



Universidad de las Ciencias Informáticas

Facultad 1

Trabajo de Diploma para optar por el Título de

Ingeniero en Ciencias Informáticas

Título: *Módulo para el análisis de los resultados obtenidos durante la ejercitación utilizando juegos didácticos.*

Autor: *Radik Jair Rodríguez Naranjo*

Tutores: *Msc. Dunia Suárez Ferreiro*

Ing. Dasiel Alberto Pérez Suárez

La Habana

19 de Junio del 2013

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____días del mes de _____ del año 2013.

Firma del Autor

(Radik Jair Rodríguez Naranjo)

Firma del Tutor

(Msc. Dunia Suárez Ferreiro)

Firma del Tutor

(Ing. Dasiel Alberto Pérez Suárez)

DATOS DE CONTACTO

Msc. Dunia Suárez Ferreiro: Graduada de Licenciatura en Ciencia de la Computación en la Universidad de la Habana, 2005. Profesor Asistente. Máster en Informática Aplicada, 4 años de experiencia en el tema propuesto, 8 años de graduada.

Email: dsuarezf@uci.cu

Ing. Dasiel Alberto Pérez Suárez: Graduado de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2008.

Email: daperez@uci.cu

Agradecimientos:

Resumen:

El presente trabajo “Módulo para el análisis de los resultados obtenidos durante la ejercitación utilizando juegos didácticos.” se centra en dar solución a la necesidad que tiene el sistema SMProg de contar con un módulo que permita realizar reportes a partir de análisis estadísticos durante la ejercitación de los estudiantes a través de juegos didácticos con los que cuenta esta herramienta. El módulo sigue las pautas establecidas en la Universidad de las Ciencias Informáticas dado que está desarrollado como una aplicación web además de estar creado con herramientas completamente libres y ser de código abierto. Con esta investigación se pretende facilitar el trabajo de los profesores a la hora de llevar un seguimiento y control de la evolución de los estudiantes a través del uso de juegos didácticos. El resultado más relevante de la investigación es la creación de una herramienta que permite el manejo y control detallado de los resultados obtenidos durante la ejercitación de contenidos.

El control y seguimiento sistemático de los estudiantes por parte de los profesores, apoya en gran medida al Proceso de Enseñanza- Aprendizaje y estimula a los educandos a esforzarse más en el momento de ejercitar las materias.

Abstract

This work "Module for the analysis of the results obtained during exercise using learning games." Focuses on a solution to the need for the system SMProg to have a module that allows reports from statistical analysis during exercise of students through educational games are there in this tool. The module follows the guidelines established in the University of Information Sciences as it is developed as a web application created in addition to being completely free tools and being open source. This research aims to facilitate the work of teachers at the time of need to monitor and control the evolution of the students through the use of educational games. The most relevant result of the research is to create a tool that allows the management and detailed control of the results obtained during the exercise content.

The systematic control and monitoring of students by teachers, largely supports the teaching-learning process and encourages learners to do more at the time of exercising subjects.

Índice de contenido

Índice de ilustraciones.....	10
Índice de tablas	11
Introducción:	12
Capítulo 1: Fundamentación teórica.....	17
Introducción:	17
1.1 Conceptos inherentes al dominio del problema	17
1.1.1 Proceso de Enseñanza-Aprendizaje.....	17
1.1.2 Juegos didácticos	17
1.1.3 Generadores de reportes.....	17
1.2 Sistemas generadores de reportes	18
1.2.1 Generadores de reportes utilizados en el mundo.....	18
1.2.2 Generadores de reportes utilizados en Cuba.....	20
Conclusiones del estudio de homólogos.....	22
1.4 Tecnologías para el desarrollo.....	23
1.4.1 Metodología de desarrollo de software	23
1.4.2 Herramientas CASE	25
1.4.3 Lenguaje Unificado de Modelado UML	26
1.4.4 Lenguaje de programación.....	26
1.4.5 Servidor web.....	27
1.4.6 Framework (marco de trabajo)	28
1.4.6 Entorno de Desarrollo Integrado	28

1.4.7 Sistema Gestor de Bases de Datos.....	29
1.4.8 Herramienta de administración de bases de datos	30
Conclusiones	30
Capítulo 2: Características del sistema.....	31
Introducción	31
2.1 Especificación de los requisitos de software	31
2.2 Técnicas de captura de requisitos.....	31
2.1.1 Requisitos funcionales (RF)	32
2.1.2 Requisitos no funcionales (RNF).....	32
2.2 Modelado del sistema.....	34
2.2.1 Casos de uso del sistema	34
2.2.2 Actores del sistema	35
2.2.3 Resumen de descripción de los casos de uso del sistema.....	36
2.3 Descripción ampliada de los casos de uso del sistema	41
2.3.1 Descripción de caso de uso Generar reportes.....	41
2.3.2 Descripción de caso de uso Consultar evaluación.....	43
2.3.3 Descripción de caso de uso Graficar reportes	44
2.3.4 Descripción de caso de uso Exportar reportes	44
2.4 Arquitectura de software	45
2.4.1 Patrón Modelo Vista Controlador	46
2.5 Clases del análisis	47
2.6 Diagrama de despliegue	49
2.7 Modelo de datos.....	50

Conclusiones parciales	51
Capitulo 3: Implementación y prueba	52
Introducción	52
3.1 Estándares de codificación.....	52
3.2 Métodos de prueba.....	53
3.2.1 Prueba de caja negra.....	53
3.3 Diseño de los casos de prueba.....	53
3.3.1 Descripción las variables	54
3.3.2 Diseño de casos de prueba.....	54
3.4 Análisis de los resultados de las pruebas realizadas.....	58
3.4.5 No conformidades detectadas.....	59
Conclusiones parciales	61
Conclusiones	62
Recomendaciones	63
Anexo 2. Encuesta a profesores	68

Índice de ilustraciones

ILUSTRACIÓN 1 DIAGRAMA DE CASO DE USO	35
ILUSTRACIÓN 2: ARQUITECTURA MVC EN SPRING.....	47
ILUSTRACIÓN 3: DIAGRAMA DE CLASES	48
ILUSTRACIÓN 4: DIAGRAMA DE DESPLIEGUE	49
ILUSTRACIÓN 5: MODELO DE DATOS	50
ILUSTRACIÓN 6: DIAGRAMA DE COMPONENTES	51
ILUSTRACIÓN 7: RESULTADOS DE LAS ITERACIONES DE LOS CASOS DE PRUEBA.....	61

Índice de tablas

TABLA 1: ACTORES DEL SISTEMA Y SU DESCRIPCIÓN.....	36
TABLA 2: RESUMEN DEL CASO DE USO: GENERAR REPORTES.....	37
TABLA 3: RESUMEN DEL CASO DE USO: REPORTES DE LISTADO.....	38
TABLA 4: RESUMEN DEL CASO DE USO: REPORTES DE PROMOCIÓN.....	39
TABLA 5: RESUMEN DEL CASO DE USO: EXPORTAR REPORTES.....	39
TABLA 6: RESUMEN DEL CASO DE USO: GRAFICAR REPORTES.....	40
TABLA 7 RESUMEN DE CASO DE USO: CONSULTAR EVALUACIÓN.....	41
TABLA 8 DESCRIPCIÓN DE CASO DE USO GENERAR REPORTE.....	43
TABLA 9: DESCRIPCIÓN DE CASO DE USO CONSULTAR EVALUACIÓN.....	43
TABLA 10 DESCRIPCIÓN DE CASO DE USO GRAFICAR REPORTES.....	44
TABLA 11: DESCRIPCIÓN DE CASO DE USO EXPORTAR REPORTES.....	45
TABLA 12: DESCRIPCIÓN DE DE LAS VARIABLES DEL DISEÑO DE CASOS DE PRUEBA.....	54
TABLA 13: DESCRIPCIÓN DEL CASO DE PRUEBA EVALUACIÓN POR TEMA.....	56
TABLA 14: DESCRIPCIÓN DEL CASO DE PRUEBA EVALUACIÓN POR NIVEL DE COMPLEJIDAD.....	58
TABLA 15: ANÁLISIS DE LOS RESULTADOS DE LAS PRUEBAS REALIZADAS.....	59

Introducción:

En la Universidad de las Ciencias Informáticas, es objetivo primordial el aprendizaje de las distintas materias que se imparten en esta casa de altos estudios. Las asignaturas que pertenecen al núcleo básico-específico de la carrera son complejas (como las de Programación), lo cual dificulta el aprendizaje por parte de los estudiantes. Estudios realizados a través de encuestas aplicadas a estudiantes de todos los años, constatan una falta de estudio sistemático y motivación por parte de los estudiantes hacia estas asignaturas. (FERREIRO, 2011)

Los docentes han buscado alternativas para darle solución a este inconveniente, desarrollando aplicaciones y juegos didácticos que mejoren el interés de los estudiantes por estas asignaturas, así como lograr aumentar la interacción estudiante-profesor.

En el año 2005 se comienza a utilizar en la universidad el Entorno Virtual de Aprendizaje(PRESSMAN), el cual según su página web es un espacio de apoyo al proceso de formación de la Carrera de Ingeniería en Ciencias Informáticas. En él los profesores pueden implementar estrategias de enseñanza-aprendizaje complementarias a las clases presenciales, así como diseñar cursos semipresenciales o totalmente a distancia, disponiendo los estudiantes de un medio en el cual pueden obtener, utilizar o compartir materiales didácticos.(TELEFORMACIÓN, 2013)

No obstante, es posible aprovechar de una mejor manera las potencialidades de las tecnologías de la información y las comunicaciones(INFORMÁTICA) para la educación, a través de herramientas educativas que favorezcan la motivación y ejercitación de contenidos. Es por ello que en el año 2011 es creado el sistema SMProg, un *software* que permite al profesor el diseño de juegos didácticos estilo tablero, así como la definición de sistemas de preguntas en función de los contenidos que este desea que los estudiantes ejerciten a través del juego, de forma tal que contribuya positivamente con el Proceso de Enseñanza-Aprendizaje (PEA) de la disciplina Técnicas de Programación de Computadoras (TPC).

Por otra parte es esencial destacar que el sistema *SMProg* solo cuenta con una variedad de juegos didácticos estilo tablero. A partir del año 2012 se comenzó el desarrollo de la segunda versión de la herramienta, concebida como una migración a una plataforma web, manteniendo los principios de

funcionamiento de la herramienta pero agregándole entornos tridimensionales y extendiendo su uso al resto de las disciplinas impartidas en la universidad.

Esta herramienta puede resultar muy valiosa para la ejercitación de los contenidos pero, es necesario tener una constancia de los resultados obtenidos por cada estudiante para mantener un recorrido, análisis y control del estudiante, así como dar seguimiento al estado y evolución del mismo, teniendo en cuenta la utilización de tablas, gráficas y reportes. En la actualidad la herramienta solo está concebida para ser utilizada por los estudiantes en la ejercitación de contenidos y por el profesor en la creación de juegos didácticos que serán usados como material para la ejercitación. Sin embargo, no se explotan las facilidades que ofrece SMProg como una herramienta para ayudar al profesor a determinar cuáles son los estudiantes que más atención diferenciada requieren, pues no se puede obtener la evaluación de los estudiantes en ninguno de los juegos ejercitados. Por tanto muchas de las estadísticas que pudieran ser útiles al profesor en la toma de decisiones con respecto a cómo dirigir el Proceso de Enseñanza-Aprendizaje frente a un grupo de estudiantes no son utilizadas.

Además, la herramienta no brinda la posibilidad de que el profesor conozca el desempeño individual de los estudiantes en la ejercitación de los contenidos, de ahí que no pueda ser utilizada como un referente para conocer el estado y/o evolución de un estudiante en un tema determinado. Esto imposibilita que se tengan en cuenta los resultados de los estudiantes para dirigir una atención diferenciada intencionada y mejorar los resultados docentes en temáticas específicas, o incluso desarrollar el nivel de conocimientos en estudiantes aventajados. Tampoco es posible recurrir a la herramienta para conocer cuáles son los temas que más dificultades generan en los estudiantes y a partir de ahí trazar estrategias de aprendizaje que puedan contribuir a elevar los resultados docentes, dejando esta tarea solo a los métodos tradicionales, sin utilizar SMProg como una herramienta para crear juegos didácticos que centren su atención en los problemas particulares de los estudiantes y no en cómo el profesor perciba los problemas.

A partir de la problemática existente se identifica el siguiente **problema de la investigación**:

¿Cómo facilitar el análisis de los resultados obtenidos a través de juegos didácticos en la herramienta SMProg 2.0?

El **objeto de estudio** lo constituyen los sistemas de reportes estadísticos, enmarcados en el **campo de acción** sistemas de análisis estadísticos docentes.

