



Universidad de las Ciencias Informáticas
Facultad 4

Módulo Académico para el Juez en Línea Caribeño
(COJ)

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

Autores: Olga Nelys Olay González
José Nolberto Isac González

Tutores: Msc. Tomás Orlando Junco Vázquez
Msc. Leandro González Vallejo

Declaración de autoría

Declaramos que somos los únicos autores del trabajo “**Módulo Académico para el Juez en Línea Caribeño**” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

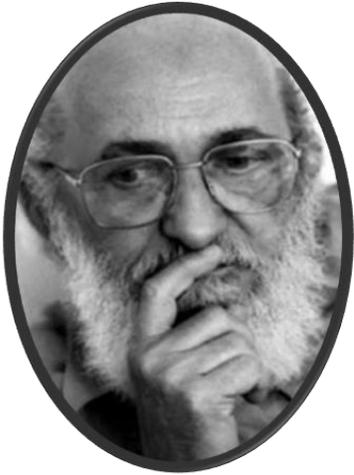
Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autor: Olga Nelys Olay González

Autor: José Nolberto Isac González

Tutor: Tomás Orlando Junco Vázquez

Tutor: Leandro González Vallejo



“La educación no cambia el mundo: cambia a las personas que van a cambiar el mundo.”

Paulo Freire

Dedicatoria

Olga

A mi familia, principalmente a mi mamá y hermanas por su apoyo incondicional durante mis años de estudiante.

Nolberto

A mi familia por siempre apoyarme. A mis padres por su amor, confianza y apoyo incondicional, son las personas más importantes en mi vida. A mis abuelos y en especial a mi abuela Isabel, que donde quiera que esté siempre la llevo en el corazón.

Agradecimientos

Olga

A mi mamá María Elena, por ser madre y padre y enseñarme a transitar por el buen camino. Por su empuje y dedicación a sus hijas. Te quiero mami.

A mis hermanas Rosita y Dianelys, por ser mis confidentes y mis mejores amigas, por secar mis lágrimas y por compartir tantas cosas juntas.

A mi papá Olay, por ayudarme a crecer y siempre creer en mí.

A mi familia, por sacarme muchas veces de apuros y alentarme durante la carrera.

A Pepito, por haber sido mi otro padre y darme el tamaño que tengo.

A mi novio Robert por su amor incondicional, por las largas horas a mi lado apoyándome y dándome aliento, por su manera de sacarme una sonrisa cuando más lo necesitaba. Te amo bebé.

A mi compañero de tesis Nolberto por haberme dado tantos dolores de cabeza y brindarme su amistad.

A Jennifer, por ser mi hermanita de no sangre y estar ahí en todo momento para aconsejarme.

A mis amigas Liena, Lisandra y Edelín por todos los años juntos y los grandes recuerdos de amistad.

A todos los profesores que me impartieron clases, en especial a Rosalba, Rafael, Maritza por su gran maestría y sus consejos como si fueran nuestros padres.

A mis compañeros de aula, gracias por los buenos y malos momentos juntos.

A las amistades que conocí en el transcurso de la carrera y me brindaron su cariño, en especial a Gabriel, Maide, el Jimmy, Yisel, Mayara, Elaine, Yamila, Adrián, Dairelis, Yadira, Alexo, Zapato, el Rafa, Richard, Angelito, el Viki, Yuliet, Yeidy, Sardina, Neysa, Niurka y Dainier.

A mis tutores, por sus consejos y ayuda durante esta tesis.

Por último, doy gracias a Dios por haber entrado en esta universidad y llevarme de ella los recuerdos más lindos de mi vida.

Nolberto

A mi mamá Iraida por ser la persona más especial en el mundo, por darme consejos en los momentos más difíciles de la vida, mima te quiero mucho.

A mi papá José Miguel por brindarme todo su apoyo y confianza, y por forma a la persona que soy hoy en día.

A mis abuelos, en especial a mi abuelo Norberto por todo el cariño, confianza y consejos que me brindo.

A mis tías y tíos por su preocupación y consejos a lo largo de mi carrera y de mi vida.

A mi hermano Yoan y mis tres primas que más quiero Karime, Isnalvis y Mariacari (tata) por todo el cariño que me han brindado.

A mi compañera de tesis Olga por todos los momentos difíciles que pasamos en la carrera y los dolores de cabeza que le hice pasar, te quiero Olguita.

A mis tutores Tomas y Leandro, por todo el conocimiento y confianza que me brindaron no solo en este curso, sino en los anteriores.

A mis compañeros de aula, en especial a Gabriel por todos los momentos buenos y malos que compartimos en la Universidad.

A mis compañeros de equipo de la ACM Eduardo y Broche por todos los resultados que alcance, que sin ellos no hubiese sido posible.

A la profesora Maritza por todos los consejos que me dio en estos cinco años de la carrera y por todos los dolores de cabeza que le hice pasar.

A mis compañeros del IPVCE que no estarán leyendo esta dedicatoria, pero que han significado mucho para mí y que voy a estar en deuda con ellos para toda la vida.

A mis dos mejores amigos de la infancia y de la vocacional Yaumel y Bismar que no pudieron estar aquí.

Y por último a todas esas personas que han significado algo para mí en esta universidad que siempre estuvieron ahí para apoyarme en todo lo que me hizo falta, en especial Yolanda, Alejandro Camp, Rolando, Cesar, Lázaro, Maidevis, Rey, Carlos Alberto, Heriberto.

Resumen

Los jueces en línea de programación son aplicaciones web creadas con el objetivo de incrementar el desarrollo de habilidades en las técnicas de programación y el diseño de algoritmos. Generalmente son empleados para la preparación de concursos pero al fomentar la capacidad resolutoria de los estudiantes ante complejos problemas, han despertado el interés de extrapolar sus potencialidades al ámbito docente. El Juez en Línea Caribeño (COJ, por sus siglas en inglés) fue concebido para entrenar a los universitarios cubanos y del Caribe. Sin embargo, el COJ, con las funcionalidades adecuadas, puede convertirse en una herramienta de apoyo en el proceso de enseñanza-aprendizaje. La presente investigación tiene como objetivo desarrollar un módulo académico para gestionar el proceso de enseñanza-aprendizaje de la Programación en el Juez en Línea Caribeño. Para el desarrollo de la investigación fue necesario caracterizar soluciones semejantes con funcionalidades académicas y la tendencia actual de las mismas. Se estudiaron diferentes herramientas, tecnologías y metodologías de desarrollo. Finalmente se obtuvo una solución informática que responde a los objetivos planteados en la investigación.

Palabras claves: jueces, funcionalidades, módulo, metodología.

Abstract

The programming online judges are web applications created with the aim of increasing the development of skills in programming techniques and design of algorithms. They are generally used for the preparation for contests but encourage decisiveness of students to complex problems have aroused the interest of extrapolating their potential by teaching field. Judge in Caribbean Line (COJ, for its acronym in English) is designed to train the Cuban and Caribbean university. However, the COJ, with the right features, can become a support tool in the teaching-learning process. This research aims to develop an academic module for managing the teaching and learning of programming in the Caribbean Online Judge. For the development of the investigation it was necessary to characterize such solutions with academic capabilities and the current trend of them. Different tools, technologies and development methodologies were studied. Finally, a solution that meets the objectives in the research was obtained.

Keywords: judges, functionality, module methodology.

Índice de contenido

Introducción	14
Estructura Capitular	16
1. Capítulo 1. Fundamentación teórica	17
1.1 Introducción	17
1.2 Definición de juez en línea	17
1.3 Antecedentes de los jueces en línea.....	17
1.4 Jueces en línea para entornos competitivos	18
1.4.1 Juez en Línea de la Universidad de Valladolid.....	18
1.4.2 Juez en Línea de la Universidad de Tecnología de Gdansk	19
1.4.3 Juez en Línea Caribeño.....	19
1.5 Jueces en línea con funcionalidades académicas.....	19
1.5.1 Sistema de Envíos Online.....	20
1.5.2 Entorno Virtual de Aprendizaje Judge.org.....	20
1.5.3 Juez en Línea Académico URI.....	20
1.7 Arquitectura	21
1.8 Tecnologías	22
1.9 Tecnologías de desarrollo de software	22
1.9.2 Framework de desarrollo	25
1.9.3 Servidores de bases de datos.....	26
1.9.4 Entorno Integrado de Desarrollo	28
1.9.5 Servidor web.....	29
1.9.6 Metodologías de desarrollo de software.....	30
2. Capítulo 2. Descripción del Sistema	33
2.1 Introducción	33
2.2 Descripción de la solución	33
2.3 Usuarios del sistema.....	33
2.4 Lista de Reservas del Producto	34
2.5 Aspectos no funcionales del sistema	35
2.6 Exploración.....	35
2.6.1 Historias de Usuario	35
2.7 Planificación	38
2.7.1 Iteraciones.....	38
2.7.2 Plan de Entrega	39
2.8 Diseño	40

2.8.1	Tarjetas CRC.....	40
2.9	Modelo de datos	41
3.	Capítulo 3. Implementación y pruebas.....	43
3.1	Introducción	43
3.2	Implementación	43
3.2.1	Tareas de implementación.....	43
3.3	Pruebas	45
3.3.1	Pruebas unitarias.....	46
3.3.2	Resultados.....	46
3.3.3	Pruebas de aceptación	47
	Conclusiones generales.....	50
	Recomendaciones	51
	Glosario de términos.....	52
	Referencias Bibliográficas.....	53

Índice de Tablas

Tabla 1. Usuarios del sistema	34
Tabla 2. HU 1: Planificación de actividades	36
Tabla 3. HU 2: Soporte para los lenguajes de programación	36
Tabla 4. HU 3: Estadísticas.....	37
Tabla 5. HU 4: Gestión de roles	37
Tabla 6. HU 5: Gestión de cursos	37
Tabla 7. HU 6: Gestión de contenidos.....	38
Tabla 8. HU 7: Revisión de soluciones.....	38
Tabla 9. Plan de iteraciones.....	39
Tabla 10. Plan de entrega.....	40
Tabla 11. Tarjeta CRC course.....	40
Tabla 12. Tarjeta CRC chapter	40
Tabla 13. Tarjeta CRC chapter_content.....	41
Tabla 14. Tarjeta CRC course_logs	41
Tabla 15. Tarjeta CRC course_user.....	41
Tabla 16. Tarea de implementación Eliminar curso.....	44
Tabla 17. Tarea de implementación Adicionar curso.....	44
Tabla 18. Tarea de implementación Editar curso.	44
Tabla 19. Tarea de implementación Editar usuario de un curso.....	44
Tabla 20. Tarea de implementación Editar capítulo.	45
Tabla 21. Tarea de implementación Adicionar capítulo.	45
Tabla 22. Tarea de implementación Adicionar capítulo.....	45
Tabla 23. Prueba de aceptación Planificación de actividades con tiempo límite de resolución.....	48
Tabla 24. Prueba de aceptación Soporte para los lenguajes usados en la docencia.	48
Tabla 25. Prueba de aceptación Gestión de estudiantes	48
Tabla 26. Prueba de aceptación Gestión de roles.....	49
Tabla 27. Prueba de aceptación Gestión de cursos.	49
Tabla 28. Prueba de aceptación Gestión de contenidos.	49

Índice de Ilustraciones

Ilustración 1. Fases de desarrollo de XP	32
Ilustración 2. Modelo de Base de Datos del módulo académico.....	42
Ilustración 3. Resultados de las pruebas unitarias.....	47

Introducción

El Concurso Internacional Universitario ACM¹ de Programación (ACM-ICPC² por sus siglas en inglés) se realiza cada año con la participación de varios equipos en representación de diferentes universidades del mundo. Este acontecimiento constituye en la actualidad uno de los más importantes y prestigiosos eventos de programación de computadoras. Fomentándose aquí el trabajo en equipo, la resolución de tareas y el desarrollo rápido de software. Además de poner a prueba la capacidad de los estudiantes para la búsqueda de soluciones a disímiles problemas.

