

Universidad de las Ciencias Informáticas

Facultad 2



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

CMFMail: Sistema automático de respuesta por correo electrónico del Caribbean Mind Forge.

Autor: Eduardo Oscar Pereira Dennes

Tutor: MSc. Madelis Pérez Gil

Co-Tutor: Ing. Andy Eloy Yero Guevara

"No hay secretos para el éxito. Éste se alcanza preparándose, trabajando arduamente y aprendiendo del fracaso."

Colin Powell

Declaración de Autoría

Declaro ser el legítimo autor del trabajo titulado: “**CMFMail: Sistema automático de respuesta por correo electrónico del Caribbean Mind Forge.**”, y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autor

Eduardo Oscar Pereira Dennes

Tutor

MSc. Madelis Pérez Gil

Resumen

La informatización cubana avanza aceleradamente dando paso a nuevas áreas de acceso a internet. El servicio de correo electrónico brindado por la empresa de telecomunicaciones de Cuba (ETECSA) ha tomado auge y se expande por la población. Siendo este un medio de acceso a “Los jueces en línea” de agilidad mental. Estos son soluciones web creadas con el objetivo de desarrollar habilidades mentales a través de juegos de razonamiento, lógica y memoria. Dichas soluciones web no son aprovechadas al máximo debido a intermitencias en la conexión a Internet además del alto costo del mismo.

La Fragua Caribeña de la Mente o Caribbean Mind Forge (CMF), es un jurado en línea de agilidad mental el cual se encuentra publicado en Internet desde noviembre de 2013. Actualmente es accedido por usuarios de las diferentes universidades del país y centros de trabajo con acceso a Internet, permaneciendo desconocido por usuarios que no tienen acceso a internet. Debido a esto se hace necesario desarrollar un sistema capaz de interactuar con el jurado en línea a través de correo electrónico, aun en condiciones de acceso a Internet desfavorables.

El resultado de esta investigación es una aplicación informática para el servidor de correo de la Universidad de las Ciencias Informáticas que posibilita extender los servicios del Caribbean Mind Forge permitiendo el acceso al mismo desde cualquier cliente de correo electrónico. Este software permite satisfacer a todas aquellas personas que están ansiosas por participar en el CMF y que no cuentan con intranet o internet y sí con un correo electrónico.

Palabras clave: CMFMail, CMF, correo electrónico.

Introducción.....	9
Capítulo.1 Fundamentación Teórica	14
1.1 Introducción.....	14
1.2 Análisis de soluciones similares	14
1.2.1 Juez en línea de la Universidad de Valladolid (UVA)	14
1.2.2 Sphere Online Judge (SPOJ).....	15
1.2.3 Caribbean Online Judge (COJ).....	15
1.3 Conclusiones del análisis de sistemas similares	16
1.4 Análisis de Protocolos de Correo Electrónico	17
1.4.1 Protocolo de transporte de correo SMTP	17
1.4.2 Protocolos de acceso a correo electrónico POP e IMAP.....	17
1.5 Seguridad en los protocolos de correo electrónico	18
1.6 Conclusiones del Análisis de los Protocolos de Correo	19
1.7 Metodología, Herramientas, Tecnología y Lenguajes de desarrollo.	19
1.7.1 Metodología.....	19
1.7.2 Herramientas	20
1.7.3 Tecnología.....	21
1.7.4 Lenguajes de desarrollo.....	22
1.8 Conclusiones.....	24
Capítulo.2 Exploración, Planificación y Diseño del Sistema.....	25
2.1 Introducción.....	25
2.2 Propuesta de solución.....	25
2.3 Funcionalidades	27
2.4 Características no funcionales.....	28
2.5 Micro lenguaje.....	29
2.6 Exploración	31
2.6.1 Historias de usuario	31
2.7 Planificación	33
2.7.1 Estimación de Esfuerzo	33
2.8 Iteraciones.....	34
2.9 Plan de entregas	35
2.10 Diseño.....	36
2.11 Patrón arquitectónico	36

Índice

2.11.1	Patrones de Diseño	36
2.12	Modelo de datos.....	37
2.13	Tarjetas CRC	39
2.14	Conclusiones.....	45
Capítulo.3 Implementación y Pruebas.....		46
3.1	Introducción.....	46
3.2	Implementación.....	46
3.2.1	Tareas de implementación.....	46
3.3	Pruebas.....	48
3.3.1	Pruebas unitarias.....	48
3.3.2	Resultados.....	49
3.3.3	Pruebas de aceptación	54
3.4	Conclusiones.....	63
Conclusiones.....		64
Recomendaciones.....		65
Bibliografía		66

ÍNDICE DE TABLAS

Tabla 1: Protocolos de correo electrónico sobre SSL/TLS.....	19
Tabla 2: Descripción del Micro lenguaje a usar para acceder a los servicios del CMF	30
Tabla 3: Plantilla de Historias de Usuario.....	31
Tabla 4: Historia de Usuario Aplicación básica.	32
Tabla 5: Historia de Usuario Leer un problema	32
Tabla 6: Historia de Usuario Comentarios.....	32
Tabla 7: Velocidad del Proyecto.....	33
Tabla 8: Plan de entregas	35
Tabla 9: Plantilla de Tarjetas CRC	39
Tabla 10: Tarjeta CRC LeerCorreos.....	40
Tabla 11: Tarjeta CRC TareaAutomatica	40
Tabla 12: Tarjeta CRC Validar	41
Tabla 13: Tarjeta CRC Operaciones	41
Tabla 14: Tarjeta CRC Email.....	41
Tabla 15: Tarjeta CRC Especial	42
Tabla 16: Tarjeta CRC Vistas.....	42
Tabla 17: Tarjeta CRC Instalador CMFMail	42
Tabla 18: Tarjeta CRC Instalador1	43
Tabla 19: Tarjeta CRC CopiarDirectorio.....	43
Tabla 20: Tarjeta CRC Instalador2.....	43
Tabla 21: Tarjeta CRC FormulaCMF.....	44
Tabla 22: Tarjeta CRC Formulario2	44
Tabla 23: Tarjeta CRC PreguntaDia.....	44
Tabla 24: Plantilla de las Tareas de implementación	46

Índice

Tabla 25: Tarea de implementación Configurar cuenta de correo.....	47
Tabla 26: Tarea de implementación Crear servicios.	47
Tabla 27: Plantilla Pruebas de Aceptación.....	55
Tabla 28: Prueba de aceptación Registrar usuario	55
Tabla 29: Prueba de aceptación Rango de problemas	56
Tabla 30: Prueba de aceptación Responder problema	56
Tabla 31: Prueba de aceptación Leer un problema.....	57
Tabla 32: Prueba de aceptación Ver Comentarios.....	57
Tabla 33: Prueba de aceptación Ranking	57
Tabla 34: Prueba de aceptación Posición en el ranking.....	58
Tabla 35: Prueba de aceptación Últimas sentencias.....	58
Tabla 36: Prueba de aceptación Sentencias	59
Tabla 37: Prueba de aceptación Manual del CMF	60
Tabla 38: Prueba de aceptación Configurar cuenta de correo	60
Tabla 39: Prueba de aceptación Configurar servidor de servicios	61
Tabla 40: Prueba de aceptación Configurar servidor de envío de correo	61
Tabla 41: Prueba de aceptación Configurar servidor de descarga de correo	61
Tabla 42: Prueba de aceptación Configurar puerto de descarga de correo	62
Tabla 43: Prueba de aceptación Configurar puerto de envío de correo	62

Figura 1: Estadísticas de CMF del 25 de enero de 2017	10
Figura 2: Propuesta de solución.....	25
Figura 3: Diagrama Entidad- Relación	38
Figura 4: Resultado de la prueba unitaria a la clase Vista.	49
Figura 5: Resultado corregido de la prueba unitaria a la clase Vista.....	49
Figura 6: Resultado de la prueba unitaria a la clase Validar.	50
Figura 7: Resultado corregido de la prueba unitaria a la clase Validar.	50
Figura 8: Resultado de la prueba unitaria a la clase Vistas.....	51
Figura 9: Resultado corregido de la prueba unitaria a la clase Vistas.....	52
Figura 10: Resultado de la prueba unitaria a la clase Validar.	52
Figura 11: Resultado de la prueba unitaria a la clase Validar.	53
Figura 12: Resultado de la prueba unitaria a la clase Validar.	53
Figura 13: Resultado corregido de la prueba unitaria a la clase Validar.	53
Figura 14: Resultado de la prueba unitaria a la clase Vistas.....	54
Figura 15: Resultado corregido de la prueba unitaria a la clase Vistas.....	54

Introducción

En el mundo existe una gran cantidad de aplicaciones y páginas web para la práctica y desarrollo de habilidades mentales, tales como rapidez de cálculo, memoria fotográfica, memoria a corto y largo plazo. Todo esto favorece el mantenimiento en forma del cerebro, para realizar cualquier actividad, ya sea académica, laboral, cotidiana o de cualquier otra envergadura.

Los Jueces o Jurados en línea de agilidad mental son aplicaciones web creadas principalmente para entornos académicos. Proveen problemas de variadas complejidades, y a su vez, evalúan de forma automática las soluciones que los usuarios encuentran a dichos problemas. Son también empleados no solo por jóvenes estudiantes pertenecientes a cualquier enseñanza sino también por toda persona que desea ampliar su sabiduría y entendimiento. En los últimos años se ha potenciado el uso de los jueces en línea de agilidad mental. Siendo usados por estudiantes y entrenadores de concursos. Teniendo como propósito lograr buenos resultados en eventos para el desarrollo de la mente. Dichos eventos fueron desarrollados con el objetivo de fomentar conocimientos en diversas ramas de las ciencias (matemáticas, probabilidades, estadísticas, lógica, álgebra, etc.) y lograr en la sociedad un expandir intelectual.

En Cuba, específicamente en la Universidad de las Ciencias Informáticas se ha desarrollado un jurado en línea de agilidad mental denominado Fragua Caribeña de la Mente (Caribbean Mind Forge, CMF), realizado por estudiantes. Se encuentra publicado en Internet desde noviembre de 2013. Actualmente es accedido por usuarios de las universidades del país y centros de trabajo con acceso a Internet. El propósito de esta modalidad es promover el desarrollo del conocimiento de manera recreativa y competitiva de las siguientes materias:

1. Memoria fotográfica
2. Cálculo de fechas
3. Operaciones matemáticas
4. Tecleo rápido
5. Sudoku
6. Resolución de problemas de agilidad mental, lógica, matemática y cultura general
7. Geografía visual

En varias instituciones cubanas resulta difícil tener pleno acceso al CMF. La causa es que la disponibilidad de la red de Internet es afectada por el poco ancho de banda y la cantidad de usuarios

conectados simultáneamente. Además, estas instituciones deben pagar una cuota establecida para la contratación del servicio de Internet. La tarifa aumenta en la medida que se quiera adquirir más ancho de banda (mayor velocidad de conexión). Por tal motivo, no es posible mantener un acceso a Internet constante y sin fallos, lo cual impide el uso eficiente de los servicios que brinda el CMF.

Con el creciente auge de la informatización de la sociedad cubana se han venido incrementando en el país diversos puntos de acceso a internet. Sin embargo, aún no son suficientes para el abastecimiento total de la población. El internet en Cuba es deficiente pues los precios para poder acceder al mismo no son asequibles para todo integrante de la sociedad. Además, resulta ser lento e inaccesible en cualquier lugar del país que no cuente con un punto de acceso. No todas las escuelas poseen el privilegio de contar con este avance tecnológico y solo una pequeña porción de las empresas pueden acceder al mismo. Los demás integrantes de la sociedad como las amas de casa, trabajadores por cuenta propia, personas de la 3era edad y demás que no pertenecen a una escuela o empresa que cuente con este recurso, no pueden acceder a internet, a menos que tengan una buena solvencia económica. En contraste a esto la gran mayoría de las escuelas y empresas poseen servicio de correo nacional. Este servicio es brindado por la Empresa de Telecomunicaciones de Cuba (ETECSA), el cual está tomando gran auge y se mantiene expandiéndose por toda la población. Para contar con esta facilidad no es necesario estar cerca de un punto de acceso a internet, el costo de este servicio es bastante asequible para la sociedad además de ser un servicio gratis para escuelas y empresas.

Desde el momento de su lanzamiento hasta la fecha, CMF ha logrado las estadísticas que se muestran en la

Figura 1, disfrutando de popularidad entre los usuarios que disfrutan navegar por sus páginas a diario.



Figura 1: Estadísticas de CMF del 25 de enero de 2017

Sin embargo, los servicios del sistema no les llegan a todas las personas. Un ejemplo son todas aquellas que desean ejercitar su capacidad cerebral resolviendo problemas de agilidad mental pero no tienen acceso a internet y si poseen un correo electrónico.