El **objetivo general** de la presente investigación se centra en: desarrollar un módulo que permita el análisis estadístico a partir de la configuración, cálculo y graficado de los resultados alcanzados durante la ejercitación de contenidos, utilizando juegos didácticos en la herramienta SMProg 2.0.

Para la realización de este módulo se han trazado los siguientes **objetivos específicos**:

- ✓ Construir el marco teórico referencial relativo al desarrollo de módulos para realizar análisis estadísticos de resultados docentes.
- ✓ Describir la propuesta de solución del módulo para el análisis estadístico de los resultados a través de juegos didácticos en la herramienta SMProg 2.0.
- ✓ Implementar las funcionalidades del módulo para el análisis estadístico de los resultados obtenidos durante la ejercitación de contenidos utilizando juegos didácticos en la herramienta SMProg 2.0.
- ✓ Verificar el correcto funcionamiento del módulo para el análisis estadístico de los resultados obtenidos durante la ejercitación de contenidos utilizando juegos didácticos en la herramienta SMProg 2.0.

Para dar cumplimiento a los objetivos trazados se proponen las siguientes **tareas de investigación**:

- Caracterización del estado del arte acerca de las herramientas para el análisis estadístico sobre información almacenada en los sistemas de gestión de bases de Datos (SGBD), en particular sobre PostgreSQL.
- Identificación de los recursos informáticos a utilizar para el desarrollo del módulo.
- Definición de la arquitectura del módulo.
- Identificación de los requisitos funcionales y no funcionales.
- Desarrollo del módulo para el análisis de los resultados docentes alcanzados durante la ejercitación de contenidos utilizando juegos didácticos.
- Realización de las pruebas funcionales al *software*.

- Configuración de al menos 5 reportes estadísticos diferentes que incluyan el graficado de datos.

Para realización de este documento se tuvieron en cuenta varios **métodos científicos** investigativos como:

Métodos teóricos:

Durante el desarrollo de la investigación se emplea el **Método Histórico–Lógico**, el mismo se utilizó para analizar la evolución a nivel nacional e internacional el uso de los sistemas informáticos que tengan similitud con el que se va a implementar.

Analítico-Sintético:

Partiendo del estudio de sistemas relacionados con el objeto de estudio se analizan sus principales características y funcionalidades, se sintetizan en una propuesta de solución que asegure el correcto funcionamiento del módulo.

Métodos empíricos:

Entrevistas: se utilizó en la realización de entrevistas a diferentes profesores involucrados en el proceso docente, para obtener toda la información necesaria acerca de qué funcionalidades serían imprescindibles en el desarrollo de la herramienta.

Observación: Se utilizó para conocer la forma de organización de los reportes en los sistemas estudiados y cómo pudieran ajustarse al campo de acción.

Justificación de la investigación

Con este trabajo se propone crear un módulo que permita la generación de reportes, mediante el cual el profesor consulte en tiempo real los resultados obtenidos por los estudiantes durante la interacción con juegos didácticos. La aplicación le permitirá al profesor tener constancia de una detallada información estadística de los resultados obtenidos durante la ejercitación de los contenidos utilizando juegos didácticos. Este resultaría de gran ayuda al profesorado de la universidad pues tendría la posibilidad de llevar un registro de los estudiantes, ver su evolución y de esta forma saber sobre qué deficiencias trabajar para lograr un mejor aprendizaje.

El presente trabajo está conformado por 3 capítulos que se describen a continuación:

Capítulo 1: Fundamentación teórica: se exponen los diferentes aspectos teóricos relacionados con el desarrollo de este trabajo. Se abordan conceptos fundamentales para un mejor entendimiento del mismo, el estado del arte, un análisis de las tecnologías en las cuales se sustentará el desarrollo del módulo, así como las herramientas que se utilizarán para el desarrollo del mismo.

Capítulo 2: Características del sistema: se detallan las características generales y específicas del módulo. Se describen los casos de uso, los requisitos tanto funcionales como no funcionales. Se define la arquitectura y se describe el sistema a través de diagramas de clases del análisis y diseño.

Capítulo 3: Implementación y prueba: se presenta el estándar de codificación a seguir para la implementación del módulo. Se diseña y ejecutan los casos de prueba basados en los requisitos. Se presenta la implementación de módulo a través de la distribución física de los componentes de hardware necesarios.

Capítulo 1: Fundamentación teórica

Introducción:

El control y seguimiento sistemático de los estudiantes por parte de los profesores, apoya en gran medida al Proceso de Enseñanza- Aprendizaje y estimula a los educandos a esforzarse más en el momento de ejercitar las materias. En el presente capítulo se describen algunos conceptos fundamentales para la comprensión de la investigación, además se abordarán estudios realizados a sistemas homólogos tanto en el ámbito nacional como en el internacional. Así como una fundamentación de las herramientas, lenguajes y metodologías utilizados para dar solución a la problemática presentada.

1.1 Conceptos inherentes al dominio del problema

A continuación se definen una serie de conceptos relacionados con la investigación realizada, los cuales permitirán una mejor comprensión del documento.

1.1.1 Proceso de Enseñanza-Aprendizaje

El proceso de enseñanza-aprendizaje es un proceso consciente, organizado y dialéctico de apropiación de los contenidos y las formas de conocer, hacer, vivir y ser, en el cual se producen cambios que le permiten al estudiante adaptarse a la realidad, transformarla y crecer como personalidad. (PERDIGÓN y MONTENEGRO, 2008)

1.1.2 Juegos didácticos

Constituyen un método que moviliza la actividad de variadas formas de organización de la enseñanza y propician el desarrollo de la capacidad cognitiva, práctica y variada de los conocimientos en forma activa y dinámica.(VALLEJO, 2002)

1.1.3 Generadores de reportes

Los generadores de reportes son herramientas complementarias de los sistemas de información. Utilizan una especie de lenguaje transparente para el usuario por medio del cual éste realiza consultas a la base de datos y obtiene información de ella en forma de reporte.(NAVAS, 2012)

1.2 Sistemas generadores de reportes

Para este estudio se consultaron varios sistemas emisores de reportes existentes en la actualidad, tanto en el ámbito nacional como internacional con el fin de observar las tendencias y técnicas de estos sistemas y que sirvan de apoyo para la confección del módulo para el análisis estadístico de los resultados obtenidos mediante la ejercitación con juegos didácticos. A continuación se hace una descripción de algunos de estos sistemas.

1.2.1 Generadores de reportes utilizados en el mundo.

1.2.1.1 Jasper Reports

Jasper Reports es una librería de código abierto para la creación de informes en Java, la misma permite la generación y gestión de reportes. Posibilita la creación de reportes que pueden ser presentados en la Web o se pueden crear ficheros en diversos formatos, como por ejemplo, PDF, HTML, XLS, ODT y RTF para que los datos puedan ser procesados con otras aplicaciones. También soporta diferentes fuentes de datos: PostgreSQL, MySQL, Oracle, MSSQL, ODBC, Interbase, Informix, XML, CSV. Este motor brinda además, la posibilidad de generar textos o imágenes de fondo para ser utilizados como marcas de agua por motivos de seguridad o simplemente para su identificación. Soporta varios tipos de gráficos. Admite la creación de reportes dentro de reportes, permite añadir enlaces en los reportes, lo que haría más fácil y rápida la navegación entre las diferentes secciones del informe cuando este es muy largo. *Jasper Reports* se usa comúnmente con *iReport*, un *front-end* gráfico (parte del *software* que interactúa con él o los usuarios) de código abierto para la edición de reportes. Con una solución de este tipo, se dispone en el tiempo deseado de los datos indispensables para lograr una gestión eficaz. También pueden seguirse fácilmente los resultados obtenidos y la manera en que la actividad progresa en función de los objetivos fijados. Pueden descubrirse las tendencias y los factores claves que afectan a la actividad y los resultados de la empresa. Las principales características de *Jasper Reports* son: genera informes y sub informes con múltiples niveles de anidamiento, librería de cálculos predefinidos, internacionalización, herramientas para diseño gráfico de los informes, fácil integración en aplicaciones. (REPORTS, 2013)

1.2.1.2 Crystal Reports

Es una herramienta fácil de usar para el diseño y generación de informes a partir de datos almacenados en una base de datos u otra fuente de información. Es una herramienta ampliamente utilizada en su categoría, no solo entre quienes podrían considerarse sus usuarios puros (aquellos que necesitan obtener periódicamente información para la toma de decisiones a partir de los datos de la empresa), sino también entre los programadores, que la han convertido en su herramienta favorita a la hora de embeber capacidades de generación de informes dentro de las aplicaciones a la medida. A este último hecho indudablemente ha contribuido mucho el que, desde hace más de diez años, Crystal Reports venga siendo incorporado de serie a las herramientas de desarrollo de Microsoft (Visual Basic y luego Visual Studio). Al igual que estas últimas mencionadas Crystal Reports es privativa. Entre sus características fundamentales se pueden destacar:

Diseño integral y opciones de formato: la tecnología flexible de diseño de *Crystal Reports* provee control completo sobre el acceso y la presentación de los datos en los reportes. Posee más de 100 opciones de formato, incluyendo parámetros, mapas, tablas cruzadas, gráficos e hipervínculos, para incrementar el impacto de los reportes. También se incluyen más de 160 fórmulas, funciones y operadores para un control completo de la presentación de los datos.

Productividad: el repositorio Crystal permite almacenar elementos clave de los reportes, tales como objetos de texto, imágenes, sentencias SQL y funciones personalizadas. Gracias a este repositorio central se pueden reutilizar estos objetos en múltiples reportes. Este almacén centralizado de objetos permite minimizar los esfuerzos de mantenimiento de los reportes y al mismo tiempo ser más productivo en el diseño de reportes nuevos.

De la misma manera, se pueden crear plantillas corporativas para ser aplicadas a todos los reportes y así evitar dar formato una y otra vez a cada uno de los nuevos reportes que se creen. (DANYSOFT, 2013)

1.2.1.3 MySchool

Es un *software* en línea para instituciones de educación superior. Abarca las áreas financiero - contable, administrativa, análisis de información académica y comunicacional (con los alumnos, padres y la comunidad escolar en su conjunto). El mismo aprovecha las tecnologías de última generación y cumple el

objetivo de apoyar la gestión de un establecimiento educacional, satisfaciendo los requerimientos de la comunidad escolar y del Ministerio de Educación del país o centro de estudios que contrate la herramienta (actas, certificados, informes, otros). Este *software* posee varias áreas, pero el estudio se centra en la académica pues es la que concierne directamente a los objetivos del trabajo. Su principal objetivo es brindar a los profesores y al colegio, las herramientas y facilidades, con el fin de apoyar la gestión docente. (MYSCHOOL, 2009) Entre los aspectos principales que MySchool cubre en esta área, están:

- Análisis estadístico de calificaciones por alumno, subsector, curso y colegio.
- Análisis gráfico de calificaciones.
- Exportación de datos a Excel y PDF.
- Información histórica.

1.2.2 Generadores de reportes utilizados en Cuba

1.2.2.1 Generador Dinámico de Reportes

El Generador Dinámico de Reportes (GDR), es una aplicación web para la gestión de la información de cualquier empresa o institución, mediante el cual se muestra el cumplimiento de las metas establecidas y contribuye a la toma de decisiones. Posibilita diseñar reportes tabulares (con gráficos incluidos), tabla pivote y cruzada desde los gestores de bases de datos: SQL Server, SQLite, Oracle, MySQL y PostgreSQL. Dicha herramienta permite a los usuarios abstraerse de los conocimientos relacionados con los gestores de bases de datos y generar reportes en varios formatos con gran variedad de opciones en su diseño, marcando una diferencia entre los reportes tradicionales y los reportes dinámicos. El Generador Dinámico de Reportes ha obtenido resultados tanto nacionales como internacionales, se encuentra actualmente instalado en más de 20 proyectos de la Universidad de las Ciencias Informáticas y entidades del país, ha sido contratado por varios clientes de la República Bolivariana de Venezuela. (MEDINA *et al.*, 2012)

1.2.2.2 EMIPAS

Es un *software* para la emisión de pasaportes que trabaja con los estándares de seguridad y calidad internacionales, en un ambiente amigable y flexible, con una interfaz sencilla e intuitiva. El mismo fue diseñado e implementado para la personalización de pasaportes de Lectura Mecánica, según las normas de la Organización Internacional de Aeronáutica Civil. Está compuesto por tres módulos, Personalización, Administración y Reportes, que garantizan la fiabilidad y seguridad de los datos.