Para brindar soporte a la preparación de este tipo de concursos y gracias al desarrollo que han alcanzado las Tecnologías de la Información y las Comunicaciones (TIC), existen variadas herramientas en Internet que incluyen un amplio conjunto de problemas a resolver de distintas áreas de conocimiento y que además son utilizadas por estudiantes y entrenadores con el propósito de lograr buenos resultados. Estas aplicaciones son conocidas como Jueces en Línea de programación. Un juez en línea es una aplicación web para entornos generalmente académicos, que incluye varios problemas de distintas materias para ser resueltos mediante técnicas de programación y, lo más importante, evalúa automáticamente las soluciones de sus usuarios en los varios lenguajes de programación disponibles. Se caracterizan por contener una interfaz bien definida para la interacción con el usuario y un alto nivel de disponibilidad, permitiendo el libre acceso a la aplicación en todo momento a las personas registradas (1).

Con el auge del movimiento ACM-ICPC en Cuba surge el Juez en Línea Caribeño (COJ, por sus siglas en inglés), desarrollado y mantenido por la Universidad de las Ciencias Informáticas (UCI). Este sistema que está disponible en Internet (<http://coj.uci.cu>) desde el 5 de junio de 2010 ha sido fundamental en el desarrollo de múltiples concursos de programación en la UCI, Cuba y el Caribe. (2) En Cuba, debido a los problemas que existen para acceder a aplicaciones semejantes, tiene una amplia difusión el COJ en la enseñanza universitaria, específicamente en las carreras con perfil computacional. Gracias a sus potencialidades proporciona un ambiente de auto-preparación y de intercambio de conocimientos. Estas ventajas que ofrece permiten que sea utilizado para orientar tareas de estudio independiente.

Sin embargo, las opciones que ofrece ahora mismo la plataforma COJ no cubren todas las necesidades de los docentes para gestionar las actividades con sus grupos de estudiantes. Lo primero es que el profesor no puede revisar las implementaciones de los estudiantes para observar el avance ni realizarles sugerencias para tener una respuesta aceptada. No está concebido el concepto de grupo de clases, no existe una lógica de negocio que facilite la retroalimentación, no hay forma de

¹ Association for Computing Machinery

² ACM International Collegiate Programming Contest.

guardar un histórico de las actividades docentes realizadas ni de conocer sus resultados, no se facilita el seguimiento sobre las frecuencias de acceso a la plataforma, las clasificaciones por áreas de conocimientos actuales no se corresponden con las necesarias para apoyar el proceso docente, no existe la posibilidad de que los profesores agreguen problemas a la aplicación que sean convenientes dadas sus asignaturas y los momentos del curso. Por demás no existen propuestas de documentación a revisar ni otro tipo de orientación asociados a los distintos problemas a resolver por parte de los estudiantes.

A partir de la situación problemática anteriormente planteada se define para la presente investigación el siguiente **problema a resolver**: ¿Cómo gestionar el proceso de enseñanza-aprendizaje de la Programación en el Juez en Línea Caribeño? Este problema enmarca como **objeto de estudio** los jueces en línea de programación. Delimitando como **campo de acción** la aplicación de los jueces en línea en la enseñanza de la programación.

Por lo que se define como **objetivo general**: Desarrollar un módulo con funcionalidades académicas para gestionar el proceso de enseñanza-aprendizaje de la Programación en el Juez en Línea Caribeño.

En correspondencia con lo planteado anteriormente se formulan las siguientes **preguntas científicas**:

- ¿Qué características debe tener un juez en línea para gestionar el proceso de enseñanza-aprendizaje de la Programación?
- ¿Qué tecnologías facilitarían la implementación de las características necesarias para gestionar el proceso de enseñanza-aprendizaje de la Programación en el Juez en Línea Caribeño?
- ¿Cuál es el funcionamiento y estructura que debe tener un módulo con funcionalidades académicas para gestionar el proceso de enseñanza-aprendizaje de la Programación?
- ¿Cuáles son las funcionalidades necesarias que debe tener un módulo académico para gestionar el proceso de enseñanza-aprendizaje de la Programación?
- ¿Cómo es recomendable la propuesta de solución desarrollada?

Para dar cumplimiento a las preguntas científicas se trazaron las siguientes **tareas a cumplir**:

1. Realización de un estudio sobre los principales jueces en línea existentes en la UCI, Cuba y el resto del mundo.
2. Caracterización de los jueces en línea con funcionalidades docentes.
3. Identificación de las tecnologías necesarias para desarrollar un módulo académico para los jueces en línea.
4. Identificación del funcionamiento y estructura de un módulo académico en los jueces en línea.

5. Identificación de las funcionalidades necesarias en un módulo académico para los jueces en línea.
6. Recomendación de la propuesta de solución desarrollada.

Métodos de investigación utilizados:

Métodos teóricos:

- **Histórico-Lógico:** Posibilitó estudiar la trayectoria histórica y la lógica del desarrollo de los jueces en línea, permitiendo conocer de forma general el funcionamiento de los mismos.
- **Analítico-Sintético:** Permitió dividir y estudiar las aplicaciones similares, extrayendo de esta manera los elementos que podían ser útiles para dar solución al problema.
- **Modelación:** Posibilitó la creación y esbozo de los diferentes diagramas y modelos generados durante el proceso de desarrollo del módulo académico.

Métodos empíricos:

- **Observación:** A través de ella se recopiló información de los jueces en línea con funcionalidades académicas, permitiendo vincular a la solución aquellas funcionalidades que resultasen de utilidad.

Estructura Capítular

El presente documento se encuentra estructurado en tres capítulos. En el primero se hace un análisis de los principales conceptos relacionados con los jueces en línea y posibles herramientas o utilidades que pudieran complementar sus prestaciones. Se realiza un estudio del estado del arte de los más importantes jueces en línea y de su empleo en varias instituciones del mundo, principalmente en el clásico contexto docente que son las asignaturas de una carrera.

En el segundo capítulo se describe la propuesta de solución, enfatizando los elementos más importantes que permiten su inclusión en el ambiente docente.

En el tercer y último capítulo se valida el resultado obtenido teniendo en cuenta principalmente los resultados de la aplicación desarrollada.

1. Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se abordarán los antecedentes de los jueces en línea que incluyen funcionalidades para apoyar la docencia y se realizará un estudio acerca de los principales sistemas de este tipo que existen en la actualidad en los diferentes ámbitos, para obtener conocimiento de cuáles son las tendencias que rigen el desarrollo de los mismos. Además, se seleccionarán las herramientas a utilizar durante el proceso de desarrollo de la solución propuesta.

1.2 Definición de juez en línea

Un juez en línea es un sistema, por lo general web, que permite juzgar programas de computación que intenten solucionar tareas propuestas. Estos sistemas pueden compilar y ejecutar códigos fuente, y ponerlos a prueba con los juegos de datos definidos para la tarea seleccionada. Las posibles soluciones se ejecutan con restricciones tales como: límite de tiempo de ejecución, límite de memoria, tamaño de código fuente, restricciones de seguridad, entre otras. (3) La salida de cada programa enviado por el usuario es capturada por el sistema y comparada contra la salida que se tiene de la tarea en cuestión o será evaluada por un evaluador externo.

1.3 Antecedentes de los jueces en línea

Los concursos de programación al estilo ACM-ICPC tienen sus antecedentes en un concurso local celebrado en 1970 en la Universidad A&M de Tejas, Estados Unidos, y no es hasta 1977 que se desarrollan varias rondas clasificatorias como parte del concurso, realizándose por primera vez una final mundial organizada con el auspicio de la ACM. A pesar de ser durante mucho tiempo el único evento de su tipo en el mundo, prácticamente sólo equipos de Estados Unidos y Canadá participaron entre los años 1977 y 1989. No es hasta 1997 cuando con el patrocinio de la empresa IBM se aprecia un aumento considerable en las cantidades de equipos, instituciones y países participantes, evidenciándose un crecimiento anual entre el 10 y el 20% (4). En la realización de dichos concursos un conjunto de jueces proponen varias tareas a resolver por cada uno de los equipos participantes, cada tarea se compone de uno o varios casos de prueba³ y puede traer aparejado un evaluador externo⁴. Los equipos pueden solicitar la evaluación de una tarea en cualquier momento de la competencia sin ningún tipo de restricción de cantidad de solicitudes, esperando una respuesta en el menor tiempo posible. En cada competencia se realizan un número de solicitudes de evaluación que está condicionado por la cantidad de equipos y la dificultad de las tareas propuestas, efectuándose un

³ Ficheros que contienen la entrada y salida original de cada tarea.

⁴ Programa informático cuyo objetivo es la evaluación de la salida generada por una solución en las tareas con múltiples salidas para un mismo caso de prueba.

cúmulo significativo de envíos por cada competencia, lo cual convierte la evaluación en un proceso tan complejo que es casi imposible realizarlo eficientemente desde un enfoque manual.

Generalmente, el proceso de evaluación de las tareas era realizado por evaluadores automáticos, los cuales son sistemas informáticos cuyo propósito general es el procesamiento eficiente de las solicitudes de evaluación. En el año 1988 surge el proyecto PC²⁵, siendo uno de los primeros evaluadores automáticos del cual se tiene conocimiento, el mismo fue desarrollado inicialmente en Turbo Pascal y se mantuvo el desarrollo con dicha tecnología hasta 1999 que fue remplazada por Java, brindando por primera vez soporte para sistemas Unix⁶, FreeBSD⁷ y Solaris. Hasta el año 2001 se utilizaron varias versiones del sistema para la realización de competencias alrededor de todo el mundo, ya en el año 2000 se realiza la primera final mundial haciendo uso del sistema PC² sobre plataforma Unix (5). Con el surgimiento y la popularidad de la web, los evaluadores automáticos fueron evolucionando a lo que se conoce en la actualidad como Jueces en línea.

1.4 Jueces en línea para entornos competitivos

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), el Juez en Línea de la Universidad de Tecnología de Gdansk (SPOJ), y el Juez en Línea de la Universidad de Pekín (PKU). En la Universidad de las Ciencias Informáticas (UCI), se han desarrollado y empleados jueces tal como el de la Cátedra de Programación Avanzada (CPAV), que aunque en estos momentos no se encuentra brindando servicios, logró un paso de avance en cuanto a la programación competitiva en la Universidad. Además, existe el Juez en Línea Caribeño (por sus siglas en inglés, COJ), que es empleado también en el resto del país y en la región del Caribe.

