



**Facultad 4**

*Implementación y prueba de un Jurado Online sobre correo electrónico*

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

**Autores**

**Nersa Doraines Acosta Labrada**

**Tutores**

**Ing. Jorge Amado Soria Ramírez**

**Ing. Enrique José Altuna Castillo**

**La Habana, junio de 2011**

**“Año 53 de la Revolución”**

### **Declaración de autoría**

Declaro que soy la única autora del trabajo titulado: **“Implementación y prueba de un Jurado Online sobre correo electrónico”** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Nersa Doraines Acosta Labrada    Tutor: Jorge Amado Soria Ramírez    Tutor: Enrique José Altuna Castillo

*“La alegría está en la lucha, en el esfuerzo, en el sufrimiento que supone la  
lucha y no en la victoria misma. Nuestra recompensa se encuentra en el  
esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa”*

*Mohandas Karamchand Gandhi.*

*Dedicatoria*

*A mi abuelo Acacio, que siempre ha sido un padre para mí, que me malcrió, porque ha estado pendiente en cada uno de los momentos de mi vida. Espero que dure 120.*

*A mi abuela Toty, que aunque no pudo verme graduada se que estaba muy orgullosa de mí y sabía que yo iba a lograrlo, por ser una madre para mí.*

*A mi mamita, por ser tan especial y simpática, por su dedicación y apoyo, por hacer que me sienta orgullosa de que sea mi mamá, porque nadie como ella para merecerse este título.*

*A mi papa, por ser un ejemplo para mí, por estar ahí, por inculcar en mí valores morales con el ejemplo y no con la palabra.*

*A mi hermano el cabezón porque no importa que nos hayamos pasado la vida entera fajándonos, siempre nos hemos tenido el uno al otro y nos queremos en la misma medida que nos fajamos, porque estoy muy orgullosa de él y se que siempre ha disfrutado de mis logros más que yo misma.*

*A todas las personas que me han apoyado:*

*A toda mi familia. Mi padrastro que es como un padre para mí. Mi hermana que me obliga a ser mejor cada día para darle un buen ejemplo. Mi prima que ha sido siempre mi hermana y compañera de parrandas desde que no sabíamos ni hablar. Mi hermano Andy que desde el pre cargamos el uno con el otro en la escuela.*

*A mis hermanitas Nani, Yamile y Mailin por haberse convertido en hermanas porque juntas compartimos muchos momentos agradables.*

*A mi coquito prieto por su comprensión durante todo este curso, por su amor, por su dedicación, por su compañía, por ser quien es.*

*A las personas que me ayudaron sin protestar para la confección de esta tesis: micha, nidelso, milena, manolo, osmel, en especial mi compañera de tesis Anabel, que me abandonó en la recta final para coger unas vacaciones por los países hermanos. A los que se me quedan.*

*A mis tutores, en especial amado, por proponerme el tema y aguantarme durante todo el trayecto con mis dudas y pesadeces, por ser amigo.*

*A mis amigos de estos 5 años. Los que han compartido conmigo en fiestas, en estudios, en los ensayos para los festivales, en fin, en mi estancia por esta universidad.*

*A mis amistades de afuera.*

## **Resumen**

Los jurados en línea son aplicaciones web disponibles todo el tiempo, que automatizan la evaluación de las respuestas a los problemas de programación de competencia. Estos sistemas se asocian normalmente con entornos académicos (en especial universidades). En la generalidad de los entornos universitarios cubanos no se pueden aprovechar bien estos sistemas debido a que la disponibilidad de la red está sujeta a intermitencias o su velocidad es muy baja.

El presente trabajo tiene como objetivo la implementación de un jurado online con intercambio de información sobre protocolos de correo electrónico que provea el conjunto mínimo de funcionalidades de un jurado online estándar y genere información en formato PDF. Este formato es el estándar de visualización de documentos, mientras que el correo, al ser más desconectado que la comunicación por el protocolo HTTP (Hypertext Transfer Protocol) permite mantener una conexión más estable aun con dificultades de conectividad.

Para conciliar la condición de conexión constante necesaria para el uso del jurado online con las posibilidades disminuidas, se estudiaron las metodologías, herramientas y tecnologías de desarrollo de software a emplear durante la implementación. Por último, se obtuvo un sistema que responde a los objetivos planteados, permitiendo la interacción con el usuario a través del correo electrónico.

## **Abstract**

Online judges are web applications available all the time, which automate the evaluation of solutions to programming problems. These systems are usually associated with academic environments (especially universities). Most Cuban universities cannot take advantage of these systems because their network availability is subject to intermittency or latency.

This work aims at implementing an online judge that exchanges information over e-mail protocols, that provides the minimum set of functionalities of a standard online judge and generate information in PDF format. This format is the standard for document viewing, while e-mail being more disconnected than HTTP (Hypertext Transfer Protocol) and thus permits a more stable connection even with connectivity issues.

To reconcile the condition of constant connection needed to use the online judge with the diminished connection possibilities it was necessary to study different methodologies, tools and software development technologies in order to use them for development. Finally, a system that allows interaction with the user via email was built, thus achieving the declared goals of this paper.

**Índice de tablas**

Tabla 1 Perfil tecnológico de desarrollo ..... 19

Tabla 2 Secciones a probar en el CU Consultar el estado del servidor ..... 40

Tabla 3 Descripción de la variable caso de prueba CU Consultar el estado del servidor ..... 42

Tabla 4 Matriz de datos caso de prueba CU Consultar el estado del servidor..... 44

Tabla 5 Secciones a probar en el CU Ver problema ..... 45

Tabla 6 Descripción de la variable caso de prueba CU Ver problema..... 46

Tabla 7 Matriz de datos caso de prueba CU Ver problema ..... 47

Tabla 8 Secciones a probar en el CU Someter solución de un problema ..... 48

Tabla 9 Descripción de la variable caso de prueba CU Someter solución de un problema ..... 50

Tabla 10 Matriz de datos caso de prueba CU Someter solución de un problema..... 51

Tabla 11 No conformidades del caso de prueba CU Consultar el estado del servidor..... 52

Tabla 12 No conformidades del caso de prueba CU Someter solución de un problema ..... 53

Tabla 13 No conformidades del caso de prueba CU Consultar ranking ordenado..... 53

Tabla 14 No conformidades del caso de prueba CU Consultar el estado de un problema ..... 53

Tabla 15 No conformidades del caso de prueba CU Consultar listado de problemas ..... 54

Tabla 16 No conformidades del caso de prueba CU Modificar información de usuario ..... 54

Tabla 17 No conformidades del caso de prueba CU Ver solución de problema ..... 54

Tabla 18 No conformidades del caso de prueba CU Ver ayuda ..... 55



**Índice de figuras**

Figura 1 Ejemplo de cómo ver un problema .....	24
Figura 2 Modelo de despliegue .....	31
Figura 3 Diagrama de componentes general .....	33
Figura 4 Diagrama de componentes del paquete beans .....	34
Figura 5 Diagrama de componentes del paquete comandos .....	35
Figura 6 Diagrama de componentes del paquete vistas.....	36
Figura 7 Cantidad de no conformidades por caso de uso .....	52

**Índice**

Introducción .....	1
Capítulo 1 .....	5
1.1    Jurados online de programación .....	5
3.1.1    UVa .....	6
3.1.2    SPOJ .....	6
3.1.3    PKU .....	7
3.1.4    Xtreme .....	8
3.1.5    CPAV .....	9
1.2    Metodologías de desarrollo .....	9
1.2.1    XP .....	10
1.2.2    Scrum .....	10
1.2.3    RUP .....	11
1.3    Herramientas CASE .....	12
1.3.1    Rational Rose .....	12
1.3.2    Visual Paradigm for UML .....	13
1.4    Herramientas de desarrollo .....	13
1.5    Protocolos de correo .....	14
1.5.1    POP3 .....	14
1.5.2    IMAP4 .....	15
1.6    Tecnología para el envío y recepción de correos .....	15
1.7    Tecnologías para la generación de documentos en formato PDF .....	16
1.7.1    iText .....	16
1.7.2    LaTeX .....	16

1.7.3	JasperReports .....	17
1.8	Conclusiones .....	17
Capítulo 2	.....	20
2.1	Requisitos .....	20
2.1.1	Requerimientos funcionales .....	20
2.1.2	Requerimientos no funcionales .....	21
2.2	Descripción del funcionamiento del sistema .....	23
2.3	Descripción del micro lenguaje .....	24
2.4	Modelo de implementación .....	30
2.4.1	Diagrama de despliegue .....	31
2.4.2	Diagrama de componentes .....	32
2.5	Conclusiones .....	36
Capítulo 3	.....	37
3.1	Modelo de prueba .....	37
3.1.1	Método de prueba .....	37
3.1.2	Descripción de los casos de pruebas .....	38
3.1.3	Resultados .....	51
3.2	Conclusiones .....	55
Conclusiones Generales	.....	56
Recomendaciones	.....	57
Referencias Bibliográficas	.....	58
Glosario de Términos	.....	60
Anexos	.....	64
Anexo 1: Diseño de caso de prueba CU Consultar ranking ordenado	.....	64

## Introducción

En la actualidad, la carrera de Ingeniería Informática incluye como parte central del plan de estudio la enseñanza de la programación. Es a través de esta materia que los estudiantes desarrollan las habilidades que serán necesarias en su vida laboral. Sin embargo, a partir de la experiencia de La Universidad de las Ciencias Informáticas es palpable en los alumnos que ingresan a la carrera la presencia de dificultades en el dominio de este contenido. Generalmente estas carencias están dadas por la falta de habilidades algorítmicas y de abstracción y el desconocimiento de los paradigmas que rigen la programación.

Para enfrentar este problema han surgido como soluciones los concursos de programación. Gestados en la década del 70 por William B. Poucher quien se desempeña como director general del ACM International Collegiate Programming Contest (ACM-ICPC), evento organizado anualmente por la Association for Computing Machinery (ACM). Estas competencias han favorecido el desarrollo de habilidades en las técnicas de análisis y diseño de algoritmos, la cooperación en un ambiente de equipo, la productividad y el pensamiento algorítmico, contribuyendo a la formación de mejores profesionales.

Para apoyar la preparación de estos concursos y la enseñanza de la programación en sentido general, surgen los jurados online. Estos básicamente son aplicaciones web disponibles todo el tiempo que automatizan la evaluación de las respuestas a los problemas de programación de competencia.

Es necesario tener en cuenta que en la generalidad de los entornos universitarios cubanos no se pueden aprovechar bien estos sistemas debido a que la disponibilidad de la red está sujeta a intermitencias o su velocidad es muy baja. Se afecta así uno de los pilares fundamentales de este tipo de software: su disposición constante a los estudiantes. En ausencia de estas condiciones, los educandos solo pueden adquirir conocimientos a través de los programas docentes y el autoestudio, que pueden no ser suficientes.

Aun cuando la red universitaria donde esté presente el jurado online no sufra de las afectaciones anteriormente mencionadas, otros factores pueden interferir en la disponibilidad de la interacción del usuario con el sistema. Los mismos pueden ser organizacionales (horarios de clases, pases, o demás afectaciones del cronograma de actividades del estudiante); tecnológicos (disponibilidad de un puesto de

trabajo o de herramientas de programación adecuadas a la tecnología que se desea utilizar); o simplemente humanos (para prácticas en equipo, dificultad de coordinar la presencia de todos los integrantes del mismo en un solo lugar, o para prácticas individuales, la disposición particular de una persona en resolver un problema determinado en un momento dado).

Conciliar la condición de conexión constante necesaria para el uso del jurado online con las posibilidades disminuidas es el objetivo de una serie de medidas que tienen como objeto lograr la disponibilidad de la información mínima para un uso eficiente del sistema. Dichas medidas son: la disponibilidad de la información del jurado en documentos offline en formato PDF y la accesibilidad del mismo por el protocolo SMTP (Simple Mail Transfer Protocol) en unión con otro protocolo que posibilite la descarga de los mensajes desde el servidor de correos.

El formato PDF es el estándar de visualización de documentos que permite independientemente del sistema operativo o la aplicación en la que se consulte el documento su visualización e impresión uniformes, no siendo así con otros formatos.

SMTP, al ser más desconectado que el protocolo HTTP (Hypertext Transfer Protocol), permite mantener una conexión más estable aun con dificultades de conectividad. Además es el protocolo soportado por los servidores de correo de las universidades del país y está ampliamente difundido, existiendo implementaciones de alta calidad que han demostrado ser muy robustas. Se hace necesario seleccionar además otro protocolo de correo, pues SMTP permite el envío de los mensajes, pero no la descarga de los mismos desde el servidor de correo.

En los últimos años se ha incrementado el acceso a los jurados online desde otras universidades cubanas fruto del despliegue y uso del Caribbean Online Judge (<http://coj.uci.cu>). Sin embargo, se puede decir que continúa siendo muy limitado el aprovechamiento de estos sistemas ya que en estas universidades el acceso a internet es restringido en cuanto a su velocidad y disponibilidad constante.

A partir de los resultados obtenidos en la Universidad de las Ciencias Informáticas (UCI) en cuanto a la aplicación de jurados online en competencias y a la auto preparación de los estudiantes se puede decir que contar con uno de estos sistemas debe ser un requisito de cualquier facultad de Ciencias Informáticas o ingenierías relacionadas.

Atendiendo a la problemática anteriormente explicada y la necesidad de desarrollar un jurado online con intercambio de información sobre protocolos de correo electrónico se define como **problema a resolver**: ¿Cómo lograr que estudiantes de universidades con redes de poco ancho de banda o limitados en su acceso a un sistema de evaluación automática, puedan interactuar con el mismo obteniendo resultados semejantes que con un jurado online estándar?

El problema a resolver identificado anteriormente se enmarca en el **objeto de estudio**: Proceso de desarrollo de aplicaciones de evaluación automática y sus funcionalidades, protocolos de transferencia de correo y generación de documentos.

El **objetivo general** de la investigación es: Implementar y probar un sistema de evaluación automática con intercambio de información sobre protocolos de correo electrónico que provea el conjunto mínimo de funcionalidades de un jurado online estándar y genere información en formato PDF.

Se tiene como **campo de acción**: Desarrollo de un jurado online accesible desde correo electrónico y generación de documentos en formato PDF.

Para dar solución al problema se plantean los siguientes **objetivos específicos**:

- ❖ Estudiar el estado del arte de los jurados online actuales haciendo énfasis en sus funcionalidades básicas y herramientas a utilizar en su desarrollo y seleccionar las herramientas más adecuadas.
- ❖ Estudiar el estado del arte de las tecnologías para la generación de documentos en formato PDF y seleccionar la más adecuada.
- ❖ Implementar y probar un sistema de evaluación automática con intercambio de información sobre correo electrónico, que provea el conjunto mínimo de funcionalidades de un jurado online estándar y genere documentos a formato PDF con la información necesaria para el uso offline de los usuarios.

Se sustenta la **Idea a defender** siguiente: Con el desarrollo de un jurado online accesible mediante correo electrónico que provea información en formato PDF se permitirá mejorar la disponibilidad de este tipo de software en universidades con redes de poco ancho de banda o limitadas en su acceso continuo a internet.

En función de dar cumplimiento a los objetivos trazados, se decide desarrollar las siguientes **tareas de investigación**:

- ❖ Elaboración del diseño teórico de la investigación.
- ❖ Revisión del estado del arte de los jurados online actuales y de las herramientas a utilizar en su desarrollo.
- ❖ Revisión del estado del arte de las tecnologías para la generación de documentos en formato PDF.
- ❖ Elaboración de la propuesta de solución.
- ❖ Evaluación de la factibilidad de la propuesta de solución en un entorno lo más real posible.