Módulo de reportes

Proporciona un conjunto de informes orientados al control de los procesos de trabajo en cada turno y aporta la información necesaria para garantizar la dirección en todas las oficinas emisoras que componen el sistema. Cuenta con las siguientes características:

- Soportado sobre el sistema operativo Windows 2003 Server y el SGBD Oracle y Windows XP para las estaciones.
- No se establecen límites en la cantidad de información a registrar en la base de datos.
- Los procesos de personalización son escalables, las oficinas se configuran de acuerdo a la producción que se requiera.
- Los procesos de transmisión y recepción de información son seguros y se auto-controlan.
- Cuenta con un Plan de Preparación con su documentación y medios didácticos, para el personal designado a trabajar con el sistema.
- Control de Calidad y Supervisión, que garantizan fiabilidad y exactitud de la información contenida.
- Mecanismos de seguridad que personalizan los roles de cada usuario con acceso al sistema.
- Visualización del registro de todos los eventos ocurridos en el sistema y de las alertas sobre violaciones o eventos no previstos en su funcionamiento.(DATYS, 2013)

Conclusiones del estudio de homólogos

En el estudio de homólogos realizado se tomaron en cuenta las funcionalidades y particularidades de estos sistemas para el posterior desarrollo de la aplicación. Es importante destacar que tras el estudio de estas aplicaciones, ninguna posee un emisor de reportes específico para juegos didácticos. Otro de los aspectos que se tomó en cuenta es que, a pesar de la potencia en términos de funcionalidades que ofrecen muchos de estos sistemas, como es el caso del Crystal Reports y MySchool, estos son privativos, pero este último sirvió de base para la definición de funcionalidades básicas que se deben incluir en el sistema. Por otra parte aunque no se utiliza el EMIPAS por estar dirigido exclusivamente a la emisión de pasaportes este aportó una buena visión de cómo tratar la seguridad.

En otros sistemas la dificultad radica en la incompatibilidad de lenguajes y la dependencia de un soporte técnico como es el caso del Generador Dinámico de Reportes. Este sistema está desarrollado en Symfony, el cual es un *framework* (marco de trabajo) de PHP y utiliza como motor de reportes PHP Reports, que es un motor absoluto para PHP; otra de sus limitaciones es que presenta problemas en el graficado, además solo puede cargar un elemento por vez. Por otra parte una de las mayores limitantes es que, aunque sus desarrolladores aseguran que es posible la integración con otros sistemas, el nivel de complejidad de esta herramienta hace necesario el apoyo de un grupo de soporte para efectuar dicha integración.

Para el desarrollo de un módulo que permita el análisis de los resultados obtenidos durante la ejercitación de contenidos utilizando juegos didácticos, se decide utilizar el motor de reportes *Jasper Reports* ya que:

- Es uno de los que más formatos de salida y fuentes de datos soporta.
- Permite generar sub-reportes para facilitar el diseño.
- Brinda la posibilidad de generar textos o imágenes de fondo para utilizarlo como marcas de agua con el propósito de identificar el reporte o simplemente por motivos de seguridad.
- Es libre y multiplataforma.

- Ofrece diferentes componentes dentro de los cuales se encuentran tablas de contingencia dinámica, etiquetas, $2D^1$, $3D^2$.

1.4 Tecnologías para el desarrollo

Actualmente existen diversas tecnologías viables para el desarrollo de aplicaciones, en este caso específico: aplicaciones web. Se tuvieron en cuenta una serie de tecnologías con el fin de encontrar las más viables para el desarrollo de este módulo, tomando como punto de referencia las tendencias actuales y las que mejor se ajustan a las particularidades del proyecto.

1.4.1 Metodología de desarrollo de software

Las metodologías de desarrollo de *software* son un conjunto de técnicas, procedimientos y herramientas que ayudan a los desarrolladores a realizar un nuevo *software*. Indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, señalan además, qué papel tendrán las personas que estarán presentes en el desarrollo de las actividades. Detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. (Jacobson, 2000)

1.4.1.1 Metodología de desarrollo a utilizar

Esta investigación se deriva de la herramienta SMProg, por lo que el desarrollo es considerado como un proyecto de larga duración que necesita una estructura organizativa legible y bien documentada para que pueda ser seguida por posteriores desarrolladores.

En otro orden, es válido destacar que no se tiene un cliente definido ya que el producto está dirigido a una comunidad de profesores, no a una empresa en específico, por lo que se hace un tanto complejo contar con un cliente concreto dentro del equipo de desarrollo, así como realizar un levantamiento de requisitos lo más completo posible desde el comienzo.

¹ Bidimensional: Algo es **bidimensional** si tiene dos dimensiones, por ejemplo, ancho y largo, pero no profundidad. Los planos son bidimensionales, y sólo pueden contener cuerpos unidimensionales o bidimensionales.

² Tridimensional: un objeto o ente es **tridimensional** si tiene tres dimensiones. Por ejemplo ancho largo y profundidad.

Entre las principales metodologías utilizadas en la actualidad se encuentran: Proceso Unificado de Desarrollo (RUP), adaptable para proyectos de largo plazo que genera abundante documentación, aunque puede adaptarse para proyectos de menor alcance sin la necesidad de generar todos los artefactos; Programación Extrema (XP), que se recomienda para proyectos de corto plazo que involucren al cliente dentro del equipo de desarrollo, SCUM como apoyo a otras metodologías en la gestión de proyectos. Estas dos últimas asumen que la respuesta al cambio es parte importante del proceso de desarrollo.(JACOBSON *et al.*, 2000)

Teniendo en cuenta que es necesario realizar un estudio minucioso de las características del sistema, que permitan determinar, en ausencia de un cliente definido y por tanto ausente del equipo de desarrollo, los requisitos del sistema con la mayor severidad posible para evitar cambios importantes en el desarrollo, además de considerar esta investigación como una continuación o mejora de SMProg, el cual fue creado utilizando la metodología RUP, se decide utilizar la misma como metodología de desarrollo, aunque en este caso solo se generan algunos artefactos que esta metodología propone porque a pesar de ser un módulo pequeño no se cuenta con un cliente dentro del proceso de desarrollo. Por otra parte el equipo de desarrollo está compuesto casi en su totalidad por estudiantes, los cuales necesitan una mejor manera de organizar su trabajo dada su corta experiencia. Y no se considera viable que los requisitos sean inestables durante la elaboración de la propuesta, que son características propias de las metodologías ágiles, por tanto se escoge RUP para guiar el proceso.

1.4.1.1 Proceso Unificado del Rational (RUP)

El *Proceso Unificado de Rational* o RUP por sus siglas en inglés (*Rational Unified Processes*) es una metodología que permite el desarrollo de aplicaciones a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Presenta entre sus características ser un proceso de desarrollo orientado a objetos, utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de representación visual. Organiza los proyectos en términos de disciplinas y fases, consistiendo cada una en una o más iteraciones. Mantiene al equipo enfocado en producir incrementalmente *software* operativo a tiempo, con las características y calidad requeridas.(JACOBSON *et al.*, 2000)

Para el desarrollo del módulo se emitirán los artefactos:

- Diagrama de caso de uso.
- Descripción de caso de uso
- Diagrama de componente.
- Diagrama de comunicación.
- Diagrama de despliegue.

1.4.2 Herramientas CASE

Se puede definir a las Herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering, por sus siglas en inglés, o Ingeniería de *Software* Asistida por Ordenador) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software*. (Informática, Noviembre de 1999)

Se define también como: una innovación en la organización, un concepto avanzado en la evolución de tecnologías con un potencial de efecto profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de *software* y las metodologías de desarrollo de *software* formales.

1.4.2.1 Visual Paradigm 8.0

Visual Paradigm es una herramienta de modelado multiplataforma. Soporta el ciclo de vida completo de desarrollo de *software*, ayuda a una rápida construcción de aplicaciones de calidad y a obtener un menor costo en las mismas. Permite dibujar todos los tipos de diagramas de clases, obtener código inverso, generar código desde diagramas, así como la documentación que se necesite de cada uno de ellos Visual Paradigm hace uso de un lenguaje común a todo el equipo de desarrollo facilitando la comunicación entre los mismos. También presenta capacidades de ingeniería directa e inversa además de presentar disponibilidad de múltiples versiones para resolver cada necesidad. Es un producto de elevada claridad y altamente usado en la actualidad, pues es fácil de instalar, actualizar y manipular (INFORMÁTICA, Noviembre de 1999). Tomando como referencia la disponibilidad de la herramienta en múltiples plataformas (Windows, GNU/Linux). El diseño centrado en casos de uso enfocado al negocio que genera un *software* de mayor calidad, además de que en la UCI se cuenta con la licencia para el uso de la misma,

opción que facilita las condiciones de su uso, se selecciona Visual Paradigm 8.0 para el modelado del módulo.

1.4.3 Lenguaje Unificado de Modelado UML

El lenguaje unificado de modelado o UML (*Unified Modeling Language* por sus siglas en inglés) es el lenguaje estándar especificado por el OMG (*Object Management Group* por sus siglas en inglés) para visualizar, especificar, construir y documentar los artefactos de un sistema y además, sirve para el modelado del negocio y sistemas de *software*. Ofrece un estándar para describir los modelos, incluyendo aspectos conceptuales como procesos de negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. UML cuenta con un conjunto de notaciones y diagramas para modelar sistemas orientados a objetos y describe la semántica esencial de lo que estos diagramas y símbolos significan. En la actualidad UML y RUP se emplean de forma amplia en proyectos orientados a objetos de todos los tipos. (PRESSMAN, 1998)

1.4.4 Lenguaje de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo de cómputo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (SILVA, 2008)

1.4.4.1 Java 7

Java se creó como parte de un proyecto de investigación para el desarrollo de *software* avanzado para una amplia variedad de dispositivos de red y sistemas embebidos. La meta era diseñar una plataforma operativa sencilla, fiable, portable, distribuida y de tiempo real. Cuando se inició el proyecto, C++ era el lenguaje del momento, pero a lo largo del tiempo las dificultades encontradas en C++ crecieron hasta el punto en que se pensó que los problemas podrían resolverse mejor creando una plataforma de lenguaje completamente nueva. Se extrajeron decisiones de diseño y arquitectura de una amplia variedad de lenguajes como Eiffel, SmallTalk, Objective C y Cedar/Mesa. El resultado es un lenguaje que se ha mostrado ideal para desarrollar aplicaciones, distribuidas y basadas en red en un amplio rango de entornos.(BUSCHMANN *et al.*, 2001)

1.4.5 Servidor web

Un servidor web es un programa que sirve datos en forma de páginas web, hipertextos o páginas HTML (*HyperText Markup Language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP. Con esto, un servidor web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que se suele conocer como un navegador web. (GROUSSARD, 2012)

1.4.5.1 Apache Tomcat 7.0.34

Puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que su uso de forma autónoma era sólo recomendable para entornos de desarrollo, con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. El motor de *servlets*³ de Tomcat a menudo se presenta en combinación con el servidor web Apache. Tomcat constituye uno de los más completos contenedores de *Servlet* gratuito. Sin embargo, se trata de algo más que un servidor de pruebas, dado que muchas empresas lo emplean en la actualidad en entornos de producción debido a su contrastada estabilidad. (KHARE, 2012)

Características:

- Es un servidor HTTP y un contenedor de *servlets*.
- Es *software* libre (licencia Apache 2.0) gestionado por la fundación Apache.
- Puede funcionar como servidor HTTP o conectado a otro servidor HTTP como Apache HTTP Server o IIS.
- Puede ejecutar servicios web mediante Apache Axis.

³ Los *servlets* son objetos que corren dentro y fuera del contexto de un contenedor de *servlets* (ej: TomCat) BUSCHMANN, F.; MEUNIER, R., *et al.* *Pattern Oriented Software Architecture A system of patterns*. 2001. vol. 1, 459 p. ISBN 0 471 95889 7. y extienden su funcionalidad.