El CMF por su parte no cuenta con una funcionalidad que permita a los usuarios que no tienen acceso a internet o intranet y sí a un correo electrónico nacional poder establecer una comunicación con el CMF. Por este motivo no pueden disfrutar de sus servicios, privándose de los beneficios que este sitio aporta a la sociedad.

Partiendo de la situación problemática expuesta con anterioridad se plantea el siguiente **problema a resolver**: ¿Cómo extender los servicios del jurado en línea Fragua Caribeña de la Mente (CMF) a los usuarios que no cuentan con un acceso regular a Internet o Intranet?

Se define como **objeto de estudio**: La evaluación automática en aplicaciones informáticas.

Para darle solución al problema planteado se propone como **objetivo general**: Desarrollar un sistema automático de respuesta por correo electrónico que permita extender los servicios del Jurado en Línea Fragua Caribeña de la Mente, a los usuarios con correo electrónico sin acceso a Internet o Intranet.

Campo de acción: La evaluación automática mediante el uso de correo electrónico en los Jurados en línea.

Teniéndose como objetivos específicos para el desarrollo de la investigación:

- Elaborar la Fundamentación Teórica de la investigación.
- Definir las funcionalidades que debe cumplir el sistema.
- Implementar las funcionalidades definidas para el sistema.
- Aplicar pruebas al sistema resultante.

Para dar cumplimiento a los objetivos específicos se definieron las siguientes **tareas de investigación**:

- Revisión bibliográfica para generar el marco teórico conceptual en lo referente al desarrollo de sistemas automáticos de respuesta por correo electrónico.
- Estudio de sistemas homólogos para conocer aspectos regulares en el diseño de sistemas automáticos de respuesta por correo electrónico.

- Adaptación del sistema a desarrollar a las herramientas informáticas y metodologías de desarrollo de software definidas en el sistema base para su posterior integración.
- Investigación de los tipos de pruebas correspondientes a la metodología empleada, para su posterior aplicación sobre el sistema desarrollado.

Para la solución de la investigación se hace uso de métodos científicos que servirán como guía para la realización de la misma. Estos métodos son procedimientos sistemáticos e instrumentos que llevan a un conocimiento científico y son capaces de garantizar cualquier tipo de solución.

“La Informática utiliza dos grandes clases de métodos de investigación: los métodos lógicos y los empíricos. Los primeros son aquellos que se basan en la utilización del pensamiento en sus funciones de deducción, análisis y síntesis, mientras que los métodos empíricos, se aproximan al conocimiento del objeto mediante el conocimiento directo y el uso de la experiencia.” (22)

Métodos teóricos

- **Método Analítico-Sintético:** Método utilizado para analizar la situación existente respecto a la conexión al Caribbean Mind Forge mediante correo. Empleado además para determinar las características que debe tener el sistema de respuesta automática por correo electrónico del CMF. Usado también para analizar las particularidades de diversos Jueces en línea y protocolos de correo electrónico, posibilitando sintetizar y resumir aquellos elementos más relevantes para la investigación.
- **Modelación:** Se pone en práctica para la realización del diagrama entidad-relación correspondiente a la fase de exploración y diseño, proporcionando una mejor perspectiva del sistema que se quiere desarrollar.

Método empírico

- **Observación:** Método utilizado para observar los problemas de acceso de la sociedad cubana al CMF y las características de aplicaciones similares a la propuesta en este trabajo.

El documento de la investigación estará estructurado en 3 capítulos:

Capítulo 1: Fundamentación teórica: Se realiza un estudio de las soluciones similares a la propuesta de la investigación. Se estudian, además algunos protocolos de correo. Incluye una caracterización de las herramientas, tecnologías y metodologías usadas en el desarrollo del software.

Capítulo 2: Exploración, Planificación y Diseño del Sistema: En este capítulo se elabora un modelo de datos. Se especifican las funcionalidades y las características no funcionales con los que debe cumplir el sistema y se definen las tarjetas CRC (Clase-Responsabilidad-Colaboración).

Capítulo 3: Implementación y Pruebas: En este capítulo se detalla el proceso de implementación de la solución a través de tarjetas llamadas: Tarea de Ingeniería. Se describen las pruebas funcionales realizadas para validar el sistema desarrollado.

Capítulo.1 Fundamentación Teórica

1.1 Introducción

En este capítulo se hace un estudio de los principales conceptos asociados a la investigación, así como el estado del arte sobre sistemas similares vinculados al campo de acción. Se realiza un estudio de aquellas soluciones informáticas que por su funcionamiento y características son consideradas de referencia obligatoria para la investigación. Se hace un análisis de las herramientas y tecnologías a utilizar, así como la metodología de desarrollo de software seleccionada para el desarrollo del software.

1.2 Análisis de soluciones similares

En el mundo hay diversos sistemas de evaluación automática ya sean juegos didácticos, jurados en línea u otras aplicaciones informáticas que permiten el desarrollo de la mente humana. Las mismas hacen uso de técnicas de aprendizaje como la visualización, memorización auditiva o alguna otra técnica para la adquisición de agilidad mental. La agilidad o habilidad mental es la capacidad que tienen los seres humanos de dar solución a problemas rápidamente usando el razonamiento lógico, la memoria y el conocimiento.

En la sociedad cubana se aboga por una propagación intelectual. Siendo su enfoque el mejoramiento de las capacidades del ser humano como es la respuesta certera, rápida y concisa que facilita la resolución de problemas en la vida cotidiana.

Un jurado en línea es una aplicación web que tiene como tarea la evaluación y calificación de problemas de diversa índole.

Un juego didáctico o software de habilidad mental es un sistema informático diseñado por neuropsicólogos. Estos tienen como destino ganar memoria, ganar capacidad de concentración y mantener el nivel de estrés bajo control mientras te diviertes. (8)

Algunos softwares¹ de evaluación automática son los siguientes:

1.2.1 Juez en línea de la Universidad de Valladolid (UVA)

Uno de los primeros jueces en línea que surgieron fue el UVA. Está integrado a la plataforma EduJudge, convirtiéndolo en uno de los pocos jueces en línea que poseen una integración con el proceso de enseñanza-aprendizaje. Algunas de sus características son:

¹ Programas o aplicaciones informáticas.

- Problemas de competencias oficiales del mundo.
- Un módulo de estadísticas, el cual representa información respecto a las estadísticas de uso de los problemas, usuarios y concursos.
- Una tabla de posiciones, dedicada al desempeño de los usuarios en competencias. (30)
- El UVA no cuenta con conexión mediante correo electrónico a sus servicios en línea.

1.2.2 Sphere Online Judge (SPOJ)

SPOJ es un juez en línea desarrollado por el Sphere Interest Project, entidad afiliada a la Universidad de Tecnología de Gdansk en Polonia, dedicado al desarrollo de tecnologías en línea para proveer herramientas de evaluación automática de habilidades. SPOJ se ha convertido en uno de los jueces en línea más populares en la comunidad internacional. Posee más de 5859 problemas publicados, convirtiéndolo en uno de los archivos más grandes de problemas. (31)

Posee varias características que lo diferencian de la mayoría de los jueces en línea, tales como:

- Soporta 48 lenguajes, la mayor cantidad de lenguajes soportados por un juez en internet.
- Tiene un foro de discusión como espacio de intercambio para sus usuarios.
- Permite comparar los usuarios.
- Posee tiempos de evaluación muy rápidos. (31)
- El SPOJ no cuenta con conexión mediante correo electrónico a sus servicios en línea.

1.2.3 Caribbean Online Judge (COJ)

El desarrollo del Juez en línea caribeño de programación (COJ por sus siglas en inglés) tiene sus antecedentes en el año 2006. El COJ está disponible en internet desde el 5 de junio de 2010. La característica más considerable del sistema es los tiempos de respuesta en la mayoría de las interfaces, no así en la vista del estado de los envíos. (27)

- Soporta 14 lenguajes.
- Miles de problemas publicados en su archivo.
- Hay usuarios de al menos 144 países distintos y dentro de ellos de 1010 Instituciones.
- Ha sido anfitrión de cientos de competencias desde el nivel local hasta el nivel regional del ACM-ICPC del Caribe y de otras varias competencias nacionales de programación.
- Mostrar los nuevos registros de usuarios en el COJ en un periodo de tiempo.
- Comparar usuarios en cuanto a veredictos.

- Mostrar línea de tiempo por desempeño de usuario en el concurso.
- Posee una aplicación nombrada COJMail 2.0, capaz de interactuar con el Juez en Línea Caribeño (COJ) a través de correo electrónico.

COJMail 2.0

COJMail 2.0 es un daemon². Esta aplicación aún en condiciones de acceso a Internet desfavorable es capaz de interactuar con el COJ. Algunas de sus deficiencias son el empleo de un micro lenguajes de sintaxis compleja y la ausencia de funcionalidades relacionadas con el registro de usuarios y concursos. (27)

A continuación se describe su principio de funcionamiento:

1. Realiza la descarga de los correos desde el buzón del COJ, usando el protocolo que soporta el servidor de correo de la UCI, IMAP.
2. En caso que existan correos, obtiene el asunto de cada correo en particular e identifica la petición realizada a través de un proceso de análisis gramatical.
3. Valida los parámetros especificados para cada petición en particular. En caso que los parámetros sean correctos, se ejecuta dicha petición en la base de datos, construyendo un correo de respuesta a partir de los datos obtenidos.
4. Construye una vista en formato HTML o un documento PDF y la envía como respuesta al usuario a través del correo electrónico, usando el API JavaMail y el protocolo SMTP. Estas vistas son similares a las presentadas en el COJ. En caso de que hayan sido identificados errores en la capa de validación, es construida una vista de ayuda en formato HTML.
5. Se espera un tiempo definido y se ejecuta nuevamente este proceso (desde el paso 1). (27)

1.3 Conclusiones del análisis de sistemas similares

Luego de realizar un estudio del UVA, el SPOJ y el COJ en busca de algún medio de conexión con los mismos mediante correo electrónico se pudo arribar a la conclusión de que no fueron encontrados sistemas encargados de gestionar los servicios brindados por el UVA o el SPOJ a través de correo electrónico.

²Programa o proceso que se ejecuta constantemente, por lo general en plataformas UNIX, con el fin de realizar determinadas tareas, por ejemplo la invocación de un servicio web, la recepción y envío de correos electrónicos, entre otros.

Sin embargo, se halló una solución informática similar a la propuesta en este trabajo, cuyo nombre es COJMail que brinda una conexión al COJ mediante correo electrónico.

Aprovechando la sencillez del micro lenguaje desarrollado para el CMFMail y la facilidad que brindan los múltiples clientes de correo que existen, así como su capacidad para descargar, enviar y organizar correos, se decide no implementar un cliente de correo y en su lugar elaborar un manual de ayuda al usuario, así como formularios de registro con validaciones de ayuda.

1.4 Análisis de Protocolos de Correo Electrónico

Un protocolo de correo electrónico es un estándar o procedimiento que siguen dos máquinas para enviar o recibir e-mails³. En este trabajo se observan los protocolos que soportan los clientes de correo. Los protocolos de correo electrónico estudiados fueron los siguientes:

1.4.1 Protocolo de transporte de correo SMTP

SMTP (*Simple Mail Transfer Protocol*, Protocolo para la transferencia simple de correo electrónico) es el protocolo que se emplea en el envío de correo electrónico entre dos computadoras y otros dispositivos electrónicos. En las RFC (Request for Comments) 2822 y RFC 2821 se puede hallar lo referente a su uso y el formato de los mensajes.

El protocolo SMTP es un protocolo de comunicaciones basado en texto, y utilizado para intercambiar mensajes de correo electrónico. Debido a que presenta algunas limitaciones en el procesamiento de la cola de mensajes recibidos en el servidor de destino (algunas de sus características son consideradas inseguras), se emplea para gestionar el correo saliente. (28).

1.4.2 Protocolos de acceso a correo electrónico POP e IMAP

Para acceder a los mensajes desde los servidores de correo existen dos protocolos fundamentales; Post Office Protocol (POP) e Internet Message Access Protocol (IMAP).

POP

El protocolo POP3 (Postal Office Protocol, o protocolo de oficina postal versión 3), definido en el documento RFC 1939, descarga los mensajes de correo en el ordenador (u otro dispositivo), donde quedan almacenados y organizados en las carpetas que se haya creado. Una vez que se han bajado del servidor, los correos sólo serán accesibles desde la máquina a la que se haya bajado, ya que por defecto, al descargarse los mensajes, éstos quedan eliminados del servidor.

³ Correo electrónico

Es ideal para conectarse siempre desde un mismo ordenador, y además ofrece la ventaja de poder acceder a los mensajes antiguos (ya descargados) sin necesidad de tener conexión a internet.