Para lograr un mejor aprovechamiento de este documento, el mismo está organizado en 3 capítulos, además cuenta con conclusiones, recomendaciones, bibliografía, glosario y anexos. El primero capítulo es Fundamentación teórica, incluye la revisión del estado del arte comenzando por las funcionalidades de los principales jurados online actuales y la investigación de las posibles tecnologías, metodologías y herramientas a utilizar en el desarrollo de la solución, así como la argumentación de la selección hecha. El capítulo 2, Implementación, documenta el proceso de implementación de la solución con el modelo de implementación y da una breve explicación del funcionamiento del sistema y como trabajar con él. El tercer capítulo, Pruebas, contiene el modelo de pruebas, en el que se aclara cuál es el método y técnica utilizados y se recogen los resultados de la ejecución de las pruebas.

# Capítulo 1

En este capítulo se realiza un estudio de las soluciones semejantes del sistema a desarrollar, haciendo énfasis en sus funcionalidades básicas y determinando las que serán aplicables a la solución. Se abordan diferentes tecnologías que existen para generar documentos PDF y se elige la adecuada para la aplicación. También se hace un análisis de las metodologías, herramientas, otras tecnologías y una breve justificación de las seleccionadas.

### 1.1 Jurados online de programación

Los jurados en línea son aplicaciones web disponibles todo el tiempo, que automatizan la evaluación de las respuestas a los problemas de programación de competencia y poseen un público objetivo muy especializado. Estos sistemas se asocian normalmente con entornos académicos (en especial universidades), debido a que en estos la búsqueda de conocimiento teórico se estimula de forma más activa. Los jurados presentan estas características porque sus objetivos y funcionamiento no son fáciles de convertir en lucrativos, excepto por las actividades colaterales de capacitación, publicaciones y eventos especiales por invitación. (1)

Estos sistemas constan generalmente de dos aplicaciones que funcionan de manera conjunta, una aplicación web para interactuar con los concursantes y otra encargada de la calificación de las soluciones enviadas por los clientes. Tal es el caso de los jurados Xtreme y CPAV, actualmente en uso en la comunidad universitaria de la UCI.

Aunque de estos sistemas existe una amplia documentación no hay un estándar definido para su desarrollo, que básicamente está regido por las necesidades de la organización donde estará presente y/o de las habilidades o visión del equipo de desarrollo que lo implemente. De modo general poseen un alto nivel de disponibilidad, permitiendo el libre acceso al sistema en todo momento a los usuarios.

Entre los principales jurados online de programación a nivel mundial que forman parte del estado del arte de esta investigación, se destacan:

- ❖ el Jurado Online de la Universidad de Valladolid (UVa) (<http://acm.UVa.es>)



- ❖ el Jurado Online de la Universidad de Gdansk en Polonia (SPOJ) (<http://www.spoj.pl>)
- ❖ el Jurado Online de la Universidad de Pekín(PKU) (<http://poj.org>)

En la UCI hace aproximadamente cuatro años se comenzaron a desplegar jurados online de programación. Actualmente los más importantes son:

- ❖ el Jurado Online Xtreme (<http://onlinejudgef8.uci.cu:5800/JudgeOnline>; <http://coj.uci.cu>)
- ❖ el Jurado Online de la Cátedra de Programación Avanzada (CPAV) (<http://cpav.uci.cu>)

### 3.1.1 UVa

El jurado online de la Universidad de Valladolid (UVa) es el más antiguo y reconocido sistema de evaluación automática de problemas de programación. Es fuente de problemas de los demás jurados, al contar con un amplio banco de estos organizados en volúmenes que se nutren de las competencias de la ACM. Posee entre sus servicios: una interfaz amigable, un detallado conjunto de estadísticas, rankings y un mecanismo sencillo para recibir sumisiones. (2)

El responsable de su surgimiento fue un estudiante de informática llamado Ciriaco García de Celis. Aunque este sistema fue desarrollado para el entrenamiento de los competidores de los concursos de programación internacional, se ha convertido en una herramienta utilizada para el autoestudio a nivel mundial. (2)

El grupo que se encarga de administrar este jurado forma parte de la Universidad de Valladolid y sus miembros gozan de gran prestigio a nivel mundial (3). Un ejemplo lo constituye el autor Miguel Revilla que es reconocido como experto en el tema de la programación de competencia a partir de sus libros publicados. Los integrantes de este equipo han estado involucrados en las competencias de la ACM a escala internacional.

En general, se puede afirmar que el UVa es el jurado más grande y prestigioso de todos los que serán objeto de análisis en esta investigación. Contiene los volúmenes más numerosos de problemas presentes en Internet y para la Universidad de las Ciencias Informáticas (UCI) constituye la principal referencia.

### 3.1.2 SPOJ

El Sphere Online Judge (SPOJ) es un proyecto del Sphere Interest Project, entidad afiliada a la Universidad de Tecnología de Gdansk en Polonia. Es una solución técnica muy sofisticada. Su sistema de

puntuación es el más complicado de todos los jurados que serán objeto de análisis, al igual que el sistema de distribución de los problemas, de los cuales reconoce varios tipos distintos. Estos aunque no muy numerosos son mantenidos en varios volúmenes y han sido replicados en ruso, portugués y vietnamita. En los volúmenes se encuentran todos los problemas presentados en las diversas ediciones de la ICPC, así como adiciones de los mismos usuarios y administradores del sitio. (1)

El SPOJ cuenta con varios rankings, en los cuales es determinante la puntuación de cada usuario, siendo agrupados por países, por categorías (problemsetter, challenger, normal). Dichos rankings, a diferencia de los de otros sistemas, no cambian solamente cuando el usuario en cuestión realiza una operación, sino cada vez que se realiza una sobre uno de los problemas que tributan a la puntuación del usuario. Esto da como resultado un sistema muy dinámico de puntuación y asegura que los problemas más complejos valgan más y que los usuarios que pasen cierto tiempo sin realizar envíos bajen en el ranking a favor de los usuarios más activos. (1)

Por otro lado es importante destacar que este jurado cuenta con un módulo especializado de competencia, muy versátil y eficiente, que ha sido anfitrión de muchas competencias (oficiales y de los propios usuarios). (4)

Soporta más de 30 lenguajes, lo que representa la mayor cantidad de estos asumidos por un jurado en Internet. Sin embargo, es importante destacar que la mayoría de estos lenguajes son muy especializados, anticuados o simplemente con una base de usuarios muy pequeña. (1)

En resumen, el SPOJ es una solución tecnológicamente elegante, muy flexible y que soporta todas las funcionalidades básicas de un jurado online maduro. Se destaca por tener una adecuada variedad de problemas y una base estable de usuarios y competencias.

### 3.1.3 PKU

El PKU Judge Online (PKU) además de las funcionalidades estándares que presentan el resto de estos sistemas, ofrece correo interno. Su módulo de estadísticas de sumisión es muy interesante, ya que permite apreciar el devenir del jurado desde que fue creado en 2003. El funcionamiento de este es muy suave, intuitivo y rápido. (1)

Es un sitio muy utilizado, según estadísticas publicadas en él, se realizaron 1 768 727 sumisiones en el año 2010, lo cual junto al análisis de los envíos diarios lleva a suponer que este sistema es el más visitado de todos los analizados. Tiene la interfaz más intuitiva de todos y se basa en una comunidad activa de usuarios. (5)

La mayor diferencia entre este y los demás jurados online es la opción de descargar una versión libre de la aplicación. Esta es mucho más pequeña y restrictiva que el original, además de componerse de ficheros compilados que no proporcionan código fuente. La opción de descarga está diseñada para poner en funcionamiento rápidamente una versión del jurado en cualquier entorno. (1)

En resumen el PKU es el más visitado de todos los jurados analizados, y el que tiene la interfaz más intuitiva. Alberga muchos servicios adicionales y como punto importante a destacar es que se basa en una comunidad activa de usuarios.

### 3.1.4 Xtreme

El jurado Xtreme está basado funcionalmente en la versión libre del jurado de Pekín y es muy sencillo. Su desarrollo se hizo en Java y tiene como principios fundamentales: simplicidad en las herramientas utilizadas en su construcción, velocidad de respuesta y estabilidad del sistema. Fue desarrollado en la UCI y ha permanecido en activo durante 4 años, cosechando numerosos logros en este período de tiempo.

Consta de las funcionalidades básicas de cualquier jurado online estándar y otras que se le han ido adicionando, entre las que se destacan: agrupación de ranking, sistema de puntuación dinámico (tanto para volumen público como para competencias), funcionalidad de hora muerta y otras mejoras menores. (6)

Al ser un sistema en desarrollo y mantenimiento activo se nutre de las ideas de las demás aplicaciones de su tipo en el mundo, además de constituir la línea oficial para dichas herramientas en el marco de la asociación de ACM-ICPC en Cuba.

El sistema del jurado Xtreme consta actualmente de 3 instancias confirmadas: una para el uso interno de la UCI, (Jurado Xtreme de la facultad 4), una en Internet para el uso público internacional (Caribbean Online Judge) y una para el uso interno de la UO (Jurado Online de la Universidad de Oriente). Las 3

instancias corren diferentes versiones del código base y se espera que se unifiquen a medida que el desarrollo del sistema continúe avanzando.

### 3.1.5 CPAV

El jurado de la cátedra avanzada de programación es el más joven de los jurados de la UCI. Rompe con los esquemas de los jurados anteriores, en filosofía, herramientas y objetivos. Está construido usando un CMS de PHP (más exactamente, el PHP-Fusion) y para evaluar las soluciones no utiliza el mismo software, sino un daemon aparte construido en C++. (1)

Entre las competencias que ha organizado se destacan: GrundyPC, She Programmer y la Copa Troya. El CPAV ofrece varios tipos de ranking: por año, por facultades, por lenguajes y general. Soporta los lenguajes clásicos de competencia (Java, C, C++ y Pascal), C#, python3.1 y python2.5. Recientemente (enero/2011) hospedó la copa individual Time vs Memory, convocada por el colectivo de administración de la CPAV. (7)

Ofrece una serie de servicios que derivan de su calidad de CMS, como son: la galería, el shoutbox, los anuncios y los últimos usuarios conectados. En contraste, no brinda la funcionalidad básica de ver el estado actual de las sumisiones fácilmente, lo cual es una deficiencia grave. El sistema de clasificación de los problemas en volúmenes por temas puede ser contraproducente dado que ofrece información sobre los conocimientos necesarios para resolver el problema. El uso de un CMS puede resultar poco efectivo a la hora de aumentar las funcionalidades de la aplicación o de evolucionar su interfaz gráfica, dado que no se tiene control completo sobre la fuente, al ser código legado con sus propias funcionalidades y arquitectura. (1)

Después de analizar las soluciones existentes de jurados online se decidió que las funcionalidades a desarrollar en la solución serían aquellas que respondieran a las operaciones básicas para el uso de los jurados online y la presentación de contenido en formato PDF. Esta última funcionalidad juega un papel fundamental para el jurado sobre correo electrónico. Las funcionalidades seleccionadas como conjunto mínimo imprescindible se modelaron como requisitos funcionales del sistema.

### 1.2 Metodologías de desarrollo

Para asegurar el éxito del proceso de desarrollo y la producción de una documentación consistente durante la implementación de la solución, es altamente recomendable utilizar una metodología de

desarrollo. Las más utilizadas y conocidas a nivel mundial y académico son: XP, Scrum y RUP, las cuales se analizan someramente a continuación.

### 1.2.1 XP

La Programación Extrema (XP, eXtreme Programming) es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. (8)

Es la más destacada de los procesos ágiles de desarrollo de software para proyectos de corto plazo y pequeño equipo. Pone más énfasis en la adaptabilidad que en la previsibilidad, los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.

La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Los defensores de XP creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

“XP supone:

- ❖ Las personas son claves en los procesos de desarrollo.
  - ❖ Los programadores son profesionales, no necesitan supervisión.
  - ❖ Los procesos se aceptan y se acuerdan, no se imponen.
  - ❖ Desarrolladores y gerentes comparten el liderazgo del proyecto.
  - ❖ El trabajo de los desarrolladores con las personas que conocen el negocio es regular, no puntual.”
- (8)

### 1.2.2 Scrum

“Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Es una metodología ágil, y como tal:

- ❖ Es un modo de desarrollo de carácter adaptable más que predictivo.

- ❖ Orientado a las personas más que a los procesos.
- ❖ Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.” (9)

Establece un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores.

Durante cada sprint, un período entre 15 y 30 días, el equipo crea un incremento de software potencialmente entregable. Estas iteraciones son la base del desarrollo ágil, gestionándose su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado hasta la fecha y la previsión para el día siguiente.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada.

### 1.2.3 RUP

EL Rational Unified Process (RUP) es un proceso de desarrollo de software que constituye el estándar más usado para el análisis y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso (CU). Incluye artefactos (que son los productos tangibles del proceso, por ejemplo, el modelo de casos de uso) y roles. Está basado en componentes, consta de nueve flujos de trabajos fundamentales, de ellos seis denominados “de Ingeniería”, que son: Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue. Los restantes tres flujos de trabajo son considerados de apoyo: Administración de Proyectos, Gestión de Configuración y Cambios y Ambiente. (10)

Es más apropiada para proyectos grandes (aunque es configurable para proyectos pequeños).

#### Características

- ❖ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- ❖ Pretende implementar las mejores prácticas en Ingeniería de Software

- ❖ Desarrollo iterativo
- ❖ Administración de requisitos
- ❖ Uso de arquitectura basada en componentes
- ❖ Control de cambios
- ❖ Modelado visual del software
- ❖ Verificación de la calidad del software

Luego de realizar un estudio de las metodologías anteriores se elige RUP para el sistema porque la misma tiene bien definido y organizado el proceso de desarrollo de software, lo que permite que los involucrados estén claros de los pasos a seguir para obtener un producto con calidad. Provee un entorno de proceso de desarrollo configurable, atendiendo a las características específicas del sistema. Teniendo en cuenta que la aplicación a desarrollar es de nuevo tipo y se espera que tenga continuidad, es importante gestionar una documentación detallada para apoyar el futuro desarrollo. Una de las vías para alcanzar esto es a través de artefactos específicos de RUP. Estas características representan algunas de las razones que justifican la elección para guiar el desarrollo del jurado online sobre correo electrónico con dicha metodología.

### 1.3 Herramientas CASE

CASE corresponde a las iniciales de: Computer Aided Software Engineering; y en su traducción al Castellano significa Ingeniería de Software Asistida por Computadoras. Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores en el modelado visual, durante todos los pasos del ciclo de vida de desarrollo de un software. (11)

#### 1.3.1 Rational Rose

Es una herramienta para el modelado visual mediante UML (Unified Modeling Language) de sistemas de software. Proporciona un marco de desarrollo que permite cubrir todo el ciclo de vida de un proyecto, en el que se puede especificar, analizar y diseñar un sistema antes de codificarlo; ayudando a establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Además, permite generar código a partir de los modelos y realizar ingeniería inversa (crear modelo a partir de código). (12)

Aunque sus características la convierten en una herramienta muy potente y deseable, la particularidad de ser propietaria hace que sea rechazada por muchas empresas y desarrolladores de software.

### 1.3.2 Visual Paradigm for UML

Es una herramienta profesional de modelado que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite representar todos los tipos de diagramas de clases, hacer ingeniería inversa, generar código desde diagramas y generar documentación. Brinda soporte para varias versiones de UML. Es una herramienta multiplataforma que ayuda a una rápida construcción de aplicaciones de mejor calidad y a un menor costo. Posibilita la integración con los principales IDEs de desarrollo. (13)

A partir del análisis de las herramientas anteriores, se decidió que Visual Paradigm for UML será la seleccionada para llevar a cabo el modelado visual. Para esto se tiene en cuenta que sus características la convierten en una de las herramientas más potentes y utilizadas en la actualidad. El hecho de que en el entorno donde se está desarrollando la solución se posee su licencia comprada y que se integre con los principales IDEs de programación constituyen también razones de peso.