1.4.6 Framework (marco de trabajo)

1.4.6.1 Spring 3

Spring es un *framework* (marco de trabajo) de código abierto de desarrollo de aplicaciones para la plataforma Java, cuenta con un conjunto de librerías en las que es posible escoger aquellas que faciliten el desarrollo de la aplicación. Entre sus posibilidades está su contenedor de Inversión de Control (Inversión de Control, también llamado Inyección de Dependencias, es una técnica alternativa a las clásicas búsquedas de recursos vía JNDI. Permite configurar las clases en un archivo XML y definir en él las dependencias. De esta forma la aplicación se vuelve muy modular y a la vez no adquiere dependencias con Spring), la introducción de aspectos, plantillas de utilidades para Hibernate, iBatis y JDBC así como la integración con JSF. Es uno de los proyectos más sorprendentes en el panorama actual en Java por el grado en que ayuda a que los diferentes componentes que forman una aplicación trabajen entre sí; pero no establece apenas dependencias consigo mismo. Esta es la primera característica de este framework. Sería posible retirarlo sin prácticamente cambiar líneas de código. Lo único necesario es, lógicamente, añadir la funcionalidad que provee, ya sea con otro Framework similar o mediante el propio código. (HUNTER y CRAWFORD, 1998)

1.4.6 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado, llamado también IDE (siglas en inglés de *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. Un Entorno de Desarrollo Integrado es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. (HUNTER y CRAWFORD, 1998)

1.4.6.1 NetBeans 7.1

El Entorno de Desarrollo Integrado NetBeans es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, puede ser usado por cualquier otro lenguaje de programación. Consta de una gran base de usuarios y una comunidad en constante

crecimiento al igual que muchos otros sistemas libres. La plataforma *NetBeans* permite que las aplicaciones se desarrollen a partir de un conjunto de módulos o componentes de *software*. Es fácil de instalar, de uso instantáneo y se ejecuta en varias plataformas incluyendo Windows, GNU/Linux, Mac OS X y Solaris. El IDE NetBeans es un producto libre y sin restricciones de uso. NetBeans permite crear aplicaciones web con Java Web además de un tener compilador integrado. Presenta soporte para Spring lo que permite que más desarrolladores lo utilicen como IDE de desarrollo.(FERREIRO, 2011)

1.4.7 Sistema Gestor de Bases de Datos

1.4.7.1 PostgreSQL 8.4

Es un Sistema de Gestión de Bases de Datos objeto-relacional, multiusuario, centralizado y de propósito general, que está siendo desarrollado desde 1977 y liberado bajo la licencia de *Berkeley Software Distribution* (BSD).(Martínez, 2010).

Principales características:

- Organiza los datos mediante un modelo objeto-relacional.
- Capaz de manejar procedimientos, rutinas complejas y reglas.
- Ofrece transacciones que permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- Es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos declarados por el usuario; soporta además sobrecarga de operadores, sobrecarga de procedimientos, vistas, partición de tablas y datos.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de la información dentro de las bases de datos.
- Ofrece control de acceso simultáneo a través de la gestión de múltiples versiones de un mismo registro.

- Las restricciones y disparadores tienen la función de mantener la integridad y consistencia en las bases de datos.

1.4.8 Herramienta de administración de bases de datos

1.4.8.1 PgAdmin III

Es una aplicación gráfica para administrar bases de datos *PostgreSQL*, siendo utilizada con licencia *Open Source*. Está diseñada para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita la administración. Es desarrollado en todo el mundo por una comunidad de expertos de PostgreSQL y está disponible en más de una docena idiomas. Es de *software* libre liberado bajo la licencia de PostgreSQL. Tiene variedad de características entre las que se encuentran: Multiplataforma, diseñado para las versiones de PostgreSQL múltiples y derivados, una amplia documentación, acceso a los datos, acceso a todos los objetos PostgreSQL, soporta la mayoría de las codificaciones del lado del servidor. (MATHEW, 2005)

Conclusiones

En este capítulo se realizó un estudio del estado del arte, el cual sirvió para mostrar la necesidad de desarrollar el módulo para el análisis estadístico de los resultados obtenidos en la ejercitación de contenido con juegos didácticos. El estudio de homólogos mostró que la mayor parte de las aplicaciones que poseen sistemas de reportes no están enfocadas a los juegos didácticos, además de ser en su gran mayoría herramientas privativas, por lo que se seleccionó *Jasper Reports* como motor de reportes. Tras analizar los lenguajes, metodologías y herramientas de desarrollo a tener en cuenta, se decide emplear RUP como metodología de desarrollo de *software*, como herramienta CASE Visual Paradigm 8.0 utilizando como lenguaje de modelado UML, para el desarrollo se utilizará el lenguaje de programación Java 7 en el ambiente de desarrollo o IDE Net Bean 7.1 apoyado por el *framework* Spring, como servidor web se usará el Apache TomCat 7.0.34 y como gestor de bases de datos PostgreSQL 8.4.

Capítulo 2: Características del sistema

Introducción

En el presente capítulo se hace una descripción del levantamiento de requisitos tanto funcionales como no funcionales mediante el uso de los métodos utilizados para la identificación de los mismos. Se declaran los actores del sistema involucrados y se identifican los casos de uso que contienen las funcionalidades con las que debe contar el módulo a partir de la especificación de los requisitos funcionales, así como la descripción tanto resumida como extendida de los casos de uso. Además se hace el análisis y diseño del sistema.

2.1 Especificación de los requisitos de software

Los requisitos son cualidades o funcionalidades que el sistema debe cumplir para que tenga un correcto funcionamiento. Una buena captura de requisitos es fundamental para el desarrollo exitoso de la aplicación, pues ellos proveen al equipo de desarrollo de un entendimiento específico y común de las cualidades del sistema. El objetivo fundamental de la captura de los requerimientos del software es traducir de forma sencilla el problema tal y como lo ve el cliente, logrando un mejor entendimiento entre el usuario final y los desarrolladores. Este entendimiento sobre lo que debe y no debe hacer el sistema permite identificar las funcionalidades requeridas y las restricciones que se imponen. (JACOBSON *et al.*, 2000)

2.2 Técnicas de captura de requisitos

Dentro de las técnicas de captura de requisitos que se emplearon para la investigación se encuentran:

Entrevistas: Se utilizaron para obtener toda la información necesaria acerca de qué funcionalidades serían imprescindibles en el desarrollo de la herramienta. Las entrevistas a diferentes profesores involucrados en el proceso docente permitieron entender el dominio del problema y definir de una manera clara cómo resolver las necesidades.

Prototipos: El uso de prototipos es una herramienta valiosa para clarificar requisitos confusos. En la investigación se usan los prototipos para que el usuario entienda mejor qué información necesitan obtener del sistema y cómo visualizarla.

Tormenta de ideas: Esta técnica sirvió de apoyo al proceso de generación de ideas, mediante la cual se obtienen opiniones y señalamientos de posibles mejoras, así como la identificación de nuevos requisitos.

2.1.1 Requisitos funcionales (RF)

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir para darle solución al problema identificado. Ellos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. (Jacobson 2004).

A continuación se muestra el listado de requisitos funcionales obtenidos luego de una entrevista con el cliente dada sus necesidades y sugerencias.

RF 1: Obtener evaluación de un estudiante en un tema.

RF 2: Obtener evaluación de un estudiante en un nivel de complejidad asociado a un tema.

RF 3: Obtener promedio de evaluaciones de un estudiante.

RF 4: Listar los estudiantes suspensos en un tema.

RF 5: Listar los estudiantes con una nota específica en un tema.

RF 6: Obtener porciento de aprobados por tema.

RF 7: Obtener porciento de suspensos por tema.

RF 8: Graficar los resultados obtenidos en un juego sobre un tema.

RF 9: Exportar resultados a formato PDF.

2.1.2 Requisitos no funcionales (RNF)

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Los requerimientos no funcionales son fundamentales en el éxito del producto y normalmente están vinculados a requisitos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se

puede determinar cómo ha de comportarse, qué cualidades debe tener, cuán rápido o grande debe ser.(Jacobson 2004). Existen múltiples categorías para clasificar a los requisitos no funcionales, a continuación se muestran los más destacados en este sistema:

Apariencia o interfaz externa.

- Cada funcionalidad debe estar descrita explícitamente con un texto aclaratorio en correspondencia con la función que realiza.
- Las ventanas de trabajo deben ser sencillas, es decir no estar cargadas en su composición para que sea más fácil de entender por el usuario.
- Estará diseñada para una óptima visualización siendo adaptable a cualquier resolución.

Rendimiento.

- El sistema deberá atender y responder a una determinada petición por parte de los usuarios en un período de tiempo no mayor de 1 segundo, para mostrar los gráficos se considera eficiente un tiempo de respuesta de hasta 3 segundos.

Legales.

- Las herramientas de desarrollo deben ser herramientas libres o de código abierto en su mayoría y para las que no lo son, las licencias deben estar avaladas para su uso.

Restricciones de software.

- Máquina virtual de Java instalada versión 6.0 o superior.
- Servidor PostgreSQL como SGBD para establecer y mantener la base de datos, en su versión 8.4.

Restricciones de hardware.

En la PC servidor

- 40 GB o más de capacidad en el disco duro.
- Microprocesador de 2.4 GHz o superior.

- Memoria RAM de 2 GB o superior.
- Tarjeta de red de 100 Mbps o superior.

En la PC cliente

- 15 GB o más de capacidad en el disco duro.
- Microprocesador 1 GHz o superior
- Memoria RAM de 512 MB o superior.
- Tarjeta de red de 100 Mbps o superior.

Restricciones de la implementación.

- Se utilizará como lenguaje de programación el Java usando como IDE de desarrollo el NetBean 7.1.

Confiabilidad.

- La precisión en los reportes obtenidos debe ser la más exacta posible, midiéndose en términos de calidad y exactitud no así de cantidad.

Seguridad.

- Disponibilidad: Significa que a los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado. El sistema deberá permanecer en funcionamiento las 24 horas, permitiendo al usuario interactuar con el mismo en cualquier momento, deberá recuperarse de cualquier fallo del sistema además debe contar con un equipo de mantenimiento.

2.2 Modelado del sistema

2.2.1 Casos de uso del sistema

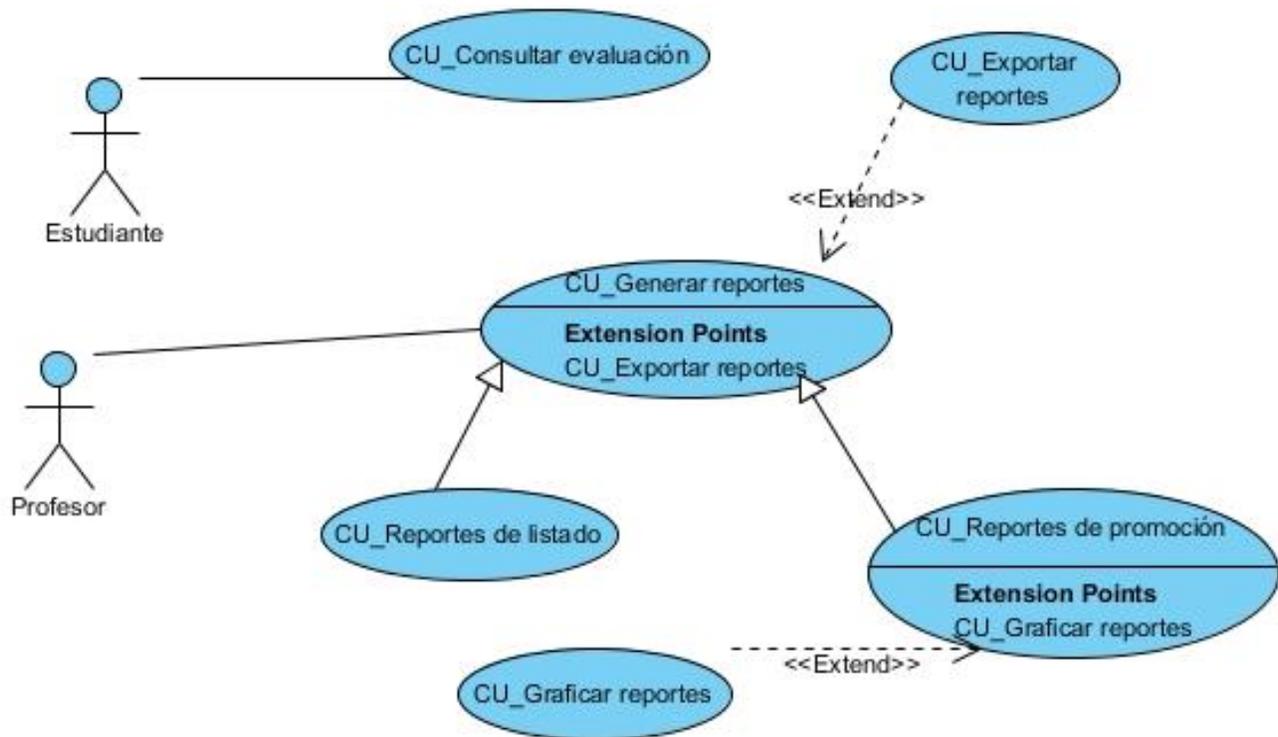


Ilustración 1 Diagrama de caso de uso

2.2.2 Actores del sistema

Un **Actor** es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema. (PRESSMAN, 1998)

Actores del sistema	Justificación
Profesor	Representa el rol encargado de realizar las peticiones de los diferentes tipos de reportes, ya sean de listado o de promoción. Tiene la posibilidad de graficar y exportar los reportes a formato PDF. Tiene acceso a todas las

	funcionalidades del módulo.
Estudiante	Representa el rol encargado de consultar sus evaluaciones en la ejercitación de contenidos con juegos didácticos. Solo tiene acceso a las funcionalidades Consultar evaluación por tema, por asignatura y por nivel de complejidad.

