

Universidad de las Ciencias Informáticas

Facultad 3

Grupo de Investigación AIRI



**HERRAMIENTA WEB PARA LA GESTIÓN DEL PROYECTO
EXTENSIONISTA QQSÍ**

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

Autores:

Oswaldo Cintra García

Michel Sed Araóz

Tutor(es):

Ing. Michel Álvarez Cancio

Ing. Yaicel Torres Garcés

Ing. Yoannys Gustavo Dueñas Pérez

Ciudad de La Habana, junio de 2016

“Año 58 de la Revolución”

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los _ días del mes de _____ del año 2016.

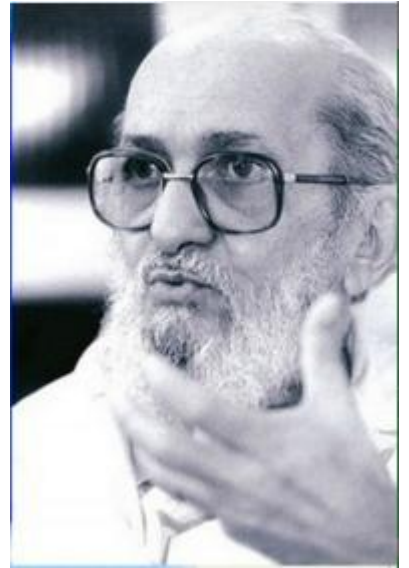
Oswaldo Cintra García

Michel Sed Araóz

Ing. Michel Álvarez Cancio

Ing. Yoannys G. Dueñas Pérez

Ing. Yaicel Torres Garcés



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

Paulo Freire

Datos de contacto

Síntesis de los tutores:

Tutor: Ing. Michel Álvarez Cancio.

Ingeniero en Ciencias Informáticas graduado con honores en el 2014, mención CITMA de “Estudiante Investigador”. Profesor del departamento de Técnicas de Programación de la Facultad 3.

Dirección: Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km. 2 ½. Torrens, municipio de La Lisa. La Habana, Cuba

E-mail: mcancio@uci.cu

Tutor: Ing. Yaicel Torres Garcés.

Ingeniero en Ciencias Informáticas en el 2015, desarrollador en el proyecto Aduana del centro CEIGE, de la Facultad 3.

Dirección: Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km. 2 ½. Torrens, municipio de La Lisa. La Habana, Cuba

E-mail: ygarces@uci.cu

Tutor: Ing. Yoannys Gustavo Dueñas Pérez

Ingeniero en Ciencias Informáticas en el 2014, desarrollador en el proyecto Tribunales del centro CEGEL, de la Facultad 3.

Dirección: Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km. 2 ½. Torrens, municipio de La Lisa. La Habana, Cuba

E-mail: yoannys@uci.cu

Agradecimientos

Agradecimientos

Osvaldo:

A mi mamá Caridad L. García que siempre me ha dado su apoyo, su pasión, su amor puro y sincero y ha confiado en mí, ha sido mi compañera de batallas interminables, ha sido mi madre, mi padre, mi amiga, mi profesora, en fin, gracias por traerme a la vida, por inculcarme valores, gracias por existir y por ser la excelente persona que eres. Gracias a mi Tatik por su enorme paciencia, comprensión e inteligencia para ayudarme a asimilar los contenidos y hacerme entender que sí podía, solo tenía que proponérmelo y esforzarme un poco más. A mi compañero de tesis y hermano Michel por ayudarme a cumplir esta meta que un día tracé y hoy cumplo. A mi familia por darme apoyo y ayudarme a salir de los momentos amargos por los que nos toca pasar en el transitar de la vida. A los profesores que durante este caminar me han brindado sus conocimientos, sus valores. A mi tutor Michel por su paciencia, su preocupación, sus sugerencias y apoyo incondicional. A mi tutor, mi amigo y hermano Yaicel por su apoyo desinteresado y las largas noches de código y debugueo cuando aparecía un error nuevo, las cuales lograron que tuviera presente su voz diciéndome: revisa bien qué está pasando, revisa que envías y que viene, mira la consola, esa no, la del servidor, en fin, siempre hizo un espacio para mis dudas y errores. A los miembros del tribunal por sus aclaraciones y su apoyo para que el documento contara con la calidad requerida. A Sosamania por su apoyo con las herramientas de desarrollo y por tratar de convencerme que Firefox es mejor que Chrome. A los titos por su apoyo incondicional y su amistad sincera. A la gran familia que he hecho en estos años conformada por Roly, Bauta, Chicho, Yasniel, Yadiesky, Omar, Carlos Javier, Leandro, Oslandy, Maiño, el Ruso y los que lamentablemente después de graduarse no he podido ver más. A todos ellos muchas gracias.

Agradecimientos

Michel:

A mi compañero de tesis, amigo y hermano Osvaldo. Gracias por todo el apoyo y ayuda incondicional. Las palabras son pocas para agradecerte.

A todos los miembros del tribunal por su gran ayuda y por contribuir a que el trabajo se realizara con la calidad requerida. Gracias por sus consejos.

A Michel mi tutor, gracias por la idea y contar con nosotros para este proyecto, por el apoyo y el tiempo que nos dedicaste. Yo estuve en tu posición alguna vez y se lo que se siente, si en algo te pudiera ayudar no dudes en contar conmigo, a parte de mi tutor te considero un amigo nunca lo olvides.

A Yaicel, hermano gracias, por aguantar al Osva y ayudarnos como lo hiciste.

A Alice por la gran ayuda y cuidar de Osva como tú sabes hacerlo.

A mi amiga Selma por su paciencia y toda la ayuda que me brindó.

A las amistades que han estado a mi lado en estos años y las otras que han tomado caminos separados, Yasniel, Bauta, Rolo, Oslíandy, Maico, Leo, Andy.

A todos los profesores que me entendieron y apoyaron durante mi estancia en la Universidad, los tendré que nombrar en los anexos porque no alcanzaría en esta sección. Entre ellos Maigret, Yanet, Leandro, Yaniesis, Dariela y Elizabeth. Gracias.

A Anet por estar gran parte de mi vida a mi lado y apoyarme en los momentos que más lo necesite. Gracias y ojalá pueda seguir contando contigo. Ahora te regalo todo mi tiempo aprovéchalo.

A todos mis hermanos Meme, Tico, Edua. Gracias por su preocupación y apoyo durante toda mi vida y mi carrera, los amo incondicionalmente. Espero siempre contar con su ayuda, así como espero que cuenten con la mía, en las malas y en las peores también.

Agradecimientos

A mi hermano Marlon, porque de manera inconsciente has sido un impulso enorme, ya que si hoy estoy graduándome es por cumplir una de dos grandes promesas que le hice a mami. La primera era graduarme para que tengas un ejemplo a seguir y la segunda la terminaré cuando este junto a ella, que es cuidarte mi hermano te amo.

Dedicatoria

Dedicatoria

Osvaldo:

Dedico el presente trabajo de diploma a mi madre por todo su amor, sacrificio y confianza a lo largo de esta etapa.

A la familia que he hecho en estos años en la universidad.

A la Universidad de las Ciencias Informáticas.

Michel:

A mi pequeño talismán, por cuidar de mi como lo hiciste y lo sigues haciendo. Sé que este momento lo estás disfrutando muchísimo, aquí estoy cumpliendo mi promesa sabes que nunca te falló, te amo mami.

A el mejor de mis amigos, el mayor de mis hermanos, mi padre. Gracias por todo lo que soy, por el amor sin límites, por el apoyo y la preocupación incesante. Te dejé para último pues después de ti no tengo más nada. Te amo, mientras que la tierra gire.

Resumen

Resumen

La extensión universitaria constituye el cuarto principio fundamental de la formación de profesionales en Cuba, siendo una de las principales tareas elevar la calidad de los futuros graduados de la educación superior cubana (Álvarez Cancio, y otros, 2015). En el departamento docente de Técnicas de Programación de la Facultad 3, de la Universidad de las Ciencias Informáticas (UCI), se encuentra el proyecto “Quién Quiere Ser Ingeniero”, el cuál es un proyecto extensionista que integra los contenidos de las asignaturas de la carrera del actual programa de estudio del Ingeniero en Ciencias informáticas, elementos de cultura general y algunas de las líneas de investigación existentes en la UCI. Este proyecto cuenta con una herramienta que no permite la gestión de sus competencias, las preguntas y perfiles de los participantes. Las competencias es la forma de interacción de los participantes con la herramienta, donde hasta la actualidad, solo pueden realizarse de manera presencial con la intervención de un mediador, imposibilitando la participación de otros competidores en diferentes espacios físicos. El presente trabajo propone una herramienta para la informatización del proyecto, de manera que se gestionen las competencias, las preguntas y pueda visualizarse un *ranking* de los usuarios con las mejores puntuaciones.

Palabras clave: actividad lúdica, extensión universitaria, juego online.

Abstract

Abstract

The university extension is the fourth fundamental principle of professional training in Cuba, one of the main tasks of improving the quality of future graduates of Cuban higher education (Álvarez Cancio, y otros, 2015). In the educational department Programming Techniques Faculty 3 University of Informatics Sciences (UIS), is the project "Who Wants To Be Engineer", which is an extension project that integrates the contents of the subjects of the current career study program in computer Science Engineering, elements of general culture and some of the existing lines of research in the UIS. This project is a tool that allows management of its powers, questions and profiles of the participants. Competences is the form of interaction of participants with the tool, where even today, can only be done in person with the intervention of a mediator, preventing the participation of other competitors in different physical spaces. This paper proposes a tool for the computerization project, so the skills are managed, questions and can display a ranking of users with the best scores.

Keywords: recreational activity, university extension, online game.

Índice de contenidos

Índice

Introducción	1
Capítulo 1: Fundamentación teórica.....	5
1.1 Introducción	5
1.2 Conceptos asociados al dominio del problema	6
Entorno educativo	6
Herramientas con fines didácticos.....	7
Juego didáctico	6
Actividad lúdica	7
Juegos online multiusuario	8
Sistemas de juego en línea en entornos educativos.....	8
1.3 Proyecto extensionista QQSI	10
1.4 Sistemas analizados	11
1.5 Herramientas y tecnologías utilizadas en el desarrollo	13
Arquitectura cliente-servidor.....	13
Tecnologías del lado del servidor	14
Tecnologías del lado del cliente	18
MEAN.IO.....	20
Herramientas IDE.....	21
Sistema gestor de base de datos	22
1.6 Proceso de desarrollo de software.....	26
Metodología de desarrollo de software.....	26
1.7 Conclusiones parciales	31
Capítulo 2: Descripción del sistema	32
2.1 Herramienta existente para la gestión del proyecto extensionista QQSI	32
2.2 Sistema web para el proyecto QQSI	33
Juego online.....	33
Sistema de ranking y puntuación	34
Actores del sistema	34
2.3 Requisitos funcionales y no funcionales	35
Requisitos funcionales	35
Requisitos no funcionales	37
2.4 Fase de planificación	39
Historias de usuarios.....	39
Estimación de esfuerzos por Historias de Usuario	40

Índice de contenidos

2.5	Plan de iteraciones	41
2.6	Plan de entregas.....	42
2.7	Interfaz de usuario	42
2.8	Tarjetas CRC.....	47
2.9	Conclusiones parciales.....	48
Capítulo 3: Validación y prueba		49
3.1	Pruebas	49
■	Pruebas unitarias	50
■	Pruebas de aceptación.....	52
3.2	Resultados obtenidos	56
3.3	Conclusiones parciales.....	58
Conclusiones generales		59
Recomendaciones.....		60
Bibliografía		61
Anexos		66
Anexo: Historias de Usuarios.....		66
Anexo 2.....		66
Anexo 3.....		67
Anexo 4.....		67
Anexo 5.....		68
Anexo 6.....		68
Anexo 7.....		69
Anexo 8.....		69
Anexo 9.....		70
Anexo 10.....		70
Anexo: Tarjetas CRC		71
Anexo 11		71
Anexo 12.....		71
Anexo 13.....		72
Anexo 14.....		72
Anexo 15.....		72
Anexo: Casos de prueba		73
Anexo 16.....		73
Anexo 17.....		73
Anexo 18.....		74
Anexo 19.....		74

Índice de contenidos

- Anexo 2075
- Anexo 2175
- Anexo 2276
- Anexo 2376
- Anexo 2477
- Anexo 2578
- Anexo 2678
- Anexo 2779
- Anexo: Acta de liberación80
 - Anexo 2880
- Anexo: Actas de aceptación.....81
 - Anexo 2981
 - Anexo 3082
- Glosario83

Introducción

Introducción

Las universidades cubanas inmersas en el contexto social contemporáneo abogan por mantener su pertinencia y la calidad de sus egresados, con el propósito de satisfacer la necesidad de formar hombres integrales dispuestos a contribuir con el desarrollo sostenible del país (Ruiz Echevarría, 2013). En este proceso debe conservarse la visión de que la educación superior contemporánea no se circunscribe solamente a la formación de profesionales mediante la docencia, ya que es necesario mantener la relación dialéctica con la función desarrolladora de la investigación y como parte integrante de dicha misión universitaria se involucra la promoción de la ciencia y la cultura hacia la sociedad, estas se definen como extensión universitaria (Sánchez Cuervo, 2009). En este contexto, el desarrollo y reconocimiento de las competencias profesionales están encaminados a fomentar una formación significativa e integral de los educandos.

El estudiante necesita aprender a resolver problemas, a analizar críticamente la realidad y transformarla, a identificar conceptos, aprender a aprender, aprender a hacer, aprender a ser y descubrir el conocimiento de una manera amena, interesante y motivadora (Martínez González, 2008). Los estudios realizados en las universidades cubanas adscritas al Ministerio de Educación Superior sobre los resultados docentes alcanzados por los estudiantes (modalidad presencial) durante los últimos cursos académicos, han arrojado una tendencia a la disminución en las diferentes carreras (Vargas Jimenez, y otros, 2011).

Los estudiantes opinan que no están motivados por el estudio y citan diversas causas, entre las que se encuentra el bajo desarrollo de intereses profesionales, debido fundamentalmente al deficiente trabajo de orientación profesional que recibieron en los niveles de educación precedentes. Esta realidad afecta la formación integral de los estudiantes al no desarrollar el interés por lograr una buena formación profesional, que incentive la necesidad del estudio (Ruiz Echevarría, 2013).

La utilización de actividades lúdicas en el entorno educativo actual y la falta de motivación de los estudiantes por la carrera propició la creación, en la Facultad 3 de la UCI, del proyecto “Quién Quiere Ser Ingeniero” (QQSI). El proyecto extensionista está conformado por un conjunto de competencias que abordan los contenidos de las asignaturas del plan de estudio del Ingeniero de las Ciencias Informáticas, así como elementos de la cultura general integral. El mismo tiene como objetivo incentivar la motivación de los estudiantes por los

Introducción

conceptos y prácticas de la carrera que estudian, utilizando métodos y técnicas para conseguir la consolidación de los conocimientos por los educandos y estimular el autoestudio (Álvarez Cancio, y otros, 2015).

Como resultado de la investigación se identifica el siguiente **problema a resolver**: ¿Cómo informatizar la competencia QQSI de manera que contribuya a la gestión de las preguntas y la visualización de las trazas?

Como **objetivo general** de la investigación se plantea: desarrollar una herramienta web que permita la realización de la actividad lúdica “QQSI” en modalidad online multiusuario¹ y que almacene sus trazas. Del objetivo general de la investigación se derivaron los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación.
2. Diseñar la propuesta de solución.
3. Implementar los elementos del diseño obtenidos.
4. Validar la solución propuesta.

Para cumplir con el objetivo planteado se definen las siguientes tareas de la investigación:

1. Análisis de los referentes teóricos asociados al tema de juegos didácticos.
2. Caracterización de la metodología de desarrollo de software, los lenguajes de programación y las herramientas vinculadas al desarrollo web.
3. Realización del análisis y diseño de la herramienta.
4. Implementación y desarrollo del sistema.
5. Definición de los casos de prueba para la validación
6. Análisis de la validación de las pruebas de software.

Del problema a resolver expuesto y el objetivo general planteado, se decide centrar la investigación en el **objeto de estudio**: proceso de desarrollo de herramientas informáticas

¹ Online multiusuario: Se le denomina así a la modalidad de competencia cuando se juega con más de un usuario en línea (Álvarez Cancio, y otros, 2015).

Introducción

con fines didácticos y como **campo de acción**: sistemas de juego en línea en entornos educativos.

Como **idea a defender** se tiene que:

Con el desarrollo de un sistema web en forma de actividad, se dotará al proyecto extensionista de una herramienta que le permita realizar sus competencias, contribuyendo a la gestión de las preguntas y visualizando un ranking de los usuarios mejor puntuados.

Para el correcto desarrollo del presente trabajo de diploma se emplearon **Métodos científicos** que sustentan la investigación a través de métodos teóricos y empíricos. A continuación, se explican los seleccionados:

Métodos teóricos:

Método Histórico – Lógico: permitió realizar un estudio del origen y evolución de las principales soluciones relacionadas con la investigación a nivel mundial.

Método Analítico-Sintético: Se aplica en la presente investigación durante el análisis de aplicaciones multiusuarios y para el estudio de fuentes bibliográficas seguras, conceptos y técnicas con soluciones previamente desarrolladas. Este método, además, permite descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución propuesta.

Métodos empíricos:

Entrevista: la entrevista cerrada fue utilizada para obtener información acerca del proceso a investigar. A través de la entrevista se obtiene la información necesaria sobre cómo se realizan las actividades que permiten arribar a la situación problemática planteada.

El presente trabajo de diploma quedará estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica.

En este capítulo se abordan de forma general los aspectos teóricos que sustentan esta investigación. Se realiza un análisis de las metodologías y herramientas de desarrollo que se pueden utilizar y se justifica la elección de cada una de ellas.

Capítulo 2: Descripción del sistema.

En este capítulo se describe de forma general la propuesta de solución. Se especifican los requisitos funcionales y no funcionales. Se exponen, además, los artefactos que se generan a partir de la metodología de desarrollo utilizada.

Capítulo 3: Validación y prueba.

Introducción

En este capítulo se muestra el conjunto de pruebas realizadas a la herramienta y los resultados que arrojaron. Además, se valida que el diseño realizado cumpla con la calidad requerida, y que el sistema implementado satisface las necesidades del cliente.