### 1.4 Herramientas de desarrollo

Debido a la naturaleza de la aplicación que se quiere lograr y su relación estrecha con el software ya existente (Jurado Xtreme) la mayoría de las herramientas a utilizar tienen su selección bien fundamentada en el trabajo de diploma: “Análisis, diseño e implementación de un Jurado Online”, de los autores Jorge A. Soria Ramírez y Enrique José Altuna Castillo. Se difiere de la decisión de utilizar algunas herramientas de dicho perfil y serán especificadas a continuación:

- ❖ El IDE utilizado por aplicaciones anteriores de la iniciativa ha sido Eclipse, en varias configuraciones y versiones, según el momento. Para este trabajo se selecciona en su lugar NetBeans en su versión 6.9, por considerarse que como IDE durante su evolución ha alcanzado al Eclipse. Atendiendo al conjunto de funcionalidades básicas provistas a los desarrolladores, es un IDE más ligero que el Eclipse (por ejemplo, en cuanto al uso de memoria) y se tienen mayores conocimientos técnicos sobre su uso y funcionamiento.
- ❖ Adicionalmente, la solución técnica prevista no requiere ningún plug-in o funcionalidad específica del IDE, por tanto sus características no afectan la decisión particular.



- ❖ En las versiones anteriores de la aplicación se usó Windows XP como sistema operativo. Sin embargo, producto de que en la universidad donde se va a desarrollar la solución se está llevando a cabo un proceso de migración de software propietario a software libre, se decide adecuar el desarrollo a esta política. Quedando seleccionados Linux, particularmente Ubuntu 10.10, como sistema operativo.
- ❖ De acuerdo a la decisión de migrar a software libre, se reemplaza al cliente de datos SQL Manager 2005 forMySQL por phpMyAdmin en su versión 3.3.7. Avalan esta decisión su condición de ser multiplataforma y que existe en internet basta información sobre su uso y funcionamiento. Es una herramienta muy completa que permite acceder a todas las funciones típicas de bases de datos MySQL a través de una interfaz web muy intuitiva.

El perfil tecnológico completo queda especificado en detalle en las conclusiones del presente capítulo.

### 1.5 Protocolos de correo

Producto de que SMTP posibilita el envío de correos y la recepción de ellos en el servidor pero no permite descargar un mensaje en una estación de trabajo, se hace necesario realizar un estudio de otros protocolos que brinden esta posibilidad. Para ello se analizarán POP3 e IMAP4 que son los más usados y difundidos.

#### 1.5.1 POP3

El Protocolo de Oficina de Correos (POP3, Post Office Protocol - Versión 3) permite que una estación de trabajo pueda acceder dinámicamente a un buzón en un servidor anfitrión de una manera eficiente. Por lo general esto significa que el protocolo POP3 se utiliza para permitir a un ordenador recuperar correo de un servidor con pocos recursos de conectividad. POP3 no está destinado a proporcionar amplias operaciones de manipulación de correo en el servidor, normalmente el correo se descarga y se elimina.  
(14)

A pesar de ser el mecanismo más usado por los usuarios de Internet para conseguir su correo, ya que les permite consultar el mismo sin estar conectados, existen funcionalidades con que los usuarios se familiarizan que no las soporta POP como la capacidad de ver cuántos mensajes de correo nuevos tienen. Estas capacidades se construyen en programas como Eudora o Microsoft Outlook, que recuerdan los últimos correos recibidos y calculan cuántos se tienen nuevos.

### 1.5.2 IMAP4

El Protocolo de Acceso a Mensajes de Internet, versión 4 (IMAP4, Internet Message Access Protocol) permite a un cliente acceder y manipular mensajes de correo electrónico en un servidor. Así como la manipulación de buzones de correo de una manera que es funcionalmente equivalente a carpetas locales de correo. IMAP4 también proporciona la capacidad de una línea cliente para volver a sincronizar con el servidor. (15)

Este protocolo incluye operaciones para crear, eliminar y cambiar el nombre de carpetas de buzones de correo, comprobar si hay mensajes nuevos, eliminar mensajes de forma permanente, manipular banderas y brinda criterios que pueden utilizarse para ordenar y obtener los correos electrónicos. (15)

IMAP es un protocolo más avanzado que POP3 para la obtención de mensajes desde servidores de correo electrónico. Debido a esto se podría pensar que sería utilizado por todos pero no es así, pues sobrecarga mucho el servidor de correo, requiriendo que este reciba los nuevos mensajes y los entregue a los usuarios cuando sean solicitados, manteniéndolos en el servidor. Aunque esto centraliza las copias de seguridad, también hace que las carpetas de correo a largo plazo de los usuarios se hagan más grandes. Para usar el IMAP, el servidor de correo debe soportar este protocolo.

Para descargar los mensajes que se encuentran en el servidor de correo se usará el protocolo IMAP4 debido a las ventajas que presenta sobre el protocolo POP3. No se corre el riesgo de que el buzón asignado se haga muy grande pues los correos serán eliminados inmediatamente que sean respondidos.

### 1.6 Tecnología para el envío y recepción de correos

El API JavaMail es un conjunto de funcionalidades y clases que modelan un sistema de correo. Proporciona un framework independiente de la plataforma o el protocolo para construir aplicaciones clientes de correo electrónico basadas en tecnología Java. Brinda los medios para leer y enviar correo electrónico. Los proveedores de servicios implementan protocolos en particular. Varios proveedores de servicios se incluyen con el paquete del API JavaMail, mientras que otros están disponibles por separado. El API JavaMail se implementa como un paquete opcional de Java que puede ser utilizado en el JDK 1.4 y versiones posteriores en cualquier sistema operativo. (16)

Se selecciona esta tecnología debido a su gratuidad y por las ventajas que brinda para el programador su fácil uso. JavaMail permite, al implementar desde cero las funcionalidades de correos, el dominio a fondo

del código fuente. Esto flexibiliza el proceso de agregar nuevas prestaciones si así se requiere y que solamente tenga las que necesita, haciendo más ligera la solución, en contraste con la situación que se crea al usar una herramienta previamente construida, la cual brindaría funcionalidades innecesarias y sería más compleja de extender. El hecho de que Java sea la tecnología escogida por otros aplicativos de la Iniciativa Xtreme, en cuyo marco se encuentra el resultado esperado de este trabajo, también es un punto a su favor.

### 1.7 Tecnologías para la generación de documentos en formato PDF

#### 1.7.1 iText

iText es un API de código abierto creada por Bruno Lowagie y Paulo Soares que permite a los desarrolladores producir archivos PDF sobre la marcha y/o manipular documentos existentes de ese formato. Puede ser fácilmente integrado en una aplicación web para servir a los navegadores de documentos PDF de una manera dinámica. La premisa de iText es que no se precisa ser un especialista en PDF para generar documentos. (17)

Hoy en día, iText se utiliza en muchos servicios en línea y productos de software, directa o indirectamente. Los informes creados con una de las herramientas de generación de reportes más importantes del momento, JasperReports o el Eclipse/BIRT-II usan iText pues este se utiliza como su motor PDF. Se puede encontrar gran cantidad de documentación y ejemplos en internet sobre su funcionamiento, lo que facilita su uso y comprensión. Una de sus principales ventajas es que puede convertir contenido HTML a formato PDF soportando la mayoría de las etiquetas. (17)

#### 1.7.2 LaTeX

Es un sistema de composición tipográfica de documentos formado por un conjunto de macros de TeX (sistema de tipografía escrito por Donald E. Knuth). Es muy adecuado para la producción de materiales científicos y matemáticos de alta calidad tipográfica por su capacidad de representar ecuaciones, fórmulas complicadas, notación científica y las facilidades aportadas para estructurar el documento (con capítulos, secciones, notas, bibliografía e índices analíticos). Por estas razones hoy en día es muy popular en dichas comunidades, también es ampliamente utilizado en la industria. Los científicos envían sus documentos por vía electrónica a sus colegas de todo el mundo en el formato de entrada de LaTeX. (18)

A diferencia de otros procesadores de texto, LaTeX permite concentrar la atención en el contenido y dejar a un lado las preocupaciones por el formato. Está basado en comandos, cuestión que es considerada una desventaja de dicha tecnología. Es software libre bajo licencia LPPL.

### 1.7.3 JasperReports

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 alta calidad que se pueden ver, imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, Excel, OpenOffice y Word. (19)

JasperReports genera reportes a partir de archivos XML que especifican exactamente el formato que deben llevar los mismos. La construcción de este archivo sin el auxilio de una herramienta visual no es práctica. Para hacer esto se usa iReport, herramienta para obtener archivos XML que proporciona un entorno WYSIWYG (what you see is what you get) para diseñar informes. (20)

Debido a que no permite añadir imágenes de una forma dinámica (solo permite hacerlo si se ha predefinido con anterioridad en el archivo XML), no se selecciona esta potente librería. Esta limitación es un inconveniente en el contexto del presente trabajo pues la base de datos a utilizar posee mucha información en forma de contenido HTML con imágenes embebidas. Queda de esta forma como mejor opción iText, ya que LaTeX, a pesar de ser una de las tecnologías más usadas en la actualidad, no es una opción viable debido a su complejidad de uso.

### 1.8 Conclusiones

En este capítulo se realizó un estudio de las soluciones similares de sistemas de evaluación automática haciendo énfasis en sus principales funcionalidades y características, así como de los antecedentes a la aplicación que se desarrollará: los jurados online, los protocolos de transferencia de correo y las tecnologías de generación de documentos en formato PDF. Para el posterior desarrollo de la solución se investigaron las herramientas disponibles, seleccionando las más adecuadas según diversos criterios.

Finalmente, la metodología de desarrollo a utilizar será el Proceso Unificado de Desarrollo (RUP), el lenguaje de programación, las herramientas y tecnologías quedan especificados en el perfil tecnológico de desarrollo (**Tabla 1**).

Clasificación	Herramienta	Versión
Lenguaje de programación		6.0
IDE		6.9
Gestor de datos		5.1.x
Cliente de datos		3.3.7
Control de versiones		1.6.12
Cliente subversion		0.12.0
Tecnología para generar documentos en formato PDF		5.0.6

Tecnología para el envío y recepción de correos		1.4.3
Herramienta CASE		6.4
SO utilizado		10.10

**Tabla 1 Perfil tecnológico de desarrollo**

# Capítulo 2

En este capítulo se hace una descripción del funcionamiento de la aplicación y se recogen los requisitos encontrados por el analista. Se describe el micro lenguaje establecido para la comunicación de los usuarios con el sistema. Se modelan los artefactos que se generan a partir del flujo de trabajo de Implementación, o sea, se describe cómo los elementos del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

### 2.1 Requisitos

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir y los no funcionales son propiedades o cualidades que el producto debe tener. Previo a este trabajo se desarrolló una investigación, donde se realizó el levantamiento de requisitos, así como el análisis y el diseño del sistema. Para la correcta comprensión de los distintos artefactos, se decide incluir la descripción de los requisitos en el presente capítulo.

#### 2.1.1 Requerimientos funcionales

**RF1:** Convertir a PDF información seleccionada.

**RF1.1:** Convertir a PDF las FAQs.

**RF1.2:** Convertir a PDF la ayuda.

**RF1.3:** Convertir a PDF un volumen de problemas.

**RF1.4:** Convertir a PDF un problema.

**RF2:** Ver ayuda.

**RF3:** Consultar el estado de un problema.

**RF4:** Ver noticias.

**RF5:** Someter solución de un problema.

**RF6:** Modificar la información de un usuario.

**RF7:** Consultar el estado del servidor.

**RF8:** Ver FAQs.

**RF9:** Consultar el ranking de una cantidad n de usuarios especificada.

**RF10:** Consultar el listado de problemas de un volumen determinado.

**RF11:** Ver un problema.

**RF12:** Ver solución de un problema determinado.

**RF13:** Enviar correo.

### 2.1.2 Requerimientos no funcionales

#### Hardware

**RNF1:** Funcionar en una estación de trabajo estándar: 512 Mb RAM, 5 GB, conexión a red local o Internet.

#### Software

**RNF2:** Tener acceso a un servidor de correo y a una cuenta en el mismo.

**RNF3:** Las estaciones de trabajo clientes deben tener acceso a un servidor de correo.

#### Usabilidad

**RNF4:** Debe proporcionar ayuda sobre cada funcionalidad disponible.

**RNF5:** El servidor de correo de las estaciones de trabajo clientes debe poseer acceso al servidor de correo del sistema.

#### Seguridad

**RNF6:** Las soluciones de los problemas solo pueden ser vistas por el propietario de las mismas.

**RNF7:** Solo se permitirá someter soluciones a los usuarios registrados.



**RNF8:** Únicamente los usuarios registrados podrán modificar su información personal.

### **Apariencia**

**RNF9:** Debe presentar la información al usuario de manera clara y comprensible.

### **Soporte**

**RNF10:** El correo y otros datos de contacto de los desarrolladores del sistema deben quedar claros para los usuarios.

### **Portabilidad**

**RNF11:** Podrá ser utilizado bajo cualquier sistema operativo.

**RNF12:** Los sistemas clientes podrán interactuar con el contenido producido utilizando cualquier cliente de correo estándar.

### **Rendimiento**

**RNF13:** Los adjuntos de los correos enviados no deben exceder el límite de tamaño establecido por el servidor de correo del sistema.

### **Restricciones de diseño e implementación**

**RNF14:** NetBeans 6.9 como IDE de programación.

**RNF15:** Java 6.0 como lenguaje de programación.

**RNF16:** MySQL 5.1.x como sistema gestor de base de datos.

**RNF17:** VisualParadigm 6.4 como herramienta CASE para el modelado UML.

**RNF18:** JavaMail 1.4.3 como API para implementar las funcionalidades del correo.

**RNF19:** iText 5.0.6 como API para la implementación de las funcionalidades de la generación de documentos.

### Estándares

**RNF20:** La comunicación entre los usuarios y el sistema debe estar regida por el micro lenguaje establecido en el **epígrafe 2.3**.

### 2.2 Descripción del funcionamiento del sistema

La aplicación es un daemon, no posee interfaz visual, se ejecuta continuamente en el sistema de la máquina en que se despliegue. Su funcionamiento se basa en descargar de manera constante los correos que se encuentren en la bandeja de entrada del buzón asignado. Una vez realizada la descarga, el sistema da respuesta a la petición y elimina el correo del servidor. En caso de que no haya correos para responder, la aplicación espera un tiempo definido por los administradores antes de ejecutar una nueva descarga, de esta forma no se sobrecarga el buzón de forma innecesaria.

El nombre escogido para la aplicación es MailJudge, vocablo que está compuesto por dos palabras del inglés: mail o correo en español y judge o jurado. La comunicación de los usuarios con el MailJudge es a través del micro lenguaje definido para este fin. Para que un usuario pueda interactuar de forma satisfactoria, debe enviar un correo a la dirección establecida para la comunicación. En el asunto debe especificar lo que desea ver, respetando el formato que define el micro lenguaje. Ejemplo: para ver el problema 1362, en el asunto del correo que debe enviar a la dirección de contacto, debe escribir **“mj:problema -i 1362” (Figura 1)**.

Debido a que el buzón se utiliza para otra labores se hace necesario que todos los correos que se envíen a la aplicación contengan en el asunto la cadena **“mj:”**. Para facilitar la comunicación con los usuarios, todo correo que sea enviado con un formato incorrecto, es respondido por el sistema con una explicación detallada de cómo se debe utilizar el micro lenguaje.

El proceso de autenticación en el sistema se realiza de la siguiente forma: si existe un usuario en la base de datos que posea la dirección de correo de la cual se está enviando la petición al sistema, entonces ese usuario está autenticado. Dentro de las funcionalidades que se implementan en la aplicación no se incluyen las de administración ni se permite registrar usuarios. Por esta razón, para que un usuario que no está registrado en la base de datos se autentique, debe entrar al jurado Xtreme y hacerlo o escribir a la dirección de contacto que se envía en cada correo de respuesta del sistema.