Tabla 1: Actores del sistema y su descripción

2.2.3 Resumen de descripción de los casos de uso del sistema

2.2.3.1 Resumen de caso de uso Generar reportes

Caso de uso	Generar reportes
Actor	Profesor
Resumen	El caso de uso comienza cuando el actor selecciona la opción Generar reportes. El actor selecciona el tipo de reporte que desea emitir. Si el actor selecciona la opción Reportes de listado. Ver caso de uso Reportes de listado. Si el actor selecciona la opción Reportes de promoción. Ver caso de uso Reportes de promoción.
Precondiciones	El usuario debe estar autenticado en el sistema.
Referencias	RF1, RF2, RF3, RF4, RF5, RF6, RF7

Prioridad	Crítica
------------------	---------

Tabla 2: Resumen del caso de uso: Generar reportes

2.2.3.2 Resumen de caso de uso Reportes de listado

Caso de uso	Reportes de listado
Actor	Profesor
Resumen	<p>El caso de uso comienza cuando el actor selecciona la opción Reportes de listado, mostrándose una interfaz en la que puede obtener los siguientes reportes:</p> <ul style="list-style-type: none"> • Obtener evaluación por tema • Obtener evaluación por nivel de complejidad. • Obtener estudiantes suspensos en un tema • Obtener estudiantes con una nota específica. <p>El sistema muestra los resultados de acuerdo al reporte seleccionado.</p> <p>Si el actor desea exportar alguno de estos reportes selecciona la opción Exportar a PDF: ver caso de uso Exportar reportes.</p>
Precondiciones	El usuario debe estar autenticado en el

	sistema.
Referencias	RF1, RF2, RF3
Prioridad	Crítica

Tabla 3: Resumen del Caso de uso: Reportes de listado.

2.2.3.3 Resumen de caso de uso Reportes de promoción

Caso de uso	Reportes de promoción
Actor	Profesor
Resumen	<p>El caso de uso comienza cuando el actor selecciona la opción Reportes de promoción. Mostrándose una interfaz en la que puede obtener los siguientes reportes:</p> <ul style="list-style-type: none"> • Obtener promedio por tema. • Obtener porciento de aprobados por tema. • Obtener porciento de suspensos por tema. <p>Si el actor selecciona la opción Graficar los resultados obtenidos en un juego sobre un tema: Ver caso de uso Graficar reportes. Si el actor desea exportar alguno de estos reportes selecciona la opción Exportar a PDF: ver caso de uso</p>

	Exportar reportes.
Precondiciones	El usuario debe estar autenticado en el sistema.
Referencias	RF4, RF5, RF6, RF7
Prioridad	Crítica

Tabla 4: Resumen del Caso de uso: Reportes de promoción

2.2.3.4 Resumen de caso de uso Exportar reportes

Caso de uso	Exportar reportes
Actor	Profesor
Resumen	El caso de uso comienza cuando el actor selecciona la opción Exportar reporte. El actor selecciona la ubicación de salva, se ejecuta la petición y se muestra un mensaje de éxito.
Precondiciones	El usuario debe estar autenticado en el sistema.
Referencias	RF1, RF2, RF3, RF4, RF5, RF6, RF7
Prioridad	Crítica

Tabla 5: Resumen del Caso de uso: Exportar reportes.

2.2.3.5 Resumen de caso de uso Graficar reportes

Caso de uso	Graficar reportes.
--------------------	--------------------

Actor	Profesor
Resumen	El caso de uso comienza cuando el actor selecciona la opción graficar reporte. El actor selecciona el tipo de gráfica que desea (barra o pastel), se ejecuta la petición y se muestra la grafica deseada, dándole la opción de exportarlo.
Precondiciones	El usuario debe estar autenticado en el sistema.
Referencias	RF4, RF5, RF6, RF7
Prioridad	Crítica

Tabla 6: Resumen del Caso de uso: Graficar reportes.

Caso de uso	Consultar evaluación
Actor	Estudiante
Resumen	El caso de uso comienza cuando el actor desea conocer sus evaluaciones personales en un nivel de complejidad, o tema determinados y selecciona una de estas opciones en el sistema.
Precondiciones	El usuario debe estar autenticado en el sistema.
Referencias	RF4, RF5, RF6, RF7
Prioridad	Crítica

Tabla 7 Resumen de Caso de uso: Consultar evaluación.

2.3 Descripción ampliada de los casos de uso del sistema

2.3.1 Descripción de caso de uso Generar reportes

CU1-Generar reportes	
Actores:	Profesor
Propósito:	Generar un reporte que puede ser de listado o de promoción.
Resumen: Permite al usuario generar un reporte específico que puede ser de listado o de promoción. Esto está sujeto a la acción que seleccione el usuario una vez autenticado en el sistema.	
Referencias:	RF1, RF2, RF3, RF4, RF5, RF6, RF7
Precondiciones	El usuario se debe haber autenticado previamente en el sistema SMProg 2.0, ninguna tabla de la base de datos puede estar vacía.
Acción del actor	Respuestas del sistema
1. El profesor selecciona del menú la opción generar reporte.	2. El sistema muestra las opciones de reporte que puede generar: <ul style="list-style-type: none"> • Reportes de listado • Reportes de promoción
3. El profesor selecciona la opción deseada.	4. Si el profesor selecciona la opción Reportes de listado, ver sección " Generar reportes de listado ". Si el profesor selecciona la opción Reportes de promoción, ver sección " Generar reportes de promoción ".
Sección <i>Generar reportes de listado</i>	
	1. Muestra las opciones de todos los reportes de

	listado posibles a generar por el módulo.
2. El profesor selecciona la opción de reporte de listado que desee generar	3. Muestra la interfaz con los campos correspondientes al reporte seleccionado
4. El profesor selecciona los campos requeridos y presiona el botón Ver reporte.	5. El sistema muestra una interfaz con los resultados correspondientes al reporte seleccionado y las opción de Exportar reporte a PDF
6. En caso de que el profesor seleccione la opción Exportar reporte a PDF ver caso de uso Exportar reporte a PDF	
Curso alternativo de los eventos	
	4.1 Si están vacíos o incompletos los campos, muestra un mensaje de error, el sistema regresa al paso 3 del curso normal de los eventos.
Sección Generar reportes de promoción	
	1. Muestra las opciones de todos los reportes de promoción posibles a generar por el módulo.
2. El profesor selecciona la opción de reporte de promoción que desee generar	3. Muestra la interfaz con los campos correspondientes al reporte seleccionado
4. El profesor selecciona los campos requeridos y presiona el botón Ver reporte.	5. El sistema muestra una interfaz con los resultados correspondientes al reporte seleccionado y las opciones de Exportar reporte a PDF y Graficar reporte.
6. En caso de que el profesor seleccione la opción Exportar reporte a PDF ver caso de uso Exportar reporte a PDF , si selecciona la opción Graficar reporte ver caso de uso Graficar reporte	
Curso alternativo de los eventos	
	4.1 Si están vacíos o incompletos los campos, muestra un mensaje de error, el sistema regresa al

	paso 3 del curso normal de los eventos.
--	---

Tabla 8 Descripción de caso de uso Generar reporte

2.3.2 Descripción de caso de uso Consultar evaluación

CU2-Consultar evaluación	
Actores:	Estudiante
Propósito:	Conocer la evaluación de un estudiante en un nivel de complejidad, tema o asignatura determinados.
Resumen: Permite al estudiante conocer sus evaluaciones personales en un nivel de complejidad, tema o asignatura determinados. Esto está sujeto a la acción que seleccione el usuario una vez autenticado en el sistema.	
Referencias:	RF1
Precondiciones	El usuario se debe haber autenticado previamente en el sistema SMProg 2.0 y haber jugado al menos una vez, ninguna tabla de la base de datos puede estar vacía.
Acción del actor	Respuestas del sistema
1. El estudiante selecciona del menú la opción Consultar evaluación.	2. El sistema muestra una interfaz con los campos necesarios para la consulta de las evaluaciones.
3. El estudiante selecciona los campos requeridos para la consulta de la evaluación y presiona el botón Ver evaluación	4. El sistema muestra una interfaz con los resultados correspondientes al reporte seleccionado.
Curso alterno de los eventos	
	4.1 Si están vacíos o incompletos los campos, muestra un mensaje de error, el sistema regresa al paso 3 del curso normal de los eventos.

Tabla 9: Descripción de caso de uso Consultar evaluación

2.3.3 Descripción de caso de uso Graficar reportes

CU3-Graficar reportes	
Actores:	Profesor
Propósito:	Graficar los reportes de promoción según la selección del profesor.
Resumen: Permite al profesor observar de una manera gráfica los resultados de los reportes de promoción.	
Referencias:	RF8
Precondiciones:	El usuario se debe haber autenticado previamente en el sistema SMProg 2.0 y haber jugado al menos una vez, ninguna tabla de la base de datos puede estar vacía.
Acción del actor	Respuesta del sistema
	1. El sistema muestra una interfaz con las opciones de graficado de reportes que se brindan
2. El profesor selecciona la opción deseada y presiona el botón Ver gráfico	3. Muestra una interfaz con el tipo de gráfico especificado por el usuario.

Tabla 10 Descripción de caso de uso Graficar reportes

2.3.4 Descripción de caso de uso Exportar reportes

CU4-Exportar reportes	
Actores:	Profesor

Propósito:	Exportar los reportes según la selección del profesor.
Resumen: Permite al profesor exportar los reportes a PDF.	
Referencias:	RF1, RF2, RF3, RF4, RF5, RF6, RF7
Precondiciones:	El usuario se debe haber autenticado previamente en el sistema SMProg 2.0 y haber generado al menos un reporte.
Acción del actor	Respuesta del sistema
	1. El sistema muestra una interfaz con la vista previa del reporte en formato PDF y la opción de seleccionar dónde se va a guardar el archivo.
2. El usuario selecciona la ubicación donde desea guardar el archivo y pulsa el botón Guardar.	3. Muestra un mensaje de éxito en la operación.

Tabla 11: Descripción de caso de uso Exportar reportes

2.4 Arquitectura de software

Una arquitectura de *software*, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del *software* para un sistema de información. La Arquitectura de *Software* establece los fundamentos para que analistas, diseñadores, probadores, programadores, y el resto de los integrantes del equipo o grupo de desarrollo trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. Una arquitectura de *software* se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad que es la propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez se ha presentado un evento de falla, la flexibilidad y

escalabilidad, que es la capacidad de un sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes. (PRESSMAN, 1998)

2.4.1 Patrón Modelo Vista Controlador

Es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control entre componentes distintos. Su principal finalidad es mejorar la reusabilidad y que las modificaciones en las vistas impacten en menor medida en la lógica de negocio o de datos.

El Modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema).
- Llevar un registro de las vistas y controladores del sistema.
- Ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

El Controlador es responsable de:

- Recibir los eventos de entrada.
- Contiene reglas de gestión de eventos, estas acciones pueden suponer peticiones al modelo o a las vistas.

Las Vistas son responsables de:

- Recibir datos del modelo y lo muestra al usuario.
- Tienen un registro de su controlador asociado.
- Pueden dar el servicio de “Actualización”, para que sea invocado por el controlador o por el modelo cuando es un modelo activo.
- Es un patrón que convierte una aplicación en un paquete mantenible, modular y de desarrollo rápido. La modularidad y el diseño independiente permiten a los desarrolladores y diseñadores hacer cambios en alguna parte de la aplicación sin afectar a los demás. (González, 2012)

En la aplicación se evidencia el uso de este patrón ya que se separan las clases por distintas capas:

- En la capa de la vista se encuentran las clases JSP Y CSS agrupadas en la carpeta *jsp* las cuales son las encargadas de interactuar con el usuario.

- La capa del modelo se encuentran separadas en la carpeta *dao* las cuales Contienen las consultas de acceso a datos que encierra una serie de clases con métodos básicos predefinidos para el acceso a datos.
- Las clases controladoras se encuentran en la carpeta *controller*.

