

Universidad de las Ciencias Informáticas

Facultad 1



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

Título:

“Sistema de recomendación de problemas para el
Juez en Línea Caribeño”

Autores:

Junet Doria Martínez
José Carlos González Fernández

Tutores:

MSc. Delly Lien González Hernández
Ing. Hubert Viltres Sala
Ing. Dovier Antonio Ripoll Méndez

La Habana
Junio, 2014

Declaración de autoría

Declaro ser autor de la presente tesis y ofrezco a la Universidad de las Ciencias Informáticas los derechos primordiales de la misma. Para que así conste firmo la presente a los ____ días del mes de _____ del año_____.

Firma del Autor

Junet Doria Martínez

Firma del Autor

José Carlos González Fernández

Firma del Tutor

MSc. Delly Lien González Hernández

Firma del Tutor

Ing. Hubert Viltres Sala

Firma del Tutor

Ing. Dovie Antonio Ripoll Méndez

Datos de contactos

Autores:

Junet Doria Martínez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: jdoria@estudiantes.uci.cu

José Carlos González Fernández

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: jcfernandez@estudiantes.uci.cu

Tutores:

MSc. Delly Lien González Hernández

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: delly@uci.cu

Ing. Hubert Viltres Sala

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: hviltres@uci.cu

Ing. Dovier Antonio Ripoll Méndez

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo electrónico: daripoll@uci.cu

Dedicatoria

De Junet

*A mis padres, por siempre apoyarme
durante todos mis estudios.*

*A mi abuelo Ciprian, que aunque no me acompaña
ahora, se que siempre me hubiera apoyado.*

*A Amhed que de una forma u otra siempre me apoyó
en los momentos difíciles.*

De José Carlos

A mi abuelo Antonio por ser mi ejemplo como persona.

Agradecimientos

De Junet

A mi mamá por siempre apoyarme sin esperar nada a cambio.

A mi papá por guiarme en mis estudios.

A mi novio Amhed por estar a mi lado sobre todo en los momentos más difíciles.

A mi tía Ivette por prestarme ayuda siempre que pudo.

A Manuel por ayudarme siempre que lo necesité.

A mis compañeras de apartamento y que alguna vez fueron de aula Estela y Laura.

A mis dos amigas del alma Ariagna y Mayvis.

A José Carlos por ser un compañero de tesis ejemplar.

A mis tutores por todo el apoyo brindado.

A Yurelkys que una vez fue mi tutora, y siempre me ayudó.

A todos aquellos que de una forma u otra me brindaron su ayuda.

De José Carlos

A todos los que de una forma u otra han confiado en mí todos estos años y en especial a mi familia por todo su apoyo.

Resumen

El presente trabajo propone un sistema de recomendación de problemas para el Juez en Línea Caribeño. Se estudiaron las características y funcionalidades de los principales jueces en línea a nivel mundial, los sistemas de recomendación y su relación. La solución se concibió por la necesidad de facilitarle al usuario la selección de los problemas más relacionados con sus competencias y afinidades. Para la propuesta se escogió *Spring 3.0* como *framework* de desarrollo y *SXP* como metodología de software para guiar el proceso de desarrollo. Se emplea como servidor web *Apache-Tomcat 7.0.40*, como Sistema Gestor de Base de Datos *PostgreSQL 9.1*, como IDE de desarrollo *Netbeans 7.3*, para el modelado del proceso, *Visual Paradigm 8.0*, *HTML 5*, *CSS 3* y *Java* como lenguajes de programación y como herramienta para las pruebas de carga y estrés *Apache JMeter 2.5*. La solución consiste en un sistema de recomendación híbrido en cascada, que combina la técnica de un sistema de recomendación basado en filtrado colaborativo con filtrado basado en contenido. Como resultado, el sistema recomienda problemas de acuerdo a los gustos y preferencias del usuario. Se aplicaron métricas para validar la calidad y exactitud de la recomendación. Las pruebas de carga y estrés, integración y aceptación permitieron detectar errores y corregirlos oportunamente, y evidenciaron que el sistema de recomendación facilita el acceso a los problemas disponibles en el Juez en Línea Caribeño y posibilita que los usuarios aprovechen mejor sus competencias y aumenten su preparación.

Palabras clave: algoritmos, juez en línea, problema y sistema de recomendación.

Índice

Introducción	1
Capítulo 1: Antecedentes de los jueces en línea y los sistemas de recomendación, principales características y funcionalidades, tecnologías utilizadas en su desarrollo	5
1.1 Análisis de los principales jueces en línea	5
1.2 Sistemas de recomendación	9
1.2.1 Limitaciones de los algoritmos en sistemas de recomendación	12
1.3 Tecnologías de desarrollo de software	14
1.3.1 Lenguaje de programación	14
1.3.2 <i>Framework</i>	15
1.3.3 Sistema Gestor de Bases de Datos.....	15
1.3.4 Servidor web	16
1.3.5 Entorno Integrado de Desarrollo.....	16
1.3.6 Herramienta de modelado	17
1.3.7 Metodología de desarrollo de software.....	17
1.4 Métodos de validación.....	18
1.4.1 Métricas de validación	18
1.4.2 Pruebas de sistema	19
1.4.3 Pruebas de integración.....	20
1.4.4 Pruebas de validación	20
1.5 Conclusiones parciales	21
Capítulo 2: Descripción del sistema de recomendación para el Juez en Línea Caribeño	22
2.1 Características del COJ	22
2.2 Sistema de recomendación híbrido	24
2.3 Modelo de negocio	25
2.4 Perfil algorítmico propuesto para el sistema de recomendación.....	26
2.5 Requerimientos	29
2.5.1 Requisitos funcionales	29
2.5.2 Requisitos no funcionales.....	30
2.6 Lista de Reserva del Producto	31
2.7 Historias de usuario.....	34
2.8 Tareas de Ingeniería	40
2.9 Diagrama de clases del diseño	41
2.10 Modelo de datos.....	44

2.11 Propuesta de arquitectura	45
2.12 Patrones de diseño	45
2.13 Conclusiones parciales	47
Capítulo 3: Implementación y validación del sistema de recomendación para el Juez en Línea Caribeño	48
3.1 Plan de release	48
3.2 Diagrama de componentes	48
3.3 Diagrama de paquetes	50
3.4 Diagrama de despliegue	51
3.4.1 Descripción de los componentes	51
3.5 Pruebas	51
3.5.1 Evaluación de la recomendación	51
3.5.2 Métricas de validación	51
3.5.3 Pruebas de sistema	53
3.5.4 Pruebas de integración	56
3.5.5 Pruebas de aceptación	57
3.6 Conclusiones parciales	58
Conclusiones	59
Bibliografía	61
Anexos	61

Índice de figuras

Figura 1. Algoritmos de filtrado colaborativo.....	11
Figura 2. Precisión.....	18
Figura 3. Conjunto de elementos valorados por el usuario	19
Figura 4. MAE para un usuario.....	19
Figura 5. Promedio del MAE.....	19
Figura 6. Propuesta de solución.....	24
Figura 7. Modelo de negocio 1	25
Figura 8. Modelo de negocio 2	26
Figura 9. Diagrama de clases de diseño: Mostrar recomendación a usuarios no autenticados.....	42
Figura 10. Diagrama de clases de diseño: Mostrar recomendación a usuarios autenticados.....	42
Figura 11. Diagrama de clases de diseño: Mostrar recomendación de problemas similares a usuarios autenticados.....	43
Figura 12. Diagrama de clases de diseño: Mostrar recomendación de problemas similares a usuarios no autenticados.....	43
Figura 13. Diagrama de clases de diseño: Valorar problema.....	44
Figura 14. Modelo de datos	44
Figura 15. Diagrama de componentes.....	49
Figura 16. Diagrama de paquetes	50
Figura 17. Diagrama de despliegue.....	51
Figura 18. Prueba carga y estrés a usuarios autenticados.....	54
Figura 19. Prueba carga y estrés a usuarios autenticados.....	55
Figura 20. Reporte resumen 50 usuarios autenticados	70
Figura 21. Reporte resumen 100 usuarios autenticados	70
Figura 22. Reporte resumen 150 usuarios autenticados	70
Figura 23. Reporte resumen 50 usuarios no autenticados	70
Figura 24. Reporte resumen 100 usuarios no autenticados	70
Figura 25. Reporte resumen 150 usuarios no autenticados	70

Índice de tablas

Tabla 1. Lista de Reserva del Producto	31
Tabla 2. Historia de Usuario No.1	35
Tabla 3. Historia de Usuario No.2	36
Tabla 4. Historia de Usuario No.3	37
Tabla 5. Historia de Usuario No.4	38
Tabla 6. Historia de Usuario No.5	39
Tabla 7. Tarea de Ingeniería No.1	40
Tabla 8. Tarea de Ingeniería No.2	40
Tabla 9. Tarea de Ingeniería No.3	40
Tabla 10. Tarea de Ingeniería No.4	41
Tabla 11. Tarea de Ingeniería No.5	41
Tabla 12. Plan de <i>release</i>	48
Tabla 13. Valores de Precisión	52
Tabla 14. Valores del MAE	53
Tabla 15. Caso de prueba de integración No.1	56
Tabla 16. Caso de prueba de integración No.2	56
Tabla 17. Caso de prueba de aceptación No.1	57
Tabla 18. Caso de prueba de aceptación No.2	57
Tabla 19. Caso de prueba de aceptación No.3	57
Tabla 20. Caso de prueba de aceptación No.4	58
Tabla 21. Caso de prueba de aceptación No.5	58
Tabla 22. Muestra escogida para la evaluación de la recomendación	68
Tabla 23. Recomendaciones obtenidas.....	69

Introducción

El Concurso Internacional Universitario ACM de Programación (ACM-ICPC por sus siglas en inglés), efectuado anualmente con la participación de varios equipos en representación de diferentes universidades del mundo, constituye uno de los eventos más importantes y prestigiosos de programación de computadoras donde se fomenta el trabajo en equipo, la resolución de problemas y el desarrollo rápido de software.

El ACM-ICPC tiene sus raíces en un concurso realizado en el Texas A & M¹ en 1970. Cada año se realiza en distintas sedes por todo el mundo con la participación de equipos conformados por tres estudiantes guiados por un profesor-entrenador. Estos equipos pasan por tres etapas de competición que van desde los concursos locales (efectuados en cada una de las universidades que participan en el evento), hasta los nacionales y regionales. De esta forma los ganadores de cada regional avanzan a la Final Mundial cada año (Silveira-Romero y Hernández-Legrá, 2012) (Vaillant y Labaut, 2013).

Cuba se ha ido insertando paulatinamente en el movimiento ACM-ICPC, posibilitando que se cree en la región una final caribeña. A través del Movimiento de Programación Competitiva “Tomás López Jiménez” (MPC-TLJ) en la Universidad de las Ciencias Informáticas (UCI) se ha fortalecido la participación cada vez mayor de concursantes en los eventos locales, nacionales y regionales del ACM-ICPC. En la UCI se realizan diferentes actividades enfocadas a incrementar la preparación de los participantes cubanos en todos los niveles del concurso.

En sus inicios este concurso contaba con pocos participantes, de manera que se realizaba manualmente la gestión de la información asociada al mismo, el proceso de evaluación de las soluciones y la confección de la tabla de posiciones. Con el transcurso de los años, el crecimiento en el número de participantes complejizó el proceso de evaluación según *ACM International Collegiate Programming Contest*², lo cual posibilitó el surgimiento de los evaluadores automáticos³, que fueron utilizados por largo tiempo, pero con el desarrollo de la web surgió lo que se conoce en la actualidad como jueces en línea.

“El 5 de junio de 2010 se publica en Internet la versión 1 del Juez en Línea Caribeño (COJ por sus siglas en inglés) con el objetivo fundamental de brindar más autonomía al movimiento ACM-ICPC en la región del Caribe, proporcionando a sus miembros (concurantes, entrenadores, directores y administradores) un espacio destinado a la auto-preparación y el intercambio de conocimientos” (Lobaina y Roque, 2012).

¹ Originalmente abreviatura de "Agricultura y Mecánica". Se conserva este nombre como un vínculo con el pasado de la universidad.

² <http://icpc.baylor.edu/>

³ Sistema cuyo propósito general es el procesamiento y evaluación de las soluciones propuestas por los concursantes.

En la actualidad el COJ es una de las principales herramientas que se utilizan para facilitar la preparación de los concursantes, mediante un amplio conjunto de problemas y concursos. Constituye un referente para potenciar las habilidades en cada usuario, aun cuando el mismo no explota todas sus funcionalidades.

Según lo antes mencionado se considera que es muy frecuente que los usuarios al interactuar con el juez en línea, no hagan una selección correcta de los ejercicios publicados u otros contenidos del sistema debido a la gran disponibilidad de información y son incapaces de elegir los ejercicios que mejor se adaptan a sus preferencias, competencias y habilidades como programadores. El COJ propone mucho contenido (problemas categorizados, códigos fuente y competencias) que no se brinda de forma adecuada a los usuarios según sus preferencias, limitando con ello la experiencia del usuario en el sistema y el valor agregado de éste respecto a sistemas similares. Existen por tanto insuficiencias para potenciar las habilidades y competencias individuales de los usuarios, pues no se identifican sus necesidades y preferencias relacionadas con los problemas disponibles en el COJ. Además de la variedad y profundidad de conocimientos previos, los usuarios pueden tener diferentes estilos de aprendizaje, por lo que se requiere evaluar y aplicar técnicas que permitan seleccionar los problemas que más se ajustan a su perfil y esta es otra insuficiencia actual del sistema.

Considerando la situación problemática descrita se presenta el siguiente **problema de la investigación**: ¿Cómo facilitar el acceso a los problemas disponibles en el Juez en Línea Caribeño para que los usuarios aprovechen mejor sus competencias y aumenten su preparación?

Con el fin de dar respuesta a este problema, se selecciona como **objeto de estudio** el análisis y clasificación de los problemas en los jueces en línea.

El **objetivo general** de la investigación es desarrollar un sistema de recomendación para el Juez en Línea Caribeño que permita facilitar el acceso a los problemas disponibles para que los usuarios aprovechen mejor sus competencias y aumenten su preparación.

Para darle solución al problema planteado se han definido los siguientes **objetivos específicos**:

1. Caracterizar los principales jueces en línea y sus funcionalidades.
2. Caracterizar los fundamentos teóricos y tendencias en el desarrollo de sistemas de recomendación.
3. Definir las tecnologías, las herramientas y la metodología para implementar/developar el sistema de recomendación para el COJ.
4. Diseñar las funcionalidades del sistema de recomendación para el COJ.
5. Implementar las funcionalidades del sistema de recomendación para el COJ.
6. Validar las funcionalidades del sistema de recomendación para el COJ.

Se determina como **campo de acción** el análisis y clasificación de los problemas en el Juez en Línea Caribeño.

La **idea a defender** en el presente Trabajo de Diploma es que con el desarrollo de un sistema de recomendación para el Juez en Línea Caribeño se facilitará el acceso a los problemas disponibles en él para que los usuarios aprovechen mejor sus competencias y aumenten su preparación.

Para darle cumplimiento a los objetivos específicos se plantean las siguientes **tareas de la investigación**:

- Realización de un estudio sobre las tendencias en el desarrollo de sistemas de recomendación según las características de los jueces en línea.
- Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
- Selección de la metodología de desarrollo.
- Definición de los requisitos funcionales y no funcionales de la propuesta de solución.
- Implementación de la propuesta de solución.
- Documentación de las pruebas de carga y estrés, de aceptación y de integración y las métricas de evaluación del sistema de recomendación.

Para guiar el desarrollo de la investigación se usaron los siguientes **métodos científicos**:

Métodos Teóricos:

- Histórico-Lógico: se utilizó para consultar la bibliografía referente al ACM-ICPC en sus diferentes niveles, su evolución y comportamiento. Permitió un mayor conocimiento sobre el ACM-ICPC así como estudiar la trayectoria histórica y la lógica del desarrollo de jueces en línea.
- Analítico - Sintético: posibilitó analizar la bibliografía y realizar una síntesis de las fuentes para extraer los elementos útiles y de esta forma poder dar solución al problema.

Métodos Empíricos:

- Análisis documental: permitió buscar, seleccionar y consultar bibliografía sobre los temas de interés para el desarrollo de la propuesta de solución.
- Encuesta: se realizó una encuesta a los usuarios del COJ, arrojando la información necesaria para conocer cuáles son los contenidos que se pueden tomar como referencia para las recomendaciones. Para así poder mejorar el sistema de recomendación ya existente.

La presente investigación se justifica en la necesidad de facilitar el acceso a los problemas disponibles en el Juez en Línea Caribeño.

El presente trabajo de diploma está compuesto por un resumen, una introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos. A continuación se describen los principales aspectos abordados en cada uno de los capítulos:

- **Capítulo 1: Antecedentes de los jueces en líneas y los sistemas de recomendación, principales características y funcionalidades, tecnologías utilizadas en su desarrollo.** En este capítulo se abordan los antecedentes de los jueces en línea, sus características y

funcionalidades así como los que son reconocidos tanto a nivel mundial como nacional. Se analizan los sistemas de recomendación, características principales, sus clasificaciones y aplicación. Además se definen las tecnologías, herramientas y metodología que se utilizan, así como las métricas de validación y pruebas que se le aplicaron al sistema.

- **Capítulo 2: Descripción del sistema de recomendación para el Juez en Línea Caribeño.** En el presente capítulo se describe la propuesta de solución basada en un sistema de recomendación que permita facilitar la interacción de sus usuarios con el COJ. El sistema permite recomendar problemas ajustados a las preferencias del usuario, basándose en el historial y el perfil de estos. Se describen las fuentes de datos y la técnica a utilizar para la selección del algoritmo para su implementación, se muestran segmentos de código que representan los algoritmos principales y se generan artefactos según la metodología utilizada.
- **Capítulo 3: Implementación y validación del sistema de recomendación para el Juez en Línea Caribeño.** En este capítulo se realiza el plan de *release* del sistema y una vez concluida la fase de implementación se aplican las métricas de validación a la solución y se le realizan las pruebas de aceptación e integración las cuales validarán si el sistema cumple con los requisitos planteados por el cliente.

Capítulo 1: Antecedentes de los jueces en línea y los sistemas de recomendación, principales características y funcionalidades, tecnologías utilizadas en su desarrollo

En este capítulo se realizará un estudio sobre los antecedentes de los jueces en línea, los que son reconocidos tanto a nivel mundial como nacional, y sus características y funcionalidades. Se analizan los sistemas de recomendación más utilizados, características principales, sus clasificaciones y aplicación, además de los algoritmos y métodos que utilizan. Después de analizados los resultados se tratarán las principales herramientas, tecnologías y metodología que se utilizan en su desarrollo y validación.

1.1 Análisis de los principales jueces en línea

En la realización de un concurso ACM-ICPC los jueces principales proponen un conjunto de problemas⁴, cada uno de los cuales está compuesto por uno o varios conjuntos de pruebas⁵. Los equipos participantes pueden solicitar la evaluación de los problemas en cualquier momento. Esta posibilidad hace que se acumule gran cantidad de envíos por cada competencia en correspondencia con la cantidad de equipos y la dificultad de cada problema propuesto. De esta forma la evaluación se convierte en un proceso que resulta ser un poco complicado y engorroso al realizarlo de manera manual.

La mayoría de las veces este proceso de evaluación es realizado por evaluadores automáticos, los cuales son sistemas informáticos cuyo propósito es el procesamiento eficiente de las solicitudes de evaluación. Los evaluadores automáticos con el surgimiento y la popularidad alcanzada por la web fueron utilizados por largo tiempo hasta el surgimiento de lo que se conoce en la actualidad como jueces en línea.

Según plantean Acosta y otros(2010) los jueces en línea son aplicaciones web disponibles todo el tiempo, que automatizan la evaluación de las respuestas a problemas de programación competitiva. Surgieron para apoyar la preparación de estos concursos y potenciar el desarrollo de habilidades en la resolución de problemas de matemática, lógica y algoritmia.

En la actualidad se han desarrollado disímiles jueces en línea, los cuales posibilitan que los problemas sean evaluados de la forma más eficiente posible. En los estudios realizados se identificaron 74 jueces en línea como los más destacados a nivel mundial. Según plantea Junco (2012) entre los jueces en línea de programación más conocidos e importantes a nivel global pueden mencionarse el Juez en Línea de la Universidad de Valladolid (UVA)⁶, el Juez en Línea de la Universidad de Tecnología de Gdansk en Polonia (SPOJ)⁷, y el Juez en Línea de la Universidad de Pekín (POJ)⁸.

⁴ *Problem Set*

⁵ *Test cases: Ficheros* que contienen la entrada y salida original de cada problema, las principales restricciones.

⁶ <http://uva.onlinejudge.org/>

⁷ <http://www.spoj.pl/>

⁸ <http://poj.org/>

“En materia del entrenamiento de concursantes ACM-ICPC así como en el aporte pedagógico el **UVA** es, a nivel mundial, el que goza de más prestigio en la comunidad universitaria internacional, quizás por ser uno de los más antiguos y el mejor surtido en ejercicios” (Junco, 2012). El mismo ha alcanzado recientemente los diez millones de envíos juzgados, lo que significa que el portal de Internet gestionado desde la **UVA** ha ofrecido a los usuarios un millón de problemas de algoritmia y de programación de informática, que posteriormente han sido revisados por el propio sistema. Cuenta con usuarios de 190 países de todo el mundo, principalmente de sudamérica y del sudeste asiático.

UVA tiene una de las mejores colecciones de problemas de la red, además de que contiene una gran cantidad de problemas cubriendo diferentes categorías. Periódicamente organiza concursos *online* para todo el que quiera participar y está acompañado de un juez de reconocida estabilidad.

Como rasgos distintivos se pueden resaltar:

- Módulo de estadísticas, el cual muestra información detallada respecto a los problemas, usuarios y concursos.
- Integración a la plataforma EduJudge⁹, convirtiéndolo en uno de los pocos jueces en línea que poseen una integración con el proceso de enseñanza-aprendizaje.
- Implementado sobre plataforma UNIX.
- Tabla de posiciones, dedicada al desempeño de los usuarios en competencias.

Por su parte, **SPOJ** es un juez en línea desarrollado por el proyecto *Sphere Research Labs*, grupo dedicado al desarrollo de tecnologías en línea para proveer herramientas de evaluación automática de habilidades, según lo considera *Sphere Research Business Solutions*(2013). Tiene alrededor de 13000 problemas de gran calidad y dificultad, indicado para grandes retos. Presenta 227172 usuarios registrados, 2 000 retos de programación y las soluciones a los problemas se pueden presentar en más de 40 lenguajes de programación. Posee varias características novedosas que lo diferencian de la mayoría de los jueces en línea, tales como:

- Permite la diferenciación del archivo de problemas en cuanto a dificultad y propósito.
- Brinda soporte a más de 40 lenguajes de programación, incluyendo C, C++, Python, C#, Go, Haskell, Ocaml, F#.
- Posee tiempos de evaluación muy rápidos, a pesar de las pocas prestaciones del servidor sobre el cual está desplegado.
- Brinda un abundante *kit* de herramientas asociadas al sistema.
- Facilita un espacio de intercambio para sus usuarios mediante un foro de discusión.

⁹ Proyecto perteneciente a la Universidad de Valladolid que se dedica al desarrollo de herramientas con carácter pedagógico utilizando el Juez en Línea UVA.