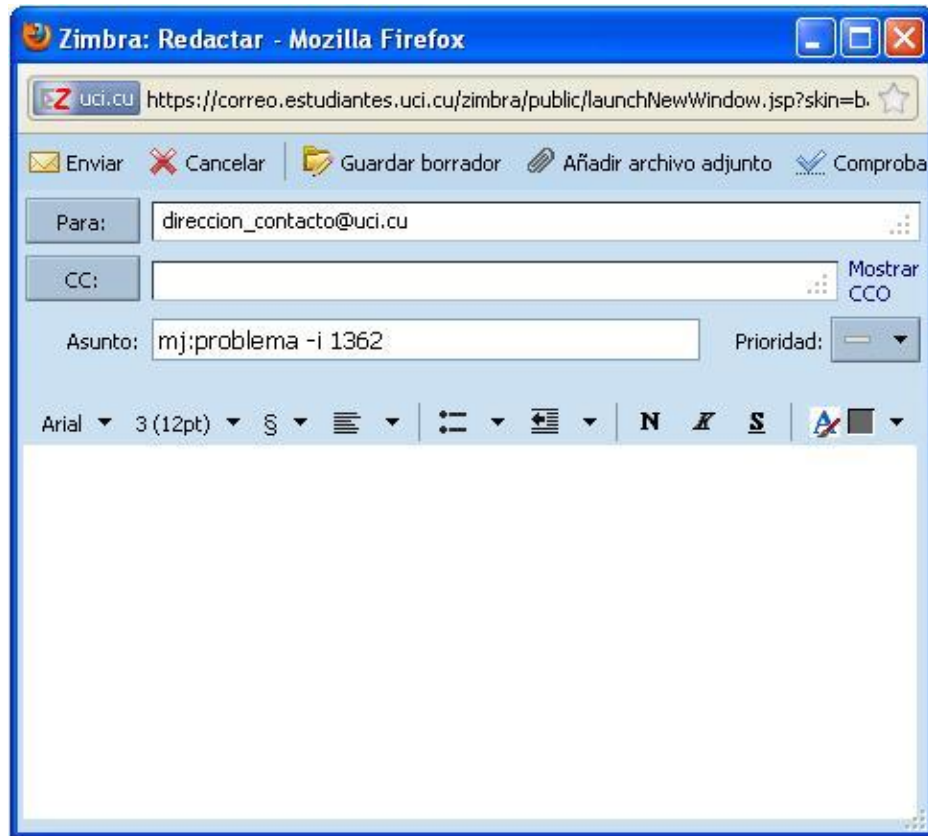


Figura 1 Ejemplo de cómo ver un problema

### 2.3 Descripción del micro lenguaje

#### Comando para consultar el estado del servidor

`mj:estado <variables>`

Las variables están formadas por los pares **<critério>** **<valor>**:

**<critério>**: son 6 los criterios del estado del servidor: tipo de estado, usuario, tipo de respuesta, rango, número de problema y lenguaje. Se representan a la hora de especificar en el asunto del correo electrónico para ser enviados al sistema con los respectivos caracteres (**-t**, **-u**, **-r**, **-n**, **-i** y **-l**).

**<valor>**: significa el conjunto de los valores que pueden tomar cada uno de los criterios anteriormente especificados. En el caso del critério **-t** (tipo de estado) el valor sería **servidor**; para **-u** (usuario) el valor

sería un usuario válido; para **-r** (tipo de respuesta) los valores serían **ac** (aceptado), **wa** (respuesta incorrecta), **rte** (error en tiempo de ejecución), **tle** (tiempo límite excedido), **mle** (memoria límite excedida), **pe** (error de presentación) o **ce** (error de compilación); para **-n** (rango) el valor sería un rango de números con la forma **#-#**, el rango que se especifique no debe exceder los 500 y el menor valor a poner en el inicio del rango es 1; para **-i** (número de problema) el valor sería un número de problema válido y para **-l** (lenguaje) los valores serían un lenguaje de programación válido.

- ❖ Para el caso del criterio tipo de estado, cuando el usuario no lo especifica, el sistema por defecto devuelve el estado del servidor.
- ❖ Para el caso del criterio usuario, cuando no se especifica, el sistema por defecto devuelve el estado de las sumisiones de todos los usuarios.
- ❖ Para el caso del criterio tipo de respuesta, cuando no se especifica, el sistema por defecto devuelve el estado de las sumisiones de todos los tipos de respuesta.
- ❖ Para el caso del criterio rango, cuando no se especifica, el sistema por defecto devuelve el estado de las últimas 500 sumisiones.
- ❖ Para el caso del criterio número de problema, cuando no se especifica, el sistema por defecto devuelve el estado de las sumisiones de todos los problemas.
- ❖ Para el caso del criterio lenguaje, cuando no se especifica, el sistema por defecto devuelve el estado de las sumisiones en cualquier lenguaje.

Si el usuario desea especificar más de una variable puede hacerlo separándolas por espacios. Por ejemplo: **mj:estado -t servidor -u arosa**.

### Comando para consultar el estado de un problema

mj:estado <variables>

Las variables están formadas por el par <criterio> <valor>:

<criterio>: son 3 los criterios del estado de un problema: tipo de estado, criterio de ordenamiento y número del problema. Se representan a la hora de especificar en el asunto del correo electrónico para ser enviados al sistema con los respectivos caracteres (**-t**, **-c** y **-i**).

**<valor>**: significa el conjunto de los valores que pueden tomar cada uno de los criterios anteriormente especificados. En el caso del criterio **-t** (tipo de estado) el valor sería **problema**; para **-c** (criterio de ordenamiento) los valores serían **memoria** (memoria consumida), **tiempo** (tiempo consumido) y **codigo** (longitud del código) y para **-i** (número de problema) el valor sería un número de problema válido.

- ❖ El criterio tipo de estado tiene que especificarse.
- ❖ Para el caso del criterio de ordenamiento, cuando no se especifica, el sistema por defecto devuelve el estado ordenado por tiempo consumido.
- ❖ El criterio número de problema tiene que especificarse.

Si el usuario desea especificar más de una variable puede hacerlo separándolas por espacios. Por ejemplo: **mj:estado -t problema -c memoria**.

### Comando para consultar listado de problemas

mj:volumen **<variables>**

Las variables están formadas por los pares **<criterio>** **<valor>**:

**<criterio>**: son 3 los criterios del listado de problemas: número de volumen, criterio de ordenamiento y tipo de formato. Se representan a la hora de especificar en el asunto del correo electrónico para ser enviados al sistema con los respectivos caracteres (**-v**, **-c** y **-f**).

**<valor>**: significa el conjunto de los valores que pueden tomar cada uno de los criterios anteriormente especificados. En el caso del criterio **-v** (# de volumen) el valor sería el número del volumen de problemas deseado; para **-c** (criterio de ordenamiento) los valores serían **ptos** (puntos del problema), **ac** (cantidad de soluciones aceptadas al problema), **envios** (cantidad de envíos hechos al problema) y **%ac** (% de soluciones aceptadas al problema) y para **-f** (tipo de formato) el valor sería **pdf**.

- ❖ Para el caso del criterio # de volumen, cuando no se especifica, el sistema por defecto devuelve el listado de problemas del volumen 1.
- ❖ Para el caso del criterio de ordenamiento, cuando no se especifica, el sistema por defecto devuelve el estado ordenado por el identificador del problema.

- ❖ Cuando se especifica el criterio tipo de formato no puede especificarse el criterio de ordenamiento y viceversa.
- ❖ Para el caso del criterio tipo de formato, cuando no se especifica, el sistema por defecto devuelve el listado de problemas en el cuerpo del correo. Cuando se especifica, entonces el sistema devuelve el listado con las descripciones de todos los problemas de ese volumen en un documento en formato PDF.

Si el usuario desea especificar más de una variable puede hacerlo separándolas por espacios. Por ejemplo: **mj:volumen -f pdf -v 2**.

### Comando para consultar el ranking ordenado

mj:ranking <variables>

Las variables están formadas por los pares <critério> <valor>:

<critério>: son 2 los criterios del ranking: critério de ordenamiento y rango. Se representan a la hora de especificar en el asunto del correo electrónico para ser enviados al sistema con los respectivos caracteres (-c y -n).

<valor>: significa el conjunto de los valores que pueden tomar cada uno de los criterios anteriormente especificados. En el caso del critério -c (critério de ordenamiento) los valores serían **ptos** (puntos acumulados), **ac** (cantidad de soluciones aceptadas), **envios** (cantidad de envíos) y **%ac** (% de soluciones aceptadas); para -n (rango) el valor tiene la forma #-#. Ejemplo 1-500. El rango que se especifique no debe exceder los 500 y el menor valor a poner en el inicio del rango es 1.

- ❖ Para el caso del critério rango, cuando no se especifica, el sistema por defecto devuelve el ranking de los 500 primeros usuarios ordenados según el critério especificado.
- ❖ Para el caso del critério de ordenamiento, cuando no se especifica, el sistema por defecto devuelve el ranking ordenado por puntos.

Si el usuario desea especificar más de una variable puede hacerlo separándolas por espacios. Por ejemplo: **mj:ranking -c ptos -n 45-60**.

### Comando para modificar información de usuario

mj:modificar <variables>

Las variables están formadas por los pares <critério> <valor>:

<critério>: son 3 los criterios de la información del usuario: nick, correo y país. Se representan a la hora de especificar el comando en el asunto del correo electrónico para ser enviados al sistema con los respectivos caracteres (-n, -e y -p).

<valor>: significa el conjunto de los valores que pueden tomar cada uno de los criterios anteriormente especificados. En el caso del criterio -n (nick) el valor sería el nuevo nick, para -e (correo) el valor sería el nuevo correo y para -p (país) el valor sería el nuevo país.

- ❖ El usuario debe especificar al menos un criterio a modificar.

Debe ser un usuario registrado para realizar esta operación. Si el usuario desea especificar más de una variable puede hacerlo separándolas por espacios. Por ejemplo: **mj:modificar -n Nersa Doraines -p Cuba.**

### Comando para someter solución de un problema

Para el asunto del correo: mj:someter <variables>

Las variables están formadas por los pares <critério> <valor>:

<critério>: son 2 los criterios de someter solución: número de problema y lenguaje. Se representan a la hora de especificar el comando en el asunto del correo electrónico para ser enviados al sistema con los respectivos caracteres (-i y -l).

<valor>: significa el conjunto de los valores que pueden tomar cada uno de los criterios anteriormente especificados. En el caso del criterio -i (# de problema) el valor sería un número de problema válido y para -l (lenguaje) el valor sería un lenguaje de programación válido.

- ❖ El criterio número de problema tiene que especificarse.
- ❖ El criterio lenguaje tiene que especificarse.

Las variables se separan con espacios. Por ejemplo: **mj:someter -i 1000 -l g++**.

En el cuerpo del correo se especifica: **<separador inicio> <código> <separador final>**

**<separador inicio>**: es la siguiente cadena: **<--PRINCIPIO DE CODIGO**.

**<código>**: código de la solución.

**<separador final>**: es la siguiente cadena: **/FIN CODIGO-->**.

Debe ser un usuario registrado para realizar esta operación.

### Comando para ver un problema

mj:problema **<variables>**

Las variables están formadas por los pares **<critério> <valor>**:

**<critério>**: son 2 los criterios de ver problema: número de problema y tipo de formato. Se representan a la hora de especificar el comando en el asunto del correo electrónico para ser enviados al sistema con los respectivos caracteres (**-i** y **-f**).

**<valor>**: significa el conjunto de los valores que pueden tomar cada uno de los criterios anteriormente especificados. En el caso del criterio **-i** (# de problema) el valor sería un número de problema válido y para **-f** (tipo de formato) el valor sería **pdf**.

- ❖ El criterio número de problema tiene que especificarse.
- ❖ Para el caso del criterio tipo de formato, cuando no se especifica, el sistema devuelve la descripción del problema en el cuerpo del correo. Cuando se especifica, entonces el sistema devuelve la descripción del problema en un documento en formato PDF.

Si el usuario desea especificar más de una variable puede hacerlo separándolas por espacios. Por ejemplo: **mj:problema -f pdf -i 1002**.

### Comando para ver una solución

mj:solucion -s **<id solución>**



**<id solución>**: es el identificador de la solución. Un número que se le asigna a cada solución atendiendo al orden en que son enviadas al jurado. Es obligado para el usuario especificar el id de la solución que desea ver y para que el sistema la devuelva debe ser propietario de la misma.

### Comando para ver las noticias

mj:noticias

### Comando para ver las preguntas frecuentes

mj:faqs **<variable>**

La variable está formada por el par **<critério> <valor>**:

**<critério>**: el único critério para ver las faqs es si desea verlo en formato pdf. Puede especificarse o no. Se representa a la hora de especificar el comando en el asunto del correo electrónico para ser enviado al sistema con el caracter (-f).

**<valor>**: el valor sería pdf.

### Comando para ver la ayuda

Solo se necesita especificar:

mj:

De querer visualizar la ayuda en formato PDF se debe poner:

mj:ayuda -f pdf

## 2.4 Modelo de implementación

El flujo de trabajo de Implementación va dirigido a implementar la arquitectura que se definió con anterioridad. Está fuertemente determinado por el lenguaje de programación. Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación al describir los componentes, su organización y la dependencia entre los nodos físicos en los que funcionará la aplicación.

### 2.4.1 Diagrama de despliegue

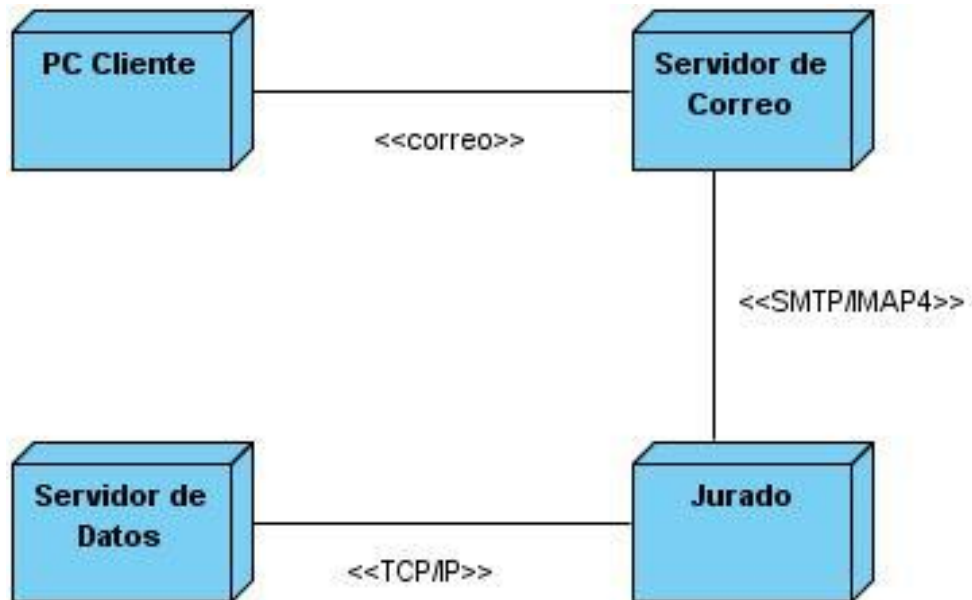


Figura 2 Modelo de despliegue

Los sistemas clientes (**PC Cliente**) se comunican mediante un protocolo cualquiera de correo electrónico (representado por el estereotipo <<correo>>) con un servidor de correo capaz de utilizar SMTP e IMAP4 (**Servidor de Correo**).