Capítulo 1: Fundamentación teórica

Capítulo 1: Fundamentación teórica

En el presente capítulo se expondrán conceptos asociados al dominio del problema. Se analizarán las tecnologías utilizadas para desarrollar sistemas de juego online, además del estudio de las herramientas a utilizar y se explicarán detalladamente los aspectos a tener en cuenta en la selección de las herramientas, tecnologías y metodología a utilizar para el desarrollo de la propuesta de solución.

1.1 Introducción

Según (Horruitiner Silva, 2007) la educación superior cubana, desde el surgimiento mismo del Ministerio de Educación Superior en el año 1976, ha venido trabajando por lograr integrar, en una concepción pedagógica coherente, el legado de nuestros más ilustres educadores. Con las actuales exigencias científico-técnicas de la formación de profesionales, la universidad cubana imprime a dicho proceso una personalidad propia que, sin desconocer las principales tendencias actuales vigentes en otros países, hoy ya comienza a tener reconocimiento internacional. Se permite hablar de una Escuela Cubana de Educación Superior, que se connota y distingue por la manera en que aborda algunas de las cuestiones que hoy son medulares para una institución de educación superior que aspire a lograr una educación de excelencia.

Las tecnologías de la información y las comunicaciones (TIC's) se han constituido en elementos sustantivos inherentes al desarrollo de todas las esferas de la vida. La educación no ha escapado a ello, donde se descubre un universo ilimitado de posibilidades como recurso para el aprendizaje con la posibilidad de expandirse a un mayor número de usuarios de diferentes escenarios con la capacidad de socializar el conocimiento. Nuestro país ha sido consecuente con el imperativo del desarrollo e implementación de las TIC's, las que ya se expanden a todos los niveles de enseñanza y cobran especial fuerza en la universalización de la educación superior, cuyos impactos fundamentales se centran en su uso como fuente de información, canal de comunicación, instrumento cognitivo y de procesamiento de la información. Esto impone nuevos roles para los protagonistas del proceso formativo e implican retos para el profesional del futuro y las instituciones formadoras, las que han de lidiar con aspectos técnicos, formación especializada, seguridad informática y otros elementos que determinan la expansión de las TIC's (Impacto de las

Capítulo 1: Fundamentación teórica

tecnologías de la información y las comunicaciones en la educación y nuevos paradigmas del enfoque educativo, 2011).

1.2 Conceptos asociados al dominio del problema

A continuación, se definirán los conceptos que se utilizarán en el presente trabajo para un mejor entendimiento. Estos conceptos están relacionados en su mayoría con el desarrollo de herramientas didácticas en entornos educativos.

Entorno educativo

La definición de entorno educativo está asociada a los conceptos de entorno: Conjunto de características que definen el lugar y la forma de ejecución de una aplicación; la definición de educativo: Perteneciente o relativo a la educación, que educa o sirve para educar. De ahí que una definición de entorno educativo sea: el lugar y la forma de ejecución de una actividad que tiene como propósito educar.

Un entorno educativo es un sistema de aprendizaje que hace más eficiente la formación de los estudiantes; que facilita aprender de manera constructiva, instructiva y en colectivo, así como el empleo de múltiples estilos para las actividades que el alumno debe realizar como parte de su proceso de instrucción, en el que pueden integrarse varios tipos de software y materiales de consulta. (Pérez TA, 2001)

Juego didáctico

El juego puede ser considerado o bien como medio, o bien como fin. La importancia del juego para el crecimiento, para la calidad de vida de la gente, para las relaciones interpersonales, ha sido tratado en múltiples textos de psicología y antropología. La pedagogía se hace eco de esa importancia y lo desarrolla también en numerosos tratados. Todo esto, da cuenta del juego como fin. La pedagogía, también lo trata como medio. En este caso, el juego forma parte del método. Así el juego es una valiosa herramienta para el aprendizaje. Ambas dirán que el ser humano aprende, también, a través del juego, especialmente en algunas de las etapas de asimilación del conocimiento. (Zuberman, 2000)

El juego didáctico es una técnica participativa de la enseñanza encaminado a desarrollar en los estudiantes métodos de dirección y conducta correcta, favorecer así la disciplina con un adecuado nivel de decisión y autodeterminación; es decir, no sólo propicia la adquisición de conocimientos y el desarrollo de habilidades, sino que además contribuye al logro de la motivación por las asignaturas; o sea, constituye una forma de trabajo docente que brinda

Capítulo 1: Fundamentación teórica

una gran variedad de procedimientos para el entrenamiento de los estudiantes en la toma de decisiones para la solución de diversas problemáticas.

El juego es una actividad, naturalmente feliz, que desarrolla integralmente la personalidad del hombre y en particular su capacidad creadora. Como actividad pedagógica tiene un marcado carácter didáctico y cumple con los elementos intelectuales, prácticos, comunicativos y valorativos de manera lúdica (López Gabriel, 2014).

Herramientas informáticas con fines didácticos

Un software educativo es un programa didáctico que permite hacer llegar el conocimiento a una audiencia de manera más sencilla, además adapta el ritmo de trabajo y las necesidades de cada educando para lograr diversas alternativas de aprendizaje (Ramos Pérez, y otros, 2008). Interactúa con distintas materias de diferentes maneras y ofrecen un entorno de trabajo sensible en dependencia de las circunstancias de los alumnos (Núñez Rojas, 2010). Se caracterizan por ser interactivos, a partir de que se le añaden programas de entretenimientos y recursos multimedia: videos, sonidos, fotografías, ejercicios, diaporamas y juegos instructivos, que apoyan la evaluación y el diagnóstico del estudiante (Coordinación General de Investigación y Desarrollo de Modelos Educativos, 2012). El software educativo cuenta con características o aspectos que son comunes y que lo distinguen independientemente de la finalidad educativa (Pérez Marqués, 2012):

- Son materiales desarrollados con finalidad didáctica.
- Utilizan la computadora como medio para realizar sus actividades.
- Son interactivos, ya que dan respuestas inmediatas a las acciones de los estudiantes.
- Permiten el intercambio de información entre la computadora y el estudiante.
- Son fáciles de usar, no se requiere conocimientos avanzados en informática para su uso.

Actividad lúdica

Los estudios sobre el juego acentúan que la actividad lúdica es una vía privilegiada de acceso al conocimiento y de potencialidad interactiva, una fuente de vivencia y aprendizajes dentro de la etapa educativa a la que nos estamos refiriendo. Constituye una actividad de importancia capital para la adquisición de ciertos aspectos básicos para la vida de las personas, tales como las conductas de apego y vinculación afectiva, las habilidades comunicativas y la capacidad de gestionar la libertad y la toma de decisiones, así como el

Capítulo 1: Fundamentación teórica

desarrollo del pensamiento creativo. En definitiva, la actividad lúdica permite a la persona poner en práctica las competencias necesarias para prepararse para la vida en sociedad (El juego: la experiencia de aprender jugando, 2003).

Juegos online multiusuario

Se consideran juego online multiusuario a las modalidades de juego en internet o mediante una intranet, que permite la interacción de varios usuarios de manera simultánea en modo competitivo. El juego online multiusuario, dentro del proyecto extensionista, se le denomina a la competencia que posee más de un usuario en juego. Mediante esta actividad el usuario competirá con otros usuarios y ganará el que más puntos acumule al finalizar la competencia (Álvarez Cancio, y otros, 2015).

Sistemas de juego en línea en entornos educativos

El concepto de juego educativo involucra un producto digital que incorpora una aventura en un entorno audiovisual, donde el usuario debe resolver ciertos problemas que conforman la esencia del juego en cuestión en pos de su desarrollo. Se crea entonces una reciprocidad entre el ámbito educativo y el recreacional que fortalece aún más la accesibilidad constante del usuario al conocimiento y a su formación integral.

Los juegos educativos representan una herramienta pedagógica de muy alto valor ya que con ellos el usuario puede aplicar los conocimientos adquiridos en el aula o en el juego como instrumento para llegar hasta la victoria. Esto aumenta en gran medida el rendimiento del estudiante pues este se verá mucho más interesado en captar la mayor cantidad de conocimientos que le sea posible. Los juegos educativos estimulan el desarrollo de habilidades en los estudiantes como la coordinación motriz y los reflejos, la capacidad observación, de concentración, la creatividad, la imaginación y la memoria (Infante Carbó, 2013).

Las características que debe cumplir un juego son las siguientes (Infante Carbó, 2013):

- Ser una actividad libre, comenzada y terminada a voluntad del usuario.
- Ser improductiva en generación de propiedad o riqueza, cuyo fin último es el juego en sí mismo.
- Está acotada, con límites de tiempo y de espacio.
- Está reglamentada, dentro de un orden específico propio de sí misma.
- Produce incertidumbre en el desarrollo y la finalización, dado que cada partida es distinta y depende de diversas ideas.

Capítulo 1: Fundamentación teórica

A las características generales sobre juegos, para que sea educativo, se debe añadir:

- Deben partir de una premisa a resolver.
- Debe tener siempre al menos una solución cierta.
- Que el usuario/jugador aprenda algo ya sea por; introducción de nuevos conocimientos; fijación de conocimientos previamente adquiridos; ejercicio de habilidades; compartición de experiencias; descubrimiento de conceptos o desarrollo de creaciones (Infante Carbó, 2013).

En la actualidad los juegos se utilizan para los niños como una herramienta muy beneficiosa en la educación, permitiendo de una forma diferente y divertida su desarrollo intelectual y cognitivo. Los juegos educativos, han traído consigo múltiples ventajas, pues permiten:

- Potenciar el desarrollo de la actividad mental y física.
- Fomentar normas de relaciones sociales y morales.
- Descubrir valores y aptitudes.
- Entender diversos conceptos mediante representaciones simbólicas.
- Fomentar la interacción entre los jugadores, que expresan y confrontan sus puntos de vista y opiniones.
- Utilizar y aplicar conocimientos y habilidades en diversos contextos y situaciones.
- Ganar en confianza y vencer dificultades.

Sin embargo, los juegos educativos pueden considerarse una actividad de enseñanza-aprendizaje efectiva si (Jiménez Rodríguez, 2006):

- Requiere la participación activa de los jugadores.
- Exige el uso de diversas habilidades, destrezas y conocimientos.
- Su objetivo radica en que los participantes se diviertan y aprendan simultáneamente.
- Estimula el razonamiento
- Fomenta la comunicación
- Es susceptible de progresar en él, de jugar cada vez mejor.

La aplicación provechosa de los juegos educativos en la enseñanza no solo posibilita el desarrollo biológico, psicológico, social y espiritual del jugador como eje de la acción educativa; también promueve la diversión y el entretenimiento mientras adquieren cultura y conocimiento.

Capítulo 1: Fundamentación teórica

La aplicación de los conceptos definidos anteriormente permitió enfocar el objetivo de la herramienta en promover el aprendizaje. Además, incita a la actividad y al pensamiento, haciendo que los estudiantes dediquen más tiempo a trabajar y, por tanto, la utilización del juego como actividad educativa que motive al educando a estudiar. Cuando se interactúa con una herramienta informática con fines didácticos, se logra un alto grado de implicación en el trabajo debido a la interacción con el ordenador. En la herramienta los estudiantes se ven obligados a tomar continuamente nuevas decisiones ante las respuestas del ordenador a sus acciones.

1.3 Proyecto extensionista QQSI

QQSI surge con el propósito de contar con una herramienta informática que sirviera de soporte o de medidor de conocimientos, en un inicio a los estudiantes de la Facultad 3 de la Universidad de las Ciencias Informáticas y posteriormente a la comunidad universitaria, en aras de aumentar la motivación de los estudiantes en las asignaturas de la carrera. QQSI es una actividad que integra el actual plan de estudio del Ingeniero en Ciencias Informáticas con elementos de cultura general y algunas líneas de investigación existentes en la UCI. El mismo puede ser adaptado a otros planes de estudio y a niveles inferiores como “Preuniversitario”, “Secundario” y “Primario”.

Las preguntas son almacenadas en una base de datos según el grado de dificultad de las mismas, donde se van eligiendo aleatoriamente las más sencillas primero. A medida que los competidores avancen de ronda la complejidad de las preguntas aumentan. El juego contará con 20 rondas como máximo y los concursantes permanecen en el juego siempre que su puntuación sea positiva. Cada participante puede usar 3 comodines y pueden utilizarlos, cada uno, una sola vez y en el orden que deseen.

Uno de los comodines será la opinión del público, este consiste en que el público alzaré la mano según la respuesta que crea correcta y le dará una dirección sobre la posible respuesta a elegir. El siguiente comodín será mitad de las respuestas, ocultando 2 de las 4 opciones de posibles respuestas que saldrán por cada pregunta. El último comodín será la posibilidad de llamar a un amigo, el cuál puede brindarle la respuesta que busca. La utilización del comodín del público y el de llamar a un amigo generan un ambiente de competición sano, propicio para que todos los presentes adquieran nuevos conocimientos o recuerden algunos que pudiesen haber olvidado.

Capítulo 1: Fundamentación teórica

El proyecto extensionista QQSI cuenta con una herramienta la cual posee las siguientes limitantes:

- La gestión se realiza de forma manual.
- No permite su acceso desde distintas ubicaciones.
- El estudiante no sabrá que puntuación obtuvo hasta que finalice la competencia.
- Las preguntas no se pueden gestionar.
- Es obligatoria la presencia de un miembro del proyecto extensionista para que la actividad se realice.

Debido a estas inconvenientes y a la poca información que se tiene del sistema existente se decide descartar esta herramienta y realizar una nueva herramienta para la gestión del proyecto extensionista.

1.4 Sistemas analizados

La elección entre la cantidad y variedad de herramientas disponibles en la actualidad para usos didácticos está determinada por las preferencias de los usuarios. Debido al auge de los dispositivos móviles durante los últimos años se ha propiciado un nuevo mercado para las editoriales: el de las aplicaciones educativas. Actualmente, se calcula que hay más de 80.000 aplicaciones educativas disponibles en las diferentes plataformas: Android, iOS, BlackberryOS y sistemas web. Este tipo de aplicaciones añaden un fuerte componente lúdico a algunos procedimientos educativos que hasta ahora solo se habían destinado al papel. (García, 2014).

Entre las aplicaciones estudiadas con similitud a los requerimientos del evento QQSI se encuentran: “Quién quiere ser millonario”, y “Verdadero o falso”. A partir de su análisis se pudo constatar que no constituyen una solución al problema identificado debido a que son aplicaciones privativas y no se puede acceder a su código fuente, por lo que no es posible adaptarlas al sistema de conocimientos deseado. Es por ello que se reafirma la necesidad de una nueva herramienta, sin embargo, su estudio permitió identificar características que constituirán requisitos funcionales del sistema a implementar. A continuación, se describen detalladamente como aportaron a la investigación cada una:

- Quién quiere ser millonario: Aplicación creada por Ludia Inc. que corre sobre sistemas Android superiores a la versión 2.3, basado en el formato de preguntas aleatorias visto en la exitosa serie de televisión del juego. En esta aplicación se puede competir contra amigos para contestar correctamente preguntas de opción

Capítulo 1: Fundamentación teórica

múltiple trivia de varias categorías, con la posibilidad de utilizar uno de los famosos comodines: doble caída, pedir a la audiencia o saltar la pregunta. Puede obtener puntos de bonificación por responder varias preguntas en una fila o respondiendo a preguntas tan rápido como pueda. Incluso se puede obtener una recompensa especial por completar las colecciones millonarias. Esta aplicación cuenta con las siguientes características (Ludia Inc, 2016):

- Capacidad de competir contra amigos o seguidores millonarios a través de la aplicación.
 - Miles de preguntas y cientos de categorías tomadas directamente del espectáculo de la televisión.
 - Efectos visuales y de sonido semejantes a los del espectáculo de la televisión.
 - Acceso a comodines como: doble caída, pedir a la audiencia y saltar la pregunta.
- Verdadero o falso: Esta aplicación fue creada por LOTUM media GmbH y corre sobre sistemas Android superiores a la versión 2.3.3. La versión más reciente de verdadero o falso es la 2.4, con fecha del 1 de octubre de 2015. Consiste en responder de forma rápida y decidir si el enunciado es correcto o no. Integra una diversidad de datos sobre temas que ponen a prueba sus conocimientos; pero cuidado, de entre todos los enunciados hay algunas afirmaciones falsas y algunas que son trampa. A medida que el usuario avance de nivel, simultáneamente lo hace la dificultad de las afirmaciones. El usuario dispone de un tiempo para responder las afirmaciones, el cual está en dependencia de la complejidad de estas. En cada nivel el usuario tiene un límite de respuestas a las que puede responder incorrectamente determinado por la cantidad de preguntas que posea ese nivel así como la posibilidad de pasar una respuesta que no sepa (LOTUM media GmbH, 2015).

De forma general los elementos que pueden aportar a la solución propuesta son:

- Sistema de comodines
- Categorizar las preguntas según su complejidad.
- Límite de tiempo para responder cada pregunta.

Capítulo 1: Fundamentación teórica

- Posibilidad de pasar una pregunta que no se sepa.

1.5 Herramientas y tecnologías utilizadas en el desarrollo

Existe gran variedad de tecnologías usadas en el desarrollo de aplicaciones web. A continuación, se abordan las que constituyen la base del desarrollo del sistema web QQSI.

Arquitectura cliente-servidor

Las aplicaciones web son soluciones informáticas que los usuarios utilizan accediendo a un servidor a través de Internet o su red interna (Intranet). Como interfaz con la aplicación se utiliza un navegador de Internet (SPL Sistemas de Información, 2012).

La arquitectura cliente/servidor utilizada por las aplicaciones web, es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo y los servidores de correo. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica sigue siendo la misma (Sun Microsystems, 2006).

Las aplicaciones web desde su surgimiento han evolucionado desde contenidos estáticos hasta mega sistemas que gestionan altos volúmenes de información generada por usuarios u otros sistemas informáticos, como son las redes sociales en Internet. En la actualidad se habla de aplicaciones web más dinámicas, sociales y que sean capaces de gestionar información en tiempo real.

