101 países en los 6 continentes. El número de equipos se incrementa en un 20% cada año, lo que complejizó el proceso de evaluación, provocando el surgimiento de los evaluadores automáticos que evolucionaron con el desarrollo de la Web hacia los jueces en línea que constituyen ejemplos de este tipo de espacios de intercambio (1).

Un juez en línea es un sistema que permite juzgar programas de computación que intenten solucionar tareas propuestas. Compilan y ejecutan código fuente, y los ponen a prueba con los datos definidos para la tarea seleccionada. La evaluación se realizará comparando la salida del programa enviado por el usuario contra la salida que se tiene de la tarea en cuestión (2).

La Universidad de las Ciencias Informáticas (UCI) no está exenta de este tipo de sistemas, en el 2006 comenzó el desarrollo del Juez en Línea Caribeño (COJ)[2] que es un juez en línea con el propósito de brindar mayor autonomía al Movimiento ACM-ICPC en la región del Caribe, facilitando el intercambio de conocimientos y la auto-preparación.

Entre las principales funcionalidades de estos sistemas informáticos, se encuentra la de proveer la información que se genera durante los distintos procesos y eventos. Para lograr un control eficiente de estas tareas, se hace necesario buscar mecanismos que faciliten exportar la información. Los mismos deben ser capaces de consolidar la información adquirida y mostrarla en formatos entendibles para el personal que espera recibir la información.

[1] Competición anual de programación y algorítmica entre universidades de todo el mundo. En la competición prima el trabajo en equipo, el análisis de problemas y el desarrollo rápido de software.

[2] coj.uci.cu

El Juez en Línea Caribeño para crear y exportar reportes utiliza iText[3], pero su implementación en el sitio no es del todo ideal dado que afecta el rendimiento del sistema, no realiza la exportación de reportes con la calidad que requiere este proceso y que sus administradores necesitan, pues se enmarca solamente en la exportación de reportes del archivo de problemas del módulo 24 Horas. Con esta acción el servidor debe crear un reporte por cada una de las solicitudes que reciba del cliente; este sitio cuenta con más de 21000 usuarios, si se supone que se realiza un concurso participando al menos 5000 usuarios, se tienen unas 5000 solicitudes, que son 5000 reportes en formato PDF[4] de aproximadamente 50kbs cada uno que tiene que crear el servidor, esto significaría disminuir la capacidad de respuesta del servidor y si es un sistema como el COJ que se caracteriza por su crítico consumo de hardware al compilar y ejecutar las soluciones de sus problemas, mayor es la probabilidad de que el sitio no brinde este servicio de forma eficiente.

A partir de la situación problemática antes expresada se define como **problema a resolver**: ¿Cómo facilitar la exportación de reportes en el Juez en Línea Caribeño?

Quedando definido como **objeto de estudio:** El proceso de exportación de reportes.

El **campo de acción** estará enfocado en la exportación de reportes en Jueces en Línea de Programación.

Se plantea como **Objetivo General:** Desarrollar un módulo que facilite la exportación de reportes en el Juez en Línea Caribeño.

Para darle solución al objetivo general, se desglosan los siguientes **Objetivos Específicos:**

- Realizar un estudio del estado del arte de los jueces en línea que utilizan la exportación de reportes de problemas, estadísticas y concursos a nivel mundial, nacional y local.
- Desarrollar un módulo para la exportación de reportes del Juez en Línea Caribeño.
- Probar la solución implementada del módulo para la exportación de reportes del Juez en Línea Caribeño.

Para dar cumplimiento al objetivo planteado se definen las siguientes **tareas de investigación:**

- Definición de los principales conceptos asociados a la exportación de reportes.

[3] iText es una biblioteca Open Source para crear y manipular archivos PDF, RTF, y HTML en Java.
[4]Documentos de formatos portable.

- Caracterización de los principales jueces en línea existentes en la Universidad de las Ciencias Informáticas, en Cuba y en el resto del mundo que utilicen la exportación de información.
- Definición de las principales tecnologías, herramientas y metodologías de desarrollo a utilizar para la creación del módulo para la exportación de reportes del Juez en Línea Caribeño.
- Desarrollo del módulo para la exportación de reportes del Juez en Línea Caribeño realizándose el diseño y la implementación.
- Validación de la solución obtenida.

El desarrollo de la solución se sustenta sobre la **Hipótesis:** Si se desarrolla un módulo para la exportación de reportes, entonces se facilitará la obtención de la información que se genera en el Juez en Línea Caribeño.

Para cumplir las tareas planteadas se utilizarán los siguientes **métodos científicos**:

Métodos teóricos:

- Analítico sintético: Para el análisis del Juez en Línea Caribeño con la información recopilada en los medios bibliográficos que ayude a desarrollar el diseño del sistema y emplear estos conocimientos para una mejor preparación en el tema.
- La modelación: Para mostrar la estructura, relaciones internas y características a través de diagramas, mediante el uso del lenguaje de modelado UML[5].
- Método sistémico: Para estudiar los procesos de exportación de reportes en su dinámica y la actividad que manifiesta estos procesos en su movimiento, en sus relaciones con el medio y sobre la base de su estructura interna.

Métodos empíricos:

- Revisión documental: Se pone en práctica al inspeccionar la bibliografía existente y estudiar la información en Internet para realizar las tareas de investigación.

[5] Lenguaje Unificado de Modelado

- Observación: Para poder conocer la realidad mediante la percepción directa de los procesos de gestión y exportación de reportes.

La estructura capitular del presente trabajo de diploma es:

Capítulo 1: Fundamentación teórica: Se aborda el estudio sobre los principales jueces en línea existentes en la UCI, en Cuba y en el resto del mundo que utilicen la exportación de información, se describen también principales conceptos asociados a la exportación de reportes, se analizarán e identificarán principales tecnologías de desarrollo, herramientas y metodologías a utilizar en el desarrollo del módulo.
Capítulo 2: Propuesta de Solución: Se presentan las principales particularidades del sistema y se realiza una propuesta general. Se especifican los requisitos funcionales, no funcionales y se presentarán los artefactos resultantes del análisis y diseño.
Capítulo 3: Implementación y Prueba: Implementación de las funcionalidades definidas y se realizan las pruebas al módulo para que cumpla con los requerimientos establecidos.

Capítulo 1: Fundamentación teórica.

1.1 Introducción

En el desarrollo de este capítulo se aborda el estudio sobre los principales jueces en línea existentes en los diferentes ámbitos que utilicen la exportación de información. Se describen los principales conceptos asociados a la exportación de reportes, se analizan e identifican las principales tecnologías de desarrollo, herramientas y metodología a utilizar en el desarrollo del módulo.

1.2 Conceptos fundamentales

En este trabajo se utilizan algunos términos técnicos, los cuales se definen a continuación para una mejor comprensión de los mismos.

Reporte

Un reporte es un informe o una noticia. Este tipo de documento que puede ser impreso, digital o audiovisual pretende transmitir una información. En el ámbito de la informática los reportes son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios (3).

Gracias a los reportes se puede proceder a realizar un resumen de datos o clasificar grupos determinados. Por todo ello, se entiende que estos documentos sean tan importantes en cualquier empresa pues gracias a ellos cuenta con sus propias bases de datos (3).

Módulo

En programación un módulo es una porción de un programa de computadora. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas. En el caso de la programación, los módulos suelen estar organizados jerárquicamente en niveles, de forma que hay un módulo principal que realiza las llamadas oportunas a los módulos de nivel inferior. Cuando un módulo es convocado, recibe como entrada los datos proporcionados por otro del mismo o superior nivel, el que ha hecho la llamada; luego realiza su tarea. A su vez este módulo convocado puede llamar a otro u otros módulos de nivel inferior si fuera necesario; cuando ellos finalizan sus tareas, devuelven la salida pertinente al módulo inmediato llamador, en secuencia reversa, finalmente se continúa con la ejecución del módulo principal (4).

Características de un módulo

Cada uno de los módulos de un programa idealmente debería cumplir las siguientes características (4):

- Tamaño relativamente pequeño: Esto facilita aislar el impacto que pueda tener la realización de un cambio en el programa, para corregir un error, o por rediseño del algoritmo correspondiente.

- Independencia modular: Cuanto más independientes son los módulos entre sí más fácil y flexiblemente se trabajará con ellos, esto implica que para desarrollar un módulo no es necesario conocer detalles internos de otros módulos.

Todo módulo, por lo tanto, forma parte de un sistema y suelen estar conectado de alguna manera con el resto de los componentes.

1.3 Jueces en línea

Un juez en línea es un sistema, por lo general web, que permite juzgar programas de computación que intenten solucionar tareas propuestas. Estos sistemas pueden compilar y ejecutar códigos fuente, y ponerlos a prueba con los juegos de datos definidos para la tarea seleccionada. Las posibles soluciones se ejecutan con restricciones tales como: límite de tiempo de ejecución, límite de memoria, tamaño de código fuente, restricciones de seguridad, entre otras. La salida de cada programa enviado por el usuario es capturada por el sistema y comparada contra la salida que se tiene de la tarea en cuestión o será evaluada por un evaluador externo (2). Existen muchos jueces en línea activos destacándose los siguientes:

Juez en Línea de Pekín

El Juez en Línea de Pekín (POJ) es uno de los jueces en línea de mayor demanda entre la comunidad universitaria china, puesto que es uno de los más antiguos de esa región. Posee características similares a varios jueces en línea de otras universidades dado que posee una interfaz intuitiva, ligera y fácil de usar, además de ser uno de los jueces en línea con menores tiempos de respuesta en su interfaz y poseer una abundante sección de preguntas (5).

Tianjin University Online Judge (TOJ)

Este surge a partir de la posibilidad brindada por el POJ de descargar una versión disminuida y restrictiva de la aplicación. Posee las mismas funcionalidades y características del POJ, solo que incluye la funcionalidad de crear Concursos Virtuales Públicos. La diferencia más significativa del TOJ, respecto al POJ, radica en la no dependencia de plataforma Windows (6).

Sphere Online Judge

El Juez en línea Esfera (SPOJ) es un juez de línea con más de 200.000 usuarios registrados y más de 20.000 problemas. La solución a los problemas puede presentarse en más de 40 lenguajes. SPOJ permite a los usuarios avanzados para organizar concursos bajo sus propias reglas y problemas. También incluye un foro donde los programadores pueden discutir cómo se puede resolver un problema particular. Este sistema fue creado originalmente para aplicar un juez de línea en la enseñanza de los estudiantes. Básicamente, se centró en los estudiantes y profesores de las universidades y de los miembros de una comunidad más amplia de programación, interesados en algoritmos[6] y concursos de programación (7).

TopCoder

Realmente es una plataforma de concursos de programación en general. La colección de problemas es perfectamente accesible y su principal objetivo es proveer una plataforma donde las diferentes empresas puedan reclutar personal talentoso, basados en el desempeño en competencias. TopCoder funciona además como una fuente de creatividad para las empresas que contratan los servicios de la plataforma (8).

Codeforces

Surge en el año 2010 como iniciativa de Mike Mirzanayov, cambiando la dinámica de los jueces en línea existentes hasta el momento. Codeforces se caracteriza por ser uno de los pocos jueces en línea que utiliza la Web 2.0 y constituye un sistema predominantemente para la realización de competencias de programación en varios estilos existentes. No permite la exportación reportes de sus problemas (9).

ACM-ICPC Live Archive

El Archivo del ACM-ICPC es el Juez en Línea con mayor número de problemas de competencias oficiales (Concursos Regionales, Finales Mundiales, entre otros). El archivo se caracteriza por tener una interfaz amigable, muy similar a la interfaz de Juez en Línea de la UVa[7]. Su funcionalidad más relevante es la posibilidad brindada a numerosos usuarios de organizar sus propias competencias sobre la plataforma y aportar problemas que aún no estén. Se considera que su principal deficiencia radica en el corto tiempo de vida de los envío de los usuarios al sistema (1). En sus diferentes archivos se puede presentar problemas, se representan las estadísticas y también se puede exportar la información de los problemas existentes para mejor estudio.

[6] Conjunto reescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien lo ejecute.
[7] Universidad de Valladolid.

UVa Online Judge

Juez Online de la UVa es un juez automatizado en línea creado en 1995 por Miguel Ángel Revilla en la Universidad de Valladolid en España. Ciriaco García de Celis, un estudiante de informática de la Universidad de Valladolid, escribió todo el código para la primera versión del sistema. El juez fue abierto al público sólo en abril de 1997 y en noviembre de 1999 y 2000 fue sede de la UVa ACM ICPC SWERC. Sus archivos de problemas contienen más de 4.300 problemas y el registro de usuarios está abierto a todo el mundo. Actualmente hay más de 100.000 usuarios registrados. Un usuario puede presentar una solución en ANSI[8], C, C ++, Java y Pascal[9]. Al igual que ACM-ICPC Live Archive este sitio comparado con los anteriores mencionados en cuanto a la exportación de la información muestra mejor rendimiento y popularidad brindando a los usuarios objetos de estudio para su preparación (10).

Juez en Línea Caribeño (COJ)

Es un juez en línea para entrenar la programación de algoritmos con diferentes lenguajes. Su desarrollo comenzó en 2006 por programadores de la UCI y otros programadores de la comunidad caribeña de la ACM-ICPC.

El sitio cuenta con más de 2000 problemas de distintas fuentes y dificultades. Esta lista de problemas va creciendo paulatinamente a medida que los usuarios de la comunidad van añadiendo. El sitio permite el envío de soluciones programadas con diversos lenguajes como Pascal, Python[10], C/C++, Java, C#, Perl, Prolog, Ruby y Text. Mantiene varias posiciones como usuarios, instituciones y países. La puntuación que aporta un problema es de acuerdo a su dificultad, la cual es medida por la cantidad de personas que lo han resuelto, lo que implica que sea dinámica, cambia con el tiempo. Periódicamente se organizan concursos en tiempo real, la mayoría de los cuales son abiertos.

El sistema automáticamente compila y ejecuta el código enviado por el usuario (solución a un problema). El código fuente enviado será probado con algunas restricciones, incluyendo el tiempo de ejecución, el uso de memoria, el tamaño del código fuente, la seguridad y otras. La salida del código será capturada por el sistema y comparada con la salida correcta proporcionada por el autor del problema. El usuario obtendrá Aceptado (como sentencia o veredicto) si todas las pruebas aplicadas a su código fuente parecen estar bien. En cualquier otro caso, obtendrá una respuesta de rechazo (11).

[8]Es considerado uno de los lenguajes más usado en el desarrollo de sistemas operativos.
[9]Lenguaje de programación pensado para ser usado con fines científicos.
[10] Es un lenguaje de programación que se desarrolla como un proyecto de código abierto.

1.3.1 Selección de los sistemas similares útiles para el desarrollo de la solución

Después de definir las principales características de los jueces en línea antes mencionados se desarrolló un análisis sobre el manejo de la información y la forma que la exportan. Los jueces ACM-ICPC Live Archive, UVa Online Judge y Sphere Online Judge son los únicos que en sus archivos de problemas incluyen la descarga de reportes. Luego de investigar las principales tecnologías para dicho proceso y de establecer comunicación con los administradores de estos tres sistemas mediante correo electrónico, solo se pudo conocer que el UVa Online Judge utilizó HTML to PDF que es una biblioteca para la generación de PDF y para la conversión de documentos HTML en archivos PDF y ahora utiliza LaTeX (pdfTeX) extensión de TEXT que es un sistema de tipografía muy popular en el entorno académico, especialmente entre las comunidades de matemáticos, físicos e informáticos. Estas tecnologías son similares a la que utiliza el COJ dado que funcionan de parte del servidor y como desventajas se puede decir que es muy difícil su aprendizaje, por lo que no son factible como puntos de apoyo para el desarrollo del módulo para la exportación del COJ. Para resolver los problemas existentes, se planteó la necesidad de construir un módulo propio que cumpla con las necesidades planteadas por el cliente para realizar el proceso de exportación de reportes.

1.4 Tecnologías para el desarrollo de la solución

El Juez en Línea Caribeño fue desarrollado utilizando las siguientes tecnologías: CSS3, Javascript, HTML5, java y Lenguaje Unificado de Modelado. Es por ello que el nuevo módulo a desarrollar debe adaptarse e incorporar las mismas tecnologías, sin entorpecer el trabajo ya desarrollado.

Javascript

Javascript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos con Javascript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, Javascript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente es una marca registrada de la empresa Sun Microsystems[11] (12). Se utilizó Javascript en el desarrollo de la solución para realizar las funcionalidades encargadas de exportar los reportes, permitiendo mayor eficiencia y rendimiento del servidor, por ser un lenguaje del lado del cliente.

CSS

[11] Es una empresa informática adquirida por Oracle Corporation y creadora de la popular suite ofimática OpenOffice.org y el lenguaje de programación Java.

Hoja de estilo en cascada o CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo y abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y otros temas. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a ella en las que aparezca ese elemento (13). Con la utilización de CSS quedaron estructurado los reportes en cuanto a estilo y diseño para su exportación, permitiendo obtener un documento más profesional y de acorde con el prestigio del Juez en Línea Caribeño.

HTML

El lenguaje de marcas de hipertexto[12] o HTML, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto e imágenes. Este lenguaje no describe la apariencia de un documento sino que ofrece a cada plataforma la información para que le dé formato según su capacidad y la de su navegador. Por eso es importante, diseñar los documentos con un contenido claro y bien estructurado que resulte fácil de leer y entender en cualquier navegador (14). Se utilizó esta tecnología para obtener los reportes ya realizados en el sistema que incluyen tablas, gráficos, imágenes y texto.

Java

Java es un lenguaje de programación orientado a objeto de alto nivel desarrollado por Sun Microsystem a principios de la década del 90. Su sintaxis es muy parecida a la de C y C++ considerando que su desarrollo fue inspirado en los mismos, siempre buscando eliminar errores que suelen inducirse en lo que respecta a la manipulación de memoria y punteros. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra (15). Con Java se podrá controlar los reportes pertinentes del sitio y permitirá acceder a la base de datos y obtener la información pertinente de los problemas para la creación de reportes.

Lenguaje Unificado de Modelado (UML)

[12]Es el texto que en la pantalla de un dispositivo electrónico conduce a otro texto relacionado.

Es el lenguaje de modelado de sistemas de software más conocidos y utilizados en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados (16). Se utilizó para elaborar el diagrama de Modelo de Datos.

1.5 Herramientas para el desarrollo de la solución

Se define como herramienta para el desarrollo de la solución a un programa informático que usa un programador para crear, depurar, gestionar o mantener un programa.

1.5.1 Herramientas Case de Modelado de Software

La herramienta CASE (Computer Aided Software Engineering)[13] es un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Este puede ser generalmente aplicado a cualquier sistema o colección de herramientas que ayudan a automatizar el proceso de diseño y desarrollo de software (17).

Objetivos de las herramientas CASE:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.

La herramienta **CASE** que se utilizará para el desarrollo de la solución será **Visual Paradigm** por ser considerada muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelente facilidades de interoperabilidad con otras aplicaciones. Permite invertir código fuente de programas,

[13]Ingeniería de Software Asistida por Computadoras

archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Su elección se debe también por ser un software libre que incorpora el soporte para trabajo en equipo, que permite a varios desarrolladores trabajar a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

1.5.2 Entorno Integrado de Desarrollo (IDE)

Un **IDE** es un entorno de programación que ha sido empaquetado como un programa de aplicación; consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (18).

NetBeans

Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Existe un número importante de módulos para extender el IDE NetBeans siendo un producto libre y gratuito sin restricciones de uso.

Soporta el desarrollo de aplicaciones Java como J2SE, web, EJB y aplicaciones móviles. Entre sus características se encuentra un sistema de proyectos basado en Ant[14], control de versiones y refactoring[15] (19).

Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Geany

Es un IDE que hasta hace poco sólo estaba disponible para sistemas Linux[16], Mac OS X[17]y BSD[18], pero ya está disponible para Windows. Este entorno es muy sencillo, pero proporciona las funcionalidades

[14]Ant es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas.
[15] Técnica de la ingeniería de software para reestructurar un código fuente.
[16] Sistema operativo libre homólogo a Unix que usualmente utiliza herramientas de Sistema GNU.
[17]Sistema operativo de Apple.
[18]Distribución de Software Berkeley, sistema operativo derivado del sistema Unix.

necesarias para desarrollar aplicaciones sin problemas. Este IDE permite programar en diferentes lenguajes como: C, C++, Java, Python, Pascal, SQL[19] o HTML (20).