El **POJ** es uno de los jueces en línea más populares entre la comunidad universitaria china, siendo uno de los más antiguos de esa región. Tiene características similares a otros jueces en línea de otras universidades chinas y de otras universidades del mundo, porque por algún tiempo era posible descargar una versión disminuida y restrictiva de la aplicación. Acepta soluciones escritas en C, C++, Pascal, Java y Fortran.

La característica más distintiva del **POJ** es el módulo de estadísticas, ya que recopila y muestra los datos de uso del sistema desde su surgimiento en el año 2003 hasta la fecha. Por otra parte, ofrece un servicio de mensajería interna para los usuarios registrados, funcionalidad que se considera muy útil en este tipo de sistema. Este sistema marcó los primeros pasos del desarrollo y despliegue de jueces en línea en la UCI mediante la versión descargable libremente que proporcionaron en el 2006.

En tal sentido, el **COJ**, desarrollado en Cuba, en la UCI, pero de alcance regional e internacional, tiene como característica más sobresaliente la rapidez en los tiempos de respuesta en la mayoría de las interfaces, no así en la vista del estado de los envíos. En estos momentos tiene 14354 usuarios registrados, 726 instituciones representadas, 144 países representados, 1892 problemas/ejercicios publicados, 553388 programas enviados por los usuarios, 324 competencias realizadas. Desarrollan competencias en varios lenguajes de programación como: Bash, C, C#, C++, Java, Pascal, Perl, PHP, Python, Ruby, Text (COJ, 2014).

El COJ posee servicios como:

- Mensajería interna para los usuarios registrados.
- Interfaz por correo electrónico para enviar soluciones y consultar el estado (COJ, 2014).

En Cuba se cuenta con la presencia de otros jueces en línea como el sistema **UCOCoach**, que surge como iniciativa de estudiantes y profesores de la Universidad de Camagüey, con el objetivo de apoyar a los concursantes de dicha institución en la preparación para competencias de programación. Posee las características comunes de todo juez en línea, además de una interfaz amigable y sugerente. La funcionalidad más relevante del sistema es la ayuda que ofrece automáticamente a los usuarios, cuando estos fallan varias veces un mismo problema. El sistema está actualmente en desarrollo y no se encuentra disponible en Internet (Lobaina y Roque, 2012).

En la UCI existe la Cátedra de Programación Avanzada (**CPAV**), que brinda servicio de competencias y la posibilidad de resolver los ejercicios disponibles. La misma dispone de un jurado que permite la realización de las Copas UCI, Copa Pascal y otros eventos competitivos de programación (Viltres y Matos, 2010).

A partir de este análisis, los autores reconocen que los jueces en línea poseen como ventaja que hacen propicio que se desarrolle el aprendizaje colaborativo que es “un sistema de interacciones cuidadosamente diseñado que organiza e induce la influencia recíproca entre los integrantes de un equipo. Se desarrolla a través de un proceso gradual en el que cada miembro se siente mutuamente

comprometido con el aprendizaje de los demás generando una interdependencia positiva que no implique competencia” (Herrera-Viedma y otros, 2012).

En cuanto a desventajas, el **COJ** en particular, tiene como inconveniencia, que presentan un gran cúmulo de problemas y otras informaciones que no se muestran adecuadamente y no se ajustan a las necesidades de los usuarios, por lo que se les torna complicado escoger los problemas más adecuados. Para darle solución ya se han realizado herramientas que solucionan esos problemas, tal es el caso de las herramientas recomendadoras que fueron desarrolladas para dos de los principales jueces en líneas mencionados anteriormente el ***UVA Online Judge*** y el ***SPOJ***, para guiar a los usuarios a la hora de seleccionar los problemas a resolver (Toledo, 2010).

Las herramientas recomendadoras desarrolladas son el ***ACM Problem Grading***, de Sebastian Urbaniak, el ***Next2Solve***, de Igor Naverniuk, ambas implementadas en apoyo al jurado en línea de la Universidad de Valladolid; y el ***Problem classification***, en el *Sphere Online Judge*.

La herramienta ***ACM Problem Grading***, básicamente se limita a proporcionar un listado de problemas a resolver, el cual puede ser ordenado por diferentes criterios, entre los que se pueden mencionar la cantidad de soluciones aceptadas para cada ejercicio, el por ciento de soluciones correctas con respecto al total de soluciones y una puntuación de simplicidad que se calcula tomando como base los dos criterios anteriores.

El principal inconveniente que presenta la herramienta es que sus recomendaciones no son enfocadas a cada usuario en específico, solo supone que los problemas más fáciles para resolver son aquellos que tengan más soluciones correctas enviadas. Otro elemento negativo que presenta es que no es ajustable a nuevas versiones; según lo considera Toledo (2010).

Next2Solve, creado por Igor Naverniuk, estudiante de doctorado de la Universidad de Toronto, se autodefine como un servicio basado en el *ACM Problem Grading* de Urbaniak. La diferencia radical entre uno y otro viene dada en que mientras *ACM Problem Grading* muestra de manera indistinta el mismo listado para todos los usuarios, *Next2Solve* solicita la entrada de un identificador del usuario y basado en este muestra una sublista de la lista generada por la aplicación de Urbaniak, en la que sólo aparecen los problemas a resolver por el usuario entrado, excluyéndose aquellos que él ya ha resuelto (Toledo, 2010).

Problem classification es una herramienta de recomendación, asociada al jurado en línea *SPOJ*, que utiliza un enfoque distinto al presentado en las dos aplicaciones anteriores. En vez de concentrarse en el perfil de los usuarios, esta se centra en los ejercicios, definiendo para cada uno de ellos una serie de datos o etiquetas que luego son utilizadas para agruparlos por categorías. Utilizando estas categorías y a través de opciones de filtrado, los usuarios pueden obtener propuestas de problemas a resolver, basándose en criterios de búsqueda, que coinciden generalmente con los valores de las etiquetas. Presenta la misma inconveniencia que la herramienta ***ACM Problem Grading*** ya que no recomienda

basado en los usuarios, porque permite hacer uso de todas las funcionalidades sin necesidad de autenticarse (Toledo, 2010).

Gran cantidad de jueces en línea no presentan una herramienta de recomendación que facilite que el usuario sea capaz de manejar y comprender mejor los contenidos que se muestran. De aquí se hace necesario el análisis más profundo de los sistemas de recomendación para poder obtener una mejor idea de su funcionamiento, como solución actual a los problemas o desventajas detectados en los jueces en línea analizados, incluido el COJ.

1.2 Sistemas de recomendación

Una solución del área de la informática a la necesidad de información de los usuarios, ha sido el surgimiento de los denominados sistemas de recomendación (en inglés *Recommender Systems*). Resnick y Varian (1997); Ricci y otros(2010) coinciden en que un sistema de recomendación es una herramienta de software que utilizando preferencias, sugerencias y valoraciones emitidas por los usuarios, permite guiarlos en su proceso de búsqueda de información relevante. Varios autores (Resnick y Varian, 1997; Burke, 2002; Resnik y otros, 2010; Ricci y otros, 2010) los definen en la literatura disponible, y a los efectos de esta investigación se considera que un sistema de recomendación es una herramienta o software que utilizando preferencias, sugerencias y valoraciones emitidas por los usuarios, permite guiarlos en su proceso de búsqueda de información, previa selección de la información más relevante, mostrando una cantidad que sea manejable por el usuario, adaptada a sus necesidades y preferencias. Según se plantea en la conferencia *ACM RecSys*(2013) una recomendación es una forma particular de filtrado de información, que explota los comportamientos pasados y similitudes de usuario para generar una lista de elementos de información que se adaptan personalmente a las preferencias de un usuario final.

Los sistemas de recomendación tienen como particularidad que se diseñan para entender las necesidades individuales de los usuarios a los que asisten, siendo claves para su adopción la percepción de los consumidores/usuarios sobre el grado en que el sistema de recomendación les entienden y les proporcionan información personalizada útil (Komiak y Benbasat, 2006). Los sistemas de recomendación aglutinan una gran variedad de necesidades de los usuarios, ayudándolos en su proceso de selección de información útil.

Pueden ser utilizados en los jueces en línea para ayudar a los usuarios a establecer mejor sus competencias y habilidades. El núcleo de los sistemas de recomendación es la fuente de datos que serían los elementos a analizar, los usuarios que interactúan con él y las transacciones que se realizan para poder hacer la recomendación, y también como parte de su núcleo están las técnicas que utilizan para hacer la recomendación, que cada técnica está basada en el tipo de recomendación que se utilice.

Según lo anteriormente analizado se considera que los sistemas de recomendación reciben diferentes clasificaciones para la selección de los elementos; basado en las técnicas que utilizan para realizar el filtrado, según Peis y otros(2012) se clasifican en sistemas de recomendación de filtrado social, también denominados de filtrado en colaboración o colaborativos, sistemas de recomendación basados en contenido y sistemas de recomendación basados en factores económicos. Como también Moya e Invarato(2014a) consideran que son clasificados en filtrado basado en contenido, filtrado demográfico y filtrado colaborativo. Yera(2010) considera que se pueden clasificar en cuatro tipos basados en el contenido, sistemas de soporte a la recomendación, sistemas basados en filtrado colaborativo, sistemas de recomendación híbridos, también Ricci y otros(2010) los clasifican en demográficos, basado en conocimientos, en filtrado colaborativo, basado en contenido, basado en comunidad y en sistema de recomendación híbrido. Después de realizar el estudio de las diferentes clasificaciones que varios autores nombran, se considera necesario profundizar en tres clasificaciones básicas: filtrado colaborativo, filtrado basado en contenido y sistema de recomendación híbrido.

- **Filtrado colaborativo**

El sistema que utiliza esta técnica de filtrado está basado en obtener recomendaciones utilizando la información proporcionada por usuarios similares al usuario en cuestión. Filmaffinity¹⁰ es uno de los sitios que es popular entre los usuarios y utiliza este tipo de filtrado según Moya e Invarato(2014a). La principal fortaleza de este enfoque está en la posibilidad de una total personalización de las recomendaciones y la simplicidad de la representación desde el punto de vista computacional. Tiene capacidad para identificar usuarios similares a pesar de que puedan parecer heterogéneos analizando estrictamente los accesos a productos que realizan. Según Moya e Invarato(2014b) el filtrado colaborativo presenta dos clasificaciones basándose en los algoritmos que utiliza para su recomendación.

- ✓ **Métodos basados en memoria:** emplean métricas de similitud para determinar el parecido entre una pareja de usuarios. Para ello calculan los elementos que han sido votados por ambos usuarios y comparan dichos votos para calcular la similitud.
- ✓ **Métodos basados en modelos:** utilizan la matriz de votaciones para crear un modelo a través del cual establecen el conjunto de usuarios similares al usuario activo. Algunos ejemplos de estos modelos son los clasificadores bayesianos, las redes neuronales, algoritmos genéticos, sistemas borrosos y la técnica de descomposición matricial basada en la técnica matemática del SVD.

¹⁰ <http://www.filmaffinity.com>)



Figura 1. Algoritmos de filtrado colaborativo (Moya, 2014b)

JMSD: Índice de Jaccard.

MSD: Diferencia cuadrática media.

SVD: Descomposición en valores singulares

Correlación: Correlación de Pearson.

Desventajas

- ✓ Este tipo de sistemas tiende a ofrecer resultados pobres cuando se dispone de poca información sobre los usuarios o estos tienen gustos muy heterogéneos.
- ✓ No funciona bien para usuarios cuyo comportamiento pueda considerarse como no nítido, en el sentido de que pertenece a dos grupos de usuarios distintos, cada uno de los cuales define un claro comportamiento.
- ✓ Cuando un nuevo elemento se añade al sistema, como no ha sido evaluado por ningún usuario, no hay forma de recomendarlo.
- ✓ Si un usuario particular no se identifica con los gustos de ningún otro usuario del sistema, no es factible hallar vecinos cercanos, y por lo tanto hacer recomendaciones.
- ✓ Se requiere un mínimo de usuarios para elaborar las predicciones.

• Filtrado basados en contenido

La recomendación viene dada por las informaciones que el usuario ha mostrado interés anteriormente. Moya en Invarato(2014a) plantean que YouTube es unos de los sitios que son reconocidos en Internet que presentan este tipo de recomendación.

Estos sistemas son poco fiables cuando no se tiene mucha información acerca del usuario. Presenta problema de la “estabilidad-plasticidad” que consiste en adaptarse ante los datos relevantes y mantenerse estable ante datos irrelevantes.

Desventajas

- ✓ Se hace un análisis automático de los elementos, considerando atributos predefinidos pero se deja de lado otros atributos relevantes.
- ✓ Las sugerencias son hechas en función del historial de elementos elegidos por el usuario, por lo tanto el usuario sólo puede recibir recomendaciones que concuerden con su perfil.

Ventajas y desventajas comunes para ambos tipos de filtrados. Según Albín(2009).

Ventajas

- ✓ No necesitan conocimiento del campo que tratan los servicios.
- ✓ Adaptabilidad, es decir, capacidad de mejorar con el tiempo.
- ✓ El sistema es capaz de auto realimentarse.

Desventajas

- ✓ Depende de disponer de un histórico de datos grande.

• **Filtrado Híbrido**

Combinan dos o más técnicas de recomendación para obtener mayor eficacia sin los inconvenientes de cada sistema (Burke, 2002). Ejemplo de un sistema de recomendación híbrido lo contiene Amazon. Los sistemas de recomendación basados en el filtrado híbrido, contienen varios tipos de hibridación.

- ✓ **Ponderado:** se ponderan los resultados para obtener una única lista.
- ✓ **Conmutación:** se elige el sistema a utilizar dependiendo del caso.
- ✓ **Mixto:** contiene varias listas de recomendación.
- ✓ **Combinación de funciones:** datos de entrada de diferentes sistemas.
- ✓ **Cascada:** se refina la lista con sistemas de recomendación en serie.
- ✓ **Aumento de características:** se añade el resultado (predicción, similitud) de una técnica como entrada de la siguiente.
- ✓ **Meta-nivel:** el modelo aprendido se usa como entrada del siguiente.

1.2.1 Limitaciones de los algoritmos en sistemas de recomendación

• **Arranque en frío**

Constituye una de las cuestiones críticas a resolver al inicio del despliegue de todo sistema de recomendación, independientemente de la línea que se haya seguido en su desarrollo. Este problema hace referencia a las situaciones en las que se dispone solamente de un reducido número de

evaluaciones o consideraciones para realizar las recomendaciones solicitadas, puede ocurrir específicamente bajo tres escenarios diferentes:

- ✓ Nuevo usuario: cuando un nuevo usuario se registra en el sistema de recomendación, existen muy pocos datos disponibles para definir con cierta precisión su perfil.
- ✓ Nuevo elemento: cuando un nuevo elemento es adicionado al catálogo, inicialmente no presenta evaluaciones.
- ✓ Nuevo sistema: cuando se pone a funcionar una nueva instancia de un sistema de recomendación, el número promedio de evaluaciones o consideraciones por usuario y por elemento es bajo, lo que puede afectar significativamente la calidad de las recomendaciones.

Las principales estrategias para la mitigación de esta problemática han sido la utilización de sistemas híbridos que combinen la recomendación basada en el contenido con el filtrado colaborativo, así como el empleo de criterios emergentes tales como la popularidad y la entropía de los elementos (Mamunur y otros, 2002).

- **Dispersión de los datos**

En todo sistema de recomendación se ha comprobado que, en la práctica, el número de evaluaciones reales que se posee acerca de un elemento específico queda muy por debajo del número mínimo que se requiere para hacer recomendaciones efectivas, haciéndose particularmente crítico para aquellos usuarios con preferencias alejadas de las de la mayoría. Esto implica que la predicción efectiva a partir de un número reducido de ejemplos juegue un papel clave para el éxito del sistema.

Una manera de enfrentar esta limitante es utilizar datos almacenados en el perfil del usuario a la hora de establecer la función de similitud. Así, para determinar si un usuario es semejante a otro, no sólo se tendría en cuenta si ambos muestran su preferencia por determinados elementos, sino que también entran en consideración la edad, el género, el nivel educacional, entre otros. Esta extensión del tradicional filtrado colaborativo, en ocasiones ha sido referenciada como filtro demográfico.

Se han desarrollado también soluciones teóricas que han ido enfocadas a la reducción de la dimensión de los datos con vistas a elevar la calidad de la información que se extrae de estos (Billsus y Pazzani, 2009).

- **Sobre-especialización**

La sobre-especialización en sistemas de recomendación está directamente asociada con los sistemas basados en el contenido. Este problema aparece cuando solamente se recomiendan elementos que se corresponden fuertemente con el perfil del usuario, limitando a estos a recibir únicamente sugerencias de elementos similares a aquellos que ya ha evaluado o considerado. Con vistas a garantizar cierta diversidad en cuanto a las recomendaciones brindadas a determinado usuario, en muchos sistemas de recomendación se ha dado cabida a la aleatoriedad, a través de técnicas como la de los algoritmos

genéticos. El uso de sistemas de recomendación híbridos, donde se utilicen como base para la recomendación tanto la información del propio usuario como la de usuarios semejantes, también se ha utilizado para evitar la tendencia a la aparición de esta problemática (Yera, 2010).

Después de analizadas las clasificaciones de los sistemas de recomendación y sus diferentes tipos de hibridación, se propone como técnica a utilizar **filtrado híbrido en cascada**. Esta técnica se basa en utilizar algún método de recomendación para elaborar una lista de posibles recomendaciones y a partir de esta lista de recomendaciones, aplicar un segundo algoritmo de recomendación, es decir, un sistema de recomendación refina las recomendaciones dadas por otro sistema de recomendación. Esta decisión de los autores de este trabajo responde a que **varias fuentes** han demostrado cómo a través de este enfoque la calidad de las recomendaciones aumenta considerablemente (Sobecki, 2002; Ricci y otros, 2010), constituyendo de esta forma una alternativa a tener en cuenta a la hora de diseñar un sistema de este tipo.

1.3 Tecnologías de desarrollo de software

Las tecnologías que se mencionan son las utilizadas en la base tecnológica con la que se desarrolló el COJ, por eso la propuesta a desarrollar debe basarse en las mismas tecnologías utilizadas. Solo se mencionan las tecnologías y una breve descripción de las mismas, destacando principalmente por qué fueron escogidas desde un inicio para el desarrollo del COJ. Además se decide también utilizar la misma metodología utilizada por los desarrolladores del COJ.

1.3.1 Lenguaje de programación

- **Java**

Es un lenguaje de programación que se caracteriza por ser simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico.

Este lenguaje fue seleccionado debido a que le permitió a los desarrolladores utilizar experiencias ya adquiridas en el uso de la plataforma java, así como reutilizar las habilidades en el uso de las herramientas asociadas al lenguaje, ayudando así en las deficiencias que pueda presentar la plataforma. Además la versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para el desarrollo de aplicaciones distribuidas.

- **HTML 5**

El lenguaje de marcado de hipertexto constituye un conjunto de etiquetas utilizadas para definir una página web mediante su estructuración en forma de texto, imágenes, vínculos y otros elementos que la componen (Gómez, 2011).

- **CSS 3**

Las hojas de estilo en cascada, es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML), constituye la mejor forma de separar los contenidos y su presentación, además es imprescindible para crear páginas web complejas (Hechavarria, 2011).

1.3.2 Framework

Según Eguluz (2014) *framework* es un conjunto de herramientas, librerías, convenciones y buenas prácticas que pretenden encapsular las tareas repetitivas en módulos genéricos fácilmente reutilizables.

Spring: Es un *framework* para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

Para la selección del *framework* los desarrolladores del COJ se basaron en las principales características que presenta *Spring*.

- Spring MVC es muy flexible, ya que implementa toda su estructura mediante interfaces. Además, todas las partes del *framework* son configurables a través de *plugins*.
- Los controladores de *Spring MVC* se configuran mediante Inversión de Control como los demás objetos.
- Resulta más fácil probar las partes de Spring MVC debido a que evita forzosamente la herencia de una clase y la dependencia directa en el controlador del *servlet*¹¹ que despacha las peticiones (Lobaina y Roque, 2012).

1.3.3 Sistema Gestor de Bases de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un sistema de *software* que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos (Bernot, 2011).

PostgreSQL

Sistema gestor de base de datos relacional orientado a objetos y libre, publicada bajo licencia BSD. Es un programa de código abierto, dirigido por una comunidad de desarrolladores llamada PGDG (*PostgreSQL Global Development Group*).

¹¹ <http://www.servlets.com>

Esta herramienta fue seleccionada por los desarrolladores del COJ debido a que es una herramienta multiplataforma y por estar licenciado bajo la licencia de GNU/GPL, además por las características distintivas que son muy a fines con el sistema desarrollado como por ejemplo soporta una capacidad de almacenamiento en el orden de los TB¹², posee mayor escalabilidad (soporta una mayor cantidad de peticiones concurrentes), y tiene la capacidad de comprobar la integridad referencial.

1.3.4 Servidor web

Servidores web

Es la tecnología que tiene implícito programas informáticos que procesan aplicaciones realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente (EcuRed, 2014).

Apache-Tomcat

Funciona como un contenedor de servidores desarrollado bajo el proyecto Jakarta en la *Apache Software Foundation*. Tomcat implementa las especificaciones de los servidores y de *Java Server Pages* de *Sun Microsystems*.

Su selección fue basada principalmente en que posee licencia libre lo cual facilita posibles actualizaciones, aunque su principal competidor *Glassfish*, también posee licencia para software libre, pero se argumenta que puede ser cerrada en cualquier momento lo que puede causar la pérdida de soporte y actualización al servidor web.

1.3.5 Entorno Integrado de Desarrollo

Netbeans

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. *Netbeans* IDE es un producto libre y gratuito sin restricciones de uso.

Los desarrolladores del COJ prefirieron *Netbeans* debido a que ya tenían experiencias en su utilización, además porque brinda mejor soporte que otras herramientas analizadas y porque se integra fácilmente con las otras herramientas que fueron seleccionadas como por ejemplo, presenta librería para el trabajo con Spring, se integra de forma natural con el servidor web Apache-Tomcat y porque tiene una interfaz más amigable e intuitiva.

¹² Tera Bytes

1.3.6 Herramienta de modelado

Herramientas CASE

Se puede definir a las herramientas de Ingeniería de Software Asistida por Ordenador (CASE por sus siglas en inglés), como diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y dinero.

Las herramientas CASE son útiles en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores (Pullés, 2012).

Existen variadas herramientas CASE dentro de las que se encuentran *Erwin*, *Rational Rose*, *Visual Paradigm*, entre otras. Se determinó que se utilizaría Visual Paradigm por ser una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software, es compatible con una amplia gestión de casos de uso y diseño de base de datos y proporciona medidas más eficaces en el análisis y diseño de sistemas (Visual-Paradigm.com, 2014).

Después de analizadas las herramientas y tecnologías a utilizar se puede seleccionar la metodología de desarrollo de software a utilizar.

1.3.7 Metodología de desarrollo de software

SXP es una metodología compuesta por SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos. XP más bien es una metodología encaminada para el desarrollo; 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 el éxito del proyecto.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Esta metodología ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo (Peñalver, 2008).

Se decide mantener la misma metodología que se utiliza para el desarrollo del COJ, debido a que se adapta fácilmente a las características del equipo de desarrollo y del sistema de recomendación a

desarrollar, y la forma de adaptación que tiene a las actividades que se deben realizar para la obtención del sistema final que da cumplimiento al objetivo de la presente investigación.

