

# UACM

Universidad Autónoma  
de la Ciudad de México

---

*Nada humano me es ajeno*

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA DE SOFTWARE

**“Desarrollo de un prototipo para la creación, aplicación y  
evaluación de cuestionarios para la UACM”**

TRABAJO RECEPCIONAL  
PARA OBTENER EL TÍTULO DE LICENCIADO EN  
INGENIERA DE SOFTWARE

PRESENTAN

**ALBERTO CARLOS PÉREZ CADENA**

Directora del trabajo recepcional

**M. en C. Araceli Liliana Reyes Cabello**

Ciudad de México, Junio de 2017.

## SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



## UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

### RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

#### DERECHOS RESERVADOS ©

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

---

# Agradecimientos

Me gustaría agradecer a todas aquellas personas que directa o indirectamente hicieron posible este trabajo.

En primera instancia a mi madre Catalina Cadena por su gran paciencia a lo largo de toda mi travesía y a mis hermanos por quererme por sobre todas las cosas.

A mi querida Universidad que creyó en mí y abrió sus puertas para que yo pudiera crecer tanto en lo académico como en lo personal.

A todos mis profesores de la academia de informática por impulsarme, sacar lo mejor de mí y darme las herramientas para mi profesión. Además quiero expresar mi agradecimiento en particular a los lectores de esta tesis: la Dra. Silvia Alejandra Andrade Rodríguez, la Mtra. Catalina Trevilla Román, la Mtra. Diana Aurora Cruz Hernández y el Mtro. José Luis Quiroz Fabián, a quienes considero mis apreciables maestros y que cada uno contribuyó de manera importante tanto en la elaboración de este trabajo (leyendo, opinando, corrigiendo, teniéndome paciencia, dando ánimo), como en mi formación académica.

Mi agradecimiento especial para la directora de este trabajo la Maestra Araceli Liliana Reyes Cabello. Te agradezco el haberme facilitado un espacio de confianza y libertad esenciales para la realización de este trabajo. Agradezco también tu tiempo, tu esfuerzo, tu orientación y tus consejos, para guiar mis ideas. Pero más allá de tu dirección en este proyecto te agradezco también porque fuiste mi guía como mentora y esto ha sido un invaluable aporte para mí.

Un agradecimiento sincero a la Dra. Silvia Alejandra Andrade Rodríguez la cual siempre quiso sacar lo mejor de mí. Gracias por toda su comprensión, disposición, paciencia y el apoyo incondicional que me brindó.

También quiero destacar el apoyo recibido por parte de Octavio León López, del área de desarrollo de software del plantel San Lorenzo Tezonco, por el apoyo para este trabajo pues sin su colaboración no habría sido posible la realización del mismo.

A mis compañeros de clase y amigos por acompañarme en esta mi odisea, en especial a Angélica Villagómez. Así mismo a mi gran amigo Fernando Estrada Sandoval, de quien siempre pude contar con su ayuda y tuvo tiempo para escucharme y me animaste a darle el rumbo que hoy tomo este trabajo.

Son muchas las personas a las que agradezco su amistad, consejos, apoyo, ánimo y compañía a lo largo de mi vida profesional. Algunas están aquí conmigo y otras entre mis recuerdos y mi corazón, sin importar en donde estén quiero decirles gracias por formar parte de mí y por todo lo que me han brindado.

Por último agradezco el apoyo económico otorgado por la Universidad Autónoma de la Ciudad de México (UACM) para la impresión y empastado de este trabajo de tesis.

Agradecimiento al apoyo del convenio SECITI/061/2016 de colaboración con la Secretaría de Ciencia y Tecnológica e Innovación (SECITI) de la ciudad de México y el centro de investigación y educación en transporte y Movilidad (CITMA), S.C., a través del proyecto “Sistema de Monitoreo y análisis para la toma de decisiones en materia de congestión y movilidad para la Ciudad de México”.

---

# Contenido

<b>Agradecimientos</b>	<b>2</b>
<b>Introducción</b>	<b>9</b>
<b>I Problemática</b>	<b>12</b>
<b>1. Introducción</b>	<b>13</b>
1.1. Identificación del problema . . . . .	13
1.2. Contexto . . . . .	15
1.3. Justificación . . . . .	20
1.4. Objetivos . . . . .	20
1.4.1. Objetivo general . . . . .	20
1.4.2. Objetivo específicos . . . . .	21
1.5. Estado del arte . . . . .	21
1.6. Alcance y limitaciones . . . . .	22
<b>II Marco Teórico</b>	<b>24</b>
<b>2. El desarrollo de software: modelos y metodologías</b>	<b>25</b>
2.1. Ciclos de vida . . . . .	25