La elección entre IMAP o POP3, una vez conocidas las ventajas e inconvenientes de cada protocolo, se deja a elección del usuario. Una solución usual es utilizar el POP3 en el ordenador principal del trabajo, o en el ordenador personal de casa y descargar en ellos los correos para gestionar mejor su almacenamiento. En movilidad, o para consultar el correo desde otros ordenadores, se puede usar el IMAP. (28)

IMAP

El protocolo IMAP (Internet Message Access Protocol, o Protocolo de acceso a mensajes de internet), en el documento RFC 2060, no descarga los mensajes al ordenador: tanto los mensajes como las carpetas que se hayan creado se mantienen en el servidor.

Esto es ventajoso cuando la conexión para leer correos se realiza desde diferentes dispositivos, por ejemplo, un portátil o el Smartphone: se sabe que siempre se puede acceder a todos los mensajes, y que el buzón estará actualizado.

También es interesante para preservar la privacidad cuando se lee correo desde un ordenador de uso público o compartido, ya que no almacena información en la máquina local.

El protocolo IMAP es el más aconsejable cuando se accede a correos desde varios dispositivos, o en movilidad.

Como precaución, se ha de borrar periódicamente el contenido de la cuenta para que no exceda del límite de espacio concedido.

El inconveniente de este protocolo es que siempre hay que disponer de conexión a internet, incluso para acceder y trabajar con los mensajes antiguos. (28)

1.5 Seguridad en los protocolos de correo electrónico

Los protocolos SMTP, IMAP, y POP, soportan al protocolo SSL/TLS (Secure Sockets Layer/Transport Layer Security) con el propósito de lograr seguridad en la información que circula en la red. En la siguiente tabla se muestra el identificador y puerto de las extensiones de SMTP, IMAP y POP que usan SSL/TLS.

Tabla 1: Protocolos de correo electrónico sobre SSL/TLS.

Protocolo con SSL/TLS	Identificador	Puerto TCP (Por defecto)
SSL/TLS con SMTP	Smtps	465
SSL/TLS con IMAP	Imaps	993
SSL/TLS con POP	Pops	995

Protocolo SSL/TLS

“El protocolo SSL y su sucesor TLS son protocolos criptográficos que proporcionan comunicaciones seguras por una red. Se basa en un proceso de cifrado de clave pública estableciendo un canal de comunicación seguro (cifrado) entre dos equipos después de una fase de autenticación.” (11)

“SSL proporciona confidencialidad, mediante el uso de encriptación, integridad, ya que los datos recibidos son exactamente iguales a los datos enviados y autenticación, ya que se realiza utilizando un certificado digital por parte del servidor. (Y cliente opcionalmente).” (12)

1.6 Conclusiones del Análisis de los Protocolos de Correo

Para realizar la descarga y envío de los correos desde la cuenta de correo impersonal del CMF se usarán los protocolos que soporta el servidor de correo de la UCI: Protocolo de acceso a correos electrónicos IMAP y protocolo de transporte de correo SMTP.

1.7 Metodología, Herramientas, Tecnología y Lenguajes de desarrollo.

Como el desarrollo del sistema automático de respuesta por correo electrónico CMFMail no afecta el ambiente de desarrollo y de despliegue del CMF, el mismo se acoge a las definidas en el proyecto de tesis titulado: “Juez en línea para el desarrollo, ejercitación y realización de concursos de habilidad mental” del autor: “Javier de León Barral”.

1.7.1 Metodología

La metodología para guiar el proceso de desarrollo es Xtreme Programming (XP). XP es una metodología ágil de desarrollo de software que surge a finales de la década de los 90 y como

principales creadores tuvo a Kent Beck y Martin Flower. Su objetivo fundamental es desarrollar el software en un ambiente donde el cliente se sienta satisfecho y donde se potencie las relaciones interpersonales, como clave para promover el trabajo cohesionado de todos los miembros del equipo en función de lograr eficiencia y rapidez. XP basa su funcionamiento en la retroalimentación continua entre el cliente y el equipo de desarrollo.

“XP mejora un proyecto de software en cinco aspectos esenciales: la comunicación, la simplicidad, la retroalimentación, el respeto y el coraje. Esto quiere decir que un proyecto guiado por la metodología XP se caracteriza por la comunicación fluida de su equipo de trabajo, las soluciones generadas poseen un diseño simple y sencillo y existe una retroalimentación continua entre el cliente y los desarrolladores. Además, existe el respeto hacia las contribuciones realizadas por cada miembro del equipo y la valentía para enfrentar posibles cambios que surjan durante la vida del proyecto.” (13)

Se escoge XP como metodología para el desarrollo del CMFMail por ser la metodología empleada en el CMF.

1.7.2 Herramientas

PostgreSQL 9.2

PostgreSQL es un sistema de administración de bases de datos relacionales orientadas a objetos (ORDBMS, object-relational database management system) basado en POSTGRES, Versión 4.2, desarrollado en el Departamento de Ciencias Computacionales de la Universidad de California, Berkeley. POSTGRES fue pionero en muchos conceptos que solo llegaron a aparecer en algunos sistemas de bases de datos comerciales mucho tiempo después. PostgreSQL es un descendiente libre del código original de Berkeley. (15)

Se escoge PostgreSQL como sistema gestor de base de datos por ser el sistema gestor de base de datos empleado en el CMF.

Visual Paradigm 8.0

Visual Paradigm para UML (VP-UML) es una poderosa plataforma-cruzada y ya el modelado UML visual y la herramienta CASE son muy fáciles de usar. VP-UML provee a los desarrolladores de software la plataforma de desarrollo de borde cortante para construir aplicaciones de calidad más rápido, mejor y barato. Facilita excelente interoperabilidad con otras herramientas CASE y la mayoría de los principales IDEs lo cual aventaja su proceso de desarrollo modelo-código-despliegue. (16)

Se escoge el Visual Paradingm como herramienta para la elaboración del diagrama entidad-relación por su reconocida capacidad de modelado y facilidad de uso.

NetBeans Entorno de Desarrollo Integrado (IDE) 7.4

Netbeans es un ambiente de desarrollo integrado de java, el cual posibilita el rápido desarrollo de aplicaciones con la mayoría de marcos de trabajo, tecnologías y servidores. Diferente a otros IDEs, Netbeans incluye paquetes con una amplia gama de funcionalidades fuera de caja, así como soporte para diferentes marcos de trabajo, servidores, bases de datos y desarrollo móvil. (17)

Se escoge Netbeans como IDE por ser el IDE empleado en CMF.

1.7.3 Tecnología

Servidor Apache Tomcat 7.0

Actualmente el CMF está desplegado usando como servidor de aplicación Apache Tomcat en su versión 7. La solución propuesta será desplegada conjuntamente con el CMF, compartiendo el mismo servidor de aplicación.

Apache Tomcat, también conocido como Tomcat o Jakarta Tomcat, es un servidor web multiplataforma que funciona como contenedor de servlets y que se desarrolla bajo el proyecto denominado Jackarta perteneciente a la Apache Software Foundation bajo la licencia Apache 2.0 y que implementa las especificaciones de los servlets y de Java Server Pages o JSP de Sun Microsystem. Dicho servidor es mantenido y desarrollado por miembros de la fundación y voluntarios independientes, los cuales tienen libre acceso al código fuente bajo los términos establecidos por la Apache Software Foundation. (14)

Se escoge el Apache Tomcat como servidor de aplicaciones web por ser el servidor de aplicaciones web empleado en el CMF.

Máquina Virtual de Java7.0

Ejecuta los bytecodes (ficheros compilados con extensión*.class) creados por el compilador de Java (javac.exe). Tiene numerosas opciones entre las que destaca la posibilidad de utilizar el denominado JIT (Just-In-Time Compiler), que puede mejorar entre 10 y 20 veces la velocidad de ejecución de un programa. Gracias a la JVM el código java es capaz de ejecutarse en cualquier tipo de máquina. Una vez compilado no es necesaria ninguna modificación por el hecho de cambiar de procesador o de ejecutarlo en otra máquina. La clave consiste en un código neutro el cual está preparado para ser ejecutado sobre una máquina hipotética o virtual, denominada Java Virtual Machine (JVM). (18)

Se escoge la JVM por ser empleada en el CMF.

1.7.4 Lenguajes de desarrollo

Groovy

Groovy es un lenguaje nacido con la misión de llevarse bien con Java, vivir en la Máquina Virtual y soportar los tipos de datos estándar, pero añadiendo características dinámicas y sintácticas presentes en otros lenguajes como Python, Smalltalk o Ruby. El código fuente Groovy se compila a bytecodes igual que Java, y es posible instanciar objetos Java desde Groovy y viceversa. Lo que Groovy aporta es una sintaxis que aumenta enormemente la productividad y un entorno de ejecución que permite manejar los objetos de formas que en Java serían extremadamente complicados.

En general, el código Groovy guarda bastante parecido con Java, lo que hace que la introducción en el lenguaje sea sencilla para programadores familiarizados con él. La mayoría de lo que se hace en Java puede hacerse también en Groovy. (29)

Se escoge el lenguaje groovy por ser empleado en el CMF y necesario para la creación de la clase controladora que brindará los servicios al CMFMail.

Grails

Grails, framework web para la plataforma Java y base del Sistema CMF. Se basa en el lenguaje dinámico Groovy. Hace uso de Lenguajes de Dominio Específico (DSLs) potentes pero a la vez sencillos. Es versátil y posee un ecosistema de Plugins que mejora la productividad en un conjunto cada vez más amplio de escenarios. Grails se ha hecho inmensamente popular, y un motor de cambio en el espacio Java.

Grails permite crear aplicaciones en días, en lugar de semanas. En comparación con el típico framework de Java, con Grails se necesita menos código para obtener el mismo resultado. Menos código significa menos errores y menos líneas de código de mantener. (29)

Se escoge el framework grails debido a que es el framework con el que se hizo el CMF y resulta necesario en la clase controladora que brinda los servicios al CMFMail.

Java

Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido.

Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada. Desarrollaron un código neutro que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una máquina hipotética o virtual denominada Java Virtual Machine (JVM). Era la JVM quien interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Esto permitía lo que luego se ha convertido en el principal lema del lenguaje: "Write Once, Run Everywhere". A pesar de los esfuerzos realizados por sus creadores, ninguna empresa de electrodomésticos se interesó por el nuevo lenguaje. (18)

Se escoge el lenguaje java por ser empleado en el CMF y por su versatilidad que posibilita la implementación de las funcionalidades del CMFMail.

JavaMail

El API (Interfaz de programación de aplicaciones) JavaMail es un paquete opcional (extensión estándar) para leer, componer, y enviar mensajes electrónicos. Se usa este paquete para crear programas del tipo MUA (Mail User Agent), similares a Eudora, Pine, y Microsoft Outlook. Su propósito principal no es transportar, enviar, o re-enviar mensajes como sendmail u otros programas del tipo MTA (Mail Transfer Agent). En otras palabras, los usuarios interactúan con los programas para leer y escribir e-mails. Los programas MUA tratan con los programas MTA para el envío real.

El API JavaMail está diseñado para proporcionar acceso independiente del protocolo para enviar y recibir mensajes dividiéndose en dos partes:

- La primera parte del API es básicamente, cómo enviar y recibir mensajes independientemente del proveedor/protocolo.
- La segunda parte habla de lenguajes específicos del protocolo como SMTP, POP, IMAP, y NNTP (Network News Transport Protocol). (19)

Se escoge el API Javamail por su capacidad de enviar y recibir correos de forma relativamente sencilla.

UML

El Lenguaje Unificado de Modelado (Unified Modeling Language, UML) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. UML incluye conceptos semánticos, notación, y principios generales. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo, pero no especifica en sí mismo qué metodología o proceso usar. (1)

Se escoge UML como lenguaje de modelado por ser empleado en el CMF y por ayudar en la elaboración del diagrama entidad-relación.

1.8 Conclusiones

Con el estudio del estado del arte se pudo conocer los conceptos más relevantes en la investigación, así como los sistemas similares existentes. Se identificaron sus características principales, buscando una conexión a los mismos a través de correo electrónico. Dicho estudio arrojó la existencia en el COJ de una aplicación de respuesta automática llamado COJMail 2.0 del cual se tomaron algunas características para el desarrollo del CMFMail. Se identificaron los protocolos de transporte de correo así como los protocolos de acceso a correo electrónico seleccionando de esto último el IMAP por ser el soportado en la universidad. Se hizo un estudio de las herramientas, tecnologías y lenguajes de programación empleados en el Caribbean Mind Forge para desarrollar el sistema automático de respuesta por correo electrónico.

Capítulo.2 Exploración, Planificación y Diseño del Sistema

2.1 Introducción

En este capítulo se presenta la propuesta de solución para el desarrollo del sistema automático de respuesta por correo electrónico CMFMail para el CMF. En este apartado de la investigación se recogen los resultados de las fases Exploración, Planificación y Diseño correspondientes a XP. Los términos más importantes de estas fases son las Historias de usuario, Iteraciones, Planificación de entregas y las Tarjetas CRC, los cuales son explicados detalladamente en el transcurso del capítulo.