1.4 Métodos de validación

1.4.1 Métricas de validación

Para determinar la precisión de un sistema de recomendación se utilizan métricas que están divididas en dos grupos fundamentales: aquellas que están enfocadas a evaluar cuán cerca está la predicción, por parte de la aplicación, de la votación que daría un usuario a un determinado elemento con respecto a la evaluación real dada por este usuario; y un segundo grupo que se dedica a medir la frecuencia con la que un sistema hace suposiciones correctas con respecto a si un elemento es realmente apropiado o no para un usuario en específico (Herlocker,2004).

Dada las características del sistema en cuestión, las métricas pertenecientes al segundo grupo son las más apropiadas para realizar la evaluación de este. Estas son usualmente llamadas métricas de precisión de la clasificación.

- **Precisión**

Precisión según Cristache(2009) es una de las métricas más populares para evaluar los sistemas de recomendación de información. Precisión mide la capacidad de presentar al usuario final sólo los elementos relevantes. Es considerada la métrica de precisión de la clasificación por excelencia a utilizar en los sistemas de recomendación.

La aplicación de Precisión en este tipo de sistema difiere ligeramente con respecto a la forma de aplicarla en otros marcos de propósito más general referidos como parte de la evaluación *offline*, normalmente se tienen datos con los elementos que cada usuario ha utilizado o preferido. Para realizar la prueba, se selecciona un usuario específico, y se divide este conjunto de elementos, en este caso problemas resueltos, en dos subconjuntos, llamados conjunto de pre-recomendaciones, y conjunto de pruebas. Seguidamente al conjunto de pre-recomendaciones se le aplica el algoritmo a validar, y se seleccionan las n-primeras recomendaciones generadas por este. A su vez, los elementos que aparecen tanto en el conjunto de pruebas como en el de las n-primeras recomendaciones, se consideran como parte de un conjunto llamado *hit set*.

Dentro de este contexto, el Precisión se definen de la siguiente manera (Cristache, 2009):

$$precision = \frac{\text{tamaño del hit set}}{\text{Cantidad de } n - \text{primeras} - \text{recomendaciones}}$$

Figura 2. Precisión (Yera, 2010)

- **Error medio absoluto (MAE)**

El error medio absoluto, también denominado MAE (*Mean Absolute Error*), calcula la distancia absoluta existente entre las predicciones realizadas y la votación real del usuario (Moya, 2013).

Se define C_u como el conjunto de elementos valorados por el usuario u para los que ha sido posible obtener la predicción.

$$C_u = \{i \in I | r_{u,i} \neq \bullet \wedge p_{u,i} \neq \bullet\}$$

Figura 3. Conjunto de elementos valorados por el usuario (Moya, 2013)

Es calculado el MAE de un usuario, MAE_u como el valor medio de la diferencia absoluta entre las predicciones realizadas y las votaciones reales del usuario:

$$MAE_u = \frac{1}{\#C_u} \cdot \sum_{i \in C_u} |r_{u,i} - p_{u,i}|$$

Figura 4. MAE para un usuario (Moya, 2013)

Es calculado el MAE como el promedio del error medio absoluto cometido en cada usuario.

$$MAE = \frac{1}{\#U} \cdot \sum_{u \in U} MAE_u$$

Figura 5. Promedio del MAE (Moya, 2013)

1.4.2 Pruebas de sistema

Las pruebas de sistema son aquellas que se hacen cuando el software ya está funcionando como un todo. Dentro de las pruebas de sistema se encuentran las pruebas de recuperación, las pruebas de seguridad, las pruebas de resistencia, y las pruebas de rendimiento (Pressman, 2002).

Las pruebas de rendimiento a su vez se clasifican en varias categorías, entre las que están las pruebas de contención, que se enfocan en la capacidad del elemento a probar para manejar de manera aceptable la demanda de múltiples actores, y las pruebas de carga y estrés, las que se usan para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga variable, simulándose la carga de trabajo a la que está sometido en tiempo real, así como para detectar posibles errores que sólo suelen aparecer en situaciones críticas.

Dadas las características del sistema que se desarrolla de tener un intercambio continuo con el usuario, se decide la ejecución de pruebas de carga y estrés como actividad prioritaria para realizar la validación del mismo.

1.4.2.1 Pruebas de carga

Realizar una prueba de carga consiste en exponer un sistema a un uso continuo y constante, controlado estadísticamente (Herlocker, 2004). Las dos principales razones por las cuales se realiza este tipo de prueba son para dar soporte a las pruebas de confiabilidad del software, y para asegurar de una manera efectiva, la eficacia de los resultados de todas las pruebas de rendimiento. En este tipo de pruebas, la carga suele ser variada desde una cantidad mínima cercana a cero, llegando hasta los valores que el sistema en términos de recursos pueda sostener.

Para realizar la prueba de carga se propone como herramienta a utilizar:

- **Apache JMeter**

Es una aplicación de escritorio de código abierto desarrollada 100% con Java. Diseñada para realizar pruebas funcionales de comportamiento y para medir el rendimiento. Originalmente se diseñó para probar aplicaciones Web, pero se ha ampliado a otras funciones de prueba. Se puede utilizar para simular una carga pesada en un servidor, en la red o a un objeto para poner a prueba su resistencia o para analizar el rendimiento global en diferentes tipos de carga. Se utiliza para realizar pruebas a la propuesta de solución planteada.

1.4.3 Pruebas de integración

Las pruebas de integración buscan probar la combinación de las distintas partes de la aplicación para determinar si funcionan correctamente en conjunto. Esto es útil para ver cómo se comunican los *servlets* con las páginas de HTML. Las pruebas de integración se verifican mediante la descripción de caso de pruebas especiales.

1.4.4 Pruebas de validación

La validación puede definirse de muchas formas, pero una simple definición es que la validación se consigue cuando el software funciona de acuerdo con las expectativas razonables del cliente.

Dentro de las pruebas de validación se encuentran las **pruebas de aceptación** para permitir que el cliente valide todos los requisitos. Las realiza el usuario final en lugar del responsable del desarrollo del sistema, una prueba de aceptación puede ir desde un informal caso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas (Pressman, 2002).

1.4.4.1 Pruebas de aceptación

Las pruebas de aceptación son de suma importancia para el correcto funcionamiento de la aplicación, antes de su uso en un ambiente de producción. Estas pruebas son creadas a partir de las historias de usuarios y desarrolladas por el cliente o un especialista de la aplicación, conducida a determinar como el sistema satisface sus criterios de aceptación validando los requisitos funcionales que han sido levantados para el desarrollo de la solución. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación.

1.5 Conclusiones parciales

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

- Realizado el estudio a los principales jueces en línea se determinó los principales elementos y ventajas relacionados con el entorno de los sistemas de recomendación en los jueces en línea.
- Después del análisis realizado a las metodologías y tecnologías utilizadas en el COJ se decide la utilización de la misma base tecnológica, SXP como metodología para guiar el proceso de desarrollo, Java(jdk 1.7) como lenguaje de programación, *Netbeans* 7.4 como IDE de desarrollo, *Spring* 3.0 como *framework* para el desarrollo de la capa de presentación, como servidor web se decide utilizar *Apache-Tomcat* 7.0.40, como sistema gestor de bases de datos *PostgreSQL* 9.1, *Visual Paradigm* como herramienta de modelado y *Apache JMeter* 2.8 como herramienta de validación.
- La selección de los métodos de validación, permitirán que la propuesta de solución pueda ser evaluada una forma menos compleja y fácil de interpretar.

Capítulo 2: Descripción del sistema de recomendación para el Juez en Línea Caribeño

En el presente capítulo se describe la propuesta de solución basada en un sistema de recomendación que permita facilitar la interacción de sus usuarios con el COJ. El sistema permite recomendar problemas ajustados a las preferencias del usuario, basándose en el historial y el perfil de estos. Además de describe las fuentes de datos y la técnica a utilizar para la selección del algoritmo para su implementación, se muestran segmentos de código que representan los algoritmos principales y se generan artefactos según la metodología utilizada.

2.1 Características del COJ

En el COJ cada usuario cuenta con su perfil, donde se les solicita informaciones que identifican al usuario, así como sus preferencias.

- ✓ Nombre de usuario
- ✓ Alias
- ✓ Nombre
- ✓ Apellidos
- ✓ Correo-e
- ✓ País (Selección)
- ✓ Institución (Selección)
- ✓ Lenguaje de la Interfaz (Selección)
- ✓ Lenguaje de Programación (Selección)
- ✓ Contraseña
- ✓ Confirmación de la Contraseña
- ✓ Género (Selección)
- ✓ Fecha de nacimiento (Selección)
- ✓ Mostrar fecha de nacimiento (Marcar)
- ✓ Mostrar correo-e (Marcar)
- ✓ Recibir notificaciones vía correo-e (Marcar)
- ✓ Pregunta de seguridad (Selección)
- ✓ Respuesta de seguridad

Dentro de las informaciones solicitadas se encuentran las que son imprescindibles y necesarias para poder realizar una recomendación que sea la más aceptada para cada usuario. Cada información

seleccionada para la recomendación es asociada a los contenidos presentes en el sitio como son los problemas, los cuales presentan metadatos que pueden ser de gran utilidad para cada recomendación.

- ✓ Metadatos de los problemas
 - Descripción
 - Especificación de entrada
 - Especificación de salida
 - Ejemplo de entrada
 - Ejemplo de salida
 - Sugerencia(s)
 - Creado por
 - Adicionado por
 - Fecha de adición
 - Tiempo límite (ms)
 - Prueba límite (ms)
 - Memoria límite (kb)
 - Salida límite (mb)
 - Tamaño límite (bytes)
 - Lenguajes activados

Una vez analizadas las características del COJ, y cuáles de ellas pueden ayudar a realizar una buena recomendación a los usuarios, se propone un **sistema de recomendación híbrido en cascada** por conmutación de un modelo colaborativo y de un modelo basado en conocimiento. Esta hibridación permite aprovechar las ventajas de cada modelo de recomendación y superar las limitaciones individuales de cada uno.

EL sistema de recomendación que se propone a los inconvenientes planteados y a las necesidades del COJ, donde realiza recomendaciones a los usuario a no autenticados utilizando la votación de los problemas, y a los usuarios autenticados enfocándose en el perfil d preferencias que presenta cada uno, permitiendo ajustar la recomendaciones a sus afinidades y preferencias



Figura 6. Propuesta de solución

2.2 Sistema de recomendación híbrido

Para crear un sistema híbrido colaborativo basado en contenido, los perfiles de usuario se mantienen según el análisis de los contenidos de los elementos, y directamente se comparan esos perfiles para determinar la similitud entre usuarios para una recomendación colaborativa.

El algoritmo híbrido-mixto de filtrado colaborativo y basado en contenido, permite la recomendación de contenidos en entornos compuestos por comunidades de personas con intereses similares, basándose en el modelado de datos en matrices y uso de métricas de similitudes, correlaciones y predicciones. Debido a la dinámica colaborativa de las comunidades, el algoritmo genera recomendaciones teniendo en cuenta las preferencias individuales y las de otros usuarios, reduciendo los efectos de la sobre-especialización. A diferencia de otros algoritmos en los que las métricas están en función del nivel de interés del usuario por los contenidos y sus atributos, en esta propuesta el algoritmo híbrido filtra información en función del nivel de interés por las comunidades, haciendo un uso eficiente de los niveles de interés registrados por cada usuario y reduciendo la dispersión de datos. De esta manera, es posible: el cálculo anticipado de vecinos que permitan la reducción del error en las predicciones; y la reducción de los efectos negativos del

problema *Cold-Start*, obteniendo recomendaciones aun con una baja cantidad de niveles de interés registrados en el sistema. Por otro lado, el algoritmo tiene en cuenta la dinámica de las comunidades, adaptando el concepto de vecino tratado en la teoría de los sistemas de recomendación y, de esta manera, reducir la carga computacional del proceso de filtrado.

2.3 Modelo de negocio

Un modelo de procesos de negocio describe cómo funciona el negocio, es decir, describe las actividades involucradas en el negocio y la manera en que se relacionan entre ellas e interactúan con los recursos necesarios para lograr la meta del proceso. El modelado de procesos de negocios se utiliza para capturar, documentar o rediseñar procesos de negocio. Para llevarlo a cabo se pueden emplear diversos lenguajes, con diferente naturaleza, características y objetivos (Mora, 2009).

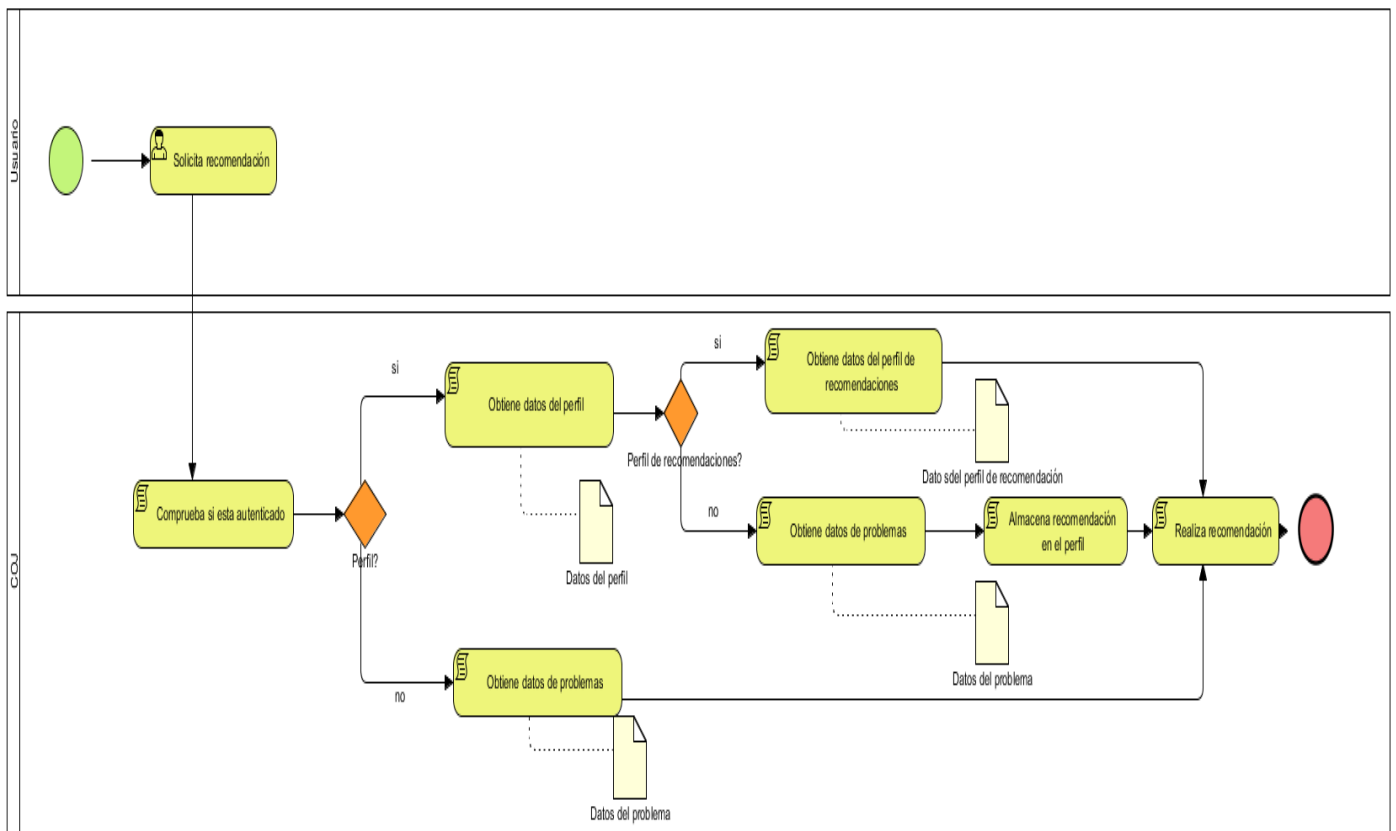


Figura 7. Modelo de negocio 1

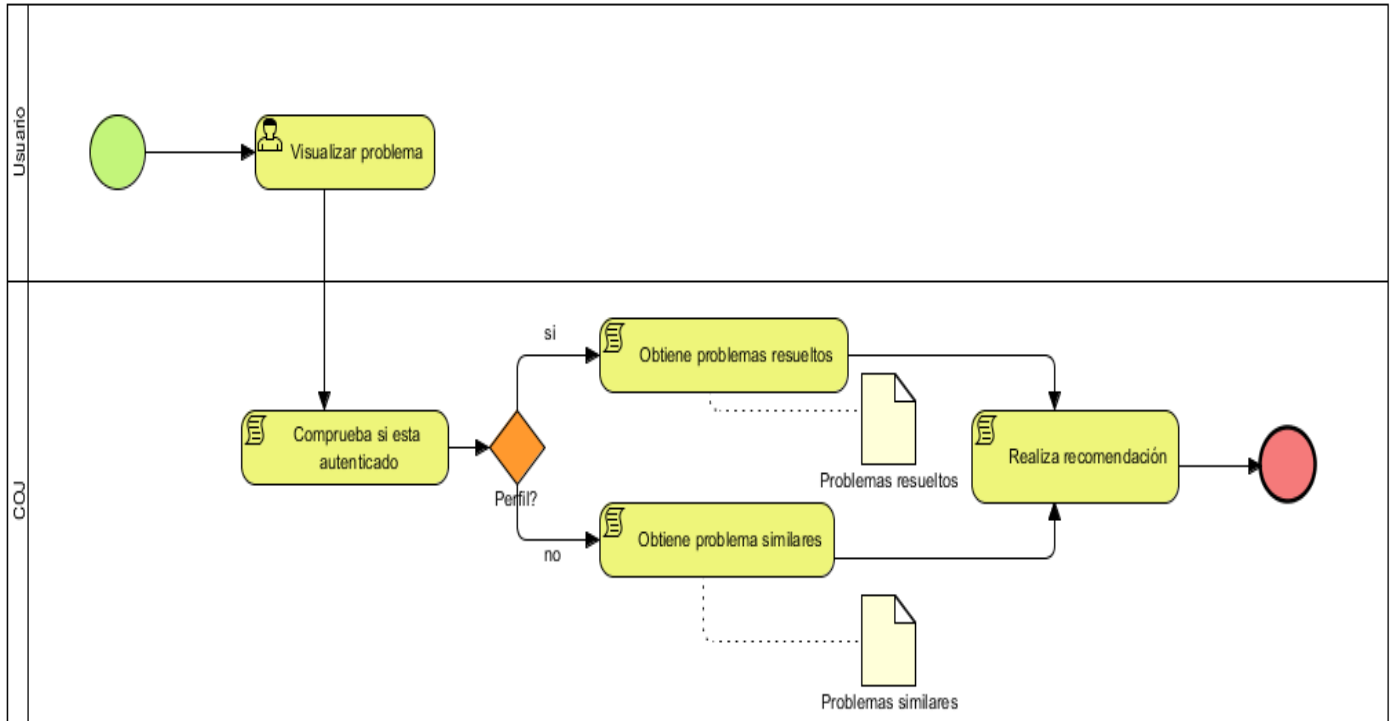


Figura 8. Modelo de negocio 2

2.4 Perfil algorítmico propuesto para el sistema de recomendación

Después de analizados los algoritmos de recomendación que se encuentran definidos para cada técnica de recomendación se decide utilizar por parte de la técnica colaborativa el algoritmo **K-vecinos**. Se basa en recomendar a un usuario (usuario activo) los elementos que le han gustado a usuarios con gustos similares al usuario activo, de ahí el nombre de los k-vecinos, considerando los vecinos como usuarios con gustos similares al usuario activo (Moya, 2014b). Además se utiliza la **Correlación de Pearson** para la selección de vecinos en casos de que se encuentren muy similares y se tenga que escoger entre uno y otro. Cuando el usuario no se encuentra autenticado se utiliza el algoritmo **arranque en frío (Cold Start)**.

- **Algoritmo K-vecinos implementado en el sistema**

```
// K-nearest neighbours
```

```
private List<UserProfile> KNN(UserProfile profile) {
```

```
    List<UserProfile> profiles = userDao.loadRelatedProfiles(profile);
```

```
    int K_INDEX_KNN = 3;
```

```
    PriorityQueue<UserProfile> cola = new PriorityQueue<UserProfile>(K_INDEX_KNN, new
```

```

    Comparator<UserProfile>() {
        @Override
public int compare (UserProfile o1, UserProfile o2) {
    return o2.getPearson ().compareTo(o1.getPearson());
}
});
for (UserProfile prof : profiles) {
    double cor = pearson(profile.getTags(), prof.getTags());
    prof.setPearson(cor);
    cola.add(prof);
    if (cola.size() > K_INDEX_KNN) {
        cola.remove();
    }
}
profiles.clear();
profiles.addAll(cola);
return profiles;
}

```

- **Algoritmo Correlación de Pearson**

```

private static double pearson(List<Double> l1, List<Double> l2) {
    double suma1 = 0, suma2 = 0, suma1q = 0, suma2q = 0, psum = 0;
    for (int i = Math.min(l1.size(), l2.size()); i >= 0; i--) {
        double a = 0, b = 0;
        if (i < l1.size()) {
            a = l1.get(i);
        }
        if (i < l2.size()) {
            b = l2.get(i);
        }
        suma1 += a;
        suma2 += b;
        suma1q += a*a;
        suma2q += b*b;
        psum += a*b;
    }
}

```

```

}
double num = psum;
double den = Math.sqrt(suma1q * suma2q);
return num / den;
}

```

- **Algoritmo de arranque en frío(Cold Start)**

```

@Transactional(readOnly = true)
public List<Problem> suggestedProblemsColdStart(String username, int uid, String orderby, String inv)
{
int PROBLEM_SUGGESTION_QTY = 5 * 2; //this is configuration
String notlogged = "";
if (username == null) {
List<Object> list = new LinkedList<Object>();
list.add(0);//complexity
list.add(2);//limit
notlogged += "AND (";
boolean first = false;
for (int c = 1; c <= 5; c++) {
list.set(0, c);
List<Problem> now = objects("recommender.notlogged.complexity", Problem.class, list.toArray());
for (Iterator<Problem> it = now.iterator(); it.hasNext() {
Problem problem = it.next();
if (first == true) {
notlogged += " OR ";
} else {
first = true;
}
notlogged += "abb ilike '%" + problem.getAbb() + "%";
}
}
notlogged += ") ";
}
String sql = getSql("recommender.notlogged");

```

```

List<Object> list = new LinkedList<Object>();
list.add(username);
list.add(PROBLEM_SUGGESTION_QTY);

if (orderby != null) {
    sql = sql.replace("<orderby>", getSql(inv.equals("asc") ? "order.all.problems.asc" :
"order.all.problems.desc"));
    sql = sql.replace("<orderby>", orderby);
} else {
    sql = sql.replace("<orderby>", "");
}
sql = sql.replace("<AND>", notlogged);
list.add(PROBLEM_SUGGESTION_QTY);
list.add(0);
List<Problem> problems = objects(sql, Problem.class, list.toArray());
return problems;
}

```

2.5 Requerimientos

Después de analizado el dominio del problema, se identifican los requisitos funcionales y no funcionales que tendrá el sistema que se desarrolla. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, mientras que los no funcionales son propiedades o cualidades que el producto debe tener. Los requisitos forman parte del modelo del sistema que se desarrolla (Pressman, 2005). A continuación se presentan los requisitos funcionales y no funcionales extraídos.