Tiempo real en el área de la informática y las telecomunicaciones se define como el conjunto de hardware y software que están sujetos a estrictas restricciones de tiempo desde la

Capítulo 1: Fundamentación teórica

ocurrencia de un evento hasta la respuesta del sistema al mismo. Generalmente los tiempos de respuesta en tiempo real son entendidos en el orden de los milisegundos y algunas veces en microsegundos. Un sistema informático en tiempo real debe soportar las restricciones de tiempo asociadas a las respuestas del sistema, aunque el mismo se encuentre altamente sobrecargado (Ben-Ari, 1990).

Se decide utilizar esta arquitectura debido a que es un modelo de aplicación que distribuye las tareas entre los proveedores de recursos y los clientes, es decir la capacidad de proceso está repartida entre ambos. Son ventajosas en un sistema operativo multiusuario distribuido a través de una red de computadoras. Son útiles en sistemas web ya que toda la interacción, de un cliente o usuario, con la aplicación se podrá llevar a cabo mediante un navegador web.

Tecnologías del lado del servidor

Estas tecnologías proporcionan un entorno rápido de creación de scripts y soporte para los estándares más importantes. Las aplicaciones dinámicas de Internet permiten la integración de las comunicaciones bidireccionales y los datos en tiempo real en las aplicaciones. Esto también es realizado por las aplicaciones tradicionales de base de datos. En este sentido, JavaScript se perfila como un ejemplo necesario del lado del servidor.

1.5.2.1 JavaScript

JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como Node.js o Apache CouchDB². Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

El JavaScript estándar es ECMAScript. A partir de 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos soportan como mínimo ECMAScript 3. Una sexta revisión del estándar está en proceso (Mozilla Developer Network y colaboradores individuales, 2016).

Entre las principales características del lenguaje se pueden encontrar:

² Gestor de base de datos no relacional de código abierto.

Capítulo 1: Fundamentación teórica

- Es simple, no hace falta tener conocimientos avanzados de programación para poder hacer un programa en JavaScript.
- Maneja objetos dentro de la página web y sobre ese objeto se pueden definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas.
- Es dinámico y responde a eventos en tiempo real.
- El auge de las tecnologías web y el nacimiento de la web 2.0 a inicios del siglo XXI provocaron el incremento del uso de JavaScript en sitios web cada vez más dinámicos. Esta situación motivó al mismo tiempo el surgimiento de numerosas bibliotecas de JS que facilitan considerablemente el trabajo con los objetos y eventos del DOM³.

Este lenguaje puede desarrollarse en el cliente, se aplica en el navegador del cliente (JavaScript en el cliente) y además en el servidor (JavaScript en el servidor).

Con JavaScript se pueden lograr efectos realmente mágicos en una página, así como formularios, imágenes y textos; tiene como ventaja que la mayoría de los navegadores no tienen problemas para interpretarlo.

En JavaScript se refiere siempre al objeto prototipo para crear las instancias, pero esto no hace al lenguaje ni más ni menos orientado a objetos que otros lenguajes. El concepto de clases en JavaScript está dado por el prototipo, con la propiedad *prototype* de los objetos prototipos podemos añadir todas las propiedades que se quieran, no a la instancia sino al mismo prototipo. Permite codificar de una manera flexible, definir un objeto prototipo y si se necesita añadir alguna propiedad lo único que se tiene que hacer es asignar el true. Se puede añadir además propiedades a los objetos prototipo predefinidos por el lenguaje (González Gutiérrez, Enrique , 2006).

JavaScript en el lado del servidor ha tomado fuerza desde el lanzamiento de Node.js en 2009, beneficiado por el alto rendimiento y rápida respuesta de Node.js. En el procesamiento de peticiones, muchos de los proyectos que se inician hoy en día escogen a JavaScript como lenguaje de programación del cliente y del servidor.

³ Siglas en inglés para Document Object Model. Es una interfaz de programación de aplicaciones (API) para documentos HTML.

Capítulo 1: Fundamentación teórica

Teniendo en cuenta esta tendencia, las características y el buen rendimiento de Node.js, JavaScript es elegido para ser empleado en el desarrollo de la solución.

1.5.2.2 Servidores Web

- **Apache.**

Lanzado en el año 1995, es un servidor web de código abierto, robusto, flexible, rápido y eficiente, continuamente actualizado, adaptado a los nuevos protocolos y cuya implementación se realiza de forma colaborativa. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo conocido como el Grupo Apache. Apache 2 es una profunda revisión del servidor Apache, centrándose en la escalabilidad, seguridad y rendimiento.

Apache es multiplataforma, posee un sistema de módulos dinámico que pueden ser habilitados y deshabilitados mediante la ejecución de un comando y después de haber reiniciado el proceso servidor. Sus módulos de multiprocesamiento (MPM⁴) en el manejo de peticiones permiten que cada solicitud sea atendida por un hilo o proceso separado y utiliza sockets sincrónicos (Kabir, 2002).

Según W3Techs.com⁵, es el servidor web más usado con un 58.4%, aunque estos números han caído debido al surgimiento y avance de novedosas soluciones como Nginx que emplean menos recursos y aportan mayor velocidad al procesar las peticiones del cliente.

- **Nginx.**

Fue creado por Ígor Sysóiev, el principal administrador de sistemas del buscador ruso Rambler. En 2002 empezó a crear su propio servidor y en 2004 publicó la primera versión de este producto. Se trata de un servidor web HTTP de código abierto, licenciado bajo la Licencia BSD simplificada. Nginx es multiplataforma, utiliza una arquitectura basada en eventos que hace que su consumo de recursos crezca de forma predecible, puede ser utilizado como un proxy inverso con opciones de caché y tiene soporte para SSL, FastCGI, streaming de video, autenticación, compresión gzip y balanceo de carga (Nedelcu, 2010).

⁴ Siglas en inglés para Multi-Processing Modules

⁵ Sitio web de W3Techs.com URL: http://w3techs.com/technologies/cross/web_server/ranking

Capítulo 1: Fundamentación teórica

Actualmente este servidor web, según W3Techs.com³ es el segundo más utilizado en el mundo por detrás de Apache con un 23.3% y va captando adeptos cada vez más. Empresas como la compañía de TV online Hulu utilizan Nginx por su estabilidad y configuración simple. Otros usuarios, como Facebook y WordPress.com, lo utilizan porque la arquitectura asíncrona del servidor web deja una pequeña huella de memoria y bajo consumo de recursos, haciéndolo ideal para el manejo de múltiples y cambiantes páginas web activas.

- **Node.js.**

Es una plataforma construida sobre el entorno de ejecución JavaScript V8 de Chrome para desarrollar fácilmente aplicaciones web rápidas y escalables. Node.js utiliza un manejador de eventos, modelo no bloqueador de entradas y salidas que permite ligereza y eficiencia, ideal para aplicaciones en tiempo real de datos intensivos que se ejecutan a través de dispositivos distribuidos (Joyent Inc, 2014).

Para el manejo de paquetes, Node.js utiliza NPM⁶, un gestor de paquetes que se encarga de solucionar las dependencias de software de terceros en la aplicación. Mediante simples comandos, los desarrolladores pueden descargar los módulos que necesitan, para esto, es necesario tener una conexión a Internet.

En comparaciones realizadas por el sitio web Changeblog⁷ entre Apache y Node.js en cuanto a velocidad, peticiones procesadas, consumo de memoria RAM⁸ y CPU⁹, dan como claro ganador a Node.js. En esta prueba se observa que Node.js es hasta 3 veces más rápido que Apache y solo un 2.5 % de las peticiones no son bien procesadas. El consumo de memoria y CPU es mayor, pero esto no debe ser una sorpresa considerando el alto rendimiento.

La tabla que a continuación se muestra (ver tabla 1), refleja los resultados de las pruebas realizadas por Changeblog.

⁶ Siglas en inglés para Node Package Manager

⁷ Sitio web de Chanblog: <http://zgadzaj.com/benchmarking-nodejs-basic-performance-tests-against-apache-php>

⁸ Siglas en inglés para Random Access Memory

⁹ Siglas en inglés para Central Processor Unit

Capítulo 1: Fundamentación teórica

Tabla 1 Comparación de rendimiento entre Apache y Node.JS (Changeblog)

	Tiempo total (s)	Peticiones fallidas	Peticiones por segundos	Tiempo por peticiones (ms)	Velocidad de transferencia (kb/s)
Apache	3570.753	2617614	280.5	3.571	10.7
Node.js	1043.076	25227	958.7	1.043	76.8

Otras pruebas de rendimiento también subrayan el rendimiento de Node.js sobre sus similares, en este caso Nginx, destacando la cantidad de transacciones por segundo que es capaz de procesar Node.js (el doble), y el mayor tiempo para una transacción tomado por Node.js, 1.10s contra 13.95s de Nginx.

Por todo lo mencionado anteriormente, se elige como servidor web a utilizar Node.js, principalmente por su velocidad al procesar peticiones y el alto rendimiento demostrado, necesario para sostener una aplicación pensada para ser empleada por más de 1000 personas y brindar una experiencia satisfactoria al usuario final. La versión a utilizar será la 0.12.

Tecnologías del lado del cliente

Las tecnologías del lado del cliente están orientadas preferentemente, como su nombre indica, para ejecutarse en los puestos cliente, por ejemplo, AngularJS, CSS, HTML, JavaScript y NodeJS. Esto proporciona las capacidades del cliente que hacen posible la creación de aplicaciones dinámicas de Internet al aprovechar el poder de procesamiento local de las computadoras y los dispositivos.

- **AngularJS**

AngularJS es un framework utilizado en el lado del cliente, cuando se está trabajado con Node.js, ya que Node permite la utilización de JavaScript en ambos lados cliente-servidor (AngularJS: Novice to Ninja, 2014).

AngularJS viene cargado con todas las herramientas que los creadores ofrecen para que los desarrolladores sean capaces de crear ese HTML enriquecido. La palabra clave que permite ese HTML declarativo en AngularJS es "directiva", que no es otra cosa que código

Capítulo 1: Fundamentación teórica

JavaScript que mejora el HTML. Puede usar el que viene con AngularJS y el que han hecho terceros desarrolladores, puesto que muchas personas están contribuyendo con pequeños proyectos “independientes del propio framework” para enriquecer el panorama de directivas disponibles. Hasta este punto será un "consumidor de directivas", y finalmente cuando vaya tomando experiencia será capaz de convertirse en un "productor de directivas", enriqueciendo las herramientas para mejorar un HTML (AngularJS: Novice to Ninja, 2014).

Las características ofrecidas por AngularJS en la creación de aplicaciones web lo convierten en una biblioteca JavaScript robusta que facilita el desarrollo de productos informáticos.

Se decide utilizar AngularJS en su versión 1.5 debido a que es un proyecto de código abierto, realizado en JavaScript que contiene un conjunto de bibliotecas útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un *framework* para el desarrollo escrito en el lenguaje JavaScript.

- **Hojas de estilo en cascada**

CSS, del inglés *Cascading Style Sheets*, es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación. Es utilizado para definir el aspecto de todos los contenidos, ya sea el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, así como la tabulación con la que se muestran los elementos de una lista.

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (Eguíluz Pérez, 2008).

Ventajas de CSS:

- CSS permite un mejor control en la presentación de un sitio web que los elementos de HTML, agilizando su actualización.
- La principal ventaja de CSS sobre el lenguaje HTML, es que el estilo se puede guardar completamente por separado del contenido, siendo posible, por ejemplo, almacenar todos los estilos de presentación para una web de 10.000 páginas en un solo archivo de CSS.

Capítulo 1: Fundamentación teórica

Se decide utilizar CSS, en el lado del cliente, ya que modifica la presentación de cada elemento sin modificar el código HTML, ahorrando esfuerzo y tiempo de edición. Así, el mantenimiento de la herramienta web se hace más sencillo. El lenguaje CSS ofrece una amplia gama de herramientas de composición más potentes que HTML y evita tener que recurrir a “trucos” para conseguir algunos efectos. Las Hojas de Estilo pueden usarse con otros lenguajes de programación (por ejemplo, JavaScript) para conseguir efectos dinámicos en las páginas y se pueden especificar para distintos navegadores.

- **Lenguaje de marcado de hipertexto**

HTML¹⁰, es el lenguaje de marcado predominante para la construcción de páginas web, utilizado normalmente en la www¹¹. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque sí le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. Aunque HTML es un estándar muy usado se considera un lenguaje estático basado en etiquetas muy limitado que no puede ser ampliado. Efectos como arrastrar y soltar; y cambios de tamaño de elementos son imposibles de realizar, al igual que la actualización de zonas específicas de la página sin la necesidad de tener que redireccionar una nueva página HTML (Musciano, y otros, 1999).

HTML se utilizó combinándose con CSS, para determinar la forma en la que aparecen las vistas en el navegador del cliente, así como también las imágenes y los demás elementos que componen la aplicación.

MEAN.IO

Es un completo marco de trabajo para JavaScript que simplifica y acelera el desarrollo de aplicaciones web. Sus siglas provienen de la unión de MongoDB, Express, Node.js y Angular, herramientas que este incluye. Está diseñado para dar una forma rápida y organizada de iniciar el desarrollo de aplicaciones web basadas en MEAN con módulos útiles como Mongoose y Passport, ambos ya configurados. Express según sus creadores, es un marco de trabajo para el desarrollo de aplicaciones web minimalista y flexible para Node.js (LINNOVATE, 2014).

¹⁰ HTML: HyperText Markup Language.

¹¹ www: World Wide Web.

Capítulo 1: Fundamentación teórica

Entre sus principales características se incluye la presencia de paquetes, los cuales se pueden instalar y configurar en dependencia de lo que el desarrollador necesita. Se decide utilizar este conjunto de herramientas, debido a que el lenguaje de programación elegido es JavaScript y este marco de trabajo logra integrar diferentes herramientas que se complementan, debido a que en todos los niveles usan el mismo lenguaje de programación y se obtienen mejores tiempos de respuestas.

■ Herramientas IDE¹²

Los IDE son un entorno de programación creado con un conjunto de herramientas para el programador: editor de código, compilador, depurador e intérprete. Estos pueden ser aplicaciones independientes o pueden formar parte de aplicaciones existentes, para un lenguaje de programación específico o multilenguaje. Proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación **Fuente especificada no válida**. Sus componentes son:

- Editor de código: aplicación para escribir código de programación en el lenguaje deseado. Contiene todas las herramientas de edición necesarias, y algunas veces características propias de IDE's.
- Compilador: programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar.
- Depurador: programa usado para probar y revisar posibles errores que contenga el código escrito.
- Intérprete: programa informático capaz de analizar y ejecutar otros programas, escritos en un lenguaje de alto nivel.
- **JetBrains WebStorm 2016.1.1**

Como entorno de desarrollo se decide utilizar WebStorm en su versión 2016.1.1. WebStorm es un IDE ligero pero potente, perfectamente equipado para el desarrollo del lado del cliente y el desarrollo del lado del servidor con Node.js. El editor de código inteligente de WebStorm proporciona soporte de primera clase para JavaScript, Node.js, HTML y CSS, así como sus sucesores modernos. Disfruta de completamiento de código, detección de errores y refactorización (Jetbrains, 2010). Se elige este IDE fundamentado en que hace posible agilizar el proceso de desarrollo en el lado del servidor con Node.js, ofrece opciones como completado inteligente de código, características avanzadas de navegación y búsqueda, así como asistencia para trabajar con Angular, React y Meteor.

¹² Siglas en inglés para Integrated Development Environment

Capítulo 1: Fundamentación teórica

- **Eclipse**

Es un entorno de desarrollo integrado (IDE), de código abierto y multiplataforma. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

Eclipse, al mismo tiempo, es una comunidad de código abierto, cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta compuesta por marcos extensibles, herramientas de tiempos de ejecución para la construcción, implementación y administración de software a través del ciclo de vida de los mismos. La Fundación Eclipse es una organización sin fines de lucro que aloja los proyectos de Eclipse y ayuda a cultivar tanto, una comunidad de código abierto y un ecosistema de productos y servicios complementarios (The Eclipse Foundation, 2015). Se

- **NetBeans**

Es un entorno de desarrollo integrado de código abierto para desarrolladores de software. Cuenta con todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, así como con C/ C + +, PHP, JavaScript y Groovy (Oracle Corporation, 2015). NetBeans recibe el soporte integrado para lenguajes dinámicos, todo en una herramienta de gran alcance. Provee de un resaltado de código sintáctico y semántico, documentación emergente, formateo de código y plegado, y el marcado de los sucesos y puntos de salida.

El IDE también ofrece un completo editor de HTML, JS y CSS, con la ventaja de proveer un resaltado de sintaxis completo, completamiento de código inteligente, documentación pop-up, y la comprobación de errores para HTML, CSS y JavaScript, incluyendo HTML 5, JavaScript 1.7. El editor reconoce código HTML en los archivos de JavaScript y viceversa.

Teniendo en cuenta las características anteriormente planteadas, se selecciona el IDE JetBrains Webstorm en su versión 2016.1.1 para desarrollar la solución propuesta, por todas las facilidades que brinda para el trabajo con proyectos JS, AngularJS y al soporte a tecnologías de última generación como HTML5.

Sistema gestor de base de datos

Un sistema gestor de bases de datos (SGBD) es una colección de programas que permiten a los usuarios crear y mantener una base de datos. También se puede considerar como un software de propósito general que facilita los procesos de definición, construcción y

Capítulo 1: Fundamentación teórica

manipulación de la base de datos para distintas aplicaciones (Sistemas gestores de bases de datos, 2010).

- Definición de la base de datos: especificar los tipos de datos, estructuras y restricciones.
- Construcción de la base de datos: almacenar los datos.
- Manipulación de la base de datos: consultar, actualizar el diseño y generar informes.

Por lo tanto, se concluye que, si la base de datos son los datos almacenados, el SGBD es el programa o conjunto de programas que gestionan y mantienen consistentes estos datos.

- **MySQL**

Es un SGBD multiplataforma de código abierto desarrollado, distribuido y soportado por Oracle Corporation. Fue creado en 1995 por David Axmark, Allan Larsson y Michael Widenius.

MySQL se caracteriza fundamentalmente por su velocidad, rendimiento y fiabilidad, motivo por el cual es ampliamente utilizado en sitios web y aplicaciones de escritorio de baja y mediana complejidad. Muchas de las organizaciones más grandes y de más rápido crecimiento del mundo, incluyendo Facebook, Google, Adobe, Alcatel Lucent y Zappos basan sus sistemas en MySQL.