2.2 Propuesta de solución

Para darle respuesta a la situación problemática se decide implementar un sistema automático de respuesta por correo denominado CMFMail para el sistema CMF. Este sistema automático contribuye a extender los servicios del Jurado en Línea Fragua Caribeña de la Mente a los usuarios con correo electrónico sin acceso a Internet o Intranet.

En la siguiente imagen se muestra cómo funciona el CMFMail. Una descripción más detallada se puede apreciar en el sub epígrafe: Principio de funcionamiento.

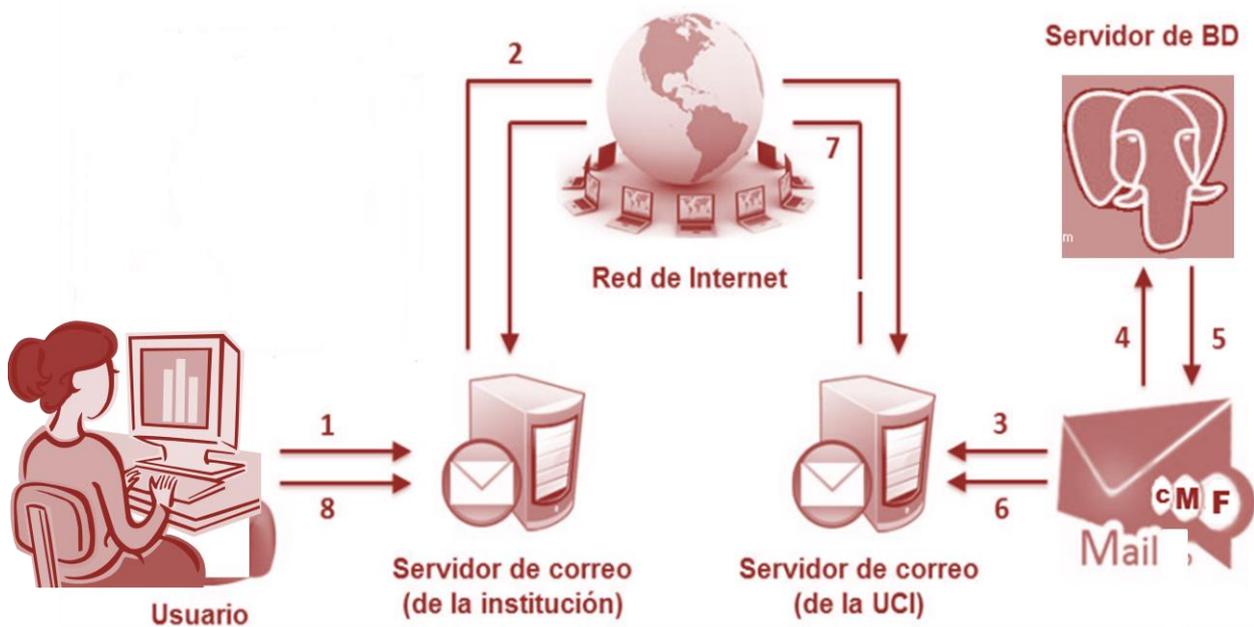


Figura 2: Propuesta de solución.

2.2.1 Funcionamiento general

El CMFMail es un daemon. “Su funcionamiento consiste en descargar de manera periódica los correos que se encuentran en la cuenta de correo del CMF. Una vez realizada la descarga, responde la petición del usuario y copia los correos en una carpeta llamada CMFmailstore. La periodicidad entre las descargas es de 10 segundos.”

2.2.2 Principio de funcionamiento

Su funcionamiento se describe a continuación:

1. Realiza la descarga de los correos desde la carpeta CMFmail usando el protocolo que soporta el servidor de correo de la UCI, IMAP.
2. Hace una copia en la carpeta CMFmailstore de los correos de CMFmail.
3. Elimina los correos de CMFmail para agilizar las descargas de correo.
4. En caso que existan correos, obtiene el asunto de cada correo en particular e identifica la petición realizada a través de un proceso de análisis gramatical.
5. Valida los parámetros especificados para cada petición en particular. En caso que los parámetros sean correctos, se ejecuta dicha petición en la base de datos, construyendo un correo de respuesta a partir de los datos obtenidos.
6. Construye una vista en formato HTML y la envía como respuesta al usuario a través del correo electrónico, usando el API JavaMail y el protocolo SMTP.
7. Se espera un tiempo definido y se ejecuta nuevamente este proceso (desde el paso 1).

Capítulo.2 Exploración, Planificación y Diseño del Sistema

2.3 Funcionalidades

No	Nombre	Descripción
1	Configurar cuenta de correo	Permite al administrador cambiar la cuenta de correo, el usuario y la contraseña.
2	Configurar servidor de servicios	Permite al administrador especificar la dirección del servidor donde se encuentra la controladora de servicios para el CMFMail
3	Configurar servidor de envío de correo	Permite al administrador especificar la dirección del servidor para el envío de correo.
4	Configurar servidor de descarga de correo	Permite al administrador especificar la dirección del servidor para la descarga de correo.
5	Configurar puerto para envío de correo	Permite al administrador especificar el puerto empleado para el envío de correo.
6	Configurar puerto para descarga de correo	Permite al administrador especificar el puerto empleado para la descarga de correo.
7	Leer un problema	Permite que el usuario a través de su cliente de correo pueda leer un problema

Capítulo.2

Exploración, Planificación y Diseño del Sistema

8	Responder un problema	Permite que el usuario envíe una respuesta y determina si es correcta o no.
9	Realizar Comentarios	Permite al usuario ver los comentarios de un problema solo si lo ha respondido correctamente
10	Registrar usuario	Permite al usuario registrarse en el CMF.
11	Ver Ranking	Permite al usuario ver las primeras 10 posiciones en el ranking del CMF.
12	Ver Posición en el ranking	Permite al usuario ver su posición en el ranking del CMF.
13	Ver Últimas sentencias	Permite al usuario ver las últimas 10 sentencias enviadas por él.
14	Ver Sentencias	Permite al usuario especificar un rango de sentencias del CMF no mayor que 100.
15	Consultar Manual del CMFMail	Permite al usuario consultar el Manual del CMFMail.
16	Ver Últimos problemas	Permite al usuario saber la lista de los últimos 10 problemas del CMF.
17	Explorar problemas	Permite al usuario especificar un rango de problemas con imagen no mayor que 25.

2.4 Características no funcionales

No	Características no funcionales
1	Tener una cuenta de correo con las carpetas CMFmail y CMFmailstore creadas.
2	Crear un filtro que envía a la carpeta CMFmail todos los correos que llegan con asunto CMF.
3	Tener instalado la JDK 7.
4	Instalar el CMFMail junto al sitio web del CMF.
5	Agregar clase controladora con los servicios implementados al sitio web del CMF.

6	PC con 2 GB de RAM
7	Al menos 1 GB de espacio en disco C
8	Servidor de base de datos PostgreSQL 9.2 o superior

2.5 Micro lenguaje

Para establecer comunicación con el CMF vía correo electrónico los usuarios deben enviar un correo a la dirección electrónica del correo impersonal del CMF (CMF@uci.cu). En el asunto de dicho correo el usuario debe escribir unos comandos particulares conocidos como micro lenguaje. Este término representa el lenguaje común entre los usuarios y el CMFMail, entendible desde luego por ambas partes. El objetivo del micro lenguaje es que los usuarios puedan especificar la información que desean gestionar u obtener. La estructura del micro lenguaje es similar a muchos de los comandos empleados en consola (existencia de pares con la forma <critério><valor>). Una explicación más detallada de la estructura del micro lenguaje puede ser consultada en el Trabajo de Diploma “Implementación y Prueba de un Jurado Online sobre correo electrónico” del autor Nersa Doraines Acosta Labrada.

2.5.1 Complejidad del micro lenguaje

La probabilidad de que un usuario escriba de forma incorrecta el micro lenguaje está estrechamente relacionada con la complejidad sintáctica que poseen estos comandos. A medida que aumenta la complejidad del micro lenguaje aumenta la probabilidad de errores, que por lo general provoca frustración en el usuario al no poder obtener lo que desea. Por este motivo se establece un micro lenguaje asequible y sencillo para cualquier persona que desee interactuar con el CMF.

2.5.2 Micro lenguaje establecido

Para garantizar que los usuarios tengan conocimiento de la existencia del sistema automático de respuesta por correo en el CMF, los administradores del Juez en línea pondrán una sección para dar a conocer las funcionalidades del mismo y su forma de uso mediante el Micro lenguaje establecido.

Micro lenguaje definido para el uso de los servicios del CMF mediante correo electrónico:

Capítulo.2

Exploración, Planificación y Diseño del Sistema

Tabla 2: Descripción del Micro lenguaje a usar para acceder a los servicios del CMF

No	Micro lenguaje	Descripción
1	CMF L #	Permite que el usuario a través de su cliente de correo pueda leer un problema.
2	CMF R #	Permite que el usuario envíe la respuesta de un problema.
3	CMF C #	Permite al usuario ver los comentarios de un problema solo si lo ha respondido correctamente
4	CMF RK	Permite al usuario ver las primeras 10 posiciones en el ranking del CMF
5	CMF P	Permite al usuario ver su posición en el ranking del CMF
6	CMF S	Permite al usuario ver las últimas 10 sentencias enviadas por él.
7	CMF T offset # max #	Permite al usuario especificar un rango de sentencias del CMF no mayor que 100.
8	CMF UP	Permite al usuario saber la lista de los últimos 10 problemas del CMF
9	CMF E offset # max #	Permite al usuario especificar un rango de problemas con imagen no mayor que 25.
10	CMF RG	Permite al usuario solicitar un formulario para registrarse.
11	CMF F	Envío de datos de registro de usuario
12	CMF manual	Permite al usuario consultar el Manual del CMFMail.

2.6 Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario (en lo adelante HU). Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. (23)

2.6.1 Historias de usuario

Las Historias de usuario son la técnica utilizada en XP para especificar las funcionalidades del software. Son escritas por el cliente, y por lo general serán descripciones cortas en el lenguaje del usuario. Las HU proporcionarán la estimación del riesgo y el tiempo que conllevará su implementación. El nivel de detalle de las HU debe ser el mínimo posible. La forma de tratar estos artefactos es muy flexible, pudiéndose romper en cualquier momento, cambiar o añadir nuevas. Son descompuestas en tareas de programación y asignadas a los programadores para poder ser implementadas.

Los clientes del sistema son parte del equipo de desarrollo y discuten escenarios con otros miembros del equipo. Desarrollan conjuntamente una «tarjeta de historias» (story card) que recoge las necesidades del cliente. (16)

A continuación se muestra la plantilla creada para las HU.

Tabla 3: Plantilla de Historias de Usuario

Historia de usuario	
Número: Número de la HU.	Nombre: Nombre de la HU.
Prioridad: Cuán importante es para el cliente o el equipo de desarrollo. Alta. Media. Baja	Complejidad: Nivel de dificultad para el desarrollador. Alta. Media. Baja.
Iteración: Iteración a la que corresponde.	Estimación: Tiempo estimado para realizar la actividad
Descripción: Breve reseña de la HU, describiendo las acciones del usuario.	
Notas: Observaciones de interés.	

A continuación las Historias de Usuario de cada una de las funcionalidades.

Capítulo.2 Exploración, Planificación y Diseño del Sistema

Tabla 4: Historia de Usuario Aplicación básica.

Historia de usuario	
Número: 1.	Nombre: Crear Aplicación básica.
Prioridad: Alta.	Complejidad: Alta.
Iteración: 1.	Estimación: 8 semanas.
Descripción: Crear una aplicación que debe cada 10 segundos descargar los correos con asunto CMF de la carpeta CMFmail, eliminarlos de la misma y copiarlos a la carpeta CMFmailstore, ambos directorios de la cuenta de correo impersonal del CMF. Posterior a la descarga debe analizar cada correo y mediante un micro-lenguaje permitir el registro de usuario, ver rango de problemas y responder un problema. Para llevar a cabo estas funcionalidades se debe crear previamente una clase controladora que brinde los servicios necesarios para interactuar con la base de datos del CMF	
Notas:	

Tabla 5: Historia de Usuario Leer un problema

Historia de usuario	
Número: 2.	Nombre: Leer un problema
Prioridad: Alta.	Complejidad: Media.
Iteración: 2.	Estimación: 1 semana
Descripción: Permite al usuario a través del cliente de correo leer un problema.	
Notas:	

Tabla 6: Historia de Usuario Comentarios

Historia de usuario	
Número: 3.	Nombre: Comentarios.
Prioridad: Media.	Complejidad: Media.
Iteración: 2.	Estimación: Tres semanas.