Eclipse

Entorno de desarrollo integrado de código abierto multiplataforma para desarrollar proyectos. Esta plataforma ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). También se puede usar para otros tipos de aplicaciones cliente, como BitTorrent[20] o Azureus[21]. En Eclipse se pueden usar diferentes lenguajes de programación como: Java, ANCI C, C++, JSP, sh, perl, php y sed (21).

Análisis de la selección del IDE

Para la implementación del trabajo se utilizará el IDE NetBeans 8.0 pues permite la reutilización de módulos, su instalación y actualización es simple, es gratis y de código libre y es un editor de código rápido e inteligente. Su elección se debe además por la experiencia anterior en el uso de esto herramienta durante estos cinco años y que se integra de forma natural con el servidor web Apache Tomcat mientras que otros de los IDE vistos se deben realizar varias configuraciones para posibilitar esta integración.

1.5.3 Control de Versiones

Software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, como código fuente, documentación o ficheros de configuración (22). Debe proporcionar:

- Mecanismo de almacenamiento de los elementos que deba gestionar.
- Posibilidad de realizar cambios sobre los elementos almacenados.
- Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos.

Sistemas de Control de Versiones Libres

CVS

[19] Lenguaje de consulta estructurado
[20] Programa de intercambio de ficheros.
[21] Cliente BitTorrent libre y multiplataforma que dispone de una interfaz agradable y estética.

CVS ha estado durante mucho tiempo, y muchos desarrolladores están ya familiarizados con él. En su día fue revolucionario: fue el primer sistema de control de versiones de código abierto con acceso a redes de área amplia para desarrolladores y el primero que ofreció anónimos de sólo lectura, los que dieron a los desarrolladores una manera fácil de implicarse en los proyectos. CVS sólo versiona ficheros, no directorios; ofrece ramificaciones, etiquetado, y un buen rendimiento en la parte del cliente, pero no maneja muy bien ficheros grandes ni ficheros binarios. Tampoco soporta cambios atómicos (23).

Subversion

Subversion fue escrito ante todo para reemplazar a CVS, es decir, para acceder al control de versiones aproximadamente de la misma manera que CVS lo hace, pero sin los problemas o falta de utilidades que más frecuentemente molestan a los usuarios de CVS. Uno de los objetivos de Subversion es encontrar una transición relativamente suave para las personas que no están acostumbrados a CVS (24).

Mercurial

Mercuriales un sistemas de control de versiones distribuido que ofrece una completa indexación cruzada de ficheros y conjuntos de cambios; unos protocolos de sincronización SSH[22] y HTTP[23] eficientes respecto al uso de CPU y ancho de banda; una fusión arbitraria entre ramas de desarrolladores; una interfaz web autónoma integrada; portabilidad a UNIX, MacOS X, y Windows (25).

Git

Git es un proyecto empezado por Linus Torvalds[24]para manejar el árbol fuente del "kernel"[25]de Linux. Al principio GIT se enfocó bastante en las necesidades del desarrollo del "kernel", pero se ha expandido más allá que eso y ahora es usado por otros proyectos aparte del "kernel" de Linux. GIT cae en la categoría de herramientas de administración de código abierto distribuido (26).

Análisis de la selección del Sistema de Control de Versiones

Se decide utilizar Git por las facilidades de infraestructura que brinda la UCI para trabajar a través del sitio codecomunidades utilizando GitLab que es un software de código abierto para colaborar en código que permite administrar repositorios Git con controles de acceso de grano fino que mantienen su código seguro, realizar revisiones de código y mejorar la colaboración con las solicitudes de combinación; en cada proyecto

[22]Protocolo de transferencia de hipertexto es el protocolo usado en cada transacción de la World Wide Web.
[23] Protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red.
[24] Ingeniero de software finlandés; es conocido por iniciar y mantener el desarrollo del "kernel".
[25] Núcleo, es un software que constituye una parte fundamental del sistema operativo.

también puede tener un seguimiento de incidencias y una wiki; utilizado por más de 100.000 organizaciones, GitLab es la solución más popular para gestionar repositorios Git en las instalaciones (27).

1.5.4 Servidor de Aplicaciones Web

Los servidores web son aquellos cuya tarea es alojar sitios o aplicaciones, las cuales son accedidas por los clientes en un navegador que se comunica con el servidor utilizando el protocolo HTTP. Sirven datos en forma de páginas web, hipertextos o páginas HTML ya sean textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

GlassFish

Es servidor de aplicaciones web que fue iniciado por Sun Microsystems para la plataforma Java EE [26]y por un tiempo fue patrocinado por Oracle Corporation. La versión comercial es denominada Oracle GlassFish Enterprise Server. Permite a los desarrolladores crear aplicaciones empresariales portables y escalables, además de su integración con las más anticuadas tecnologías de este campo. GlassFish permite que componentes opcionales puedan ser instalados para servicios adicionales (28).

IIS

Internet Information Services/Server [27]permite realizar copias de seguridad y restaurar la meta base, posee mensajes de error personalizados mejorados y posee herramientas de administración fundamentadas en web que admiten la administración remota del servidor desde casi cualquier explorador en cualquier plataforma. Aunque son servicios para los ordenadores que funcionan con Windows, representando esta su mayor desventaja (29).

Servidor HTTP Cherokee

Posee características y funcionalidades centradas en la velocidad, flexibilidad y capacidad de ser empotrado. La alta eficiencia y una arquitectura lo suficientemente flexible como para poder escalar grandes servidores son características de Cherokee que pueden suponer un paso adelante respecto a los servidores web libres existentes (30).

Servidor Tomcat

Conocido como Jakarta Tomcat o Apache Tomcat) es una implementación de software de Código abierto de Java Servlet y tecnologías JavaServerPages (JSP)[28].Tomcat puede funcionar como servidor Web por sí

[26] Plataforma de programación.
[27] Servicios de información de Internet / servidor.
[28] Tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo.

mismo. Al principio de su desarrollo existió la percepción de que la utilización de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con mínimos requisitos de velocidad y gestión de transacciones. Actualmente ya no existe esa percepción y Tomcat es usado como servidor Web independiente en entornos con alto nivel de tráfico y alta disponibilidad (31).

Análisis de la selección del Servidor de Aplicaciones Web

Como servidor de aplicaciones web se decidió trabajar con Tomcat 8.0, debido a las facilidades que proporciona esta herramienta y por su utilización para numerosos proyectos universitarios. Otra de las razones por las cuales se optó su utilización, la más importante a mencionar, es la compatibilidad que existe entre las diversas plataformas existentes, de esta manera se podrá realizar la implantación ya sea en un sistema operativo Windows o Unix y a su integración con el IDE NetBeans 8.0.

1.5.5 Sistemas Gestor de Base de Datos (SGBD)

Un Sistema Gestor de Base de Datos es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos.

Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos (32).

MySQL

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes (33).

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a

los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras (33).

PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de Postgres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multi-versión que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación (34).

Posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos y los disparadores. Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales.

El código fuente se encuentra disponible para todos sin costo alguno. Es totalmente compatible con ACID (acrónimo de Atomicidad, Consistencia, Aislamiento y Durabilidad) y tiene soporte completo para claves foráneas, combinaciones, vistas, disparadores y procedimientos almacenados (34).

Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico (34).

Análisis de la selección del Sistemas Gestor de Base de Datos

Se selecciona PostgreSQL por brindar ahorros considerables de costos de operación, estabilidad y confiabilidad, es multiplataforma, diseñado para ambientes de alto volumen, posee gran capacidad de almacenamiento y por ser la tecnología que se usa actualmente por el COJ.

1.5.6 Marcos de trabajo

Un marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Un marco tiene un comportamiento predeterminado que por defecto debe ser un comportamiento útil, definido e identificable (35).

En la programación Javascript avanzada, especialmente el complejo tratamiento de las diferencias del navegador, a menudo puede ser muy difícil y requiere mucho tiempo para trabajar. Para hacer frente a estas

dificultades, se han desarrollado bibliotecas Javascript. Estas bibliotecas son llamadas marcos de trabajo de Javascript. Todos estos marcos tienen funciones para tareas comunes de Javascript como animaciones, manipulación DOM y manejo Ajax[29] (35).

JQuery

JQuery es un software libre y de código abierto permitiendo su uso en proyectos libres y privativos. Consiste en un único fichero Javascript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. Es el marco de Javascript más popular en Internet, a causa de que utiliza selectores CSS para acceder y manipular los elementos HTML (objetos DOM) en una página web. Muchas de las mayores empresas de la Web usan JQuery como Google, Microsoft, IBM y Netflix[30] (36).

JasperReports

La biblioteca JasperReports es el motor de informes de código abierto más popular del mundo. Está escrito completamente en Java y es capaz de utilizar los datos procedentes de cualquier tipo de fuente de datos y producir documentos de píxeles perfecto que se pueden ver, imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, Excel, Open Office y Word (37).

Google Web Toolkit (GWT)

Es un marco creado por Google que permite ocultar la complejidad de varios aspectos de la tecnología AJAX y es compatible con varios navegadores. Con esta biblioteca, los desarrolladores pueden crear y depurar aplicaciones AJAX en lenguaje JAVA usando el entorno de desarrollo que prefieran. Cuando una aplicación es desplegada, el compilador GWT traduce la aplicación Java a un archivo Javascript, que puede ser usado para optimizar el rendimiento (38).

Vaadin

Vaadin es un marco de trabajo para el desarrollo de aplicaciones web de código abierto modernas, que posee una librería con numerosos componentes de usuario con los que construir aplicaciones web profesionales similares a los interfaces de una aplicación de escritorio. Se ejecuta del lado del servidor lo que significa que la mayoría de la lógica y por tanto la mayor carga del trabajo recae en el servidor (39).

Spring

El marco Spring proporciona un modelo de programación y configuración completa para las aplicaciones empresariales modernas basadas en Java en cualquier tipo de plataforma de despliegue. Posee inyección

[29] Acrónimo de Javascript asíncrono y XML, es una técnica de desarrollo Web para crear aplicaciones interactivas.
[30] Es una empresa comercial estadounidense de entretenimiento.

de dependencia y apoyo fundamental para JDBC[31], JPA y JMS[32]. Un elemento clave de Spring es el apoyo de infraestructura a nivel de aplicación para que los equipos puedan centrarse en la lógica de negocios a nivel de aplicación, sin ataduras innecesarias a los entornos de despliegue específicos (40).

jsPDF

Es una biblioteca Javascript de código abierto para la creación de reportes en documentos PDF sin utilizar en ningún momento un lenguaje del lado del servidor, realizando todo el trabajo del lado del cliente. El punto a destacar de este proyecto es como utiliza de forma creativa las Data URIs[33]para pasar la información a formato PDF, una solución sencilla y eficiente a un problema que no podía ser resuelto de manera tan sencilla. Se puede utilizar en una extensión de Firefox, en el lado del servidor y Javascript con datos URI en algunos navegadores (41).

Análisis de la selección de los marcos de trabajo

Como marcos de trabajo para apoyarse en el desarrollo del presente trabajo se utilizarán Spring y JQuery para las funcionalidades que requieran su utilización dado que el COJ está implementado con estos marcos y para el proceso de exportar la información se trabajara con jsPDF; debido a los grandes problemas que solucionaría en el sistema en cuanto a rendimiento y que permitirá crear reportes del lado del cliente con soporte de imágenes, texto y diferentes fuentes pues no necesita ningún script del lado del servidor para realizar esta tarea. Además, su elección se debe a ser un software libre y siguiendo las ideas de Cuba y de la UCI de promover el desarrollo con herramientas libres es apropiada su utilización.

1.6 Metodología de desarrollo de software

Las metodologías son el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software e indican cómo hay que obtener los distintos productos parciales y finales. Se clasifican en dos tipos metodologías pesadas y metodologías ágiles.

Para dar una idea de qué metodología se puede utilizar y cuál se adapta más al medio, se realizó un estudio de las que se consideran más importantes como Rational Unified Process[34] (RUP), Extreme Programming[35] (XP) y Scrum.

[31] Java Database Connectivity, para la integración de bases de datos.
[32] Tecnologías destacadas de la Plataforma empresarial de java.
[33] Esquema para insertar pequeños trozos de contenido en los documentos
[34] Proceso Relacional Unificado.
[35] Programación Extrema.

RUP

Es una metodología de desarrollo de software que está basada en componentes e interfaces bien definidas, y junto con el Lenguaje Unificado de Modelado, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos (42).

Es un proceso que puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo (42).

Características principales de RUP

- Unifica los mejores elementos de metodologías anteriores.
- Preparado para desarrollar grandes y complejos proyectos.
- Orientado a Objetos.
- Utiliza el UML como lenguaje de representación visual.

XP

Se puede definir la programación extrema 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 (43).

Es un enfoque de la Ingeniería de Software formulado por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change 1999. Es el más destacado de los procesos ágiles de desarrollo de software (43).

Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que

los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Con sus teorías ha conseguido el respaldo de gran parte de la industria del software y el rechazo de otra parte. La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica (43).

Metodología Scrum

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

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos de entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (44).

Análisis de la selección de la Metodología de Desarrollo de software

Luego de realizar un análisis sobre las metodologías de desarrollo de software anteriores, se decide utilizar XP porque toma en cuenta los cambios que puedan ir ocurriendo a medida que se vaya desarrollando el proyecto para de esta manera mejorar u optimizar la arquitectura del mismo a medida que este se va llevando a cabo. Trabaja de forma iterativa, de manera que en cada iteración se vaya mejorando la arquitectura del proyecto y se acerque más al resultado esperado. Estos procesos ágiles requieren de una persistente comunicación y retroalimentación entre los integrantes del proyecto para que de esta manera se realicen los cambios y se mejore el producto final, proporcionando programación organizada, menor taza de errores y satisfacción del cliente.

1.7 Conclusiones del Capítulo

Después de analizar las deficiencias existentes en el COJ para la exportación de reportes se evidencia la necesidad de implementar un módulo que permita obtener los reportes del sistema. Para ello se analizaron las características de los sistemas estudiados en los diferentes ámbitos donde se planteó la necesidad de construir un módulo propio que cumpla con las necesidades planteadas por el cliente para realizar el proceso de exportación de reportes. Por otra parte después de un estudio realizado de las principales tecnologías y herramientas para la realización del módulo de exportación de reportes para el COJ, se eligieron y se fundamentó la selección de la metodología XP para guiar el proceso de desarrollo, NetBeans como IDE

para el desarrollo, jsPDF como marco de trabajo, GIT como control de versiones, Visual Paradigm como Herramientas Case de Modelado de Software, Servidor Tomcat como Servidor de Aplicaciones Web y PostgreSQL como Sistema Gestor de Bases de Datos con el fin de crear un sistema con la calidad requerida.

Capítulo 2: Propuesta de Solución

2.1 Introducción

Ya analizado el estado del arte, seleccionadas las tecnologías, herramientas y metodología; se puede comenzar a desarrollar la propuesta de solución para ello se realizarán las dos primeras fases de la metodología elegida: exploración y planificación.

2.2 Propuesta de solución

El COJ es un juez en línea para la motivación y la obtención de conocimiento el cual evalúa las posibles soluciones a problemas de programación expuestos online. En él tanto los administradores como los usuarios necesitan descargar reportes de estadísticas, concursos y problemas, para ello se propone la integración de un módulo de exportación de reportes para dicha actividad permitiendo mayor accesibilidad a la información.

2.3 Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo.

Tabla 1: Personal asociado a la aplicación

Persona relacionada con el sistema	Justificación
Usuario anónimo	Es la persona que navega por el sistema sin haberse creado una cuenta aún. Tiene la posibilidad de registrarse en el sitio para así poder acceder a todos los servicios que como usuario autenticado tiene.
Usuario autenticado	Es la persona que se encuentra autenticada en el sitio. Puede acceder a los servicios que se ponen a su disposición en el sitio como la resolución de problemas y de concursos en los cuales envían su

	propuesta de solución y el sistema evalúa esta respuesta.
Administrador	Es la persona facultada para la gestión del sistema. Es el encargado de administrar todo lo referente a módulos, temas y secciones de la aplicación

2.4 Levantamiento de requisitos

Es el proceso de identificar las necesidades del negocio, solucionando las posibles disparidades entre las personas involucradas en el mismo, con el propósito de definir y destilar los requerimientos para cumplir las restricciones impuestas por las distintas partes.

El proceso de levantamiento de requerimientos soporta el desarrollo de la especificación de los requerimientos, de tal forma que tengan los siguientes atributos:

- Deben ser completos, consistentes y han de estar dentro del alcance del proyecto.
- Deben tener un único identificador.
- Cumplen con los objetivos de los clientes.
- Son viables y apropiados para el desarrollo.
- Los requerimientos han de ser susceptibles a pruebas. (45)

2.4.1 Lista de reserva del producto

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. Esta lista de reserva está compuesta por requerimientos funcionales, que también pueden declarar explícitamente lo que el sistema no debe hacer (46). Los requisitos funcionales del módulo de gestión y exportación de reportes para el COJ son los siguientes:

Tabla 2: Requisitos funcionales

	Requisitos funcionales
RF1	Exportar reporte de cada problema del archivo de 24 horas.
RF2	Exportar reporte del listado de problemas presentes del archivo 24 horas.
RF3	Exportar reporte de las posiciones de los usuarios en el archivo 24 horas.
RF4	Exportar reporte de las posiciones de instituciones en el archivo 24 horas.
RF5	Exportar reporte de las posiciones de los países en el archivo 24 horas.
RF6	Exportar reporte de las estadísticas en el archivo 24 horas.
RF7	Exportar reporte de las posiciones de los concursantes en los concursos reales.
RF8	Exportar reporte de las estadísticas de los concursos reales.
RF9	Exportar reporte de las posiciones en los concursos reales anteriores.
RF10	Exportar reporte del perfil de usuarios.
RF11	Configurar reportes.
RF12	Crear y exportar reporte de todos los problemas presentados en un concurso ya realizado.
RF13	Exportar reporte de los premios de un concurso.
RF14	Exportar reporte de las estadísticas en los concursos reales anteriores.
RF15	Exportar reporte de la comparación de usuarios en el archivo 24 horas.
RF16	Exportar reporte de las sentencias de archivo 24 horas
RF17	Exportar reporte de las sentencias de concursos del archivo concursos reales.

2.4.2 Aspectos no funcionales del sistema

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema. Esto significa que a menudo son más críticos que los

requerimientos funcionales particulares. Los usuarios del sistema normalmente pueden encontrar formas de trabajar alrededor de una función del sistema que realmente no cumple sus necesidades. Sin embargo el incumplimiento de un requerimiento no funcional puede significar que el sistema entero sea inutilizable (46).

A continuación se muestra los requisitos no funcionales identificados para la solución propuesta:

Portabilidad: El módulo debe funcionar en los sistemas operativos (Windows y Linux).

Usabilidad: El módulo podrá ser usado por cualquier usuario que tenga conocimientos básicos de computación.

Seguridad: Debe garantizar que el acceso a la información se realice de acuerdo al rol que desempeñan los usuarios.

Requisitos de licencia: El producto debe ser liberado bajo la licencia GNU/GPL[36].

Software: Navegador Firefox 3+, Internet Explorer, Safari 3+, Google Chrome, Opera.

Hardware: Para definir los requerimientos de hardware del servidor la investigación se basó en las tecnologías que utiliza este sistema en su implementación y por su crítico consumo de hardware. Mientras que para especificar las del cliente se tuvo en cuenta la necesidad de que la propuesta de solución a implementar no tuviera problemas de rendimiento al ser ejecutada por los usuarios.

Servidor

Requerimientos mínimos de hardware: Core 2 duo a 2.0 GHZ y 2GB RAM.