Tanto el software del servidor de MySQL en sí y las bibliotecas de cliente utilizan una doble licencia de distribución. Se ofrecen bajo licencia GPL, a partir del 28 de junio de 2000 (que Oracle ha ampliado con una excepción de licencia de software libre) o una licencia propietaria (GOSLING, y otros, 2015). Precisamente por esta razón el uso de MySQL no es recomendado para la solución informática propuesta, puesto que Oracle puede retirar en cualquier momento la licencia GPL del DBMS y se estarían incumpliendo las políticas de soberanía tecnológica de la universidad y del país.

- **PostgreSQL**

Es un avanzado servidor de base de datos SQL, disponible en una amplia gama de plataformas. Uno de los beneficios más claros de PostgreSQL es que es de código abierto, además de que sus versiones son mantenidas por largos periodos de tiempo y requiere poco o ningún mantenimiento en muchos casos (RIGGS, y otros, 2010).

Capítulo 1: Fundamentación teórica

Entre las características fundamentales de PostgreSQL se pueden encontrar (RIGGS, y otros, 2010):

- Excelente cumplimiento de los estándares SQL 2008.
- Arquitectura cliente-servidor.
- Diseño altamente concurrente donde los lectores y escritores no se bloquean entre sí.
- Altamente configurable y extensible para muchos tipos de aplicaciones.
- Integridad referencial (*foreign keys*).
- Triggers Vistas (Views).
- Control de versionado concurrente (MVCC).

PostgreSQL se caracteriza además por ser un sistema de base de datos de propósito general que permite crear sus propias funciones de servidor en una docena de idiomas diferentes. Es altamente extensible, de modo tal que es posible agregar sus propios tipos de datos, operadores y los tipos de índices.

- **MongoDB**

MongoDB (de la palabra en inglés humongous que significa enorme) es un sistema de base de datos de código abierto que lidera el movimiento NoSQL. Está escrito en C++ y ofrece alto rendimiento, alta disponibilidad y fácil escalabilidad. MongoDB comenzó a expandirse en el 2009 y es una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta. MongoDB ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidados. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria. La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel empresarial y flexibilidad operativa. Está orientada a documentos, lo que quiere decir, que en un único documento es capaz de almacenar toda la información necesaria que define un producto o un cliente, aceptando

Capítulo 1: Fundamentación teórica

todo tipo de datos (incluidos *arrays* y otros subdocumentos). Todo ello sin tener que seguir un esquema predefinido.

Entre sus principales características se incluyen (MongoDB Inc. 2014):

- **Flexibilidad:** MongoDB almacena los datos en documentos JSON¹³ (los cuales son serializados a BSON¹⁴). JSON provee un modelo de datos rico que asigna a la perfección a los tipos de lenguaje de programación nativos, y el esquema dinámico hace evolucionar fácilmente el modelo de datos más que con un sistema con esquemas forzados como RDBMS¹⁵.
- **Potencia:** MongoDB proporciona una gran cantidad de características que los RDBMS tradicionales brindan: índices secundarios, consultas dinámicas, clasificación, ricas actualizaciones, upserts (actualiza si existe el documento, inserta si no lo existe), y fácil agregación. Esto le da la amplitud de funcionalidades a la que estamos acostumbrados en un RDBMS, con la capacidad de flexibilidad y escalado que el modelo no relacional permite.

Velocidad/Escalado: Al mantener los datos relacionados juntos en documentos, las consultas pueden ser mucho más rápidas que en una base de datos relacional donde los datos relacionados son separados en múltiples tablas y luego tienen que ser unidos más tarde. MongoDB también hace que sea fácil de escalar su base de datos, permite ampliar o reducir el clúster de forma lineal mediante la adición de más máquinas. Lo cual es muy importante en la web cuando la carga puede aumentar de repente y dejar sin respuesta al sitio web.

Se concluye que se tomara MongoDB como SGBD debido que al emplear MongoDB en su versión 3.0.4 con Node.js, este aporta más velocidad al servidor web cuando se ejecutan operaciones sobre la base de datos, brinda flexibilidad al modelar la solución y no crea los cuellos de botellas como lo hacen otros SGBD ya que al utilizar JavaScript como lenguaje de programación en todos los niveles se logra un correcto funcionamiento al procesar grandes cantidades de datos y peticiones.

¹³ Siglas en inglés para JavaScript Object Notation

¹⁴ Siglas en inglés para Binary JSON

¹⁵ Siglas en inglés para Relational Database Management System

Capítulo 1: Fundamentación teórica

1.6 Proceso de desarrollo de software

El proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requisitos de software, estos requisitos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo” (Booch, y otros, 2000).

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto de software que reúna los requisitos del cliente. No existe un proceso de software único que sea efectivo para todos los contextos. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de software por lo que se hace necesario contar con una metodología que garantice la satisfacción del cliente para que se puedan alcanzar los objetivos del proyecto.

Metodología de desarrollo de software

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar, inspirado por otras disciplinas de la ingeniería. El desarrollo de software no es una tarea fácil; prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo.

Por una parte, se tienen aquellas propuestas más tradicionales o robustas que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán, o sea se basa en el uso exhaustivo de documentación durante todo el ciclo del proyecto. Por otra parte, se encuentran as metodologías ágiles o ligeras las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas (H. Canós, y otros, 2003).

En la Tabla 1 se realiza una comparación de las características de la metodología ágiles y tradicionales al momento de desarrollar un software.

Tabla 1 Comparación de metodologías de desarrollo (Letelier, y otros, 2006)

Crterios	Metodología Ágil	Metodología Tradicional
----------	------------------	-------------------------

Capítulo 1: Fundamentación teórica

Equipos de desarrollo pequeños	<ul style="list-style-type: none"> • Orientada a proyectos pequeños. • Corta duración (o entregas frecuentes) • Equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio. 	<ul style="list-style-type: none"> • Aplicables a proyectos de cualquier tamaño • Pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
Documentación que se genera	<ul style="list-style-type: none"> • Pocos artefactos • El modelado es prescindible. • Modelos desechables. 	<ul style="list-style-type: none"> • Muchos artefactos • El modelado es esencial. • Mantenimiento de modelos.
Comunicación con el cliente	<ul style="list-style-type: none"> • El cliente es parte del equipo de desarrollo. 	<ul style="list-style-type: none"> • El cliente interactúa con el equipo de desarrollo mediante reuniones
Resistencia a cambios recurrentes	<ul style="list-style-type: none"> • Asimilación de continuos cambios durante el desarrollo del proyecto 	<ul style="list-style-type: none"> • Se espera que no ocurran cambios de gran impacto durante el proyecto

Luego del análisis de las metodologías de desarrollo y el estudio de sus características, etapas de desarrollo y ventajas que posibilitan, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software. El cliente es parte del equipo de desarrollo, el equipo de desarrollo es de solo dos

Capítulo 1: Fundamentación teórica

personas, además de la dificultad que representa para un equipo de desarrollo pequeño el adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo.

1.6.1.1 Metodologías ágiles

El desarrollo ágil de software refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, estas metodologías son imprescindibles en un mundo en el que surgen cambios recurrentemente. Siempre hay que tener en cuenta como programadores que lo que es la última tendencia hoy puede que no exista mañana y por esto existe la metodología ágil donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizado y multidisciplinario. Cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación, se resumen dichas metodologías ágiles (Letelier, y otros, 2006):

- **SCRUM:** Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos diez años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de treinta días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de quince minutos del equipo de desarrollo para coordinación e integración.
- **Crystal Methodologies:** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo, Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).

Capítulo 1: Fundamentación teórica

- **Dynamic Systems Development Method (DSDM):** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.
- **Adaptive Software Development (ASD):** Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.
- **Feature-Driven Development (FDD):** Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.
- **Lean Development (LD):** Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.
- **XP o Extreme Programming:** La "programación extrema" es un proceso de la Metodología Ágil que se aplica en equipos con muy pocos programadores quienes llevan muy pocos procesos en paralelo. Consiste entonces en diseñar, implementar y programar lo más rápido posible, hasta en casos se recomienda saltar la documentación y los procedimientos tradicionales. Se fundamenta en la capacidad del equipo para comunicarse entre sí y las ganas de aprender de los errores propios inherentes en un programador.

Capítulo 1: Fundamentación teórica

La siguiente tabla (ver Tabla 2) compara las distintas aproximaciones ágiles en base a tres parámetros: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo y características más específicas de la propia metodología como son resultados, simplicidad, adaptabilidad y excelencia técnica. También incorpora como referencia no ágil el Capability Maturity Model (CMM).

Tabla 2 Comparación de metodologías ágiles (Letelier, y otros, 2006)

	CMM	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Resultados	2	5	5	4	4	4	5	5
Simplicidad	1	4	4	3	5	3	5	5
Adaptabilidad	2	5	5	3	3	4	4	3
Excelencia técnica	4	3	3	4	4	4	3	4
Prácticas de colaboración	2	5	5	4	3	3	4	5
Media CM	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
Media Total	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

XP está concebida para dirigir las necesidades específicas del desarrollo de software conducido por equipos pequeños. Esta metodología es más adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Está centrada en potenciar la comunicación desarrollador-cliente desde el primer día. Es considerada ligera, flexible, predecible, de bajo riesgo, y no por ello menos científica. Entre otras ventajas pueden mencionarse los pocos requisitos de documentación y planificación,

Capítulo 1: Fundamentación teórica

así como la exigencia de tener siempre el cliente disponible para el desarrollo, implicando una mejor correspondencia entre el producto y la necesidad del negocio (Beck, 2002).

La gran ventaja de XP es su increíble capacidad de respuesta ante imprevistos, aunque por diseño es una metodología que no construye para el largo plazo y para la cual es difícil documentar. XP es un método estupendo para equipos extremadamente pequeños que se centran en un solo cliente.

Luego de haber estudiado estas metodologías se determinó que estas pueden hacer un amplio recorte de roles y artefactos para adaptar el proyecto a pequeños equipos de personas. XP es conocida por la rapidez de la entrega del producto a corto plazo. Se selecciona esta metodología por las características siguientes: El período de desarrollo es corto, las dimensiones del proyecto son pequeñas, el cliente forma parte del desarrollo, el proyecto es sencillo, pequeño y se desarrollará en conjunto con el cliente.

1.7 Conclusiones parciales

En este capítulo se realizó un análisis sobre el estado del arte y los conceptos fundamentales en el que se basa el proyecto extensionista QQSI. Se expuso a grandes rasgos las características del proyecto extensionista en cuestión. Se obtuvo la caracterización de la metodología de desarrollo de software XP que fue la seleccionada por los autores debido a los indicadores mostrados anteriormente. Además, se define que la solución propuesta se desarrollará en el lenguaje de programación JavaScript que es el de mayor tendencia en aplicaciones de juegos multiusuario online con ranking implementado y es una ventaja utilizar el mismo lenguaje de programación en todos los niveles sobre el entorno de desarrollo integrado (IDE) WebStorm de JetBrains.

Capítulo 2: Descripción del sistema

En este capítulo se realizará una breve descripción de las características del sistema y un estudio de los requisitos funcionales y no funcionales que deberá cumplir la aplicación. Así mismo se efectuará la identificación y descripción de los artefactos que serán creados para guiar el diseño de la herramienta propuesta, los que servirán de base a su posterior implementación. Para esta etapa de construcción del sistema se realizarán las interfaces de usuario, las historias de usuario y las tarjetas CRC.

2.1 Herramienta existente para la gestión del proyecto extensionista QQSI

Durante el análisis a la herramienta con que cuenta el proyecto extensionista QQSI para su realización se constataron algunas limitantes como son: que la gestión se realiza de forma manual, solo se puede usar cuando se realiza algún evento ya que los estudiantes no pueden acceder desde distintas locaciones, el estudiante que participa no puede saber que puntuación obtuvo hasta que un miembro del proyecto califique sus respuestas. En caso de necesitar adicionarle nuevas preguntas o temas se debe contactar al encargado del proyecto extensionista físicamente, ya que no posee un mecanismo que valla avalando la calidad de cada pregunta sugerida. Es obligatoria la presencia de un miembro del proyecto extensionista para que la actividad se realice satisfactoriamente. Debido a las limitantes planteadas con anterioridad, de la herramienta existente, se decide realizar una nueva herramienta para la gestión del proyecto extensionista QQSI.

Para la gestión de este proyecto extensionista se puede usar un sistema web que permita (Álvarez Cancio, y otros, 2015):

- Gestionar las competencias y obtener un registro de las mismas por cada usuario.
- Crear competencias en la que se pueda participar desde distintas locaciones.
- Ser utilizado por estudiantes para prepararse de manera individual o colectiva creando su propia competencia de la asignatura o grupo de asignaturas que elija.
- Seleccionar los profesores responsables de cada asignatura, que puedan tener acceso al registro de las competencias creadas relacionadas con su asignatura y analizar los resultados, como así también tener la posibilidad de solicitar la creación o eliminación de preguntas del sistema.

Capítulo 2: Descripción del sistema

- Contar con un sistema de trazas que registre todas las actividades en función de realizar estudios y generar reportes sobre el estado de un alumno, un grupo, un año, una facultad o de la universidad.
- Visualizar el ranking de los resultados en las competencias creadas oficialmente por los miembros del proyecto extensionista.

2.2 Sistema web para el proyecto QQSI

El sistema web propuesto para el proyecto, permitirá al usuario registrarse en el sistema con la introducción de su nombre, usuario, correo y contraseña. Luego de registrado podrá personalizar su perfil, invitar amigos a jugar o unirse a competencias creadas, además le mostrará la información del estado de las competencias, así como notificaciones de interés. El profesor tiene la posibilidad de crear una competencia para los estudiantes de su grupo, además puede solicitar la creación de nuevas preguntas o escoger las ya presentes en el sistema de su asignatura, de todas las del semestre, del año o de la carrera en general. El alumno también podrá utilizar el software para prepararse de manera individual o colectiva creando su propia competencia de la asignatura o grupo de asignaturas que elija. Los profesores seleccionados como responsables de cada asignatura, podrán tener acceso al registro de todas las competencias creadas relacionadas con su asignatura y permitir analizar los resultados. De igual forma se podrá acceder a un registro por asignaturas y temas y contará con un sistema de trazas que inspeccionará todas las actividades que se efectúan en función de realizar estudios y generar reportes sobre el estado de un alumno, un grupo, un año, una facultad o de la universidad.

Juego online¹⁶

El juego online consiste en iniciar una competencia, la cual se puede jugar de forma individual o invitando a otros usuarios. Cuenta con modalidades de juego tales como:

1. QQSI: este modo de competencia mantendrá a los usuarios en juego hasta su primera respuesta incorrecta, después de haberse equivocado la competencia finaliza y solo acumula punto si el usuario fue capaz de alcanzar una ronda superior a la 10.

¹⁶La **definición de juego** en línea es la de un **juego** digital en el que es necesaria una conexión de red activa para poder jugar. Esto incluye no sólo **juegos** en Internet, sino también los que se juegan en línea a través de consolas o por teléfono móvil.

Capítulo 2: Descripción del sistema

2. Supervivencia: esta modalidad mantendrá al usuario en competencia mientras que su puntuación sea positiva. Cada pregunta tendrá un valor que se sumará o restará en dependencia de si su respuesta es correcta o no.
3. Máxima puntuación: en esta modalidad se jugarán las 20 rondas de la competencia sin el uso de comodines. Ganará el que mayor cantidad de puntos logre acumular.

Las modalidades están conformadas por rondas y en cada ronda se realiza una pregunta diferente. Las preguntas van aumentando su nivel de complejidad a medida que avancen las rondas. Una vez que la competencia haya iniciado, el usuario no podrá participar en ella, pero tendrá la posibilidad de crear una nueva, iniciar una que esté creada o participar como invitado u observador en una competencia ya iniciada. La modalidad de invitado u observador no permitirá al usuario responder, ni ver las preguntas que van saliendo en cada ronda, pero sí visualizar las puntuaciones de los concursantes de la competencia en ejecución hasta que la misma culmine.

█ Sistema de ranking y puntuación

Cada pregunta tendrá una puntuación en dependencia de su complejidad, la ronda en que se encuentre y el tiempo que demore en responder el estudiante. De un total de 30 segundos para responder cada pregunta, si responde en los primeros 10 segundos acumulará el total de los puntos para esa pregunta, si demora de 11 a 20 segundos, se penaliza con el 30 por ciento de los puntos, se responde en los últimos 30 segundos la penalización será del 60 por ciento y si no responde antes de que termine el tiempo no acumula puntos. Cuando acabe la competencia se sumarán todos los puntos acumulados y se le asignarán al perfil del usuario, para poder ser visualizados en el ranking.

El sistema de ranking contará con un listado decreciente teniendo en cuenta las puntuaciones de las competencias.

█ Actores del sistema

El sistema será utilizado por diferentes tipos de usuarios, agrupándose en los siguientes roles: administrador general, profesor principal del año, jefe de asignatura, profesor y estudiante. A medida que un usuario solicite la creación de nuevos elementos, estas peticiones son enviadas al usuario con el rol superior para su aprobación. A continuación, se muestran los roles ordenados por jerarquía y se explican los privilegios asociados a cada rol (Tabla 3).

Capítulo 2: Descripción del sistema

No.	Actor	Descripción
1	Administrador General	<ul style="list-style-type: none">• Posee todos los privilegios del sistema.• Aprueba cada petición.
2	Profesor principal	<ul style="list-style-type: none">• Propone crear una asignatura o una disciplina.
3	Jefe de asignatura	<ul style="list-style-type: none">• Propone crear un tema.
4	Profesor	<ul style="list-style-type: none">• Visualiza la actividad de los estudiantes en el sistema.
5	Estudiante	<ul style="list-style-type: none">• Propone crear una pregunta.• Participa en una competencia.• Participa como invitado en una competencia.

2.3 Requisitos funcionales y no funcionales

La identificación de los requisitos permite tener una correcta visión del producto que se quiere construir. Estos elementos se clasifican en dos grupos: requisitos funcionales y no funcionales. La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requisitos. A continuación, se especifican los diferentes requisitos con los que debe cumplir el sistema, capturados a través de entrevistas cerradas con el cliente.

Requisitos funcionales

El uso del método empírico de entrevista cerrada con el cliente, permitió capturar los requisitos funcionales y no funcionales para el sistema. En la Tabla 4 se muestran los resultados obtenidos.