Descripción: Permite al usuario ver los comentarios de un problema solo si lo ha respondido correctamente.

Notas:

Las HU de las restantes funcionalidades pueden ser consultadas en los Anexos del documento.

2.7 Planificación

En esta fase se controla la velocidad del proyecto, se crea un plan de entregas, se rota el personal del proyecto y se realizan sistemáticamente reuniones entre el cliente y el equipo de trabajo.

2.7.1 Estimación de Esfuerzo

La velocidad del proyecto o estimación de esfuerzo es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de HU que se pueden implementar en una iteración; de esta forma se conocerá la cantidad de HU que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto se controlará que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Una buena práctica es recalcular la velocidad del proyecto cada 3 o 4 iteraciones y si se aprecia que no es adecuada se debe negociar con el cliente para confeccionar un nuevo Plan de entregas.

Se pueden construir varias veces al día nuevas versiones del software y los incrementos se entregan al cliente cada dos meses aproximadamente. (16)

Tabla 7: Velocidad del Proyecto

	Iteración 1	Iteración 2	Iteración 3
Horas	200	150	125
Semanas	8	6	5
Horas semanales	25	25	25
Historias de usuario (cantidad)	1	6	5

2.8 Iteraciones

El desarrollo de las HU fue dividido en iteraciones. En este segmento del proyecto, el cliente vuelve a ser parte esencial en la toma de decisiones, pues es quien decide que HU se implementará en cada iteración. Estas HU están definidas por iteraciones en dependencia de su importancia funcional.

Planificación de HU de la 1º Iteración

En esta primera iteración se seleccionó la HU más básicas que se debe realizar siendo su implementación un producto que aporta funcionalidades básicas para el proceso de negocio del cliente. La funcionalidad Registrar usuario es fundamental, debido a que es el primer paso que se debe efectuar para poder utilizar el sistema automático. Explorar problemas que devuelve un rango de problemas es la 2da funcionalidad más importante ya que es imprescindible para la identificación de los problemas a resolver y finalmente Responder problema es la 3ra funcionalidad elemental para la interacción del usuario con el CMF.

Su realización llevará un total de 40 días laborables.

Planificación de HU de la 2º Iteración

Una vez que se ha implementado las funcionalidades más básicas y se ha creado un primer producto funcional se procede a desarrollar las HU más importantes para la presente iteración:

- Leer un problema
- Ver comentarios
- Ver ranking
- Posición en el Ranking
- Ultimas sentencias
- Ver sentencias

Su realización llevará un total de 30 días laborables.

Planificación de HU de la 3º Iteración

Capítulo.2 Exploración, Planificación y Diseño del Sistema

Las funcionalidades que se ejecutan de último, pero a la vez importantes para lograr el producto requerido, son las siguientes:

- Manual del CMF
- Configurar cuenta de correo
- Configurar servidor de servicio
- Configurar servidor de envío de correo
- Configurar servidor de descarga de correo
- Configurar puerto de descarga
- Configurar puerto de envío

Su realización llevará un total de 25 días laborables.

Para lograr el cumplimiento en tiempo de las HU se toman las siguientes decisiones:

- Se trabajará 5 horas diarias en la resolución de las HU.
- Se trabajará 5 días en la semana; lunes, martes, miércoles, jueves y viernes (días laborables). El sábado y domingo se descansará (días no laborables). En conclusión, se trabajará 25 horas semanales.

2.9 Plan de entregas

Al término de cada iteración se realizará una pequeña entrega funcional del CMFMail, lo que quiere decir que en el momento de la entrega el producto estará listo para ser puesto en funcionamiento.

Tabla 8: Plan de entregas

Producto	Final 1era Iteración	Final 2da Iteración	Final 3era Iteración
CMFMail	Versión 1.0	Versión 1.1	Versión 1.2

2.10 Diseño

XP propone realizar el diseño del sistema de la forma más sencilla posible y con el mínimo número de ambigüedades. De esta manera se garantiza que los desarrolladores entiendan e implementen las funcionalidades en menos tiempo.

La metodología XP propone crear durante la Fase de Diseño un artefacto llamado Tarjetas CRC (Clase-Responsabilidad-Colaboración). El objetivo de realizar estas tarjetas es definir las clases necesarias para implementar el sistema, su responsabilidad, así como las diferentes clases que colaboran con su funcionamiento. De esta manera es posible fomentar el análisis y discusión entre los miembros del equipo del proyecto con el objetivo de tomar decisiones que contribuyan a mejorar la simplicidad del diseño.

2.11 Patrón arquitectónico

Modelo Vista Controlador, o MVC, es un patrón de arquitectura de software que separa una aplicación en tres componentes o partes diferenciadas: el Modelo, la Vista y el Controlador.

El modelo: representa los datos de una aplicación.

La vista: Contiene todo lo referente a la interacción con el usuario, más conocido como interfaz.

El controlador: Incluye a menudo la lógica que procesa las acciones y eventos generados durante el tiempo de vida de la aplicación.

Es importante destacar que en este sistema va a estar presente el patrón arquitectónico Modelo- Vista- Controlador (MVC) usado por el framework Grails. El CMFMail se integra al CMF adicionando una controladora que gestiona las peticiones a la base de datos. Estas peticiones son realizadas por el daemon que se encarga a su vez de la descarga periódica de correos de la cuenta del Caribbean Mind Forge y de la validación y procesamientos de dichos correos. Emitiendo una respuesta al usuario en formato HTML a través de mensajes.

2.11.1 Patrones de Diseño

“Los patrones de diseño son descripciones de objetos y clases comunicándose, que son adaptadas para resolver un problema de diseño general en un contexto particular”. (24)

General Responsibility Assignment Software Patterns (GRASP): Es un contrato u obligación de una clase. Las responsabilidades están ligadas a las obligaciones de un objeto respecto a su comportamiento.

Capítulo.2

Exploración, Planificación y Diseño del Sistema

Patrón Experto: Determina cuál es la clase que debe asumir una responsabilidad a partir de la información que posee cada una. Estas clases son: “Correo” que es responsable del envío y recepción de correos, “Validar” que es responsable de la validación del asunto dentro de cada mensaje recibido y “Operaciones” que gestiona la petición al servidor y la respuesta al usuario.

Patrón Creador: Resuelve el problema de ¿Quién es el responsable de crear alguna nueva instancia de alguna clase? La clase LeerCorreos lo utiliza ya que es responsable de crear instancias de la clase TareaAutomatica.

Patrón Controlador: Resuelve el problema de ¿Quién debería encargarse de atender un evento del sistema? La clase PreguntaDiaController es la responsable de atender los distintos eventos de gestión de información en la base de datos del CMF.

Patrón Bajo acoplamiento: Medida de la fuerza con que una clase está conectada a otras clases. Resuelve el problema de ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización? Este patrón es utilizado por el framework Grails, y por consiguiente en el sistema CMFMail, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja. El CMFMail separa la respuesta HTML de la clase controladora que gestiona la información de la base de datos del CMF.

Patrón Alta cohesión: Medida de la fuerza que une a las responsabilidades de una clase. Resuelve el problema de ¿Cómo manejar la complejidad dentro de límites manejables? La clase Operaciones depende de la orden emitida por la clase Validar para realizar la operación correcta por lo que sus responsabilidades están estrechamente relacionadas y sus respectivos trabajos son adecuados.

(25)

2.12 Modelo de datos

El diseño de la base de datos es un proceso que requiere extremo cuidado, el principal aspecto a tener en cuenta para la creación de las tablas de una base de datos es la definición correcta de cada uno de sus atributos. Uno de los modelos más utilizados para diseñar bases de datos es el modelo entidad-relación, ya que permite una definición clara y concisa de los esquemas conceptuales y de su visión. Este modelo se encuentra basado en dos conceptos: las entidades, que son objetos sobre los cuales se desea guardar información y las relaciones, que constituyen asociaciones entre entidades.

A continuación se presenta el diseño del modelo de base de datos sobre el cual trabaja el sistema:

Capítulo.2 Exploración, Planificación y Diseño del Sistema

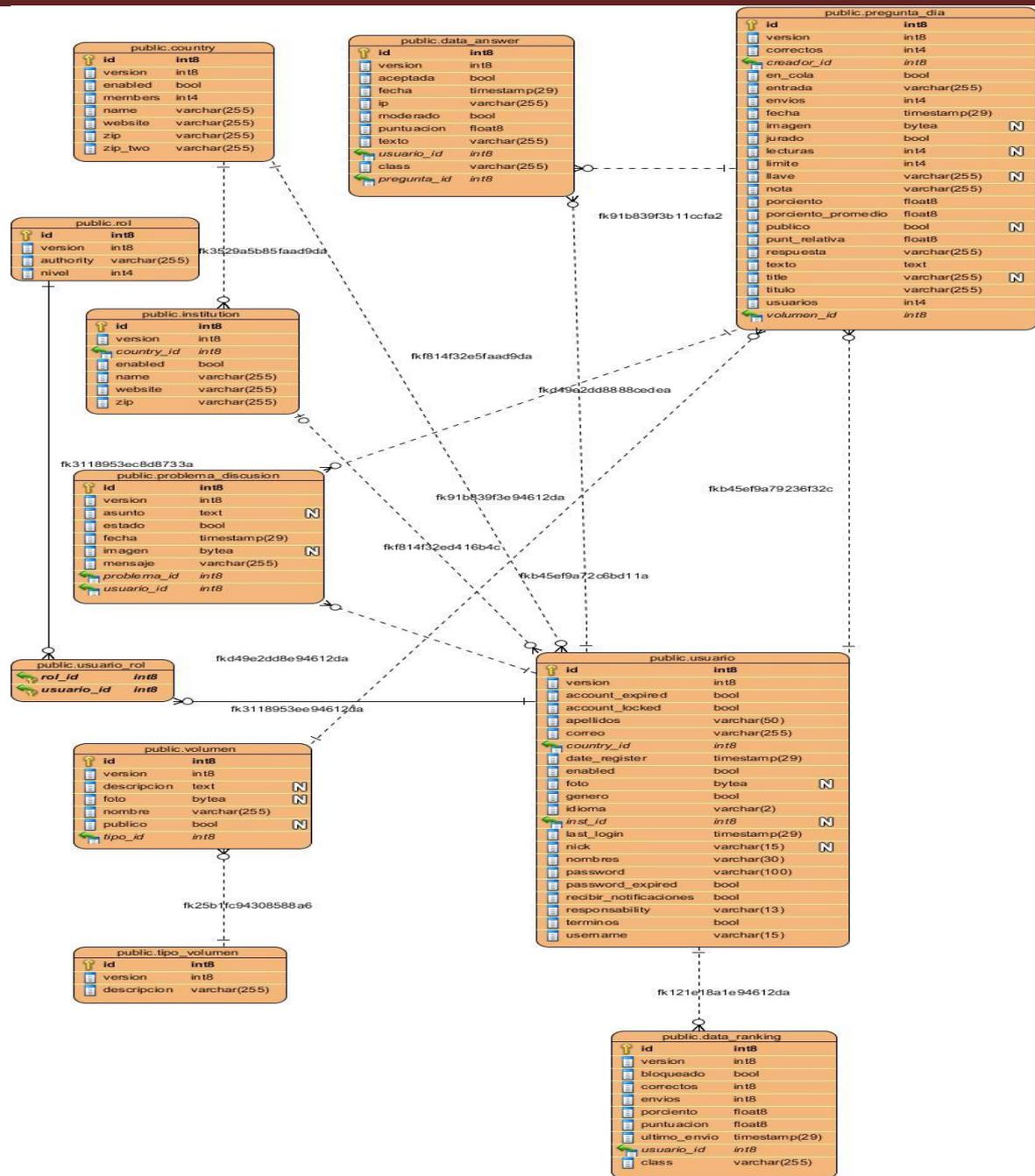


Figura 3: Diagrama Entidad- Relación

Capítulo.2 Exploración, Planificación y Diseño del Sistema

El diagrama de la figura 3 muestra las tablas de la base de datos del CMF sobre las que interactúa el CMFMail para brindar sus servicios. A continuación se evidencia como se efectúa algunas de dichas interacciones:

Cuando se responde un problema la respuesta del usuario es comparada con el campo respuesta de la tabla pregunta_dia. Si las respuestas son iguales se actualiza el campo porciento mediante la fórmula $\text{porciento} = (\text{correctos} * 100) / \text{envios}$ y los campos correctos y envíos a los cuales se les suma uno, campos todos pertenecientes a la tabla pregunta_dia.

Para leer un problema se utiliza la tabla pregunta_dia de la cual se obtienen los campos: id, titulo, texto y lecturas.

Para ver las últimas sentencias del usuario se utiliza la tabla pregunta_respuesta de la cual se obtienen los campos: id, aceptada y texto.

Para ver los comentarios de un problema se utiliza la tabla problema_discusion de la cual se obtienen los campos: usuario y mensaje.