2.5.1 Requisitos funcionales

- RF1- Generar perfil de recomendaciones de usuarios
- RF2- Generar recomendación a usuarios no autenticados
- RF3- Generar recomendación a usuarios autenticados
- RF4- Generar recomendación de problemas similares a usuarios no autenticados
- RF5-Generar recomendación de problemas similares a usuarios autenticados
- RF6-Mostrar recomendación
- RF7-Valorar los problemas

2.5.2 Requisitos no funcionales

- **Usabilidad**

RNF1- El sistema podrá ser usado sobre ambiente web por personas con pocos conocimientos de informática.

RNF2- Es sistema mostrará debajo de cada problema los problemas que serán recomendados.

- **Confiabilidad**

RNF3- En caso de que el sistema presente alguna falla, los errores se deben mostrar sin detalles de información que pueda comprometer la seguridad e integridad del sistema, solo podrá mostrarse detalles de la información para los usuarios con privilegios de administración dentro del sistema.

RNF4- El sistema establecerá mecanismos para garantizar la confiabilidad e integridad de la información ante posibles accesos no autorizados.

RNF5- Se podrá acceder a la aplicación desde cualquier navegador web, ya sea en Internet Explorer (v8.0 o superior), Mozilla Firefox (v3.0 o superior), Chrome (v5.0 o superior), Opera (v9.0 o superior) y Safari (v4.0 o superior).

- **Eficiencia**

RNF6- Se esperan alrededor de hasta 150 usuarios conectados concurrentemente.

RNF7- Cada recomendación realizada debe poseer un tiempo de respuesta rápido.

- **Soporte**

RNF8- El sistema debe permitir ser mejorado, así como de incorporarle nuevas técnicas en caso de ser necesarias.

- **Interfaz**

RNF9- El diseño cumple con los estándares internacionales de desarrollo web aplicados al COJ, porque presenta una interfaz sencilla e intuitiva que garantiza mayor nivel de usabilidad.

- **Interfaz de Hardware**

RNF10- Requisitos mínimos de hardware: Core 2 duo 2.0 GHZ y 2GB RAM.

- **Interfaz de Software**

RNF11- PC Servidor web

Sistema Operativo: Ubuntu 13.04 o superior. Servidor web: apache Tomcat 7.x o superior
Lenguaje de programación: PHP 5.3.9 o superior (librerías: pgsq, curl, gd, db, uploadprogress).

RNF12- PC Servidor de BD

Sistema Operativo: Ubuntu 13.04 o superior. Sistema Gestor de Base de Datos *PostgreSQL* v8.4 o superior. Se empleará el Sistema Gestor de Base de Datos de *PostgreSQL PgAdmin III*.

RNF13- Utilizar el entorno de ejecución java “OpenJDK-JRE” en su versión 6 o superior.

- **Requisitos legales, de derecho de autor y otros.**

RNF14- Las tecnologías y herramientas utilizadas están bajo licencia GPL.

RNF15- El sistema posee una sección donde se publican los términos y condiciones de uso que reglamentan el acceso y uso del COJ y de todos sus subdominios o dominios secundarios incluyendo sus contenidos y servicios, puestos a disposición que deben ser de estricto cumplimiento por parte de los usuarios.

RNF16- Los derechos de autor sobre la información de terceros que se publica en el COJ pertenecen a sus respectivos autores y su publicación en el COJ se realiza sin ánimos de lucro y con un propósito informativo.

2.6 Lista de Reserva del Producto

La Lista de Reserva del Producto (LRP) es una lista priorizada donde se especifican los requisitos funcionales y no funcionales para tener en cuenta en el futuro proyecto a desarrollar, o sea cuenta con los requisitos que debe llevar la aplicación establecidos por el cliente, y se define en la etapa de planificación del proyecto.

Tabla 1. Lista de Reserva del Producto

Asignado a	Elemento	Descripción	Estimación	Estimado por
Requisitos funcionales				
Prioridad Máxima				
Analista Junet Doria Martínez	1	Generar perfil de recomendaciones de usuarios.	1	Analista
Programador José Carlos González Fernández	2	Generar recomendación a usuarios no autenticados	2	Programador
Programador José Carlos González Fernández	3	Generar recomendación a usuarios autenticados	2	Programador

Programador José Carlos González Fernández	4	Generar recomendación de problemas similares a usuarios no autenticados	1	Programador
Programador José Carlos González Fernández	5	Generar recomendación de problemas similares a usuarios autenticados	1	Programador
Analista Junet Doria Martínez	6	Mostrar recomendación	1	Analista
Prioridad Alta				
Analista Junet Doria Martínez	7	Valorar problema	1	Analista
Requisitos no funcionales				
Usabilidad				
	1	El sistema podrá ser usado sobre ambiente web por personas con pocos conocimientos de informática.		
	2	Es sistema mostrará debajo de cada problema los problemas que serán recomendados.		
Confiabilidad				
	3	En caso de que el sistema presente alguna falla, los errores se deben mostrar sin detalles de información que pueda comprometer la seguridad e integridad del sistema, solo podrá mostrarse detalles de la información para los usuarios con privilegios de administración dentro del sistema.		
	4	El sistema establecerá mecanismos para garantizar la confiabilidad e integridad de la información ante posibles accesos no autorizados.		

	5	Se podrá acceder a la aplicación desde cualquier navegador web, ya sea en Internet Explorer (v8.0 o superior), Mozilla Firefox (v3.0 o superior), Chrome (v5.0 o superior), Opera (v9.0 o superior) y Safari (v4.0 o superior).		
Eficiencia				
	6	Se esperan alrededor de hasta 150 usuarios conectados concurrentemente.		
	7	Cada recomendación realizada debe poseer un tiempo de respuesta rápido.		
Soporte				
	8	El sistema debe permitir de ser mejorado, así como de incorporarle nuevas técnicas en caso de ser necesarias.		
Interfaz				
	9	El diseño cumple con los estándares internacionales de desarrollo web aplicados al COJ, porque presenta una interfaz sencilla e intuitiva que garantiza mayor nivel de usabilidad.		
Interfaz de Hardware				
	10	Requisitos mínimos de hardware: Core 2 duo 2.0 GHZ y 2GB RAM.		
Interfaz de Software				
	11	PC Servidor web -Sistema Operativo: Ubuntu 13.04 o superior. -Servidor web: <i>Apache Tomcat 7.x</i> o superior -Lenguaje de programación: PHP 5.3.9 o superior (librerías: pgsq, curl, gd, db, uploadprogress).		

	12	<p>PC Servidor de BD</p> <p>-Sistema Operativo: Ubuntu 13.04 o superior.</p> <p>-Sistema-Gestor de Base de Datos <i>PostgreSQL</i> v8.4 o superior.</p> <p>-Se empleará el Sistema Gestor de Base de Datos de <i>PostgreSQL PgAdmin</i> III.</p>		
	13	Utilizar el entorno de ejecución java “OpenJDK-JRE” en su versión 6 o superior.		
Requisitos legales, de derecho de autor y otros				
	14	Las tecnologías y herramientas utilizadas están bajo licencia GPL.		
	15	El sistema posee una sección donde se publican los términos y condiciones de uso que reglamentan el acceso y uso del COJ y de todos sus subdominios o dominios secundarios incluyendo sus contenidos y servicios, puestos a disposición que deben ser de estricto cumplimiento por parte de los usuarios.		
	16	Los derechos de autor sobre la información de terceros que se publica en el COJ pertenecen a sus respectivos autores y su publicación en dicho Portal se realiza sin ánimos de lucro y con un propósito informativo		

2.7 Historias de usuario

Las historias de usuario guían la construcción de las pruebas de aceptación son utilizadas para estimar tiempos de desarrollo. En este sentido, sólo proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas. En el momento de implementar una historia de usuario, se debe detallar a través de la comunicación con el cliente. Estas son la base para las pruebas funcionales.

Tabla 2. Historia de Usuario No.1

Historia de Usuario																																																													
Número: 1	Nombre Historia de Usuario: Mostrar recomendación a usuarios no autenticados.																																																												
Requisitos funcionales implícitos: RF2- Generar recomendación a usuarios no autenticados RF6- Mostrar recomendación																																																													
Modificación de Historia de Usuario Número: Ninguna																																																													
Usuarios: José Carlos González Fernández Junet Doria Martínez	Iteración Asignada: 1																																																												
Prioridad en Negocio: Máxima	Puntos Estimados: 3																																																												
Riesgo en Desarrollo: Alta	Puntos Reales: 3																																																												
Descripción: Cuando el usuario no está autenticado en el COJ, también es necesario hacerle recomendaciones, las recomendaciones son generadas antes de ser mostradas, y se generan según los problemas más votados en el COJ.																																																													
Observaciones: -																																																													
Prototipo de Interfaz:																																																													
 <p>The screenshot displays the COJ website header with the logo and tagline "Thinking better!". Below the header is a navigation menu and a date/time stamp. The main content area is titled "Problem Recommendations" and includes a login form on the left and a table of recommendations on the right. The table lists various problems with their IDs, titles, and statistics.</p> <table border="1" data-bbox="440 1367 1414 1682"> <thead> <tr> <th>id</th> <th>title</th> <th>sub</th> <th>ac</th> <th>ac%</th> <th>score</th> </tr> </thead> <tbody> <tr> <td>2194</td> <td>The Trough Game</td> <td>10</td> <td>8</td> <td>80.00</td> <td>4.17</td> </tr> <tr> <td>1917</td> <td>Tile Exchanging</td> <td>69</td> <td>36</td> <td>52.17</td> <td>2.86</td> </tr> <tr> <td>1487</td> <td>Alarms</td> <td>923</td> <td>187</td> <td>20.26</td> <td>1.23</td> </tr> <tr> <td>2187</td> <td>CMPSelection</td> <td>82</td> <td>28</td> <td>34.15</td> <td>3.08</td> </tr> <tr> <td>1370</td> <td>Dragon Slayer</td> <td>109</td> <td>14</td> <td>12.84</td> <td>4.08</td> </tr> <tr> <td>1364</td> <td>Combination Of Integers</td> <td>43</td> <td>16</td> <td>37.21</td> <td>3.92</td> </tr> <tr> <td>1368</td> <td>Network Problem</td> <td>400</td> <td>128</td> <td>32.00</td> <td>1.42</td> </tr> <tr> <td>1624</td> <td>Counting Palindromes</td> <td>250</td> <td>98</td> <td>39.20</td> <td>1.87</td> </tr> <tr> <td>2325</td> <td>Eco-driving</td> <td>91</td> <td>22</td> <td>24.18</td> <td>4.00</td> </tr> </tbody> </table>		id	title	sub	ac	ac%	score	2194	The Trough Game	10	8	80.00	4.17	1917	Tile Exchanging	69	36	52.17	2.86	1487	Alarms	923	187	20.26	1.23	2187	CMPSelection	82	28	34.15	3.08	1370	Dragon Slayer	109	14	12.84	4.08	1364	Combination Of Integers	43	16	37.21	3.92	1368	Network Problem	400	128	32.00	1.42	1624	Counting Palindromes	250	98	39.20	1.87	2325	Eco-driving	91	22	24.18	4.00
id	title	sub	ac	ac%	score																																																								
2194	The Trough Game	10	8	80.00	4.17																																																								
1917	Tile Exchanging	69	36	52.17	2.86																																																								
1487	Alarms	923	187	20.26	1.23																																																								
2187	CMPSelection	82	28	34.15	3.08																																																								
1370	Dragon Slayer	109	14	12.84	4.08																																																								
1364	Combination Of Integers	43	16	37.21	3.92																																																								
1368	Network Problem	400	128	32.00	1.42																																																								
1624	Counting Palindromes	250	98	39.20	1.87																																																								
2325	Eco-driving	91	22	24.18	4.00																																																								

Tabla 3. Historia de Usuario No.2


Historia de Usuario																																																									
Número: 2	Nombre Historia de Usuario: Mostrar recomendación a usuarios autenticados.																																																								
Requisitos funcionales implícitos: RF1- Generar perfil de recomendaciones de usuarios. RF3- Generar recomendación a usuarios autenticados RF6- Mostrar recomendación																																																									
Modificación de Historia de Usuario Número: Ninguna																																																									
Usuarios: José Carlos González Fernández Junet Doria Martínez	Iteración Asignada: 1																																																								
Prioridad en Negocio: Máxima	Puntos Estimados: 4																																																								
Riesgo en Desarrollo: Alta	Puntos Reales: 4																																																								
Descripción: Cuando el usuario se autentica en el COJ, el sistema genera las recomendaciones usando como ayuda el perfil, a través del cual se genera un perfil de recomendaciones que se utiliza para mostrar las recomendaciones al usuario en cuestión.																																																									
Observaciones: -																																																									
Prototipo de Interfaz:																																																									
 <p>The screenshot shows the COJ website interface. The header features the COJ logo and the text 'Caribbean Online Judge' with the tagline 'Thinking better!'. Below the header is a navigation menu with links: HOME, ABOUT, DOC, TOOLS, FORUM, FAQ, LINKS, CONTACT US. The user is logged in as 'HELLO, JCFERNANDEZ'. The main content area is titled 'Problem Recommendations' and displays a table with the following data:</p> <table border="1"> <thead> <tr> <th>id</th> <th>status</th> <th>title</th> <th>sub</th> <th>ac</th> <th>ac%</th> <th>score</th> </tr> </thead> <tbody> <tr> <td>2325</td> <td>★</td> <td>Eco-driving</td> <td>91</td> <td>22</td> <td>24.18</td> <td>4.00</td> </tr> <tr> <td>1624</td> <td>★</td> <td>Counting Palindromes</td> <td>250</td> <td>98</td> <td>39.20</td> <td>1.87</td> </tr> <tr> <td>1917</td> <td>★</td> <td>Tile Exchanging</td> <td>69</td> <td>36</td> <td>52.17</td> <td>2.86</td> </tr> <tr> <td>1487</td> <td>★</td> <td>Alarms</td> <td>923</td> <td>187</td> <td>20.26</td> <td>1.23</td> </tr> <tr> <td>2194</td> <td>★</td> <td>The Trough Game</td> <td>10</td> <td>8</td> <td>80.00</td> <td>4.17</td> </tr> <tr> <td>2187</td> <td>★</td> <td>CMPSelection</td> <td>82</td> <td>28</td> <td>34.15</td> <td>3.08</td> </tr> <tr> <td>1627</td> <td>★</td> <td>Counting Zeros</td> <td>381</td> <td>262</td> <td>68.77</td> <td>0.70</td> </tr> </tbody> </table>		id	status	title	sub	ac	ac%	score	2325	★	Eco-driving	91	22	24.18	4.00	1624	★	Counting Palindromes	250	98	39.20	1.87	1917	★	Tile Exchanging	69	36	52.17	2.86	1487	★	Alarms	923	187	20.26	1.23	2194	★	The Trough Game	10	8	80.00	4.17	2187	★	CMPSelection	82	28	34.15	3.08	1627	★	Counting Zeros	381	262	68.77	0.70
id	status	title	sub	ac	ac%	score																																																			
2325	★	Eco-driving	91	22	24.18	4.00																																																			
1624	★	Counting Palindromes	250	98	39.20	1.87																																																			
1917	★	Tile Exchanging	69	36	52.17	2.86																																																			
1487	★	Alarms	923	187	20.26	1.23																																																			
2194	★	The Trough Game	10	8	80.00	4.17																																																			
2187	★	CMPSelection	82	28	34.15	3.08																																																			
1627	★	Counting Zeros	381	262	68.77	0.70																																																			

Tabla 4. Historia de Usuario No.3

Historia de Usuario					
Número: 3	Nombre Historia de Usuario: Mostrar recomendación de problemas similares a usuarios autenticados				
Requisitos funcionales implícitos: RF4- Generar recomendación de problemas similares a usuarios autenticados RF6- Mostrar recomendación					
Modificación de Historia de Usuario Número: Ninguna					
Usuarios: José Carlos González Fernández Junet Doria Martínez	Iteración Asignada: 1				
Prioridad en Negocio: Máxima	Puntos Estimados: 2				
Riesgo en Desarrollo: Alta	Puntos Reales: 2				
Descripción: Cuando el usuario se autentica en el COJ, el sistema muestra dentro de los metadatos de los problemas una sección que es en la que recomienda problemas similares al que se está mostrando y que el usuario no ha resuelto.					
Observaciones: -					
Prototipo de Interfaz:					
<div style="border: 1px solid #ccc; padding: 10px;"> <p>Sample output</p> <div style="border: 1px solid #ccc; background-color: #ffffcc; padding: 5px; margin-bottom: 10px;">1010</div> <p>Hint(s)</p> <p>Recommend</p> <p>We have carefully selected several similar problems for you: 1617 1917 1624 2325</p> <p>Your opinion</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%; border: 1px solid #ccc; padding: 5px;">Problem statement</td> <td style="text-align: center;">★ ★ ★ ★ ★</td> </tr> <tr> <td style="border: 1px solid #ccc; padding: 5px;">Problem solution</td> <td style="text-align: center;">★ ★ ★ ★ ★</td> </tr> </table> <p style="margin-top: 5px;"><input type="button" value="Vote"/></p> </div>		Problem statement	★ ★ ★ ★ ★	Problem solution	★ ★ ★ ★ ★
Problem statement	★ ★ ★ ★ ★				
Problem solution	★ ★ ★ ★ ★				

Tabla 5. Historia de Usuario No.4

Historia de Usuario					
Número: 4	Nombre Historia de Usuario: Mostrar recomendación de problemas similares a usuarios no autenticados				
Requisitos funcionales implícitos: RF4- Generar recomendación de problemas similares a usuarios no autenticados RF6- Mostrar recomendación					
Modificación de Historia de Usuario Número: Ninguna					
Usuarios: José Carlos González Fernández Junet Doria Martínez	Iteración Asignada: 1				
Prioridad en Negocio: Máxima	Puntos Estimados: 2				
Riesgo en Desarrollo: Alta	Puntos Reales: 2				
Descripción: Cuando el usuario se conecta en el COJ, el sistema muestra dentro de los metadatos de los problemas una sección que es en la que recomienda problemas similares al que se está mostrando.					
Observaciones: -					
Prototipo de Interfaz:					
<div style="border: 1px solid #ccc; padding: 10px; width: 80%; margin: auto;"> <p>Sample output</p> <div style="border: 1px solid #ccc; background-color: #ffffcc; padding: 5px; margin-bottom: 10px;">1010</div> <p>Hint(s)</p> <p>Recommend</p> <p>We have carefully selected several similar problems for you: 1617 1917 1624 2325</p> <p>Your opinion</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Problem statement</td> <td style="text-align: center;">★ ★ ★ ★ ★</td> </tr> <tr> <td>Problem solution</td> <td style="text-align: center;">★ ★ ★ ★ ★</td> </tr> </table> <p style="text-align: center;"><input type="button" value="Vote"/></p> </div>		Problem statement	★ ★ ★ ★ ★	Problem solution	★ ★ ★ ★ ★
Problem statement	★ ★ ★ ★ ★				
Problem solution	★ ★ ★ ★ ★				

Tabla 6. Historia de Usuario No.5

Historia de Usuario							
Número: 5	Nombre Historia de Usuario: Valorar problema						
Requisitos funcionales implícitos: RF7- Valorar problema							
Modificación de Historia de Usuario Número: Ninguna							
Usuario: Junet Doria Martínez	Iteración Asignada: 1						
Prioridad en Negocio: Alta	Puntos Estimados: 1						
Riesgo en Desarrollo: Medio	Puntos Reales: 1						
<p>Descripción: Para realizar recomendaciones a los usuarios no autenticados, es necesario tener la valoración de los problemas, porque las recomendaciones son generadas a según la valoración realizada a cada problema.</p>							
Observaciones: -							
<p>Prototipo de Interfaz:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Sample output</p> <div style="border: 1px solid #ccc; background-color: #ffffcc; padding: 5px; margin: 5px 0;">1010</div> <p>Hint(s)</p> <p>Recommend</p> <p>We have carefully selected several similar problems for you: 1617 1917 1624 2325</p> <p>Your opinion</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%; border-bottom: 1px solid #ccc; padding: 5px;">Problem statement</td> <td style="text-align: center; border-bottom: 1px solid #ccc;">★ ★ ★ ★ ★</td> </tr> <tr> <td style="border-bottom: 1px solid #ccc; padding: 5px;">Problem solution</td> <td style="text-align: center; border-bottom: 1px solid #ccc;">★ ★ ★ ★ ★</td> </tr> <tr> <td style="padding: 5px;"><input type="button" value="Vote"/></td> <td></td> </tr> </table> </div>		Problem statement	★ ★ ★ ★ ★	Problem solution	★ ★ ★ ★ ★	<input type="button" value="Vote"/>	
Problem statement	★ ★ ★ ★ ★						
Problem solution	★ ★ ★ ★ ★						
<input type="button" value="Vote"/>							

2.8 Tareas de Ingeniería

Tabla 7. Tarea de Ingeniería No.1

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Programación de la propuesta de recomendación para los usuarios no autenticados.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 semanas
Fecha Inicio: 21/02/2014	Fecha Fin: 14/03/2014
Programador Responsable: José Carlos González Fernández y Junet Doria Martínez	
Descripción: Se lleva a cabo la programación de las recomendaciones para los usuarios no autenticados.	

Tabla 8. Tarea de Ingeniería No.2

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 2
Nombre Tarea: Programación de la propuesta de recomendación para los usuarios autenticados.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 4 semanas
Fecha Inicio: 17/03/2014	Fecha Fin: 14/04/2014
Programador Responsable: José Carlos González Fernández y Junet Doria Martínez	
Descripción: Se lleva a cabo la programación de las recomendaciones para los usuarios autenticados.	

Tabla 9. Tarea de Ingeniería No.3

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 3
Nombre Tarea: Estudio de cómo mostrar recomendación de problemas similares para los usuarios autenticados.	
Tipo de Tarea : Investigación	Puntos Estimados: 2 semanas
Fecha Inicio: 15/04/2014	Fecha Fin: 29/04/2014
Programador Responsable: José Carlos González Fernández y Junet Doria Martínez	
Descripción: Se realiza un estudio de los principales elementos que se muestran en la recomendación de problemas similares para los usuarios autenticados.	

Tabla 10. Tarea de Ingeniería No.4

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 3
Nombre Tarea: Estudio de cómo mostrar recomendación de problemas similares para los usuarios no autenticados.	
Tipo de Tarea : Investigación	Puntos Estimados: 2 semanas
Fecha Inicio: 30/04/2014	Fecha Fin: 14/05/2014
Programador Responsable: José Carlos González Fernández y Junet Doria Martínez	
Descripción: Se realiza un estudio de los principales elementos que se muestran en la recomendación de problemas similares para los usuarios no autenticados.	

Tabla 11. Tarea de Ingeniería No.5

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 4
Nombre Tarea: Estudio de la forma en la que se pueden valorar los problema.	
Tipo de Tarea : Investigación	Puntos Estimados: 1 semanas
Fecha Inicio: 15/05/2014	Fecha Fin: 22/01/2014
Programador Responsable: Junet Doria Martínez	
Descripción: Se realiza un estudio de los principales elementos a tener en cuenta para valorar los problemas.	