Cliente

Requerimientos mínimos de hardware: Celeron a 1.5 GHZ y 1 GB RAM

2.5 Exploración

En esta fase los clientes realizan las historias de usuarios que desean que estén para la primera entrega. Cada historia describe una de las funcionalidades que el programa tendrá. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, la tecnología y las prácticas a ser utilizadas durante el proyecto. En algunos casos se utiliza un prototipo para probar la nueva tecnología y explorar algunos aspectos de la arquitectura a ser implementada. La duración de esta fase puede extenderse desde unas pocas semanas a varios meses dependiendo de la adaptación del equipo de desarrollo (47).

[36] Licencia Pública General

Durante la exploración, o en cualquier momento que el equipo necesite resolver incertidumbres o mitigar el riesgo, ellos pueden preparar un punto de fijación. Un punto de fijación es una prueba muy rápida que llega a profundizar sobre el aspecto en cuestión

2.5.1 Historias de usuario

Descripción de una funcionalidad utilizada en XP que debe incorporar un sistema de software, y cuya implementación aporta valor al cliente. La exploración da lugar a un documento de requisitos. Los programadores y el cliente se reúnen y discuten las necesidades del cliente. El cliente escribe historias que describen estas necesidades (48).

En la discusión con el cliente, los programadores eliminan la ambigüedad[37] de las historias y aseguran que éstas puedan ser estimadas y probadas. Los clientes se aseguran de que las historias sean significativas en términos de valor para su negocio. Las historias se escriben normalmente en tarjetas. No contienen mucho texto. Más bien, sirven como recordatorio de las conversaciones mantenidas entre los programadores y el cliente. Más adelante, durante cada iteración, el cliente proporciona un detalle escrito sobre las historias por medio de pruebas de aceptación.

Una historia debe ser lo suficientemente pequeña como para que el equipo la desarrolle durante una iteración (una a tres semanas). Las historias pequeñas son mejores que las grandes. Una historia debe ser probada. El cliente deberá ser capaz de especificar las pruebas de aceptación que verifiquen que la historia es correcta y completa. Superar estas pruebas es la definición de haber concluido una historia. Finalmente, el cliente debe ser capaz de priorizar cada historia. Una historia debería tener una única responsabilidad que condujese a una única prioridad (48).

Plantilla de historia de usuario

Tabla 3: Plantilla de historia de usuario

Historia de usuario

[37] Posibilidad de que algo pueda entenderse de varios modos.

Número: (Número de la historia de usuario, incremental en el tiempo)	Nombre historia de usuario:(Nombre de la historia de usuario para identificarla fácilmente entre los desarrolladores y los clientes)
Usuario: (El usuario del sistema que utiliza o protagoniza la historia)	
Prioridad en negocio: (Alta, Media y Baja), que tan importante es para el cliente.	Riesgo en desarrollo: (Alto, Medio y Bajo), que tan difícil es para el desarrollador.
Iteración asignada: La iteración (liberación en el proceso) a la que corresponde.	
Descripción: La descripción de la historia, detallando las operaciones del usuario y opcionalmente las respuestas del sistema.	
Observaciones: Algunas observaciones de interés, como glosario, información sobre usuarios	

Prioridad en el negocio

Alta: Se le otorga a las historias de usuario que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

Media: Se le otorga a las historias de usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando

Baja: Se le otorga a las historias de usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo

Riesgo en su desarrollo

Alta: Cuando en la implementación de las historias de usuario se consideran la posible existencia de errores que lleven a la inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la historia de usuario que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto. El cliente y el equipo de desarrollo trabajan en conjunto para definir como agrupar las historias de usuario para su lanzamiento**.**

Definición de las historias de usuario

Durante la fase de exploración se identificaron 17 HU para la realización de la aplicación, a continuación se describen las HU 1 y 2, las restantes se pueden encontrar en el **Anexo I**:

Tabla 4: Historia de usuario No 1

Historia de usuario	
Número: 1	Nombre historia de usuario: Exportar reporte de cada problema del archivo de 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Descripción: Los usuarios pueden acceder a los problemas en el archivo 24 horas y pueden crear reportes de estos y exportarlos en documento de formato portable, para obtener las estadísticas, los límites de tiempo, por quien es adicionado, los lenguajes activados, en los idiomas que está disponible, las descripciones del problema, las especificaciones, ejemplos, sugerencias y recomendaciones.	
Observaciones:	

Tabla 5: Historia de usuario No 2

Historia de usuario	
Número: 2	Nombre historia de usuario: Exportar reporte del listado de problemas del archivo 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	

Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Descripción: Los usuarios pueden acceder al listado de problemas en el archivo 24 horas y crear reportes de ellos y exportarlos para obtener información relevante como el título, cantidad de envíos de solución, cantidad de aceptados, porciento de aceptados y puntuación de cada problema.	
Observaciones:	

2.6 Estimación y Planificación

El objetivo de esta fase es fijar la prioridad de cada una de las historias de usuario y se establece cual va a ser el contenido de la primera entrega. Los programadores estiman cuanto esfuerzo requiere cada historia y se establece el cronograma. La duración del calendario para la entrega de la primera liberación no suele superar los dos meses. Duración de la fase de planificación en sí no toma más de dos días (47).

Un proyecto XP se divide en una serie de versiones. Cada versión proporciona un valor de negocio al cliente. Cuando se planifica una versión el cliente selecciona las historias que serán implementadas. Esta selección puede que no sea la más eficiente desde el punto de vista técnico, pero asegura que cada versión proporciona el máximo valor al negocio. El valor de negocio deja a un lado la eficiencia técnica.

En XP se realizan dos actividades de planificación fundamentales, la planificación de liberaciones, donde participa el cliente de forma decisiva, y la planificación de la iteración, donde participan los desarrolladores definiendo las tareas, estimándolas, comprometiéndose e implementándolas (47).

2.6.1 Estimación de esfuerzos por HU

En conjunto con el cliente se realizó una estimación para cada una de las HU, posibilitando tener una visión global del tiempo de desarrollo del sistema. Un punto, equivale a una semana ideal de programación. Los resultados obtenidos se muestran a continuación:

Tabla 6: Estimación de Esfuerzos

Número	Historias de usuarios	Puntos de estimación

1	Exportar reporte de cada problema del archivo de 24 horas.	**0.5**
2	Exportar reporte del listado de problemas del archivo 24 horas.	**0.5**
3	Exportar reporte de las posiciones de los usuarios en el archivo 24 horas.	**0.5**
4	Exportar reporte de las posiciones de instituciones en el archivo 24 horas.	**0.5**
5	Exportar reporte de las posiciones de los países en el archivo 24 horas.	**0.5**
6	Exportar reporte de las estadísticas en el archivo 24 horas.	**0.5**
7	Exportar reporte de las posiciones de los concursantes en los concursos reales.	**0.5**
8	Exportar reporte de las estadísticas de los concursos reales.	**0.5**
9	Exportar reporte de las posiciones en los concursos reales anteriores.	**1**
10	Exportar reporte del perfil de usuarios.	**0.5**
11	Configurar reportes	**1**
12	Crear y exportar reporte de todos los problemas presentados en un concurso ya realizado.	**1**
13	Exportar reporte de los premios de un concurso.	**0.5**

14	Exportar reporte de las estadísticas en los concursos reales anteriores.	**0.5**
15	Exportar reporte de las posiciones en los concursos reales anteriores.	**1**
16	Exportar reporte de las sentencias del archivo 24 horas	**0.5**
17	Exportar reporte de las sentencias de concursos del archivo concursos reales.	**0.5**
	Total	**10.5**

2.6.2 Planificación de iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación.

Para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. Los elementos que deben tomarse en cuenta durante la elaboración del plan de la iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior.

Después de haber definido las historias de usuarios y estimado el esfuerzo propuesto para la realización de cada una de ellas, se tomó la decisión de realizar el sistema en 3 iteraciones, las cuales se detallan a continuación:

Tabla 7: Planificación de iteraciones

Planificación de iteraciones	
Primera iteración	**1** Exportar reporte de cada problema del archivo de 24 horas. **2** Exportar reporte del listado de problemas del archivo 24 horas. **3** Exportar reporte de las posiciones de los usuarios en el archivo 24 horas.

	4 Exportar reporte de las posiciones de instituciones en el archivo 24 horas. **5** Exportar reporte de las posiciones de los países en el archivo 24 horas.
Segunda iteración	**6** Exportar reporte de las estadísticas en el archivo 24 horas. **7** Exportar reporte de las posiciones de los concursantes en los concursos reales. **8** Exportar reporte de las estadísticas de los concursos reales. **9** Exportar reporte de las posiciones en los concursos reales anteriores. **10** Exportar reporte del perfil de usuarios.
Tercera iteración	**11** Configurar reportes **12** Crear y exportar reporte de todos los problemas presentados en un concurso ya realizado. **13** Exportar reporte de los premios de un concurso. **14** Exportar reporte de las estadísticas en los concursos reales anteriores. **15** Exportar reporte de la comparación de usuarios en el archivo 24 horas. **16** Exportar reporte de las sentencias del archivo 24 horas **17** Exportar reporte de las sentencias de concursos del archivo concursos reales.

2.6.3 Plan de duración de iteraciones

El plan de duración de las iteraciones es el encargado de mostrar las historias de usuario que serán implementadas en cada una de las iteraciones, así como la duración estimada y el orden de implementación de cada una de ellas. Este plan se crea para lograr una mayor organización del trabajo.

A continuación se muestran las historias de usuario organizadas por iteraciones:

Tabla 8: Plan de duración de iteraciones

Iteraciones	Orden de las historias de usuarios a implementar	Prioridad	Duración de la iteración
1	**1**-Exportar reporte de cada problema del archivo de 24 horas. **2**-Exportar reporte del listado de problemas del archivo 24 horas. **3**-Exportar reporte de las posiciones de los usuarios en el archivo 24 horas. **4**-Exportar reporte de las posiciones de instituciones en el archivo 24 horas. **5**-Exportar reporte de las posiciones de los países en el archivo 24 horas.	**Media**	**2.5 semanas**
2	**1-**Exportar reporte de las estadísticas en el archivo 24 horas. **2**-Exportar reporte de las posiciones de los concursantes en los concursos reales. **3**-Exportar reporte de las estadísticas de los concursos reales. **4**-Exportar reporte de las posiciones en los concursos reales anteriores. **5**-Exportar reporte del perfil de usuarios.	**Media**	**3 semanas**
3	**1**-Configurar reportes	**Alta**	**5 semanas**

	2-Crear y exportar reporte de todos los problemas presentados en un concurso ya realizado. **3**-Exportar reporte de los premios de un concurso. **4**-Exportar reporte de las estadísticas en los concursos reales anteriores. **5**-Exportar reporte de la comparación de usuarios en el archivo 24 horas. **6**-Exportar reporte de las sentencias del archivo 24 horas **7**-Exportar reporte de las sentencias de concursos del archivo concursos reales.		

2.6.4 Plan de entregas

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores y gerentes).Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

Tabla 9: Plan de Entrega

Entregable	Final 1ra iteración 4ta semana de marzo	Final 2da iteración 4ta semana de abril	Final 3ra iteración 4ta semana de mayo

Módulo para la exportación de reportes del Juez en Línea Caribeño	Versión 0.1	Versión 0.2	Versión 1.0

2.7 Diseño

En esta fase la metodología XP sugiere que hay que conseguir diseños simples y sencillos, procurar hacerlo todo lo menos complicado posible para conseguir un diseño entendible y fácil de implementar que a la larga costará menos tiempo y esfuerzo desarrollar. XP no define una técnica específica de modelado, pueden utilizarse indistintamente esquemas sencillos, tarjetas CRC (Clase, Responsabilidad, Colaboración) o diagramas de clase utilizando UML, siempre que sean útiles y no requieran mucho tiempo en su creación. A continuación se exponen los elementos relevantes para el diseño del sistema (47).

2.7.1 Patrones de diseño

Los patrones de diseño expresan esquemas para definir estructuras de diseño para construir sistemas de software. Describe un problema que se repite una y otra vez y brinda una solución genérica a ese problema, de forma que se utilice dicha solución en todas y cada una de las ocasiones en que se afronte el problema descrito (49).

En el diseño e implementación de la propuesta de solución, se tuvo en cuenta el estudio realizado sobre los patrones de diseño y de asignación general de responsabilidades.

Inversión del control

El principio establece una diferenciación entre el concepto de biblioteca y marco de trabajo, definiendo el primero como un simple conjunto de clases, métodos y funciones que son invocadas por el flujo del programa y que posteriormente devuelven el control a éste (control normal) y el segundo como un diseño más abstracto y elaborado que se encargará, en algún momento, de invocar el código que el programador se encargue de codificar (inversión de control) (50).

El flujo habitual se da cuando es el código del usuario quien invoca a un procedimiento de una biblioteca. La inversión de control sucede cuando es la biblioteca la que invoca el código del usuario. Típicamente sucede cuando la biblioteca es la que implementa las estructuras de alto nivel y es el código del usuario el que implementa las tareas de bajo nivel. La utilización de interfaces y la aparición de los marcos de trabajo han popularizado este término. De hecho es el concepto central del marco de trabajo Spring, dado que implementa un "Contenedor" que se encarga de gestionar las instancias (así como sus creaciones y destrucciones) de los objetos del usuario. Por tanto las aplicaciones que utilicen el marco de trabajo Spring utilizarán Inversión de Control (50).

Inyección de dependencia.

La inyección de dependencias o DI por sus siglas en inglés, es una herramienta utilizada en el diseño orientado a objetos, consiste en inyectar comportamientos a componentes. Esto no es más que extraer responsabilidades a un componente para delegarlas en otro, estableciendo un mecanismo a través del cual el nuevo componente pueda ser cambiado en tiempo de ejecución (51).

Se puede decir que utilizando este patrón las piezas del software serán independientes comunicándose únicamente a través de una interface.

Alta cohesión

El patrón Alta Cohesión es la meta principal que ha de tenerse en cuenta en cada momento en todas las decisiones de diseño. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande, mejorando la claridad y la facilidad con que se entiende el diseño, se simplifican el mantenimiento y las mejoras en funcionalidad y la ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico (52).

La utilización de este patrón permitirá que cada elemento realice una única labor dentro del sistema mejorando la calidad y facilidad del diseño, donde las clases no incorporen demasiadas tareas.

Bajo acoplamiento

El bajo acoplamiento es un principio que se debe tener siempre en cuenta durante las decisiones de diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño y asigna una responsabilidad de manera que el acoplamiento permanezca bajo (52). El uso de este patrón se tuvo en

cuenta con vistas a evitar las dependencias entre las clases y para que los componentes no se afectaran por cambios de otros componentes, siendo fáciles de entender por separado y fáciles de reutilizar.

Patrón decorator

La utilidad principal del patrón Decorator, es la de dotar funcionalidades dinámicas a objetos mediante composición. Se realizará la decoración de los objetos para darles más funcionalidad de la que tienen en un principio. El uso de este patrón permitirá evitar jerarquías de clases complejas, pues la herencia es una herramienta poderosa, pero puede hacer que el diseño sea mucho menos extensible. Ofrecerá más flexibilidad que la herencia estática, pues se podrá añadir y eliminar responsabilidades en tiempo de ejecución y se reducirá el número de clases y del árbol de herencia de clases, permitiendo simplificar el diseño y la implementación de la propuesta de solución (53).

2.7.2 Tarjetas CRC (Clase-Responsabilidad-Colaboración)

La metodología XP propone que todo proyecto que se realice tenga un diseño lo más sencillo posible. Para esto define el uso de tarjetas CRC, las que permiten al programador apreciar el desarrollo orientado a entidades específicas.

A continuación se muestran la tabla muestra de una tarjeta y la tarjeta de la clase ReporteController, las restantes se pueden encontrar en el **Anexo II.**

Tabla muestra de una tarjeta CRC

Tabla 10: Muestra de una Tarjeta CRC

Tarjeta CRC : Nombre de la clase	
Clase: Nombre de la clase que se está modelando.	
Súper clase: Nombre de la clase padre en la herencia.	
Sub clases: Nombre de las clases hija en la herencia.	
Responsabilidades: Es una descripción de alto nivel del propósito de la clase.	**Colaboraciones:** Indica con cuales clases se requiere relación para cumplir la responsabilidad.

Tarjeta CRC (ReporteController)

Tabla 11 Tarjeta CRC ReporteController

Tarjeta CRC : ReporteController	
Clase: ReporteController.	
Súper clase: BaseController.	
Sub clases:	
Responsabilidades: Esta clase es la encargada de controlar la configuración y de proporcionar un contexto adecuado para el acceso a los reportes.	**Colaboraciones:** ReporteDAO.

2.8 Modelo de Datos

Un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia. Es el proceso de analizar los aspectos de interés para una organización y la relación que tienen unos con otros. Resulta en el descubrimiento y documentación de los recursos de datos del negocio. El modelado hace la pregunta " ¿Qué? " en lugar de " ¿Cómo? ", ésta última orientada al procesamiento de los datos. Es una tarea difícil, pero es una actividad necesaria cuya habilidad solo se adquiere con la experiencia (54).

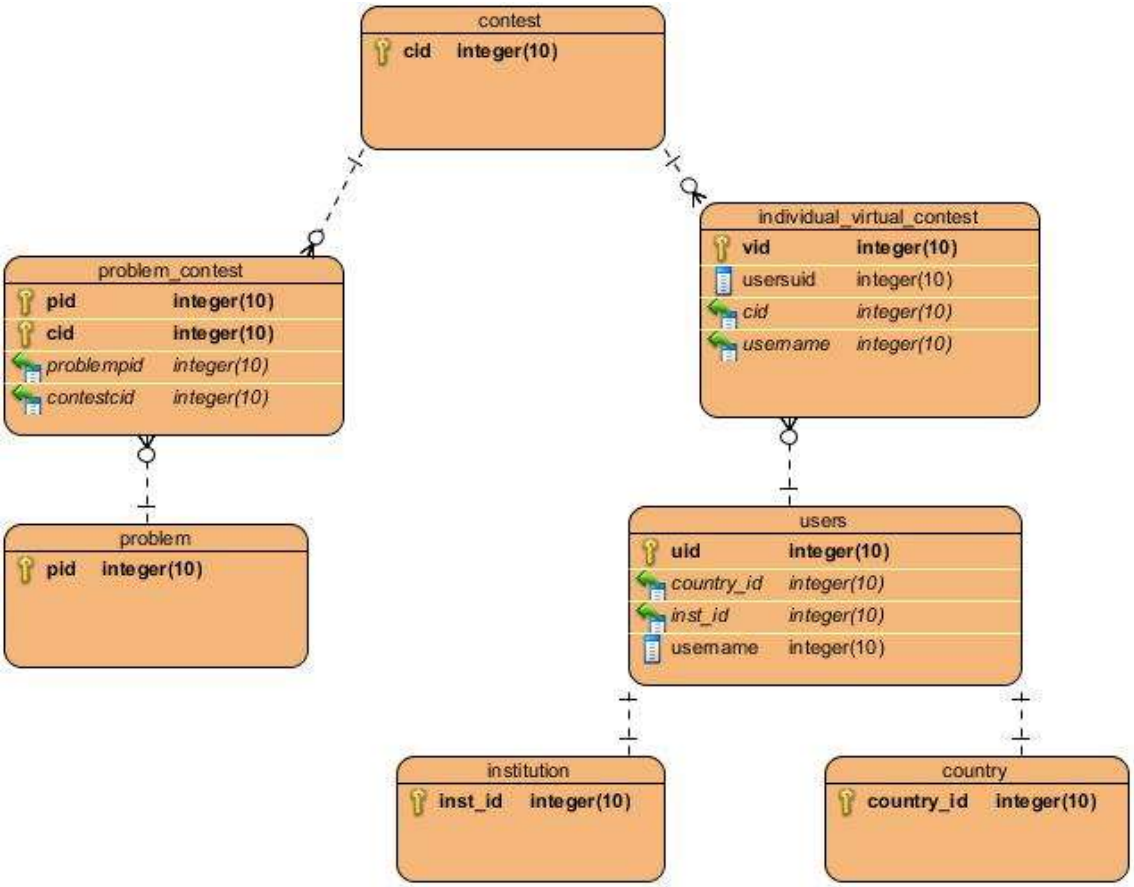

Figura 1: Diagrama de Modelo de Datos

2.9 Conclusiones del capítulo.

En el presente capítulo se abordó todo lo referente a las primeras fases de la metodología de desarrollo de software XP, exploración, planificación y diseño, describiendo cada uno de los artefactos generados durante el transcurso de las mismas. Por lo que se concluye que durante la exploración se definieron 17 HU que se estimaron y planificaron para realizarse en tres iteraciones durante 70 días aproximadamente, donde se realizarán tres entregas de versiones del producto al terminar cada iteración, permitiendo optimizar el trabajo y perfeccionar la organización. Se definieron los patrones de diseño que permitirán la implementación del sistema propuesto, los que representaron mecanismos efectivos para la reutilización del código, permitiendo la obtención de una aplicación que cumpla con las funcionalidades identificadas y sin problemas. Por todo

lo expuesto anteriormente, se evidenció el conocimiento adquirido de las funcionalidades del módulo y se creó las bases para la implementación del sistema.