Para ver el ranking del usuario se utiliza la tabla data_answer de la cual se obtienen los campos: usuario y puntuación.

2.13 Tarjetas CRC

A continuación se expone de manera diferenciada una selección de las Tarjetas CRC de la aplicación a desarrollar.

Tabla 9: Plantilla de Tarjetas CRC

Tarjetas CRC	
Nombre de la clase: Nombre de la clase que está modelando.	
Responsabilidades: Descripción de alto nivel de las responsabilidades	Colaboradores: Indica con cuáles otras clases requiere

de la clase.	relación para cumplir su responsabilidad.
--------------	---

Tabla 10: Tarjeta CRC LeerCorreos

Tarjetas CRC	
Nombre de la clase: LeerCorreos	
Responsabilidades: Clase principal. Ejecuta a la clase TareaAutomatica cada 10 segundos	Colaboradores: TareaAutomatica.

Tabla 11: Tarjeta CRC TareaAutomatica

Tarjetas CRC	
Nombre de la clase: TareaAutomatica	
Responsabilidades: Copiar correos de CMFmail a CMFmailstore Obtener correos de la carpeta CMFmail Borrar correos de CMFmail Llamar al método val de la clase Validar para Validar el asunto de cada mensaje Llamar al método ProcesarOrden de la Clase Operaciones para consumir el servicio correspondiente a la sintaxis del asunto validado	Colaboradores: Correo Validar Operaciones

Tabla 12: Tarjeta CRC Validar

Tarjetas CRC	
Nombre de la clase: Validar	
Responsabilidades: Valida que la sintaxis del asunto esté correctamente escrito y emite una orden a ejecutar como resultado	Colaboradores:

Tabla 13: Tarjeta CRC Operaciones

Tarjetas CRC	
Nombre de la clase: Operaciones	
Responsabilidades: Consume el servicio correspondiente a la orden emitida por la clase Validar Emite respuestas al usuario con la petición hecha o con un mensaje de error.	Colaboradores: Correo.

Tabla 14: Tarjeta CRC Email

Tarjetas CRC	
Nombre de la clase: Email	
Responsabilidades: Representa a un correo enviado por el usuario	Colaboradores:

Tabla 15: Tarjeta CRC Especial

Tarjetas CRC	
Nombre de la clase: Especial	
Responsabilidades: Clase que permite cambiar caracteres extraños por palabras claves y a su vez cambiar palabras claves por los caracteres extraños correspondientes.	Colaboradores:

Tabla 16: Tarjeta CRC Vistas

Tarjetas CRC	
Nombre de la clase: Vistas	
Responsabilidades: Posee todas las vistas HTML de respuesta al usuario y los métodos para devolverlas de acuerdo al caso que corresponda.	Colaboradores:

Tabla 17: Tarjeta CRC Instalador CMFMail

Tarjetas CRC	
Nombre de la clase: InstaladorCMFMail	
Responsabilidades: Es la clase principal del instalador y se encarga de visualizar la ventana de instalación del CMFMail	Colaboradores: Instalador1.

Tabla 18: Tarjeta CRC Instalador1

Tarjetas CRC	
Nombre de la clase: Instalador1	
Responsabilidades: Visualiza un botón para comenzar la instalación del programa copiando los archivos a la carpeta de usuario en una carpeta nombrada CMFMail.	Colaboradores: CopiarDirectorio.

Tabla 19: Tarjeta CRC CopiarDirectorio

Tarjetas CRC	
Nombre de la clase: CopiarDirectorio	
Responsabilidades: Copia todos los archivos del programa a un nueva carpeta que se crea en el directorio de usuario nombrada CMFMail	Colaboradores:

Tabla 20: Tarjeta CRC Instalador2

Tarjetas CRC	
Nombre de la clase: Instalador2	
Responsabilidades: Pregunta si desea ejecutar el CMFMail ahora o si no y termina el instalador	Colaboradores:

Tabla 21: Tarjeta CRC FormulaCMF

Tarjetas CRC	
Nombre de la clase: FormulaCMF	
Responsabilidades: Clase principal del formulario de registro. Su funciones visualizar la ventana de registro.	Colaboradores: Formulario2

Tabla 22: Tarjeta CRC Formulario2

Tarjetas CRC	
Nombre de la clase: Formulario2	
Responsabilidades: Valida la entrada correcta en los campos del formulario a través de expresiones regulares, procesa los datos organizándolos para el envío por correo y encripta la contraseña con el algoritmo RSA.	Colaboradores: Especial

Tabla 23: Tarjeta CRC PreguntaDia

Tarjetas CRC	
Nombre de la clase: PreguntaDia	
Responsabilidades: Brinda servicios para la gestión con la base de datos del CMF	Colaboradores:

2.14 Conclusiones

En este capítulo se inició el proceso de desarrollo de la solución, mediante las fases de Planificación y Diseño propuestas por XP. A continuación se muestran los resultados de mayor relevancia obtenidos luego de concluir estas dos fases:

- Se describieron cada una de las funcionalidades del sistema por medio de las HU y se estimó el tiempo que llevaría la implementación de las mismas.
- Se dividió el proyecto en tres iteraciones.
- Se precisó que al final de cada iteración se realizará una pequeña entrega del sistema con el fin de que una vez concluido el proceso de desarrollo el sistema ya esté listo para ser usado.
- Se controla la velocidad del proyecto, a través de lo cual se determina cuantas HU serán implementadas en cada iteración. Tal y como propone XP se realizó un diseño simple y sin ambigüedades de las clases del sistema.

Capítulo.3 Implementación y Pruebas

3.1 Introducción

Luego de haberse definido las HU y las Tarjetas CRC propias de la metodología XP, descritas en el capítulo anterior, el equipo de desarrollo cuenta con suficiente conocimiento sobre el sistema y está en condiciones de comenzar la implementación de las HU según la planificación concebida. Paralelamente al desarrollo de las funcionalidades y también después de esto se realizarán las pruebas para detectar y corregir posibles errores.

El objetivo de este capítulo es describir las Fases de Implementación y Pruebas que plantea el proceso XP, dando cumplimiento al desarrollo del sistema. Al final del capítulo se presentan los resultados obtenidos en ambas fases.

3.2 Implementación

La Fase de Implementación es el espacio dentro del ciclo de vida del proyecto donde el equipo de desarrollo implementa las funcionalidades que fueron especificadas en la Fase de Exploración a través de las HU.

3.2.1 Tareas de implementación

XP propone realizar unas tarjetas nombradas Tareas de implementación, donde se exponga brevemente cuales fueron las tareas realizadas para implementar las HU. Cada Tarea de implementación estará vinculada a una HU; de esta forma se comprobará el cumplimiento de todas las funcionalidades del sistema. A continuación se muestra la planilla empleada para las Tareas de implementación.

Tabla 24: Plantilla de las Tareas de implementación

Tarea de implementación	
Número de tarea: Número de la tarea.	Número de HU: Número de HU a la cual va dirigida la tarea.
Nombre de la tarea: Nombre de la tarea a desarrollar.	
Tipo de tarea: Tipo de tarea. Configuración. Desarrollo, entre otros.	Estimación: Tiempo que se estime para el desarrollo de la tarea.

Fecha de Inicio: Fecha en que se inicia el desarrollo de la tarea.	Fecha de Fin: Fecha en que se le da cumplimiento a la tarea.
Programador Responsable: Autor de la tarea.	
Descripción: Breve descripción de la tarea.	
Clases: Nombre de las clases implementadas.	

A continuación las Tareas de implementación por cada HU.

Tabla 25: Tarea de implementación Configurar cuenta de correo

Tarea de implementación	
Número de tarea: 1.	Número de HU: 9.
Nombre de la tarea: Configurar cuenta de correo.	
Tipo de tarea: Desarrollo.	Estimación: Tres semanas
Fecha de Inicio: 1 de septiembre de 2016.	Fecha de Fin: 9 de septiembre de 2016.
Programador Responsable: Eduardo Pereira Dennes.	
Descripción: Crear un clase y configurarla para el envío y recepción de correo empleando los protocolos de correo soportados por el servidor de correo de la UCI.	
Clases: Correo.	

Tabla 26: Tarea de implementación Crear servicios.

Tarea de implementación	
Número de tarea: 2.	Número de HU: 1.
Nombre de la tarea: Crear servicios.	

Tipo de tarea: Desarrollo.	Estimación: Tres semanas.
Fecha de Inicio: 12 de septiembre de 2016.	Fecha de Fin: 20 de septiembre de 2016.
Programador Responsable: Eduardo Pereira Dennes.	
Descripción: Crear una controladora llamada PreguntaDia y agregarle 3 acciones. Cada acción deberá implementar uno de los siguientes requerimientos: 1-Explorar problemas con imágenes del CMF definiendo un rango no mayor de 25. Debe devolver los siguientes campos de la tabla de usuarios: id, título, texto, entrada, envíos, correctos, creador, lecturas y fecha. Solo se efectúa si el usuario que la invoca está registrado. 2-Responder problemas comparando la respuesta en el cuerpo del mensaje del usuario con la respuesta en la base de datos, usando el id del problema para identificarlo, el cual vendrá en el asunto del correo. 3-Registrar usuario recopilando mediante un formulario los siguientes datos: usuario, contraseña, nombres, apellidos, correo, género, país e idioma. Clases: PreguntaDia.	

Las Tareas de implementación restantes pueden ser consultadas en los Anexos del documento.

3.3 Pruebas

Las pruebas son un elemento importante dentro del ciclo de vida de un proyecto guiado por XP. Desde luego las pruebas permiten la detección de errores y eliminación posterior de los mismos, a la vez que proporcionan una medida de la calidad del software.

La metodología XP propone realizar pruebas unitarias y pruebas de aceptación. A continuación se describen las características de estas pruebas, así como los resultados obtenidos.

3.3.1 Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, las pruebas deben ser definidas antes de realizar el código ("Test-driven programming"). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser

utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo. (26)

3.3.2 Resultados

A continuación se presentan los resultados de las pruebas unitarias realizadas al código por iteración. Se realizaron los test⁴ utilizando TestNG como framework de prueba unitaria, los cuales arrojaron errores que fueron corregidos hasta llegar a un producto libre de los mismos.

Iteración 1

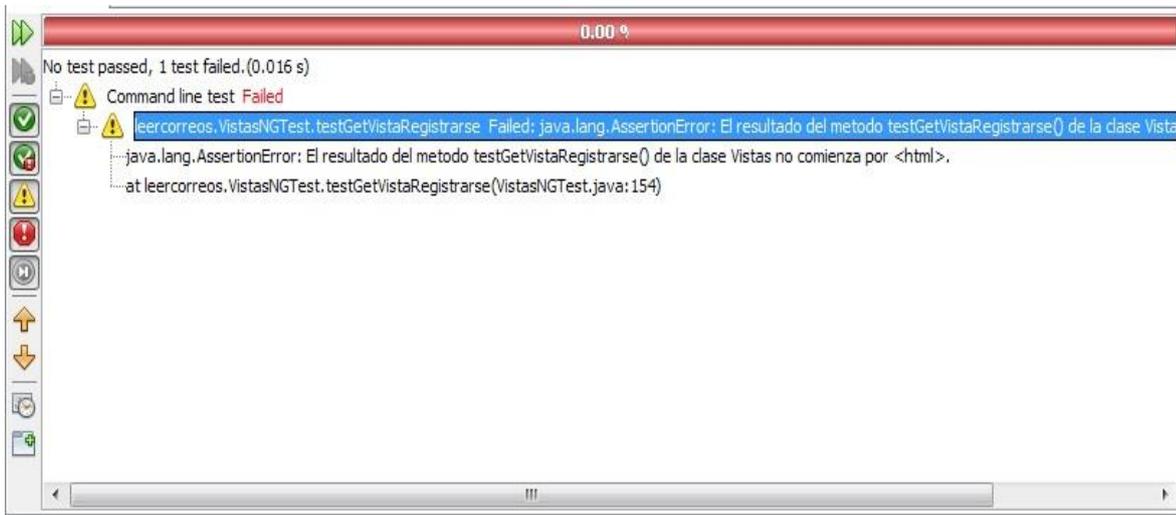


Figura 4: Resultado de la prueba unitaria a la clase Vista.