1.4.1 Juez en Línea de la Universidad de Valladolid

El UVA es uno de los jueces en línea más antiguos y prestigiosos del mundo. Surge en el año 1995 con el propósito de ser usado como herramienta de entrenamiento para concursos de programación, pero no es hasta el año 1997 que comienza a cobrar fuerza en la comunidad internacional (3). El UVA se caracteriza por tener una interfaz amigable y fácil de usar, además de ofrecer las funcionalidades típicas de todo juez en línea y de poseer uno de los mejores archivos de problemas del mundo. Sin embargo, se le puede señalar la escasa disponibilidad de lenguajes de programación, pues sus usuarios solamente pueden enviar soluciones implementadas en C⁸, C++, JAVA⁹ y PASCAL¹⁰.

⁵ Sistema de control de concursos de programación creado en la Universidad Estatal de California —Sacramento.

⁶ Sistema operativo portable, multitarea y multiusuario desarrollado en principio en los laboratorios Bell de AT&T.

⁷ Sistema operativo libre basado en las CPU de arquitecturas INTEL aunque en la actualidad es compatible con otro número de arquitecturas.

⁸ Es considerado uno de los lenguajes más usado en el desarrollo de sistemas operativos.

⁹ <http://java.com>

¹⁰ Lenguaje de programación pensado para ser usado con fines científicos.

1.4.2 Juez en Línea de la Universidad de Tecnología de Gdansk

El 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 (6). El SPOJ se caracteriza por tener una interfaz intuitiva, sencilla y ligera, propiciando a usuarios nuevos e inexpertos la fácil comprensión del propósito de la herramienta. Quizás las dos características más distintivas del sistema son el uso de MOSS¹¹, como herramienta de detección de plagios, y la posibilidad brindada a determinados usuarios de organizar concursos bajo sus propias reglas y problemas. Se considera que el SPOJ hace un uso excesivo de anuncios publicitarios en las principales vistas del sistema; además de no brindar información referente a su historia. Por otra parte, también carece de referencias a documentación o tutoriales destinados a la preparación de los usuarios en los temas relacionados con la programación competitiva. Se considera que el POJ brinda pocos lenguajes de programación (solamente cinco) y que la principal deficiencia del sistema radica en estar implementado sobre plataforma Windows.

1.4.3 Juez en Línea Caribeño

El desarrollo del sistema base (Xtreme Online Judge) comenzó en el año 2006 a partir de la versión limitada y restrictiva del POJ, bajo la "Iniciativa Xtreme", compuesta principalmente por estudiantes y profesores de la antigua Facultad 8 de la UCI. Luego de que la UCI se uniera al Movimiento ACM-ICPC y liderara la creación de la Comunidad Caribeña del ACM-ICPC, el Xtreme Online Judge fue seleccionado para publicarse en Internet como COJv1, siendo la fecha de publicación el 5 de junio de 2010. (2) El COJ posee las funcionalidades típicas de todos los sistemas que tuvieron como punto de partida al POJ, aunque la interfaz y la arquitectura de la información cambiaron notablemente. La característica más considerable del sistema es la rapidez en los tiempos de respuesta en la mayoría de las interfaces, no así en la vista del estado de los envíos.

La versión actual (2.0) incluye archivo 24 horas, posibilidades de revivir competencias anteriores, solución de comunicación interna entre los usuarios, un excelente y completo módulo de administración, ejecución en ambiente seguro de las soluciones de los usuarios y posibilidad de realizar la evaluación de soluciones de manera distribuida.

1.5 Jueces en línea con funcionalidades académicas

Durante la investigación realizada se han identificado alrededor de 100 jueces en línea, de los cuales la mayor cantidad está concentrada en Asia; usualmente dichos sistemas están hospedados en instituciones de Educación Superior, debido a que el mayor número de usuarios son estudiantes

¹¹ Herramienta desarrollada en la Universidad de Stanford desde el año 1994 para la detección de plagio
<http://theory.stanford.edu/~aiken/moss/>

universitarios. Por otra parte, estos jueces en línea han sido utilizados como herramientas de apoyo al proceso de enseñanza-aprendizaje.

1.5.1 Sistema de Envíos Online

El Sistema de Envíos Online (BOSS¹², por sus siglas en inglés) fue desarrollado hace unos cuantos años por la Universidad de Warwick, como una herramienta para facilitar el envío y el procesamiento de ejercicios de programación. (7) Cuenta con un módulo para incluir los listados de estudiantes y demás aspectos relacionados con las características específicas de la institución, lo cual constituye un elemento interesante para la inclusión de un juez en línea al contexto general universitario. La principal limitación de BOSS es que no tiene en cuenta ninguna funcionalidad de orientación a los estudiantes y además por estar tan ligado a los conceptos módulo y tareas para los estudiantes no permite disponer de actividades competitivas o del concepto de ranking, aspectos típicos y motivadores existentes en otros jueces en línea.

1.5.2 Entorno Virtual de Aprendizaje Jutge.org

Jutge.org¹³ es un juez en línea de acceso abierto a la enseñanza de la programación, utilizado principalmente por la Facultad de Informática de Barcelona y la Facultad de Estadística de la Universidad de Cataluña, donde el estudiante trata de resolver más de 800 problemas usando alrededor de 22 lenguajes de programación. Al contrario de otros populares jueces en línea, Jutge.org está diseñado para el estudiante y el profesor: Por un lado, el repositorio de problemas está conformado para novatos que utilizan por primera vez el juez, con gran claridad en la organización. Por otra parte, el sistema fue diseñado para recrear un ambiente de enseñanza virtual, donde los profesores puedan administrar sus propios cursos y a sus estudiantes, adicionar problemas, concursos y exámenes. (8)

1.5.3 Juez en Línea Académico URI

El Juez en Línea Académico (URI¹⁴), desarrollado en el Departamento de Ciencias de la Computación de la Universidad Regional Integrada de Brasil, es una herramienta web que ayuda a los profesores en las clases de programación y motiva a los estudiantes a practicar más la base teórica aprendida en dichas clases, además de que pueden compartir sus conocimientos sobre lógica, algoritmización y programación. El módulo académico permite que el profesor administre las asignaturas y el listado de ejercicios en temas específicos en un organizado ambiente en línea, el cuál presenta variados beneficios en comparación con el método tradicional de resolver los ejercicios escritos a mano. (9)

¹² <http://boss.org.uk/>

¹³ <https://www.jutge.org/>

¹⁴ <https://www.urionlinejudge.com.br>

1.6 Tendencias de los Jueces en Línea con funcionalidades académicas

Los jueces en línea son aplicaciones que se caracterizan por el amplio consumo de recursos de hardware causado en gran medida por la demanda de procesamiento de las soluciones a evaluar. Un juez en línea medianamente popular recibe en un día promedio de 800 a 3000 solicitudes de evaluación, pudiendo incrementarse considerablemente en un día de competencia, por lo que los jueces en línea deben basar su implementación en arquitecturas eficientes, lenguajes robustos y en la optimización de sus funcionalidades. Por otra parte, los jueces en línea son sistemas propensos a recibir ataques por lo que la implementación de los mismos debe estar sustentada por una plataforma lo más segura posible. Los jueces en línea generalmente están compuestos por dos módulos fundamentales, el módulo de presentación (Generalmente una aplicación web) y el módulo de evaluación que puede estar embebido en el módulo de presentación o ser una aplicación independiente, generalmente conocida como evaluador automático o motor de evaluación, el cual es usado de forma centralizada o distribuida según la arquitectura seleccionada para la implementación del mismo.

1.7 Arquitectura

Los jueces en línea convencionales (POJ, SPOJ, COJ v1, entre otros) implementan evaluadores de forma centralizada o embebidos como funcionalidad de la aplicación Web; los evaluadores centralizados o embebidos se caracterizan por poseer tiempos de respuesta más rápidos que los evaluadores distribuidos pues no incluyen en su funcionamiento el factor “rapidez de la red”, pero al mismo tiempo, son propensos a sobrecargar el servidor y su funcionamiento pone en riesgo la integridad de los datos del juez en Línea y la seguridad del servidor donde se encuentra el sistema principal.

Los jueces en línea más modernos (Codeforces, UCOCoch, entre otros) implementan los evaluadores automáticos aplicando **programación distribuida**¹⁵. EL uso de evaluadores distribuidos puede incluir una tercera aplicación que es el distribuidor/balanceador de tareas/carga, encargado de comunicarse con cada uno de los servicios distribuidos por los nodos de la red. Los motores de evaluación distribuidos se caracterizan por minimizar los daños al servidor ante un posible ataque sobre el sistema, además de optimizar el procesamiento de las soluciones por la posibilidad de realización de tareas simultáneas en diferentes servidores. Por otra parte, el uso de sistemas distribuidos no limita la posibilidad de usarse de forma centralizada.

Existen varias tendencias/arquitecturas en la implementación de los jueces en línea para el proceso de almacenamiento de los juegos de datos de cada problema.

¹⁵ Paradigma de programación enfocado en desarrollar sistemas distribuidos, abiertos, escalables, transparentes y tolerantes a fallos. Este paradigma es el resultado natural del uso de las computadoras y las redes (10)

- Almacenamiento físico en el servidor donde se ejecuta el evaluador o motor de calificación.
- Almacenamiento en Base de Datos.
- Almacenamiento mediante un sistema de archivos de red (NFS por sus siglas en ingles).

El almacenamiento físico en el servidor es la variante más empleada entre todos los jueces en línea por ser la menos compleja de implementar aunque no es la más segura y eficiente debido a que la distribución de los juegos de datos entre varios evaluadores distribuidos es un proceso complejo. El almacenamiento en Base de Datos es la menos eficiente y es usada generalmente por sistemas personalizados para competencias, que manejan pequeños volúmenes de problemas y no se usan en entrenamiento las 24 horas. El almacenamiento usando NFS es el que mayor dificultad presenta para su configuración pero al mismo tiempo es el que exhibe mayor seguridad y eficiencia pues puede funcionar de forma centralizada y distribuida, permitiendo gestionar la seguridad e integridad de los ficheros usados por cada evaluador automático, además de mantener cada servidor actualizado de forma automática ante los cambios que puedan ocurrir en el servidor principal.

1.8 Tecnologías

El estudio realizado sobre los diferentes jueces en línea develó que tanto los de rasgos competitivos como los jueces con funcionalidades académicas presentan las siguientes similitudes tecnológicas:

- Las 2 tecnologías más utilizadas para el desarrollo de la capa de presentación entre los jueces en línea analizados son Java y PHP. Java alcanza este auge debido a que el POJ ha sido el punto de partida para muchos jueces en línea alrededor de todo el mundo mayoritariamente en el continente asiático y PHP por su condición de lenguaje más usado y popular para el desarrollo de aplicaciones Web dinámicas en los últimos años (11).
- Para el desarrollo del módulo de evaluación las tecnologías utilizadas han sido C++, Java y Python¹⁶ siendo el primero el que ha sido usado mayoritariamente por integrarse fácilmente con las diferentes arquitecturas y caracterizarse por ser uno de los lenguajes más rápidos de los últimos tiempos; por otra parte, Java y Python han sido las otras opciones de desarrollo destacándose Java debido a la popularidad alcanzada por la plataforma en los últimos años.
- Las tecnologías de procesamiento de datos más utilizadas entre estos tipos de aplicaciones han sido PostgreSQL y MySQL; sin existir una diferencia marcada en el uso de una u otra para el desarrollo de jueces en línea

1.9 Tecnologías de desarrollo de software

1.9.1 Lenguajes de programación

¹⁶ <http://www.python.org>

Un lenguaje de programación es aquel elemento dentro de la Informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis, poniéndose a disposición del programador para que éste pueda comunicarse con los dispositivos de software y hardware (12).