Capítulo 3: Implementación y Prueba

Introducción

En este capítulo se define la arquitectura a utilizar, se exponen las tareas de la ingeniería generadas por cada historia de usuario y se realizarán las pruebas al software las cuales se derivan de las historias de usuario.

3.1 Propuesta de arquitectura

La arquitectura de un software es determinante para su éxito pues se encarga de establecer los fundamentos para que los miembros del equipo de desarrollo en sus diferentes roles se guíen por una línea de trabajo común y así alcanzar los objetivos de manera exitosa (55).

Para el desarrollo del módulo para la exportación de reportes se selecciona la arquitectura Cliente/Servidor, que consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor), el cual consulta en caso de ser necesaria la base de datos y le da respuesta al cliente.

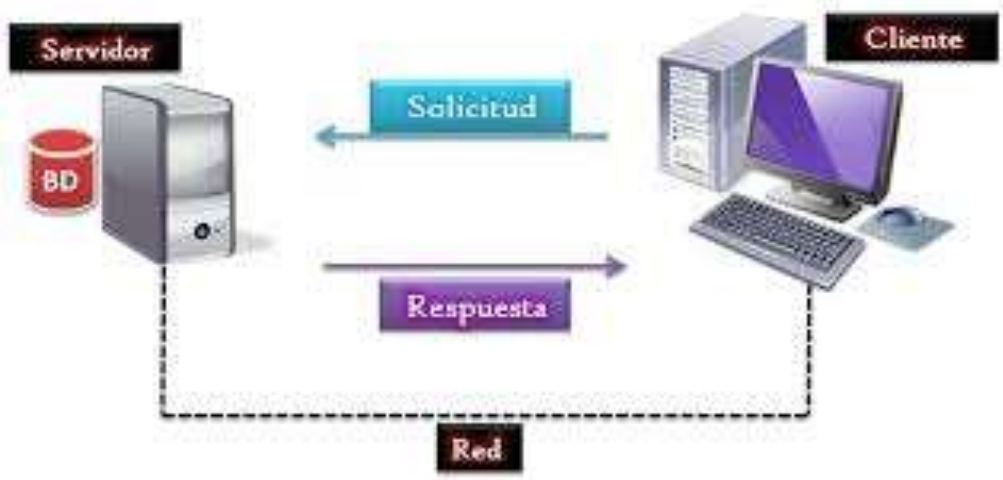

Figura 2: Arquitectura Cliente/Servidor

Cliente

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN[38] o WAN[39]. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente (55).

Servidor

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes (55).

Características del modelo Cliente/Servidor

- El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un servidor da servicio a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final (55).

[38] Red de área local(grupo de equipos que pertenecen a la misma organización y están conectados dentro de un área geográfica pequeña a través de una red)

[39] Red de Área Amplia(red de computadoras que se extiende en una gran franja de territorio(Internet))

Ventajas del esquema Cliente/Servidor

- Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas.
- Una ventaja adicional del uso del esquema Cliente/Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC).
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global (55).

En conclusión, Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. Esto tiene como fin que el usuario de un sistema de información soportado por una arquitectura cliente / servidor, trabaje desde su estación de trabajo con distintos datos y aplicaciones, sin importarle dónde están o dónde se ejecuta cada uno de ellos.

3.1.1 Estilo arquitectónico

Se determinó el uso también del estilo arquitectónico Modelo–Vista–Controlador (MVC) pues el trabajo se puede llevar a cabo en varios niveles y en caso de que exista algún cambio o errores solo se ataca al nivel requerido sin tener que revisar entre código mezclado.

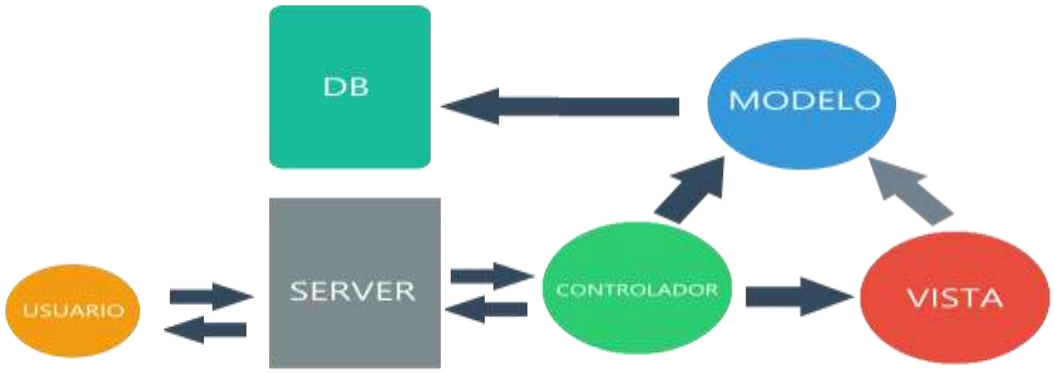

Figura 3: Estilo arquitectónico MVC

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de marcos de trabajos basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores (56).

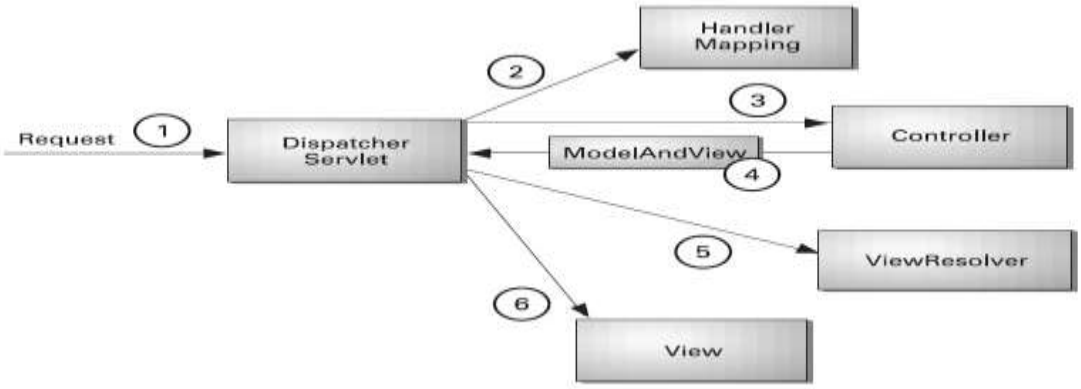

Figura 4: Ciclo de Vida de Spring MVC

En este ciclo el navegador envía una petición (Request) y el distribuidor de servlets[40] (Dispatcher Servlet) se encarga de recoger esta petición (paso 1) para pasárselo al controlador de mapeos (Handler Mapping) (paso 2) que comprueba que dicha petición esté mapeada y le devuelve el controlador asociado a dicha petición al distribuidor de servlets. Una vez que se sabe que controlador se necesita el distribuidor de servlets le pasará el control a dicho controlador (Controller) (paso 3) para que este se encargue de realizar toda la lógica de negocio de la aplicación, este controlador devolverá un objeto modelo y vista (Model, View) (paso 4), el modelo es la información que se desea mostrar y la vista es donde se desea mostrar dicha información. Una vez el distribuidor de servlets tiene el objeto modelo y vista tendrá que asociar el nombre de la vista retornado por el controlador con una vista concreta (View Resolver) es decir una página jsp[41] o jsf[42], (Paso 5). Una vez resulto esto el distribuidor de servlets tendrá que pasar a la vista el modelo, es decir los datos a presentar, y mostrar la vista (View) (paso 6).

Componentes del patrón MVC

Vista

Presenta el modelo en un formato adecuado para interactuar con la interfaz de usuario por tanto requiere de dicho modelo la información que debe representar como salida (57).

Controlador

Es el responsable de gestionar la interacción con el usuario. El usuario interactúa únicamente con controladores, que gestiona los eventos y acciones sobre la interfaz gráfica, traduciéndolos a invocaciones de servicios proporcionados por el modelo o a peticiones a funcionalidades propias de la vista (57).

Modelo

Encapsula las funcionalidades y los datos del sistema. Es independiente de los mecanismos de presentación de la información (interfaces gráficas) y de interacción con el usuario. Es la representación específica de la información con la cual el sistema opera permitiendo derivar nuevos datos y asegurando la integridad de estos (57).

[40] Clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.
[41] JavaServer Pages: Tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML.
[42] JavaServer Faces: Framework para aplicaciones Java basadas en web

3.2 Fase de implementación

En el transcurso de las iteraciones se desarrollará la implementación de las HU seleccionadas para ser realizadas en cada una de ellas y conjuntamente se efectuará una revisión del plan de iteraciones permitiendo la creación de tareas para ayudar a organizar la implementación exitosa de la HU. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

3.2.1 Tareas de ingeniería

Cada una de las historias de usuario se transformará en tareas que serán desarrolladas por programadores. Estas tareas no son nada más que la representación gráfica de las responsabilidades asignadas, de cada miembro del equipo de desarrollo en XP.

Tabla muestra de una tarea de ingeniería

Tabla 12: Muestra de tarea de ingeniería.

Tareas de ingeniería	
No. de la tarea: [Los números deben ser consecutivos]	No. de la HU: [Número de la historia de usuario a la que pertenece la tarea]
Nombre de la tarea: [Nombre que identifica a la tarea.]	
Tipo de tarea: [Las tareas pueden ser de: Desarrollo, Corrección o Mejora]	Puntos estimados: [Tiempo en semanas que se le asignará. (Estimado)]
Fecha inicio:[Inicio de la tarea]	Fecha fin:[Culminación de la tarea]
Programador responsable: [Los programadores encargados]	
Descripción: [Breve descripción de la tarea.]	

Con respecto a la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se muestran las tablas sobre los tiempos de implementación de

cada HU en las tres iteraciones y después las tareas de ingeniería desglosadas de la HU #1, las restantes están expuestas en el **Anexo III.**

Primera iteración

En esta iteración se implementan las HU 1, 2, 3, 4 y 5.

Tabla 13: HU abordadas en la primera iteración.

	Tiempo de implementación	
Historias de usuarios	**Estimación**	**Real**
1-Exportar reporte de cada problema del archivo de 24 horas.	0.5	0.5
2-Exportar reporte del listado de problemas del archivo 24 horas.	0.5	0.5
3-Exportar reporte de las posiciones de los usuarios en el archivo 24 horas.	0.5	0.5
4-Exportar reporte de las posiciones de instituciones en el archivo 24 horas.	0.5	0.5
5-Exportar reporte de las posiciones de los países en el archivo 24 horas.	0.5	0.5

Tabla 14: Tarea de ingeniería #1 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 1
Nombre de la tarea: Utilizar la biblioteca jsPDF.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5

Fecha inicio: 1/03/2015	Fecha fin: 4/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Utilizar jsPDF para generar y exportar reportes en la implementación de la solución propuesta.	

Tabla 15: Tarea de ingeniería #2 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 1
Nombre de la tarea: Encontrar recursos para el reporte.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1/03/2015	Fecha fin: 4/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Los recursos para generar los reportes se encuentran ubicados en el HTML del sitio, para su captura se hace necesario utilizar selectores de JQuery.	

Tabla 15: Tarea de ingeniería #2 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 1
Nombre de la tarea: Implementación de las pruebas unitarias definidas para la exportación del reporte de cada problema del archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1/03/2015	Fecha fin: 4/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 16: Tarea de ingeniería #3 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 1
Nombre de la tarea: Diseñar reportes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1/03/2015	Fecha fin: 4/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Diseñar un formato para la elaboración de los reportes en PDF.	

Tabla 17: Tarea de ingeniería #4 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 4	No. de la HU: 1
Nombre de la tarea: Cargar imagen	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1/03/2015	Fecha fin: 4/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: La imagen que contienen los reportes se encuentra ubicada en la dirección src/main/webapp/images del proyecto. Deben cargarse desde allí en tiempo de ejecución haciendo uso de la biblioteca jsPDF.	

Tabla 18: Tarea de ingeniería #5 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 5	No. de la HU: 1
Nombre de la tarea: Exportar reporte a formato PDF	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1/03/2015	Fecha fin: 4/03/2015

Programador responsable: Jorge Gabriel Díaz Reinoso.
Descripción: Exportar reporte a formato PDF y mostrarlo por pantalla en una nueva ventana.

Tabla 19: Tarea de ingeniería #2 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 6	No. de la HU: 1
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 1/03/2015	Fecha fin: 4/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Segunda iteración

Tabla 20: HU abordadas en la segunda iteración.

	Tiempo de implementación	
Historias de Usuarios	Estimación	Real
1-Exportar reporte de las estadísticas en el archivo 24 horas.	0.5	0.5
2-Exportar reporte de las posiciones de los concursantes en los concursos reales.	0.5	0.5
3-Exportar reporte de las estadísticas de los concursos reales.	0.5	0.5

4-Exportar reporte de las posiciones en los concursos reales anteriores.	1	1
5-Exportar reporte del perfil de usuarios.	0.5	0.5

Tercera iteración

Tabla 21: HU abordadas en la tercera iteración.

	Tiempo de implementación	
Historias de Usuarios	Estimación	Real
1-Configurar reportes	1	1
2-Crear y exportar reporte de todos los problemas presentados en un concurso ya realizado.	1	1
3-Exportar reporte de los premios de un concurso.	0.5	0.5
4-Exportar reporte de las estadísticas en los concursos reales anteriores.	0.5	0.5
5-Exportar reporte de la comparación de usuarios en el archivo 24 horas.	1	1
6-Exportar reporte de las sentencias del archivo 24 horas	0.5	0.5

7-Exportar reporte de las sentencias de concursos del archivo concursos reales.	0.5	0.5

3.3 Pruebas

Uno de los pilares de la metodología XP es el uso de las pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Para validar el cumplimiento de las funcionalidades del software construido, se realizaron pruebas unitarias, pruebas de integración, pruebas de carga y pruebas de aceptación.

3.3.1 Pruebas unitarias

Se define como prueba unitaria, a la prueba de uno de los módulos que componen un programa. Todos los módulos deben de pasar por las pruebas antes de ser publicados en caso contrario se considerará que el código no está completo. En los últimos años se han desarrollado un conjunto de herramientas que facilitan la elaboración de pruebas unitarias en diferentes lenguajes. Como la solución se realiza utilizando Java y Javascript se utilizaron dos marcos de trabajo:

JUnit

Para el trabajo realizado en java se utilizó JUnit, librería que está desarrollada para poder probar el funcionamiento de las clases y métodos que componen la aplicación, y asegurarse de que se comportan como deben ante distintas situaciones de entrada (58). En la actualidad las herramientas de desarrollo como NetBeans y Eclipse cuentan con plug-ins [43]que permiten que la generación de las plantillas necesarias para la creación de las pruebas de una clase Java se realice de manera automática, facilitando al programador enfocarse en la prueba y el resultado esperado, y dejando a la herramienta la creación de las clases que permiten coordinar las pruebas.

[43] Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica

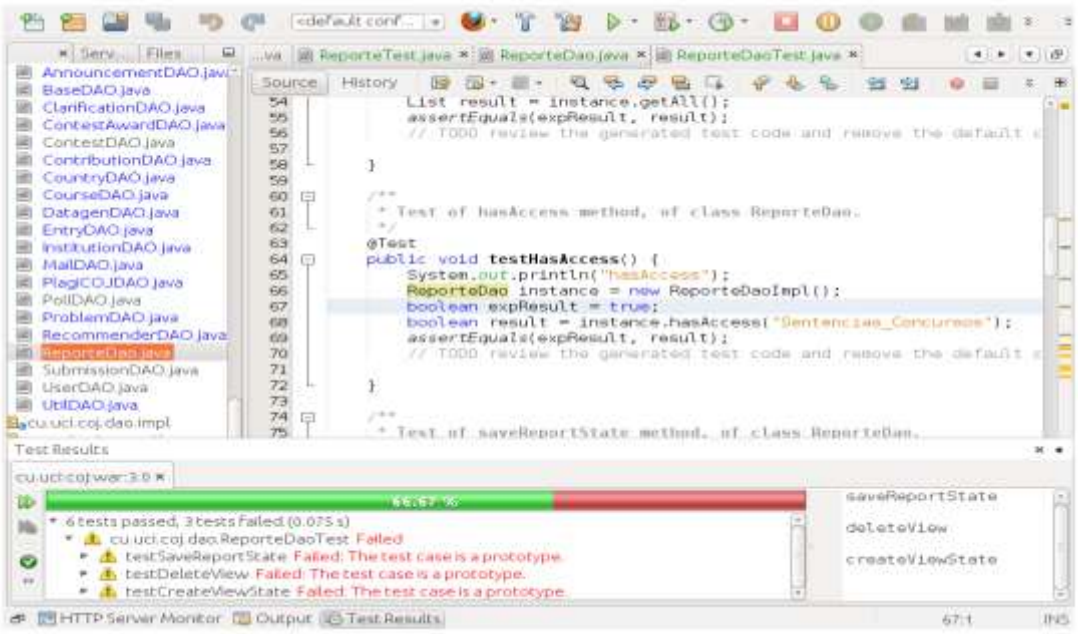

Figura 5: Interfaz del plug-ins Junit en Netbeans

A medida que se fue programando el módulo se fue realizando esta prueba unitaria, comprobando los resultados obtenidos en las diferentes pruebas, algunos de los resultados se pueden evidenciar en el **Anexo IV**.

QUnit

Para la validación del trabajo realizado en Javascript se utilizará QUnit que es un marco para pruebas unitarias creado por jQuery, donde, en lugar de utilizar la consola para mostrar los resultados, crea un reporte en HTML con los resultados de las pruebas realizadas. En QUnit, cada comparación que se hace se llama assert, mientras que el conjunto de asserts es llamado pruebas (test) (59). QUnit fue desarrollado originalmente por John Resig como parte de jQuery. En 2008 obtuvo su propia casa, el nombre y la documentación de la API, permitiendo que otros lo utilizan para su unidad de pruebas también. En el momento en que todavía dependía de jQuery. Una reescritura en 2009 fijó eso, y ahora QUnit funciona totalmente independiente (60).