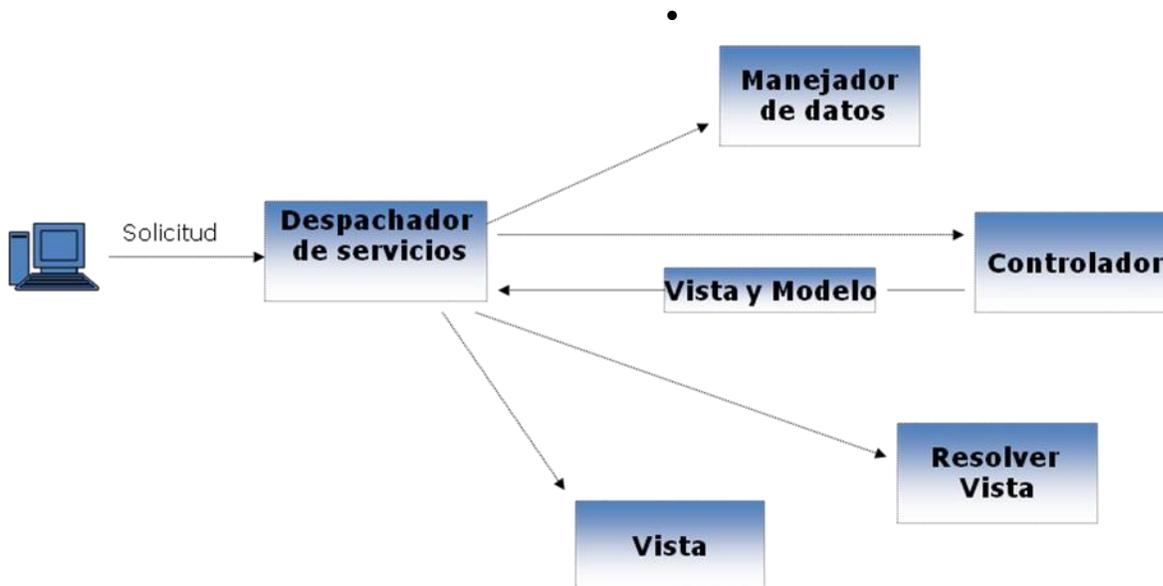


Ilustración 2: Arquitectura MVC en Spring

2.5 Clases del análisis

Las clases de análisis son una abstracción de una o varias clases y/o subsistemas del diseño del sistema que se centra en el tratamiento de los requisitos funcionales, ofrecen una interfaz en términos de operaciones, definen atributos normalmente conceptuales y reconocibles en el dominio del problema, participan en relaciones más conceptuales que en el diseño y la implementación. Los estereotipos de las clases de análisis son interfaz, control y entidad, cada uno expresa una semántica específica que constituye un método potente para identificar y describir las clases del análisis. (Larman, 1999)

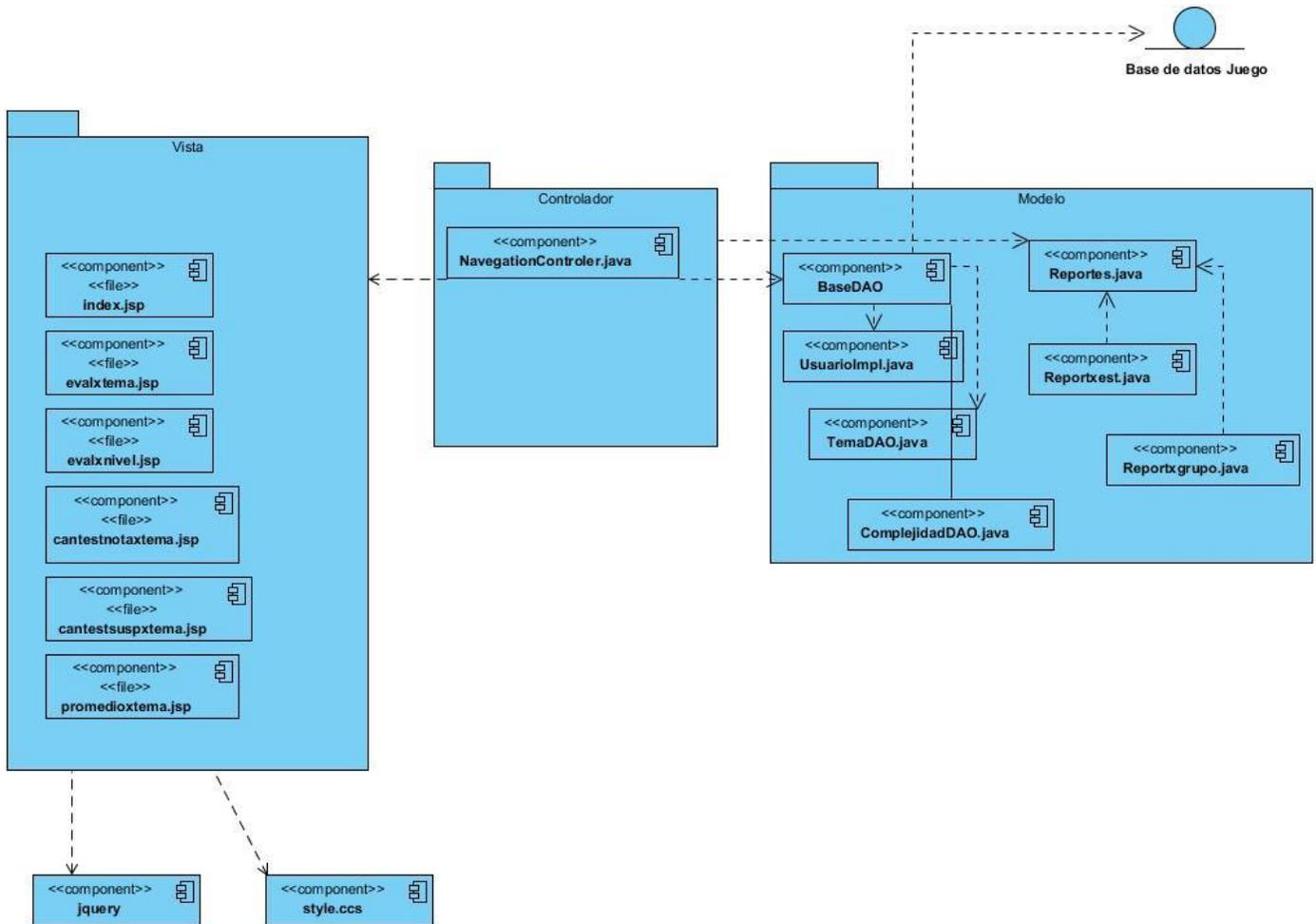


Ilustración 4: Diagrama de despliegue

2.6 Diagrama de despliegue

El Diagrama de Despliegue es un diagrama que se utiliza para modelar el *hardware* utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Describen la topología del sistema la estructura de los elementos de *hardware* y el *software* que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP, microondas, etc.

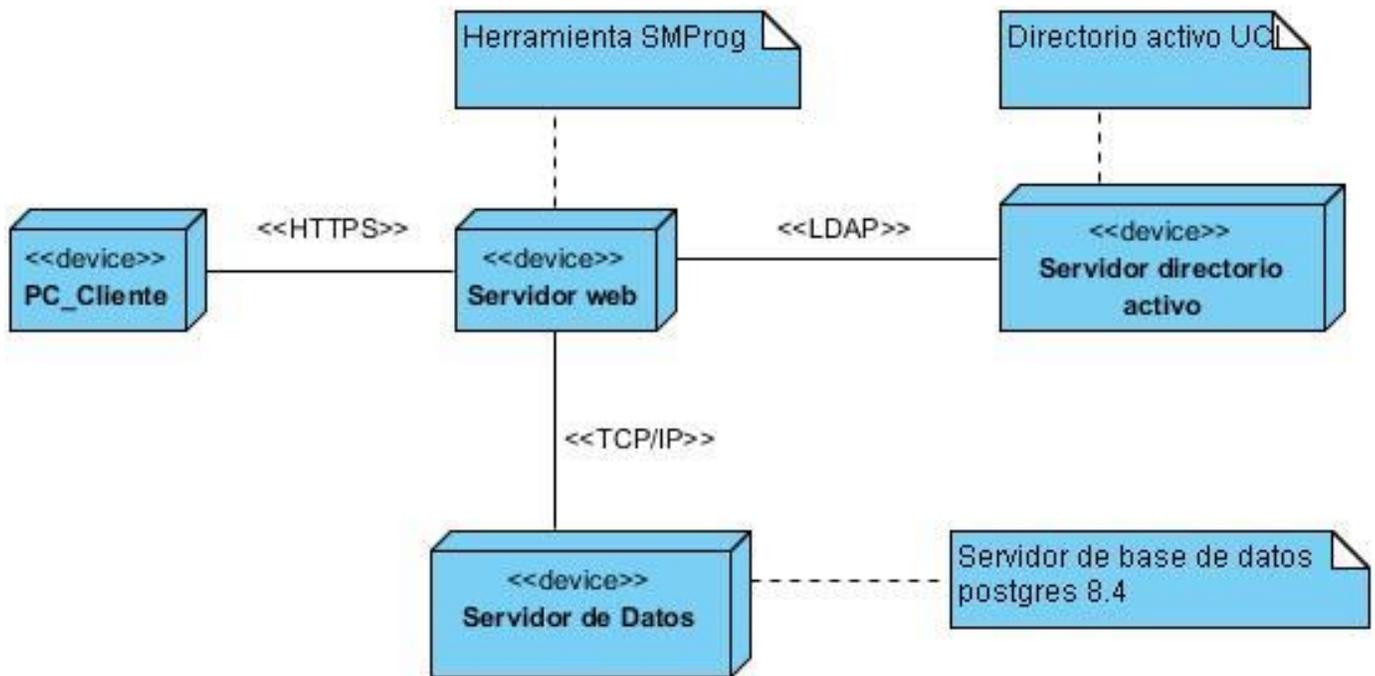


Ilustración 5: Modelo de datos

El usuario al ingresar al sistema desde una PC cliente se conectará al servidor web de la aplicación SMProg 2.0 mediante un protocolo HTTPS la cual permite una conexión de mayor seguridad. Una vez en el sistema, el cliente deberá autenticarse contra el servidor del directorio activo mediante el protocolo LDAP para poder acceder al módulo de reportes. Ya establecida la autenticación del usuario tendrá la opción de generar reportes mediante consultas a la base de datos. Por lo cual el servidor conectará con el servidor de datos mediante un protocolo TCP/IP.

2.7 Modelo de datos

Un modelo de datos es la combinación de una colección de estructuras de datos, operadores o reglas de inferencia y de reglas de integridad, las cuales definen un conjunto de estados consistentes. El cual puede ser usado como una herramienta para especificar los tipos de datos y la organización de los mismos. Además para la manipulación de consultas y datos, así mismo es el elemento clave en el diseño de la arquitectura de un manejador de base de datos.