---

2.2.	Construcción de prototipos . . . . .	26
2.3.	Tendencias y metodologías para la construcción de prototipos . . . . .	30
2.4.	Artefactos de software . . . . .	35
2.5.	Acerca de los requisitos . . . . .	39
2.6.	Modelo entidad relación y la importancia de la información . . . . .	40
2.7.	Documento de métrica para estimación del proyecto . . . . .	41
2.8.	Aplicaciones web y frameworks . . . . .	42
 <b>III Marco metodológico</b>		<b>43</b>
 <b>3. Ingeniería de Requisitos y Análisis de Software</b>		<b>44</b>
3.1.	Ingeniería de Requerimientos . . . . .	45
3.2.	Características del sistema . . . . .	46
3.2.1.	Historias de usuario . . . . .	49
3.2.2.	Casos de uso . . . . .	49
3.2.2.1.	Definición de actores . . . . .	49
3.2.2.2.	Caso de uso: Crear cuestionario . . . . .	52
3.2.2.3.	Caso de uso: Editar un cuestionario . . . . .	53
3.2.2.4.	Caso de uso: Resolver un cuestionario . . . . .	54
3.3.	Métrica puntos de función lite . . . . .	55
3.4.	Estimación . . . . .	56
3.4.1.	Tipo de conteo . . . . .	56
3.4.2.	Delimitación de los límites / frontera de la aplicación . . . . .	56
3.4.3.	Identificación de las funciones de datos . . . . .	57
3.4.3.1.	Identificación de ILFs . . . . .	57
3.4.3.2.	Identificación de ELFs . . . . .	58

---

3.4.4.	Identificación de las funciones transaccionales . . . . .	58
3.4.4.1.	Determinación de los procesos elementales . . . . .	59
3.4.4.2.	Determinación de Entradas Externas (EI) . . . . .	59
3.4.4.3.	Determinación de EOs . . . . .	60
3.4.4.4.	Determinación de EQs . . . . .	60
3.4.5.	Valor final de los PF para proyectos de desarrollo . . . . .	61
3.4.6.	Estimación del esfuerzo . . . . .	61
3.4.7.	Estimación de la duración . . . . .	62
3.4.8.	Estimación de cantidad de personal . . . . .	63
3.4.9.	Estimación del costo . . . . .	63
3.4.10.	Estimación de productividad y velocidad de entrega . . . . .	64
3.4.11.	Estimaciones por fase . . . . .	65
<b>4.</b>	<b>Modelado de datos</b>	<b>66</b>
4.1.	Bases de datos relacionales . . . . .	66
4.1.1.	Diseño de la base datos . . . . .	67
4.1.2.	Entidades y relaciones . . . . .	68
<b>5.</b>	<b>Framework Laravel</b>	<b>77</b>
5.1.	Introducción . . . . .	77
5.2.	Objetivos de los framework web . . . . .	78
5.3.	Características de un framework web . . . . .	79
5.4.	Diseño . . . . .	79
5.5.	Arquitectura . . . . .	80
5.6.	Contexto tecnológico . . . . .	81
5.6.1.	Composer . . . . .	81

---

5.6.2.	Artisan . . . . .	83
5.6.3.	Migraciones . . . . .	84
5.6.4.	Eloquent . . . . .	87
5.6.5.	Seeders y el componente faker . . . . .	88
5.7.	Presentación web . . . . .	89
5.7.1.	Bower . . . . .	90
5.7.2.	Bootstrap . . . . .	91
5.7.3.	Blade . . . . .	91
5.7.4.	Otros componentes . . . . .	94
5.8.	Interacción en la aplicación . . . . .	95
5.8.1.	Rutas . . . . .	95
5.8.2.	Controladores . . . . .	97
<b>6.</b>	<b>Recorrido por el Sistema</b>	<b>102</b>
6.1.	Funcionalidades del sistema . . . . .	102
6.1.1.	Funcionalidades para el Administrador . . . . .	103
6.1.1.1.	Creación de nuevo cuestionario . . . . .	103
6.1.1.2.	Inserción de temas . . . . .	105
6.1.1.3.	Otras funcionalidades . . . . .	105
6.1.2.	Funcionalidades para el Editor . . . . .	106
6.1.2.1.	Editar un cuestionario . . . . .	107
6.1.2.2.	Inserción de nueva pregunta . . . . .	109
6.1.2.3.	Inserción de una pregunta de baúl . . . . .	114
6.1.2.4.	Aprobación de un cuestionario . . . . .	115
6.1.3.	Funcionalidades para el Colaborador . . . . .	116
6.1.3.1.	Agregar una observación a un cuestionario . . . . .	117

---

6.1.4. Funcionalidades para el participante . . . . .	119
6.1.4.1. Resolver un cuestionario . . . . .	121
6.1.4.2. Ver los resultados de la aplicación de un cuestionario . . .	122
<b>7. Conclusiones</b>	<b>126</b>
7.1. Trabajos futuros y perspectivas del producto . . . . .	130
<b>Anexos</b>	<b>132</b>
<b>A. Requisitos y casos de uso</b>	<b>133</b>
<b>B. Modelo entidad relación</b>	<b>171</b>
<b>C. Diccionario de datos</b>	<b>173</b>
<b>Bibliografía</b>	<b>180</b>