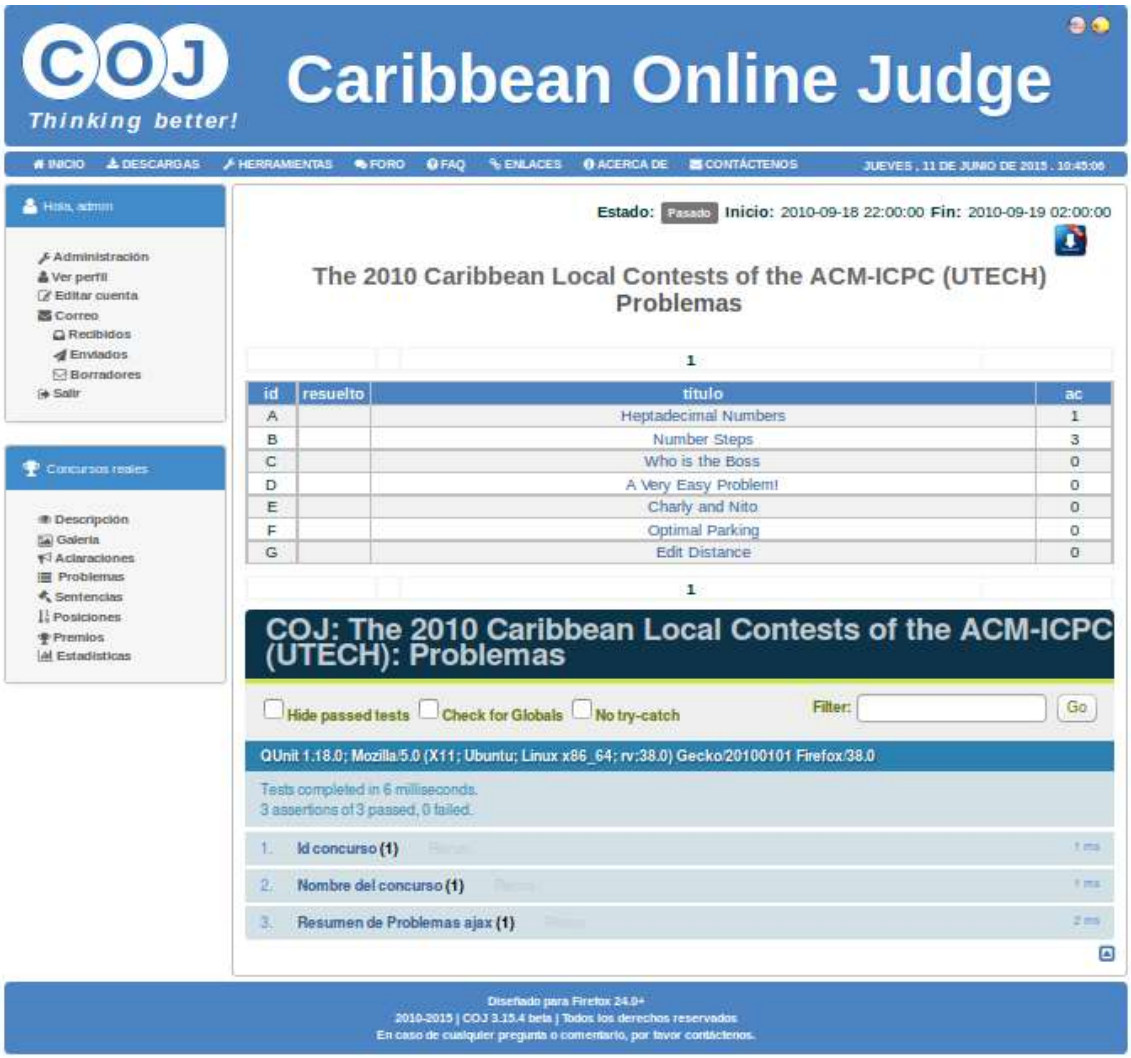

Figura 6: Interfaz del plug-ins Qunit

Durante el desarrollo de la solución se fue implementando la prueba unitaria con QUnit para validar las funciones utilizadas en la solución; se pueden encontrar las pruebas en el **Anexo IV.** Durante el proceso de validación desarrollado por QUnit que constó de tres iteraciones, se encontraron No Conformidades las cuales se fueran solucionando en el transcurso de las iteraciones.

3.3.2 Pruebas de integración

Las pruebas de integración son la base del software en el que los módulos individuales se combinan y se ensamblan como un grupo. Se produce después de las pruebas unitarias y antes de las pruebas de

aceptación. El propósito de las pruebas de integración es verificar funcionalidades, rendimiento y los requisitos de fiabilidad colocados en los principales elementos de diseño (61). Este tipo de pruebas son ejecutas por el equipo de desarrollo , donde los casos de prueba se construyen para comprobar que todos los componentes interactúan correctamente, por ejemplo a través de llamadas a procedimientos o activaciones de procesos, y esto se hace después de probar los módulos individuales con las pruebas unitarias.

En el caso de la solución implementada esta prueba de integración se realizó utilizando JUnit, pues cuando se validan los métodos implementados en las diferentes clases debemos acceder a otras clases para el correcto desarrollo de las pruebas. El planteamiento anterior se afirma dado que una prueba de integración ejercita el funcionamiento de una o varias clases en el contexto de la aplicación y estas verifican la interacción con una base de datos (62).

Por ejemplo si se tiene una clase como en la propuesta de solución ReporteDAOImpl y en esta se encuentra un método para crear reporte, la prueba que se debe implementar para comprobar la inserción a la base de datos necesitará de otras clases para su funcionamiento como ReporteDao. Con la puesta en práctica de esta prueba de integración aumentó el nivel de confianza de la implementación porque se sabe que el estado actual del módulo está bajo control, por otra parte la correcta ejecución de las pruebas garantizó que la integración de la solución no corrompe la base del código existente aumentando la satisfacción del trabajo realizado.

3.3.3 Pruebas de carga

Realizada la validación del código fuente mediante las pruebas de unidad y de integración también fue necesario efectuar pruebas de carga. Con el objetivo de obtener datos sobre la carga del sistema, para comprobar, de manera anticipada, el funcionamiento que tendrá el servidor cuando esté en plena operación, ayudando a localizar e identificar errores que la aplicación pueda tener.

JMeter

La herramienta utilizada para estas pruebas es el Apache JMeter. Software de código abierto, diseñado para cargar el comportamiento funcional de las pruebas y medir su rendimiento. Originalmente fue diseñado para probar las aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba (63).

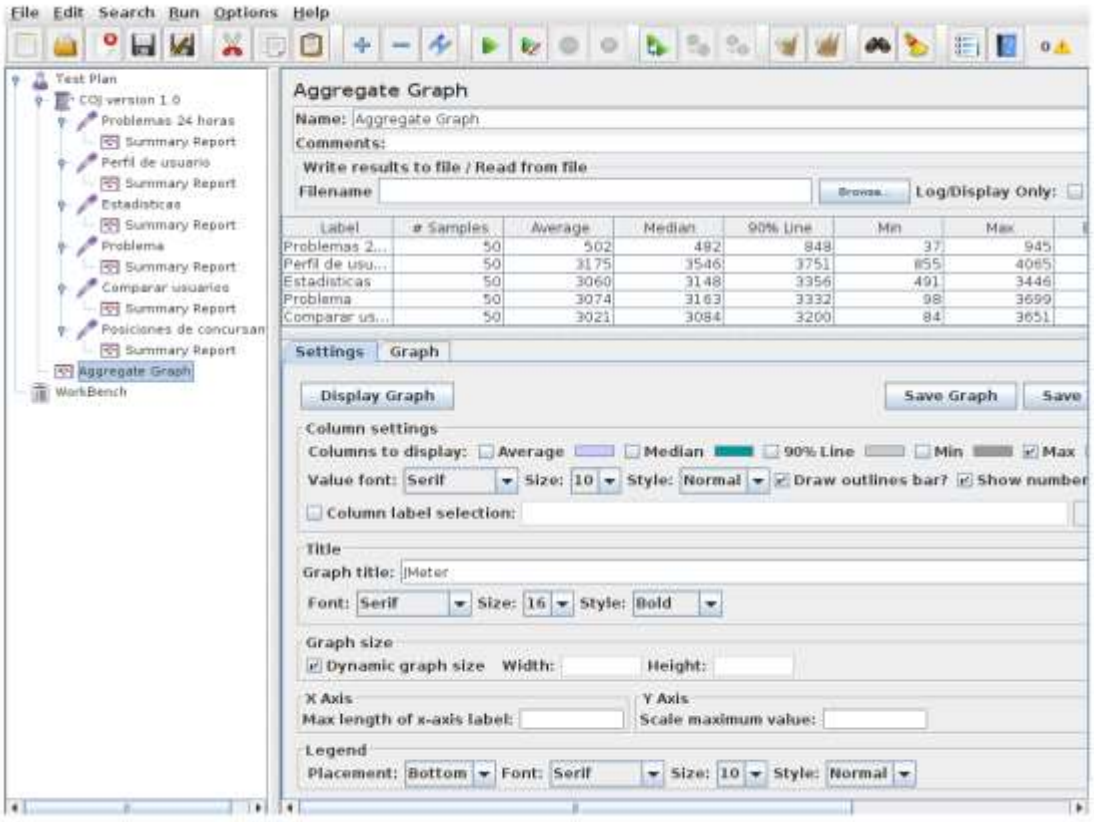

Figura 7: Plan de pruebas de carga

Para el plan de pruebas de carga se utilizaron las páginas del perfil de usuario, listado de problemas, estadísticas, problema, comparar usuarios y posiciones de usuarios; las cuales para cada petición realizan gran número de operaciones como exportar reportes y el grupo de hilos utilizado constó con 50 usuarios. El plan de pruebas se fundamentó en que la aplicación pudiera responder a las peticiones de los usuarios, sin menospreciar el tiempo de respuesta obtenido aunque no es relevante. Al concluir la prueba el COJ respondió correctamente a las peticiones realizadas por los usuarios por lo que se puede decir que las peticiones realizadas al módulo para la exportación de reportes fueron satisfactorias y se evidenció su correcta integración con el sistema. Los artefactos generados de esta prueba se pueden encontrar en el **Anexo IV**.

3.3.4 Pruebas de aceptación

Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Pruebas de aceptación"; estas pruebas son creadas y usadas por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.

Una prueba de aceptación puede ir desde un informal caso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. De hecho, las pruebas de aceptación pueden tener lugar a lo largo de semanas o meses, descubriendo así errores latentes o escondidos que pueden ir degradando el funcionamiento del sistema. Estas pruebas son muy importantes, permitiendo definir el paso de nuevas fases del proyecto como el despliegue y mantenimiento (64).

A continuación se muestra la tabla de caso de prueba de aceptación de la HU # 1, las restantes se pueden encontrar en el **Anexo IV**.

Tabla 22: Prueba de Aceptación HU#1

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_1	Nombre historia de usuario: 1
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de cada problema del archivo de 24 horas.	
Condiciones de Ejecución: La información que se desea exportar, en este caso los problemas debe de existir.	
Entrada / Pasos de ejecución: Se accede al archivo 24 horas y se selecciona un problema, posteriormente se accede a la opción PDF y se exportará todo la información relacionada con dicho problema.	
Resultado Esperado: Se muestra en formato PDF un reporte con las características del problema: descripción, especificación de entrada, especificación de salida, ejemplo de entrada, ejemplo de salida y las sugerencias.	
Evaluación de la Prueba: Prueba satisfactoria.	

Resultados obtenidos de las pruebas

Al concluir el proceso de pruebas que constó de tres iteraciones, para efectuar las entregas de las funcionalidades previamente pactadas, se documentó la cantidad de No Conformidades (NC) encontradas en las pruebas, detectándose 4 NC significativas y 8 NC no significativas. En la siguiente gráfica se evidencia los resultados de las pruebas obtenidos en las iteraciones:

Figura 8: Resultados de las pruebas.

Dentro de las NC significativas se pueden mencionar algunas como:

- Elevado tiempo de respuesta en la creación y exportación de los reportes del perfil de usuario y de las estadísticas del módulo 24 horas.

- Datos faltantes en la generación del reporte de problemas en los concursos reales.

El ciclo de pruebas en las iteraciones pertinentes concibió la corrección de las NC permitiendo la reestructuración del código, puesto que se verificó que las modificaciones no introdujeron errores, se simplificó la integración asegurando que las partes individuales funcionan correctamente y se eliminaron los errores existentes.

Conclusiones del capítulo

En este capítulo la arquitectura y el estilo arquitectónico seleccionados para guiar el desarrollo, facilitaron la comprensión de la estructura interna del sistema, de cómo están integrados los componentes y proporcionó una visión global de la propuesta a desarrollar. Las tareas de ingeniería para las HU en cada iteración aseguraron la implementación del producto, mientras que los casos de prueba de aceptación sirvieron para guiar al cliente en cuanto a las funcionalidades del sistema, para ser aceptado y que el mismo tuviera la mayor conformidad con lo desarrollado. Las pruebas unitarias realizadas utilizando los marcos de trabajo JUnit y QUnit proporcionaron la validación de la implementación realizada, así como también las pruebas de integración asegurando que el módulo no afecta el código del sistema. Todo este proceso de pruebas realizadas como parte de la validación del sistema propuesto sirvió para obtener un producto con la calidad requerida. Por lo anterior expuesto se puede afirmar que se da cumplimento al objetivo general, que ha sido elaborar la propuesta del modelo lógico y físico del módulo para la exportación de reportes del COJ, obteniéndose una herramienta que genera y exporta reportes dinámicamente del lado del cliente sin interrumpir el rendimiento del servidor.

Conclusiones Generales

Con el presente trabajo se obtuvieron resultados favorables que pueden concluirse de la siguiente forma:

- El estudio realizado de aplicaciones similares para tratar de resolver los problemas existentes, planteó la necesidad de construir un módulo propio que cumpla con los objetivos trazados por el cliente, para realizar el proceso de exportación de reportes.

- Con el resultado obtenido del diseño y de la implementación de la solución, los usuarios del COJ contarán con un módulo capaz de crear y exportar reportes del archivo 24 horas, de los concursos reales y de la información del usuario, permitiendo mayor accesibilidad de la información para su posterior estudio.

- Se realizaron disímiles pruebas para la validación de la solución informática, a través de las cuales se pudieron detectar y solucionar los errores en cada una de las iteraciones previstas, concluyendo el correcto desarrollo del módulo para la exportación de reportes del COJ.

Recomendaciones

Teniendo como base los resultados de esta investigación se recomienda para posibles mejoras:

Estudiar y utilizar la biblioteca atmosphere para realizar varias funcionalidades relacionadas con reportes como:

- Crear y exportar reporte de certificación de la publicación de un problema.
- Crear y exportar reporte de diploma para el primer lugar en las posiciones de usuarios, para cada participante de un concurso y para los jueces que participaron en dicho concurso.

Implementar una futura versión del módulo que integre funcionalidades destinadas a la seguridad de los reportes.

Referencias bibliográficas

1. **ACM-ICPC Live Archive.** [En línea] [Citado el: 23 de enero de 2015.] https://icpcarchive.ecs.baylor.edu/.

2. **Caribbean Online Judge.** [En línea] [Citado el: 22 de enero de 2015.] http://coj.uci.cu/general/about.xhtml.

3. **Reportes.** [En línea] [Citado el: 19 de febrero de 2015.] http://sipec.sep.gob.mx/WebHelp/reportes/reporte.htm.

4. **Defining Modules, Modularity and Modularization.** [En línea] [Citado el: 22 de enero de 2015.] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.454.868&rep=rep1&type=pdf. 87-89867-60-2.

5. **POJ**. [En línea] [Citado el: 15 de 1 de 2015.] http://poj.org/.

6. **Tianjin University Online Judge**. [En línea] [Citado el: 22 de enero de 2015.] http://acm.tju.edu.cn/toj/.

7. **Sphere Online Judge**. [En línea] [Citado el: 23 de enero de 2015.] http://www.spoj.pl.

8. **TopCoder.** [En línea] [Citado el: 23 de enero de 2015.] http://www.topcoder.com.

9. **Codeforces.** [En línea] [Citado el: 23 de enero de 2015.] http://www.codeforces.com.

10. **UVa Online Judge**. [En línea] [Citado el: 23 de enero de 2015.] http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=23.

11. Caribbean Online Judge (COJ). [En línea] [Citado el: 24 de enero de 2015.] http://coj.uci.cu/general/about.xhtml.

12. **Pérez, Javier Eguíluz**. Introducción a JavaScript. [En línea] [Citado el: 25 de enero de 2015.] http://www.intercambiosvirtuales.org/libros-manuales/introduccin-javascript.

13. **ÁLVAREZ, M. A**. Características y ventajas de las CSS. [En línea] 2010. [Citado el: 25 de enero de 2015.] http://www.desarrolloweb.com/articulos/182.php.

14. HTML. [En línea] [Citado el: 15 de 4 de 2015.] http://www.ri5.com.ar/ayuda03.php.

15. **Byous, Jon.** Java technology: The early years. [En línea] [Citado el: 25 de enero de 2015.] http://java.sun.com.

16. **HERNÁNDEZ ORALLO, ENRIQUE**. El Lenguaje Unificado de Modelado. [En línea] 2009. [Citado el: 26 de enero de 2015.] http://www.disca.upv.es/enheror/pdf/ActaUML.PDF.

17. **Davis, William S.** Herramientas CASE : metodología estructurada para el desarrollo de los sistemas. [En línea] [Citado el: 26 de enero de 2015.] http://www.worldcat.org/title/herramientas-case-metodologia-estructurada-para-el-desarrollo-de-los-sistemas/oclc/55299445. 8428319278 9788428319270.

18. **Maldonado, Daniel**. El CoDiGo K. Qué son los IDE de programación. [En línea] [Citado el: 27 de enero de 2015.] http://www.elcodigok.com.ar/2007/09/que-son-los-ide-de-programacin.html.
19. **NetBeans**, IDE gráfica para programación. [En línea] [Citado el: 28 de enero de 2015.] http://bitsbeta.com/netbeans-ide-grafica-programacion.
20. **Carrero, Ángel**. Geany, un IDE multiplataformas. [En línea] 24 de febrero de 2011 2011. [Citado el: 27 de enero de 2015.] http://www.programacion.com/noticia/geany_un_ide_multiplataformas_2029.
21. **Various Licenses and Comments about Them Ver sección Eclipse Public License.** [En línea] [Citado el: 28 de enero de 2015.]
22. Sistemas de control de versiones. [En línea] [Citado el: 16 de febrero de 2015.] http://producingoss.com/.
23. SCV. [En línea] [Citado el: 16 de febrero de 2015.] http://www.nongnu.org/cvs/.
24. Subversion. [En línea] [Citado el: 16 de febrero de 2015.] http://subversion.tigris.org/.
25. Mercurial. [En línea] [Citado el: 16 de febrero de 2015.] http://www.selenic.com/mercurial/.
26. Git. [En línea] [Citado el: 16 de febrero de 2015.] http://git.or.cz/.
27. GitLab. [En línea] [Citado el: 12 de 4 de 2015.] www.gitlab.com.
28. Glassfish. [En línea] [Citado el: 16 de febrero de 2015.] https://glassfish.java.net/.
29. iis. [En línea] [Citado el: 16 de febrero de 2015.] http://www.iis.net/.
30. Cherokee. [En línea] [Citado el: 16 de febrero de 2015.] http://cherokee-project.com/.
31. Apache Tomcat. [En línea] [Citado el: 16 de febrero de 2015.] http://tomcat.apache.org/.
32. **Bachman, Charles W**. Sistema Gestor de Bases de Datos. [En línea] [Citado el: 17 de febrero de 2015.] http://dl.acm.org/citation.cfm?id=362534.
33. **PETKOVIĆ, DUŠAN**. Microsoft SQL Server. [En línea] 2012. [Citado el: 28 de enero de 2015.] http://www.amazon.es/Microsoft-SQL-Server-2012-Beginners/dp/0071761608.
34. **PostgreSQ**L. [En línea] [Citado el: 28 de enero de 2015.] http://www.postgresql.org/about/.
35. **Riehle, Dirk.** framework. [En línea] [Citado el: 18 de febrero de 2015.] http://dl.acm.org/citation.cfm?id=286951.
36. jquery. [En línea] [Citado el: 18 de febrero de 2015.] http://jquery.com/.
37. jasperreports. [En línea] [Citado el: 17 de febrero de 2015.] https://community.jaspersoft.com/project/jasperreports-library.