El Jurado (**Jurado**) se comunica a su vez con un servidor de correo (que puede o no ser el mismo servidor de correo que utiliza el cliente, siempre y cuando haya comunicación entre ambos) mediante una combinación de protocolos SMTP e IMAP4. Es importante observar que el servidor de correo, aunque no restringe el protocolo para la comunicación con los sistemas clientes, si lo hace para que los protocolos necesarios en la comunicación con el Jurado sean SMTP e IMAP4. Esta es una restricción importante para servidores Exchange de Microsoft que son capaces de funcionar sin SMTP, en cuyo caso la interacción con el Jurado no es posible.

El Jurado realiza los procesos de negocio definidos para el sistema e interactúa a través del protocolo TCP/IP con un servidor de datos (**Servidor de Datos**) compartido con una instancia del Jurado Online Xtreme. En este servidor se almacenan datos correspondientes a la acción de ambos jurados. Nótese que

los jurados Xtreme y el que se está desarrollando pueden o no existir en el mismo servidor de aplicaciones siempre y cuando el hardware de este nodo satisfaga los requisitos no funcionales de ambas aplicaciones.

### 2.4.2 Diagrama de componentes

Un diagrama de componentes modela los aspectos físicos de un sistema tales como: ejecutables, tablas, librerías, archivos, documentos y las relaciones existentes entre ellos. Constituye la vista de implementación estática de un sistema.

Las imágenes que aparecen a continuación representan los 4 diagramas de componentes de la aplicación. El primero contiene una vista general de cómo interactúan los paquetes y componentes del sistema. Los tres diagramas restantes muestran la estructura de los paquetes vistas, comandos y beans.

El Diagrama de componentes general (**Figura 3**) está compuesto por seis paquetes y el componente que representa a la base de datos. A continuación se explicarán tres de estos paquetes, los otros tres son explicados con sus respectivos diagramas.

El primer paquete (**core**) contiene el componente que se encarga de dirigir todo el sistema, controlando la interacción entre los distintos manejadores de la aplicación. El segundo paquete (**manejadores**) contiene los tres manejadores del sistema: el que dirige la comunicación con el buzón de correo, el encargado de la lógica del negocio y el que controla la presentación de la información. Por último el paquete **data** contiene los componentes encargados del acceso a datos por lo que es este paquete el que utiliza la base de datos.

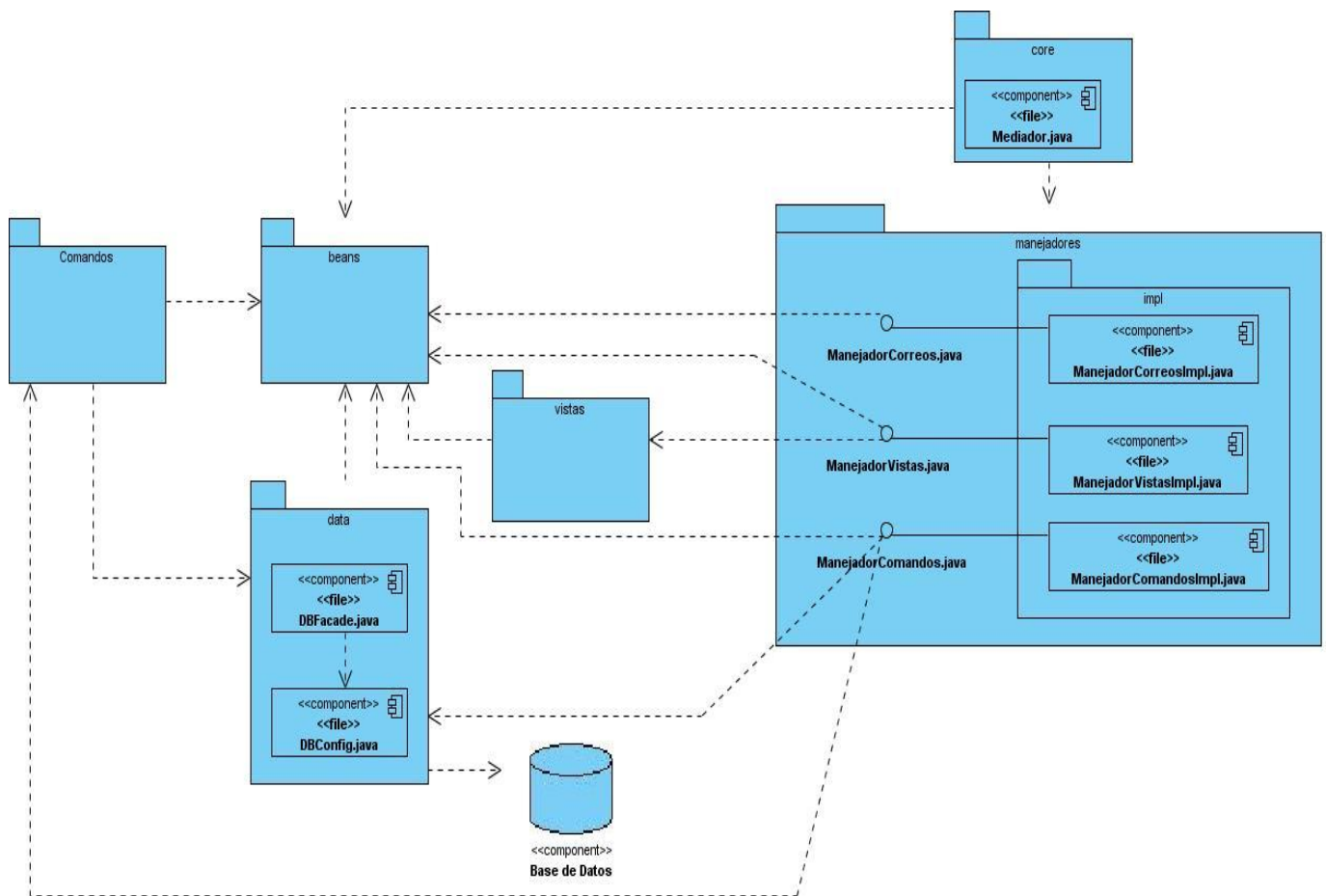


Figura 3 Diagrama de componentes general

El paquete **beans** (Figura 4) contiene los componentes que representan los conceptos del dominio o aquellos creados para la trasportación de la información entre los diferentes componentes del sistema. Por esta razón es utilizado por todos los demás paquetes.

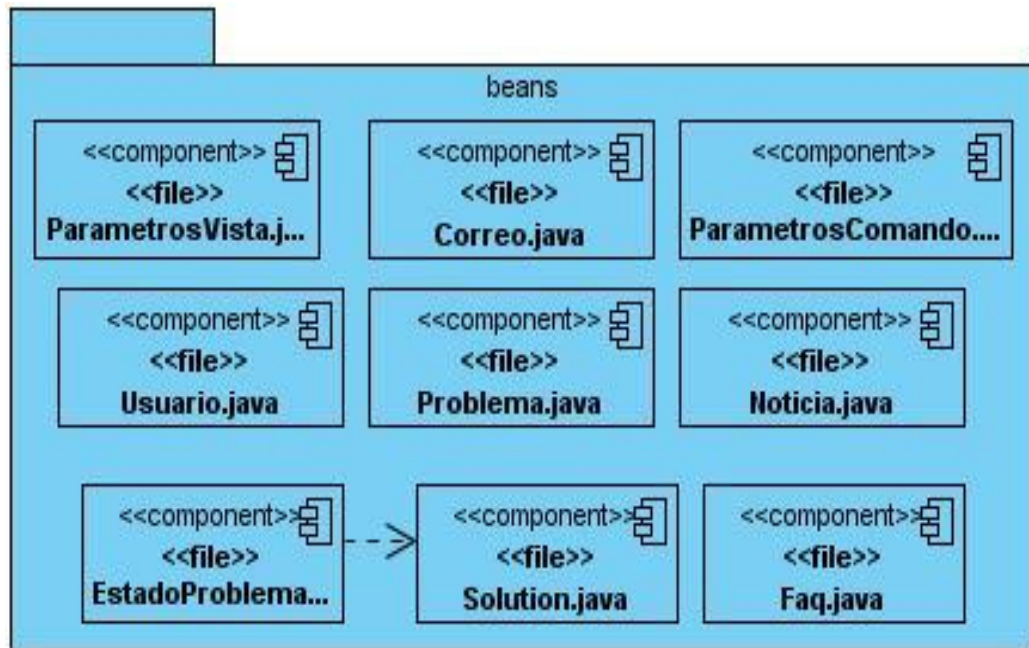


Figura 4 Diagrama de componentes del paquete beans

El paquete **comandos** (Figura 5) se encarga de ejecutar la petición enviada por el usuario. Para esto contiene un componente por cada una de las peticiones que pueden hacerse.

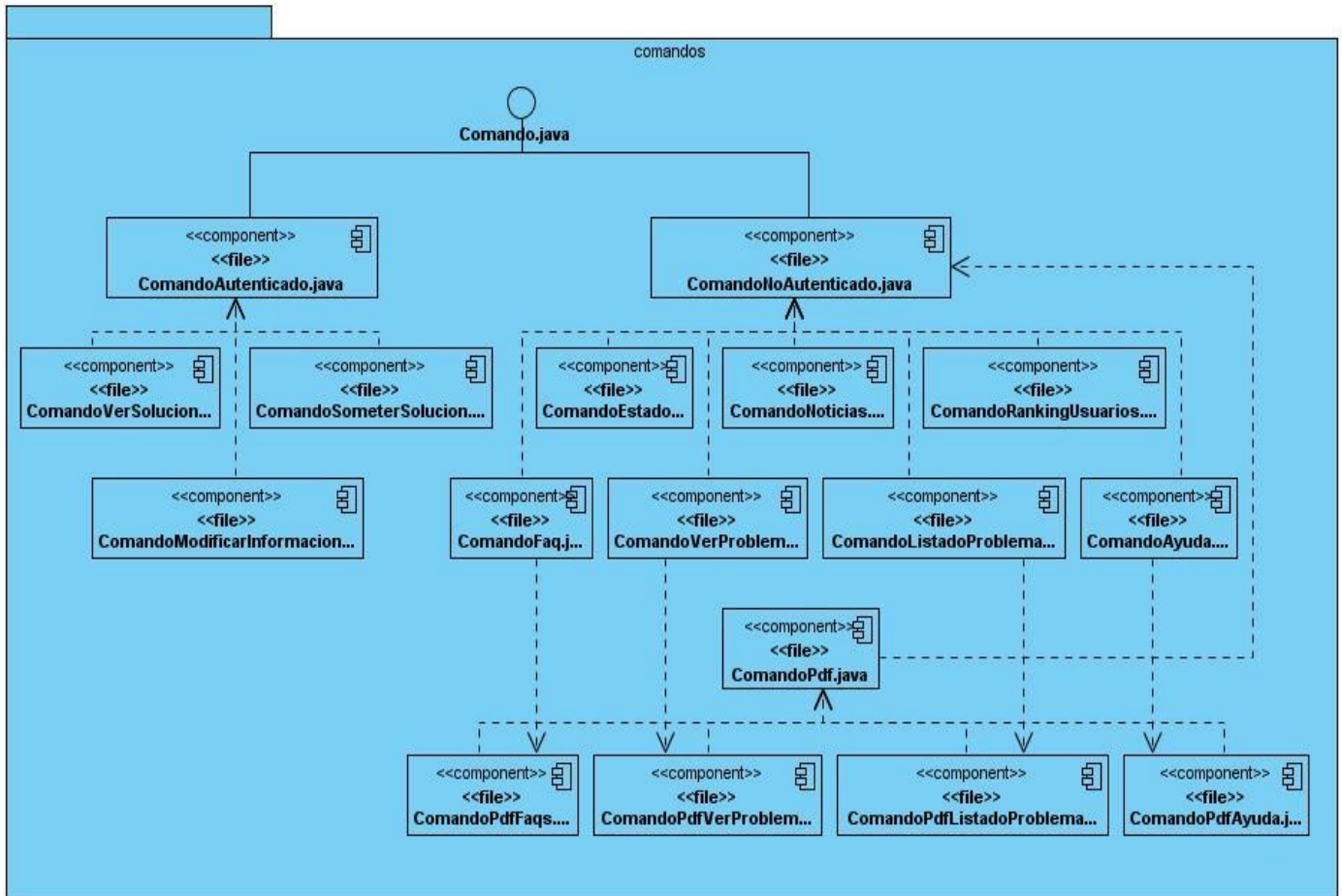


Figura 5 Diagrama de componentes del paquete comandos

El paquete **vistas** (Figura 6) se encarga de configurar la presentación de la información. Análogo al paquete **comandos** contiene un componente para cada una de las vistas que pueden mostrarse.

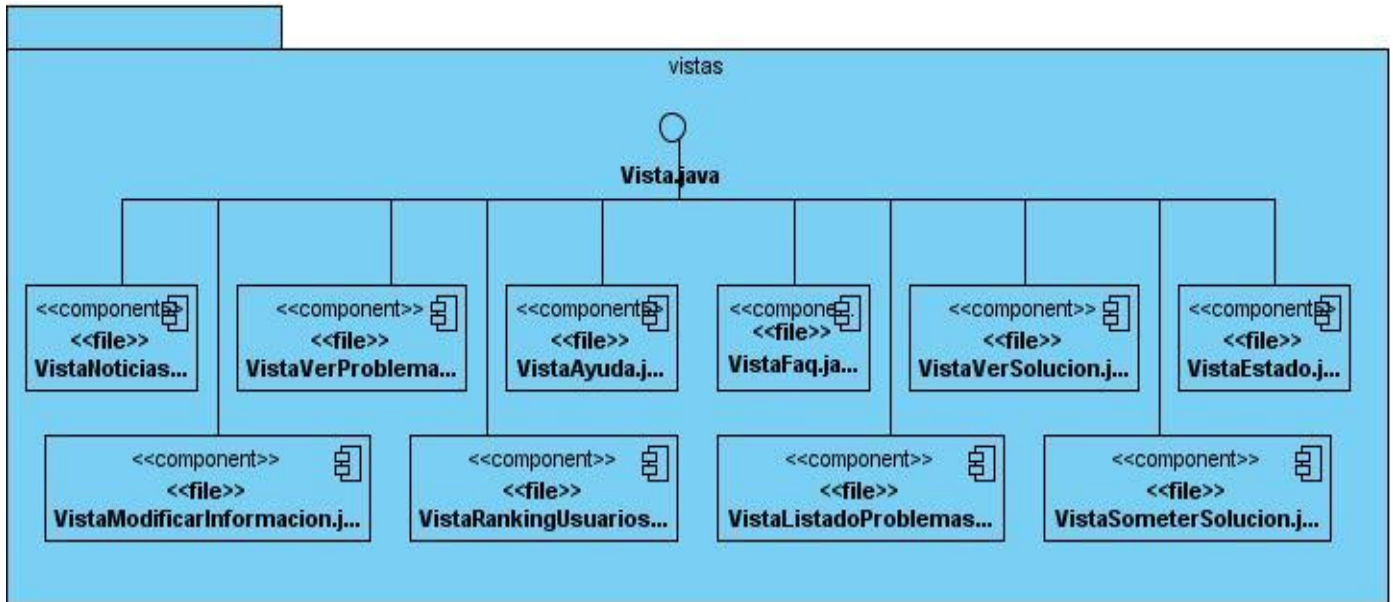


Figura 6 Diagrama de componentes del paquete vistas

## 2.5 Conclusiones

En este capítulo se hizo una breve descripción del funcionamiento de la aplicación para luego presentar cómo está construida a partir de los diagramas de componentes e ilustrar cuáles serán los nodos que se utilizarán para la implantación del sistema en el diagrama de despliegue.