---

# Introducción

Antes de empezar a diseñar e implementar un software, es necesario saber que es lo que se quiere construir, las necesidades y problemas que se deben cumplir. Primero se parte de una visión general, y poco a poco se van descubriendo, a veces paralelo al desarrollo, los detalles. Esto parece razonable y puede sonar sencillo, pero conlleva retos que solo la ingeniería de software (IS) puede superar. La IS es una disciplina relativamente nueva y puede verse como la aplicación práctica del conocimiento científico al diseño y construcción de programas computacionales, al igual que una ciencia, evoluciona, crece, incorpora nuevas herramientas, metodologías y conceptos, al mismo tiempo que se enfrenta a varios desafíos. Los desafíos de la IS van desde la gestión del proyecto hasta el mantenimiento del producto resultante, los tiempos de entrega de los sistemas, la continua evolución hacia sistemas más complejos, cambios en los procesos de negocios, por enumerar algunos. Así mismo no existe, un plan detallado y riguroso preestablecido para desarrollar sistemas de software, sin embargo la IS propone fundamentos, modelos y herramientas para la obtención de resultados más eficientes.

Este trabajo tiene la finalidad de presentar un prototipo de software en línea para el área de desarrollo de software de la Universidad Autónoma de la Ciudad de México (UACM), ya que esta área cuenta actualmente con dos sistemas para poder llevar a cabo la aplicación de evaluaciones diagnósticas a los estudiantes de nuevo ingreso. La principal desventaja es que para un mismo proceso se tiene dos sistemas descentralizados, donde la aplicación para crear y aplicar evaluaciones tiene el inconveniente de instalarse de manera local, lo que implica costos en tiempo de instalación. A raíz de esto el área de desarrollo de software propone la creación de un sistema web integral de evaluación, que proporcione una interfaz para la creación, edición y resolución de cuestionarios en línea, que no solo sirva para la población de nuevo ingreso, sino que pueda ser utilizado en cualquier ámbito en donde se desee realizar un proceso de evaluación. Es aquí donde se presenta la oportunidad para usar elementos de ingeniería de software y enfocarlos hacia dicha problemática, ya que con la aplicación de dicha disciplina se pueden reducir riesgos de fallos en un sistema en

desarrollo e incrementar la posibilidad de la entrega del producto en el tiempo estimado, con calidad y el con cumplimiento de parámetros requeridos.

Este trabajo se centra en el desarrollo de un prototipo de un sistema integral de cuestionarios en línea para la UACM que permitirá dentro de sus funcionalidades tanto la creación, como la edición y resolución de los cuestionarios con preguntas de opción múltiple. Cabe señalar que una de las primeras tareas para el desarrollo del sistema fue el levantamiento de requisitos, en el cual se trabajó con el encargado del desarrollo de los sistemas de evaluación que se mencionaron anteriormente; de lo cual se obtuvo que el sistema incluyera preguntas de opción múltiple y preguntas abiertas. Sin embargo, después de realizar las métricas de estimación de tiempo y esfuerzo se determinó desarrollar un primer prototipo integral que sólo incluya las preguntas de opción múltiple. No obstante el diseño de la base de datos contempla ambos tipos de preguntas, así como los datos que se requieren almacenar para todas las funcionalidades del sistema que se recabaron en los requerimientos.

El impacto que este sistema integral tendría, será la optimización de tiempos y horas hombre, ya que al ser en línea no se tendrá que estar instalando cada vez que se necesite o se realicen cambios en el sistema, pero además permitiría tener todos los resultados de las evaluaciones concentrados en una base de datos y de forma estructurada (ya que actualmente no es de esta manera), lo que facilitará el análisis estadístico de estos datos e, incluso, aplicar técnicas de minería de datos para conocer el avance o rendimiento del conocimientos de los estudiantes.

El procedimiento seguido para el desarrollo del prototipo se presenta en este documento conformado por seis capítulos. El primero da una explicación breve de los sistemas de evaluación actuales en la UACM con el fin de definir el contexto del problema actual. Además se enuncian los objetivos, alcance y limitaciones del sistema. También se presenta una tecnología que se ajusta en solución a algunos aspectos de la problemática; sin embargo no es utilizada para solvencia puesto que lo requerido por el área de desarrollo de la Universidad es un software “a la medida”. En el segundo capítulo se presenta el modelo de prototipo y dos metodologías de desarrollo que nos brindan un marco teórico para enfrentar la problemática presente mediante el uso coherente de conceptos dentro de la ingeniería de software. Consecuentemente se aborda, en el tercer capítulo, el análisis de requisitos y el uso de una métrica para estimar el tamaño del producto y de esta forma tener una idea del esfuerzo requerido para concluir un proyecto de este tipo. En el cuarto capítulo se habla del modelado de la base datos utilizando una técnica de moldeamiento como guía útil o referencia directa para asegurar la calidad de los datos, también dentro

de este mismo capítulo, se especifican las principales entidades que conforman la capa de persistencia del sistema. El quinto capítulo plantea el marco de trabajo o framework Laravel que fue el eje central para el desarrollo del proyecto, el cual provee una estructura y una metodología especial de trabajo que facilitó la implementación del sistema, además de ayudar a tener una mejor organización en el código. Finalmente, en el capítulo sexto se describen aspectos de la funcionalidad en la aplicación para cada uno de los principales roles de usuario con los que interactúa el sistema, es decir, se muestra las partes de la funcionalidad deseada en el prototipo enfocado a la experiencia que tendrá un usuario para que este pueda comprobar que el prototipo permite realizar las tareas que desean llevar a cabo.