C++

C++ es un lenguaje de programación de propósito general basado en el lenguaje de programación C. Además de los recursos de C, C++ proporciona clases, funciones en línea (*inline*), sobrecarga de operadores, sobrecarga de nombres de funciones, tipos constantes, referencias, operadores para el manejo del almacenamiento disponible, verificación de argumentos de funciones y conversiones de tipos (13). El lenguaje C++ se comenzó a desarrollar en 1980, por Bjarne Stroustrup; en la actualidad C++ es un lenguaje versátil y potente, se puede utilizar en la construcción de todo tipo de aplicaciones aunque su mayor aplicación radica en las aplicaciones de escritorio, sistemas operativos y servicios.

PHP

PHP es un lenguaje interpretado en el lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad. Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutados por el servidor web a través de un intérprete antes de transferir al cliente que lo ha solicitado un resultado en forma de código HTML puro. Al ser un lenguaje que sigue la corriente *open source*, tanto el intérprete como son totalmente accesibles de forma gratuita en la red. Es un lenguaje multiplataforma, que puede funcionar sobre la mayoría de los servidores web y brinda soporta a más de 20 tipos de base de datos (19).

Java

Java es un lenguaje de programación de alto nivel desarrollado por Sun Microsystem a principios de la década del 90. Su sintaxis es muy parecida a la de C y C++ ya que su desarrollo fue inspirado en los mismos, siempre buscando eliminar errores que suelen inducirse en lo que respecta a la manipulación de memoria y punteros (14).

Los programas en Java generalmente son compilados a un lenguaje intermedio llamado *bytecode*, que luego es interpretado por una máquina virtual (JVM). Esta última sirve como una plataforma de abstracción entre el hardware y el lenguaje permitiendo que se pueda ejecutar el programa sobre cualquier arquitectura. Señalar que la compilación a código máquina también es posible, brindando mayor eficiencia en la ejecución del programa pero limita su característica multiplataforma.

Entre las características principales de este lenguaje se encuentran las siguientes:

- Simple: Elimina la complejidad de los lenguajes como C y C++ dando paso al contexto de los lenguajes modernos orientados a objetos.
- Orientado a objetos: Soporta las características esenciales del paradigma de la programación orientada a objetos: encapsulamiento, herencia y polimorfismo.
- Robusto: Elimina el uso de apuntadores para referenciar áreas de memoria, además, despoja al desarrollador de la necesidad de liberar la memoria que la aplicación ya no usa. También requiere la declaración explícita tanto de los tipos de datos como de los métodos.
- Multiplataforma: El mismo código Java que funciona en un sistema operativo, funciona en cualquier otro que tenga instalada la máquina virtual de Java.
- Multitareas: Permite la ejecución concurrente de varios procesos ligeros o hilos de ejecución.
- 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.

Python

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos (15).

Python es gratuito, incluso, para propósitos empresariales.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que se puede desarrollar, como Unix, Windows, OS/2, Mac.

Análisis de la selección del lenguaje de programación

Se selecciona el lenguaje de programación Java para el desarrollo del módulo académico porque permite el desarrollo de aplicaciones funcionales en cualquier plataforma (se pueden ejecutar en cualquier sistema operativo ofreciendo los mismos resultados). Esta característica de Java es sumamente importante, pues teniendo en cuenta que en las universidades cubanas existen computadoras con cualquier plataforma, existe la posibilidad de que los usuarios tengan que acceder

al módulo académico en una computadora con cualquier sistema operativo. Esta selección permite a los autores aprovechar la experiencia obtenida con el uso de la plataforma Java, así como reutilizar el conocimiento adquirido previamente en el uso de las herramientas asociadas al lenguaje, lo cual compensa las deficiencias que pueda presentar la plataforma.

1.9.2 Framework de desarrollo

En general, con el término framework, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones (16).

Spring

Spring es un framework de aplicación desarrollado por la compañía Interface 21, para aplicaciones escritas en el lenguaje de programación Java. Fue creado gracias a la colaboración de grandes programadores, entre ellos se encuentran como principales partícipes y líderes de este proyecto Rod Johnson y Jürgen Höller. Estos dos desarrolladores, además de otros colaboradores que juntando toda su experiencia en el desarrollo de aplicaciones J2EE (Java 2 Enterprise Editions), incluyendo EJB (Enterprise JavaBeans), Servlets y JSP (Java Server Pages), lograron combinar dichas herramientas y otras más en un sólo paquete, para brindar una estructura más sólida y un mejor soporte para este tipo de aplicaciones. Además se considera a Spring un framework lightweight, es decir liviano o ligero, ya que no es una aplicación que requiera de muchos recursos para su ejecución, además el framework completo puede ser distribuido en un archivo .jar ¹⁷de alrededor de 1 MB, lo cual representa muy poco espacio, y para la cantidad de servicios que ofrece es relativamente insignificante su tamaño. (17).

Entre sus posibilidades más potentes radica su contenedor Inversión de Control también llamado Inyección de Dependencias (DI por sus siglas en inglés) que es una técnica alternativa a las clásicas búsquedas de recursos vía JNDI¹⁸, permite configurar las clases en un archivo XML¹⁹ y definir en él las dependencias. De esta forma la aplicación se vuelve muy modular y a la vez no adquiere dependencias con Spring.

¹⁷ Tipo de archivo que presenta la información comprimida.

¹⁸ Interfaz de Nombrado y Directorio Java (*Java Naming and Directory Interface*) es una Interfaz de Programación de Aplicaciones (API) de Java para servicios de directorio.

¹⁹ XML, siglas en inglés de *eXtensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C).

Struts

Struts es un proyecto de la Apache Software Foundation que tiene como objetivo proporcionar un entorno específico para la construcción de grandes aplicaciones web. Se trata de un entorno basado en el concepto de Model 2, que es una variación del paradigma de diseño clásico conocido como Modelo-Vista-Controlador (MVC, por sus siglas en inglés), donde se separan claramente los roles de los diseñadores web, los programadores java y los administradores de bases de datos. El núcleo del entorno está constituido por un conjunto de tecnologías estándar, como son los Servlets, JavaBeans, ResourceBundles y XML. (18)

Análisis de la selección del framework

Después de realizar un análisis sobre los *frameworks* anteriores, los autores de la solución deciden utilizar Spring en su versión como *framework* principal para el desarrollo de la solución, porque luego de realizar un estudio comparativo entre ambos, se demostró que Spring posee las siguientes ventajas sobre su homólogo:

- Spring MVC es muy flexible, implementa toda su estructura mediante interfaces; a diferencia de Struts, el cual obliga a heredar de clases concretas tanto en sus Controladores como en sus Formularios. Además, todas las partes del framework son configurables a través de *plugins*.
- Los controladores de Spring MVC se configuran mediante Inversión de Control, lo cual los hace fácilmente integrables con otros objetos que estén en el contexto de Spring y, por consiguiente, que sean manejables por éste.
- Resulta más fácil probar las partes de Spring MVC que las de Struts, debido a que el primero evita forzosamente la herencia de una clase y la dependencia directa en el controlador del servlet²⁰ que despacha las peticiones.

1.9.3 Servidores de bases de datos

Los servidores de Bases de datos, también conocidos como DBMS (acrónimo en inglés de *DataBase Management Systems*), son programas que permiten organizar datos en una o más tablas relacionadas. Los servidores de Bases de Datos se utilizan en todo el mundo en una amplia variedad de aplicaciones. Prácticamente cualquier aplicación que necesite almacenamiento, acceso y análisis de datos estructurados hace uso de algún tipo de DBMS. La mayoría de los servidores de bases de datos ofrecen una interfaz de texto rudimentaria que permite interactuar con el servidor usando SQL (*Structured Query Language*).

²⁰ <http://www.servlets.com>

MySQL

Este gestor de base de datos es bien aceptado debido a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. MySQL es un sistema de gestión de base de datos relacional. Su diseño multi-hilo le permite soportar una gran carga de forma muy eficiente (19).

Por un lado se ofrece bajo la GNU-GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSIC.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el *copyright* de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios.

PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS). Basado en el proyecto POSTGRES, de la universidad de Berkeley. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. (20)

A continuación se enumeran las principales características de este gestor de base de datos (20):

- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen).
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Cuenta con un rico conjunto de tipos de datos, permitiendo además, su extensión mediante tipos, operadores definidos y programados por el usuario.
- Es altamente confiable en cuanto a estabilidad se refiere.

Análisis de la selección del servidor de bases de datos

Al finalizar la comparación se decide utilizar PostgreSQL en su versión 1.16.1 como Sistema Gestor de Bases de Datos (SGBD), debido a su condición de ser multiplataforma y estar licenciado bajo GNU/GPL. Por otra parte, posee varias características que normalmente sólo se encuentran en homólogos comerciales (por ejemplo: Oracle), al mismo tiempo que (respecto a MySQL -su principal rival-) demuestra superioridad en cuanto a las necesidades del problema planteado. Dicha superioridad se manifiesta del siguiente modo: soporta una capacidad de almacenamiento en el orden

de los TB (*Tera Bytes*), posee mayor escalabilidad (soporta una mayor cantidad de peticiones concurrentes), y tiene la capacidad de comprobar la integridad referencial (20).

1.9.4 Entorno Integrado de Desarrollo

Un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés) es un programa informático compuesto por un conjunto de herramientas de programación que le facilitan al usuario editar, compilar, depurar códigos, además de construir de manera rápida interfaces gráficas de usuarios. Puede ser usado para uno o varios lenguajes de programación.

Eclipse

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios. Presenta una estructura en forma de *plugins* que permite que los mismos puedan ser creados utilizando una API especial brindada por los desarrolladores y que puedan ser agregados fácilmente (sólo ubicarlos dentro de un directorio); gracias a esta facilidad posee una gama amplia y creciente de *plugins* desarrollados ya sea por empresas o por usuarios comunes (21).

NetBeans

NetBeans es un proyecto de código abierto de gran éxito con una enorme base de usuarios, una comunidad en constante crecimiento. Es una aplicación de código abierto diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java, dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y soporta la instalación de módulos para ampliar las funcionalidades del IDE (22).

Análisis de la selección del IDE

Los autores seleccionan NetBeans en su versión 7.3.1 como IDE para implementar la solución, teniendo en cuenta la experiencia anterior en el uso de la herramienta, además de considerar que la misma brinda un mejor soporte a las tecnologías seleccionadas para el desarrollo que Eclipse. Algunos de los principales elementos considerados fueron:

- NetBeans brinda (de forma local) las librerías necesarias para programar con el *framework* Spring, mientras que Eclipse requiere conexión a Internet para realizar la descarga de las dependencias.
- NetBeans se integra de forma natural con el servidor web Apache Tomcat, mientras que Eclipse requiere de varias configuraciones para garantizar la integración.