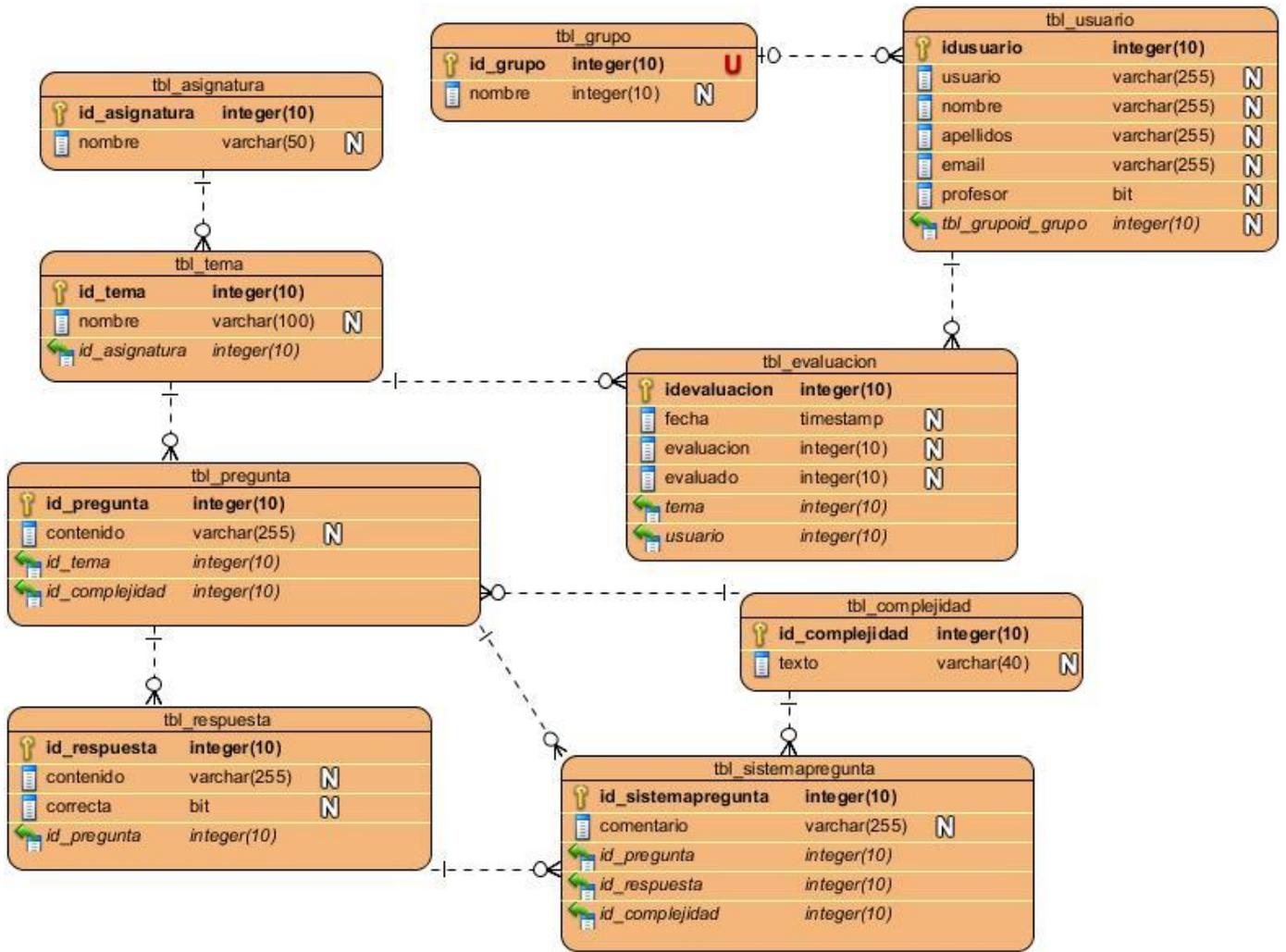


Ilustración 6: Diagrama de componentes

Conclusiones parciales

En este capítulo se realizó un profundo análisis de las características del sistema, que permitió obtener el conjunto de funcionalidades que debe cumplir el módulo. Se describieron los casos de uso, se propuso la arquitectura a utilizar, y el modelo de datos. Estos facilitarán que se cuente con los componentes y elementos necesarios para realizar la implementación. Además quedaron descritos los requisitos no funcionales que apoyarán el desarrollo del sistema.

Capítulo 3: Implementación y prueba

Introducción

En el presente capítulo se hace una descripción de los estándares de codificación que se emplearon para la implementación de módulo, además la descripción de los casos de prueba así como los resultados que estos arrojaron mediante los cuales se valida la propuesta de solución.

3.1 Estándares de codificación

Para lograr un mejor entendimiento del código en la implementación de la aplicación es necesario establecer un estándar de codificación a usar. Los estándares de codificación no son más que políticas aplicadas para permitir un mejor entendimiento en el código de un sistema. Estos estándares permitirán que el sistema sea totalmente independiente del autor y cualquier persona, ya sea desarrollador o no, pueda entender lo que plasmó el mismo. Con la aplicación de estos estándares a cualquier sistema se logra que se mantenga un estilo en la programación y que sirva de referencia a otros programadores para futuras integraciones o mantenimiento.

Norma general:

Todo el código de este sistema se desarrollará en el lenguaje de programación Java 7, apoyado por el *framework* Spring 3.

Reglas a seguir

- Para la declaración de los métodos en el código del sistema se realizará con la primera letra en mayúscula. Ejemplo (Metodo, Submetodo).
- Los nombres de las clases deben ser sustantivos o frases en forma sustantiva que no pueden usar prefijos.
- Las clases y paquetes del sistema se nombran en inglés.
- Las clases quedarán organizadas por los paquetes que representen su funcionalidad en el sistema.

- Para los parámetros y campos no constantes se deben usar nombres descriptivos que deben ser lo suficientemente descriptivos para determinar el significado de la variable y su tipo, preferentemente su significado.
- Las propiedades deben ser nombradas utilizando sustantivos o frases en forma sustantiva considerando siempre que sean nombradas con el mismo nombre de su tipo.
- Para entrada de datos se recomienda no poner prefijos. Por ejemplo en el caso de parámetros para métodos no sería necesario declararlos como p_Nombre.

3.2 Métodos de prueba.

Para verificar que un *software* funciona correctamente, es imprescindible considerar parámetros y estadísticas que brindan información acerca de la calidad del mismo. Existen métodos de pruebas diseñados para comprobar el rendimiento de la aplicación de los cuales se utilizó el método de pruebas de cajas negras para llevar a cabo las pruebas necesarias en la calidad de la propuesta de solución como sigue a continuación:

3.2.1 Prueba de caja negra.

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. Estas son las pruebas más comunes y conocidas en el caso de Transmisiones también son las más aplicadas ya que son realizadas por todo el equipo de desarrollo del software. Son guiadas por los casos de uso y los conocidos casos de pruebas, se enfocan en verificar que la aplicación cumple con lo especificado en la descripción de los casos de uso y en que la interfaz de la aplicación sea agradable al usuario y cumpla con las pautas de diseño impuestas por el proyecto. (Pressman, 2010 #2)

3.3 Diseño de los casos de prueba

Las pruebas que se realizaron a la aplicación están enfocadas a dar validez a la propuesta de solución. El tipo de prueba empleada fue pruebas de caja negra (sobre la interfaz) la cual cumple la función de detectar no conformidades de tal manera que se le puedan dar solución en horas tempranas. A continuación se presentan solo dos casos de prueba ya que no es relevante mostrarlos todos dada la similitud entre los mismos.

3.3.1 Descripción las variables

Las variables V1, V2, V3, V4, V5 y V6 que se representan en la siguiente tabla, significan los valores de entrada de datos para los casos de pruebas.

No	Nombre del campo	Clasificación	Valor nulo	Descripción
V1	Seleccione el grupo	Select	No	Permite la selección de un grupo para filtrar el rango de estudiantes.
V2	Seleccione el nombre	Select	No	Cadena de caracteres, nombre del estudiante del que se desea saber la evaluación.
V3	Seleccione una asignatura	Select	No	Tiene definido por defecto una serie de asignaturas de las cuales una será a la que pertenecerá la evaluación a reportar.
V4	Seleccione el tema	Select	No	Tiene una lista de temas del cual será extraída la evaluación.
V5	Seleccione el nivel	Select	No	Lista los distintos niveles de complejidad de las preguntas.
V6	Seleccione la nota	Select	No	Permite la selección de una nota para definir la búsqueda.

Tabla 12: Descripción de de las variables del diseño de casos de prueba.

3.3.2 Diseño de casos de prueba

- V: indica válido
- I: indica inválido

3.3.2.1 Descripción del caso de prueba Evaluación por tema.

Escenario	Descripción	V1	V2	V3	V4	Respuesta del sistema	Flujo central
1.1 Obtener evaluación por tema seleccionando los campos correctamente.	El usuario selecciona los valores correspondientes en cada uno de los campos del formulario y presiona el botón ver reporte	V (1206)	V (Juan Pérez)	V (Programación 1)	V (Polimorfismo)	Muestra una interfaz con los resultados del reporte solicitado por el profesor.	1-Hacer clic en Obtener evaluación por tema. 2- El usuario selecciona los campos y presiona el botón "Ver reporte". 3- El sistema muestra una vista previa del reporte y se le da la opción de exportar.

<p>1.2 Obtener evaluación por tema dejando todos los campos vacíos.</p>	<p>El usuario deja todos los campos del formulario y presiona el botón ver reporte</p>	<p>I(vacío)</p>	<p>I(vacío)</p>	<p>I(vacío)</p>	<p>Muestra una notificación de error informando que ningún campo debe estar vacío.</p>	<p>1-Hacer clic en Obtener evaluación por tema. 2- El usuario no llena los campos y aun así presiona el botón "Ver reporte". 3- El sistema muestra una notificación de error "debe llenar todos los campos".</p>
<p>1.3 Obtener evaluación por tema dejando uno de los campos vacío.</p>	<p>El usuario deja uno de los campos del formulario vacío y presiona el botón ver reporte</p>	<p>V(1206)</p>	<p>I(vacío)</p>	<p>V(Matемática I)</p>	<p>V(Integrales)</p>	<p>Muestra una notificación de error informando que ningún campo debe estar vacío.</p> <p>1-Hacer clic en Obtener evaluación por tema. 2- El usuario no llena uno de los campos y aun así presiona el botón "Ver reporte". 3- El sistema muestra una notificación de error "debe llenar todos los campos".</p>

Tabla 13: Descripción del caso de prueba evaluación por tema.

3.3.2.2 Descripción del caso de prueba Evaluación por nivel de complejidad.

Escenario	Descripción	V1	V2	V3	V4	V5	Respuesta del sistema	Flujo central
1.1 Obtener evaluación nivel de complejidad llenando los campos correctamente.	El usuario introduce los valores correspondientes en cada uno de los campos del formulario y presiona el botón ver reporte	V (1206)	V (Juan Pérez)	V (Matemática I)	V (Integrales)	V (Alta)	Muestra una vista previa del reporte y la opción de exportar.	1-Hacer clic en Obtener evaluación nivel de complejidad 2- El usuario se le da la opción de presionar el botón "Ver reporte". 3- El sistema muestra una vista previa del reporte y se le da la opción de exportar.
1.2 Obtener evaluación nivel de complejidad dejando todos los campos vacíos.	El usuario deja todos los campos del formulario vacíos y presiona el botón ver reporte	l(vacío)	l(vacío)	l(vacío)	l(vacío)	l(Vacío)	Muestra una notificación de error informando que ningún campo debe estar vacío.	1-Hacer clic en Obtener evaluación nivel de complejidad. 2- El usuario no llena los campos y así presiona el botón "Ver reporte". 3- El sistema muestra una notificación de error "debe llenar todos los campos".

1.3	El usuario deja uno de los campos del formulario vacío y presiona el botón "Ver reporte"	V (1206)	V (Pedro Hernán dez)	V (Matemática)	V (Trigonometría)	I(vacío)	Muestra una notificación de error informando que ningún campo debe estar vacío.	1-Hacer clic en Obtener evaluación nivel de complejidad. 2- El usuario no llena uno de los campos y aun así presiona el botón "Ver reporte". 3- El sistema muestra una notificación de error "debe llenar todos los campos".
------------	--	-------------	-------------------------------	-------------------	----------------------	----------	--	---

Tabla 14: Descripción del caso de prueba Evaluación por nivel de complejidad.

3.4 Análisis de los resultados de las pruebas realizadas

Una vez realizados los casos de pruebas estos arrojan no conformidades en cada una de sus tres iteraciones, lo cual enfoca los esfuerzos en dar solución a estas.

No de la iteración	Fecha	Casos de prueba	No conformidades			Estado		
			S	NS	R	RA	PD	NP
1	15-05-2013	Generar reportes	-	-	-	-	-	-
2	19-05-2013		-	-	-	-	-	-
3	21-05-2013		-	-	-	-	-	-
1	15-05-2013	Consultar evaluación	-	2	2	1	1	-
2	19-05-2013		-	1	-	1	-	-
3	21-05-2013		-	-	-	-	-	-
1	15-05-2013	Reportes de listado	-	4	3	3	1	-
2	19-05-2013		-	1	-	1	-	-

3	21-05-2013		-	-	-	-	-	-
1	15-05-2013	Reportes de promoción	1	3	1	2	2	-
2	19-05-2013		-	2	-	2	-	-
3	21-05-2013							
1	15-05-2013	Graficar reportes	-	2	2	2	-	-
2	19-05-2013		-	-	-	-	-	-
3	21-05-2013		-	-	-	-	-	-
1	15-05-2013	Exportar reportes	-	2	1	1	1	-
2	19-05-2013		-	1	-	1	-	-
3	21-05-2013		-	-	-	-	-	-
No conformidades: S – Significativa NS – No significativa R- Recomendación Estado: RA – Resuelta PD – Pendiente NP - No procede								

Tabla 15: Análisis de los resultados de las pruebas realizadas.

3.4.5 No conformidades detectadas.

- **Caso de prueba Generar reportes.**

No se detectaron no conformidades

- **Caso de prueba Consultar evaluación.**

1. No se le debe mostrar el campo de selección de estudiantes al usuario estudiante.
2. En el formulario evaluación por nivel, el campo de nivel no debe permitir entrada de números.

- **Caso de prueba Reportes de listado.**

1. El formulario promedio no permite filtrar el campo de búsqueda de los estudiantes por grupo.
2. Mostrar en los campos del formulario "Evaluación por tema" la opción de exportar reporte.
3. El formulario "Evaluación por tema" no permite filtrar la selección de temas por asignaturas.