---

**Parte I**

**Problemática**

---

# Capítulo 1

## Introducción

En este capítulo se presenta los antecedentes de este trabajo de tesis, haciendo una introducción al problema que se pretende resolver, así como de los objetivos, alcances y limitaciones.

### 1.1. Identificación del problema

Actualmente el área de desarrollo de software de la Universidad Autónoma de la Ciudad de México (UACM) cuenta con dos sistemas (sin nombre) para poder llevar a cabo la aplicación de evaluaciones diagnósticas a los estudiantes de nuevo ingreso. El primero de estos sistemas consiste en una aplicación de escritorio (figura 1.1) que permite crear y aplicar evaluaciones; esta aplicación tiene el inconveniente de instalarse de manera local, lo que significa invertir tiempo en la instalación cada vez que se tiene que hacer el proceso de evaluación a los estudiantes, además si surge un cambio de última hora se hace la modificación pertinente y se repite el proceso de instalación en cada computadora. En segunda instancia, se tiene una página web donde los profesores califican las preguntas abiertas y llenan las actas cualitativas (figura 1.2), cabe decir que las evaluaciones cuentan con dos tipos de pregunta las cuales son de opción múltiple y abiertas, las preguntas de opción múltiple son evaluadas de forma automatizada. La principal desventaja es que para un mismo poseso se tiene dos sistemas descentralizados; uno de ellos ni siquiera garantiza la integridad de la base datos, con lo que no podemos estar seguros de la calidad, coherencia y exactitud de la información contenida. Por otro lado, el hecho de que el sistema de creación y aplicación de evaluaciones sea de escritorio, limita la posibilidad de que sea utilizado para evaluaciones de otra materias, ya que cada vez que se quiere aplicar una

nueva evaluación tendría que volverse a instalar en los equipos correspondientes.

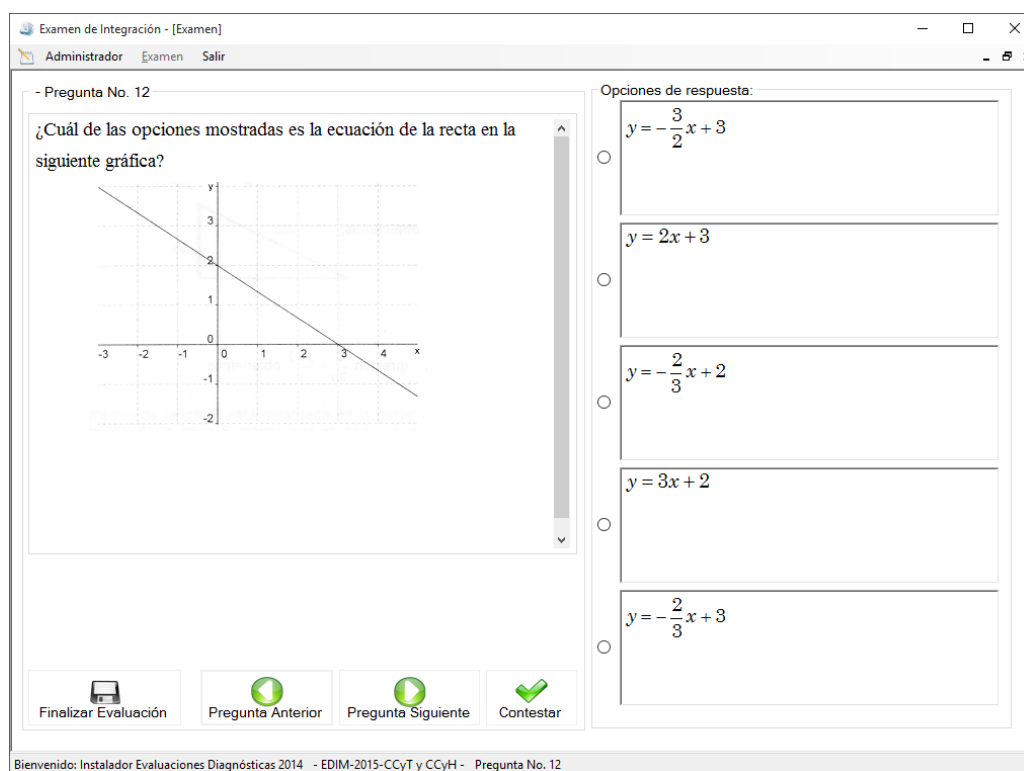


Figura 1.1: Aplicación de escritorio para crear y editar evaluaciones.