- Los autores consideran que la interfaz de NetBeans es más fácil de usar e intuitiva que la de Eclipse.

1.9.5 Servidor web

Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente.

El servidor Web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador Web) y que responde a estas peticiones, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Apache Tomcat

Apache Tomcat es desarrollado, en un entorno abierto y participativo y publicado bajo la licencia Apache versión 2, por miembros de la *Apache Software Foundation* y voluntarios independientes. El proyecto tiene la intención de ser una colaboración de los mejores desarrolladores de su clase de todo el mundo. Tomcat puede funcionar como servidor web por sí mismo. Al principio de su desarrollo existió la percepción de que la utilización de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con mínimos requisitos de velocidad y gestión de transacciones. Actualmente ya no existe esa percepción y Tomcat es usado como servidor Web independiente en entornos con alto nivel de tráfico y alta disponibilidad (23).

Glassfish

Glassfish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Oracle Glassfish Enterprise Server (antes *Sun Glassfish Enterprise Server*). Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU-GPL (24).

Análisis de la selección del servidor web

Los autores deciden utilizar Apache Tomcat en su versión 7.0.34.0, pues el mismo posee una licencia completamente libre, facilitando la obtención de nuevas versiones. Sin embargo, el licenciamiento dual que posee su principal competidor (Glassfish) condiciona el uso de la versión libre, la cual puede ser cerrada en cualquier momento (provocando la pérdida de soporte y actualización del servidor web).

1.9.6 Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos y técnicas que permiten estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Se basan en los principios de uno o más modelos para guiar el proceso de desarrollo del software. Son las encargadas de detallar la información que se debe producir como resultado de una actividad durante el proceso de desarrollo y la información necesaria para comenzarla.

Para la selección de la metodología de desarrollo a utilizar fueron consideradas tres de las más utilizadas; estas son el Proceso Unificado de Desarrollo (RUP), la Programación Extrema (XP) y Scrum (S).

Rational Unified Process

La metodología de desarrollo RUP se caracteriza por estar basada en componentes y utilizar el Lenguaje Unificado de Modelado (UML) para visualizar, especificar, construir y documentar un software. (25)

El ciclo de vida de RUP se caracteriza por ser dirigido por casos de uso, siendo los casos de uso los encargados de iniciar el proceso de desarrollo y servir como guía para el desarrollo de las siguientes fases y de los artefactos que estas generan, centrado en la arquitectura ya que presta especial atención al establecimiento temprano de una buena organización y estructura, que le permite tener una visión común entre todos los involucrados (desarrolladores y usuarios), además de brindar una perspectiva clara del sistema completo, iterativo e incremental, lo que significa que en cada una de las fases (inicio, elaboración, desarrollo y transición) se ejecutaran una o varias iteraciones que comprenden actividades de todos los flujos de trabajo.

Consta de nueve flujos de trabajo fundamentales, de ellos seis denominados “de Ingeniería”, que son: modelado del negocio, requerimientos, análisis y diseño, implementación, prueba y despliegue. Los restantes tres flujos de trabajo son considerados de apoyo y son la administración de proyectos, la gestión de configuración y cambios, y finalmente el ambiente.

La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

Scrum

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que se pueda mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. (26).

XP

XP es una metodología ágil para pequeños y medianos equipos, desarrollando software cuando los requerimientos son ambiguos o rápidamente cambiantes. A diferencia de los procesos tradicionales para desarrollar software, XP asume el cambio como algo natural, y que, indefectiblemente, en alguna etapa de un proyecto sucede. En XP se realiza el software que el cliente solicita y necesita, en el momento que lo precisa, alentando a los programadores a responder a los requerimientos cambiantes que plantea el cliente en cualquier momento. Esto es posible porque está diseñado para adaptarse en forma inmediata a los cambios, con bajos costos asociados, en cualquier etapa del ciclo de vida. En pocas palabras, XP “abrazo” el cambio.

El ciclo de vida de XP consiste básicamente de seis fases: Exploración, Planificación, Iterations to Release²¹, Producción, Mantenimiento y Muerte. (27)

Uno de los factores que hacen que XP sea tan utilizado y efectivo son las prácticas que se realizan durante el proyecto. XP es un conjunto de ideas y prácticas de metodologías ya existente y por eso deben ser muy tenidas en cuenta a la hora de su implementación. A continuación se presentan las prácticas más utilizadas.

- 40 Horas semanales
- Cliente on-site
- Diseño simple
- El juego de la planificación
- Estándares de codificación
- Integración continua
- Metáfora
- Pequeñas entregas
- Programación por pares
- Propiedad colectiva
- Testing
- Refactoring

No se debe olvidar que la base de XP es que el proceso se mantenga ágil en todo momento, lo cual significa que se pueda adaptar al cambio sin mayores complicaciones. (28)

²¹ Conjunto de funcionalidad que es integrada para realizar un programa ejecutable.

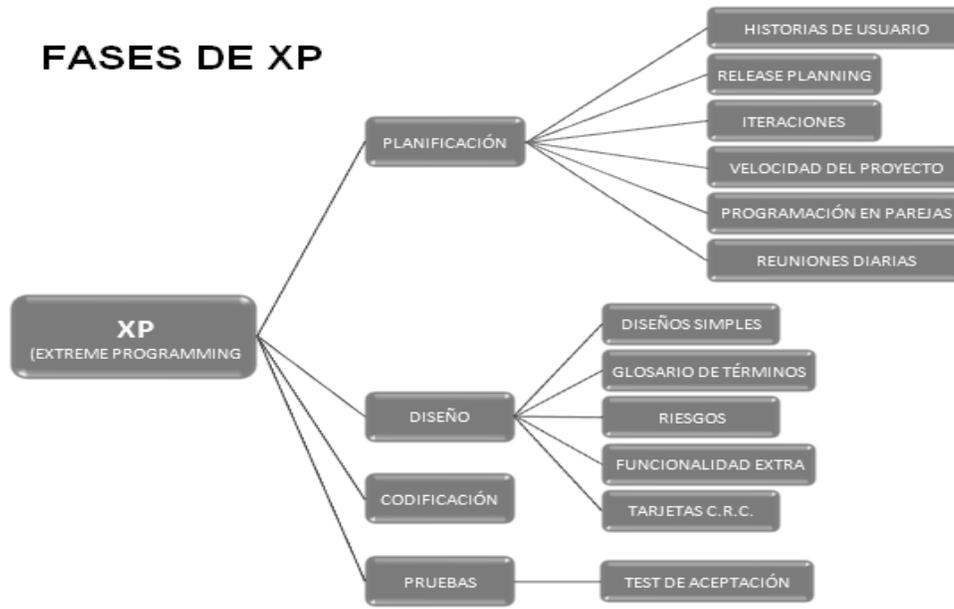


Ilustración 1. Fases de desarrollo de XP

Análisis de la selección de la metodología de desarrollo

La metodología Programación Extrema (XP, por sus siglas en inglés) es una metodología ágil para pequeños equipos, que no exige un gran cúmulo de artefactos, se moldea a los cambios bruscos que puedan surgir en el proceso de desarrollo y permite realizar un producto de calidad en un corto periodo de tiempo. Por tales razones, se ha elegido como metodología de desarrollo a XP.

2 Capítulo 2. Descripción del Sistema

2.1 Introducción

En el presente capítulo se recogen las funcionalidades de la solución propuesta, reflejadas en las Historias de Usuarios con sus prototipos de interfaces de usuario no funcionales, así como la planificación de las iteraciones del ciclo de desarrollo. Además se realiza el diseño de la solución propuesta y se describen los patrones a emplear en el desarrollo.

2.2 Descripción de la solución

Los Jueces en Línea son aplicaciones web que permiten ejercitar el conocimiento sobre programación a través de complejos ejercicios lógicos. A pesar de que han sido creados con el objetivo de facilitar la preparación para eventos competitivos se pueden utilizar sus potencialidades para vincularlo a la docencia. La solución que se propone a continuación persigue como objetivo fundamental diseñar un módulo académico para el Juez en Línea Caribeño. Dicho módulo permitirá que los profesores puedan gestionar las actividades con sus grupos de estudiantes, llevar un seguimiento sobre los ejercicios que han realizado y la frecuencia de acceso a la plataforma, orientar problemas de acuerdo al tema que esté impartiendo en ese momento, entre otras funcionalidades.

2.3 Usuarios del sistema

Los usuarios del sistema son todas aquellas personas o sistemas que interactúan con el mismo con el objetivo de obtener un resultado específico. El módulo académico presenta ciertas restricciones de acceso a las funcionalidades en dependencia del usuario a través del cual se acceda a la aplicación.

Usuarios del sistema	Justificación
<p style="text-align: center;">Profesor</p>	<p>Este usuario se refiere a los profesores que acceden a la aplicación. Los mismos pueden tener vasta o poca experiencia con el uso del módulo ya que presenta una interfaz amigable y entendible.</p>
<p style="text-align: center;">Estudiante</p>	<p>Este usuario se refiere a los estudiantes que acceden a la aplicación. Los mismos pueden tener vasta o poca experiencia con el uso del módulo ya que presenta una interfaz amigable y entendible.</p>

Administrador	Este usuario es el encargado de realizar todas las modificaciones pertinentes en la aplicación, además de poseer acceso absoluto a todas las interfaces.
----------------------	--

Tabla 1. Usuarios del sistema

2.4 Lista de Reservas del Producto

Según define la metodología XP la **Lista de Reservas del Producto** está compuesta por los aspectos funcionales con los que debe contar la aplicación para satisfacer con eficiencia las necesidades requeridas por el cliente. A continuación se muestra el listado de los mismos.

R1. Planificación de actividades con tiempo límite de resolución. Una de las actividades más comunes realizadas en los jueces en línea son las competencias. Una competencia tiene un momento de inicio, tiempo de duración y problemas a resolver en ese tiempo. Esta característica se mantendrá en el módulo académico permitiendo a los profesores realizar actividades en un tiempo determinado. El uso de esta práctica estaría directamente sincronizado con los elementos de aprendizaje competitivo y podría incentivar mucho la preparación de los estudiantes.

R2. Soporte para los lenguajes usados en la docencia. Queda claro que la posible utilidad de un módulo como este depende de que los lenguajes enseñados en los distintos cursos de Programación sean soportados por el sistema. Se tendrían en cuenta, principalmente, Java, C++, C#, Python, PHP y Prolog por ser los más utilizados.

R3. Gestión de estudiantes. La aplicación permitirá que el usuario profesor/ administrador lleve constancia del listado de los estudiantes matriculados en el curso. Si el curso es privado el profesor/administrador puede aceptar o no un estudiante determinado

R4. Gestión de roles. Como mismo ocurre en el EVA y otras plataformas educativas el conjunto de interfaces y funcionalidades disponibles para un usuario dependen de su función en el proceso. Se identifican inicialmente los roles de Administrador, Estudiante y Profesor.

R5. Gestión de cursos. El sistema deberá suministrar o permitir incluir la información de qué usuarios son profesores y cuáles estudiantes. Se controlará qué grupos de clases son atendidos por cada profesor y en qué curso. Además de que el profesor podrá adicionar y eliminar sus propios cursos. Mientras que el estudiante tiene la opción de ver un listado con todos los cursos en los que está matriculado.