38. google web toolkit. [En línea] [Citado el: 16 de febrero de 2015.] http://es.slideshare.net/emiliobg/desarrollo-de-aplicaciones-web-con-google-web-toolkit.
39. Vaadin. [En línea] [Citado el: 18 de febrero de 2015.] http://carlospesquera.com/vaadin-aplicaciones-web-java-como-si-fuesen-de-escritorio/.
40. Spring. [En línea] [Citado el: 18 de febrero de 2015.] http://projects.spring.io/spring-framework/.
41. jsPDF. [En línea] [Citado el: 18 de febrero de 2015.] https://parall.ax/products/jspdf.
42. **Philippe Kruchten**. The Rational Unified Process: An Introduction. [En línea] [Citado el: 29 de enero de 2015.] http://www.amazon.com/The-Rational-Unified-Process-Introduction/dp/0321197704. 978-0321197702.
43. **Beck, Kent.** Extreme programming explained: embrace change. [En línea] [Citado el: 6 de 4 de 2015.] http://www.amazon.com/Extreme-Programming-Explained-Embrace-Edition/dp/0321278658#reader_0321278658. 0-201-61641-6 .
44. **Proyectos Ágiles**. [En línea] [Citado el: 29 de enero de 2015.] http://www.proyectosagiles.org/que-es-scrum/.
45. *Guia Practica de Gestión de Requisitos.* s.l. : INTECO.
46. Requerimientos funcionales y no funcionales. [En línea] [Citado el: 15 de 4 de 2015.] http://www.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales#scribd.
47. Fases de la Programación Extrema. [En línea] [Citado el: 16 de 4 de 2015.] http://programacionextrema.tripod.com/fases.htm.
48. Historias de Usuarios. [En línea] [Citado el: 16 de 4 de 2015.] http://es.slideshare.net/MiquelMora/historias-de-usuario.
49. Patrones de diseño. [En línea] [Citado el: 6 de marzo de 2015.] http://es.slideshare.net/mcapeltu/patrones-diseno.
50. **Fowler, Martin. Inversion del control**. [En línea] [Citado el: 18 de 4 de 2015.] http://martinfowler.com/bliki/InversionOfControl.html.
51. Inyección de dependencias. [En línea] [Citado el: 6 de marzo de 2015.] https://msdn.microsoft.com/es-es/library/jj635998.aspx.
52. **Roberto Canales Mora**. Patrones de GRASP. [En línea] [Citado el: 19 de 4 de 2015.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=grasp.

53. Patrón Decorator. [En línea] [Citado el: 12 de 5 de 2015.] http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-decorator.
54. Modelo de Datos. [En línea] [Citado el: 20 de 4 de 2015.] http://ict.udlap.mx/people/carlos/is341/bases02.html.
55. **Avalos, Sandra Adanary Ibarra**. Cliente Servidor. [En línea] Universidad de Colima. [Citado el: 15 de 4 de 2015.] http://docente.ucol.mx/sadanary/public_html/bd/cs.htm.
56. *Patrón Modelo-Vista-Controlador.* **Yenisleidy Fernández Romero, Yanette Díaz González.** No. 1, La Habana : s.n., 2012, Vol. Vol. 11. 1729-3804.
57. **Alonso, G., Casati, F., Kuno, H., Machiraju, V.** *Web Services Concepts, Architectures and Applications.* s.l. : Springer, 2004. 3-540-44008-9.
58. Casos de prueba: JUnit. [En línea] [Citado el: 16 de 4 de 2015.] http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion04-apuntes.html.
59. QUnit. [En línea] [Citado el: 13 de 4 de 2015.] http://cevichejs.com/5-pruebas.
60. Qunit. [En línea] [Citado el: 2 de 5 de 2015.] http://qunitjs.com/.
61. Pruebas de integracion. [En línea] [Citado el: 29 de 5 de 2015.] http://docsetools.com/articulos-educativos/article_11443.html.
62. TEST DE PRUEBAS. [En línea] [Citado el: 15 de 5 de 2015.] http://webcache.googleusercontent.com/search?q=cache:ySqGDvWX-2oJ:www.madrid.org/arquitecturasw/images/documentacion/fwjusticia/manuales/Test_de_pruebas.doc+&cd=1&hl=es-419&ct=clnk&gl=cu.
63. **Apache JMeter**. [En línea] [Citado el: 16 de 5 de 2015.] https://jmeter.apache.org/index.html.
64. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. Quinta edición. [En línea] [Citado el: 26 de enero de 2015.] http://es.slideshare.net/jdbg16/ingenieria-de-software-un-enfoque-prctico-pressman-5th-ed.

Anexos

Anexo I

Tabla 23: Historia de usuario No 3

Historia de usuario	
Número: 3	Nombre historia de usuario: Exportar reporte de las posiciones de los usuarios en el archivo 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Descripción: Permitirá la creación y exportación de reportes de las posiciones de los usuarios para obtener la posición, el país, el usuario, la cantidad de soluciones enviadas, la cantidad de soluciones aceptadas, el porciento de aceptadas y la puntuación.	
Observaciones:	

Tabla 24: Historia de usuario No 4

Historia de usuario	
Número: 4	Nombre historia de usuario: Exportar reporte de las posiciones de instituciones en el archivo 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media

Iteración asignada: 1
Descripción: Permitirá la creación y exportación de reportes de las posiciones de las instituciones para obtener la posición, el país, la institución, cantidad de usuarios, la cantidad de soluciones aceptadas y la puntuación.
Observaciones:

Tabla 25: Historia de usuario No 5

Historia de usuario	
Número: 5	Nombre historia de usuario: Exportar reporte de las posiciones de los países en el archivo 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 1	
Descripción: Permitirá la creación y exportación de reportes de las posiciones de los países para obtener la posición, el país, las instituciones que posee, cantidad de usuarios, la cantidad de soluciones aceptadas y la puntuación.	
Observaciones:	

Tabla 26: Historia de usuario No 6

Historia de usuario	
Número: 6	Nombre historia de usuario: Exportar reporte de las estadísticas en el archivo 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	

Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 2	
Descripción: El usuario tendrá la opción de crear y exportar un reporte sobre las estadísticas de los múltiples lenguajes de programación en el archivo 24 horas.	
Observaciones:	

Tabla 27: Historia de usuario No 7

Historia de usuario	
Número: 7	Nombre historia de usuario: Exportar reporte de las posiciones de los concursantes en los concursos reales.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 2	
Descripción: Permitirá la creación y exportación de reportes de las posiciones de los usuarios en los concursos reales para obtener la posición, el país, el usuario, los concursos participados, la cantidad de soluciones enviadas, la cantidad de soluciones aceptadas y el porciento de aceptadas.	
Observaciones:	

Tabla 28: Historia de usuario No 8

Historia de usuario	
Número: 8	Nombre historia de usuario: Exportar reporte de las estadísticas de los concursos reales.

Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 2	
Descripción: El usuario tendrá la opción de crear y exportar un reporte sobre las estadísticas de los múltiples lenguajes de programación en el archivo concursos reales.	
Observaciones:	

Tabla 29: Historia de usuario No 9

Historia de usuario	
Número: 9	Nombre historia de usuario: Exportar reporte de las posiciones en los concursos reales anteriores.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 2	
Descripción: El usuario podrá acceder a los concursos reales anteriores y exportar un reporte de las posiciones de los equipos de usuarios que hayan realizado dicho concurso, obteniendo el primer aceptado en el concurso, el primer aceptado en el problema, los envíos aceptados, los envíos rechazados y los envíos pendientes diferenciándose por colores.	
Observaciones:	

Tabla 30: Historia de usuario No 10

Historia de usuario

Número: 10	Nombre historia de usuario: Exportar reporte del perfil de usuarios.
Usuario: Administrador y usuario autenticado.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 3	
Descripción: Exportar un reporte con información del perfil del usuario, especificando sus datos personales y los datos que posee en el sistema como las notificaciones, el idioma predeterminado, el lenguaje de programación predeterminado, estado, las fechas de registro, del ultimo envió, del ultimo problema aceptado, la puntuación y la posición en usuarios, en la institución y en el país.	
Observaciones:	

Tabla 31: Historia de usuario No 11

Historia de usuario	
Número: 11	Nombre historia de usuario: Configurar reportes.
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Descripción: El administrador puede activar o desactivar los reportes para determinadas páginas en dependencia de la necesidad de los usuarios.	
Observaciones:	

Tabla 32: Historia de usuario No 12

Historia de usuario	
Número: 12	Nombre historia de usuario: Crear y exportar reporte de todos los problemas presentados en un concurso realizado.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Descripción: Permitirá la creación y exportación de un reporte de todos los problemas que contiene una competencia ya realizada, en él se obtendrán los datos de cada uno de los problemas como el nombre, autor, fecha en que fue adicionado, límites, descripción, especificación de entrada, especificación de salida, ejemplo de entrada y ejemplo de salida.	
Observaciones:	

Tabla 33: Historia de usuario No 13

Historia de usuario	
Número: 13	Nombre historia de usuario: Exportar reporte de los premios de un concurso.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 3	
Descripción: Permitirá crear un reporte que contenga los equipos que alcanzaron premios.	

Observaciones:

Tabla 34: Historia de usuario No 14

Historia de usuario	
Número: 14	Nombre historia de usuario: Exportar reporte de las estadísticas en los concursos reales anteriores.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 3	
Descripción: Permitirá exportar un reporte con las estadísticas de las concursos pasados.	
Observaciones:	

Tabla 35: Historia de usuario No 15

Historia de usuario	
Número: 15	Nombre historia de usuario: Exportar reporte de la comparación de usuarios en el archivo 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 3	
Descripción: Permitirá exportar un reporte de la comparación de dos usuarios seleccionados, obteniendo los usuarios, los problemas resueltos por cada uno, los	

problemas resueltos por ambos, los problemas intentados por cada uno y los problemas intentados por ambos.
Observaciones:

Tabla 36: Historia de usuario No 16

Historia de usuario	
Número: 16	Nombre historia de usuario: Exportar reporte de las sentencias del archivo 24 horas.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 3	
Descripción: Permitirá exportar un reporte con las sentencias del archivo 24 horas obteniendo las sentencias realizadas por un usuario seleccionado en un problema específico y sus respectivas propiedades.	
Observaciones:	

Tabla 37: Historia de usuario No 17

Historia de usuario	
Número: 17	Nombre historia de usuario: Exportar reporte de las sentencias de concursos del archivo concursos reales.
Usuario: Usuario anónimo, autenticado y administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 3	

Descripción: Permitirá exportar un reporte con las sentencias del archivo concursos reales obteniendo las sentencias realizadas en un concurso por un usuario seleccionado en un problema específico y sus respectivas propiedades.
Observaciones:

Anexo II

Tarjetas CRC:

Tabla 38: Tarjeta CRC ReporteDao

Tarjeta CRC : ReporteDao	
Clase: ReporteDAO	
Súper Clase: BaseDAO	
Sub Clases:	
Responsabilidades: Esta clase tiene el objetivo de declarar los principales métodos a utilizar.	**Colaboraciones:**

Tabla 39: Tarjeta CRC ReporteModel

Tarjeta CRC : ReporteModel	
Clase: Reporte	
Súper Clase:	
Sub Clases:	
Responsabilidades: Esta clase tiene el objetivo modelar los objetos de la base de datos en este caso los reportes.	**Colaboraciones:**

Tabla 40: Tarjeta CRC ReporteDaoImpl

Tarjeta CRC : ReporteDaoImpl	
Clase: ReporteDAOImpl	
Súper Clase: BaseDAOImpl	
Sub Clases:	
Responsabilidades: Esta clase tiene el objetivo de realizar las consultas pertinentes a la base de datos.	**Colaboraciones:** ReporteDao Reporte

Anexo III

Tareas de Ingeniería

Primera Iteración

Tabla 41: Tarea de ingeniería #1 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 2
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte de listado de problemas del archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 5/03/2015	Fecha fin: 8/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 42: Tarea de ingeniería #2 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 2
Nombre de la tarea: Exportar reporte del listado de problemas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 5/03/2015	Fecha fin: 8/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de los problemas que aparecen al acceder a la sesión de Problemas.	

Tabla 43: Tarea de ingeniería #3 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 2
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 5/03/2015	Fecha fin: 8/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 44: Tarea de ingeniería #1 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 3
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las posiciones de los usuarios en el archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5

Fecha inicio: 9/03/2015	Fecha fin: 12/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 45: Tarea de ingeniería #2 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 3
Nombre de la tarea: Exportar reporte de las posiciones de los usuarios en el archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 9/03/2015	Fecha fin: 12/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las posiciones de los primeros usuarios.	

Tabla 46: Tarea de ingeniería #3 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 3
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 9/02/2015	Fecha fin: 12/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 47: Tarea de ingeniería #1 de la HU #4.

Tarea de ingeniería

No. de la tarea: 1	No. de la HU: 4
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las posiciones de las instituciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 13/03/2015	Fecha fin: 16/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 48: Tarea de ingeniería #2 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 4
Nombre de la tarea: Exportar reporte de las posiciones de las instituciones en el archivo 24 horas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 13/03/2015	Fecha fin: 16/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las posiciones de los primeras instituciones.	

Tabla 49: Tarea de ingeniería #3 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 4
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 13/03/2015	Fecha fin: 16/03/2015

Programador responsable: Jorge Gabriel Díaz Reinoso.
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.

Tabla 50: Tarea de ingeniería #1 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 5
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las posiciones de los países.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 17/03/2015	Fecha fin: 20/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 51: Tarea de ingeniería #2 de la HU #5.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 5
Nombre de la tarea: Exportar reporte de las posiciones de los países en el archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 17/03/2015	Fecha fin: 20/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las posiciones de los países en el archivo 24 horas.	

Tabla 52: Tarea de ingeniería #3 de la HU #5.

Tarea de ingeniería

No. de la tarea: 3	No. de la HU: 5
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 17/03/2015	Fecha fin: 20/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Segunda Iteración

Tabla 53: Tarea de ingeniería #1 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 1
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las estadísticas del archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 21/03/2015	Fecha fin: 24/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 54: Tarea de ingeniería #2 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 1
Nombre de la tarea: Cargar gráficas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5

Fecha inicio: 21/03/2015	Fecha fin: 24/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Obtener las gráficas de las estadísticas del archivo 24 horas e insertarlas en el PDF.	

Tabla 55: Tarea de ingeniería #3 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 1
Nombre de la tarea: Exportar reporte de las estadísticas en el archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 221/03/2015	Fecha fin: 24/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las estadísticas en el archivo 24 horas.	

Tabla 56: Tarea de ingeniería #4 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 4	No. de la HU: 1
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 21/03/2015	Fecha fin: 24/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 57: Tarea de ingeniería #1 de la HU #2.

Tarea de ingeniería

No. de la tarea: 1	No. de la HU: 2
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las posiciones de los usuarios en los concursos reales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 25/03/2015	Fecha fin: 29/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 58: Tarea de ingeniería #2 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 2
Nombre de la tarea: Exportar reporte de las posiciones de los usuarios en los concursos reales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 25/03/2015	Fecha fin: 29/03/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las posiciones de los primeros usuarios en los concursos reales.	

Tabla 59: Tarea de ingeniería #3 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 2
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 25/03/2015	Fecha fin: 29/03/2015

Programador responsable: Jorge Gabriel Díaz Reinoso.
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.

Tabla 60: Tarea de ingeniería #1 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 3
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las estadísticas de los concursos reales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 30/03/2015	Fecha fin: 5/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 61: Tarea de ingeniería #2 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 3
Nombre de la tarea: Cargar gráficas de las estadísticas de los concursos reales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 30/03/2015	Fecha fin: 5/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Obtener las gráficas de las estadísticas de los concursos reales e insertarlas en el PDF.	

Tabla 62: Tarea de ingeniería #3 de la HU #3.

Tarea de ingeniería

No. de la tarea: 3	No. de la HU: 3
Nombre de la tarea: Exportar reporte de las estadísticas de los concursos reales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 30/03/2015	Fecha fin: 5/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las estadísticas de los concursos reales	

Tabla 63: Tarea de ingeniería #4 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 4	No. de la HU: 3
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 30/03/2015	Fecha fin: 9/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 64: Tarea de ingeniería #1 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 4
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las posiciones de los usuarios en los concursos anteriores.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10/04/2015	Fecha fin: 16/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	

Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.

Tabla 65: Tarea de ingeniería #2 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 4
Nombre de la tarea: Exportar reporte de las posiciones en los concursos anteriores.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10/04/2015	Fecha fin: 16/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las posiciones en los concursos anteriores.	

Tabla 66: Tarea de ingeniería #3 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 4
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10/04/2015	Fecha fin: 16/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 67: Tarea de ingeniería #1 de la HU #5.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 5
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene el perfil de usuario.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 17/04/2015	Fecha fin: 21/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 68: Tarea de ingeniería #2 de la HU #5.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 5
Nombre de la tarea: Exportar reporte del perfil de usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 17/04/2015	Fecha fin: 21/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar la información del perfil de usuario.	

Tabla 69: Tarea de ingeniería #3 de la HU #5.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 5
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 17/04/2015	Fecha fin: 21/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tercera Iteración

Tabla 70: Tarea de ingeniería #1 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 1
Nombre de la tarea: Implementación de las pruebas unitarias para la accesibilidad de los reportes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 22/04/2015	Fecha fin: 28/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 71: Tarea de ingeniería #2 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 1
Nombre de la tarea: Configurar reportes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 22/04/2015	Fecha fin: 28/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Crear formulario de accesibilidad para visualizar las descargas del sitio que se pretendan mostrar.	

Tabla 72: Tarea de ingeniería #3 de la HU #1.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 1
Nombre de la tarea: Ejecución de las pruebas unitarias.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 22/04/2015	Fecha fin: 28/04/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 73: Tarea de ingeniería #1 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 2
Nombre de la tarea: Implementación de las pruebas unitarias para la creación y exportación del reporte que contiene el resumen de problemas de un concurso seleccionado.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 29/04/2015	Fecha fin: 5/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 74: Tarea de ingeniería #2 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 2
Nombre de la tarea: Crear y exportar reporte de todos los problemas presentados en un concurso ya realizado.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 29/04/2015	Fecha fin: 5/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	

Descripción: Obtener la información de los problemas de la base de datos y utilizar AJAX para llamarla desde los jsp para su posterior descarga.

Tabla 75: Tarea de ingeniería #3 de la HU #2.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 2
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 29/04/2015	Fecha fin: 5/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 76: Tarea de ingeniería #1 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 3
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene los premios obtenidos en un concurso.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 6/05/2015	Fecha fin: 10/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 77: Tarea de ingeniería #2 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 3

Nombre de la tarea: Exportar reporte de los premios de un concurso.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 6/05/2015	Fecha fin: 10/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar los resultados de las premiaciones en una competencia.	

Tabla 78: Tarea de ingeniería #3 de la HU #3.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 3
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 6/05/2015	Fecha fin: 10/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 79: Tarea de ingeniería #1 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 4
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las estadísticas de los concursos reales anteriores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 11/05/2015	Fecha fin: 15/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	

Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.

Tabla 80: Tarea de ingeniería #2 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 4
Nombre de la tarea: Exportar reporte de las estadísticas en los concursos reales anteriores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 11/05/2015	Fecha fin: 15/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar los resultados de las estadísticas de los concursos reales anteriores.	

Tabla 81: Tarea de ingeniería #3 de la HU #4.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 4
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 11/05/2015	Fecha fin: 15/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 82: Tarea de ingeniería #1 de la HU #5.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 5

Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene la comparación de dos usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/05/2015	Fecha fin: 23/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 83: Tarea de ingeniería #2 de la HU #5.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 5
Nombre de la tarea: Exportar reporte de la comparación de usuarios en el archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/05/2015	Fecha fin: 23/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar los resultados de la comparación de dos usuarios seleccionados en el archivo 24 horas.	

Tabla 84: Tarea de ingeniería #3 de la HU #5.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 5
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/05/2015	Fecha fin: 23/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	

Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.

Tabla 85: Tarea de ingeniería #1 de la HU #6.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 6
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las sentencias del archivo 24 horas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 24/05/2015	Fecha fin: 28/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 86: Tarea de ingeniería #2 de la HU #6.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 6
Nombre de la tarea: Exportar reporte de las sentencias del archivo 24 horas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 24/05/2015	Fecha fin: 28/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Exportar reporte de las sentencias del archivo 24 horas.	

Tabla 87: Tarea de ingeniería #3 de la HU #6.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 6
Nombre de la tarea: Ejecución de las pruebas unitarias.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 24/05/2015	Fecha fin: 28/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Tabla 88: Tarea de ingeniería #1 de la HU #7.

Tarea de ingeniería	
No. de la tarea: 1	No. de la HU: 7
Nombre de la tarea: Implementación de las pruebas unitarias para la exportación del reporte que contiene las sentencias de los concursos reales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 29/05/2015	Fecha fin: 31/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Elaborar las pruebas unitarias a partir de la implementación de la HU para probar su funcionamiento.	

Tabla 89: Tarea de ingeniería #2 de la HU #7.

Tarea de ingeniería	
No. de la tarea: 2	No. de la HU: 7
Nombre de la tarea: Exportar reporte de las sentencias en los concursos reales anteriores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 29/05/2015	Fecha fin: 31/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	

Descripción: Exportar reporte de las sentencias de concursos del archivo concursos reales.