2.9 Diagrama de clases del diseño

El diagrama de clases del diseño no es más que la representación gráfica de las clases que serán implementadas en el sistema. Este diagrama muestra las especificaciones y detalles más concretos de cada clase del sistema, así como su relación de asociación, composición o agregación existentes entre ellas (Ambler, 2003a).

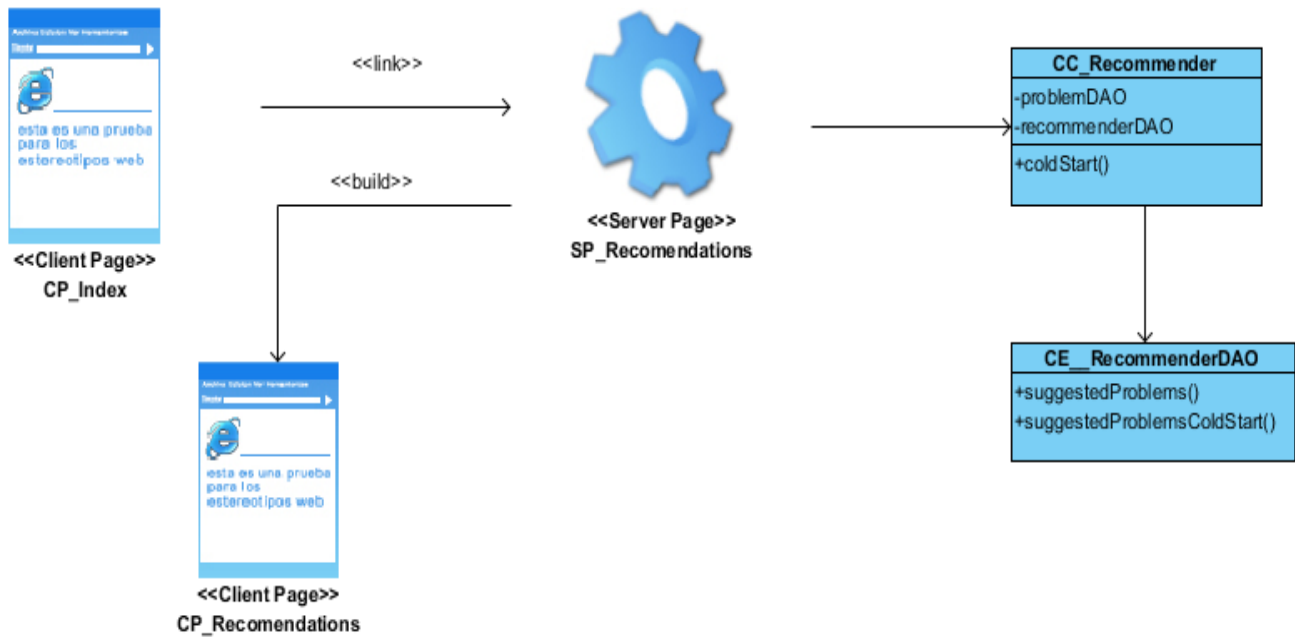


Figura 9. Diagrama de clases de diseño: Mostrar recomendación a usuarios no autenticados

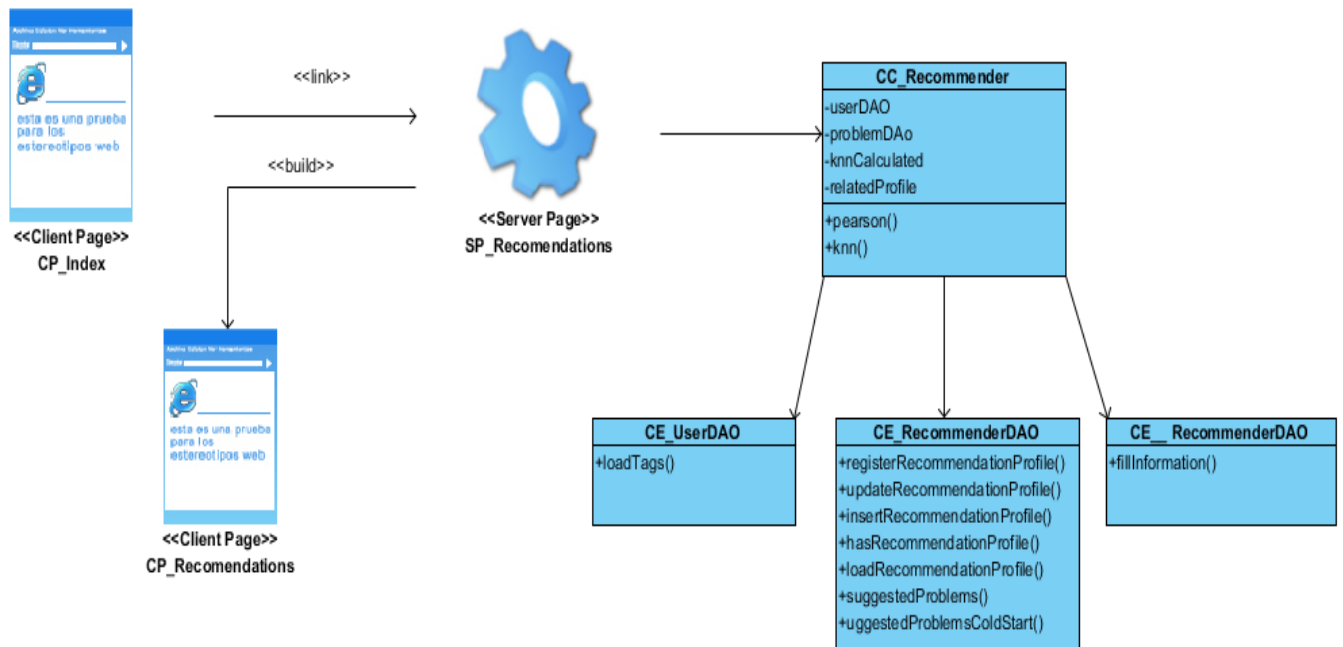


Figura 10. Diagrama de clases de diseño: Mostrar recomendación a usuarios autenticados

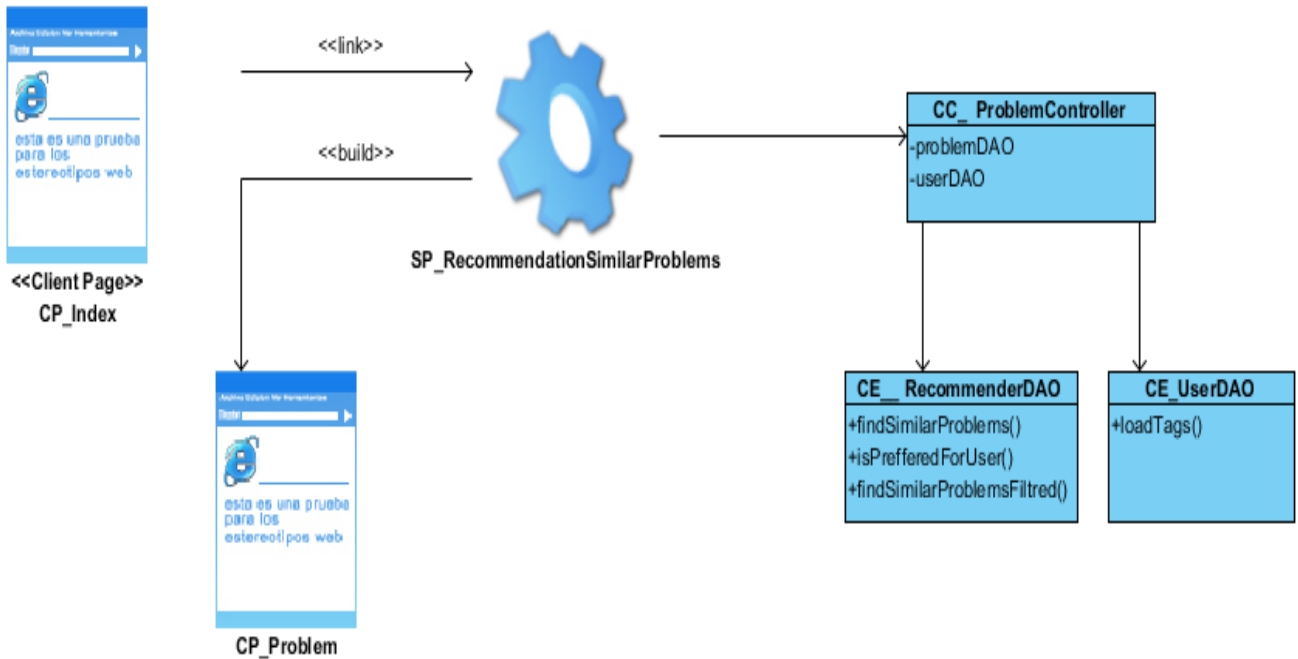


Figura 11. Diagrama de clases de diseño: Mostrar recomendación de problemas similares a usuarios autenticados

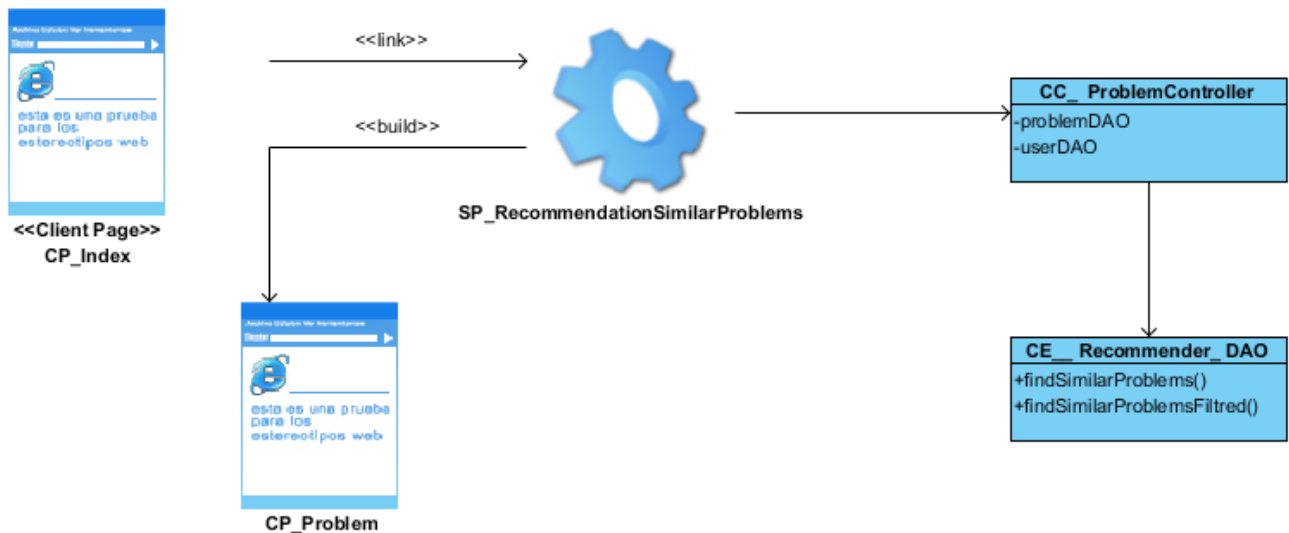


Figura 12. Diagrama de clases de diseño: Mostrar recomendación de problemas similares a usuarios no autenticados

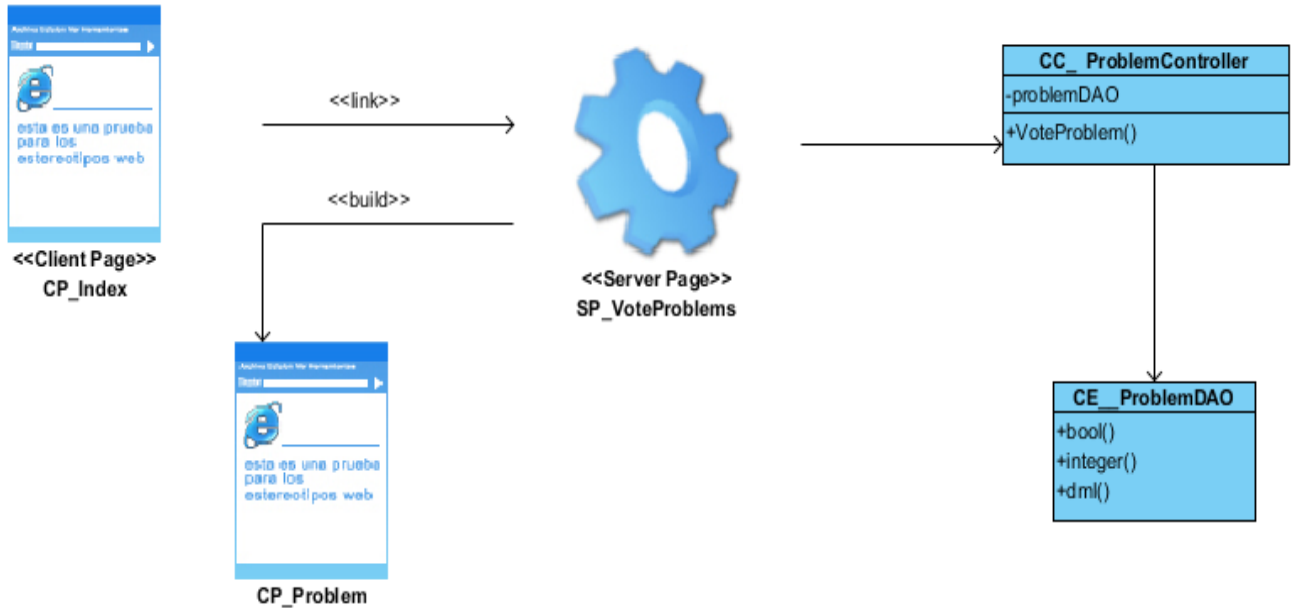


Figura 13. Diagrama de clases de diseño: Valorar problema

2.10 Modelo de datos

El Modelo de datos es el lenguaje orientado a describir la Base de Datos, permite además describir los elementos de la realidad que intervienen en el problema dado y la forma en que se relacionan entre sí.

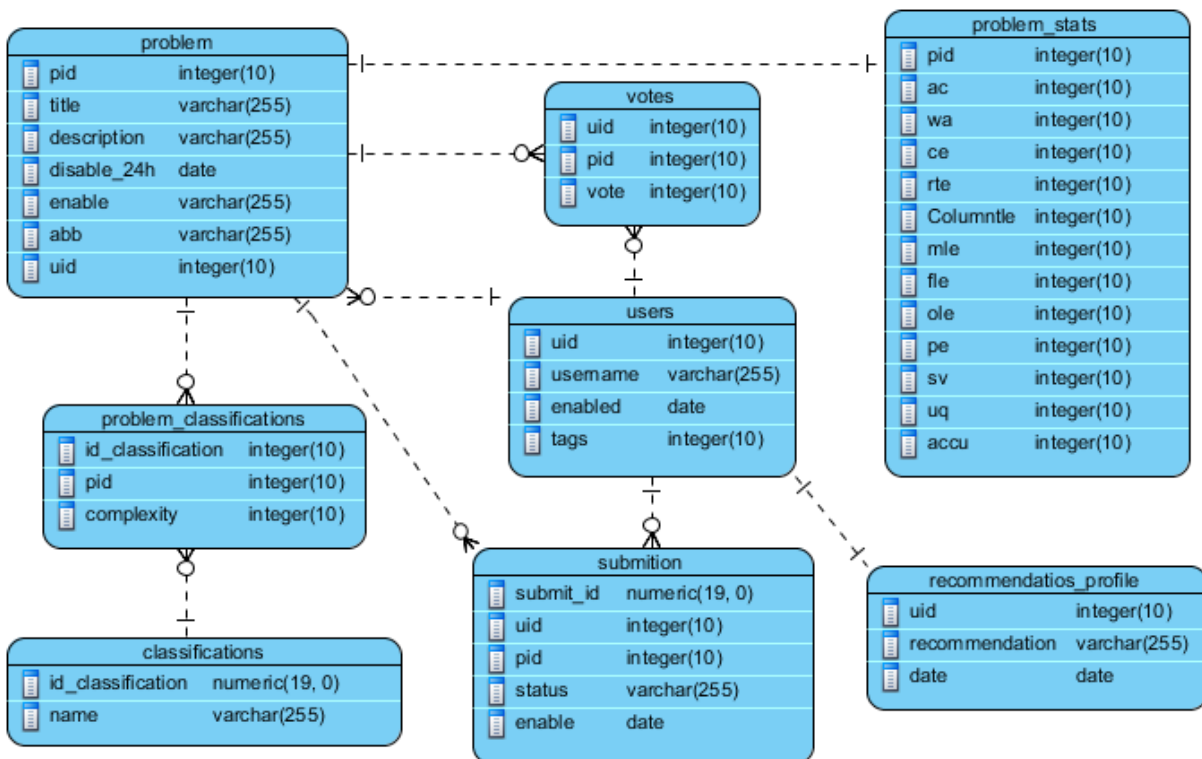


Figura 14. Modelo de datos

2.11 Propuesta de arquitectura

El estilo arquitectónico o el patrón de arquitectura de software empleado para el desarrollo de la aplicación es la arquitectura Modelo Vista Controlador (MVC). Se basa en separar los datos de la aplicación, la interfaz de usuario, y la lógica del negocio en tres capas diferentes, es decir, que agrupa el código según su función. El modelo define el comportamiento, los datos de la aplicación y la lógica del negocio; la base de datos pertenece a esta capa. La vista muestra la información que emplean los usuarios para interactuar con la aplicación. La capa controlador es la que descifra las acciones realizadas por los usuarios, realiza las llamadas al modelo para obtener la información y se las envía a la vista para que las muestre a los usuarios. La vista y el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual (Bascón, 2011).

El framework *Spring* utilizado para el desarrollo de la solución basa su funcionamiento en la implementación del patrón MVC pues propone una estructura basada en modelos (Clases DAO), vistas (Ficheros “.jsp”) y controladores (Ficheros “.java”) que interactúan entre sí del mismo modo que define el patrón. Es seleccionado este estilo por las ventajas que brinda su estructura, además de ser el que utiliza el COJ, plataforma donde será desplegado el sistema de recomendación.

2.12 Patrones de diseño

En ingeniería del software, un patrón es una solución ya probada y aplicable a un problema que se presenta una y otra vez en el desarrollo de distintas aplicaciones y en distintos contextos (*Pattern Definitions*, 2012). Es importante destacar que, en general, un patrón no es una solución en forma de código directamente, sino una descripción de cómo resolver el problema y ante qué circunstancias es aplicable.

En el diseño de la aplicación se hace uso de los patrones **GRASP**; acrónimo de *General Responsibility Assignment Software Patterns*, patrones generales de software para asignar responsabilidades, destacándose los que se mencionan a continuación:

- El patrón **Experto**: Se basa en que la responsabilidad de realizar una tarea es de la clase que posee los datos involucrados; una clase contiene toda la información necesaria para realizar la tarea que tiene encomendada.

Ejemplo: Este patrón se evidencia en las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio.

- El patrón **Creador**: este patrón como su nombre lo indica es el que crea, el que guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un objeto de la clase A.

Ejemplo: Este patrón se evidencia en la clase *Recommender* que es el objeto creado por la clase *RecommenderController*.

- El patrón **Controlador**: Se asocia con operaciones del sistema y respuestas a sus eventos, tal como se relacionan los mensajes y los métodos. El controlador delega en otros objetos el trabajo que se necesita hacer pero coordina o controla la actividad.

Ejemplo: Este patrón se evidencia en la clase *RecommenderController* que es quien delega las funcionalidades a la clase *Recommender*.

- El patrón **Alta Cohesión**: Se manifiesta en la medida en que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Ejemplo: En la solución este patrón es el resultado de asignar responsabilidades únicas a cada uno de los componentes (controladores *RecommenderController*, vistas: *Recommendations* y modelos: *RecommenderDAO*).

- El patrón **Bajo Acoplamiento**: Asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades como las validaciones y la seguridad de la aplicación.

Además se aplican algunos patrones *Gang of Four*(GoF), también conocidos como los patrones de la pandilla de los cuatro, entre los que es válido mencionar **Abstract Factory** que proporciona una interfaz para crear familias de objetos o que dependen entre sí, sin especificar sus clases concretas. Permite que las clases que la usen sean las que decidan qué clase instanciar. Se emplean otros patrones GoF pero están directamente incluidos en la tecnología que se usa en el desarrollo de la aplicación como el framework Spring. También se usa el patrón **Data Access Object** (DAO), este crea una interfaz para encapsular los accesos a las fuentes de datos.

Además se utiliza el patrón Agrupación de Objetos (**Object Pool**), La agrupación de objetos puede ofrecer un significativo aumento del rendimiento, es más eficaz en situaciones donde el costo de la inicialización de una instancia es alto, la tasa de instancia de una clase es alta, y el número de instancias en uso en cualquier momento es bajo. En la implementación del patrón es deseable mantener todos los objetos que no están siendo usados en una misma agrupación para ello la clase de agrupación de objetos es creada usando el patrón *Singleton* (*Object Pool Design Pattern*, 2012). El patrón permite en el sistema la utilización de un *pool* de conexiones a la base de datos para minimizar el costo de creación de nuevas conexiones y poder reutilizar las existentes.

Igualmente, en la aplicación se utiliza otro patrón de diseño que a pesar de no incluirse en ninguna categoría específica, juega un papel clave dentro de las nuevas tendencias del desarrollo de este tipo de proyectos, la **Inyección de dependencias** base del funcionamiento del *Framework Spring*. Especial

atención requiere el uso de dicho patrón en los objetos DAO que se conectan a la base de datos, ya que evita la creación de múltiples conexiones por los diferentes usuarios. Además se utiliza para inyectar en los controladores cada uno de los modelos requeridos para su correcto funcionamiento.

2.13 Conclusiones parciales

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

- Los requisitos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, se utilizaron de guía para desarrollar las funcionalidades y de este modo satisfacer las necesidades detectadas.
- Los artefactos generados según la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.
- El diagrama de despliegue y componentes propuesto permitió exponer de una manera más precisa cómo se estructuran los elementos de hardware y software del sistema y de este modo obtener una mejor comprensión del mismo.

Capítulo 3: Implementación y validación del sistema de recomendación para el Juez en Línea Caribeño

En este capítulo primeramente quedará confeccionado el “Plan de *release*” de la solución, permitiendo al cliente y los desarrolladores trazar una línea imaginaria de tiempo para la obtención del producto. Por otra parte se confeccionarán algunos artefactos, tales como: el diagrama de componentes, el diagrama de despliegue y el diagrama de paquete. Finalmente se diseñarán y ejecutarán las pruebas a la solución obtenida.

3.1 Plan de *release*

De acuerdo a las funcionalidades descritas en las Historias de Usuarios y la prioridad asignada a cada una, se planificaron 2 iteraciones para obtener la solución propuesta.

Tabla 12. Plan de *release*

<i>Release</i>	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	Implementar las HU de máxima prioridad	1-4	11 semanas
2	Implementar las HU de prioridad alta	5	1 semanas

3.2 Diagrama de componentes

El diagrama de componentes modela el empaquetado físico del sistema en unidades reutilizables llamadas componentes y sus relaciones. Un componente es una unidad física de implementación que encapsula una o más clases del diseño. Este diagrama describe la descomposición del software en capas y subsistemas de implementación al igual que sus dependencias (Ambler, 2003b).