Tabla 4 Requisitos funcionales (elaboración propia)

No	Nombre	Descripción	Prioridad	Complejidad
----	--------	-------------	-----------	-------------

Capítulo 2: Descripción del sistema

RF1	Gestionar disciplina	Adiciona, lista, modifica y elimina una disciplina. Se requieren de forma obligatoria los campos nombre y descripción.	Alta	Media
RF2	Gestionar asignatura	Adiciona, lista, modifica y elimina una asignatura. Se requieren de forma obligatoria los campos nombre, descripción y disciplina.	Alta	Media
RF3	Gestionar tema	Adiciona, lista, modifica y elimina un tema. Se requieren de forma obligatoria los campos nombre, descripción y asignatura.	Alta	Media
RF4	Gestionar pregunta	Adiciona, lista, modifica y elimina una pregunta. Se requieren de forma obligatoria los campos nombre, descripción y asignatura.	Alta	Alta
RF5	Gestionar competencia	Adiciona, lista, modifica y elimina una competencia. Se requieren de forma obligatoria los campos nombre, lugar, fecha, modalidad, nivel, rondas, asignaturas, temas, preguntas y usuarios.	Alta	Alta
RF6	Gestionar perfil de usuario	El sistema debe brindar la posibilidad de crear y mostrar los datos del perfil del usuario autenticado.		
RF7	Gestión de administración y funcionamiento	El sistema debe brindar la posibilidad de invitar usuarios a la competencia que ha creado el administrador, el profesor principal o un participante.	Alta	Media

Capítulo 2: Descripción del sistema

RF8	Modo invitado de juego online	El sistema debe brindar la posibilidad de acceder a la competencia que se está desarrollando en modo invitado u observador, es decir, solo va a visualizar el desarrollo de la competencia.	Baja	Baja
RF9	Gestionar trazas	El sistema debe brindar la posibilidad de almacenar las trazas de los usuarios del sistema para su posterior análisis.	Media	Alta
RF10	Iniciar juego online	El sistema debe brindar la posibilidad que se realice la competencia.	Alta	Alta
RF11	Módulo de ranking y sistema de puntuación	El sistema debe brindar la posibilidad de gestionar el ranking de los competidores y un sistema de puntuación que almacene sus logros.	Alta	Baja
RF12	Autenticar usuario	El sistema debe brindar la posibilidad de autenticarse al usuario, introduciendo su usuario y contraseña.	Media	Media

Requisitos no funcionales

Los requisitos no funcionales (RnF) refieren las propiedades o características que tiene el producto, que lo hacen atractivo, usable, rápido y confiable. Se listan a continuación:

- RnF 1. Se debe lograr una correcta estructura de la información, con el empleo de menús, que proporcionen una navegación sencilla.
- RnF 2. Se asignarán los permisos de acceso, escritura, lectura en dependencia del rol que desempeñe cada usuario en el sistema.
- RnF 3. En caso de que el sistema presente alguna falla, los errores se deben mostrar sin detalles de información que pueda comprometer la seguridad e integridad del mismo.

Capítulo 2: Descripción del sistema

- RnF 4. El sistema debe demorar en cada transición un tiempo menor a dos segundos aproximadamente.
- RnF 5. El sistema debe permitir trabajar de manera concurrente a los usuarios.
- RnF 6. El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades en caso de ser necesarias.
- RnF 7. Las paradas de cambios o mantenimiento no deberán interferir en el correcto desempeño del resto del sistema.
- RnF 8. El servidor debe contar con los requisitos mínimos de hardware como son: un microprocesador de 2 núcleos a 2.4GHZ, 4 GB de RAM, 320 GB de almacenamiento y una interfaz de red de 100 Mbit/s.
- RnF 9. El servidor que alojará la aplicación debe estar conectado a una red de computadoras.
- RnF 10. Los ordenadores clientes deben contar como mínimo con un microprocesador Pentium 4 y 1GB de RAM.
- RnF 11. Los ordenadores clientes deben contar con un navegador web como el Mozilla Firefox 42 o superior, o Google Chrome 45 o superior.

Se identificaron doce requisitos no funcionales, teniendo en cuenta el rendimiento que el cliente propone para el sistema en cuanto a fiabilidad, tiempos de respuestas y las condiciones mínimas de hardware para el servidor y los clientes, logrando un adecuado balance entre costo y rendimiento.

Como resultado se obtuvieron un total de doce requisitos funcionales. Luego se dio paso a clasificarlos según su prioridad y complejidad, identificando 9 de prioridad alta, 2 de prioridad media y 1 de baja prioridad, en cuanto a su complejidad: 4 de complejidad alta, 5 de prioridad media y 4 de baja prioridad. Esta clasificación permitirá en la etapa de planificación establecer un orden de desarrollo para cada elemento según su impacto en el sistema.

Capítulo 2: Descripción del sistema

2.4 Fase de planificación

La planificación es la etapa del proceso de desarrollo de software que propone XP como fase inicial para comenzar la construcción de un producto. Durante el transcurso de esta etapa se realiza el proceso de identificación y elaboración de las historias de usuario (HU), además el equipo de trabajo se familiariza con las tecnologías y herramientas seleccionadas para el desarrollo. El cliente define el nivel de prioridad con que se deben implementar las HU, así como la estimación del esfuerzo que costará implementar las mismas. El resultado de la presente fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su elaboración, de forma tal que sea una guía durante el desarrollo.

Historias de usuarios

Las historias de usuario (HU) son como los casos de uso en RUP, con la diferencia de que deben ser escritos por el cliente con un lenguaje sencillo. En esencia, no son más que las ideas del cliente organizadas y agrupadas de acuerdo con su funcionalidad, estableciéndose un orden que permita priorizar sus necesidades, así como definir las que resultan críticas o claves en el momento de desarrollo de la solución. A continuación, aparecen descritas dos de las HU con mayor prioridad y riesgo en el desarrollo (Tabla 5 y Tabla 6). Para consultar todas las tablas de HU puede acudir del [Anexo 1](#) al [Anexo 10](#)

Tabla 5 Historia de usuario no. 8 (elaboración propia)

Historia de usuario	
Número: 8	Nombre: Juego online
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 3 semana
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 3 semana

Capítulo 2: Descripción del sistema

Descripción: El sistema debe brindar la posibilidad de acceder a la competencia que se está desarrollando en modo invitado u observador, es decir, solo va a visualizar el desarrollo de la competencia.

Observaciones:

Tabla 6 Historia de usuario no. 11 (elaboración propia)

Historia de usuario	
Número: 11	Nombre: Módulo de ranking y sistema de puntuación
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: El sistema debe brindar la posibilidad de gestionar el ranking de los competidores y un sistema de puntuación que almacene sus logros.	
Observaciones:	

Estimación de esfuerzos por Historias de Usuario

Para el desarrollo de la solución propuesta, se realizó una estimación de esfuerzo para cada una de las HU, quedando distribuido como se muestra a continuación (Tabla 7):

Tabla 7 Estimación de esfuerzos por historias de usuario (elaboración propia)

No	Historias de usuario	Puntos de estimación
1	Gestionar disciplina	2

Capítulo 2: Descripción del sistema

2	Gestionar asignatura	1
3	Gestionar tema	2
4	Gestionar pregunta	2
5	Gestionar competencia	3
6	Gestionar perfil de usuario	3
7	Autenticarse en el sistema	2
8	Gestión de administración y funcionamiento	3
9	Juego online	3
10	Modo invitado de juego online	1
11	Gestionar trazas	3
12	Módulo de ranking y sistema de puntuación	2

2.5 Plan de iteraciones

Luego de identificar y definir las HU y estimar el esfuerzo propuesto para la realización de cada una de ellas, se precisa establecer el contenido de trabajo de las mismas. Se establece un plan de iteraciones, regulando la cantidad de HU a implementar dentro del rango establecido por la estimación efectuada. Teniendo en cuenta los aspectos antes tratados, la aplicación se desarrollará en tres iteraciones (Tabla 8).

Tabla 8 Plan de iteraciones (elaboración propia)

Iteraciones	Orden de las historias de usuario a implementar	Duración de las iteraciones	
1	1	Gestionar disciplina	2
	2	Gestionar asignatura	1
	3	Gestionar tema	2
	4	Gestionar pregunta	2

Capítulo 2: Descripción del sistema

2	5	Gestionar competencia	3
	6	Gestionar perfil de usuario	2
	7	Autenticarse en el sistema	3
	8	Gestión de administración y Funcionamiento	3
	9	Juego online	3
3	10	Modo invitado de juego online	1
	11	Gestión de trazas	3
	12	Módulo de ranking y sistema de Puntuación	2

2.6 Plan de entregas

El plan de entrega es el compromiso final del equipo de desarrollo con el cliente, pues en él se define cuándo será entregado el producto, dónde se lleva a cabo la estimación del tiempo necesario para entregar al cliente las versiones del producto a medida que se cumplan los requisitos del software. La siguiente Tabla 9 muestra el plan de duración con un aproximado del tiempo en que deben estar las versiones.

Tabla 9 Plan de entregas (elaboración propia)

	Iteración 1	Iteración 2	Iteración 3
Cantidad de HU	4	5	3
Fecha de entrega	07/03/2016	20/04/2016	30/05/2016

2.7 Interfaz de usuario

Los prototipos de interfaz de usuario pueden ser prototipos formales o informales, ejecutables o no ejecutables, de baja fidelidad o de alta fidelidad. Por ejemplo, un prototipo de interfaz de usuario puede variar desde una serie de imágenes representando capturas de pantalla de algunas páginas HTML interactivas. El formato que toma el prototipo no es

Capítulo 2: Descripción del sistema

relevante. Lo que es importante recordar es el objetivo del prototipo de interfaz de usuario (para explorar o validar un diseño de interfaz de usuario) (Betania Technologies, 2006). A continuación se muestran algunas capturas de pantalla de la aplicación:

En la Ilustración 1 se muestra la interfaz correspondiente al menú de adicionar asignatura, donde el usuario podrá ver un listado con todas las asignaturas que existen dentro de la aplicación. En esta interfaz podrá adicionar nuevas asignaturas (opción 1), buscar la que desee (opción 3) o marcarla activa o no como se muestra en la opción 2.

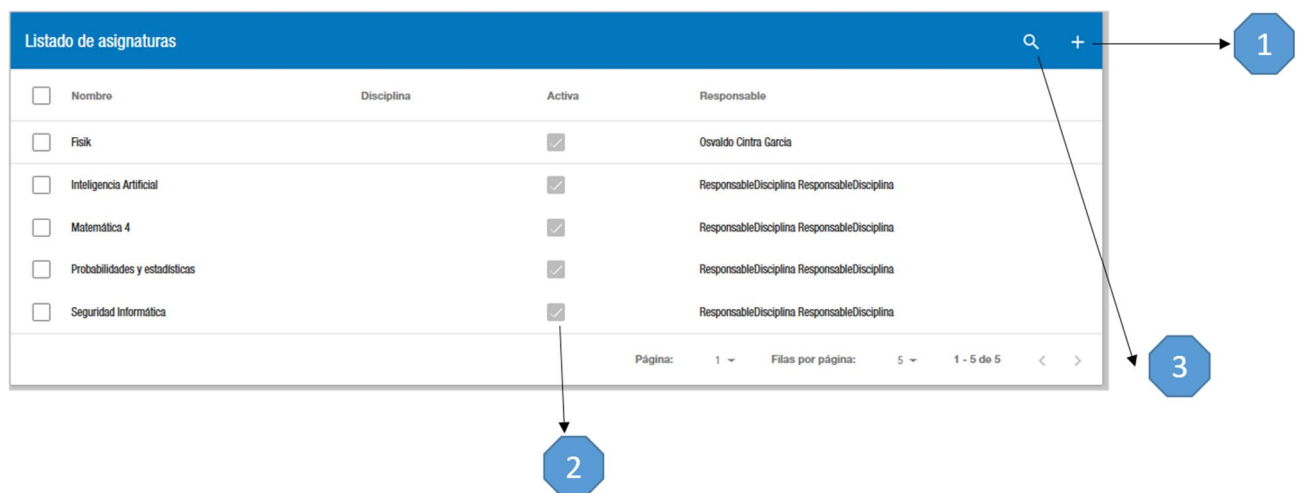


Ilustración 1 Interfaz adicionar asignatura (elaboración propia)

Si el usuario desea editar una asignatura, podrá marcarla y se mostrará la Ilustración 2, donde tendrá la opción 1 para eliminarla y la opción 3 para editarla. Las interfaces mostradas en la ilustración 1 y 2 son similares para agregar o editar los temas, disciplinas y preguntas.

Capítulo 2: Descripción del sistema

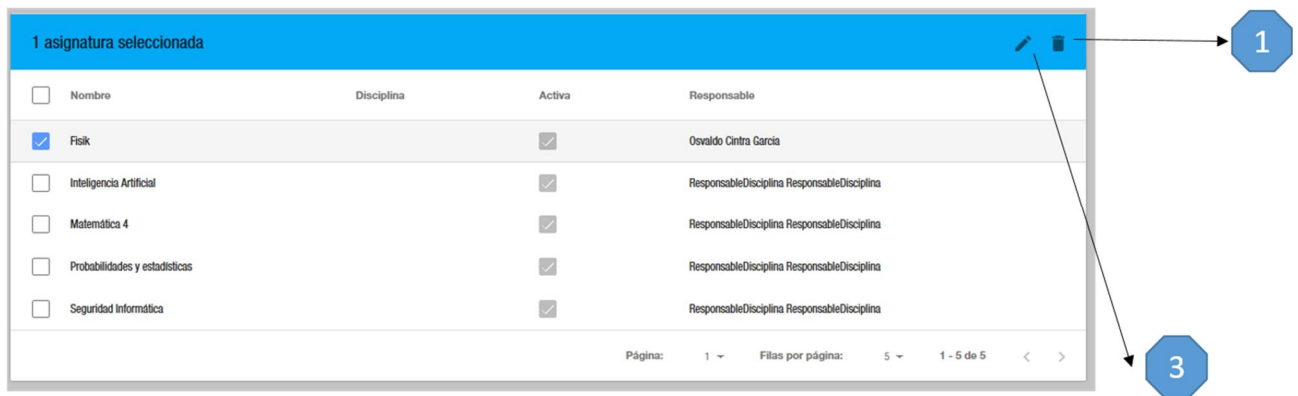


Ilustración 2 Interfaz editar asignatura (elaboración propia)

La Ilustración 3 muestra los detalles de una competencia, así como las posibles operaciones que se les puede realizar. En la parte superior la opción 1 mostrará el nombre del usuario que esté registrado en ese momento. En esta interfaz se podrá:

- Iniciar una competencia (opción 2).
- Ver los participantes elegidos para competir (opción 3).
- Ver los detalles de la competencia como son: lugar, fecha, modalidad y nivel (opción 4).
- Ver las asignaturas y temas de los que tratará la competencia (opciones 5 y 6 respectivamente).

Capítulo 2: Descripción del sistema

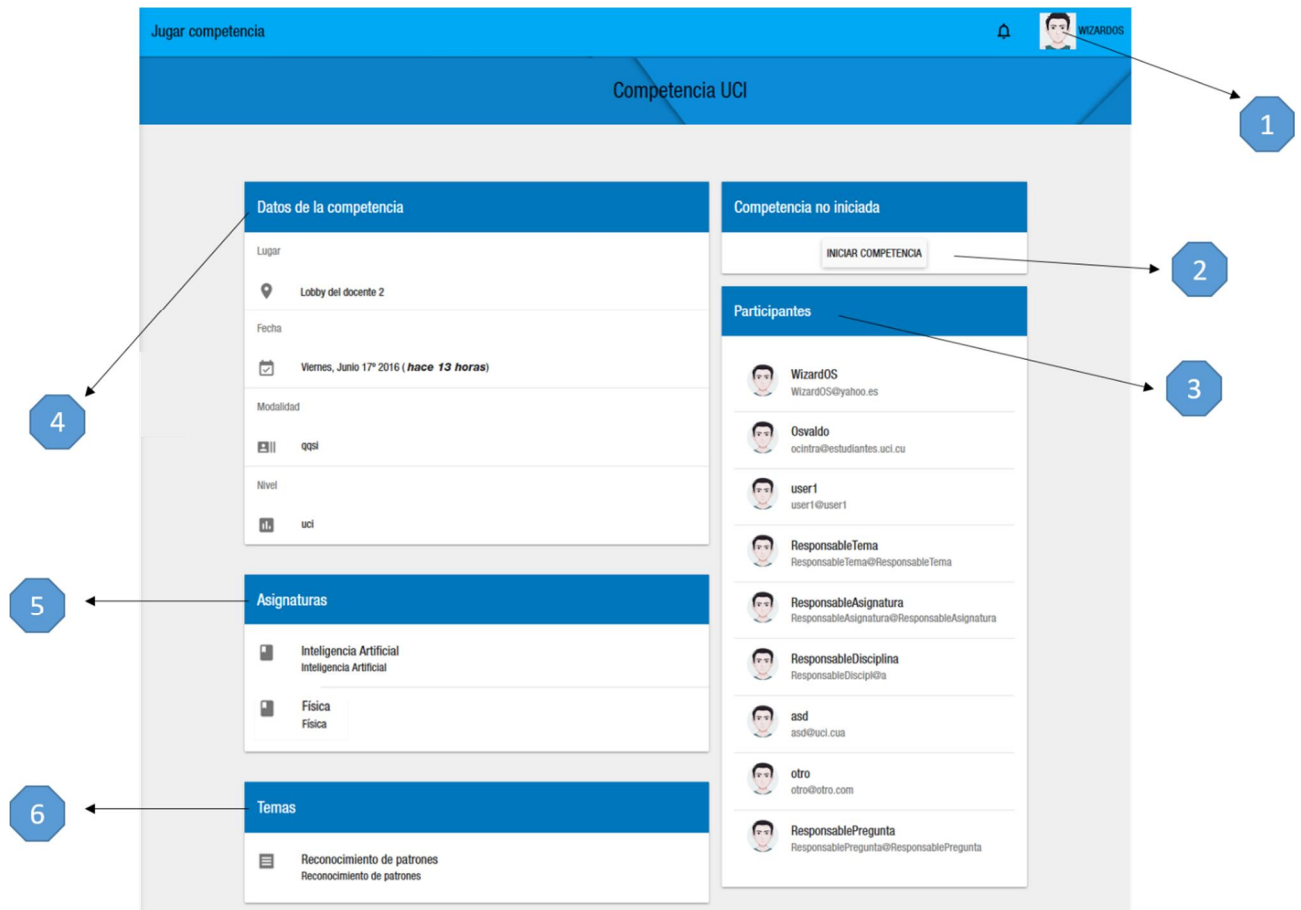


Ilustración 3 Datos de la competencia

En la Ilustración 4, se muestra la consola de administración. El administrador podrá:

- acceder a gestionar disciplinas, asignaturas, temas y preguntas (opción 1).
- Ver los usuarios registrados en el sistema (opción 2).
- Ver las notificaciones que le han sido enviadas (opción 3).
- Ver las solicitudes que faltan por su aprobación (opción 4).