Tabla 90: Tarea de ingeniería #3 de la HU #7.

Tarea de ingeniería	
No. de la tarea: 3	No. de la HU: 7
Nombre de la tarea: Ejecución de las pruebas unitarias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 29/05/2015	Fecha fin: 31/05/2015
Programador responsable: Jorge Gabriel Díaz Reinoso.	
Descripción: Realizar las pruebas unitarias para corregir las diferentes deficiencias existentes en la implementación de la solución.	

Anexo IV

Prueba unitaria con JUnit

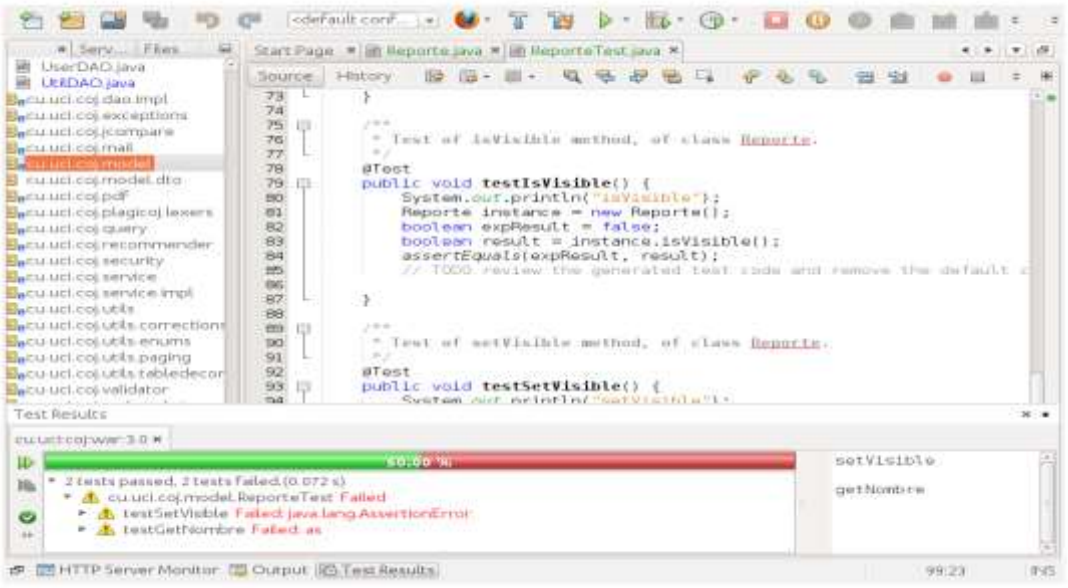

Figura 9: Prueba unitaria con JUnit clase Reporte

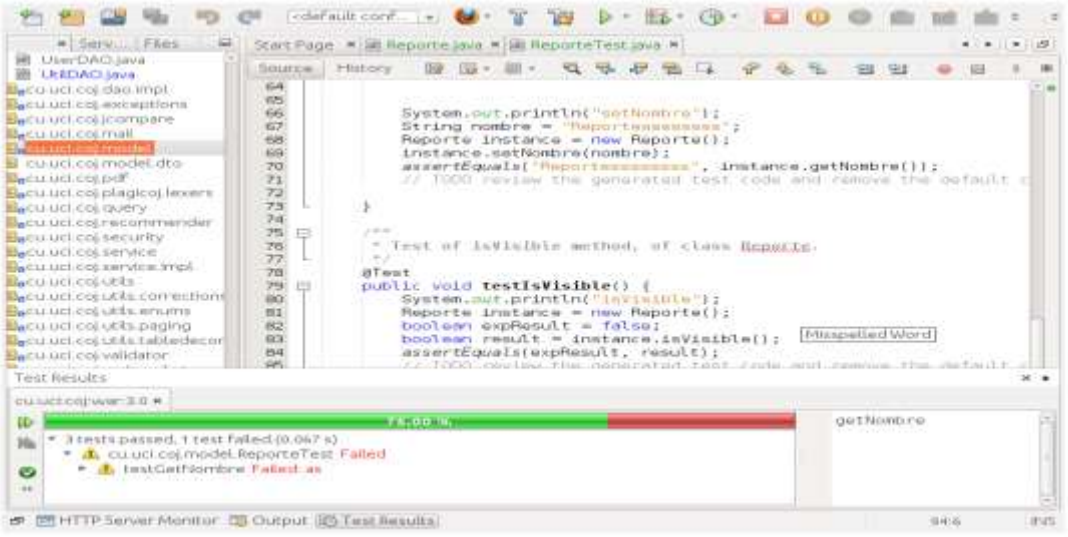

Figura 10: Prueba unitaria con JUnit clase Reporte

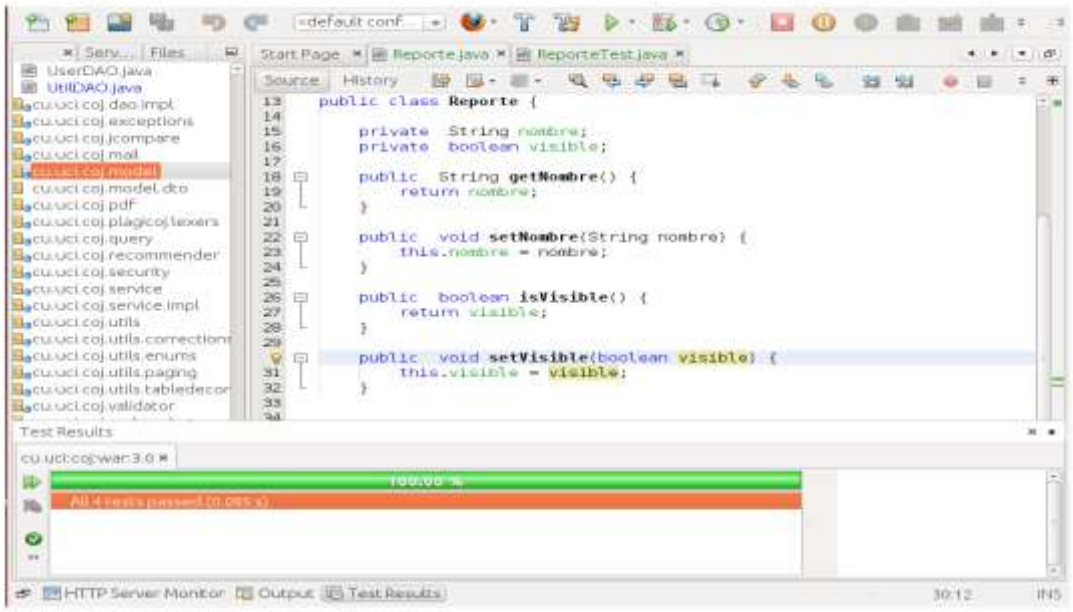

Figura 11: Prueba unitaria con JUnit clase Reporte

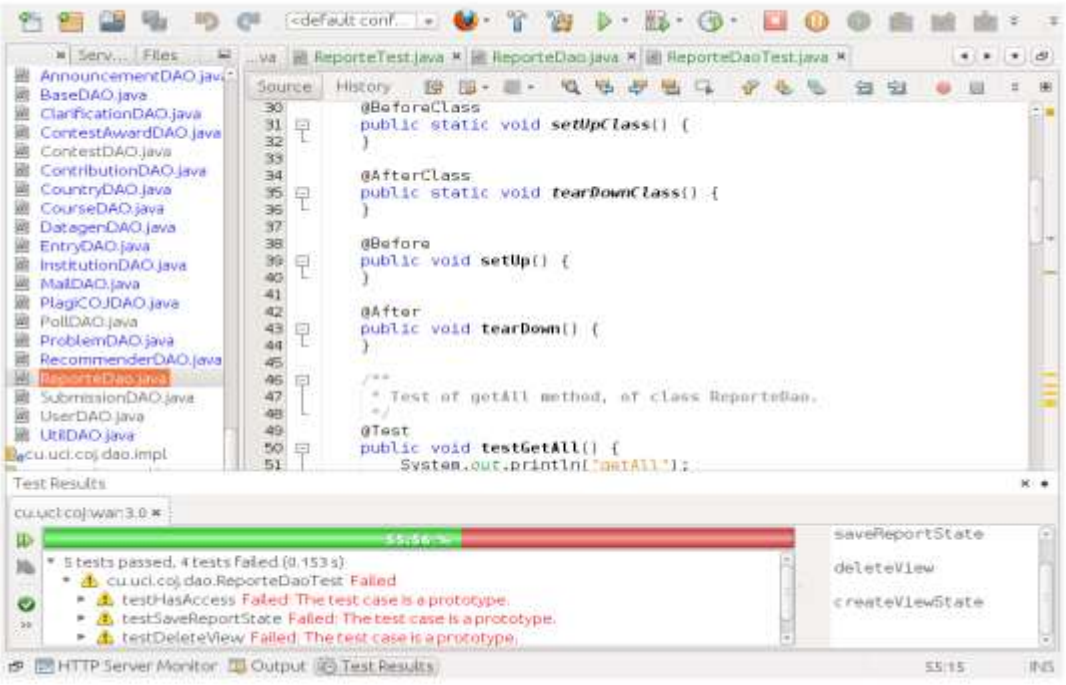

Figura 12: Prueba unitaria con JUnit clase ReporteDao

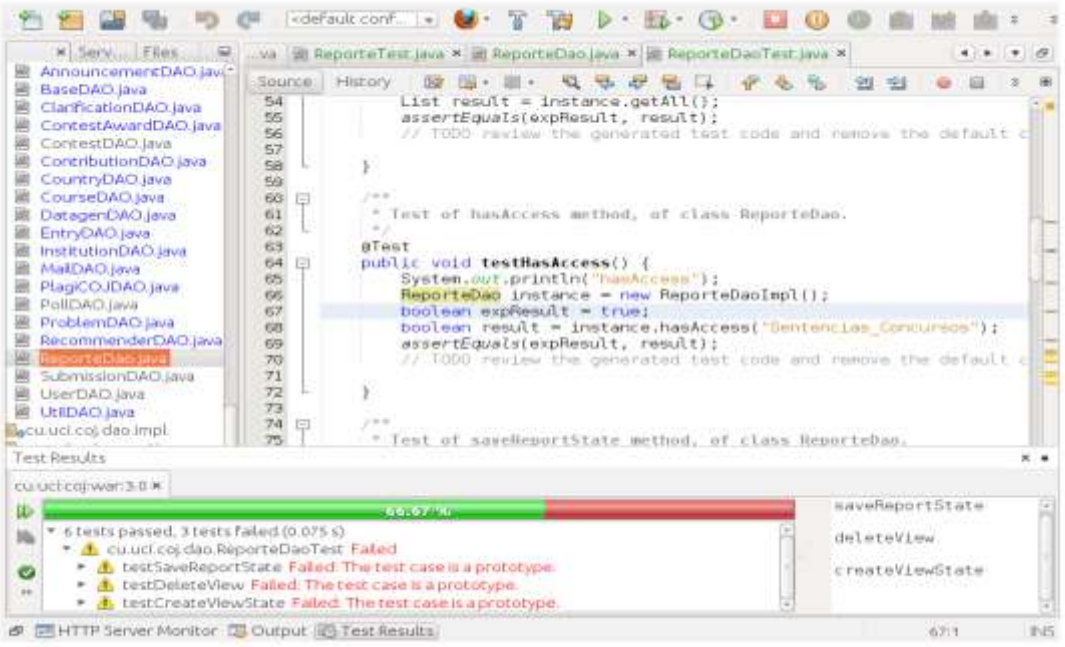

Figura 13: Prueba unitaria con JUnit clase ReporteDao

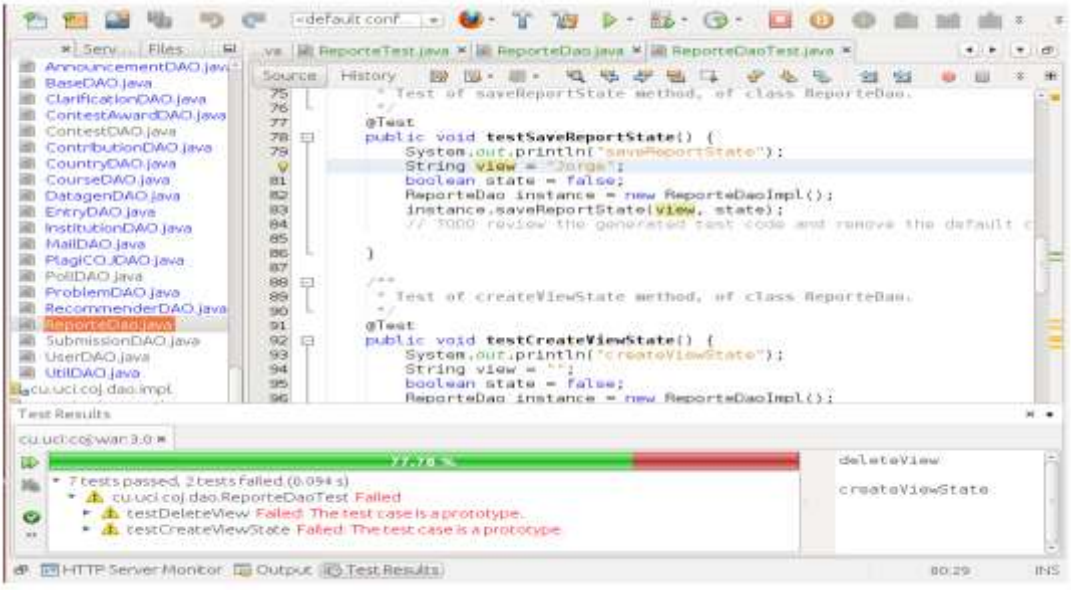

Figura 13: Prueba unitaria con JUnit clase ReporteDao

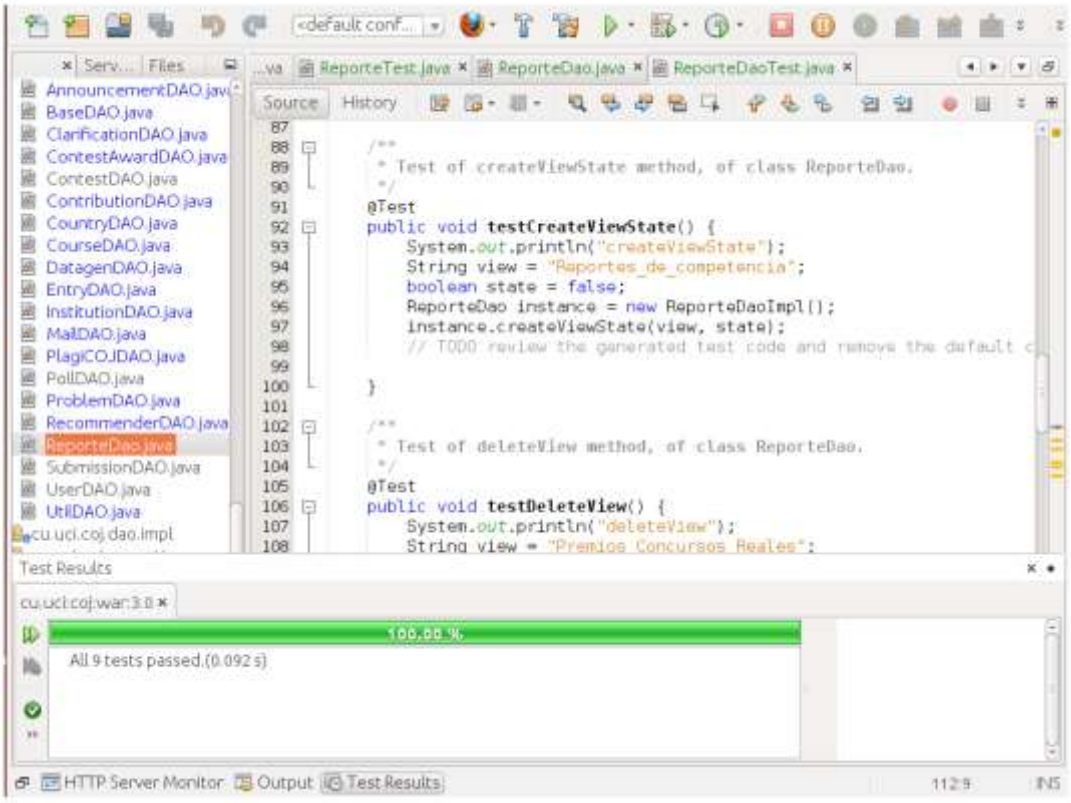

Figura 14: Prueba unitaria con JUnit clase ReporteDao

Prueba unitaria con QUnit

Figura 15: Test del concurso The 2010 Caribbean Local Contests of the ACM-ICPC (UTECH)

Figura 16: Test del concurso The 2010 Caribbean Local Contests of the ACM-ICPC (UTECH)

Figura 17: Test del concurso The 2010 Caribbean Local Contests of the ACM-ICPC (UTECH)

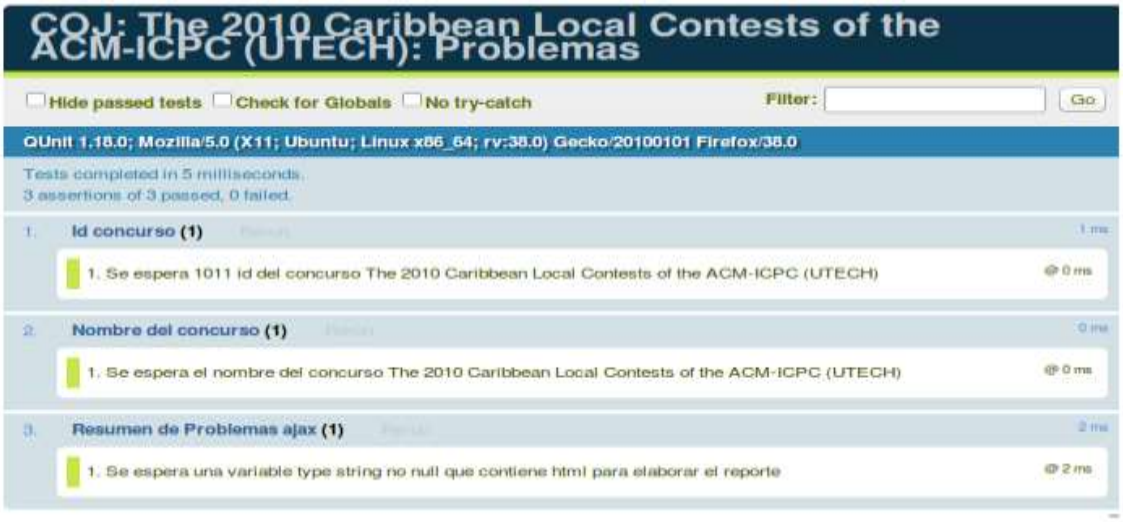

Figura 18: Test del concurso The 2010 Caribbean Local Contests of the ACM-ICPC (UTECH)

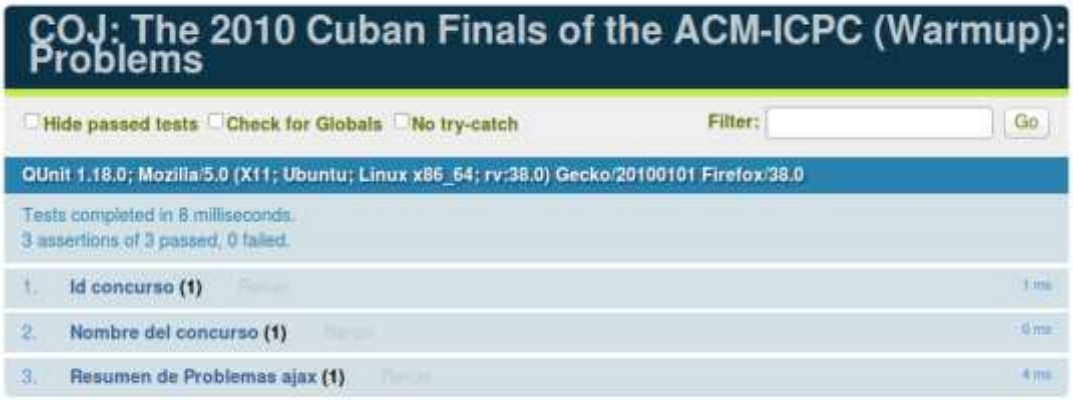

Figura 19: Test del concurso The 2010 Cuban Finals of the ACM-ICPC (Warmup)