R6. Gestión de contenidos. El sistema deberá permitir que se puedan gestionar los capítulos que tendrá un curso (adicionar, eliminar, editar, listar) por parte del profesor.

R7. Facilidad para la revisión de soluciones. Teniendo en cuenta el flujo de procesos de funcionamiento de un juez en línea, conociendo que para cada problema y sus datos de prueba las

soluciones de los estudiantes generarán una salida (correcta o no), el profesor podrá comunicarle a sus estudiantes, a través de un correo, sus errores o aciertos en cuanto a la solución.

2.5 Aspectos no funcionales del sistema

Los aspectos no funcionales son propiedades o cualidades que el producto debe tener. Son características del mismo que lo hacen atractivo, usable, rápido y confiable.

Software

- Se requiere de un navegador web a partir de la versión en adelante.
- Se requiere de acceso a Internet.

Hardware

- Memoria RAM: 8 GB
- Espacio en disco duro: 20 GB

Usabilidad

- El módulo será diseñado para que integre todos los aspectos relacionados con la usabilidad. Esta característica permitirá que el sistema sea eficiente, fácil de aprender con un nivel de abstracción adecuado para la retención de ideas, fácil de usar tanto por los profesores como por los estudiantes y muy amigable para los que lo utilizan por vez primera.

Rendimiento

- Rapidez y eficiencia tanto en los tiempos de respuesta como en la velocidad de procesamiento. Garantizar velocidad estable de navegación a nivel de aplicación. Proporcionar tiempos de respuesta mínimos para los procesos del sistema.

Portabilidad

- La herramienta deberá funcionar sobre las distribuciones de GNU/Linux.

2.6 Exploración

Como punto de partida en el ciclo de vida de un proyecto, XP propone la **Exploración** como la primera etapa donde los clientes plantean a grandes rasgos las Historias de Usuario que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. (28)

2.6.1 Historias de Usuario

Las Historias de Usuario (HU) son un instrumento para el levantamiento de requerimientos para el desarrollo de un software, que ha emergido con la aparición de los nuevos marcos de trabajo de

desarrollo ágil, son descripciones cortas de una necesidad de un cliente del software que se esté desarrollando.

Algunas características deseables de las HU son: (27)

- Que sean escritas por el usuario o por un analista de negocio que le represente.
- Frase corta que encaje en una tarjeta de tres por cinco pulgadas.
- Debe describir el rol desempeñado por el usuario en el sistema, descrito de forma explícita.
- Debe describir el beneficio para el área de negocio que representa esta funcionalidad.

A continuación se muestran las HU que representan a las funcionalidades que fueron implementadas para la propuesta de solución.

Historia de Usuario	
Número: 1	Usuario: Administrador
Nombre historia: Planificación de actividades con tiempo límite de resolución.	
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 2 semanas	Iteración asignada: 1
Descripción: El usuario profesor al publicar una actividad, podrá darle un tiempo límite de resolución para que el usuario estudiante la realice.	
Información adicional (Observaciones):	

Tabla 2. HU 1: Planificación de actividades

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre historia: Soporte para los lenguajes usados en la docencia	
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 15 días	Iteración asignada: 1
Descripción: El sistema debe permitir que el usuario estudiante pueda enviar sus respuestas en los siguientes lenguajes de programación fundamentalmente: Java, C++, C#, Python, PHP y Prolog.	
Información adicional (Observaciones):	

Tabla 3. HU 2: Soporte para los lenguajes de programación

Historia de Usuario	
Número: 3	Usuario: Administrador

Nombre historia: Gestión de estudiantes	
Prioridad en negocio: Alta	Nivel de Complejidad: Baja
Estimación: 1 semana	Iteración asignada: 1
Descripción: El sistema debe mostrar un listado con todos los estudiantes matriculados en un curso y el usuario profesor puede adicionarlos o eliminarlos en dependencia del estado (privado o público) del curso.	
Información adicional (Observaciones):	

Tabla 4. HU 3: Estadísticas

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre historia: Gestión de roles	
Prioridad en negocio: Alta	Nivel de Complejidad: Baja
Estimación: 10 días	Iteración asignada: 1
Descripción: El sistema debe permitir el acceso a los usuarios a través de contraseña y mostrará una interfaz diferente teniendo en cuenta los permisos que tenga el usuario registrado.	
Información adicional (Observaciones):	

Tabla 5. HU 4: Gestión de roles

Historia de Usuario	
Número: 5	Usuario: Administrador
Nombre historia: Gestión de cursos	
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 2 semana	Iteración asignada: 1
Descripción: El sistema debe permitir adicionar, listar, eliminar y editar un curso.	
Información adicional (Observaciones):	

Tabla 6. HU 5: Gestión de cursos

Historia de Usuario	
Número: 6	Usuario: Administrador
Nombre historia: Gestión de contenidos	

Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 2 semana	Iteración asignada: 1
Descripción: El sistema debe permitir adicionar, listar, eliminar y editar el o los capítulos que tendrá un curso.	
Información adicional (Observaciones):	

Tabla 7. HU 6: Gestión de contenidos

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre historia: Revisión de soluciones	
Prioridad en negocio: Alta	Nivel de Complejidad: Baja
Estimación: 1 semana	Iteración asignada: 1
Descripción: El sistema le permitirá al usuario profesor ver las soluciones de sus estudiantes y enviarles un correo con observaciones acerca de las mismas.	
Información adicional (Observaciones):	

Tabla 8. HU 7: Revisión de soluciones

2.7 Planificación

La fase de Planificación se realiza en pocos días, no siendo menos importante, ya que su objetivo principal es priorizar las HU. Los programadores estiman el esfuerzo de cada una de ellas, se define el cronograma y se acuerda el alcance de la entrega, el cual debe incluir varias iteraciones. El cliente prioriza las HU seleccionando cuales se realizarán en cada iteración, logrando en la primera iteración un sistema con la arquitectura de proyecto.

Otro aspecto a tener en cuenta durante esta fase es la velocidad del proyecto, la cual se estima utilizando el tiempo empleado en el desarrollo de las iteraciones terminadas. Esta medida debe reevaluarse una vez concluida 3 o 4 iteraciones y en caso de no cumplir el tiempo estimado, debe ser negociando con el cliente un nuevo Plan de Entregas. (27)

2.7.1 Iteraciones

XP enfatiza en el carácter iterativo e incremental del desarrollo, la fase de **Planificación** es la principal en el ciclo de vida de un proyecto. Las iteraciones agrupan un conjunto de HU a implementar en un período de tiempo, generando al final de cada una un entregable funcional. Son relativamente cortas. La entrega con rapidez de módulos al cliente aumenta la retroalimentación y resulta más

provechoso en términos de calidad del producto que una entrega a largo plazo.

Las HU no cuentan con suficientes detalles como para permitir su análisis y desarrollo sin la presencia del cliente durante la fase. Por lo que se hace necesario al comienzo de cada iteración realizar las tareas necesarias de análisis en conjunto con el cliente, consolidado los datos necesarios para la implementación. La participación del cliente durante esta fase del ciclo es de vital importancia. Culminar sin errores una iteración constituye un avance notable en el desarrollo del proyecto.

Iteraciones	Orden de las Historias de Usuario a implementar	Cantidad de tiempo de trabajo
Iteración 1	HU1. Planificación de actividades con tiempo límite de resolución. HU2. Soporte para los lenguajes usados en la docencia. HU3. Gestión de estudiantes. HU4. Gestión de roles. HU5. Gestión de cursos. HU6. Gestión de contenidos. HU7. Revisión de soluciones.	10 semanas

Tabla 9. Plan de iteraciones.

2.7.2 Plan de Entrega

El Plan de Entrega es el resultado final de la fase de Planificación acordado en una reunión entre todos los actores del proyecto, estableciendo cuáles HU serán agrupadas para conformar una entrega. Es un compromiso que establece el grupo de trabajo con el cliente, aspecto de vital importancia, debido a que entregas tardías podrían tener grandes repercusiones en la economía y moral del equipo de trabajo.

El cronograma de entregas es realizado en dependencia de las estimaciones de tiempos de desarrollo realizadas por los programadores. La estimación es uno de los temas más complicados del desarrollo de un proyecto de software, es por ello que se recomienda luego de algunas iteraciones realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el Plan de Entregas y ajustarlo en caso de ser necesario.

Entregable	Final 1ra iteración
Módulo Académico para el Juez en Línea Caribeño(COJ)	versión 1.0

Tabla 10. Plan de entrega

2.8 Diseño

XP establece prácticas especializadas que inciden directamente en la realización del diseño para lograr un sistema robusto y reutilizable tratando de mantener su simplicidad, es decir, crear un diseño evolutivo que se va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente.

2.8.1 Tarjetas CRC

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

Tarjetas CRC	
Nombre de la clase: course	
Responsabilidades: Definir una entidad que represente un curso con los atributos: course_id, course_name, course_type, is_public, username, initdate, enabled, chapter, course_overview, problem_points, score_active.	Colaboradores: Date User

Tabla 11. Tarjeta CRC course

Tarjetas CRC	
Nombre de la clase: chapter	
Responsabilidades: Definir una entidad que represente un capítulo con los atributos: course_id, chapter_id, chapter_name, chapter_number.	Colaboradores: Date Problem Chapter_Content

Tabla 12. Tarjeta CRC chapter

Tarjetas CRC	
Nombre de la clase: Chapter_Content	

<p>Responsabilidades: Definir una entidad que represente el contenido de un capítulo con los atributos: content_id, chapter_id, content_type, content_address, pid.</p>	<p>Colaboradores: Chapter Problem</p>
--	--

Tabla 13. Tarjeta CRC chapter_content

Tarjetas CRC	
Nombre de la clase: course_logs	
<p>Responsabilidades: Definir una entidad que maneje las imágenes y mensajes de un curso con los atributos: username, message, course_logs_id, course_id.</p>	<p>Colaboradores: Course User</p>

Tabla 14. Tarjeta CRC course_logs

Tarjetas CRC	
Nombre de la clase: course_user	
<p>Responsabilidades: Definir una entidad que represente la relación entre curso y user con los atributos: username, course_id, banned.</p>	<p>Colaboradores: Course User</p>

Tabla 15. Tarjeta CRC course_user

2.9 Modelo de datos

Es la descripción de la organización de una base de datos, constituyéndose en una representación gráfica orientada a la obtención de la estructura de datos mediante métodos. En un enfoque más amplio, un Modelo de Datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí. El Modelo de Datos está formado por dos componentes: componente estática, relacionada con el lenguaje de definición de datos (LDD) y dinámica, relacionada con el lenguaje de manipulación de datos (LMD). (29)