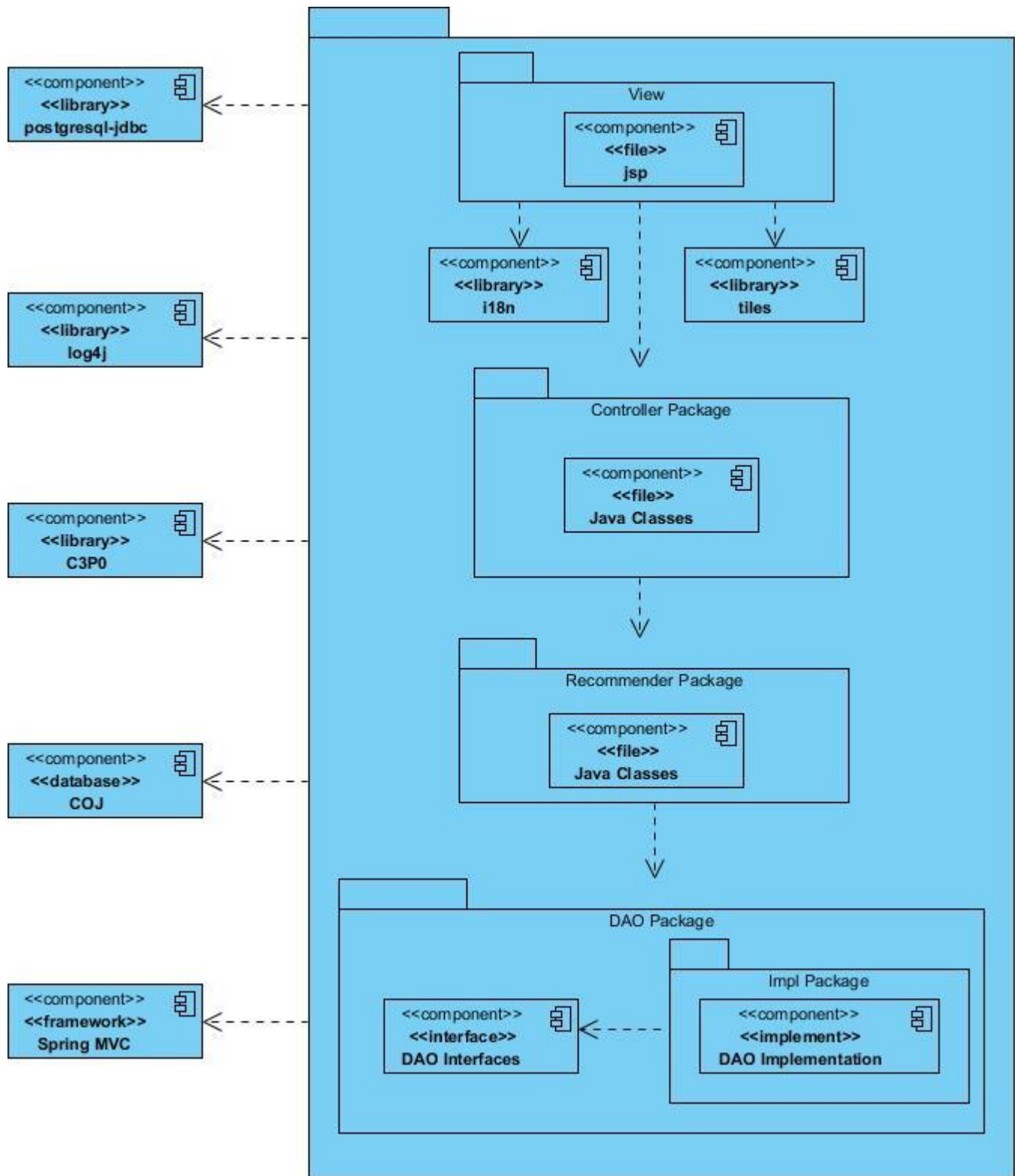


Figura 15. Diagrama de componentes

3.3 Diagrama de paquetes

Permite organizar los elementos modelados con UML, facilitando de ésta forma el manejo de los modelos de un sistema complejo. Además permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales y se pueden utilizar para plantear la arquitectura del sistema a nivel macro. El diagrama muestra como está estructurado el sistema. Cada paquete puede contener otros paquetes o clases, que tienen interfaces y realizan cierta funcionalidad. También se pueden mostrar clases dentro de los paquetes, así como las relaciones de dependencia de estas clases con otras clases o paquetes (Tutorial - UML *Package Diagram*, 2011).

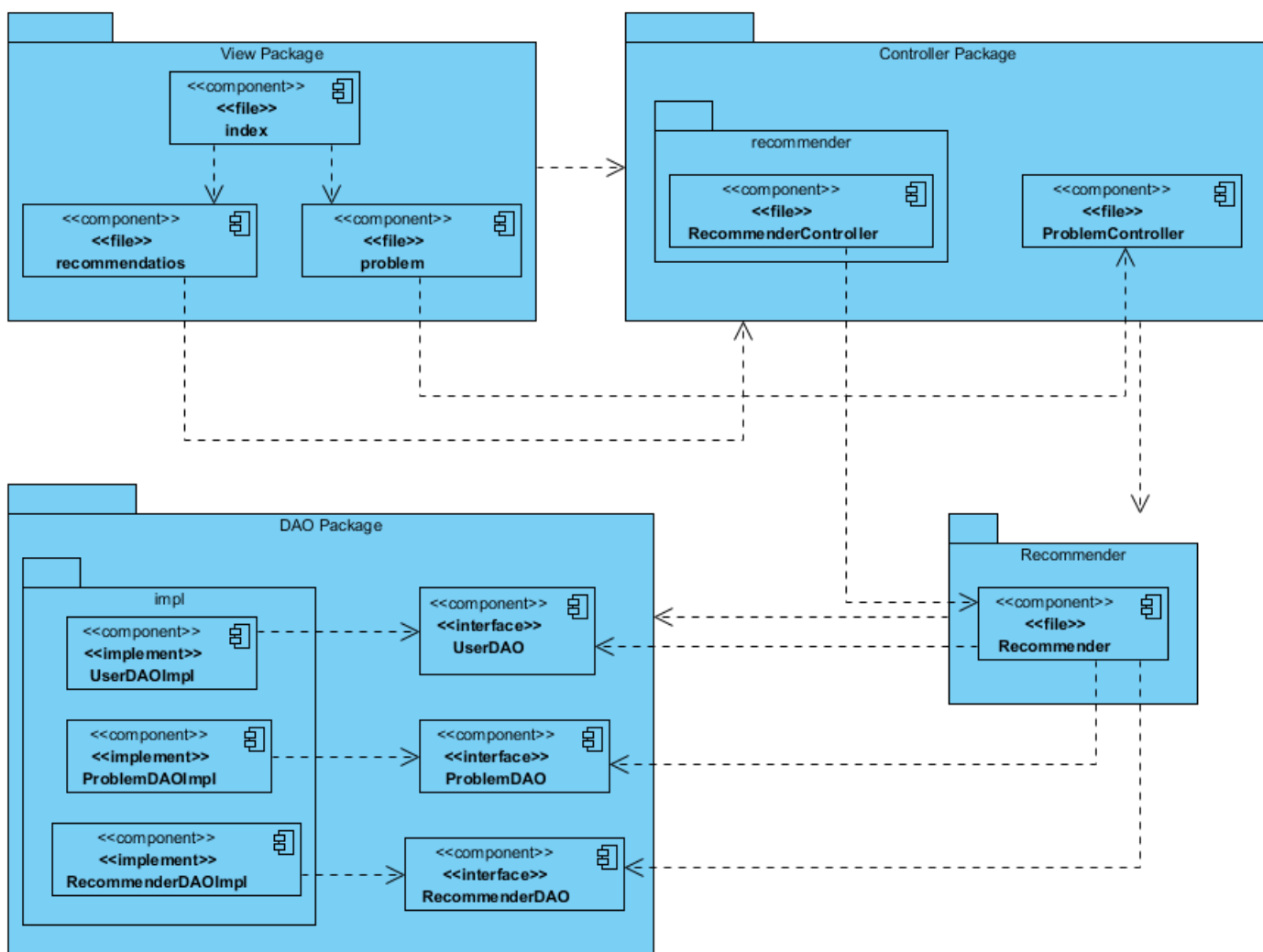


Figura 16. Diagrama de paquetes

3.4 Diagrama de despliegue

El diagrama de despliegue muestra la configuración física sobre la que será desplegado el software. Este presenta los nodos computacionales que intervienen en el funcionamiento del sistema, las conexiones entre estos y los protocolos de comunicación que serán utilizados, estableciendo posibles configuraciones que se ilustran mediante los diagramas de despliegue (Ambler, 2003c).

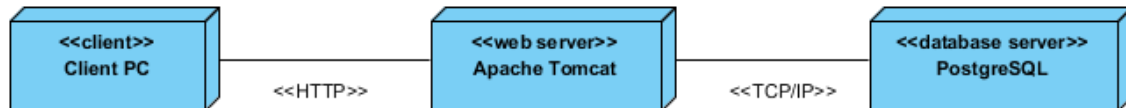


Figura 17. Diagrama de despliegue

3.4.1 Descripción de los componentes

- **Client PC:** PC desde donde se podrá visualizar e interactuar con el juez en línea a través de un navegador web (previamente instalado en la máquina).
- **Apache Tomcat:** Servidor donde se encuentra desplegada la aplicación web, a la que se conectan los clientes por medio de sus estaciones de trabajo.
- **PostgreSQL:** Servidor en el que se encuentran almacenados todos los datos persistentes del sistema.
- **TCP/IP:** Protocolo de comunicación entre el servidor web y el servidor de base de datos se emplea, este provee la transferencia confiable de paquetes de datos e información sobre la red.
- **HTTPS:** Para la conexión al servidor web desde las computadoras clientes, el mismo implementa un canal de comunicación seguro, basado en SSL (*Secure Socket Layers*) entre el navegador del cliente y el servidor HTTP.

3.5 Pruebas

3.5.1 Evaluación de la recomendación

Para esta prueba que se realizó de manera manual se tomó como muestra 5 usuarios de un total de 14354, que representa un 0,03% del total, 20 problemas de 1892 problemas presentes en el sitio, que representa 1,06 % aproximadamente. Para una corrida manual resultaba sumamente difícil elevar las cifras de las muestras seleccionadas. Esta prueba se realizó para simular todo el proceso de recolección de los datos que efectúa el sistema hasta arrojar los resultados, que son las recomendaciones. Para

realizar la prueba se tomó como límite 10 problemas recomendados que se encuentran dentro de las clasificaciones escogidas por el usuario, y dos problemas no incluidos dentro de esas clasificaciones. En caso de que los problemas que se encuentran dentro de sus preferencias no cumplan con esa cantidad se completa con los que no se encuentran dentro de ellas hasta lograr completar 12 problemas si existe esa cantidad. Los problemas son ordenados según la votación que han obtenido. Las recomendaciones obtenidas en la prueba manual concuerdan con las propuestas por el sistema para cada usuario escogido, lo cual significa que el sistema funciona de manera correcta (Anexo 2).

3.5.2 Métricas de validación

- **Precisión**

Usuarios:

- jcfernandez(José Carlos González Fernández)
- jfuentes(Jorge Fuentes Rodríguez)
- Slash(Yaniel Velázquez Bruceta)

Tabla 13. Valores de Precisión

N=5				Promedio
Usuario	jcfernandez	Slash	jfuentes	
Precisión	1.0	0.79	0.7	0.83
N=10				Promedio
Usuario	jcfernandez	Slash	jfuentes	
Precisión	0.8	0.75	0.5	0.68
N=15				Promedio
Usuario	jcfernandez	Slash	jfuentes	
Precisión	0.66666	0.6	0.46666666	0.57
N=20				Promedio
Usuario	jcfernandez	Slash	jfuentes	
Precisión	0.65	0.45	0.35	0.48
N=25				Promedio
Usuario	jcfernandez	Slash	jfuentes	
Precisión	0.52	0.40	0.32	0.41

Para el algoritmo k-vecinos de filtrado colaborativo del sistema de recomendación obtenido, se ejecutó el cálculo de los valores de Precisión para los 3 usuarios punteros del jurado online, considerados los de mejores criterios a la hora de seleccionar los problemas a resolver. Para cada uno de los usuarios se ejecutaron varias corridas en las que se varió, desde 5 hasta 25, el número N de recomendaciones a ser

devueltas por el algoritmo de recomendación, con el objetivo de ver el comportamiento de los dos parámetros que se miden. Los resultados obtenidos muestran la precisión del sistema de presentar al usuario final sólo los elementos relevantes.

- **Error medio absoluto (MAE)**

Promedio 0.29

Tabla 14. Valores del MAE

Usuarios	jcfernandez	jfuentes	Slash	jlicastrillon	Charlie	Night
MAE	0.23	0	0.32	0.38	0.42	0.4

El valor obtenido por el cálculo del MAE (0.32), indica que las estimaciones de las calificaciones (predicciones) son bastante buenas de acuerdo al método de cálculo de similitud utilizado. El valor obtenido se puede disminuir utilizando más cantidad de usuarios como muestra, debido a que la similitud entre cada usuario es menor.

3.5.3 Pruebas de sistema

3.5.3.1 Pruebas de carga

Se definió como servidor de aplicaciones para las pruebas un ordenador con las siguientes prestaciones:

- Microprocesador Intel *Core i5-4200* a 2.3GHz x 4.
- Memoria RAM 4GB.

El COJ tiene como promedio 50 usuarios concurrentes pero aun así se consideraron los siguientes

- 50 hilos (simulación de 50 usuarios) y un período de subida de 1 segundo.
- 100 hilos (simulación de 100 usuarios) y un período de subida de 1 segundo.
- 150 hilos (simulación de 150 usuarios) y un período de subida de 1 segundo

Se identificaron las funcionalidades principales que podrían sobrecargar el sistema como: mostrar la página principal, la página de problemas y mostrar un problema en específico.

Para la primera muestra se identificaron 50 usuarios, 100 usuarios y 150 usuarios que no se encontraran autenticados en el sistema (Anexo 3).

- Para la cantidad de **50 usuarios** no se detecta ningún error, lo cual se deduce en la columna que representa el tanto por ciento de errores para cada petición. En cuanto al rendimiento se puede observar que el servidor es capaz de aceptar un total de 2000 peticiones y por cada petición 22,4 segundos, con tiempo de respuesta máximo de 4,060 segundos (Figura 14).

- Para la cantidad de **100 usuarios** no se detecta ningún error, lo cual se deduce en la columna que representa el tanto por ciento de errores para cada petición. En cuanto al rendimiento se puede observar que el servidor es capaz de aceptar un total de 3800 peticiones y por cada petición 32,4 segundos, con tiempo de respuesta máximo de 13,252 segundos (Figura 15).
- Para la cantidad de **150 usuarios** no se detecta ningún error, lo cual se deduce en la columna que representa el tanto por ciento de errores para cada petición. En cuanto al rendimiento se puede observar que el servidor es capaz de aceptar un total de 5550 peticiones y por cada petición 64,9 segundos, con tiempo de respuesta máximo de 6,706 segundos (Figura 16).

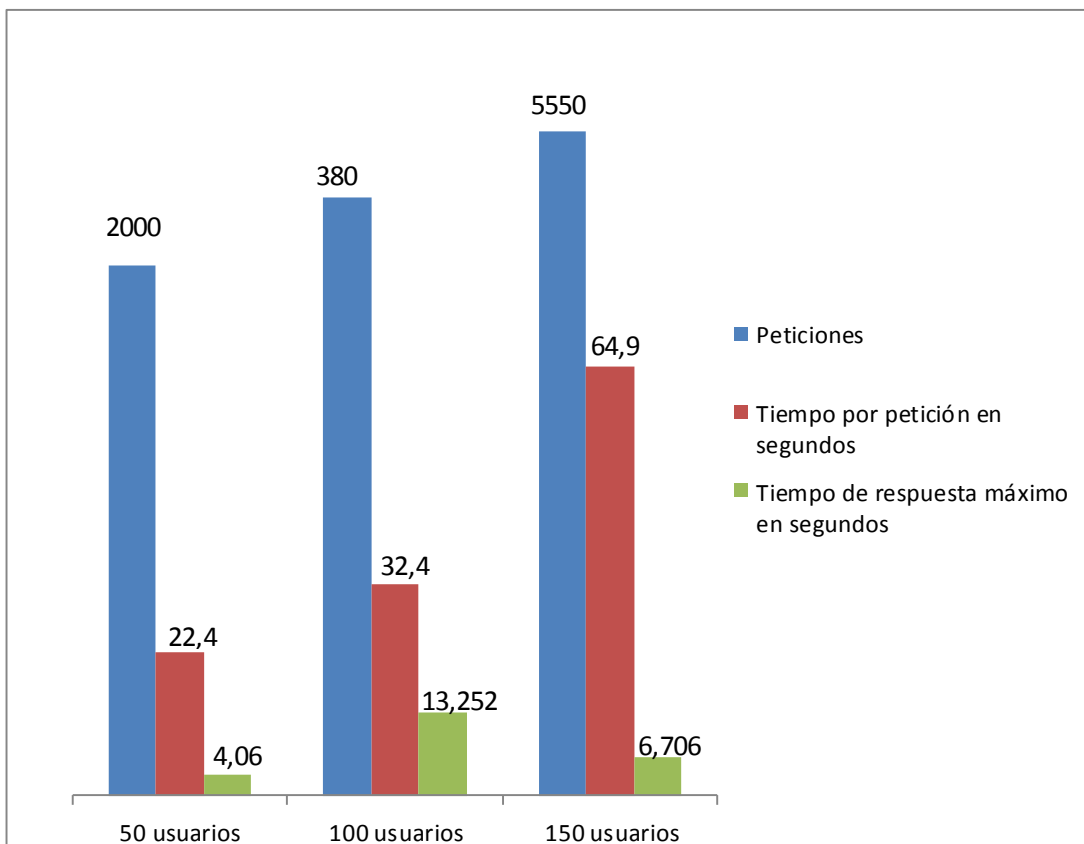


Figura 18. Prueba carga y estrés a usuarios autenticados

Para la segunda muestra se identificaron 50 usuarios, 100 usuarios y 150 usuarios que se encontraran autenticados en el sistema (Anexo 3).

- Para la cantidad de **50 usuarios** no se detecta ningún error, lo cual se deduce en la columna que representa el tanto por ciento de errores para cada petición. En cuanto al rendimiento se puede observar que el servidor es capaz de aceptar un total de 1450 peticiones y por cada petición 49,4 segundos, con tiempo de respuesta máximo de 6,861 segundos (Figura 17).

- Para la cantidad de **100 usuarios** no se detecta ningún error, lo cual se deduce en la columna que representa el tanto por ciento de errores para cada petición. En cuanto al rendimiento se puede observar que el servidor es capaz de aceptar un total de 2900 peticiones y por cada petición 54,7 segundos, con tiempo de respuesta máximo de 6,460 segundos (Figura 18).
- Para la cantidad de **150 usuarios** no se detecta ningún error, lo cual se deduce en la columna que representa el tanto por ciento de errores para cada petición. En cuanto al rendimiento se puede observar que el servidor es capaz de aceptar un total de 4350 peticiones y por cada petición 55,3 segundos, con tiempo de respuesta máximo de 10,350 segundos (Figura 19).

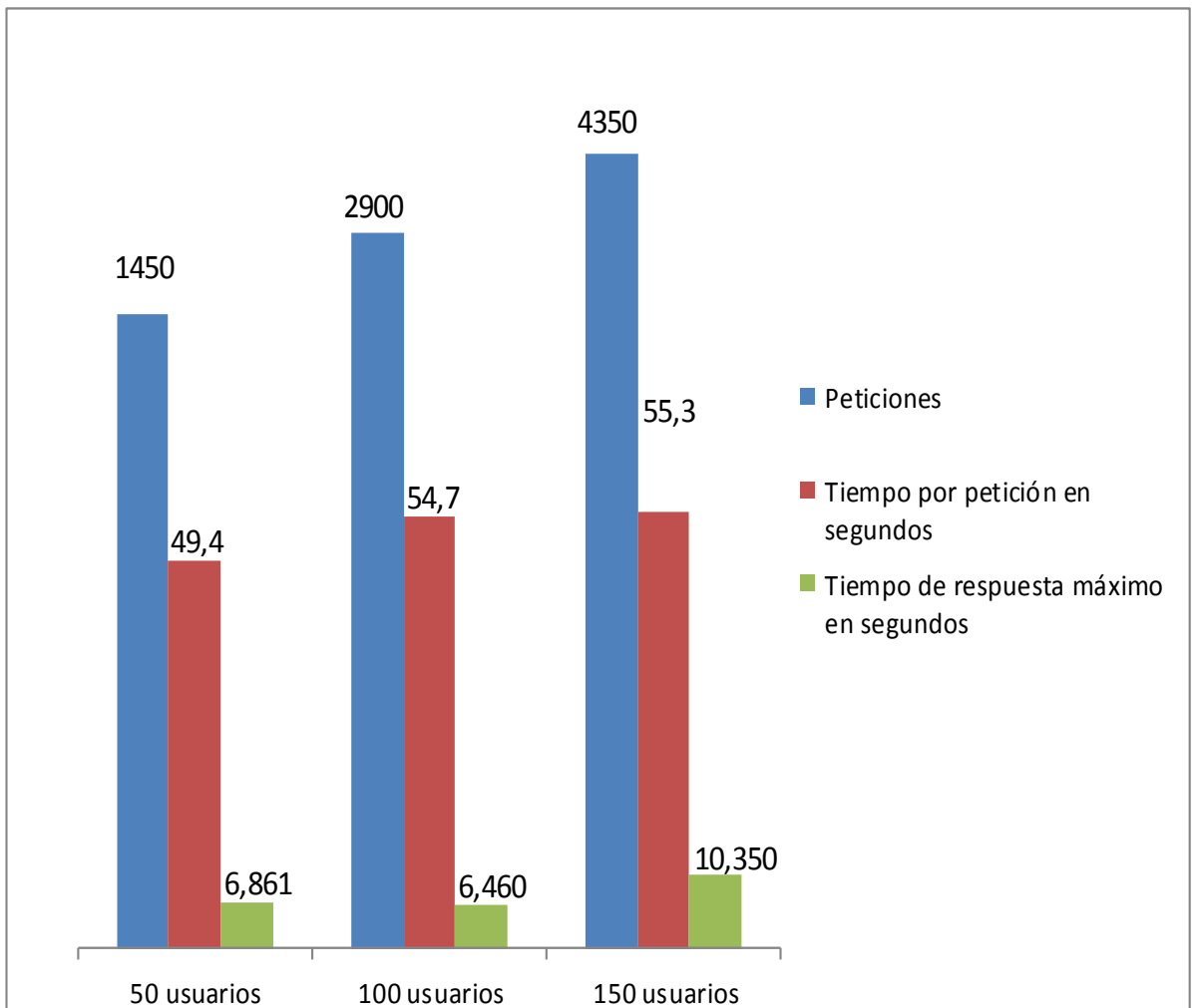


Figura 19. Prueba carga y estrés a usuarios autenticados

En los resultados en algunos casos influyen factores externos como la velocidad de las conexiones al servidor y la concurrencia de usuarios en la red, lo que trae consigo que los resultados arrojados sean variables.