Además la recuperación de los datos para calificar, conlleva otro problema ya que esta recopilación se hace por medio de un archivo de configuración; este archivo contiene la instrucción para habilitar la escritura sobre la base de datos del servidor del plantel y en caso de no poder escribir en dicha base por inconvenientes no controlados (fallo con el servidor, error con el sistema de gestión de bases de datos, fallas eléctricas, etc), el sistema guarda de manera local las respuestas. La recolección de las respuestas en modo local se hace máquina por máquina, mediante un programa llamado “respuestas local”, con este se verifica por matrícula y examen, las respuestas con respecto a la base de datos. Ahora bien, se podía dar el caso de que el sistema guardara algunas preguntas en el servidor y el restante no, si ese fuera el caso se compara el número de respuestas por cada pregunta y si estas eran mayor que el número de respuestas locales entonces se omitía la matrícula para esa aplicación de examen, en caso contrario se sobrescribe e insertan las respuestas locales.

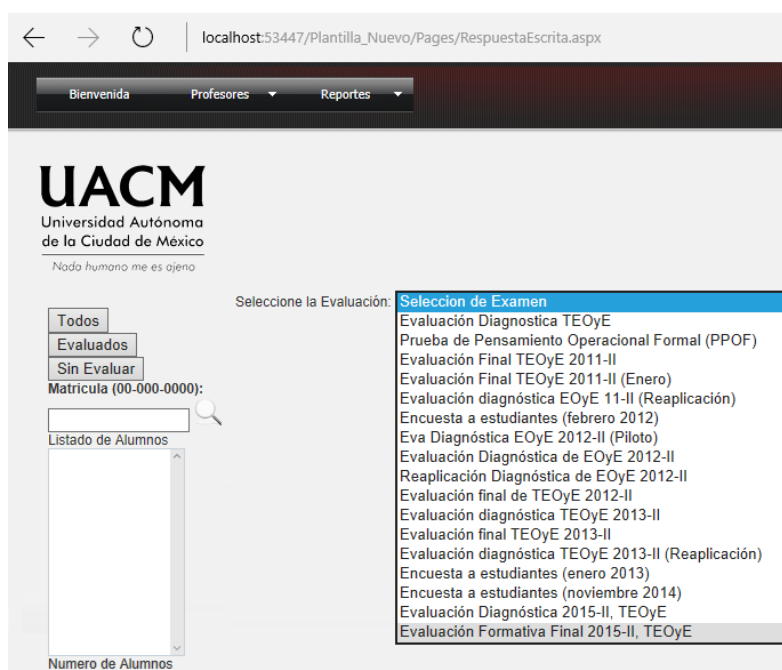


Figura 1.2: Pagina web para evaluación de examen de integración.

## 1.2. Contexto

A continuación se describe más detalladamente los dos sistemas que se tienen para crear, aplicar y calificar evaluaciones, que actualmente tiene el área de desarrollo de software de la UACM, esta descripción estará mostrando actores, tipos de pregunta, y características generales que nos permitirá tener un contexto mas amplio del problema.

Existen tres actores identificados: estudiante, profesor y administrador, que utilizan el sistema. En el caso de los estudiantes, solo pueden acceder al sistema para resolver el examen de evaluación, estos acceden por medio de su matrícula y su fecha de nacimiento, los datos son comparados con los existentes en la base de datos de la Universidad y solo así podrán tener acceso al cuestionario (figura 1.3).

Una vez con acceso, el sistema permite elegir de entre las opciones disponibles (figura 1.4) una evaluación. Estas evaluaciones contienen preguntas que internamente se componen de un número consecutivo, una descripción de la pregunta, el tipo de pregunta y las opciones de respuesta, si es el caso. En este sistema solo existen dos tipos de preguntas; de opción múltiple y preguntas con respuesta escrita o abiertas. Por un lado las preguntas de opción múltiple (figura 1.1) siempre tienen como prefijo cinco respuestas opcionales y el sistema no permite quitar o agregar más opciones. Las respuestas en un cuestionario de opción múltiple pueden tener imágenes. Las respuestas están dentro de una función

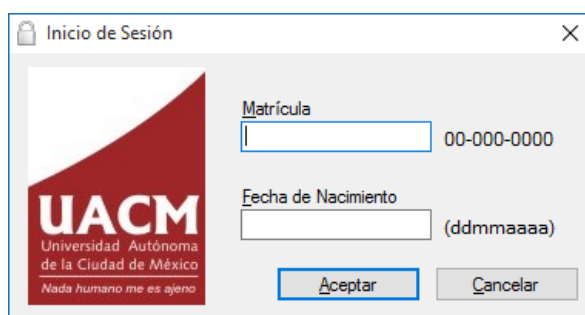


Figura 1.3: Aplicación de escritorio para cuestionarios.

de colocación aleatoria, es decir, la posición de la respuesta de opción múltiple cambia en cada cuestionario.