# Capítulo 3

El presente capítulo muestra el modelo de prueba. Se caracteriza el método de pruebas de caja negra y la técnica de partición equivalente. Se definen y describen los casos de prueba de los casos de uso (CU) definidos como arquitectónicamente significativos en el trabajo previo realizado por el analista y se muestra el resultado de las pruebas.

### 3.1 Modelo de prueba

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados y los resultados son observados y registrados. Su principal objetivo es evaluar o valorar la calidad del producto a través de la búsqueda y documentación de errores. Una prueba puede probar la presencia de errores pero no la ausencia de ellos y se considera que tiene éxito si descubre un error no detectado hasta entonces. (21)

#### 3.1.1 Método de prueba

Los métodos se encargan de definir la estrategia de verificación y validación del sistema con el objetivo de demostrar que existen fallos, no la ausencia de estos. Tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Existen dos métodos de pruebas fundamentales: el método de caja negra (denominada prueba funcional) y el de caja blanca.

#### Método de caja negra

Se decide utilizar el **método de caja negra** por la necesidad de abarcar la verificación del funcionamiento de todos los requisitos del sistema. Estas pruebas se centran en lo que se espera de un sistema, es decir, intentan encontrar casos en que el sistema no se atiene a su especificación sin tener mucho en cuenta la estructura interna del software. Permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. (22)

#### Técnica de partición equivalente



Para desarrollar este método existen varias **técnicas**, de las cuales se selecciona la Partición equivalente. Esta técnica divide el dominio de entrada de un programa en clases de datos, a partir de las cuales derivan los casos de prueba.

En el diseño de casos de prueba para partición equivalente se procede en dos pasos (21):

- ❖ Se **identifican las clases de equivalencia**. Las clases de equivalencia son identificadas tomando cada condición de entrada (generalmente una oración o una frase en la especificación) y repartiéndola en dos o más grupos.

Es de notar que dos tipos de clases de equivalencia están identificados: las clases de equivalencia válidas representan entradas válidas al programa, y las clases de equivalencia inválidas que representan el resto de los estados posibles de la condición (es decir, valores erróneos de la entrada).

- ❖ Se **definen los casos de prueba**. El segundo paso es el uso de las clases de equivalencia para identificar los casos de prueba. El proceso es como sigue: se asigna un número único a cada clase de equivalencia. Hasta que todas las clases de equivalencia válidas han sido cubiertas por los casos de prueba, se escribe un nuevo caso de prueba que cubra la clase de equivalencia válida. Por último, hasta que los casos de prueba hayan cubierto todas las clases de equivalencia inválidas, se escribe un caso de la prueba que cubra solamente una de las clases de equivalencia inválidas descubiertas.

### 3.1.2 Descripción de los casos de pruebas

Un caso de prueba es un conjunto de condiciones o variables bajo las cuales se determina si el requisito de una aplicación es parcial o completamente satisfactorio. Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Es una forma de probar el sistema para demostrar que las funciones del software son operativas, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que se debe probar.

A continuación se describen los casos de prueba desarrollados para los CU identificados como arquitectónicamente significativos en la investigación a la cual este trabajo le da continuidad. Dichos CU son: Ver problema, Someter solución de un problema y Consultar el estado del servidor. En los casos de prueba se recoge la información de entrada, los resultados obtenidos una vez ejecutado el caso de prueba

y las condiciones que deben cumplirse mientras este se ejecuta. Los casos de pruebas de los restantes CU se encuentran como anexos.

### 3.1.2.1 Diseño de caso de prueba CU Consultar el estado del servidor

#### Descripción general

El caso de uso se inicia cuando el usuario envía un mensaje de correo al sistema y en el asunto especifica los criterios que desea consultar del estado del servidor. El sistema valida que la petición del usuario esté en el formato correcto. Construye el estado del servidor atendiendo al rango y los criterios de consulta especificados por el usuario y le envía a este un mensaje de correo con los criterios del estado del servidor que solicitó consultar. Finaliza así el caso de uso.

#### Condiciones de ejecución

El usuario debe haber accedido al correo electrónico.

#### Secciones a probar en el caso de uso

Nombre de Escenarios la sección	Descripción de la funcionalidad de la sección	Flujo central
------------------------------------	--	---------------

<b>SC 1:</b> Consultar el estado del servidor.	<b>EC 1.1:</b> Consultar el estado del servidor exitosamente.	Luego de que el usuario envía un correo especificando en el asunto los criterios que desea consultar del estado del servidor, con el siguiente formato: mj:estado <variables> Ej. (mj:estado -t servidor -u arosa -r ac -n 1-500 -l java) El sistema valida el formato del asunto del correo. Construye el estado del servidor atendiendo al rango y los criterios de consulta especificados: <ul style="list-style-type: none"> <li>❖ Usuario</li> <li>❖ Tipo de respuesta</li> <li>❖ Lenguaje</li> <li>❖ Número de problema</li> </ul> Envía un mensaje de correo donde muestra los criterios del estado del servidor que se solicitó consultar.	Recibir correo de usuario Validar asunto Construir estado Enviar correo a usuario
	<b>EC 1.2:</b> Envía mensaje de correo que no se corresponde con el formato definido.	Envía mensaje de correo con la ayuda.	Recibir correo de usuario Validar asunto (asunto inválido) Construir ayuda Enviar correo a usuario

**Tabla 2 Secciones a probar en el CU Consultar el estado del servidor**

### Descripción de la variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Asunto	Texto	No	<p>Está estructurado de la siguiente forma:</p> <p><b>mj:estado &lt;variables&gt;</b> (se puede poner el <b>mj:</b> separado por espacios de la palabra <b>estado</b>)</p> <p>Las variables están compuestas por los pares <b>&lt;critério&gt; &lt;valor&gt;</b>.</p> <p>Existen 6 variables:</p> <ul style="list-style-type: none"> <li>❖ Tipo de estado</li> <li>❖ Usuario</li> <li>❖ Tipo de respuesta</li> <li>❖ Rango</li> <li>❖ Lenguaje</li> <li>❖ Número de problema</li> </ul>
2	Tipo de estado	Texto	Si	Está estructurado de la siguiente forma: <b>-t servidor</b> .
3	Usuario	Texto	Si	Está estructurado de la siguiente forma: <b>-u &lt;valor&gt;</b> donde valor es una cadena compuesta por caracteres alfanuméricos y representa el identificador de un usuario en la base de datos.
4	Tipo de respuesta	Texto	Si	<p>Está estructurado de la siguiente forma: <b>-r &lt;valor&gt;</b> donde valor puede ser:</p> <ul style="list-style-type: none"> <li>❖ <b>ac</b> (aceptado)</li> <li>❖ <b>wa</b> (respuesta incorrecta)</li> <li>❖ <b>rte</b> (error en tiempo de ejecución)</li> <li>❖ <b>tle</b> (tiempo límite excedido)</li> <li>❖ <b>mle</b> (memoria límite excedida)</li> <li>❖ <b>ce</b> (error de compilación)</li> <li>❖ <b>pe</b> (error de presentación)</li> </ul>

5	Rango	Texto	Si	Está estructurado de la siguiente forma: <b>-n &lt;valor&gt;</b> donde valor es un rango de números con la forma #-#. Este rango no debe exceder los 500 y su menor valor es 1. Ej. <b>(1-500)</b> . En caso de que no se especifique esta variable, se asume que es <b>-n 1-500</b> .
6	Lenguaje	Texto	Si	Está estructurado de la siguiente forma: <b>-l &lt;valor&gt;</b> donde valor puede ser alguno de los lenguajes que maneje el jurado Xtreme, en estos momentos serían: <ul style="list-style-type: none"> <li>❖ <b>g++</b></li> <li>❖ <b>gcc</b></li> <li>❖ <b>pascal</b></li> <li>❖ <b>java</b></li> <li>❖ <b>c#</b></li> <li>❖ <b>python</b></li> <li>❖ <b>texto</b></li> <li>❖ <b>bf</b> (brain fak)</li> </ul>
7	Número del problema	Texto	Si	Está estructurado de la siguiente forma: <b>-i &lt;valor&gt;</b> donde valor es un número a partir de mil que coincide con el identificador de un problema almacenado en la base de datos.

**Tabla 3 Descripción de la variable caso de prueba CU Consultar el estado del servidor**

### Matriz de Datos

#### SC #1: Consultar el estado del servidor

ID escenario:		EC 1.1		Escenario:		Consultar el estado del servidor exitosamente.		
V2	V3	V4	V5	V6	V7	V1	Respuesta del sistema	Resultado de la prueba
"" (V)	"" (V)	"" (V)	"" (V)	"" (V)	"" (V)	"mj:estado" (V)	Envía un mensaje de correo donde	Se envió el estado pero tenía una solución menos

							muestra los que la	critérios del especificada en el	estado del rango.	
“-t servid or” (V)	“-u ndaco sta” (V)	“-r ac” (V)	“-n 1- 1” (V)	“-l g++” (V)	“-i 1023” (V)	“mj:estado -t servid or -u ndacosta -r ac -n 1-1 -l g++ -i 1023” (V)	servid or que se solicitó consultar.	Se envió el estado pero no muestra la solución.		
<b>ID escenario:</b>		<b>EC 1.2</b>	<b>Escenario:</b>			Envía mensaje de correo que no se corresponde con el formato definido.				
<b>V2</b>	<b>V3</b>	<b>V4</b>	<b>V5</b>	<b>V6</b>	<b>V7</b>	<b>V1</b>	<b>Respuesta del sistema</b>		<b>Resultado de la prueba</b>	
“-t servid or” (I)	“(V)	“(V)	“(V)	“(V)	“(V)	“mj:estado -t servid or” (I)	Envía mensaje de correo con la ayuda.		Envía mensaje de correo con la ayuda.	
“(V)	“(V)	“-r Pe” (I)	“(V)	“(V)	“(V)	“mj:estado -r Pe” (I)				
“(V)	“(V)	“(V)	“-n 1- 501” (I)	“(V)	“(V)	“mj:estado -n 1- 501” (I)				
“(V)	“(V)	“(V)	“(V)	“-l Gcc” (I)	“(V)	“mj:estado -l Gcc” (I)				
“(V)	“(V)	“(V)	“-n -5- 60” (I)	“(V)	“(V)	“mj:estado -n -5- 60” (I)				
“(V)	“(V)	“(V)	“-n 500- 60” (I)	“(V)	“(V)	“mj:estado -n 500- 60” (I)				
“(V)	“(V)	“(V)	“(V)	“(V)	“-i 10236 50000	“mj:estado -i 1023650000000000 00000000000” (I)	No envía respuesta.			

					00000			
					00000			
					00000			
					0" (l)			

**Tabla 4 Matriz de datos caso de prueba CU Consultar el estado del servidor**

### 3.1.2.2 Diseño de caso de prueba CU Ver problema

#### Descripción general

El caso de uso se inicia cuando el usuario envía un mensaje de correo al sistema y en el asunto especifica que le muestre la información de un problema en específico. El sistema valida el formato del asunto del correo, busca el problema y le envía un mensaje de correo con la información del mismo. Finaliza así el caso de uso.

#### Condiciones de ejecución

El usuario debe haber accedido al correo electrónico.

#### Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
<b>SC 1:</b> Ver problema.	<b>EC 1.1:</b> Ver problema en el cuerpo del correo.	<p>Luego de que el usuario envía un mensaje de correo especificando en el asunto el siguiente formato:</p> <p>mj:problema -i &lt;# de problema que desea ver&gt;</p> <p>El sistema valida el formato del asunto del correo. Busca el problema.</p> <p>Envía un mensaje de correo y en el cuerpo del mismo muestra el problema.</p>	<p>Recibir correo de usuario</p> <p>Validar el asunto</p> <p>Buscar el problema</p> <p>Enviar correo a usuario</p>

	<p><b>EC 1.2:</b> Ver problema en formato PDF.</p>	<p>Luego de que el usuario envía un mensaje de correo especificando en el asunto el siguiente formato:  mj:problema -i &lt;# de problema que desea ver&gt; -f pdf  El sistema valida el formato del asunto del correo. Busca el pdf correspondiente a ese problema en caso de que exista, sino lo construye.  Envía mensaje de correo con el problema adjunto en formato PDF.</p>	<p>Recibir correo de usuario  Validar el asunto  Construir el problema  Enviar correo a usuario</p>
	<p><b>EC 1.3:</b> Envía mensaje de correo que no se corresponde con el formato definido.</p>	<p>Envía mensaje de correo con la ayuda.</p>	<p>Recibir correo de usuario  Validar el asunto (inválido)  Construir la ayuda  Enviar correo a usuario</p>

**Tabla 5 Secciones a probar en el CU Ver problema**

### Descripción de la variable

No de campo	Nombre	Clasificación	Valor	Descripción
			Nulo	



1	Asunto	Texto	No	<p>Está estructurado de la siguiente forma:</p> <p><b>mj:problema &lt;variables&gt;</b> (se puede poner el <b>mj:</b> separado por espacios de la palabra <b>problema</b>).</p> <p>Las variables están compuestas por los pares <b>&lt;critério&gt; &lt;valor&gt;</b>.</p> <p>Existen 2 variables:</p> <ul style="list-style-type: none"> <li>❖ Número del problema</li> <li>❖ Tipo de formato</li> </ul>
1	Número del problema	Texto	No	<p>Está estructurado de la siguiente forma: <b>-i &lt;valor&gt;</b> donde valor es un número a partir de mil que coincide con el identificador de un problema almacenado en la base de datos.</p>
2	Tipo de formato	Texto	Si	<p>Está estructurado de la siguiente forma: <b>-f pdf</b>.</p>

**Tabla 6 Descripción de la variable caso de prueba CU Ver problema**

### Matriz de Datos

#### SC #1: Ver problema

ID del Escenario	V2	V3	V1	Respuesta del sistema	Resultado de la prueba	
<b>EC 1.1:</b>	Ver problema en el cuerpo del correo.	"-i 1000" (V)	"" (V)	"mj:problema - i 1000" (V)	Envía un mensaje de correo y en el cuerpo del mismo muestra el problema.	Envía el problema exitosamente.
<b>EC 1.2:</b>	Ver problema en formato PDF.	"-i 1467" (V)	"-f pdf" (V)	"mj:problema - i 1467 -f pdf" (V)	Envía mensaje de correo con el problema adjunto en formato PDF.	Envía el problema adjunto en formato PDF exitosamente.

<b>EC 1.3:</b>	Envía mensaje de correo que no se corresponde con el formato definido.	“-i 999” (l)	“” (V)	“mj:problema - i 999” (l)	Envía mensaje de correo con la ayuda.	Envía mensaje de correo con la ayuda.
		“-i hji” (l)	“-f pdf” (V)	“mj:problema - i hji -f pdf” (i)		
		“-i 1256” (V)	“-f Pdf” (l)	“mj:problema - f Pdf -i 1256” (V)		
		“-i 1000000 0000000 0” (l)	“” (V)	“mj:problema - i 100000000000 0000” (l)		

**Tabla 7 Matriz de datos caso de prueba CU Ver problema**

### 3.1.2.3 Diseño de caso de prueba CU Someter solución de un problema

#### Descripción general

El caso de uso se inicia cuando el usuario envía un mensaje de correo al sistema especificando en el cuerpo del mismo el código que desea sea evaluado por el jurado. El sistema valida el formato del correo y que el usuario esté registrado. Guarda la solución y le envía al usuario un mensaje de confirmación con el id de la misma. Finaliza así el caso de uso.

#### Condiciones de ejecución

El usuario debe estar registrado en el sistema y haber accedido al correo electrónico.

#### Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
----------------------	--------------------------	---------------------------------	---------------

<b>SC 1:</b> Someter solución de un problema.	<b>EC 1.1:</b> Someter solución de un problema exitosamente.	<p>Luego de que el usuario envía un correo especificando en el asunto el siguiente formato:</p> <p>mj:someter -l &lt;lenguaje&gt; -i &lt;# de problema&gt;</p> <p>Ej. (mj:someter -l java -i 1234)</p> <p>Y en el cuerpo del correo especifica el siguiente formato:</p> <p>&lt;separador inicio&gt; &lt;código&gt; &lt;separador final&gt;</p> <p>El sistema valida el formato del asunto del correo y que el usuario tenga permiso para someter la solución. Extrae el código enviado por el usuario delimitado por los separadores. Guarda el código en la base de datos.</p> <p>Envía un mensaje de confirmación informando que la solución fue recibida y el identificador de la misma.</p>	<p>Recibir correo de usuario</p> <p>Validar asunto y permisos</p> <p>Guardar la solución</p> <p>Enviar correo a usuario</p>
	<b>EC 1.2:</b> Envía mensaje de correo que no se corresponde con el formato definido o no tiene permisos para modificar la información.	<p>Envía mensaje de correo con la ayuda.</p>	<p>Recibir correo de usuario</p> <p>Validar asunto y permisos (asunto o permisos inválido)</p> <p>Construir la ayuda</p> <p>Enviar correo a usuario</p>

**Tabla 8 Secciones a probar en el CU Someter solución de un problema**

### Descripción de la variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Asunto	Texto	No	<p>Está estructurado de la siguiente forma:</p> <p><b>mj:someter &lt;variables&gt;</b> (se puede poner el <b>mj:</b> separado por espacios de la palabra <b>someter</b>).</p> <p>Las variables están compuestas por los pares <b>&lt;critério&gt;</b> <b>&lt;valor&gt;</b>.</p> <p>Existen 2 variables:</p> <ul style="list-style-type: none"> <li>❖ Número del problema</li> <li>❖ Lenguaje</li> </ul>
2	Número de problema	Texto	No	<p>Está estructurado de la siguiente forma: <b>-i &lt;valor&gt;</b> donde valor es un número a partir de mil que coincide con el identificador de un problema almacenado en la base de datos.</p>
3	Lenguaje	Texto	No	<p>Está estructurado de la siguiente forma: <b>-l &lt;valor&gt;</b> donde valor puede ser alguno de los lenguajes que maneje el jurado Xtreme, en estos momentos serían:</p> <ul style="list-style-type: none"> <li>❖ <b>g++</b></li> <li>❖ <b>gcc</b></li> <li>❖ <b>pascal</b></li> <li>❖ <b>java</b></li> <li>❖ <b>c#</b></li> <li>❖ <b>python</b></li> <li>❖ <b>texto</b></li> <li>❖ <b>bf</b> (brain fak)</li> </ul>
4	Cuerpo	Texto	No	<p>Está estructurado de la siguiente forma:</p> <p><b>&lt;--PRINCIPIO DE CODIGO &lt;código&gt; /FIN CODIGO--&gt;</b> La estructura de <b>&lt;código&gt;</b> se explica debajo.</p>

5	Código	Texto	Si	Cadena de caracteres.
---	--------	-------	----	-----------------------

**Tabla 9 Descripción de la variable caso de prueba CU Someter solución de un problema**

### Matriz de Datos

#### SC #1: Someter solución de un problema

ID del escenario	Escenario	V2	V3	V1	V5	V4	Respuesta del sistema	Resultado de la prueba
EC 1.1:	Someter solución de un problema exitosamente.	"-i 1234" (V)	"-l java" (V)	"mj: someter -i 1234 -l java" (V)	"" (V)	"<-- PRINCIPIO DE CODIGO /FIN CODIGO->" (V)	Envía un mensaje de confirmación informando que la solución fue recibida y el identificador de la misma.	Envió la ayuda.
EC 1.2:	Envía mensaje de correo que no se corresponde con el formato definido o no tiene permisos para modificar la información.	"" (I)	"-l g++" (V)	"mj: someter -l g++" (I)	"#include <stdio.h> > int a,b; int main() { scanf("%d%d", &a, &b); printf("%	"<-- PRINCIPIO DE CODIGO #include <stdio.h> int a,b; int main() { scanf("%d%d", &a, &b); printf("%d\n", a +b);	Envía mensaje de correo con la ayuda.	Envió la ayuda.

					d\n",a+b ); return 0; }" (V)	return 0; } /FIN CODIGO- ->" (V)		
		"-i 1234" (V)	"" (I)	"mj: someter -i 1234" (I)	"" (V)	"<-- PRINCIPIO DE CODIGO /FIN CODIGO- ->" (V)		
		"-i 1234" (V)	"-l texto" (V)	"mj: someter -i 1234 -l texto" (V)	"" (V)	"/FIN CODIGO--> <- -PRINCIPIO DE CÓDIGO" (I)		
		"-i 1234" (V)	"-l texto" (V)	"mj: someter -i 1234 -l texto" (V)	"" (V)	"" (I)		

**Tabla 10 Matriz de datos caso de prueba CU Someter solución de un problema**

### 3.1.3 Resultados

Se planificaron 2 iteraciones en las cuales se probó la aplicación siguiendo lo especificado en cada caso de prueba. Cada iteración abarcó la verificación del correcto funcionamiento de todos los requisitos del sistema. Luego de la primera iteración se encontraron un grupo de no conformidades que fueron erradicadas. Luego de la segunda iteración no se encontraron no conformidades por lo que se decidió concluir las pruebas. A continuación se muestra una gráfica que ilustra la cantidad de no conformidades encontradas por caso de uso seguido de las tablas que las recogen en detalle.

## No Conformidades

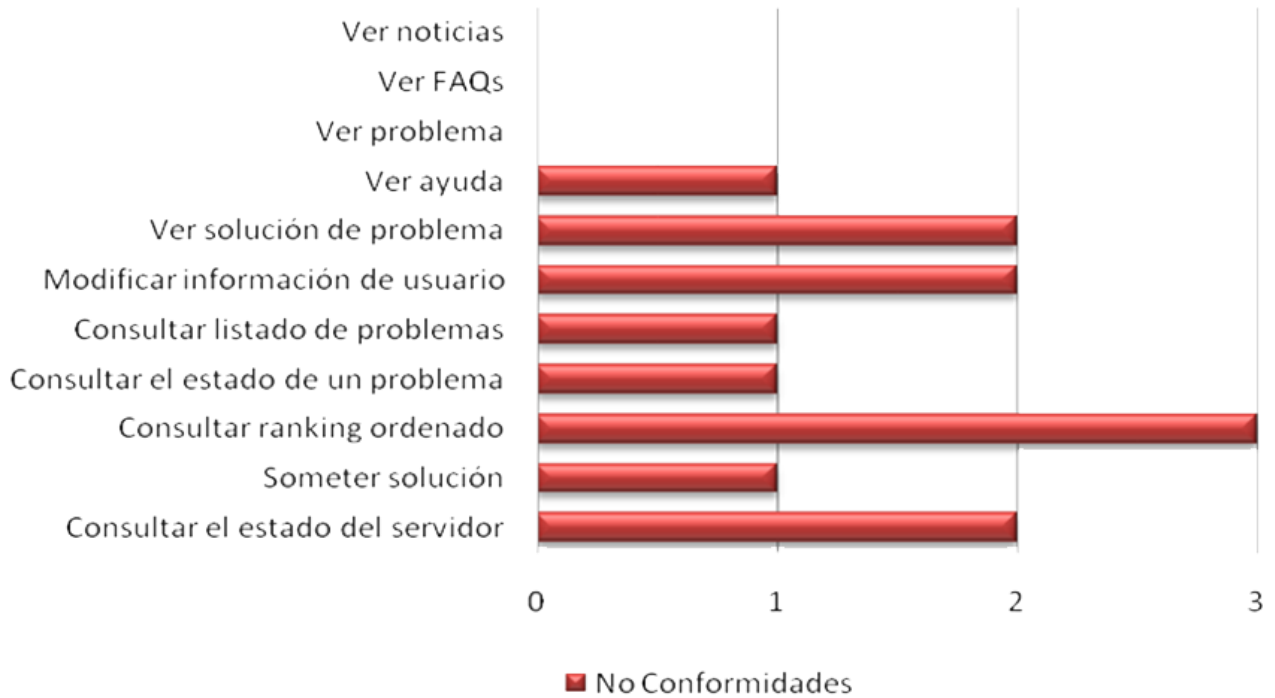


Figura 7 Cantidad de no conformidades por caso de uso

Elemento	No. No conformidad	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No significativa
Asunto correo	1	El correo enviado por la aplicación con el estado no concuerda con el que define la ECU.	El estado que se envió tenía una solución menos que la especificada en el rango.	1ra iteración	X	
Asunto correo	2	No envía correo de respuesta.	No envía correo de respuesta.	1ra iteración	X	

Tabla 11 No conformidades del caso de prueba CU Consultar el estado del servidor

Elemento	No. No conformidad	Aspecto correspondiente	Etapa de detección	Significativa	No significativa	
Cuerpo correo	1	El correo enviado por la aplicación no concuerda con el que define la ECU.	Se envió la ayuda.	1ra iteración	X	

**Tabla 12 No conformidades del caso de prueba CU Someter solución de un problema**

Elemento	No. No conformidad	Aspecto correspondiente	Etapa de detección	Significativa	No significativa	
Asunto correo	1	El correo enviado por la aplicación no concuerda con el que define la ECU.	El ranking que se envió estaba vacío.	1ra iteración	X	
Asunto correo	2	El correo enviado por la aplicación no concuerda con el que define la ECU.	El ranking que se envió tenía un usuario más que lo especificado en el rango.	1ra iteración	X	
Asunto correo	3	No envía correo de respuesta.	No envía correo de respuesta.	1ra iteración	X	

**Tabla 13 No conformidades del caso de prueba CU Consultar ranking ordenado**

Elemento	No. No conformidad	Aspecto correspondiente	Etapa de detección	Significativa	No significativa	
Asunto correo	1	El correo enviado por la aplicación no concuerda con el que define la ECU.	Manda el estado del problema 1020 ordenado por memoria cuando debiera mandar la ayuda.	1ra iteración	X	

**Tabla 14 No conformidades del caso de prueba CU Consultar el estado de un problema**



Elemento	No. No conformidad	Aspecto correspondiente	Etapa de detección	Significativa	No significativa	
Asunto correo	1	No envía respuesta.	No envía respuesta.	1ra iteración	X	

**Tabla 15 No conformidades del caso de prueba CU Consultar listado de problemas**

Elemento	No. No conformidad	Aspecto correspondiente	Etapa de detección	Significativa	No significativa	
Asunto correo	1	El correo enviado por la aplicación no concuerda con el que define la ECU.	Envía un mensaje diciendo que lo siente pero que no se pudo modificar la información.	1ra iteración	X	
Asunto correo	2	El correo enviado por la aplicación no concuerda con el que define la ECU.	Envía mensaje de correo con la ayuda.	1ra iteración	X	

**Tabla 16 No conformidades del caso de prueba CU Modificar información de usuario**

Elemento	No. No conformidad	Aspecto correspondiente	Etapa de detección	Significativa	No significativa	
Asunto correo	1	No envía respuesta.	No envía respuesta.	1ra iteración	X	
Asunto correo	2	La palabra código está mal escrita.	Envía la solución pero la palabra código está mal escrita.	1ra iteración	X	

**Tabla 17 No conformidades del caso de prueba CU Ver solución de problema**

Elemento	No. No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No significativa	
Asunto correo	1	La ayuda enviada posee errores ortográficos.	Error ortográfico. Palabra filtrándolo se encuentra sin tilde y esta puesto dirrección en vez de dirección.	1ra iteración	X	

**Tabla 18 No conformidades del caso de prueba CU Ver ayuda**

### 3.2 Conclusiones

En este capítulo se realizó el modelo de pruebas reuniéndose las descripciones de los casos de prueba para los casos de uso arquitectónicamente significativos. Se probaron todas las funcionalidades del sistema. Producto de las pruebas se obtuvieron un grupo de no conformidades a las cuales se les dio respuesta.

### Conclusiones Generales

El presente trabajo se centró en la implementación y pruebas de un jurado online con intercambio de información sobre protocolos de correo electrónico que provee información en formato PDF. La finalidad del mismo es dar una solución al problema de accesibilidad a este tipo de software que se presenta en la generalidad de los entornos universitarios cubanos. Durante el desarrollo del trabajo se le dieron cumplimiento a los diversos objetivos propuestos y se arribó a las siguientes conclusiones:

- ❖ Todos los jurados online pueden reducirse a un conjunto de funcionalidades básicas. Estas sirven de modelo para crear implementaciones de este tipo de aplicaciones fuera de su paradigma web tradicional.
- ❖ Las herramientas libres de generación de archivos PDF existentes en el mundo Java son de alta complejidad, requiriendo un conocimiento profundo de las mismas. No existe en ninguna de estas herramientas una forma sencilla y rápida de transformar contenido HTML a contenido PDF manteniendo el formato original. iText es la herramienta libre más sencilla de manipular para lograr una conversión de calidad entre estos formatos.
- ❖ Al estudiar las metodologías de desarrollo de software se pudo observar que RUP es el proceso más adecuado para generar documentación suficiente que permita continuar desarrollando versiones superiores del producto con personal distinto del equipo inicial.
- ❖ Se comprobó que es posible representar todas las funcionalidades básicas de un jurado online sobre los protocolos de correo electrónico SMTP e IMAP4. Es posible para los usuarios acceder a la misma información disponible sobre HTTP, pero beneficiándose de las características de protocolos más desconectados.
- ❖ Existe un conjunto mínimo de información necesaria y suficiente que los usuarios pueden usar de manera desconectada sin necesitar acceder a la aplicación online durante intervalos prolongados de tiempo. Esta información fue la seleccionada para convertir a formato PDF.
- ❖ Como resultado de las pruebas se considera que el sistema es factible para su utilización en aquellas universidades con problemas de conectividad en calidad de complemento a una instancia tradicional.

### Recomendaciones

Con el objetivo de hacer más eficiente y funcional el sistema que se implementó se recomienda:

- ❖ Hacer estándar el contenido HTML presente en la base de datos.
- ❖ Desplegar el sistema en las universidades del país que presenten problemas en la disponibilidad de la red.
- ❖ Incluir en el sistema las funcionalidades de competencias y administración.
- ❖ Concebir y desarrollar una segunda versión del software con las funcionalidades de competencias y administración.
- ❖ Concebir y desarrollar un sistema a utilizar por los usuarios en las máquinas clientes que envíe correos con las peticiones que se seleccionen y muestre las respuestas. De esta forma el usuario no está obligado a memorizar el micro lenguaje establecido para la comunicación, solo debe seleccionar la información que desee.

## Referencias Bibliográficas

1. **Ramirez, Jorge Amado Soria and Castillo, Enrique José Altuna.** *Análisis, diseño e implementación de un Jurado Online*. Ciudad de la Habana : s.n., 2008.
2. **Revilla, Miguel A., Manzoor, Shahriar and Liu, Rujia.** *www.edujudge.eu*. [Online] [Cited: 12 15, 2010.] [http://www.edujudge.eu/pdfs/revilla\\_ioi\\_final.pdf](http://www.edujudge.eu/pdfs/revilla_ioi_final.pdf).
3. UVa Online Judge. [Online] [Cited: noviembre 20, 2010.] <http://acm.UVa.es>.
4. Sphere Online Judge. [Online] [Cited: noviembre 14, 2010.] <http://www.spoj.pl>.
5. PKU Online Judge. [Online] [Cited: noviembre 14, 2010.] <http://poj.org>.
6. Xtreme Online Judge. [Online] [Cited: noviembre 14, 2010.] <http://onlinejudgef8.uci.cu:5800/JudgeOnline>.
7. Cátedra de programación Avanzada. [Online] [Cited: noviembre 14, 2010.] <http://cpav.uci.cu>.
8. **Solís, Manuel Calero.** *www.apolosoftware.com*. [Online] [Cited: 12 16, 2010.] <http://www.willydev.net/descargas/prev/explicaxp.pdf>.
9. **Palacio, Juan.** *navegapolis.net*. [Online] 2006. [Cited: 12 15, 2010.] [http://www.navegapolis.net/files/s/NST-010\\_01.pdf](http://www.navegapolis.net/files/s/NST-010_01.pdf).
10. **Shuja, Ahmad K. and Krebs., Jochen.** *IBM Rational Unified Process Reference and Certification Guide: Solution Designer (RUP)*. s.l. : IBM Press, 2008. ISBN-10: 0-13-156292-4; ISBN-13: 978-0-13-156292-9.
11. **Tuesta, Martín.** *El Prisma*. [Online] 2005. [Cited: enero 20, 2010.] <http://www.elprisma.com/apuntes/curso.asp?id=13324>.
12. **González Blanco, Rubén and Pérez Tobalina, Sergio.** *LESE-2 Introducción a Rational Rose*. Barcelona : s.n.
13. **Sierra, María.** *Trabajando con Visual Paradigm for UML*. Cantabria : s.n.