Capítulo 2: Descripción del sistema

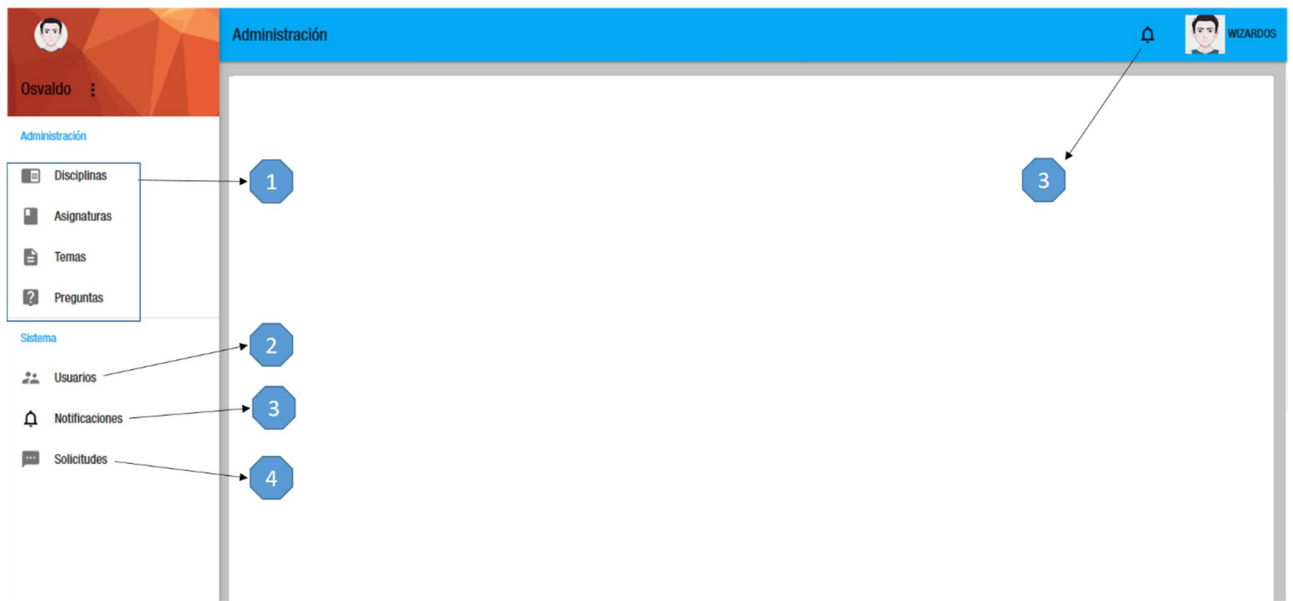


Ilustración 4 Consola de administración

La Ilustración 5 muestra los detalles de una competencia en ejecución, aquí el usuario puede:

- Visualizar la pregunta correspondiente a la ronda y sus posibles respuestas (opción 1).
- Luego de marcar la respuesta que crea correcta debe enviar su respuesta para pasar a la siguiente ronda (opción 4).
- Terminar la competencia (opción 3).
- Visualizar el tiempo que le queda para responder la pregunta mostrada (opción 2).
- Visualizar los puntos que ha ido acumulando.
- Visualizar la ronda en la que se encuentra.

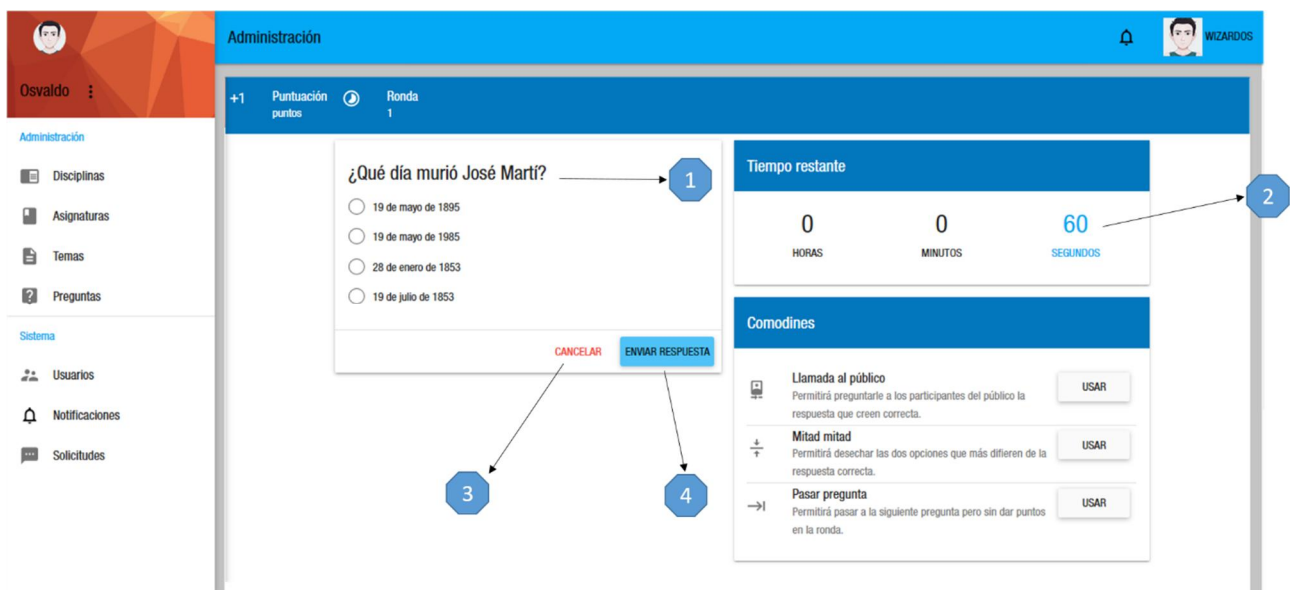


Ilustración 5 Competencia en ejecución

Capítulo 2: Descripción del sistema

2.8 Tarjetas CRC

Las tarjetas CRC constituyen una metodología creada por Kent Beck que define la relación Clase –Responsabilidad – Colaboración y ayuda a pensar en términos de objetos. Primeramente, deben identificarse las clases que compondrán al sistema.

Para comenzar este proceso se utiliza la técnica de tormenta de ideas donde el equipo de desarrollo aporta opiniones para localizar todas las posibles clases. Una vez identificadas se procede a refinarlas, es decir, dejar solo aquellas que sean básicas para el sistema que son las que poseen responsabilidades y participan activamente.

Las tarjetas CRC se crean a partir de la lista de clases básicas identificadas y constituyen una primera aproximación a los objetos que luego se van a utilizar en el desarrollo del sistema. A continuación, se muestra la tarjeta CRC correspondiente a la clase Competencia (Tabla 10). Para consultar el resto de las tarjetas CRC puede consultar del [Anexo 11](#) al [Anexo 15](#).

Tabla 10 Tarjeta CRC de la clase: Competencia (elaboración propia)

Clase: Competencia	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Posibilita añadirle asignaturas, los temas asociados a esas asignaturas y las preguntas asociadas a esos temas.2. Permite preparar la competencia, asignando como estado de la competencia “en espera”.3. Permite iniciar una competencia.4. Asigna el tiempo necesario para responder una pregunta.5. Permite la opción de responder las preguntas de una competencia.6. Permite pasar a la siguiente pregunta.7. Permite invitar un nuevo usuario.8. Permite terminar la competencia.	<ol style="list-style-type: none">1. competenciasAdd2. competenciasEdit3. competenciasList4. competenciasPlay

Capítulo 2: Descripción del sistema

2.9 Conclusiones parciales

Durante la ejecución de este capítulo se definieron las funcionalidades que debe tener la herramienta dando respuesta a la descripción del sistema. Se obtuvo la confección de los artefactos generados por la metodología XP durante la etapa de Diseño. Así como la estimación de la implementación y entrega del producto final correspondientes a la etapa de Planificación. Se analizó el flujo actual de los procesos para entender las necesidades del cliente. Además, se describieron los roles y privilegios que tendrán cada uno de ellos en el sistema, así como los requisitos no funcionales para que la herramienta funcione con la calidad requerida.

Capítulo 3: Validación y prueba

Capítulo 3: Validación y prueba

Las pruebas de software constituyen un instrumento para determinar el nivel de calidad de un producto. Dentro de las fases propuestas por la metodología XP se lleva a cabo la implementación y las pruebas del sistema. Dicha metodología divide las pruebas en dos grupos: pruebas unitarias, encargadas de verificar el código, diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al terminar una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (Shulcoper, 1999). Durante las iteraciones las historias de usuarios seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. Las pruebas unitarias se realizan con el objetivo de verificar el correcto funcionamiento de los procedimientos, agrupados en los distintos componentes.

3.1 Pruebas

El proceso de pruebas es uno de los pilares fundamentales de la metodología XP, el cual ayuda al cliente a verificar y concretar las funcionalidades de las HU, por lo que favorece la comunicación entre el cliente y el equipo de desarrollo. Esta filosofía ayuda a identificar y corregir fallos u omisiones cometidas en las mismas, por lo que se reduce el número de errores no detectados, así como el tiempo entre la introducción de estos en el sistema y su detección. Permite identificar HU adicionales que no fueran obvias para el cliente o en las que el cliente no hubiese pensado de no enfrentarse a dicha situación, contribuyendo a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones (Gutiérrez, y otros, 2010).

La metodología XP divide las pruebas en dos grupos: pruebas unitarias y las pruebas de aceptación. Las primeras son desarrolladas por los programadores, se escriben antes que el código y son las encargadas de verificar el mismo de forma automática. Mientras que las últimas son destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

Capítulo 3: Validación y prueba

Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos y componentes deben de pasar las pruebas unitarias antes de ser liberados o publicados. Todo código liberado debe pasar correctamente las pruebas unitarias lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo (Salazar Martínez, 2012). La técnica utilizada para la aplicación de esta prueba fue la del camino básico. El fragmento del código utilizado se muestra a continuación (Ilustración 6):

```
submit(valid, ev) {
  if (valid) {
    this.disciplina.owner = Meteor.user()._id;
    this.disciplina.active = false;
    Disciplinas.insert(this.disciplina);

    if (this.done) {
      this.done();
    }
    let $state = this.$state;
    var confirm = this.$mdDialog.confirm()
      .title('La disciplina se ha creado.')
      .textContent('¿Desea adicionar una disciplina nueva?')
      .ariaLabel('Adicionar disciplina')
      .targetEvent(ev)
      .ok('Si')
      .cancel('En otro momento');

    this.$mdDialog.show(confirm).then(function () {
      $state.go('admin.disciplinas add');
    }, function () {
      $state.go('admin.disciplinas list');
    });
    this.reset();
  }
}

reset() {
  this.disciplina = {};
}
```

Ilustración 6 Código fuente utilizado para realizar la prueba de caja blanca. (elaboración propia)

A continuación se enumeran las sentencias de código del método disciplinasAdd() a modo de ejemplo (Ilustración 7).

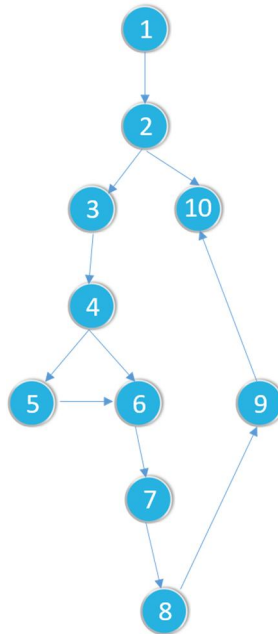


Ilustración 7 Grafo de flujo asociado a la funcionalidad disciplinas Add(). (elaboración propia)

La complejidad ciclomática es la métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Esta métrica calcula la cantidad de caminos independientes de cada una de las funcionalidades del programa. También provee el límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez (Pressman, 2005).

Luego de haber construido el grafo se realiza el cálculo de la complejidad ciclomática mediante las tres fórmulas descritas a continuación, las cuales deben arrojar el mismo resultado para asegurar que el cálculo de la complejidad sea el correcto.

1. $V(G) = (A - N) + 2$ donde **A** es el número de aristas del grafo de flujo y **N** es el número de nodos del mismo.

$$V(G) = (11 - 10) + 2$$

$$V(G) = 3$$

2. $V(G) = P + 1$ donde **P** es el número de nodos predicados contenidos en el grafo de flujo (se denomina nodo predicado a los nodos de los cuales parten dos o más aristas).

$$V(G) = 2 + 1$$

$$V(G) = 3$$

Capítulo 3: Validación y prueba

3. $V(G) = R$ donde R representa la cantidad total de regiones.

$$V(G) = 3$$

Dado que el cálculo de las tres fórmulas anteriormente mencionadas arrojó el mismo resultado se puede plantear que la complejidad ciclomática del método es tres. Esto significa que existen tres posibles caminos por donde el flujo puede circular. Este valor representa el número mínimo de casos de pruebas para el procedimiento tratado.

- **Camino básico #1:** 1 – 2 – 10
- **Camino básico #2:** 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10
- **Camino básico #3:** 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10

Para cada camino básico determinado se realiza un diseño de caso de prueba:

Caso de prueba para el camino básico #1: Si valid == false

Caso de prueba para el camino básico #2: Si valid == true, si this.done == false

Caso de prueba para el camino básico #3: Si valid == true, si this.done == true

Un camino exitoso sería aquel que finaliza en el nodo diez (10) que indica que la disciplina se agregó correctamente. La complejidad ciclomática tiene un valor de 3 lo que permitirá definir la cantidad de casos de uso para cada funcionalidad del sistema.

■ Pruebas de aceptación

Las pruebas de aceptación son las validaciones que se le realizan al sistema para ver si su funcionamiento está acorde a lo esperado. Estas pruebas son “pruebas de caja negra” que se realizan partiendo de las historias de usuario y pueden tener varias pruebas hasta que quede asegurada la aceptación del sistema desarrollado. Un sistema está completamente aceptable cuando quedan satisfechos todos los requisitos funcionales especificados por el cliente, teniendo en cuenta además los requisitos no funcionales, así como a los distintos recursos. Las técnicas de caja negra se realizan sobre la interfaz del sistema y sin conocer la lógica de este; el sistema recibe un conjunto de entradas y produce salidas, a partir de las cuales se comprobará si el comportamiento es el esperado o no. A continuación se

Capítulo 3: Validación y prueba

presenta un conjunto de casos de prueba destinados a comprobar el funcionamiento del sistema.

Iteración 1: La presente iteración de pruebas tiene como objetivo comprobar la correcta implementación de varias HU y detectar posibles no conformidades, la relación de HU involucradas en el proceso es la siguiente: gestionar disciplina, gestionar asignatura, gestionar tema y gestionar pregunta. A continuación, se muestran los casos de prueba Registrarse en el sistema e Insertar disciplina (Tablas 11 y 12). Puede consultar el resto de casos de prueba de la iteración 1 en los [Anexo 16](#) al [Anexo 22](#).

Tabla 11 Caso de prueba Registrarse en el sistema (elaboración propia)

Caso de Prueba	
Código: HU7_P1	Historia de Usuario: autenticarse en el sistema.
Nombre: Registrarse en el sistema.	
Descripción: Prueba de funcionalidad autenticarse en el sistema.	
Condiciones de ejecución: El usuario debe tener un usuario y contraseña para poder ingresar al sistema.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando usuario y contraseña.	
Resultado esperado: El usuario accedió correctamente al sistema.	
Evaluación de la prueba: Satisfactoria.	

Tabla 12 Caso de prueba Insertar disciplina (elaboración propia)

Caso de Prueba	
Código: HU1_P2	Historia de Usuario: gestionar disciplina.
Nombre: Insertar disciplina.	

Capítulo 3: Validación y prueba

Descripción: Prueba de funcionalidad para la inserción de una disciplina.
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando nombre, descripción y responsable.
Resultado esperado: La disciplina se ha creado.
Evaluación de la prueba: Satisfactoria.

Iteración 2: La presente iteración de pruebas tiene como objetivo comprobar la correcta implementación de varias HU y detectar posibles no conformidades, la relación de HU involucradas en el proceso es la siguiente: gestionar competencia, gestionar perfil de usuario, juego online, gestión de administración y funcionamiento. A continuación, se muestran los casos de prueba Crear competencia y Eliminar competencia (Tablas 13 y 14). Puede consultar el resto de casos de prueba de la iteración 2 en los [Anexo 23](#) al [Anexo 26](#).

Tabla 13 Caso de prueba Crear competencia (elaboración propia)

o	
Código: HU5_P10	Historia de Usuario: gestionar o crear competencia.
Nombre: Crear competencia.	
Descripción: Prueba de funcionalidad para la inserción de una pregunta.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando nombre, lugar y seleccionando nivel,	

Capítulo 3: Validación y prueba

modalidad y cantidad de rondas, además de seleccionar las preguntas que compondrán la competencia.

Resultado esperado: La competencia ha sido insertada.

Evaluación de la prueba: Satisfactoria

Tabla 14 Caso de prueba Eliminar competencia (elaboración propia)

Caso de Prueba	
Código: HU5_P11	Historia de Usuario: gestionar o crear competencia.
Nombre: Eliminar competencia.	
Descripción: Prueba de funcionalidad para la eliminación de una competencia.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la eliminación de la pregunta deseada.	
Resultado esperado: La pregunta ha sido eliminada.	
Evaluación de la prueba: Satisfactoria	

Iteración 3: La presente iteración de pruebas tiene como objetivo comprobar la correcta implementación de varias HU y detectar posibles no conformidades, la relación de HU involucradas en el proceso es la siguiente: modo invitado de juego online y sistema de ranking y puntuación. A continuación, se muestran el caso de prueba Acceder como invitado (Tabla 15). Puede consultar el resto de casos de prueba de la iteración 3 en el [Anexo 27](#).