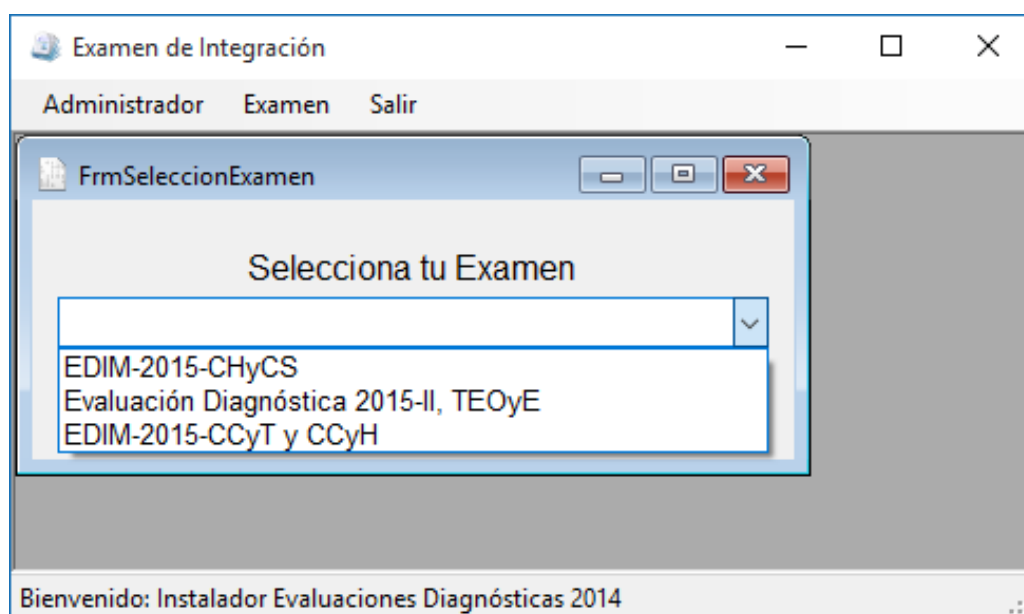


Figura 1.4: Selección de evaluaciones disponibles.

En segundo lugar se encuentra las preguntas de respuesta abierta (figura 1.5), estas preguntas contienen un texto con o sin imágenes del lado izquierdo y del lado derecho un espacio donde el estudiante puede ingresar su respuesta.

Este sistema permite “navegar” entre las respuestas para revisar todo el cuestionario o en caso de no saber la respuesta de la pregunta actual, se pueda pasar a otra pregunta para después responder la pregunta que se dejó sin contestar, es decir, admite que se puedan dejar preguntas sin contestar lanzando un mensaje de alerta (figura 1.6) que se están dejando preguntas sin responder y da la oportunidad de regresar al cuestionario para seguir contestando o también se puede dejar el cuestionario inconcluso.