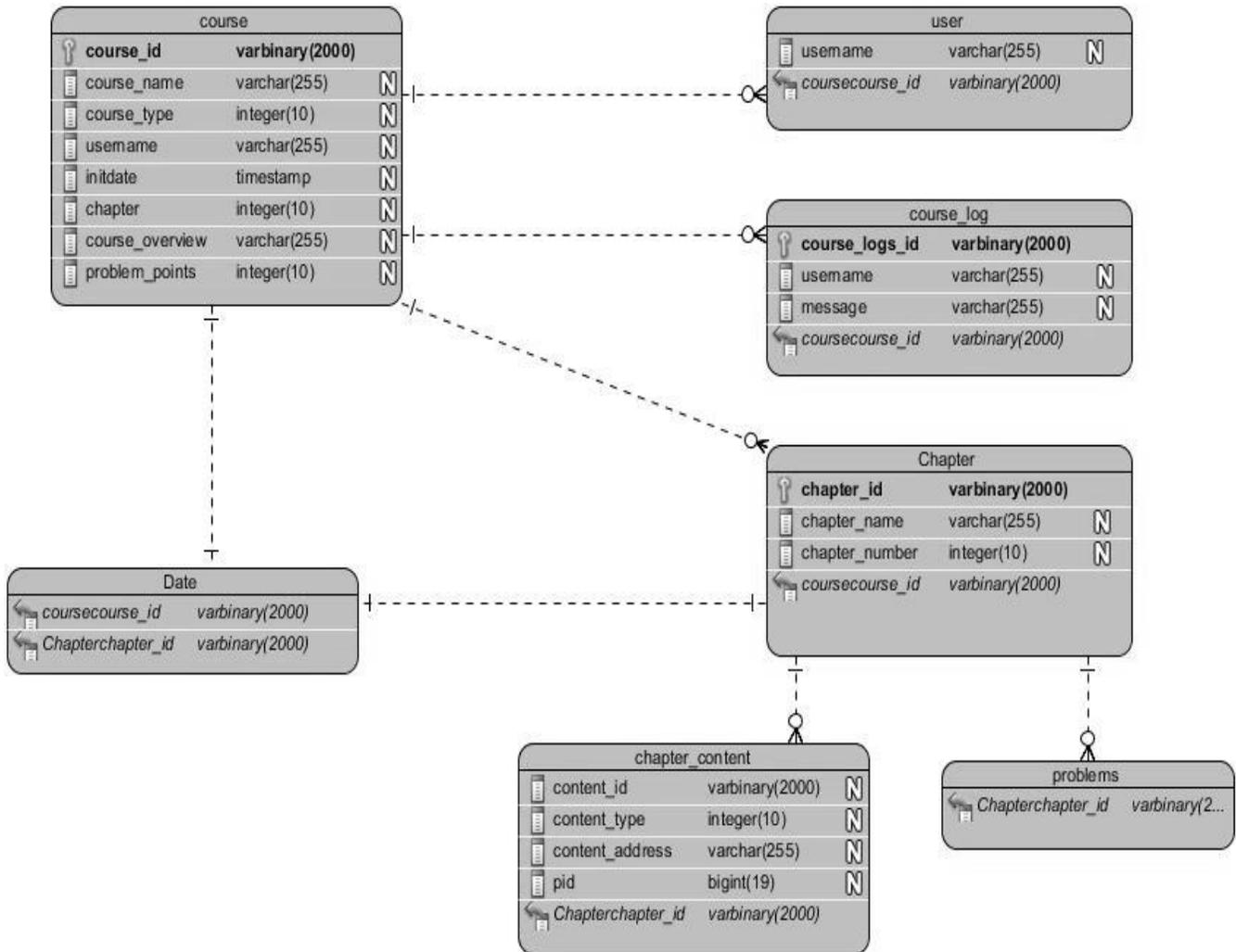


Ilustración 2. Modelo de Base de Datos del módulo académico

3. Capítulo 3. Implementación y pruebas.

3.1 Introducción

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

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

3.2 Implementación

La Fase de Implementación es el espacio dentro del ciclo de vida del proyecto donde el equipo de desarrollo implementa las funcionalidades que fueron especificadas en la Fase de Planificación a través de las HU. Las prácticas más importantes de esta etapa son:

- El cliente siempre está siempre disponible.
- Todo el código debe programarse en parejas.
- Dejar las optimizaciones para el final.
- No trabajar más de 40 horas semanales.

3.2.1 Tareas de implementación

XP propone realizar unas tarjetas nombradas Tareas de implementación, donde se exponga brevemente cuales fueron las tareas realizadas para implementar las HU. Cada Tarea de implementación estará vinculada a una HU; de esta forma se comprobará el cumplimiento de todas las funcionalidades del sistema. A continuación algunas de las Tareas de implementación por cada HU. El resto de las tareas podrán ser consultadas en los Anexos.

Tarea de implementación	
Número de tarea: 1	Número de HU: 5
Nombre de la tarea: Eliminar curso	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 5 de marzo del 2015	Fecha fin: 6 de marzo del 2015
Programador responsable: José Nolberto Isac	
Descripción: Desarrollo de la clase encargada de eliminar un curso de la aplicación.	

Tabla 16. Tarea de implementación Eliminar curso.

Tarea de implementación	
Número de tarea: 2	Número de HU: 5
Nombre de la tarea: Adicionar curso	
Tipo de tarea: Desarrollo	Estimación: 10 día
Fecha de inicio: 6 de marzo del 2015	Fecha fin: 16 de marzo del 2015
Programador responsable: José Nolberto Isac	
Descripción: Desarrollo de la clase encargada de adicionar un curso a la aplicación.	

Tabla 17. Tarea de implementación Adicionar curso.

Tarea de implementación	
Número de tarea: 3	Número de HU: 5
Nombre de la tarea: Editar curso	
Tipo de tarea: Desarrollo	Estimación: 5 día
Fecha de inicio: 17 de marzo del 2015	Fecha fin: 22 de marzo del 2015
Programador responsable: José Nolberto Isac	
Descripción: Desarrollo de la clase encargada de editar un curso de la aplicación.	

Tabla 18. Tarea de implementación Editar curso.

Tarea de implementación	
Número de tarea: 4	Número de HU: 3
Nombre de la tarea: Editar usuario de un curso	
Tipo de tarea: Desarrollo	Estimación: 1 semana
Fecha de inicio: 23 de marzo del 2015	Fecha fin: 30 de marzo del 2015
Programador responsable: José Nolberto Isac	
Descripción: Desarrollo de la clase encargada de editar un usuario de un curso.	

Tabla 19. Tarea de implementación Editar usuario de un curso.

Tarea de implementación	
Número de tarea: 5	Número de HU: 6
Nombre de la tarea: Editar capítulo	
Tipo de tarea: Desarrollo	Estimación: 2 semanas

Fecha de inicio: 1 de abril del 2015	Fecha fin: 15 de abril del 2015
Programador responsable: José Nolberto Isac	
Descripción: Desarrollo de la clase encargada de editar un capítulo de un curso.	

Tabla 20. Tarea de implementación Editar capítulo.

Tarea de implementación	
Número de tarea: 6	Número de HU: 6
Nombre de la tarea: Adicionar capítulo	
Tipo de tarea: Desarrollo	Estimación: 1 semana
Fecha de inicio: 15 de abril del 2015	Fecha fin: 22 de abril del 2015
Programador responsable: José Nolberto Isac	
Descripción: Desarrollo de la clase encargada de adicionar un capítulo a un curso.	

Tabla 21. Tarea de implementación Adicionar capítulo.

Tarea de implementación	
Número de tarea: 7	Número de HU: 6
Nombre de la tarea: Eliminar capítulo	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha de inicio: 22 de abril del 2015	Fecha fin: 27 de abril del 2015
Programador responsable: José Nolberto Isac	
Descripción: Desarrollo de la clase encargada de eliminar un capítulo de un curso.	

Tabla 22. Tarea de implementación Adicionar capítulo.

3.3 Pruebas

Las pruebas son un elemento importante dentro del ciclo de vida de un proyecto guiado por XP. Desde luego las pruebas permiten la detección de errores y eliminación posterior de los mismos, a la vez que proporcionan una medida de la calidad del software. En XP las pruebas son consideradas parte del desarrollo, ya que cada cambio realizado en el funcionamiento del software debe ser probado al instante; en caso de existir errores se realiza una nueva modificación y se vuelve a probar. Este proceso se repite continuamente hasta que no se encuentren errores, por tanto el código será implantado definitivamente una vez que supere con éxito sus correspondientes unidades de pruebas.

(25)

La metodología XP propone realizar las pruebas unitarias y de aceptación. A continuación se describen las características de estas pruebas, así como la forma en que serán aplicadas.

3.3.1 Pruebas unitarias

Las pruebas unitarias, clasificadas como pruebas de caja blanca²², son realizadas con el objetivo de comprobar el funcionamiento de pequeñas porciones del código. Por lo general son realizadas por los mismos desarrolladores ya que al tener conocimiento detallado del código se les hace mucho más fácil la tarea de generar datos de prueba. La ausencia de pruebas unitarias conlleva a que los programadores inviertan gran cantidad de tiempo en sesiones de debugging ²³buscando posibles errores en el código.

Seguidamente se describe la forma en que se realizará el proceso de pruebas unitarias:

Se crearán las clases y métodos necesarios para probar cada Tarea de implementación generada. En caso de detectar errores en el funcionamiento de alguna de las Tareas de implementación, se realizarán los cambios pertinentes en el código fuente y se le volverá a someter dicha tarea a nuevas pruebas unitarias. Esto implica que a cada Tarea de implementación se le realizará al menos una iteración de pruebas de este tipo.

Una Tarea de implementación será liberada del proceso una vez haya superado exitosamente sus correspondientes unidades de pruebas unitarias.

Se instalará un plugin del framework JUnit en el IDE NetBeans para automatizar la creación de los artefactos necesarios para desarrollar las pruebas unitarias, así como para la evaluación de los resultados.

3.3.2 Resultados

A continuación se presentan los resultados de las pruebas unitarias para las cinco primeras Tareas de implementación. Es necesario señalar que al presentar una sola iteración solamente se muestran los resultados de la misma. De las Tareas de implementación consideradas en el gráfico, sobresale como la de más errores encontrados la tarea *Editar capítulo* con un 33,33% del total.

²² Método de diseño de casos de prueba encargado de examinar los detalles procedimentales del programa, a través de la exploración de los caminos lógicos, bucles y condiciones, y examinando el estado del programa en varios puntos.

²³ Mecanismo utilizado por los programadores para ejecutar paso a paso alguna porción de código. Se realiza con el propósito de analizar y esclarecer el comportamiento de determinada función o variable.

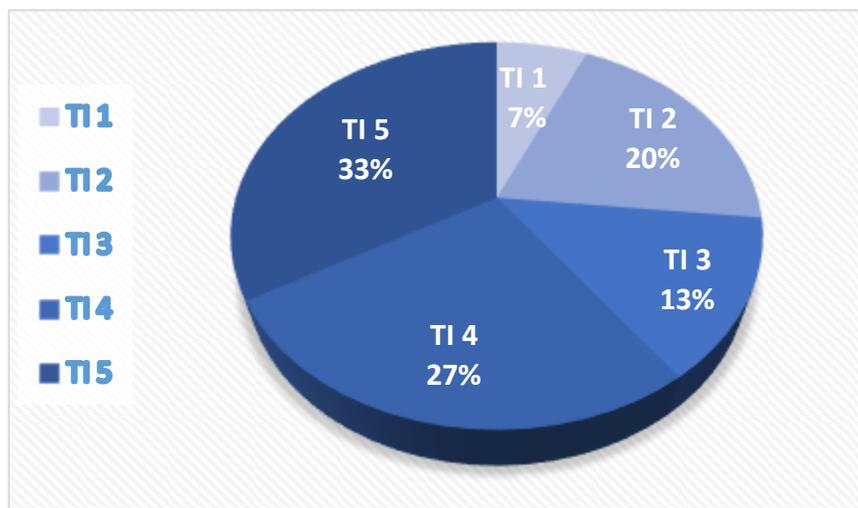


Ilustración 3. Resultados de las pruebas unitarias

3.3.3 Pruebas de aceptación

Las pruebas de aceptación son aquellas realizadas por el cliente a cada HU como acción previa a la producción de una nueva versión del software. Clasificadas como pruebas de caja negra²⁴, las pruebas de aceptación tienen el objetivo de comprobar si las funcionalidades pactadas y recogidas en las HU cumplen las expectativas del cliente. La realización de este tipo de pruebas y la publicación de los resultados debe ser lo más rápido posible para que los desarrolladores puedan realizar con rapidez los cambios que sean necesarios.