Tabla 15 Caso de prueba: Entrar en modo invitado (elaboración propia)

Capítulo 3: Validación y prueba

Caso de Prueba	
Código: HU10_P16	Historia de Usuario: modo invitado de juego online.
Nombre: Acceder como invitado.	
Descripción: Un usuario entrara en modo invitado solo para visualizar una competencia en ejecución.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema. Debe existir una competencia en ejecución.	
Pasos de ejecución: El usuario procede a presionar la opción de modo invitado en la lista de competencia en ejecución.	
Resultado esperado: El usuario pasa a la vista de modo invitado para visualizar la competencia seleccionada.	
Evaluación de la prueba: Satisfactoria	

3.2 Resultados obtenidos

En la metodología XP, la aplicación de pruebas se realiza de forma iterativa para comprobar la correcta implementación de las historias de usuario. Se aplicaron un total de 17 casos de prueba que se distribuyeron en 3 iteraciones. En la Ilustración 8 se refleja en una gráfica la

Capítulo 3: Validación y prueba

cantidad total de funcionalidades probadas y las no conformidades obtenidas por cada iteración.

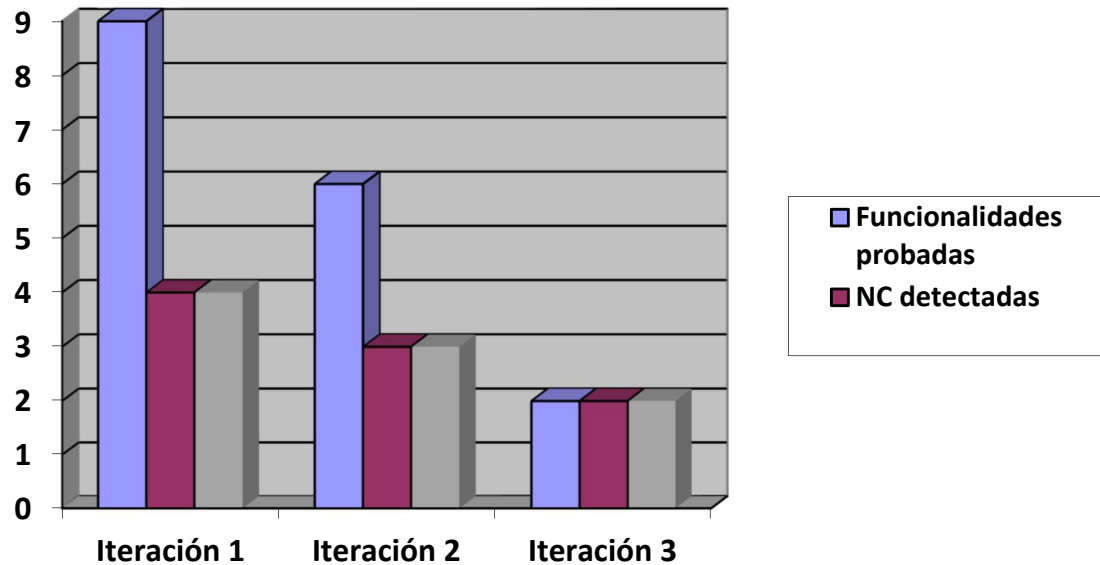


Ilustración 8 Resultados de las pruebas (elaboración propia)

A continuación, se describe para cada iteración el tipo de error detectado:

Iteración 1

Se comprobaron un total de 9 funcionalidades, detectándose un total de 4 no conformidades.

Los errores detectados fueron:

- 1 error ortográfico
- 2 errores de validaciones de campos
- 1 errores de interfaz.

Iteración 2

Se comprobaron un total de 6 funcionalidades, detectándose un total de 3 no conformidades.

Los errores detectados fueron:

- 1 error ortográfico.
- 2 errores de validación.
- 1 error de interfaz.

Iteración 3

Capítulo 3: Validación y prueba

Se comprobaron un total de 2 funcionalidades, detectándose un total de 2 no conformidades. Los errores detectados fueron:

- 2 errores de validación.

Para la realización de las pruebas de aceptación se realizaron diecisiete (17) casos de pruebas, que arrojaron nueve (9) no conformidades. Al finalizar la corrección de estas no conformidades se puso fin al proceso de pruebas de aceptación, obteniendo resultados satisfactorios.

Las no conformidades, no significativas, se centraron en errores ortográficos como: omisiones de tildes, paréntesis y no envío de notificaciones, y las significativas, en errores relacionados con la gestión de las competencias y el sistema de puntuación. Como parte de la estrategia de prueba, se presentó la herramienta al Grupo de Calidad del Centro de Gobierno Electrónico (CEGEL) detectándose 5 no conformidades referentes a problemas de interfaces y 17 no conformidades de funcionalidad para un total de 22 no conformidades, las cuales fueron resueltas en 2 iteraciones. En el [Anexo 28](#) se presenta el acta de liberación emitida por CEGEL que abala estos resultados y en el [Anexo 29](#) y [Anexo 30](#) se presentan las actas de aceptación emitidas por el Vicedecanato de Extensión y Residencia de la Facultad 3 y la Cátedra de Cultura Científica “Dra. Rosa Elena Simeón Negrín” respectivamente.

3.3 Conclusiones parciales

Al concluir este capítulo la herramienta quedó validada a partir de las pruebas unitarias realizadas a la implementación de las funcionalidades. Estas fueron divididas en tres iteraciones abarcando un total de 17 casos de pruebas y concluyendo en una herramienta libre de no conformidades. El cliente quedó satisfecho con la herramienta dando su aprobación a través de las pruebas de aceptación que se anexan. Se realizaron las tarjetas CRC que fueron de gran ayuda durante la implementación y las pruebas. Además se logró realizar una aplicación con interfaces de usuario que respondieron a los prototipos realizados por el cliente.

Conclusiones

Conclusiones generales

Con la realización de este trabajo se ha cumplido con el objetivo general. Se llegó además a las siguientes conclusiones:

Durante el desarrollo de la investigación se realizó un estudio de diversos procesos que se realizaban para llevar a cabo la competencia QQSI, así como el análisis de algunos sistemas web con fines educativos para llevar a cabo esta tarea. Se evidenció la necesidad de desarrollar un sistema que fuera capaz de automatizar este proceso para dar solución a la problemática planteada ya que las herramientas existentes no cumplían con los requerimientos necesarios para desarrollar la herramienta que se adaptara a la actividad tal y como se requería. La aplicación se desarrolló siguiendo las directrices de la metodología XP, se identificaron las funcionalidades del sistema que sirvieron como base para la implementación y se llevó a cabo con las herramientas seleccionadas. Como parte del proceso de implementación se obtuvo un sistema capaz de brindar a los usuarios un sitio donde poner en práctica y ejercitar sus conocimientos en forma de actividad lúdica y de manera competitiva. Al sistema obtenido se le aplicaron pruebas para comprobar su correcto funcionamiento y de esta forma comprobar que cumple con las expectativas del cliente.

Recomendaciones

Recomendaciones

Como resultado del proceso de investigación e implementación de la solución propuesta se propone como recomendación para el sistema:

1. Proveer a la herramienta de un sistema recomendador que sea capaz de analizar el comportamiento del usuario y recomendar preguntas acordes con el perfil de los usuarios.

Referencias bibliográficas

Bibliografía

Álvarez Cancio, Michel, y otros. 2015. Conceptualización de la herramienta QQSI con fines didácticos para los estudiantes de Ingeniería en Ciencias Informáticas. La Habana : s.n., 2015.

AngularJS: Novice to Ninja. **Sandeep Panda. 2014.** 2014.

Beck, Kent. 2002. Una explicación de la Programación extrema: aceptar el cambio. s.l. : ADDISON-WESLEY, 2002. pág. 189. ISBN 9788478290550.

Ben-Ari, M. 1990. Principles of Concurrent and Distributed Programming. 1990. ISBN 0-13-711821-X.

Betania Technologies. 2006. Betania Technologies. [En línea] 2006. [Citado el: 10 de 06 de 2016.]
http://betaniatech.com/SmallProjects/core.base_rup/workproducts/rup_user_interface_prototype_7237E5AA.html.

Booch, Grady , Ivar, Jacobson y Rumbaugh, James . 2000. El proceso unificado de desarrollo de software. 1era Edición. s.l. : Addison Wesley, 2000. pág. 464.

Castledine, E. y C, Sharkie. 2010. jQuery: Novice to Ninja. 2010.

Coordinación General de Investigación y Desarrollo de Modelos Educativos. 2012. Coordinación General de Investigación y Desarrollo de Modelos Educativos. [En línea] 2012. [Citado el: 25 de Mayo de 2016.]
http://investigacion.ilce.edu.mx/panel_control/doc/c36,evaluacsoft.pdf.

Eguíluz Pérez, Javier. 2008. Introducción a CSS. 2008.

El juego: la experiencia de aprender jugando. **Ortega Ruiz, Rosario. 2003.** 2003, Vol. 1, págs. 765-788. ISBN 84-9700-136-2.

Gáinza Gáinza, Miriam y Paz Aguilera, Armando. 2011. Fundamentos teóricos desde la extensión universitaria en el proceso de formación de los profesionales pedagógicos en el contexto de la universalización. [En línea] Mayo de 2011. [Citado el: 10 de Mayo de 2016.]
<http://www.eumed.net/rev/ced/27/ggpa.htm>.

García, David. 2014. Toyoutome. [En línea] 2014. [Citado el: 10 de 06 de 2016.]
<http://toyoutome.es/blog/%C2%BFpor-que-las-apps-educativas/23687>.

Gil Juárez, Adriana y Vida Mombiela, Tere. 2012. LOS VIDEOJUEGOS. s.l. : UNIVERSITAT OBERTA DE CATALUNYA, 2012.

González Gutiérrez, Enrique . 2006. Aprender a programar. [En línea] 2006. [Citado el: 10 de 06 de 2016.]
http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=590:ique-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192. 31.

Referencias bibliográficas

GOSLING, J, y otros. 2015. The Java® Language Specification. Redwood : Oracle Corporation, 2015. Java SE 8 Edition.

Gutiérrez, J. J, y otros. 2010. Pruebas del sistema en programación extrema. Sevilla : Universidad de Sevilla, 2010.

H. Canós, José, Patricio, Letelier y Carmen Penadés, M^a. 2003. Metodologías Ágiles en el Desarrollo de Software. Valencia : Universidad Politécnica de Valencia, 2003.

Horruitiner Silva, Pedro . 2007. El modelo de acreditación de carreras de la educación superior cubana. 2007.

Impacto de las tecnologías de la información y las comunicaciones en la educación y nuevos paradigmas del enfoque educativo. **Prieto Díaz, Vicente, y otros. 2011.** 1, Ciudad de la Habana : Educación Médica Superior, 2011, Vol. 25. ISSN 0864-2141.

Infante Carbó, César . 2013. Rimed. [En línea] 2013. [Citado el: 16 de Marzo de 2016.] http://www.roca.rimed.cu/index.php?option=com_content&view=article&id=115:el-juego-educativo-informatico-sus-caracteristicas-y-componentes-didacticos-esenciales&catid=114:juegos-educativos&Itemid=117.

Jetbrains. 2010. WebStorm: The Smartest JavaScript IDE. [En línea] Jetbrains, 2010. [Citado el: 06 de Abril de 2016.] <http://www.jetbrains.com/webstorm/?fromMenu>.

Jiménez Rodríguez, Esmeralda. 2006. La importancia del juego. España : s.n., 2006. Vol. III. ISSN 1696-7208.

Joyent Inc. 2014. Node.js. [En línea] 2014. [Citado el: 15 de Febrero de 2016.] <http://nodejs.org/>.

Kabir, Mohammed J. 2002. La biblia de servidor apache 2. Madrid : Anaya Multimedia, 2002. ISBN 84-415-1468-2.

Kirkpatrick, Marshall. 2009. [En línea] 22 de Septiembre de 2009. [Citado el: 4 de Abril de 2016.] http://www.readwriteweb.com/archives/explaining_the_realtime_web_in_100_words_or_less.php.

Letelier, Patricio y Penadés, María del Carmen. 2006. CyTA. [En línea] Universidad Politécnica de Valencia, 2006. [Citado el: 01 de 06 de 2016.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. 26.

LINNOVATE. 2014. MEAN.IO Documentation. [En línea] 2014. [Citado el: 20 de 05 de 2016.] <http://learn.mean.io/>.

López Gabriel, Filiberto. 2014. Juego didáctico y desarrollo del lenguaje. s.l. : Universidad Rafael Landívar, 2014.

LOTUM media GmbH. 2015. androidappsgame. [En línea] 1 de Octubre de 2015. [Citado el: 17 de Junio de 2016.] <http://www.androidappsgame.com/true-or-false/es>.

Referencias bibliográficas

Ludia Inc. 2016. androidappsgame. [En línea] 2016. [Citado el: 16 de Junio de 2016.] <http://www.androidappsgame.com/who-wants-to-be-a-millionaire/es>.

MacCare. 2013. Diseño y Desarrollo MacCare. [En línea] 2013. [Citado el: 18 de 11 de 2015.] <http://www.maccare.com.ar/java.htm>.

Maldonado, Enrique José y Calixto, Alejandro. 2012. Diseño de Bases de Datos. s.l. : Alfa Omega, 2012.

Martínez González, Lourdes del Carmen . 2008. Lúdica como estrategia didáctica. [En línea] e-Scholarum, 2008. [Citado el: 10 de Mayo de 2016.] <http://genesis.uag.mx/escholarum/vol11/ludica.html>.

MES. 2004. Programa Nacional de Extensión Universitaria. 2004. pág. 57. Vol. 1.

Montesino Perurena, Raydel y Porven Rubier, Joelsy . 2015. Marco de trabajo para la gestión centralizada de trazas de seguridad usando herramientas de código abierto. Ciudad Habana : s.n., 2015. págs. 18-32. Vol. 9.

Mozilla Developer Network y colaboradores individuales. 2016. Mozilla Foundation. [En línea] 12 de Mayo de 2016. [Citado el: 19 de Junio de 2016.]

Musciano, Chuck y Kennedy, Bill. 1999. HTML La guía completa. s.l. : McGraw Hill Interamericana, 1999. 2ª Edición.

Nedelcu, Clement. 2010. Nginx HTTP Server. s.l. : Aanchal Kumar, 2010.

Núñez Rojas, Nemecio . 2010. Concepción del Software educativo. La Webquest, el aula virtual y el desarrollo de competencias para la investigación en los estudiantes del I Ciclo de educación - USAT. [En línea] 2010. [Citado el: 19 de Junio de 2016.] <http://www.eumed.net/libros/2010a/669/>.

Oracle Corporation. 2015. NetBeans IDE. [En línea] 2015. <http://netbeans.org/features/index.html>.

Oracle, Corporation. 2010. Oracle Sun Microsystems. [En línea] 2010. [Citado el: 6 de Abril de 2016.] <https://www.oracle.com/sun/index.html>.

Otero, Abraham. 2007. MySQL vs PostgreSQL ¿cuándo emplear cada una de ellas? javaHispano. [En línea] 10 de Septiembre de 2007. [Citado el: 22 de Febrero de 2010.] http://www.javahispano.org/contenidos/es/mysql_vs_postgresql_cuando_emplear_cada_una_de_ellas.

Pacheco, Rodrigo. 2015. Características de una aplicación web. [En línea] 2015. [Citado el: 29 de 9 de 2015.] Características de una aplicación web.com.

Pérez Marqués, Jorge. 2012. El software educativo. [En línea] Universidad Autónoma de Barcelona, 2012. http://www.lmi.ub.es/te/any96/marques_software.

Pérez TA, Gutiérrez Serrano J, López Pietro R, González A, Vadillo Zorita JA. 2001. Hipermedia, adaptación, constructivismo e instructivismo. [En línea] Revista

Referencias bibliográficas

Iberoamericana de Inteligencia Artificial, 2001. [Citado el: 15 de Marzo de 2016.]
<http://aepia.dsic.upv.es/revista/numeros/12/Perez.pdf>.

Peru, E-LEMENTAL. 20013. ¿QUE SON LAS APLICACIONES WEB? [En línea] 20013.
[Citado el: 18 de 10 de 2015.] <http://www.elemental.com.pe/que-es-una-aplicacion-web/>.

Pressman, Roger S. 2005. Ingeniería de software: Un enfoque práctico. s.l. :
Interamericana de Mexico, 2005.

Ramos Pérez, Luis, y otros. 2008. Centro Nacional de Información de Ciencias Médicas-
Infomed. [En línea] 2008. [Citado el: 18 de Junio de 2016.]
http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.

RIGGS, S. y KROSIGN, H. 2010. PostgreSQL 9 Administration Cookbook. s.l. : Packt
Publishing Ltd, 2010.

Ruiz Echevarría, Hilda. 2013. La motivación profesional de los estudiantes universitarios
y su influencia en la dedicación al estudio: Papel del profesor. s.l. : Pedagogía, 2013.

Salazar Martínez, Eduardo. 2012. Propuesta de Procedimiento para realizar pruebas de
Caja Blanca a las aplicaciones que se desarrollan en lenguaje Python. Colombia : s.n.,
2012. Vol. 13. 1.

Sánchez Cuervo, O. 2009. Papel de la emisora "la voz de la victoria": una experiencia de
extensión universitaria para la difusión de valores. Matanzas : UMCC, 2009.

Sánchez Mendoza, Maria A. 2004. Metodologías del desarrollo de software. 2004.

**Shulcloper, José Ruiz, Arenas, Adolfo Guzmán ,Trinidad, José Francisco Martínez.
1999.** Selección de variables y clasificación supervisada. [aut. libro] José Ruiz Shulcloper.
Enfoque Lógico Combinatorio al Reconocimiento de Patrones. Mexico : Instituto
Politécnico Nacional, 1999, Vol. 1, pág. 150.

Sistemas gestores de bases de datos. **Garzón Pérez, María Teresa. 2010.** 30, Palma del
Río : IES Antonio Gala, Mayo de 2010, Innovación y experiencias educativas, pág. 5.
ISSN: 1988-6047.