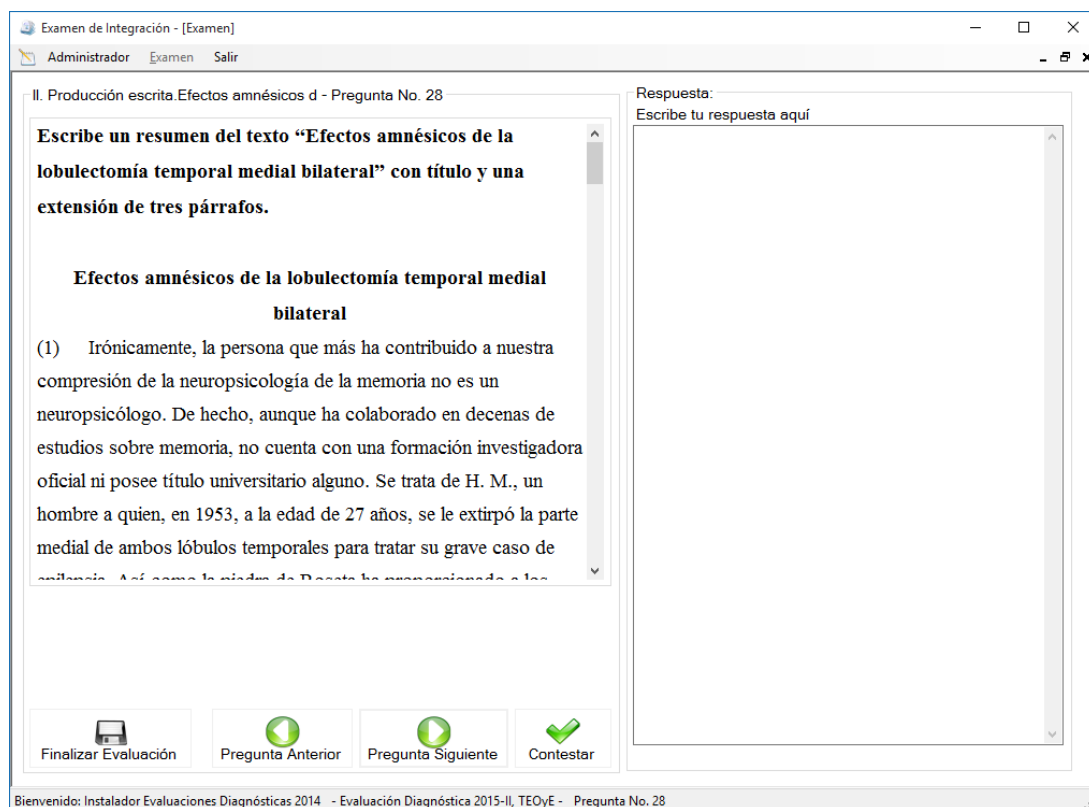


Figura 1.5: Preguntas con respuesta abierta.

Finalmente, una vez que el cuestionario fue contestado por un estudiante, el sistema califica de manera automática las preguntas de opción múltiple y las calificaciones se escriben en un archivo de Excel para su análisis. Si todas las preguntas son de opción múltiple, el sistema calcula el número de repuestas correctas y entrega el resultado. Si el cuestionario tiene preguntas abiertas, el sistema evalúa si se han contestado o se dejaron sin resolver, en caso de tener respuestas, deben ser evaluadas por un profesor, en caso contrario se entrega la calificación.

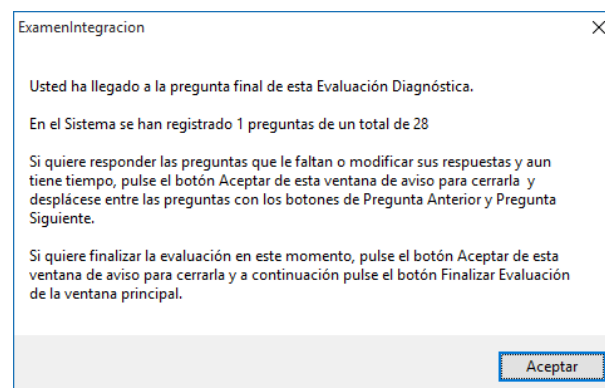


Figura 1.6: Alerta de finalización de cuestionario.

En el caso de los profesores, pueden acceder con dos diferentes roles uno es de administrador y otro es el de evaluador. Los profesores que acceden como administrador pueden realizar las siguientes tareas:

1. Acceso a todas las tablas de preguntas y respuestas.
2. Creación de cuestionario.
3. Calificar cuestionario.

Los profesores con rol de evaluador, solo pueden realizar la última tarea. Para crear los cuestionarios el sistema cuenta con un editor donde se puede insertar texto con el mismo formato del editor de Microsoft Word. Este requerimiento en el sistema surgió, por necesidades de los profesores. Cabe señalar que, desafortunadamente, este requerimiento exige que se guarde mucha información, haciendo que el tamaño las preguntas y respuestas sea excesivo, provocando bajo rendimiento en la consulta de todos los cuestionarios.

El evaluador califica los cuestionarios por medio de la página web <http://www.apps.uacm.edu.mx/integracion>. Las actividades que se pueden hacer en este sitio son: pueden calificar las respuestas abiertas (figura 1.7) y consultar el acta cualitativa o de resultados por estudiante o por evaluación. Al entrar a la página web, se muestra si existe algo por calificar, es decir, si existe una respuesta escrita de la aplicación de un examen, de ser así esta se evalúa y en caso contrario se presenta y se imprime el acta cualitativa o de resultados. Por otra parte un estudiante solo puede acceder a las actas cualitativas correspondiente a sus evaluaciones. Como dato adicional, se puede generar una calificación de un curso en el sitio web, esto es, se puede formar un acta cuantitativa con elementos cualitativos para un curso en particular (solo para calificación final, sin afectar la base de datos de certificación).

Finalmente, los cuestionarios se guardan en tablas individuales sin relación entre ellas, no existe una separación en entidades (figura 1.8). Lo que puede no garantizar la integridad referencial, así que si es necesario eliminar o actualizar un registro no se puede garantizar la congruencia en todos los registros relacionados dependientes. Al no existir una normalización de tablas esta base de datos tenderá a ser poco comprensible para otros desarrolladores y difícil para darle mantenimiento a futuro. Así mismo el análisis y procesamiento de los datos, con fines estadísticos, se vuelve complicado.

localhost:53447/Plantilla\_Nuevo/Pages/RespuestaEscrita.aspx

Bienvenida Profesores Reportes Salir

**UACM**  
 Universidad Autónoma de la Ciudad de México  
 Nada humano me es ajeno

Seleccione la Evaluación: Evaluación Formativa Final 2015-II, TEOyE

Nombre del alumno:  
 Respuesta  Calificar

Lista de Alumnos:  
 12-003-1222  
 15-001-0001  
 15-001-0063  
 15-001-0222  
 15-001-0267  
 15-001-0279  
 15-002-0021  
 15-002-0153  
 15-002-0171  
 15-002-0227  
 15-003-0004  
 15-003-0007  
 15-003-0031  
 15-003-0034  
 15-003-0046  
 15-003-0050  
 15-003-0053  
 15-003-0056  
 15-003-0108  
 15-003-0125  
 Numero de Alumnos=271