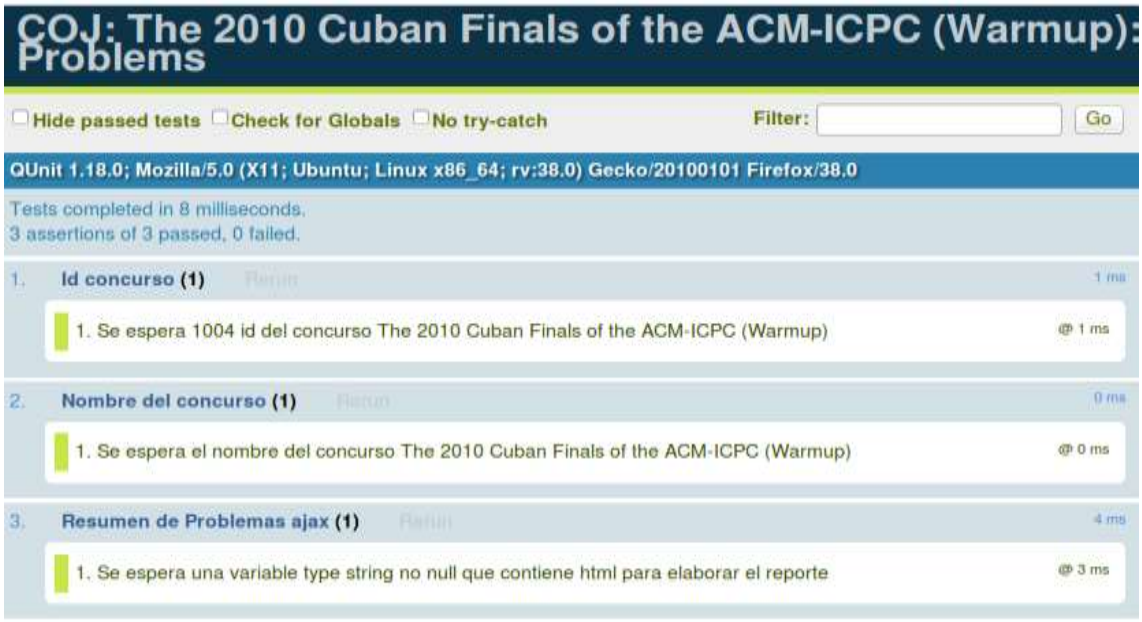

Figura 20: Test del concurso The 2010 Cuban Finals of the ACM-ICPC (Warmup)

Figura 21: Test de las estadísticas del archivo 24 horas

Pruebas de carga

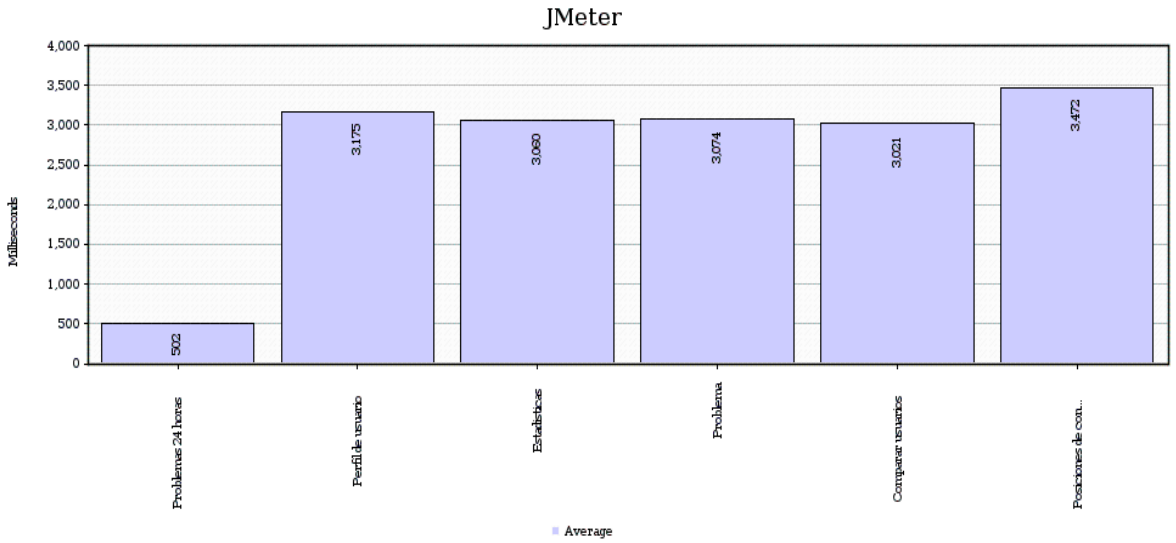

Figura 22: Average

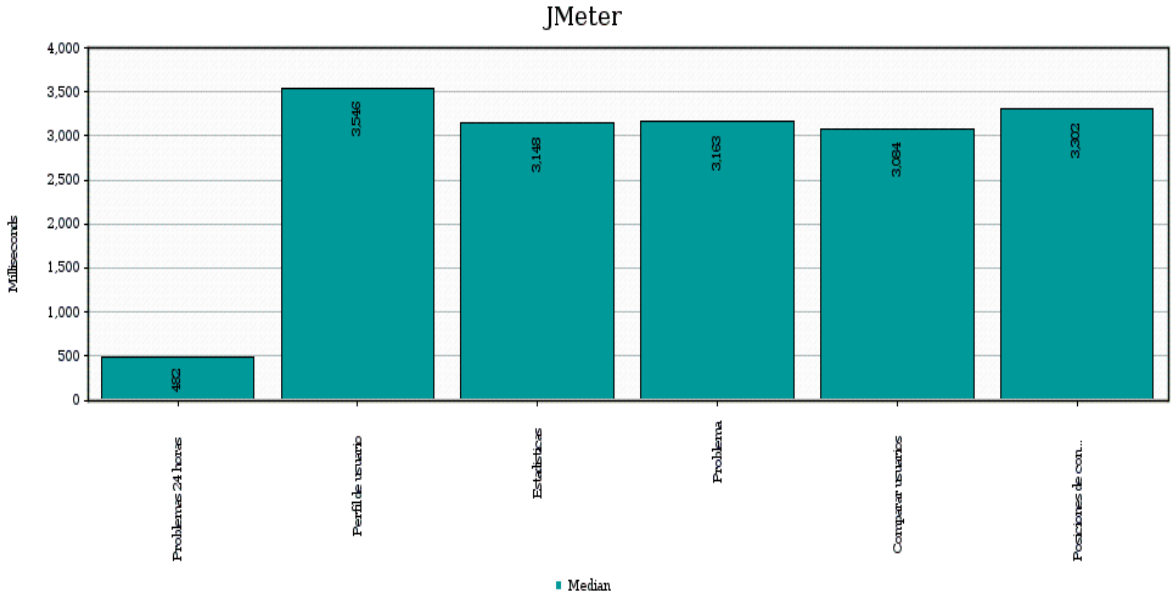

Figura 23: Tiempo promedio de respuesta

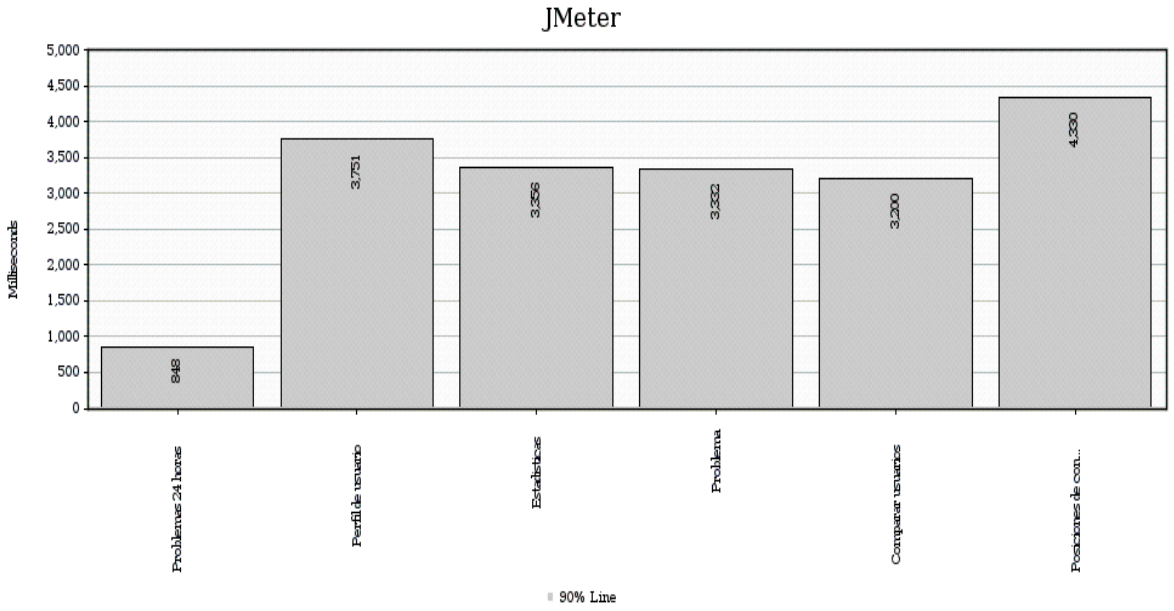

Figura 24: Línea del 90% de las peticiones

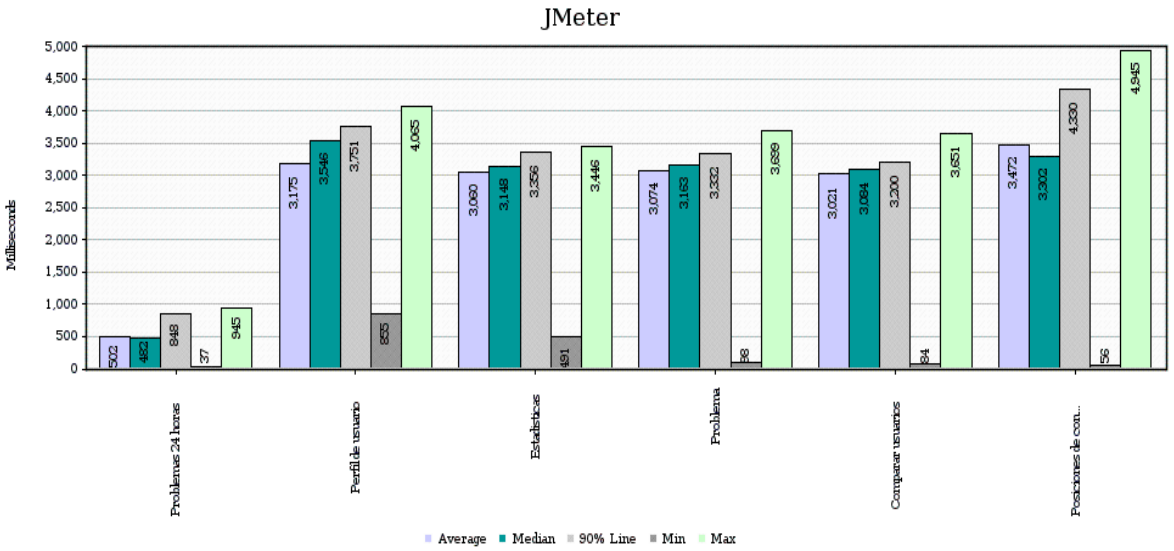

Figura: 25 Datos obtenidos

Pruebas de aceptación

Tabla 91: Prueba de Aceptación HU #2

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_2	Nombre historia de usuario: 2
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte del listado de problemas del Archivo 24 horas.	
Condiciones de Ejecución: Debe de existir al menos un problema en el archivo para que sea listado y exportado.	
Entrada / Pasos de ejecución: Se accede al Archivo 24 horas al módulo de Problemas y se selecciona un problema para después efectuar la opción de exportar PDF.	
Resultado Esperado: Se muestra en formato PDF el listado de problemas existentes.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 92: Prueba de Aceptación HU #3

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_3	Nombre historia de usuario: 3
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de las posiciones de los usuarios en el Archivo 24 horas.	
Condiciones de Ejecución: Debe de existir al menos un usuario autentificado.	
Entrada / Pasos de ejecución: Se debe ubicar en el Archivo 24 horas en las posiciones de usuarios y seleccionar la opción PDF.	
Resultado Esperado: Se muestra en formato PDF el listado de las posiciones de los usuarios.	

Evaluación de la Prueba: Prueba satisfactoria.

Tabla 93: Prueba de Aceptación HU #4

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_4	Nombre historia de usuario: 4
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de las Posiciones de instituciones en el Archivo 24 horas.	
Condiciones de Ejecución: Debe de existir al menos una institución registrada en el sistema.	
Entrada / Pasos de ejecución: Se debe ubicar en el Archivo 24 horas en las Posiciones de las Instituciones y seleccionar la opción PDF.	
Resultado Esperado: Se muestra en formato PDF el listado de las posiciones de las instituciones.	
Evaluación de la Prueba: Prueba satisfactoria	

Tabla 94: Prueba de Aceptación HU #5

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_5	Nombre historia de usuario: 5
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de las Posiciones de los Países en el Archivo 24 horas.	
Condiciones de Ejecución: Debe de existir al menos un país registrado en el sistema.	
Entrada / Pasos de ejecución: Se accede al Archivo 24 horas en las Posiciones de los Países y seleccionar la opción PDF.	
Resultado Esperado: Se muestra en formato PDF el listado de las posiciones de los países.	

Evaluación de la Prueba: Prueba satisfactoria

Tabla 95: Prueba de Aceptación HU #6

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_6	Nombre historia de usuario: 6
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de las estadísticas en el Archivo 24 horas.	
Condiciones de Ejecución: Debe existir el reporte de estadísticas.	
Entrada / Pasos de ejecución: Se accede al Archivo 24 horas en Estadísticas y luego se selecciona la opción PDF.	
Resultado Esperado: Se muestra en formato PDF las estadísticas del Archivo 24 horas.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 96: Prueba de Aceptación HU #7

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_7	Nombre historia de usuario: 7
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de las posiciones de los concursantes en los Concursos reales.	
Condiciones de Ejecución: Debe de estar registrado al menos un concursante.	
Entrada / Pasos de ejecución: Se accede a los Concursos reales en Posiciones y luego se selecciona la opción PDF.	
Resultado Esperado: Se muestra en formato PDF las posiciones de las concursantes.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 97: Prueba de Aceptación HU #8

Casos de Prueba de Aceptación

Código Caso de Prueba: CPA_8	Nombre historia de usuario: 8
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de las estadísticas de los Concursos reales.	
Condiciones de Ejecución: Debe existir el reporte de estadísticas.	
Entrada / Pasos de ejecución: Se accede a los Concursos reales en Estadísticas y luego se selecciona la opción PDF.	
Resultado Esperado: Se muestra en formato PDF las estadísticas de los Concursos reales.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 98: Prueba de Aceptación HU #9

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_9	Nombre historia de usuario: 9
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar un reporte con las posiciones de los concursantes en los diferentes concursos realizados con anterioridad, obteniendo los lugares, el tiempo empleado en cada problema, la cantidad de problemas aceptados, los envíos que fueron aceptados, rechazados y pendientes.	
Condiciones de Ejecución: Debe de existir concursos anteriores.	
Entrada / Pasos de ejecución: Se accede a los Concursos reales en Anteriores, se accede a un concurso, se elige el apartado posiciones y se selecciona la opción PDF.	
Resultado Esperado: Se muestra en formato PDF las posiciones obtenidas en concursos reales anteriores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 99: Prueba de Aceptación HU #10

Casos de Prueba de Aceptación

Código Caso de Prueba: CPA_10	Nombre historia de usuario: 10
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte del perfil de usuarios.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada / Pasos de ejecución: Se accede al módulo de usuario en Ver perfil y se selecciona la opción PDF.	
Resultado Esperado: Se muestra en formato PDF el perfil del usuario.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 100: Prueba de Aceptación HU #11

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_11	Nombre historia de usuario: 11
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Configurar reportes	
Condiciones de Ejecución: El Administrador debe autenticarse.	
Entrada / Pasos de ejecución: Se accede al módulo Administración y en el apartado reportes y selecciona los reportes que se desean exportar.	
Resultado Esperado: Se actualiza en el sistema los reportes que se desean exportar.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 101: Prueba de Aceptación HU #12

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_12	Nombre historia de usuario: 12

Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)
Descripción de la Prueba: Crear y exportar reporte de todos los problemas presentados en un concurso realizado.
Condiciones de Ejecución: Debe existir un concurso con al menos un problema.
Entrada / Pasos de ejecución: Se accede a los Concursos reales en Anteriores y se selecciona un concurso luego se selecciona el apartado problemas y se selecciona la opción de PDF.
Resultado Esperado: Se obtiene un reporte con todos los problemas presentados en dicha competencia y sus respectivas descripciones.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 102: Prueba de Aceptación HU #13

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_13	Nombre historia de usuario: 13
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de los premios de un concurso.	
Condiciones de Ejecución: Debe de haberse realizado un concurso.	
Entrada / Pasos de ejecución: Se accede a los Concursos reales en Anteriores y se selecciona un concurso luego se selecciona el apartado premios y se selecciona la opción de PDF.	
Resultado Esperado: Se obtiene un reporte con los equipos o usuarios que obtuvieron los tres primeros lugares.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 103: Prueba de Aceptación HU #14

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_14	Nombre historia de usuario: 14
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar reporte de las estadísticas en los concursos reales anteriores.	
Condiciones de Ejecución: Debe de haberse realizado un concurso y un reporte con sus respectivas estadísticas.	
Entrada / Pasos de ejecución: Se accede a los Concursos reales en Anteriores y se selecciona un concurso luego se selecciona el apartado estadísticas y se selecciona la opción de PDF.	
Resultado Esperado: Se obtiene un reporte con las estadísticas del concurso seleccionado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 104: Prueba de Aceptación HU #15

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_15	Nombre historia de usuario: 15
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar un reporte de la comparación de dos usuarios seleccionados, obteniendo los usuarios, los problemas resueltos por cada uno, los problemas resueltos por ambos, los problemas intentados por cada uno y los problemas intentados por ambos.	
Condiciones de Ejecución: Debe de haberse realizado la comparación de dos usuarios.	

Entrada / Pasos de ejecución: Se accede a al archivo 24 horas, se elige comparar usuarios, se escriben los nombres de los dos usuarios a comparar y se selecciona comparar, terminada la comparación selecciona la opción de PDF.
Resultado Esperado: Se obtiene un reporte con los datos brindados de la comparación.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 105: Prueba de Aceptación HU #16

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_16	Nombre historia de usuario: 16
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	
Descripción de la Prueba: Exportar un reporte con las sentencias del archivo 24 horas obteniendo las sentencias realizadas por un usuario seleccionado en un problema específico y sus respectivas propiedades.	
Condiciones de Ejecución: Debe de haberse realizado un reporte de las sentencias del archivo 24 horas.	
Entrada / Pasos de ejecución: Se accede al archivo 24 horas y se selecciona el apartado sentencias y se selecciona la opción de PDF.	
Resultado Esperado: Se obtiene un reporte con las sentencias presentes en la página.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 106: Prueba de Aceptación HU #17

Casos de Prueba de Aceptación	
Código Caso de Prueba: CPA_17	Nombre historia de usuario: 17
Nombre de la persona que realiza la prueba: Programador (Jorge Gabriel Díaz Reinoso)	

Descripción de la Prueba: Exportar un reporte con las sentencias del archivo concursos reales obteniendo las sentencias realizadas en un concurso por un usuario seleccionado en un problema específico y sus respectivas propiedades.
Condiciones de Ejecución: Debe de haberse realizado un reporte con las sentencias del archivo concursos reales.
Entrada / Pasos de ejecución: Se accede a los Concursos reales en Anteriores y se selecciona un concurso, luego se selecciona el apartado sentencias y se selecciona la opción de PDF.
Resultado Esperado: Se obtiene un reporte con las sentencias del concurso seleccionado.
Evaluación de la Prueba: Prueba satisfactoria.