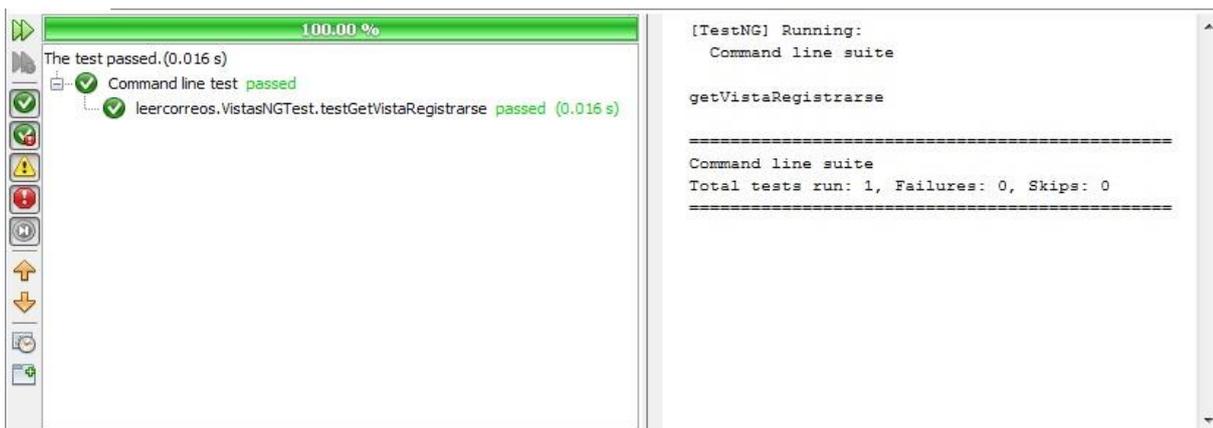


Figura 5: Resultado corregido de la prueba unitaria a la clase Vista.

⁴ Prueba



Figura 6: Resultado de la prueba unitaria a la clase Validar.



Figura 7: Resultado corregido de la prueba unitaria a la clase Validar.

Iteración 2



Figura 8: Resultado de la prueba unitaria a la clase Vistas.

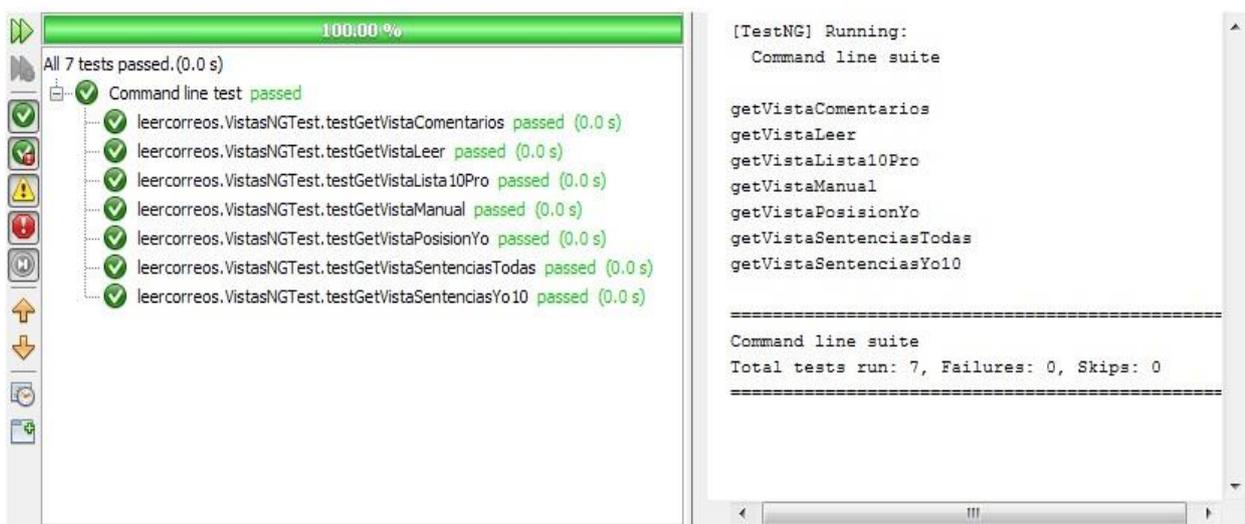


Figura 9: Resultado corregido de la prueba unitaria a la clase Vistas.



Figura 10: Resultado de la prueba unitaria a la clase Validar.

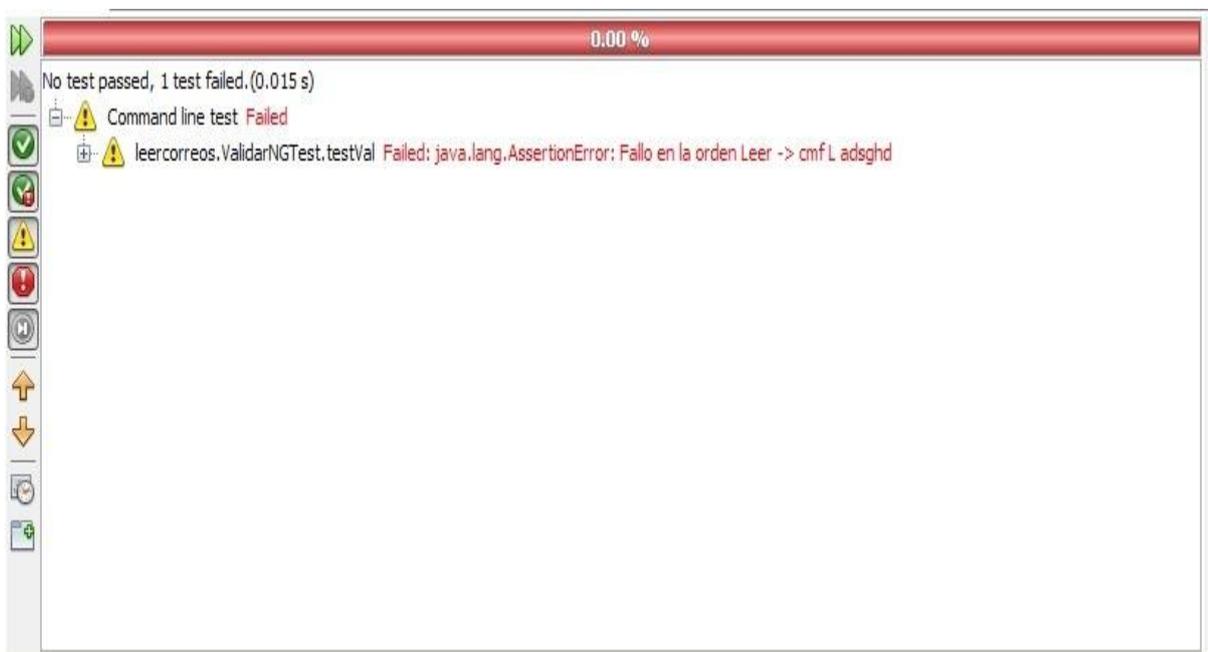


Figura 11: Resultado de la prueba unitaria a la clase Validar.



Figura 12: Resultado de la prueba unitaria a la clase Validar.



Figura 13: Resultado corregido de la prueba unitaria a la clase Validar.

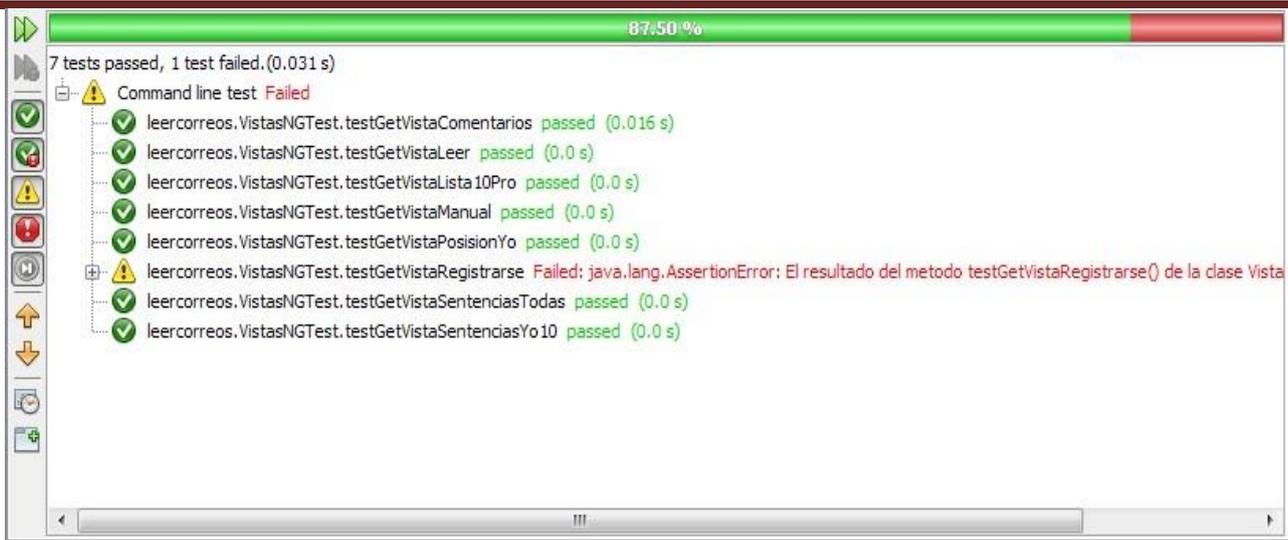


Figura 14: Resultado de la prueba unitaria a la clase Vistas.

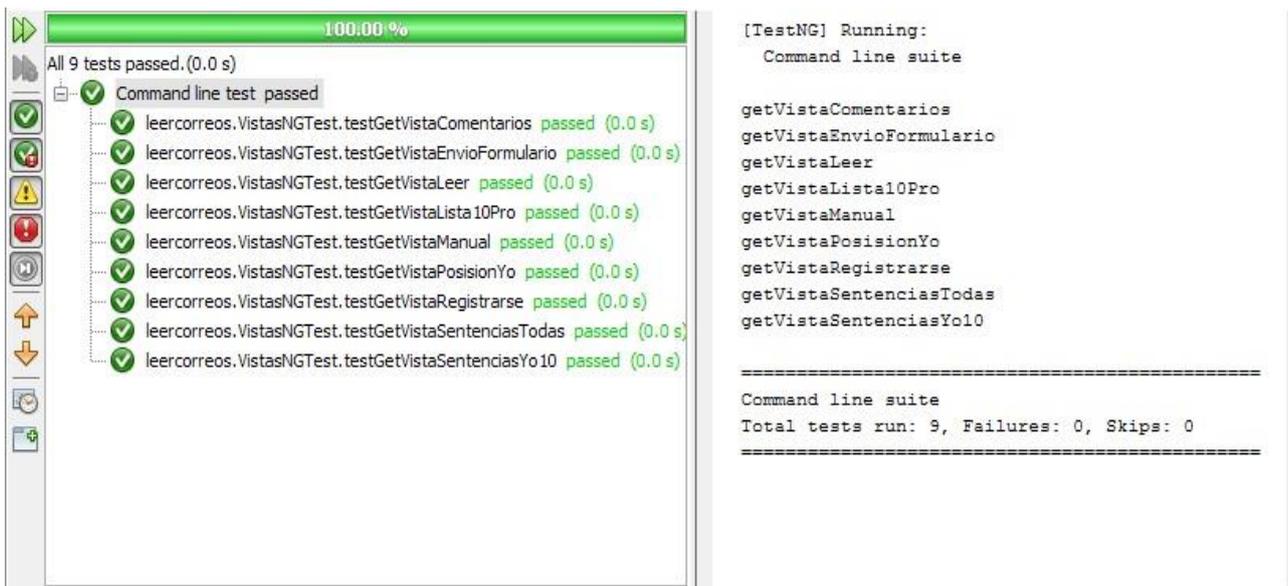


Figura 15: Resultado corregido de la prueba unitaria a la clase Vistas.

3.3.3 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. Las pruebas de aceptación son consideradas como “pruebas de caja negra” (“Black box system tests”). Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. (16)

Capítulo.3 Implementación y Pruebas

Los representantes de los clientes participan en el desarrollo y son los responsables de definir las pruebas de aceptación del sistema. (16)

Tabla 27: Plantilla Pruebas de Aceptación.

Caso de prueba de aceptación	
Número: Número de la HU.	Nombre: Nombre de la HU.
Prioridad: Cuán importante es para el cliente o el equipo de desarrollo. Alta. Media. Baja	Complejidad: Nivel de dificultad para el desarrollador. Alta. Media. Baja.
Iteración: Iteración a la que corresponde.	Estimación: Tiempo estimado para realizar la actividad
Descripción: Breve reseña de la HU, describiendo las acciones del usuario.	
Notas: Observaciones de interés.	
No conformidades: Errores encontrados en las pruebas realizadas.	
Resultado: Resultado obtenido al realizar la prueba.	

Iteración 1

Tabla 28: Prueba de aceptación Registrar usuario

Caso de prueba de aceptación	
Número: 1	Nombre: Registrar Usuario.
Prioridad: Alta	Complejidad: Media.
Iteración: 1.	Estimación: 1 semanas
Descripción: Se registra un nuevo usuario	
Notas: ninguna.	
No conformidades: ninguna	

Resultado: El registro se realiza correctamente

Tabla 29: Prueba de aceptación Rango de problemas

Caso de prueba de aceptación	
Número: 2	Nombre: Rango de problemas.
Prioridad: Alta	Complejidad: Media.
Iteración: 1.	Estimación: 1 semanas
Descripción: Permite ver un rango de problemas	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: El rango de problemas es devuelto correctamente.	