3.5.4 Pruebas de integración

Tabla 15. Caso de prueba de integración No.1

Caso de prueba: Verificar la integración del módulo de Recomendación para la funcionalidad “Mostrar recomendación solicitada”
No. De caso de prueba: 1
Módulo a integrar: Recomendación
Condiciones de ejecución: El módulo de Recomendación haya creado las recomendaciones con sus datos, quedando guardado en la base de datos.
Descripción de la prueba: Comprobar que el módulo de Recomendación pueda obtener los datos de los usuarios, para poder realizar la recomendación.
Entradas/Pasos de ejecución: El módulo de Recomendación crea las recomendaciones, quedando guardadas en la base de datos.
Resultado esperado: Se muestran en la lista recomendaciones.

Tabla 16. Caso de prueba de integración No.2

Caso de prueba: Verificar la integración del módulo de Recomendación para la funcionalidad “Mostrar problemas similares”
No. De caso de prueba: 2
Módulo a integrar: Recomendación
Condiciones de ejecución: El módulo de Recomendación haya creado las recomendaciones con sus datos, quedando guardado en la base de datos.
Descripción de la prueba: Comprobar que el módulo de Recomendación pueda obtener los datos de los problemas, para poder realizar la recomendación.
Entradas/Pasos de ejecución: El módulo de Recomendación crea las recomendaciones, quedando guardadas en la base de datos.
Resultado esperado: Se muestran en la lista recomendaciones de problemas similares.

Los casos de pruebas que se realizaron arrojaron resultados satisfactorios, demostrando que el sistema respondía según lo descrito.

3.5.5 Pruebas de aceptación

Tabla 17. Caso de prueba de aceptación No.1

Caso de prueba de aceptación	
Código de caso de prueba: HU1_CP1	Nombre de la historia de usuario: Mostrar recomendación a usuarios no autenticados.
Responsable de la prueba: Junet Doria Martínez	
Descripción de la prueba: Prueba de funcionalidad para mostrar recomendación a usuarios no autenticados.	
Condiciones de ejecución: Algún usuario debe de solicitar recomendación	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> - Un usuario debe de estar conectado en el sitio sin autenticarse. - El usuario debe solicitar la recomendación. 	
Resultado esperado: Se muestran las recomendaciones al usuario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 18. Caso de prueba de aceptación No.2

Caso de prueba de aceptación	
Código de caso de prueba: HU2_CP2	Nombre de la historia de usuario: Mostrar recomendación a usuarios autenticados.
Responsable de la prueba: José Carlos González Fernández	
Descripción de la prueba: Prueba de funcionalidad para mostrar recomendación a usuarios autenticados.	
Condiciones de ejecución: Algún usuario debe de estar autenticado y solicitar recomendación	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> - Un usuario debe de estar conectado y autenticado en el sitio. - El usuario debe solicitar la recomendación. 	
Resultado esperado: Se muestran las recomendaciones al usuario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 19. Caso de prueba de aceptación No.3

Caso de prueba de aceptación	
Código de caso de prueba: HU3_CP3	Nombre de la historia de usuario: Mostrar recomendación de problemas similares a usuarios autenticados.
Responsable de la prueba: José Carlos González Fernández	
Descripción de la prueba: Prueba de funcionalidad para mostrar recomendación de problemas similares a usuarios autenticados.	
Condiciones de ejecución: Algún usuario debe de estar conectado en el sitio	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> - Un usuario debe de estar conectado en el sitio. - El usuario debe visualizar un problema. 	
Resultado esperado: Se muestran las recomendaciones al usuario debajo de los datos del problema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 20. Caso de prueba de aceptación No.4

Caso de prueba de aceptación	
Código de caso de prueba: HU4_CP4	Nombre de la historia de usuario: Mostrar recomendación de problemas similares a no usuarios autenticados.
Responsable de la prueba: Junet Doria Martínez	
Descripción de la prueba: Prueba de funcionalidad para mostrar recomendación de problemas similares a usuarios no autenticados.	
Condiciones de ejecución: Algún usuario debe de estar conectado y autenticado en el sitio	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> - Un usuario debe de estar conectado y autenticado en el sitio. - El usuario debe visualizar un problema. 	
Resultado esperado: Se muestran las recomendaciones al usuario debajo de los datos del problema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 21. Caso de prueba de aceptación No.5

Caso de prueba de aceptación	
Código de caso de prueba: HU5_CP5	Nombre de la historia de usuario: Valorar problema
Responsable de la prueba: Junet Doria Martínez	
Descripción de la prueba: Prueba de funcionalidad para valorar problema	
Condiciones de ejecución: Algún usuario debe de estar conectado y autenticado en el sitio	
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> - Un usuario debe de estar conectado y autenticado en el sitio. - El usuario debe visualizar un problema. - El usuario realiza la votación al problema. 	
Resultado esperado: Se muestran las votaciones del problema.	
Evaluación de la prueba: Prueba satisfactoria.	

3.6 Conclusiones parciales

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

- El diagrama de componentes definido permite apreciar de forma visual la organización y las dependencias lógicas entre componentes de *software* del sistema.
- El diagrama de paquete permite obtener un mayor entendimiento de los paquetes que intervienen en la propuesta de solución y su relación entre ellos.
- Las pruebas de carga y estrés realizadas arrojaron resultados que cumplían con las expectativas que se tenían, lo cual significa que el tiempo de respuesta del sistema es considerado bueno.

Conclusiones

- La revisión bibliográfica permitió reconocer la existencia de tendencias asociadas a las recomendaciones en los Jueces en Línea estudiados y permitió la fundamentación de las características funcionales que debían ser aprovechadas en el Juez en Línea Caribeño. Entre estas tendencias sobresale la recomendación de problemas para que el usuario mejore su preparación.
- El uso de la metodología SXP permitió aligerar la modelación del sistema, evitar la elaboración de documentación innecesaria, disminuir el tiempo de desarrollo de la solución y mantener una constante retroalimentación con el cliente.
- Se obtuvo un sistema de recomendación híbrido en cascada, que cumple con todas las funcionalidades previstas por el cliente y brinda recomendaciones de problemas al usuario, cercanas a la realidad según las métricas de evaluación aplicadas, sugeridas en la literatura científica.
- Las métricas Precisión y MAE empleadas para validar los algoritmos utilizados en la implementación del sistema de recomendación demostraron la precisión y la calidad de las recomendaciones brindadas a los usuarios.
- Las pruebas de integración validaron el correcto funcionamiento de los requisitos del sistema en cuanto a su interacción con el COJ. Las pruebas de rendimiento posibilitaron demostrar la eficiencia en la respuesta del sistema a los usuarios, y las de aceptación permitieron conocer la satisfacción del cliente con el sistema obtenido.

Recomendaciones

- Optimizar los algoritmos para mejorar las recomendaciones realizadas.
- Emplear nuevas técnicas de recomendación de información.
- Aplicar otras métricas de validación de la solución.

Bibliografía

- Acosta, N.S; Soria, J.A y Reyes, A. Alternativa de comunicación para el Juez en línea Caribeño Cuba: Universidad de las Ciencias Informáticas, 2010. [En línea] [Accedido 5 de diciembre del 2013] Disponible en: <http://publicaciones.uci.cu/index.php/SC/article/view/1202/659>.
- Albín, A.P. Sistema de recomendación colaborativo basado en algoritmos de filtrado mejorados. 2009. [En línea] [Accedido 25 de febrero del 2014] Disponible en: http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc_antonio_pedro.pdf.
- Ambler, S.W. UML 2 Class Diagrams: An Introduction. 2003a. [En línea] [Accedido 9 de abril del 2014] Disponible en: <http://www.agilemodeling.com/artifacts/classDiagram.htm>.
- Ambler, S.W. UML 2 Component Diagrams: An Introduction. 2003b. [En línea] [Accedido 9 de abril del 2014] Disponible en: <http://www.agilemodeling.com/artifacts/componentDiagram.htm>.
- Ambler, S.W. UML 2 Deployment Diagrams: An Introduction. 2003c. [En línea] [Accedido 9 de abril del 2014] Disponible en: <http://www.agilemodeling.com/artifacts/deploymentDiagram.htm>.
- Apache Tiles - Framework - The Composite View Pattern. 2013. [En línea] [Accedido 23 de abril del 2014] Disponible en: <http://tiles.apache.org/framework/tutorial/pattern.html>.
- Bascón, E. El patrón de diseño Modelo Vista Controlador (MVC) y su implementación en Java Swing. 2011. [En línea] [Accedido 23 de abril del 2014] Disponible en: <http://ucbconocimiento.ucbca.edu.bo/index.php/ran/article/download/84/81>.
- Bernot, Y. M. Sistema Gestor de Base de Datos. 2011. [En línea] [Accedido 20 de marzo del 2014] Disponible en: http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos.
- Billsus, D y Pazzani, M.J. Learning Collaborative Information Filtering. 2009. [En línea] [Accedido 2 de junio de 2014] Disponible en: <http://www.aaai.org/Papers/Workshops/1998/WS-98-08/WS98-08-005.pdf>.
- Burke, R. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction. 2002 pag: 331-370.
- Caribbean Online Judge* - EcuRed. 2013. [En línea] [Accedido 9 de diciembre del 2013] Disponible en: http://www.EcuRed.cu/index.php/Caribbean_Online_Judge.

- Carrillo, G. y Ochoa, X. Recomendación de objetos de aprendizaje basado en el perfil del usuario y la información de atención contextualizada. 2012. [En línea] [Accedido 28 de marzo de 2014] Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/18962/6/gcarrilloTFM0113memoria.pdf>.
- Castellano, E.J. Evaluación del Uso de Algoritmos Colaborativos para Orientar Académicamente al Alumnado en Bachillerato. 2007. [En línea] [Accedido 28 de marzo del 2014] Disponible en: http://sinbad2.ujaen.es/cod/archivosPublicos/dea/DEA_EmilioCastellano.pdf.
- Client/Server Architectures. 2012. [En línea] [Accedido 23 de abril del 2014] Disponible en: <http://www.thedacs.com/databases/url/key/216>.
- COJ. Acerca de. 2014. [En línea] [Accedido 5 de enero del 2014] Disponible en: <http://coj.uci.cu/general/about.xhtml>.
- COJ. Enlaces. 2014. [En línea] [Accedido 23 de enero del 2014] Disponible en: <http://coj.uci.cu/general/links.xhtml#links.3>.
- Cristache, A. *Hybrid recommender system using association rules*. Auckland, New Zealand: s.n., 2009.
- Duran, D.F y Arciniega, J.L. Algoritmo de Filtrado Híbrido-Mixto para Recomendación de Contenidos Audiovisuales a Comunidades Virtuales. 2013. [En línea] [Accedido 28 de marzo del 2014] Disponible en: <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/8329/5313>.
- EcuRed. Servidores Web – EcuRed. 2014. [En línea] [Accedido 25 de febrero del 2014] Disponible en: http://www.EcuRed.cu/index.php/Servidores_Web.
- Eguiluz, J. CSS avanzado. 2014. [En línea] [Accedido 25 de febrero del 2014] Disponible en http://librosweb.es/css_avanzado/capitulo_5.html.
- García, J. ¿Qué es el Aprendizaje Colaborativo? | Mi pequeño mundo. 2009. [En línea] [Accedido 2 de diciembre del 2013] Disponible en: <http://jgarcia.wordpress.com/2006/10/29/%C2%BFaprendizaje-colaborativo/>.
- Gómez, A.C. XHTML. 2011. [En línea] [Accedido 20 de marzo del 2014] Disponible en: <http://www.ecured.cu/index.php/XHTML>.
- Griffin, C. Design Patterns: Dependency Injection.2012. [En línea] [Accedido 23 de abril de 2014] Disponible en: <http://msdn.microsoft.com/en-us/magazine/cc163739.aspx>.

- Grupo AIA. Soluciones de recomendación para nuevos modelos de negocio. 2013. [En línea] [Accedido 23 de enero del 2014] Disponible en: <http://www.aia.es/recommendation-systems/>.
- Gutiérrez, J.J. Framework. 2014. [En línea] [Accedido 25 de febrero del 2014] Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
- Gutiérrez, D. UML Diagrama de paquetes. 2009. [En línea] [Accedido 7 de mayo del 2014] Disponible en http://www.codecompiling.net/files/slides/UML_clase_05_UML_paquetes.pdf.
- Hechavarria, R. G. CSS. 2011. [En línea] [Accedido 20 de marzo del 2014] Disponible en: <http://www.ecured.cu/index.php/CSS>.
- Herlocker, J; Konsta, J.A; Terveen, L.G y Riedl, J.T. *Evaluating Collaborative Filtering Recommender Systems*. 2004. [En línea] [Accedido 3 de abril del 2014] Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.5270&rep=rep1&type=pdf>.
- Herrera-Viedma, E; Porcel. C e Hidalgo. L. Sistemas de recomendaciones: herramientas para el filtrado de información en Internet - Hipertext - (UPF).2012. [En línea] [Accedido 9 de diciembre del 2013] Disponible en: <http://www.upf.edu/hipertextnet/numero-2/recomendacion.html>.
- Junco,T.O. Un juez en línea ajustado a las necesidades de la docencia. 2012. [En línea] [Accedido 1 de abril del 2014] Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/7993/1/TM_06245%20_12.pdf.
- Krall, C. ¿Qué es y para qué sirve UML? Versiones de UML (Lenguaje Unificado de Modelado). Tipos de diagramas UML. 2014. [En línea] [Accedido 25 de febrero del 2014] Disponible en: http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:ique-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46:lenguajes-y-entornos&Itemid=163.
- Lobaina, J.C, Roque .J.L. Desarrollo de la versión 2 del Juez en línea Caribeño. 2012. [En línea] [Accedido 5 de diciembre del 2013] Disponible en: http://bibliodoc.uci.cu/RDigitales/2012/octubre/31/TD_05129_12.pdf.
- Mamunur, R; Istvan, A; Dan, C; Shyong, K. L; Sean, M. M; Joseph, A. K y John, R, Getting to know you: learning new users preferences in recommender systems. 2002. [En línea] [Accedido 2 de junio del 2014] Disponible en: <http://www.cs.cornell.edu/~danco/research/papers/getting-to-know.pdf>.

- Mo, R. Jurado Online de Programación. 2009. [En línea] [Accedido 2 de diciembre del 2013] Disponible en: <http://elcodigoweb.wordpress.com/2009/09/29/juarado-online-de-programacion/#more-12>.
- Mora, B; Ruiz, F; García, F y Piattini, F. Experiencia en transformación de modelos de procesos de negocios desde BPMN a XPD. 2009. [En línea] [Accedido 7 de mayo del 2014] Disponible en: http://www.researchgate.net/profile/Felix_Garcia2/publication/221561481_Experiencia_en_Transformacion_de_Modelos_de_Procesos_de_Negocios_desde_BPMN_a_XPD/file/d912f50c71bd937e3d.pdf.
- Martín, M. Un sistema de recomendación para sitios de ocio de Andalucía. 2009. [En línea] [Accedido 25 de febrero del 2014] Disponible en: <http://draxus.org/downloads/pfc/presentacion.pdf>.
- Menéndez-Barzanallana, R. Métodos y técnicas didácticas para la enseñanza de la informática. Máster Universitario en Formación del Profesorado de Educación Secundaria Obligatoria y Bachillerato, Formación Profesional, Enseñanzas de Idiomas y Enseñanzas Artísticas. Universidad de Murcia. 2013. [En línea] [Accedido 5 de diciembre del 2013] Disponible en: <http://www.um.es/docencia/barzana/MASTER-INFORMATICA-II/Metodos-y-tecnicas-didacticas-para-la-ensenanza-de-la-informatica.html>.
- Menéndez-Barzanallana, R. Servidores de bases de datos. 2012. [En línea] [Accedido 12 de marzo del 2014] Disponible en: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sqbd.html>.
- Model-View-Controller.2012. [En línea] [Accedido 23 de abril del 2014] Disponible en <http://ootips.org/mvc-pattern.html>.
- Monroy, B. Aprendizaje colaborativo. 2008. [En línea] [Accedido 2 de diciembre del 2013] Disponible en: <http://www.slideshare.net/lexlator/aprendizaje-colaborativo-4581012>.
- Moya R e Invarato, R. ¿Que son los sistemas de recomendación? : Jarroba. 2014a. [En línea] [Accedido 25 de febrero del 2014] Disponible en: <http://jarroba.com/que-son-los-sistemas-de-recomendacion/>.
- Moya, R e Invarato,R. Sistemas de recomendación basados en Filtrado Colaborativo (K-Vecinos): Jarroba. 2014b. [En línea] [Accedido 25 de febrero del 2014] Disponible en: <http://jarroba.com/sistemas-de-recomendacion-basados-en-filtrado-colaborativo-k-vecinos/>.
- Moya, R. SVD aplicado a sistemas de recomendación basados en filtrado colaborativo. 2013. [En línea] [Accedido 9 de abril del 2014] Disponible en: <http://informatico.ricardomoya.es/proyectoMaster.html>.
- Object Pool Design Pattern. 2012. [Accedido 23 de abril del 2014] Disponible en: http://sourcemaking.com/design_patterns/object_pool.

Pattern Definitions. 2012. [En línea]. [Accedido 23 de abril del 2014] Disponible en: <http://st-www.cs.illinois.edu/patterns/definition.html>.

Peis, E; Morales-del-Castillo, J. M y Delgado-López, J. A. Sistemas de recomendación Semánticos. Un análisis del estado de la cuestión - Hipertexto - (UPF). 2012. [En línea] [Accedido 25 de febrero del 2014] Disponible en: <http://www.upf.edu/hipertextnet/numero-6/recomendacion.html#Criterios>.

Peñalver, G. M. MA-GMPUR2 Metodología ágil para proyectos de software libre. 2008. [En línea] [Accedido 1 de abril de 2014] Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_1309_08/1/TD_1309_08.pdf.

Postgresql.org. *Documentation*. 2009. [En línea] [Accedido 20 de marzo del 2014] Disponible en: <http://www.postgresql.org/docs>.

Pressman, R.S. 2002. *Ingeniería del Software Un enfoque práctico*: Quinta edición. 2001. [En línea] [Accedido 14 de mayo, 2014] Disponible en: <http://unpocodejava.wordpress.com/2013/05/09/tecnicas-para-la-captura-de-requisitos/>.

Pressman, R.S. *Ingeniería del Software Un enfoque práctico*: Sexta edición. 2005. [En línea] [Accedido 14 de mayo, 2014] Disponible en: <http://unpocodejava.wordpress.com/2013/05/09/tecnicas-para-la-captura-de-requisitos/>.

Pullés, Y. D. *Herramienta CASE*. 2003. [En línea] [Accedido 20 de marzo del 2014] Disponible en: <http://www.sisman.utm.edu.ec/libros/FACULTAD%20DE%20CIENCIAS%20INFORM%20C3%81TICAS/CARRERA%20DE%20INGENIER%20C3%8DA%20DE%20SISTEMAS%20INFORMATICOS/09/TECNICAS%20DE%20IV%20GENERACION/Libro.pdf>.

Ricci, F; Rokach, L; Shapira, B y Kantor, P.B. *Recommender Systems Handbook*. 2010.

Ripoll, D.A. La UCI: novena en ranking del Juez en línea Polaco | Portal de la Universidad de las Ciencias Informáticas. 2012. [En línea] [Accedido 9 de diciembre del 2013] Disponible en: <http://www.uci.cu/la-uci-novena-en-ranking-del-juez-en-linea-polaco>.

Ros-Martín, M. Monitorización y evaluación en medios sociales filtrado de la información en la Web: basado en contenido, colaborativo y social. 2009. [En línea] [Accedido 9 de diciembre del 2013] Disponible en: <http://comunidad20.sedic.es/?p=160>.

Sierra, M. Qué es un servidor y cuáles son los principales tipos de servidores. 2010. [En línea] [Accedido 12 de marzo del 2014] [Disponible en

http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftpsmtpt&catid=57:herramientas-informaticas&Itemid=179].

Silveira-Romero, D y Hernández-Legrá, L. Convocatoria de participación. 2012. [En línea] [Accedido 20 de noviembre del 2013] Disponible en: <http://www.instec.cu/acm-icpc/>.

Sobecki, J. Implementations of Web-based Recommender Systems Using Hybrid Methods. 2006. International Journal of Computer Science & Applications.

Sphere Research Business Solutions. 2013. [En línea] [Accedido 23 de enero del 2014]. Disponible en: <http://sphere-research.com/en/>.

Tutorial - UML Package Diagram. 2011. [En línea] [Accedido 9 de abril de 2014] Disponible en: <http://www.visual-paradigm.com/product/vpuml/tutorials/packageDiagram.jsp>.

UML, BPMN and *Enterprise Architecture Tool for Software Development*. 2014. [En línea] [Accedido 25 de febrero del 2014] Disponible en: <http://www.visual-paradigm.com/>.

Universidad Federal de los Urales. ICPC FACTSHEET. 2014. [En línea] [Accedido 20 de noviembre del 2013] Disponible en: <http://icpc.baylor.edu/download/worldfinals/pdf/Factsheet.pdf>.

Vaillant, M y Labaut, R. Portal web de la subsede cubana de la final caribeña del ACM-ICPC. 2013. [En línea] [Accedido 23 de enero del 2014] Disponible en: http://bibliodoc.uci.cu/RDigitales/2013/septiembre/24/TD_06531_13.pdf.

Viltres, I y Matos, A. Sistema de apoyo al proceso de enseñanza-aprendizaje desde juegos didácticos. 2010. [En línea] [Accedido 5 de diciembre del 2013] Disponible en: <http://publicaciones.uci.cu/index.php/SC/article/download/356/318>.

World Finals Rules. 2014. [En línea] [Accedido 23 de enero del 2014] Disponible en: <http://icpc.baylor.edu/worldfinals/rules>.

Yera, R. Concepción y desarrollo de un sistema de recomendación para jurados *online* de programación. 2010. [En línea] [Accedido 22 de enero del 2014] Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_02825_10/1/TD_02825_10.pdf.

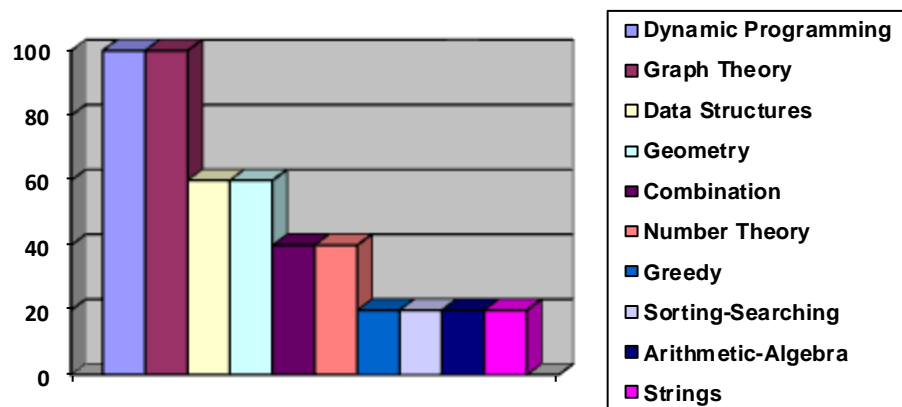
Anexo1: Encuesta realizada a los usuarios del COJ.

La encuesta persigue el objetivo de obtener sus criterios acerca de los contenidos necesarios para la realización del sistema de recomendación.