14. **Myers, J., Mellon, Carnegie and Rose, M.** The Internet Engineering Task Force (IETF). [Online] [Cited: 12 20, 2010.] <http://www.ietf.org/rfc/rfc1939.txt>.
15. **Crispin, M.** The Internet Engineering Task Force (IETF). [Online] marzo 2003. [Cited: diciembre 15, 2010.] <http://www.ietf.org/rfc/rfc3501.txt>.
16. ORACLE. [Online] [Cited: enero 13, 2011.] <http://www.oracle.com/technetwork/java/faq-135477.html#1>.
17. **Lowagie, Bruno.** *iText in Action*. New York : Manning Publications Co., 2007.
18. **Lampport, Leslie.** *Latex: A Document Preparation System(2nd Edition)*. ISBN 0-201-52983-1.
19. JasperForge. [Online] [Cited: enero 17, 2011.] <http://jasperforge.org/projects/jasperreports>.
20. **Derin, Onur.** docstoc: Document for Small Bussiness & Professionals. [Online] marzo 01, 2005. [Cited: enero 14, 2011.] <http://www.docstoc.com/docs/23951918/A-Tutorial-on-Reporting-in-JAVA-using-JasperReports-iReport>.
21. **Myers, Glenford J.** *The art of sotware Testing*. Segunda Edición. s.l. : Wiley, 2004. p. 234.
22. Quality Assurance & Software Testing. [Online] 2009. [Cited: 04 29, 2011.] [http://www.calidadyssoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadyssoftware.com/testing/pruebas_funcionales.php).

### Glosario de Términos

**AC:** Código de respuesta aceptada. Estado de una sumisión exitosa.

**ACM:** Acrónimo de **Association for Computing Machinery**, organización fundada en 1947 como la primera sociedad científica y educativa acerca de la Computación. Publica varias revistas y periódicos científicos relacionados con la computación; patrocina conferencias y otros eventos relacionados con las ciencias de la computación como por ejemplo el **ACM International Collegiate Programming Contest (ICPC)**.

**CMS:** Un sistema de manejo de contenido o CMS es un software usado para crear, editar, administrar y publicar contenido de forma consistente. Los CMSs son utilizados frecuentemente para almacenar, controlar, versionar y publicar documentación específica como artículos de noticias, manuales, guías de venta y propaganda de marketing. El contenido que puede manejarse incluye ficheros, imágenes, audio, documentos electrónicos y contenidos Web.

**Ayuda:** Especificación del micro lenguaje utilizado para permitir la comunicación vía correo electrónico entre el usuario y el jurado.

**Código legado:** Dentro de un proyecto de software dicese del código anterior al desarrollo que coexiste en producción junto al código de la aplicación que está siendo escrito. El código legado comporta problemas de compatibilidad con el resto de la aplicación, pero puede ser necesario mantenerlo debido a razones diversas de funcionalidades, facilidad de uso del cliente, velocidad, etc.

**Correo:** El correo electrónico o email es un mensaje enviado por el usuario al jurado o viceversa, que contiene la información referida a cada una de las peticiones o funcionalidades del sistema.

**CPAV:** Cátedra de programación avanzada, nombre con el que surge un proyecto en la UCI en el curso 2006-2007, con el objetivo de agrupar a los estudiantes con interés o conocimiento para formar un grupo que promueva el estudio de la algoritmia así como su aplicación en la producción.

**Daemon:** Programa de computadora que se ejecuta en el trasfondo y no bajo el control de un usuario. Normalmente se usan para tareas de mantenimiento constante que no requieren de la presencia de un usuario.

**Envío:** Ver **sumisión**.

**Estado o estado de un problema:** El estado de un problema determinado se compone del conjunto de soluciones que se le ha dado. El estado aporta mucha información al usuario experto sobre el problema, información que se puede inferir a partir de los datos públicos de consumo de memoria, longitud de código y tiempo.

**Estado actual o del servidor:** El estado actual de la aplicación es el subconjunto más actual de sumisiones que se han hecho, sin tomar en cuenta problema o usuario. El estado actual es una herramienta útil para evaluar la actividad actual de la aplicación, los ejercicios que despiertan más interés y los lenguajes más utilizados, entre otros datos de interés. El tamaño del subconjunto que se considera estado actual varía de una aplicación a otra.

**Evaluador automático:** Aplicación que alivia el trabajo de calificación de soluciones programáticas, moviendo su responsabilidad del humano a la máquina y tratando de lograr mejoras en cuanto al chequeo de correctitud, robustez y eficiencia.

**FAQ (Frequently asked questions):** Constituyen las preguntas frecuentes dentro del contexto del jurado online.

**Framework:** es una estructura conceptual y tecnológica de soporte, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

**ICPC:** El ACM **International Collegiate Programming Contest** (abreviado como ACM-ICPC o solo ICPC) es una competición anual de programación en varios niveles que se celebra entre numerosas universidades del mundo. Este concurso es patrocinado por IBM, y tiene su sede en la Universidad de Baylor. El ICPC define regiones autónomas en seis continentes y opera bajo los auspicios de la ACM.

**Información:** Es la información del usuario, incluye: su información personal (nombre, usuario, país y correo), los puntos que acumula en el jurado, la cantidad de soluciones que tiene aceptadas, el por ciento de soluciones aceptados con respecto al total de soluciones y la cantidad total de soluciones.



**Jurado en línea:** Evaluador automático desplegado en entornos generalmente académicos presentado en la Web (o en cualquier otro soporte igualmente masivo). Provee problemas y sirve para evaluar automáticamente las soluciones algorítmicas que los usuarios desarrollan en uno de entre un número determinado de lenguajes de programación. Un jurado online usualmente ofrece además otros servicios de consulta y suele servir como herramienta para la realización de competencias de programación.

**Nick:** Apodo o identificador que se utiliza en entornos cibernéticos por los usuarios. La diferencia principal entre un nick y un login es que el primero es una etiqueta elegida por el usuario para representarse individualmente, mientras el segundo es un contrato entre el usuario y el sistema y como tal puede no ser modificable.

**Noticia:** Noticia o aviso de algún tema relacionado con el jurado, comúnmente competencias, período de mantenimiento o cambio de versiones o compiladores.

**Pdf:** Documento en formato PDF que se le envía al usuario conteniendo cierta información de interés, por ejemplo el listado de problemas, un problema determinado o la ayuda del sistema.

**Problema:** Enunciado preciso de una situación que presenta un problema a solucionar mediante la aplicación de técnicas algorítmicas, lógica, geometría o cualquier otra rama matemática. Los problemas generalmente se dividen en secciones muy específicas, siendo la división más común: descripción, especificación de entrada, especificación de salida, ejemplo de entrada, ejemplo de salida y fuente del problema. Tradicionalmente el enunciado del problema debe proveer cualquier conocimiento que caiga fuera de las áreas algorítmicas o matemáticas y que sea necesario para resolverlo. Se sigue el razonamiento de que lo que se mide no es el nivel de conocimiento o cultura general del usuario, sino su conocimiento y habilidad en la rama de programación de competencias.

**Programación de competencia:** Se refiere la programación centrada en el análisis y la implementación de programas con el objetivo de medir conocimientos teóricos en aspectos particulares de algoritmia, matemática o lógica. Esta rama de programación presenta características diferentes de la programación empresarial o educativa, en que no presta atención a los principios de orientación a objetos, chequeo de errores, patrones de diseño, etc.; concentrándose exclusivamente en aspectos de eficiencia y correctitud de los programas. Una aplicación para competencias suele ser de código ofuscado, sin diseño claro y con poco mantenimiento, con poca o ninguna extensibilidad ni modularidad debido a que el interés principal

está puesto en la eficiencia en el uso de memoria y tiempo y en la implementación correcta del algoritmo correspondiente. Opcionalmente algunas personas tienden a desarrollar sus propios estilos de programación, buscando elegancia en sus soluciones lo cual puede contrarrestar las propiedades negativas anteriormente expuestas.

**Ranking:** Es una lista de los usuarios del jurado, ordenados atendiendo a diferentes criterios.

**Shoutbox:** es un cuadro de diálogo donde los visitantes de un sitio web pueden dejar sus mensajes directamente en el sitio. Es similar a un chat, pero los mensajes también se almacenan como en un libro de visitas.

**Solución:** Sumisión exitosa.

**Sumisión:** Intento de solución de un problema. Una sumisión generalmente se identifica con un número secuencial, y corresponde a un intento realizado por un usuario para resolver un problema determinado. Constituye la unidad básica de interacción del usuario con la aplicación, y puede tener asociados distintos estados en dependencia de si el intento es exitoso o no, y si no, del error que provocó el fracaso.

**Usuario:** Individuo que utiliza el sistema mediante el envío y recepción de correos electrónicos. El público objetivo del mismo se espera esté compuesto principalmente por estudiantes universitarios de Informática.

**Volumen de problemas:** Conjunto de problemas que se encuentran en un jurado online o en un evento competitivo. Los volúmenes pueden ser creados siguiendo distintos criterios, como complejidad, asunto, tamaño o simplemente ordenados cronológicamente, dependiendo el criterio del jurado o evento donde se encuentre.

**WA:** Código de respuesta incorrecta. Estado de una sumisión desacertada.

## Anexos

### Anexo 1: Diseño de caso de prueba CU Consultar ranking ordenado

#### Descripción general

El caso de uso se inicia cuando el usuario envía un mensaje de correo al sistema y en el asunto especifica el rango del ranking que desea consultar y el criterio de ordenamiento. El sistema valida que la petición del usuario esté en el formato correcto. Construye el ranking ordenado de acuerdo al criterio y rango especificados por el usuario y le envía un mensaje de correo con dicho ranking. Finaliza así el caso de uso.

#### Condiciones de ejecución

El usuario debe haber accedido al correo electrónico.

#### Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC 1: Consultar ranking ordenado por puntos.	EC 1.1: Consultar ranking ordenado por puntos exitosamente.	Luego de que el usuario envía un mensaje de correo especificando en el asunto el siguiente formato: mj:ranking -n <rango> -c ptos Ej. (mj:ranking -n 1-500 -c ptos) El sistema valida el formato del asunto del correo. Construye el ranking ordenado por puntos (ptos) dentro del rango especificado. Envía mensaje de correo y en el cuerpo del mismo muestra el ranking de usuarios ordenados por puntos dentro del rango especificado.	Recibir correo de usuario Validar asunto Construir ranking Enviar correo a usuario

	EC 1.2: Envía mensaje de correo que no se corresponde con el formato definido.	Envía mensaje de correo con la ayuda.	<p>Recibir correo de usuario</p> <p>Validar asunto (asunto inválido)</p> <p>Construir ayuda</p> <p>Enviar correo a usuario</p>
SC 2: Consultar ranking ordenado por soluciones aceptadas.	EC 2.1: Consultar ranking ordenado por soluciones aceptadas exitosamente.	<p>Luego de que el usuario envía un mensaje de correo especificando en el asunto el siguiente formato: mj:ranking -n &lt;rango&gt; -c ac</p> <p>Ej. (mj:ranking -n 1-500 -c ac)</p> <p>El sistema valida el formato del asunto del correo. Construye el ranking ordenado por soluciones aceptadas (ac) dentro del rango especificado. Envía mensaje de correo y en el cuerpo del mismo muestra el ranking de usuarios ordenados por soluciones aceptadas dentro del rango especificado.</p>	<p>Recibir correo de usuario</p> <p>Validar asunto</p> <p>Construir ranking</p> <p>Enviar correo a usuario</p>
	EC 2.2: Envía mensaje de correo que no se corresponde con el formato definido.	Envía mensaje de correo con la ayuda.	<p>Recibir correo de usuario</p> <p>Validar asunto (asunto inválido)</p> <p>Construir ayuda</p> <p>Enviar correo a usuario</p>

SC 3: Consultar ranking ordenado por cantidad de envíos.	EC 3.1: Consultar ranking ordenado por cantidad de envíos exitosamente.	Luego de que el usuario envía un mensaje de correo especificando en el asunto el siguiente formato: mj:ranking -n <rango> -c envios Ej. (mj:ranking -n 1-500 -c envios) El sistema valida el formato del asunto del correo. Construye el ranking ordenado por cantidad de envíos (envios) dentro del rango especificado. Envía mensaje de correo y en el cuerpo del mismo muestra el ranking de usuarios ordenados por cantidad de envíos dentro del rango especificado.	Recibir correo de usuario Validar asunto Construir ranking Enviar correo a usuario
	EC 3.2: Envía mensaje de correo que no se corresponde con el formato definido.	Envía mensaje de correo con la ayuda.	Recibir correo de usuario Validar asunto (asunto inválido) Construir ayuda Enviar correo a usuario
SC 4: Consultar ranking ordenado por % de soluciones aceptadas.	EC 4.1: Consultar ranking ordenado por % de soluciones aceptadas exitosamente.	Luego de que el usuario envía un mensaje de correo especificando en el asunto el siguiente formato: mj:ranking -n <rango> -c %ac Ej. (mj:ranking -n 1-500 -c %ac) El sistema valida el formato del asunto del correo. Construye el ranking ordenado por % de soluciones aceptadas (%ac) dentro del rango especificado. Envía mensaje de correo y en el cuerpo del mismo muestra el ranking de usuarios ordenados por % de soluciones aceptadas dentro del rango especificado.	Recibir correo de usuario Validar asunto Construir ranking Enviar correo a usuario
	EC 4.2: Envía	Envía mensaje de correo con la ayuda.	Recibir correo de

	mensaje de correo que no se corresponde con el formato definido.		usuario Validar asunto (asunto inválido) Construir ayuda Enviar correo a usuario
--	--	--	---

### Descripción de la variable

No	Nombre de campo	de Clasificación	Valor Nulo	Descripción
1	Asunto	Texto	No	Está estructurado de la siguiente forma: <b>mj:ranking &lt;variables&gt;</b> (se puede poner el <b>mj:</b> separado por espacios de la palabra <b>ranking</b> ). Las variables están compuestas por los pares <b>&lt;criterio&gt; &lt;valor&gt;</b> . Existen dos variables: criterio de ordenamiento y rango.
2	Criterio de ordenamiento	Texto	Si	Está estructurado de la siguiente forma: <b>-c &lt;valor&gt;</b> donde valor puede ser: <b>ptos</b> (puntos), <b>ac</b> (soluciones aceptadas), <b>envios</b> (cantidad de envíos), <b>%ac</b> (% de soluciones aceptadas).
3	Rango	Texto	Si	Está estructurado de la siguiente forma: <b>-n &lt;valor&gt;</b> donde valor es un rango de números con la forma #-#. Este rango no debe exceder los 500 y su menor valor es 1. Ej. (1-500). En caso de que no se especifique esta variable se asume que es -n 1-500.

### Matriz de Datos

#### SC #1: Consultar el ranking ordenado por puntos