Tabla 30: Prueba de aceptación Responder problema

Caso de prueba de aceptación	
Número: 3	Nombre: Responder problema.
Prioridad: Alta	Complejidad: Media.
Iteración: 1.	Estimación: 1 semanas
Descripción: Se responde a un problema.	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: El problema es respondido correctamente.	

Iteración 2

Tabla 31: Prueba de aceptación Leer un problema

Caso de prueba de aceptación	
Número: 4	Nombre: Leer un problema.
Prioridad: Alta	Complejidad: Media.
Iteración: 2.	Estimación: 1 semanas
Descripción: Se lee un problema	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: El problema se lee correctamente.	

Tabla 32: Prueba de aceptación Ver Comentarios

Caso de prueba de aceptación	
Número: 5	Nombre: Ver Comentarios.
Prioridad: Alta	Complejidad: Media.
Iteración: 2.	Estimación: 1 semanas
Descripción: Devolver los comentarios de un problema.	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se devuelven correctamente los comentarios de un problema	

Tabla 33: Prueba de aceptación Ranking

Caso de prueba de aceptación

Capítulo.3 Implementación y Pruebas

Número: 6	Nombre: Ranking.
Prioridad: Alta	Complejidad: Media.
Iteración: 2.	Estimación: 1 semanas
Descripción: Devuelve el ranking de los primeros 10 usuarios	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se devuelve el ranking de los primeros 10 usuarios correctamente.	

Tabla 34: Prueba de aceptación Posición en el ranking

Caso de prueba de aceptación	
Número: 7	Nombre: Posición en el ranking
Prioridad: Alta	Complejidad: Media.
Iteración: 2.	Estimación: 1 semanas
Descripción: Devuelve la posición del usuario en el ranking el CMF	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se devuelve la posición del usuario en el ranking del CMF correctamente.	

Tabla 35: Prueba de aceptación Últimas sentencias

Caso de prueba de aceptación	
Número: 8	Nombre: Últimas sentencias.
Prioridad: Alta	Complejidad: Media.
Iteración: 2.	Estimación: 1 semanas

Descripción: Devuelve las últimas sentencias del usuario.
Notas: ninguna.
No conformidades: ninguna
Resultado: Se devuelve las últimas sentencias del usuario correctamente.

Tabla 36: Prueba de aceptación Sentencias

Caso de prueba de aceptación	
Número: 9	Nombre: Sentencias.
Prioridad: Alta	Complejidad: Media.
Iteración: 2.	Estimación: 1 semanas
Descripción: Se devuelve un rango de sentencias	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se devolvió el rango de sentencias correctamente.	

Iteración 3

Tabla 37: Prueba de aceptación Manual del CMF

Caso de prueba de aceptación	
Número: 10	Nombre: Manual del CMF.
Prioridad: Alta	Complejidad: Media.
Iteración: 3.	Estimación: 1 semanas
Descripción: Devuelve el Manual del CMF	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se devolvió el Manual del CMF correctamente.	

Tabla 38: Prueba de aceptación Configurar cuenta de correo

Caso de prueba de aceptación	
Número: 11	Nombre: Configurar cuenta de correo.
Prioridad: Alta	Complejidad: Media.
Iteración: 3.	Estimación: 1 semanas
Descripción: Se configura la cuenta de correo	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se configuró la cuenta de correo correctamente.	

Tabla 39: Prueba de aceptación Configurar servidor de servicios

Caso de prueba de aceptación	
Número: 12	Nombre: Configurar servidor de servicios.
Prioridad: Alta	Complejidad: Media.
Iteración: 3.	Estimación: 1 semanas
Descripción: Se configura el servidor de servicios	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se configuró el servidor de servicios correctamente.	

Tabla 40: Prueba de aceptación Configurar servidor de envío de correo

Caso de prueba de aceptación	
Número: 13	Nombre: Configurar servidor de envío de correo.
Prioridad: Alta	Complejidad: Media.
Iteración: 3.	Estimación: 1 semanas
Descripción: Se configura el servidor de envío de correo	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se configuró el servidor de envío de correo correctamente.	

Tabla 41: Prueba de aceptación Configurar servidor de descarga de correo

Caso de prueba de aceptación	
------------------------------	--

Capítulo.3 Implementación y Pruebas

Número: 14	Nombre: Configurar servidor de descarga de correo.
Prioridad: Alta	Complejidad: Media.
Iteración: 3.	Estimación: 1 semanas
Descripción: Se configura el servidor de descarga de correo	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se configuró el servidor de descarga de correo correctamente.	

Tabla 42: Prueba de aceptación Configurar puerto de descarga de correo

Caso de prueba de aceptación	
Número: 15	Nombre: Configurar puerto de descarga de correo.
Prioridad: Alta	Complejidad: Media.
Iteración: 3.	Estimación: 1 semanas
Descripción: Se configura el puerto de descarga de correo	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se configuró el puerto de descarga de correo correctamente.	

Tabla 43: Prueba de aceptación Configurar puerto de envío de correo

Caso de prueba de aceptación	
Número: 16	Nombre: Configurar puerto de envío de correo.

Capítulo.3 Implementación y Pruebas

Prioridad: Alta	Complejidad: Media.
Iteración: 3.	Estimación: 1 semanas
Descripción: Se configura el puerto de envío de correo	
Notas: ninguna.	
No conformidades: ninguna	
Resultado: Se configuró el puerto de envío de correo correctamente.	

3.4 Conclusiones

En este capítulo se continuó con el proceso de desarrollo de la solución con la aplicación de las Fases de Implementación y Pruebas. A continuación se muestran los resultados de mayor relevancia obtenidos luego de concluir estas dos fases:

- Se implementaron las HU definidas por el cliente, dejando constancia de este proceso a través de la Tareas de implementación.
- Se ejecutaron pruebas unitarias al código del CMFMail, las cuales ayudaron a corregir varios errores de implementación surgidas en el proceso de desarrollo.
- Se llevaron a cabo pruebas de aceptación donde el cliente comprobó el cumplimiento de las historias de usuario.

Conclusiones

La presente investigación se centró en la concepción y el posterior desarrollo de un sistema automático de respuesta por correo denominado CMFMail. Esta herramienta informática representa una solución económicamente factible para aquellas instituciones o personas que desean o necesitan trabajar con el Juez en Línea Caribeño y que por sus limitaciones de acceso a Internet no pueden hacerlo.

En el transcurso de la investigación se dio cumplimiento a los objetivos propuestos inicialmente y se arribó a las siguientes conclusiones:

- Se comprobó que es posible utilizar los protocolos SMTP e IMAP4 para la gestión de correos en el CMFMail.
- Se elaboró la Fundamentación Teórica de la investigación.
- Se definieron 17 funcionalidades que debía cumplir el sistema.
- Se implementó el sistema informático y se le realizaron pruebas unitarias y de aceptación para comprobar su funcionamiento. Se obtuvieron índices aceptables para cada prueba realizada, por lo que se considera un resultado general positivo.

Recomendaciones

- Agregar funcionalidades de mayor complejidad que por el alcance del trabajo no fueron implementadas en el CMFMail como es la participación en concursos.
- Realizar pruebas de integración.
- Realizar pruebas de carga y estrés.

Bibliografía

1. **Jacobson, Ivar.***El Proceso Unificado de Desarrollo de Software*. s.l. : Addison-Wesley, 2007. ISBN-10: 8478290362/ISBN-13: 978-8478290369.
2. **CSIC, Instituto de Física Aplicada del.** Características del lenguaje Java. [Online] 1999. [Cited: enero 15, 2014.] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>..
3. **Sitio Web Oficial de Lenguaje Unificado de Modelado.** Lenguaje Unificado de Modelado (UML). [Online] [Cited: enero 14, 2014.] <http://www.uml.org>..
4. **Foundation., Apache Software.** Apache Tomcat. *Sitio Web Oficial de Apache Tomcat*. [Online] [Cited: enero 13, 2014.] <http://www.apachetomcat.org>..
5. **Sitio Web Oficial de PostgreS QL.**The PostgreSQL Global Development Group. [Online] 1996-2013. [Cited: enero 13, 2014.] <http://www.postgresql.org>..
6. **Sitio oficial de Visual Paradigm.**Boost Productivity with innovative and intuitive Technologies. [Online] [Cited: enero 25, 2014.] <http://www.visual-paradigm.com>.
7. **Netbeans.** Oracle Corporation and affiliates or its NetBeans. [Online] 2000. [Cited: enero 25, 2014.] http://netbeans.org/community/releases/68/index_es.html.
8. **UNOBRAIN.** unobrain. [Online] 2012. [Cited: Diciembre 10, 2015.] www.unobrain.com.
9. Stallings, William. *Comunicaciones y Redes de Computadoras*. S.l.: Prentice Hall, 2012. pág. 54.
10. Tutorial y descripción técnica de TCP/IP. [En línea] [Citado el: 13 de Diciembre de 2015.] http://www.cicei.com/ocon/gsi/tut_tcpip/3376imap.html
11. David Wagner, Bruce Schneier. *Analysis of the SSL 3.0 protocol*. 1997.
12. Alonso-Rodriguez, Antonio Jesus Caro. *MAN IN THE MIDDLE ATTACKS ON SSL/TLS*. 2013. Tesis de Maestría.
13. Extreme Programming: A gentle introduction. [En línea] [Citado el: 7 de Diciembre de 2015.] <http://www.extremeprogramming.org/index.html>.
14. Apache Software Foundation. [En línea] [Citado el: 13 de 6 de 2013.] <http://apachefoundation.wikispaces.com/Apache+Tomcat>.
15. **Acevedo, Luis Felipe López.** <http://www.ask.com>. [En línea] [Citado el: 20 de 1 de 2016.]
16. **E, IAN SOMMERVILL.***Ingeniería del software*. Madrid : PEARSON EDUCACIÓN, 2005. 84-7829-074-5.

17. **Dantas, Rhawi.** [En línea] mayo de 2011. [Citado el: 11 de 3 de 2016.] www.packtpub.com. 978-1-849512-50-3.
18. [book auth.] Javier García de Jalón, et al., et al. *Aprenda Java como si estuviera en primero*. San Sebastian : s.n., 2000.
19. **Palos, Juan Antonio.** Programacion en Castellano. [Online] [Cited: 4 3, 2015.] <http://www.programacion.com>.
20. Mozilla. [En línea] Mozilla, 2005-2013. [Citado el: 18 de Enero de 2013.] <https://www.mozilla.org/es-ES/thunderbird/features/>
21. **Informática, Instituto de Investigaciones Biológicas Clemente Estable (IIBCE) -.** Manual básico para usuarios de . 5 2013. 2.
22. **Barchini, Graciela Elisa.** *Métodos "I + D" de la Informática*. Santiago del Estero : Revista de Informática Educativa y Medios Audiovisuales, 2005. ISSN 1667-8338.
23. **Penadés, Patricio Letelier y M^a Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* . Valencia : Laboratorio de Sistemas de Información.
24. **Díaz., Moisés D.** Diseño de aplicaciones internet usando los Patrones de diseño J2EE. *moisesdaniel*. [Online] [Cited: 10 6, 2006.] www.moisesdaniel.com.
25. *Diseño orientado a objetos. Patrones de diseño y diseño de la Base de Datos.* **Software, Departamento de Ingeniería de.** La Habana : Universidad de las Ciencias Informáticas, 2013.
26. **Joskowicz, Ing. José.** *Reglas y Prácticas en eXtreme Programming* . Vigo : s.n., 2008.
27. Fleitas, Nellis Margarita Cabrera Mallea y Michael Horta. Alternativa para el uso del Juez en Línea Caribeño en. [Documento] La Habana : Universidad de las Ciencias Informáticas, 2013.
28. moodle. [Online] [Cited: 3 6, 2017.] moodle2016_17.ua.es.
29. **Brito, Nacho.** *Manual de desarrollo web con Grails*. [Documento] s.l. : ImaginaWork, 2009. 978-84-613-2651.
30. UVA Online Judge. [En línea] [Citado el: 15 de junio de 2017.] <http://uva.onlinejudge.org>.
31. Sphere Online Judge (SPOJ). [En línea] [Citado el: 15 de junio de 2017.] <http://www.spoj.com>.

Referencias Bibliográficas

Product: Rational Software Corporation. 2002. *Rational Software Corporation*. 2002.

Roberto Hernández Sampieri, Carlos Fernández, Collado, Pilar Baptista Lucio. 2006. *Metodología de la Investigación Científica*. Cuarta edición digital. s.l. : McGraw Hill, 2006. p. 882.