- ¿Cuáles son las áreas de conocimiento en las que se siente más a gusto?
- ¿Cuáles son los lenguajes de programación que más utiliza en las competencias?
- ¿Cuál es su ranking en el COJ?

Se les realizó la encuesta a 5 usuarios del COJ donde fueron respondidas satisfactoriamente todas las preguntas necesarias.

1. El estudiante Nelson González Peñate registrado con el usuario npenate y ubicado en el *ranking* 18 prefiere problemas que se encuentren en las siguientes áreas de conocimiento: *Combination, Data Structures, Dynamic Programming, Graph Theory*.
2. El estudiante Eddy Roberto Morales Pérez registrado con el usuario *Night* y ubicado en el *ranking* 7 prefiere problemas que se encuentren en las siguientes áreas de conocimiento: *Data Structures, Dynamic Programming, Geometry, Graph Theory*.
3. El estudiante José Luis Castrillón Garrido registrado con el usuario jlcastrillon y ubicado en el *ranking* 69 prefiere problemas que se encuentren en las siguientes áreas de conocimiento: *Dynamic Programming, Geometry, Graph Theory, Greedy, Number Theory, Strings*.
4. El estudiante Leandro Hernández Miro registrado con el usuario Keyfu y ubicado en el *ranking* 40 prefiere problemas que se encuentren en las siguientes áreas de conocimiento: *Dynamic Programming, Graph Theory, Greedy, Sorting-Searching*.
5. El estudiante Jorge Bárbaro Piñeiro Cruz registrado con el usuario jbpineiro y ubicado en el *ranking* 43 prefiere problemas que se encuentren en las siguientes áreas de conocimiento: *Arithmetic-Algebra, Combination, Data Structures, Dynamic Programming, Geometry, Graph Theory, Number Theory*.



Anexo2: Evaluación de la recomendación.

Tabla 22. Muestra escogida para la evaluación de la recomendación

Usuario	Clasificaciones de preferencias del usuario	Clasificación del problema resuelto	Identificador del problema resuelto.	Votación del problema
jcfernandez	4-5-23-24	1	1377	★
		2	1300	★
			1374	
			1417	
		3	1376	
			1372	
		9	1364	★
			1370	
		10	1394	★
23	1368	★★		
ymondelo20	4-5-23-24	1	1377	★
			1627	★★★
		3	1372	
		5	1917	★★★★
			1624	★★★★★★
Night	4-5-23-24	0	2187	★★★★
		1	1627	★★★★
		3	1372	
			2194	★★★★
		5	1917	★★★★
			1624	★★★★★★
			2325	★★★★★★
7	1487	★★★★★★		
Charlie	4-5-23-24			
jlcastrillon	4-5-23-24			

Tabla 23. Recomendaciones obtenidas

Usuario	Problemas con clasificaciones de preferencia			Problemas con clasificaciones que no son de preferencia		
	Clasificación del problema recomendado	Identificador del problema recomendado	Votación del problema	Clasificación del problema recomendado	Identificador del problema recomendado	Votación del problema
jcfernandez	5	1624	☆☆☆☆☆	7	1487	☆☆☆☆☆
		2325	☆☆☆☆☆	1	1627	☆☆☆☆☆
		1917	☆☆☆	3	2194	☆☆☆☆
				0	2187	☆☆☆☆
ymondelo20	5	2325	☆☆☆☆☆	7	1487	☆☆☆☆☆
	23	1368	☆☆	0	2187	☆☆☆☆
				3	2194	☆☆☆☆
				2	1300	☆☆
				9	1364	☆☆
				10	1394	☆☆
				2	1374	
				2	1417	
				9	1370	
				3	1376	
Night	23	1368	☆☆	2	1300	☆☆
				9	1364	☆☆
				10	1394	☆☆
				1	1377	☆☆
				9	1370	
				2	1374	
				3	1376	
				2	1417	
Charlie	5	1624	☆☆☆☆☆	7	1487	☆☆☆☆☆
		2325	☆☆☆☆☆	1	1627	☆☆☆☆☆
		1917	☆☆☆	3	2194	☆☆☆☆
		1368	☆☆	0	2187	☆☆☆☆
				2	1300	☆☆
				9	1364	☆☆

				10	1394	☆
				2	1374	
					1417	
				9	1370	
				3	1376	
				1	1377	☆
jlcatrillon	5	1624	☆☆☆☆☆	7	1487	☆☆☆☆☆
		2325	☆☆☆☆☆	1	1627	☆☆☆☆☆
		1917	☆☆☆	3	2194	☆☆☆☆
		1368	☆☆	0	2187	☆☆☆☆
				2	1300	☆
				9	1364	☆
				10	1394	☆
				2	1374	
					1417	
				9	1370	
				3	1376	
				1	1377	☆

Anexo3: Pruebas de carga y estrés realizadas al sistema.

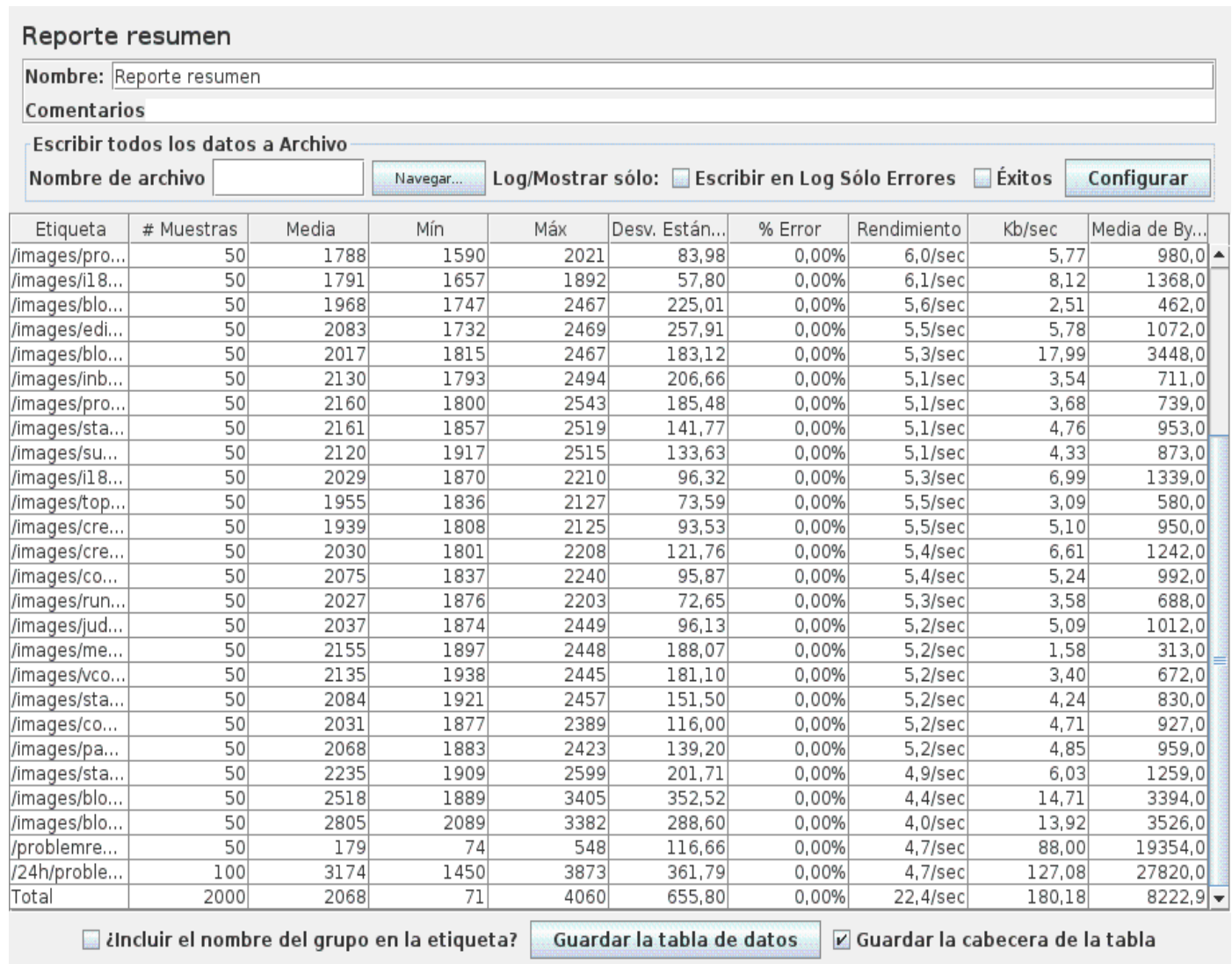


Figura 20. Reporte resumen 50 usuarios autenticados

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo:

Escribir en Log Sólo Errores Éxitos

Configurar

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de B...
/images/blockco...	100	2627	2051	2982	159,84	0,00%	5,6/sec	2,53	462,0
/images/blockhe...	100	2559	2305	3005	175,94	0,00%	5,5/sec	18,46	3440,0
/images/top.gif	100	2696	2304	3005	231,72	0,00%	5,4/sec	3,07	580,0
/images/staroff.p...	100	2731	2345	3018	211,77	0,00%	5,5/sec	53,66	9999,0
/images/compar...	100	2669	2334	3008	183,99	0,00%	5,5/sec	5,33	992,0
/images/standin...	100	2724	2268	3006	188,12	0,00%	5,4/sec	5,06	953,0
/images/blockhe...	100	2759	2404	3007	176,22	0,00%	5,3/sec	18,01	3448,0
/images/logout.p...	100	2769	2334	3025	177,01	0,00%	5,3/sec	4,81	927,0
/images/submit....	100	2773	2269	3039	151,96	0,00%	5,3/sec	4,51	873,0
/images/profile.p...	100	2790	2416	3008	143,92	0,00%	5,1/sec	4,92	980,0
/images/vcontes...	100	2782	2335	3007	147,33	0,00%	5,1/sec	3,37	672,0
/images/menuse...	100	2721	2311	3007	142,13	0,00%	5,2/sec	1,58	313,0
/images/judgme...	100	2707	2270	3007	154,63	0,00%	5,2/sec	5,14	1012,0
/images/arrow_of...	100	2682	2367	3006	146,93	0,00%	5,2/sec	19,63	3844,0
/images/create.gif	100	2628	2368	3005	156,78	0,00%	5,2/sec	6,32	1242,0
/images/coming....	100	2780	2399	3004	159,92	0,00%	5,2/sec	4,72	927,0
/images/past.png	100	2848	2282	3029	177,67	0,00%	5,2/sec	4,90	959,0
/images/blockhe...	100	2791	2202	3039	218,79	0,00%	5,3/sec	17,55	3394,0
/images/stats.png	100	2675	1102	3007	273,15	0,00%	5,2/sec	4,25	830,0
/images/blockhe...	100	2754	2134	3034	245,16	0,00%	5,1/sec	17,55	3526,0
/images/created....	100	2865	2142	3396	199,37	0,00%	5,0/sec	4,67	950,0
/images/problem...	100	2833	1095	3322	400,26	0,00%	5,1/sec	3,66	739,0
/images/running....	100	2542	1026	3389	697,20	0,00%	5,5/sec	3,67	688,0
/images/inbox.png	100	2216	344	3208	823,25	0,00%	6,2/sec	4,34	711,0
/problemrec/rec...	100	5441	98	10562	2988,88	0,00%	4,8/sec	86,84	18348,6
/24h/problem.xht...	200	4383	1246	13252	2016,24	0,00%	6,3/sec	120,59	19615,0
Total	3800	2921	77	13252	1084,18	0,00%	32,4/sec	172,99	5462,4

¿Incluir el nombre del grupo en la etiqueta?

Guardar la tabla de datos

Guardar la cabecera de la tabla

Figura 21. Reporte resumen 100 usuarios autenticados

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Configurar

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de By...
/images/top...	150	1829	1010	3080	899,92	0,00%	8,1/sec	4,59	580,0
/images/pro...	150	2018	1009	3089	912,25	0,00%	8,3/sec	7,91	980,0
/images/blo...	150	2045	1004	3113	913,83	0,00%	8,2/sec	27,60	3440,0
/images/blo...	150	2015	1006	3163	925,79	0,00%	8,5/sec	28,72	3448,0
/images/inb...	150	2046	1010	3086	934,49	0,00%	7,7/sec	5,32	711,0
/images/edi...	150	1922	1008	3160	937,30	0,00%	7,2/sec	7,51	1072,0
/images/blo...	150	1689	1009	3255	859,01	0,00%	7,2/sec	3,26	462,0
/images/pro...	150	1821	1010	3260	925,14	0,00%	7,2/sec	5,22	739,0
/images/log...	150	1706	1003	3168	889,64	0,00%	7,3/sec	6,59	927,0
/images/sta...	150	1726	1013	3365	879,95	0,00%	6,8/sec	6,33	953,0
/images/jud...	150	1843	1006	3405	895,71	0,00%	6,8/sec	6,69	1012,1
/images/co...	150	1879	1008	3197	878,79	0,00%	7,3/sec	7,05	992,0
/images/blo...	150	1899	1013	3127	903,12	0,00%	7,3/sec	24,08	3394,0
/images/su...	150	2018	1009	3334	911,36	0,00%	7,4/sec	6,30	873,0
/images/co...	150	1896	1009	3366	908,35	0,00%	7,4/sec	6,68	927,0
/images/sta...	150	2136	1003	3301	913,42	0,00%	7,0/sec	5,68	830,0
/images/pa...	150	2005	1002	3346	907,64	0,00%	6,9/sec	6,51	959,0
/images/vco...	150	1913	1016	3304	927,00	0,00%	7,0/sec	4,56	672,0
/images/run...	150	1913	1006	3378	918,88	0,00%	7,2/sec	4,84	688,0
/images/cre...	150	1920	1015	3426	913,15	0,00%	7,1/sec	6,58	950,0
/images/arr...	150	1907	1006	3394	872,48	0,00%	7,5/sec	28,02	3844,0
/images/me...	150	1884	1006	3394	845,40	0,00%	7,8/sec	2,39	313,0
/images/blo...	150	1713	1020	3394	761,78	0,00%	7,8/sec	26,73	3526,0
/images/cre...	150	1635	1019	3447	653,76	0,00%	7,3/sec	8,83	1242,0
/problemre...	150	1009	92	3290	888,79	0,00%	7,3/sec	137,34	19354,0
/24h/proble...	150	2662	1075	5301	1150,93	0,00%	6,5/sec	127,51	20230,0
Total	5550	2045	92	6706	1084,60	0,00%	64,9/sec	324,93	5129,4

¿Incluir el nombre del grupo en la etiqueta?

Guardar la tabla de datos

Guardar la cabecera de la tabla

Figura 22. Reporte resumen 150 usuarios autenticados

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Configurar

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de By...
/css/style.css	50	598	83	1672	329,89	0,00%	8,2/sec	156,09	19462,0
/css/datata...	50	547	86	1404	323,86	0,00%	8,3/sec	67,51	8344,0
/images/i18...	50	542	81	2118	466,11	0,00%	8,7/sec	11,56	1368,0
/js/script.js	50	431	107	1409	290,04	0,00%	8,2/sec	97,97	12291,0
/images/blo...	50	404	82	980	199,19	0,00%	8,8/sec	32,77	3831,0
/images/top...	50	345	62	1369	218,13	0,00%	10,0/sec	5,69	580,0
/images/for...	50	343	107	1034	158,26	0,00%	10,1/sec	10,38	1053,0
/images/reg...	50	310	100	534	128,14	0,00%	12,1/sec	11,62	985,0
/images/i18...	50	318	89	762	160,11	0,00%	12,6/sec	16,49	1339,0
/images/blo...	50	319	71	810	162,60	0,00%	12,7/sec	42,21	3394,0
/images/vco...	50	333	78	598	133,27	0,00%	13,4/sec	8,79	672,0
/images/me...	50	336	71	658	132,03	0,00%	14,6/sec	4,45	313,0
/images/jud...	50	303	64	709	148,58	0,00%	14,1/sec	13,95	1012,0
/images/blo...	50	291	62	677	149,51	0,00%	13,5/sec	46,54	3526,0
/images/co...	50	328	115	1141	180,96	0,00%	10,8/sec	9,73	927,0
/images/sta...	50	369	84	1089	177,26	0,00%	10,6/sec	8,62	830,0
/images/pa...	50	355	79	1146	243,48	0,00%	9,7/sec	9,09	959,0
/images/co...	50	412	86	1145	235,22	0,00%	9,5/sec	9,23	992,0
/images/blo...	50	466	59	2684	416,12	0,00%	6,6/sec	2,96	462,0
/images/sta...	50	592	93	3489	648,46	0,00%	6,3/sec	5,87	953,0
/images/run...	50	521	61	2657	388,03	0,00%	6,1/sec	4,08	688,0
/images/pro...	50	646	64	2753	664,78	0,00%	6,0/sec	4,32	739,0
/images/blo...	50	877	114	3178	925,20	0,00%	5,8/sec	19,69	3448,0
/problemre...	50	1367	266	2921	637,00	0,00%	4,9/sec	93,46	19354,0
/images/arr...	50	2153	287	4392	1154,58	0,00%	3,8/sec	14,13	3844,0
/24h/proble...	50	1088	55	2092	583,17	0,00%	4,0/sec	75,84	19575,0
Total	1450	761	55	6861	950,13	0,00%	49,4/sec	243,76	5055,5

¿Incluir el nombre del grupo en la etiqueta?

Guardar la tabla de datos

Guardar la cabecera de la tabla

Figura 23. Reporte resumen 50 usuarios no autenticados

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Configurar

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de By...
/css/style.css	100	1404	540	3483	565,72	0,00%	8,0/sec	152,19	19462,0
/js/script.js	100	1221	589	2462	470,19	0,00%	7,8/sec	93,82	12291,0
/css/main.css	100	1050	545	2497	381,48	0,00%	7,7/sec	37,86	5045,0
/images/i18...	100	923	516	2326	343,55	0,00%	8,5/sec	11,06	1339,0
/images/i18...	100	862	579	1977	232,84	0,00%	8,9/sec	11,93	1368,0
/images/for...	100	900	541	1718	303,61	0,00%	9,7/sec	10,01	1053,0
/images/blo...	100	1034	562	1732	318,06	0,00%	10,3/sec	38,64	3831,0
/images/top...	100	974	585	2084	277,62	0,00%	10,2/sec	5,80	580,0
/images/reg...	100	920	530	1629	244,02	0,00%	11,5/sec	11,09	985,0
/images/co...	100	935	616	1584	246,16	0,00%	11,4/sec	10,31	927,0
/images/pro...	100	986	573	1620	258,31	0,00%	11,5/sec	8,27	739,0
/images/co...	100	960	173	1707	262,71	0,00%	11,1/sec	10,78	992,0
/images/blo...	100	1056	519	1557	240,33	0,00%	10,8/sec	36,44	3448,0
/images/me...	100	1004	619	2135	230,49	0,00%	10,5/sec	3,21	313,0
/images/sta...	100	1009	394	1463	165,85	0,00%	10,8/sec	10,09	953,0
/images/vco...	100	1179	644	2331	369,36	0,00%	9,4/sec	6,18	672,0
/images/jud...	100	1430	796	2401	488,96	0,00%	8,9/sec	8,76	1012,0
/images/sta...	100	1502	844	2388	454,70	0,00%	8,6/sec	6,98	830,0
/images/blo...	100	1473	755	2384	409,72	0,00%	8,0/sec	27,61	3526,0
/images/pa...	100	1491	668	2368	361,12	0,00%	7,9/sec	7,41	959,0
/images/blo...	100	1604	707	3035	449,71	0,00%	6,8/sec	22,38	3394,0
/images/run...	100	1948	1023	3180	608,31	0,00%	5,9/sec	3,99	688,0
/images/blo...	100	2317	700	3411	720,66	0,00%	5,3/sec	2,41	462,0
/problemre...	100	905	122	2891	664,57	0,00%	5,8/sec	109,14	19354,0
/24h/proble...	100	3462	1367	4999	797,28	0,00%	4,9/sec	94,44	19575,0
/24h/proble...	100	3415	1147	5086	699,24	0,00%	5,0/sec	135,02	27820,0
Total	2900	1556	122	6460	1028,71	0,00%	54,7/sec	314,24	5882,2

¿Incluir el nombre del grupo en la etiqueta?

Guardar la tabla de datos

Guardar la cabecera de la tabla

Figura 24. Reporte resumen 100 usuarios no autenticados

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Mostrar sólo:

Escribir en Log Sólo Errores

Éxitos

Configurar

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de By...
/css/datata...	150	2092	759	4066	744,50	0,00%	9,0/sec	73,13	8344,0
/css/main.css	150	1747	786	4714	641,91	0,00%	9,8/sec	48,28	5045,0
/js/script.js	150	1571	767	3550	531,13	0,00%	11,7/sec	139,91	12291,0
/images/i18...	150	1509	798	3648	522,90	0,00%	12,2/sec	16,30	1368,0
/images/top...	150	1403	764	2415	463,94	0,00%	13,1/sec	7,40	580,0
/images/pro...	150	1502	860	2587	471,89	0,00%	13,1/sec	9,47	739,0
/images/blo...	150	1527	833	2809	509,06	0,00%	12,3/sec	46,02	3831,0
/images/i18...	150	1584	864	3014	583,23	0,00%	12,0/sec	15,70	1339,0
/images/blo...	150	1588	863	3005	538,94	0,00%	11,1/sec	38,08	3526,0
/images/vco...	150	1785	864	3006	578,73	0,00%	10,5/sec	6,89	672,0
/images/me...	150	1771	1032	3060	541,51	0,00%	10,1/sec	3,08	313,0
/images/pa...	150	1742	945	2996	602,65	0,00%	9,4/sec	8,76	959,0
/images/run...	150	1838	933	2999	543,62	0,00%	8,4/sec	5,66	688,0
/images/blo...	150	1760	1007	3014	604,27	0,00%	8,9/sec	29,38	3394,0
/images/co...	150	1917	949	3077	599,50	0,00%	8,7/sec	8,46	992,0
/images/co...	150	1834	1007	3025	584,44	0,00%	8,9/sec	8,03	927,0
/images/sta...	150	1816	1010	3125	610,11	0,00%	8,0/sec	6,51	830,0
/images/jud...	150	1792	1024	3302	620,56	0,00%	7,5/sec	7,39	1012,0
/images/blo...	150	1973	1040	3251	705,79	0,00%	7,1/sec	3,20	462,0
/images/sta...	150	2098	1042	3381	768,04	0,00%	7,3/sec	6,76	953,0
/images/blo...	150	2162	1017	3612	830,83	0,00%	7,1/sec	23,75	3448,0
/images/reg...	150	2373	1014	3590	845,41	0,00%	6,4/sec	6,13	985,0
/images/for...	150	2479	1015	3598	810,52	0,00%	6,1/sec	6,23	1053,0
/problemre...	150	3452	116	10160	2325,32	0,00%	5,1/sec	95,46	19354,0
/24h/proble...	150	5152	1143	10358	2004,20	0,00%	4,5/sec	89,87	20230,0
/24h/proble...	150	4705	1260	9401	1719,54	0,00%	4,4/sec	118,60	27820,0
Total	4350	2357	115	10358	1525,59	0,00%	55,3/sec	318,93	5904,8

¿Incluir el nombre del grupo en la etiqueta?

Guardar la tabla de datos

Guardar la cabecera de la tabla

Figura 25. Reporte resumen 150 usuarios no autenticados