3.3.4 Resultados

Casos de pruebas de aceptación	
Número: 1	Nombre: Planificación de actividades con tiempo límite de resolución
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 3 días	Iteración asignada: 1
Descripción: El usuario profesor/administrador tiene la posibilidad de subir actividades para que los estudiantes la realicen en un determinado tiempo.	
Información adicional (Observaciones):	
No Conformidades: Ningunas	

²⁴ Método de diseño de casos de prueba encargado de comprobar que tras la ejecución del programa se cumplan correctamente las especificaciones definidas para cada requisito funcional. Conocidas también como pruebas funcionales, tienen como propósito probar las funcionalidades del software a través de datos de entrada y analizar la salida, sin tener en cuenta el funcionamiento interno del programa.

Resultado: Luego de vencido el tiempo, la publicación no estará accesible.

Tabla 23. Prueba de aceptación Planificación de actividades con tiempo límite de resolución.

Casos de pruebas de aceptación	
Número: 2	Nombre: Soporte para los lenguajes usados en la docencia.
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 3 días	Iteración asignada: 1
Descripción: El usuario profesor al adicionar un problema especificará en que lenguaje desea que sea resuelto y la aplicación tendrá soporte para los lenguajes impartidos en la docencia. El profesor tendrá la opción de dejar activo solo aquel lenguaje que especificó anteriormente.	
Información adicional (Observaciones):	
No Conformidades: Ningunas	
Resultado: El estudiante subirá la respuesta en el lenguaje indicado.	

Tabla 24. Prueba de aceptación Soporte para los lenguajes usados en la docencia.

Casos de pruebas de aceptación	
Número: 3	Nombre: Gestión de estudiantes
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración asignada: 1
Descripción: El usuario profesor tiene la opción de gestionar los estudiantes que estarán matriculados en su curso a través de la opción Edit User.	
Información adicional (Observaciones):	
No Conformidades: Ningunas	
Resultado: El profesor tendrá un listado con todos los estudiantes matriculados.	

Tabla 25. Prueba de aceptación Gestión de estudiantes

Casos de pruebas de aceptación	
Número: 4	Nombre: Gestión de roles
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración asignada: 1
Descripción: Cada usuario que interactúe con el módulo tendrá un rol diferente y a su vez privilegios diferentes. El administrador es el que presenta total acceso a la gestión de todas las funcionalidades.	

Información adicional (Observaciones):
No Conformidades: Ningunas
Resultado: La aplicación mostrará una interfaz diferente para cada tipo de usuario.

Tabla 26. Prueba de aceptación Gestión de roles

Casos de pruebas de aceptación	
Número: 5	Nombre: Gestión de cursos
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración asignada: 1
Descripción: El usuario profesor/administrador tiene la posibilidad de gestionar un curso en la aplicación. Puede realizarlo por la siguiente vía: <ul style="list-style-type: none"> • Nombrar su curso y adicionarlo presionando el botón Add. • En la vista posterior podrá editar su curso, editar los usuarios del curso y adicionarle materiales al curso. 	
Información adicional (Observaciones):	
No Conformidades: Ningunas	
Resultado: Si los datos registrados son correctos, el usuario podrá ver en la opción My courses su nuevo curso registrado.	

Tabla 27. Prueba de aceptación Gestión de cursos.

Casos de pruebas de aceptación	
Número: 6	Nombre: Gestión de contenidos
Prioridad en negocio: Alta	Nivel de Complejidad: Media
Estimación: 2 semana	Iteración asignada: 1
Descripción: El usuario profesor/administrador tiene la posibilidad de gestionar el contenido de un curso. Puede realizarlo por la siguiente vía: <ul style="list-style-type: none"> • Luego de tener su curso adicionado, en la vista Administrar curso tiene la opción de editar capítulos. • En esta vista podrá adicionarle capítulos al curso y a su vez a cada capítulo contenido. 	
Información adicional (Observaciones):	
No Conformidades: Ningunas	
Resultado: Si los datos registrados son correctos, el usuario contará con los capítulos y el contenido de su curso.	

Tabla 28. Prueba de aceptación Gestión de contenidos.

Conclusiones generales

La presente investigación se centró en la concepción y el posterior desarrollo de un módulo con funcionalidades académicas para el Juez en Línea Caribeño. Cumplimentando así los objetivos propuestos y arribando a las siguientes conclusiones:

- El estudio realizado a soluciones similares permitió definir las características y tecnologías fundamentales para el desarrollo del módulo con funcionalidades académicas del Juez en Línea Caribeño.
- Se obtuvo un módulo que responde a todas las características que se plantearon en el levantamiento de requisitos.
- La solución propuesta cumple con el objetivo trazado, pues existe una satisfacción aceptable por parte del cliente, lo cual fue demostrado a partir de las pruebas unitarias y de aceptación realizadas.
- Se logró el módulo con funcionalidades académicas que permitirá la gestión del proceso docente en el Juez en Línea Caribeño.

Recomendaciones

Con el objetivo de continuar enriqueciendo el funcionamiento del módulo desarrollado, se someten a consideración las siguientes recomendaciones:

- Continuar integrándole funcionalidades al módulo académico para que el COJ pueda ser utilizado tanto en el ámbito competitivo como en el docente.
- Integrar el COJ a partir de una plataforma de servicios web con otras herramientas como Moodle y la Plataforma Zera diseñadas para la gestión del proceso docente.

Glosario de términos

Concurso ACM-ICPC: Prestigioso concurso de programación realizado anualmente a diferentes niveles (Local, Nacional, Regional y Mundial) donde solo participan equipos universitarios, buscando obtener buenas posiciones que le clasifiquen hacia el nivel superior. Consiste en intentar resolver un conjunto de problemas en un tiempo determinado (por lo general 4 horas), donde cada equipo obtiene una puntuación según el tiempo en que tardó en solucionar determinado problema; por cada envío incorrecto se penaliza el tiempo total.

TIC: Tecnologías de la Información y las Comunicaciones. Conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones.

ACM: Es una organización a nivel mundial que reúne a los profesionales, estudiantes, investigadores y personas interesadas en la ciencia de la computación. Esta organización cuenta con miembros a nivel mundial y es reconocida como la asociación número uno para los profesionales en computación.

IBM: International Business Machines es una empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York. IBM fabrica y comercializa hardware y software para computadoras, y ofrece servicios de infraestructura, alojamiento de Internet, y consultoría en una amplia gama de áreas relacionadas con la informática, desde computadoras centrales hasta nanotecnología.

UNIX: Designa el núcleo de un sistema operativo multiusuario y multitarea. En un sentido más amplio, comprende el núcleo del sistema operativo más un conjunto de programas que permiten compilar lenguajes de programación, editar texto, interpretar comandos, manejar archivos y discos, acceder a otras máquinas, establecer comunicaciones telefónicas, enviar y recibir correo electrónico, manejar las colas de impresión y un sinnúmero de tareas más. Algunos de estos programas pueden haber sido desarrollados por los propios usuarios.

Referencias Bibliográficas

1. Revilla, Miguel A. Competitive Learning in Informatics: The UVa. Olympiads in Informatics. 2008. págs. 131–148. Vol. Vol. 2.
2. [En línea] <http://coj.uci.cu>
3. SHAHRIAR MANZOOR, RUJIA LIU and MIGUEL A. REVILLA. Competitive Learning in Informatics: The UVa Online Judge Experience.
4. [En línea] [Citado el: 7 de abril de 2015.] <http://www.icpc2014.ru/en/about/history>
5. [En línea] [Citado el: 7 de abril de 2015.] <http://www.ecs.csus.edu/pc2/pc2history.html>
6. Sphere Research Business Solutions. [En línea] [Citado el: 10 de abril de 2015.] <http://sphere-research.com/en/>
7. The BOSS Online Submission and Assessment System. MIKE JOY, NATHAN GRIFFITHS, and RUSSELL BOYATT. No. 3, University of Warwick : ACM Journal on Educational Resources in Computing, 2005, Vol. Vol 5.
8. Jutge.org: An Educational Programming Judge. Omer Giménez, Jordi Petit, Salvador Roura.
9. URI Online Judge Academic: A Tool for Algorithms and Programming Classes. Jean Luca Bez, Neilor A. Tonin, Paulo R. Rodegheri. Universidade Regional Integrada : s.n.
10. [En línea] [Citado el: 2 de abril de 2015.] <http://www.scribd.com/doc/40114693/PROGRAMACION-DISTRIBUIDA>
11. Rasmus Lerdorf, Kevin Tatroe, Peter MacIntyre. Programming PHP - Google Libros.
12. Definición de lenguaje de programación. [En línea] [Citado el: 5 de abril de 2015.] <http://www.definicion.org/lenguaje-de-programacion..>
13. [En línea] [Citado el: 5 de abril de 2015.] http://www.amazon.com/C-Programming-Language-4th/dp/0321563840/ref=la_B000AQ349S_1_1/192-9673349-0873432?s=books&ie=UTF8&qid=1423514483&sr=1-
14. Harvey M. Deitel, Paul J. Deitel. Cómo programar en Java.
15. Python_para_todos.
16. [En línea] [Citado el: 15 de abril de 2015.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
17. [En línea] [Citado el: 15 de abril de 2015.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/sanchez_r_ma/capitulo3.pdf.
18. Hervella, Juan F. Rodríguez. Java Struts Framework.
19. Tecnología para el desarrollo de aplicaciones web-Google books.
20. PostgreSQL: About. [En línea] <http://www.postgresql.org/about/>.
21. About the Eclipse Foundation. [En línea] <http://www.eclipse.org/org/>.

22. **An Introduction to NetBeans.** [En línea] <http://netbeans.org/about/index.html>.
23. **Apache Tomcat - Welcome!** [En línea] <http://tomcat.apache.org/>.
24. **GlassFish: servidor de aplicaciones de código abierto — Java.net.** [En línea] <http://glassfish.java.net/es/>.
25. **Sommerville, Ian.** Ingeniería del software. Séptima edición.
26. **Sutherland, Ken Schwaber y Jeff.** La Guía Definitiva de Scrum: Las Reglas del Juego. .
27. **Metodología XP.** Mousques, Gastón. s.l. : Universidad ORT Uruguay, Facultad de Ingeniería.
28. **Ciclo de vida de un proyecto XP.** Wells, Donovan J.
29. **Data Desing System.** Data Desing System. [En línea] [Citado el: 2 de mayo de 2015.] <http://www.dds-cad.net>.