4. Los campos de selección deben iniciarse en blanco.

- **Caso de prueba Reportes de promoción.**

1. El formulario por ciento de suspensos por tema no permite filtrar el campo de búsqueda de los temas por asignatura.

2. La tipografía de los enunciados de los reportes de promoción es distinta al de los de listado.

3. Estandarizar los campos de selección del formulario "Estudiantes con una nota en un tema".

4. Los botones del menú vertical correspondientes a los reportes de promoción no se muestran activos.

- **Caso de prueba Graficar reportes.**

1. Las gráficas no se corresponden con los resultados obtenidos en el reporte.

2. Corregir los colores de las gráficas de pastel.

- **Caso de prueba Exportar reportes.**

1. Corregir tipografía de las plantillas de reportes.

2. No permite la selección de la ubicación de los archivos exportados.

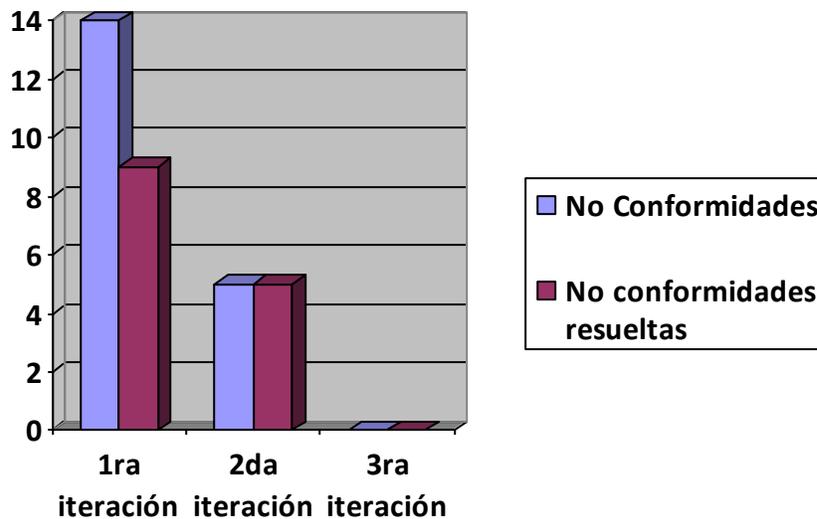


Ilustración 7: Resultados de las iteraciones de los casos de prueba.

Como se puede observar en la ilustración número 7 las pruebas realizadas se llevaron a cabo en tres iteraciones donde en la primera iteración se detectaron catorce no conformidades de las cuales solamente fueron detectadas dos de carácter significativo. En la segunda iteración se detectaron cinco no conformidades entre las cuales no figuran ninguna significativa, la tercera iteración no mostró ninguna no conformidad, concluyendo así que las pruebas de funcionalidad en el sistema fueron realizadas satisfactoriamente quedando listo para su funcionamiento.

Conclusiones parciales

Con las pruebas realizadas al *software* se verifica su correcto funcionamiento. El mismo permite la configuración de reportes estadísticos que facilitan a los profesores el análisis de de los resultados obtenidos por los estudiantes durante la ejercitación de contenidos utilizando juegos didácticos.

Conclusiones

Con el desarrollo de esta investigación se dio cumplimiento a los objetivos trazados de una manera satisfactoria llegando a la conclusión de que:

- La versión anterior del sistema SMProg carece de un emisor de reportes que le permita al profesor tener constancia y control de los resultados obtenidos a través de la ejercitación de contenidos usando juegos didácticos, de ahí que se demuestre la necesidad de contar con un módulo que facilite el proceso de análisis de los resultados a través de reportes y gráficas.
- Se implementó la propuesta de solución dando paso a una herramienta de generación de reportes capaz de proporcionar al profesor la posibilidad de graficar los mismos y comparar resultados en la ejercitación con juegos didácticos.
- El Módulo para el análisis de los resultados obtenidos durante la ejercitación utilizando juegos didácticos representa un paso de avance en el aprovechamiento de las tecnologías en función de la mejoría del Proceso de Enseñanza-Aprendizaje en la Universidad de las Ciencias Informáticas.

Recomendaciones

Se recomienda:

- Se amplíe la gama de de formatos a exportar.
- La inclusión de nuevos tipos de gráficas
- Agregar nuevos tipos de consultas para ampliar de esta manera las funcionalidades del sistema.

Bibliografía:

BUSCHMANN, F.; MEUNIER, R., *et al. Pattern Oriented Software Architecture A system of patterns*. 2001. vol. 1, 459 p. ISBN 0 471 95889 7.

DANYSOFT. *Danysoft* Danysoft Internacional, Última actualización: 1ro de marzo del 2013. [Consultado el: 13 de febrero del 2013 de 2013]. Disponible en: <http://www.danysoft.com/>.

DATYS. *DATYS Tecnología y sistemas* DATYS Tecnología y Sistemas. ed. Cuba: Disponible en: <http://www.datys.cu/wpinfoproducto.aspx?5>.

FERREIRO, D. S. *Herramienta para la creación y uso de juegos didácticos en la enseñanza de la Programación*. Tutor: Rasúa, R. a. T. Maestría, Departamento Técnicas de Programación. Universidad de las Ciencias Informáticas, 2011.

GROUSSARD, T. *Java 7 los fundamentos del lenguaje Java*. Editado por: Informáticos, C. R. Estela Dechenaud ed. 2012. ISBN 978-2-7460-7318-0.

HUNTER, J. y CRAWFORD, W. *Java Servlet Programming*. O'Reilly & Associates, Inc, 1998. ISBN 1-56592-391-X.

INFORMÁTICA, S.-J. D. *Herramientas Case*. 300 Ejemplares, Noviembre de 1999. 53 p. Colección Cultura linformática. ISBN 875-99-OI-OTDETI-INEI.

JACOBSON, I.; BOOCH, G., *et al. Proceso unificado de desarrollo de software*. Traducido por: Español. Editado por: Wesley, A. España: 2000. 464 p. La guía completa del proceso unificado escrita por sus creadores. ISBN 84-7829-036-2.

KHARE, T. *Apache Tomcat 7 Essentials*. Community Experience Distiller ed. PARKT, 2012. ISBN 978-1-84951-662-4.

MATHEW, N. *Beginning Databases whit PostgresSQL*. 2005. 589 p. ISBN 978-1-59059-478-0.

MEDINA, A. J. A.; RAMOS, M. L., *et al.* Generador dinámico de reportes. *Serie Científica de la Universidad de las Ciencias Informáticas*, 15 de noviembre 2012, vol. 5, nº p. 6. Disponible en: <http://publicaciones.uci.cu/index.php/SC>. ISSN 2353.

MYSCHOOL, C. *MySchool Software* MySchool Software S.A, [Consultado el: 2 de febrero de 2013]. Disponible en: www.myschool.cl/.

NAVAS, J. *Generador de reporte* Disponible en: <http://www.datapronet.com>.

PERDIGÓN, A. M. L. y MONTENEGRO, S. L. Ambientes informáticos para la impartición de la maestría en ciencias. *6to Congreso Internacional de Educación Superior*, 15-02-2008 2008, nº p. 14.

PRESSMAN, R. *Ingeniería del software un enfoque práctico*. Editado por: Hill, M. G. New York: 1998. vol. 6, ---.

Software Engineering. Editado por: Education, H. Boston Burr Ridge New York San Francisco London Madrid Mexico City Santiago Seoul Singapore Sydney Taipei Toronto: McGraw-Hill, 2010. vol. 7, 889 p. A Practitioner's Approach. ISBN 978-0-07-337 597 -7.

REPORTS, C. J. *JASPERSOFT COMMUNITY the intelligence inside* Jaspersoft Corporation. ed. Última actualización: 2 de marzo del 2013. Disponible en: www.community.jaspersoft.com/.

SILVA, S. *Definición de lenguajes de programación. Tipos. Ejemplos* UNEFA, Disponible en: <http://catedraprogramacion.foroactivos.net/>.

TELEFORMACIÓN, D. D. *Entorno Virtual de Aprendizaje* Universidad de las Ciencias Informáticas La Habana Cuba: Disponible en: <http://eva.uci.cu/>.

VALLEJO, J. M. B. *El juego didáctico como estrategia de atención a la diversidad* Disponible en: http://www.uhu.es/agora/version01/digital/numeros/04/04-articulos/miscelanea/pdf_4/03.PDF.

Anexo 1. Encuestas a estudiantes

Encuesta 1

Estudiantes: Esta encuesta va destinada a ustedes con los siguientes objetivos:

1. Conocer sus preferencias en materia de juegos.
2. Medir su grado de aceptación por los juegos didácticos.

Sus respuestas serán tomadas en cuenta para la confección de una herramienta educativa que permita consolidar sus conocimientos y encontrar nuevas formas de evaluación a partir del propio juego interactivo.

Esperamos su sincera colaboración.

1. ¿En qué enseñanza recibiste clases donde se utilizaron juegos didácticos?

Primaria Secundaria Preuniversitario o Tecnológico UCI

2. En la universidad se han utilizado juegos didácticos en:

Matemática Física Inglés

Filosofía Programación

Otras Ninguna asignatura

3. En Programación se emplean Juegos Didácticos:

a menudo

a veces

casi nunca

nunca

4. Consideras que si se utilizaran juegos didácticos en las clases las mismas serían:

amenas

interesantes

aburridas

se podrían realizar más actividades.

los juegos no son para nuestra edad.

Una pérdida de tiempo.

5. ¿Te gustaría que se emplearan actividades donde aprendas a la vez que te entretienes?

Si No No sé

6. ¿Te gustaría poderte evaluar sistemáticamente sin presión?

Si No No sé

7. ¿Quisieras jugar los juegos de tu preferencia y a la misma vez consolidar tus conocimientos?

Si No No sé

8. ¿Cuáles son tus tipos de juegos preferidos?

De carreras de autos

Competitivos, de deportes

Dominó

Ajedrez

De estrategia

De rompecabezas

De Mesa (Tableros y dados)

De simulación

Otros

9. De los juegos por computadora los que más te llaman la atención son: (+)

Need for speed (o de este estilo)

Detective (o de este estilo)

Monopolio

Laberinto del saber

Parchís

War Craft (o de este estilo)

Zuma (o de este estilo)

Monctezuma (o de este estilo)

MVP (o de ese estilo)

Tetris

Otros

Encuesta 2

Estudiantes: Esta encuesta va destinada a ustedes con el objetivo de conocer su grado de aceptación por los juegos didácticos, una vez aplicados en la asignatura Programación 2.

1. Con la utilización de juegos didácticos en la asignatura Programación 2 las clases te resultaron:

amenas

motivadoras

interesantes

aburridas

se pueden realizar más actividades

los juegos no son para nuestra edad

una pérdida de tiempo

Anexo 2. Encuesta a profesores

Profesor: Con esta entrevista pretendemos conocer cuáles son las diferentes actitudes del profesor respecto al uso de los juegos didácticos en sus clases.

edad sexo años de experiencia en la docencia

años de experiencia en la Educación Superior

años de experiencia en la UCI

Especialidad _____

1. ¿Motiva a los alumnos en sus clases?

Si No A veces

2. ¿Cree que se cumplan los objetivos de una clase si no se motiva adecuadamente?

Si No No se

3. ¿Ha utilizado los juegos didácticos en alguna ocasión?

Si No

*Utilizo los juegos didácticos: *No utilizo los juegos didácticos:

porque es obligatorio porque complica la práctica docente

porque me siento más cómodo porque hay que invertir mucho tiempo en su preparación

porque me racionaliza el tiempo porque no contamos con los medios para su preparación

por ser más atractivo porque no me benefician económicamente

porque los alumnos mantienen la atención porque tengo mucha carga docente

___porque mejoran las relaciones con mis alumnos ___porque me resulta difícil adaptarlo a las clases
otras_____ otras_____

4. ¿En qué tipo de clase cree que se deben utilizar los juegos didácticos?

___ De ejercitación ___ en conferencias ___ en clases prácticas ___ en laboratorios
___ en ninguna ___ en todas

5. ¿Le gustaría poder contar con una herramienta interactiva que le permita crear fácilmente juegos didácticos adaptados a sus necesidades docentes?

___ Si ___ No ___ No sé

6. ¿Le gustaría poder contar con una herramienta educativa que permita nuevas formas de estudio y evaluación?

___ Si ___ No ___ No sé

7. ¿Le gustaría contar con una herramienta educativa que motive al estudiantado por el estudio hacia su asignatura?

___ Si ___ No ___ No sé

8. ¿Cuáles tipos de juegos considera que se podrían aplicar en el entorno docente?

___De carreras de autos

___Competitivos, de deportes

___Dominó

___Ajedrez

___De estrategia

___De rompecabezas

___De Mesa (Tableros y dados)

___De simulación

___Otros