SPL Sistemas de Información. 2012. SPL Sistemas de Información. [En línea] 2012.
[Citado el: 6 de Noviembre de 2015.]
http://www.splssi.com/?sec=articulos&subsec=descripcion&v=aplicaciones_web.

Sun Microsystems. 2006. Arquitectura de aplicaciones. [En línea] 2006. [Citado el: 15 de
Febrero de 2014.] <http://java.sun.com/developer/Books/jdbc/ch07.pdf>.

The Eclipse Foundation. 2015. The Eclipse Foundation. [En línea] 2015. [Citado el: 01
de 05 de 2016.] <http://www.eclipse.org/org/>.

The jQuery foundation. 2015. jQuery. [En línea] 2015. [Citado el: 01 de 06 de 2016.]
<http://jquery.com/>.

Referencias bibliográficas

Vargas Jimenez, Antonio, Almuinas Rivero, Jose Luis y Galarza López, Yudith. 2011. Principales problemas que influyen en el tiempo de dedicación al estudio de los estudiantes del curso diurno en los CES adscritos al MES. s.l. : Comunicación interna., 2011.

Zuberman, J. 2000. Juegos en el Agua. 2000.

Anexos

Anexos

Anexo: Historias de Usuarios

Anexo 1 Historia de usuario no. 1 (elaboración propia)

Historia de usuario	
Número: 1	Nombre: Gestionar disciplina
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: El sistema debe brindar la posibilidad de adicionar, modificar, eliminar y listar una disciplina.	
Observaciones:	

Anexo 2

Anexo 2 Historia de usuario no. 2 (elaboración propia)

Historia de usuario	
Número: 2	Nombre: Gestionar asignatura
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1 semana

Anexos

Descripción: El sistema debe brindar la posibilidad de adicionar, modificar, eliminar y listar una asignatura.

Observaciones:

Anexo 3

Anexo 3 Historia de usuario no. 3 (elaboración propia)

Historia de usuario	
Número: 3	Nombre: Gestionar tema
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: El sistema debe brindar la posibilidad de adicionar, modificar, eliminar y listar un tema.	
Observaciones:	

Anexo 4

Anexo 4 Historia de usuario no. 4 (elaboración propia)

Historia de usuario	
Número: 4	Nombre: Gestionar pregunta
Iteración asignada: 1	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Alto	Puntos reales: 1 semana

Anexos

(Alto / Medio / Bajo)	
Descripción: El sistema debe brindar la posibilidad de adicionar, modificar, eliminar y listar una pregunta.	
Observaciones:	

Anexo 5

Anexo 5 Historia de usuario no. 5 (elaboración propia)

Historia de usuario	
Número: 5	Nombre: Gestionar competencia
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: El sistema debe brindar la posibilidad de adicionar, modificar, eliminar y listar una competencia.	
Observaciones:	

Anexo 6

Anexo 6 Historia de usuario no. 6 (elaboración propia)

Historia de usuario	
Número: 6	Nombre: Gestionar perfil de usuario
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana

Anexos

Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: El sistema debe brindar la posibilidad de crear y mostrar los datos del perfil del usuario autenticado.	
Observaciones:	

Anexo 7

Anexo 7 Historia de usuario no. 7 (elaboración propia)

Historia de usuario	
Número: 7	Nombre: Gestión de administración y funcionamiento
Iteración asignada: 2	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 1 semana
Riesgo en desarrollo: Bajo (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: El sistema permitirá tener una jerarquía según los roles para llevar a cabo la aprobación de crear nuevas asignaturas, temas o preguntas.	
Observaciones:	

Anexo 8

Anexo 8 Historia de usuario no. 9 (elaboración propia)

Historia de usuario	
Número: 9	Nombre: Modo invitado de juego online
Iteración asignada: 3	
Prioridad en negocio: Media	Puntos estimados: 1 semana

Anexos

(Alta / Media / Baja)	
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 1 semana
Descripción: El sistema debe brindar la posibilidad de almacenar las trazas de los usuarios del sistema para su posterior análisis.	
Observaciones:	

Anexo 9

Anexo 9 Historia de usuario no. 10 (elaboración propia)

Historia de usuario	
Número: 10	Nombre: Gestionar trazas
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semana
Riesgo en desarrollo: Medio (Alto / Medio / Bajo)	Puntos reales: 2 semana
Descripción: El sistema debe brindar la posibilidad que se realice la competencia.	
Observaciones:	

Anexo 10

Anexo 10 Historia de usuario no. 12 (elaboración propia)

Historia de usuario	
Número: 12	Nombre: Autenticarse en el sistema
Iteración asignada: 3	
Prioridad en negocio: Alta	Puntos estimados: 1 semana

Anexos

(Alta / Media / Baja)	
Riesgo en desarrollo: Medio	Puntos reales: 1 semana
(Alto / Medio / Bajo)	
Descripción: El sistema debe brindar la posibilidad autenticarse, con un usuario y contraseña personal.	
Observaciones:	

Anexo: Tarjetas CRC

Anexo 11

Anexo 11 Tarjeta CRC de la clase: asignatura (elaboración propia)

Clase: asignaturas	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Permite crear una asignatura.2. Permite editar una asignatura.3. Permite listar las asignaturas.	<ol style="list-style-type: none">1. asignaturasAdd2. asignaturasEdit3. asignaturasList

Anexo 12

Anexo 12 Tarjeta CRC de la clase: notificaciones (elaboración propia)

Clase: notificaciones	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Permite crear una notificación.2. Muestra las notificaciones en el ícono de notificación.3. Muestra la lista de las notificaciones en la barra lateral después de haber interactuado con el ícono de notificación.	<ol style="list-style-type: none">1. notificacionesList2. notificacionesSidebar

Anexos

Anexo 13

Anexo 13 Tarjeta CRC de la clase: disciplinas (elaboración propia)

Clase: disciplinas	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Permite crear una disciplina.2. Permite editar una disciplina.3. Permite listar las disciplinas.	<ol style="list-style-type: none">1. disciplinasAdd2. disciplinasEdit3. disciplinasList

Anexo 14

Anexo 14 Tarjeta CRC de la clase: preguntas (elaboración propia)

Clase: preguntas	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Permite crear una pregunta.2. Permite editar una pregunta.3. Permite listar las preguntas.	<ol style="list-style-type: none">1. preguntasAdd2. preguntasEdit3. preguntasList

Anexo 15

Anexo 15 Tarjeta CRC de la clase: temas (elaboración propia)

Clase: temas	
Responsabilidades	Colaboradores
<ol style="list-style-type: none">1. Permite crear un tema.2. Permite editar un tema.3. Permite listar los temas.	<ol style="list-style-type: none">1. temasAdd2. temasEdit3. temasList

Anexos

Anexo: Casos de prueba

Anexo 16

Anexo 16 Caso de prueba Eliminar disciplina (elaboración propia)

Caso de Prueba	
Código: HU1_P3	Historia de Usuario: gestionar disciplina.
Nombre: Eliminar disciplina.	
Descripción: Prueba de funcionalidad para la eliminación de una disciplina.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la eliminación de la disciplina deseada.	
Resultado esperado: La disciplina ha sido eliminada.	
Evaluación de la prueba: Satisfactoria.	

Anexo 17

Anexo 17 Caso de prueba Adicionar asignatura (elaboración propia)

Caso de Prueba	
Código: HU2_P4	Historia de Usuario: gestionar asignatura.
Nombre: Adicionar asignatura.	
Descripción: Prueba de funcionalidad para la inserción de una asignatura.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	

Anexos

Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando nombre, descripción, responsable y seleccionando la disciplina a la que pertenecerá.

Resultado esperado: La asignatura ha sido insertada.

Evaluación de la prueba: Satisfactoria.

Anexo 18

Anexo 16 Caso de prueba Eliminar asignatura (elaboración propia)

Caso de Prueba	
Código: HU2_P5	Historia de Usuario: gestionar asignatura.
Nombre: Eliminar asignatura.	
Descripción: Prueba de funcionalidad para la eliminación de una asignatura.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la eliminación de la asignatura deseada.	
Resultado esperado: La asignatura ha sido eliminada.	
Evaluación de la prueba: Satisfactoria.	

Anexo 19

Anexo 19 Caso de prueba Adicionar tema (elaboración propia)

Caso de Prueba	
Código: HU3_P6	Historia de Usuario: gestionar tema.
Nombre: Adicionar tema	
Descripción: Prueba de funcionalidad para la inserción de un tema.	

Anexos

Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.

Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando nombre, descripción y seleccionando la asignatura a la que pertenecerá.

Resultado esperado: El tema ha sido insertado.

Evaluación de la prueba: Satisfactoria.

Anexo 20

Anexo 20 Caso de prueba Eliminar tema (elaboración propia)

Caso de Prueba	
Código: HU3_P7	Historia de Usuario: gestionar tema.
Nombre: Eliminar tema.	
Descripción: Prueba de funcionalidad para la eliminación de un tema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la eliminación del tema deseado.	
Resultado esperado: El tema ha sido eliminado.	
Evaluación de la prueba: Satisfactoria.	

Anexo 21

Anexo 21 Caso de prueba Adicionar pregunta (elaboración propia)

Caso de Prueba	
Código: HU4_P8	Historia de Usuario: gestionar pregunta.
Nombre: Adicionar pregunta.	

Anexos

Descripción: Prueba de funcionalidad para la inserción de una pregunta.
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.
Pasos de ejecución: El usuario procede a la inserción de los datos necesarios especificando la pregunta deseada, selecciona el tema al que pertenece, define la dificultad mínima y máxima que poseerá inicialmente, escribe la respuesta correcta y 3 posibles falsas en caso de que no sea marcada como pregunta principal.
Resultado esperado: La pregunta ha sido insertada.
Evaluación de la prueba: Satisfactoria

Anexo 22

Anexo 22 Caso de prueba Eliminar pregunta (elaboración propia)

Código: HU4_P9	Historia de Usuario: gestionar pregunta.
Nombre: Eliminar pregunta.	
Descripción: Prueba de funcionalidad para la eliminación de una pregunta.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Pasos de ejecución: El usuario procede a la eliminación de la pregunta deseada.	
Resultado esperado: La pregunta ha sido eliminada.	
Evaluación de la prueba: Satisfactoria	

Anexo 23

Anexo 23 Caso de prueba: Cambio de rol a un usuario (elaboración propia)

Código: HU8_P12	Historia de Usuario: gestión de administración y funcionamiento.
Nombre: Cambio de rol a un usuario.	
Descripción: Prueba de funcionalidad para cambiarle el rol a un usuario.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema como administrador general. Debe existir un usuario al que se le va a cambiar el rol.	
Pasos de ejecución: El usuario con privilegios de administrador general, selecciona el usuario que va a editar y presiona editar, luego procede a seleccionar el rol deseado de la lista.	
Resultado esperado: El usuario es cambiado de rol.	
Evaluación de la prueba: Satisfactoria	

Anexo 24

Anexo 24 Caso de prueba: Iniciar juego online (elaboración propia)

Caso de Prueba	
Código: HU09_P13	Historia de Usuario: juego online.
Nombre: Juego online.	
Descripción: El usuario procede a iniciar un juego online junto a otros competidores.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema. Debe estar creada una competencia pero no iniciada.	

Pasos de ejecución: El usuario selecciona la opción iniciar juego online, de la lista de competencias disponibles.

Resultado esperado: Se inicia la competencia.

Evaluación de la prueba: Satisfactoria

Anexo 25

Anexo 25 Caso de prueba: Responder pregunta correctamente (elaboración propia)

Caso de Prueba	
Código: HU09_P14	Historia de Usuario: juego online.
Nombre: Responder pregunta correctamente.	
Descripción: El usuario responderá una pregunta de la competencia.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema. El usuario debe estar participando en una competencia.	
Pasos de ejecución: El usuario selecciona la respuesta correcta a la pregunta mostrada.	
Resultado esperado: Incrementa su puntuación y pasa a la siguiente pregunta.	
Evaluación de la prueba: Satisfactoria	

Anexo 26

Anexo 26 Caso de prueba: Responder pregunta incorrectamente (elaboración propia)

Caso de Prueba	
Código: HU09_P15	Historia de Usuario: juego online.
Nombre: Responder pregunta incorrectamente.	
Descripción: El usuario responderá una pregunta de la competencia.	

Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema. El usuario debe estar participando en una competencia.
Pasos de ejecución: El usuario selecciona la respuesta incorrecta a la pregunta mostrada.
Resultado esperado: Se decrementa su puntuación y pasa a la siguiente pregunta.
Evaluación de la prueba: Satisfactoria

Anexo 27

Anexo 27 Caso de prueba: Visualizar ranking (elaboración propia)

Caso de Prueba	
Código: HU12_P17	Historia de Usuario: sistema de ranking y puntuación.
Nombre: Visualizar ranking.	
Descripción: Un usuario podrá ver el ranking con los usuarios mejores puntuados.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema.	
Pasos de ejecución: El usuario procede a presionar el botón de Ranking.	
Resultado esperado: Se muestra una lista con el nombre y los puntos de los usuarios, ordenados descendientemente.	
Evaluación de la prueba: Satisfactoria	

Anexo: Acta de liberación

Anexo 28



FACULTAD # 3
CENTRO DE GOBIERNO ELECTRÓNICO




Acta de Liberación Interna de Productos Software

Fecha de emisión del acta: 21/06/2016

Emitida a favor de: Tesis "Herramienta Web para la Gestión del Proyecto Extensionista QQSI."


Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
App: Herramienta Web para la Gestión del Proyecto Extensionista QQSI.	1.0	0	2	Evaluación dinámica Pruebas de Funcionalidad	21/06/2016


Ing. Yordanis Garcia Leiva
Asesor de Calidad CEGEL


Michel Sed Araoz


Osvaldo Cintra Garcia
Autores



Jorien Machado Suastegui
Responsable de la Liberación



1

Anexo: Actas de aceptación

Anexo 29


	Vicedecanato de Extensión y Residencia Facultad 3 <i>Universidad de las Ciencias Informáticas</i>
---	---

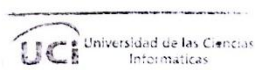

La Habana 15 de Junio de 2016

A tribunal de ejercicio de culminación de estudios:

Carta de aceptación de solución informática para el proyecto extensionista "Quién quiere ser ingeniero" (QQSI) del Departamento Técnicas de Programación de la Facultad 3.

Por este medio hacemos constar que los resultados del trabajo de diploma titulado "Herramienta web para la gestión del proyecto extensionista QQSI" de los estudiantes: Michel Sed Araóz y Osvaldo Cintra García satisfacen las expectativas del Vicedecanato de Extensión y Residencia. La propuesta cuenta con las funcionalidades requeridas hasta esta fase de la investigación, es de gran utilidad para las ejecuciones del proyecto extensionista y por ende contribuye a elevar la calidad de las actividades que se realizan en la Facultad 3 y en la comunidad. Entre los principales aportes de esta nueva versión sobresalen; el refinamiento de elementos que contribuyen a la disponibilidad de la aplicación, la gestión de preguntas online, así como la posibilidad de establecer ranking de puntuaciones para los usuarios.

Entregan:	Reciben:
<hr/> <i>Estudiantes Osvaldo Cintra García</i> <i>Michel Sed Araóz</i>	 <hr/> <i>Eliober Cleger Despaigne</i> Vicedecano de Extensión y Residencia Facultad 3


Decanato Facultar
 <hr/> <i>Michel Álvarez Cancio</i> Coordinador del proyecto extensionista QQSI

Anexo 29: Acta de aceptación #1

Anexo 30



Vicedecanato de Extensión y Residencia Facultad 3
Universidad de las Ciencias Informáticas

La Habana 15 de Junio de 2016

A Tribunal de Ejercicio de culminación de estudios:

Carta de aceptación de solución informática para el proyecto extensionista "Quién quiere ser ingeniero" (QQSI) del Departamento Técnicas de Programación de la Facultad 3.

Por este medio hacemos constar que los resultados del trabajo de diploma titulado "*Herramienta web para la gestión del proyecto extensionista QQSI*" de los estudiantes: Michel Sed Araóz y Osvaldo Cintra García cumplen con las funcionalidades requeridas para esta fase de la investigación y que por consiguiente es aceptada para su aplicación por la **Cátedra de Cultura Científica "Dra. Rosa Elena Simeón Negrín"**. La herramienta desarrollada ha sido gran utilidad para la realización de actividades coordinadas por la Cátedra, entre ellas: "*Cerebros en acción*" y en los festivales de la Ciencia, organizados anualmente por el CITMA y que se enfocan en el trabajo de promoción científica con niños entre 6 y 15 años.


Por todo lo anterior y para que conste la aceptación de los resultados de las pruebas y los entregables especificados, se extiende la presente acta, rubricada por:

Representante de las partes:

Entregan:

*Estudiantes Osvaldo Cintra García
Michel Sed Araóz*

Reciben:



**Dr. Jorge Gulín González
Presidente de la Cátedra de
Cultura Científica "Dra. Rosa
Elena Simeón Negrín"**

**Ing. Michel Álvarez Cancio
Coordinador del proyecto
extensionista QQSI**

Anexo 30: Acta de aceptación #2

Glosario

- **BSON**: Siglas en inglés para Binary JSON.
- **CouchDB**: Gestor de base de datos no relacional de código abierto.
- **CPU**: Siglas en inglés para Central Processor Unit.
- **DOM**: Siglas en inglés para Document Object Model. Es una interfaz de programación de aplicaciones (API) para documentos HTML.
- **HTML**: Siglas en inglés para HyperText Markup Language..
- **IDE**: Siglas en inglés para Integrated Development Environment.
- **JSON**: Siglas en inglés para JavaScript Object Notation.
- **Juego en línea**: La definición de juego en línea es la de un juego digital en el que es necesaria una conexión de red activa para poder jugar. Esto incluye no sólo juegos en Internet, sino también los que se juegan en línea a través de consolas o por teléfono móvil.
- **MPM**: Siglas en inglés para Multi-Processing Modules.
- **Online multiusuario**: Se le denomina así a la modalidad de competencia cuando se juega con más de un usuario en línea (Álvarez Cancio, y otros, 2015).
- **RAM**: Siglas en inglés para Random Access Memory.
- **RDBMS**: Siglas en inglés para Relational Database Management System.
- **WWW**: Siglas en inglés para World Wide Web.