**Rúbrica para evaluar la producción escrita**

Propiedad textual	Indicadores				
Formato	El texto cumple con una extensión mínima para ser evaluado.	Si <input type="radio"/> No <input type="radio"/>			
	El texto tiene un título pertinente.	Si <input type="radio"/> No <input type="radio"/>			
Adecuación	Destacado	Acceptable	Suficiente	Insuficiente	
	Registro	Evita totalmente las marcas de oralidad o de código escrito informal.	Presenta sólo una marca de oralidad o de código escrito informal.	Presenta algunas marcas de oralidad o de código escrito informal.	En el texto se encuentran considerables marcas de oralidad o de código escrito informal.
	Vocabulario	Parafrasea correctamente, es decir, usa sus propias palabras y mantiene sólo conceptos o ideas clave del texto original.	Expresa con sus propias palabras la mayor parte de las ideas esenciales del texto original, pero también reproduce partes prescindibles de éste.	Expresa con sus propias palabras algunas ideas del texto original y reproduce otras textualmente.	Copia textualmente, o bien, usa sus propias palabras, pero tergiversa el texto original.
	Voz enunciativa	Mantiene una sola voz enunciativa formal en tercera persona.	Mantiene una sola voz enunciativa informal en primera persona.	Utiliza formas discursivas impersonales.	Mixcia distintas voces enunciativas.
Coherencia	Complejidad de las ideas	Presenta las ideas completas.	Presenta una o dos ideas que quedan incompletas o carecen de antecedente, comienzan con un nexo, sin antecedente de idea principal, o bien, como si responderan a una pregunta.	Presenta hasta tres ideas que quedan incompletas o carecen de antecedente, comienzan con un nexo, sin antecedente de idea principal, o bien, como si responderan a una pregunta.	Presenta cuatro o más ideas incompletas o carecen de antecedente, comienzan con un nexo, sin antecedente de idea principal, o bien, como si responderan a una pregunta.
	Orden de las ideas	Expone las ideas en un orden lógico.	Aunque expone las ideas con cierto orden, éste no resulta consistente.	Algunas de las ideas que presenta no se derivan de la información previa.	Las ideas se presentan de modo desordenado y sin lógica.
		Introduce, desarrolla y concluye.	Ubica al lector en el tema, lo desarrolla, pero no concluye.	No ubica al lector en el tema y directamente lo desarrolla.	El texto carece de una organización en introducción, desarrollo y conclusión.
Unidad temática del resumen	Elabora un discurso narrativo con la información esencial para comprender el texto original.	Ofrece datos y hechos del texto original resaltando varias de las ideas esenciales.	Rescata algunas ideas esenciales del texto original y otras no esenciales.	Elabora un texto donde predominan las ideas no esenciales o sus omisiones.	
Signos de puntuación	Utiliza correctamente los signos de puntuación.	Algunos signos de puntuación están mal empleados o ausentes.	La ausencia de signos de puntuación necesarios o los errores en su uso entorpecen la lectura del texto.	La ausencia de signos de puntuación necesarios o los errores en su uso dificultan la comprensión del texto.	
	Utiliza correctamente los signos de puntuación.	Algunos signos de puntuación están mal empleados o ausentes.	La ausencia de signos de puntuación necesarios o los errores en su uso entorpecen la lectura del texto.	La ausencia de signos de puntuación necesarios o los errores en su uso dificultan la comprensión del texto.	
Cohesión	Nexos y marcadores discursivos	Utiliza correctamente las preposiciones, conjunciones y marcadores discursivos.	Algunas preposiciones, conjunciones y marcadores discursivos están mal empleados o ausentes.	La ausencia de preposiciones, conjunciones y marcadores discursivos necesarios o los errores en su uso entorpecen la lectura del texto.	La ausencia de preposiciones, conjunciones y marcadores discursivos necesarios o los errores en su uso dificultan la comprensión del texto.
	Concordancias	No hay errores de concordancia.	Presenta algunos errores de concordancia.	La lectura del texto se entorpece por los errores de concordancia.	La comprensión del texto se dificulta por los errores de concordancia.
Ortografía	Acentúa correctamente	Si <input type="radio"/>	No <input type="radio"/>	No <input type="radio"/>	No <input type="radio"/>
	Empieza las grafías sin errores	Si <input type="radio"/>	No <input type="radio"/>	No <input type="radio"/>	No <input type="radio"/>
	Identifica el límite de las palabras	Si <input type="radio"/>	No <input type="radio"/>	No <input type="radio"/>	No <input type="radio"/>

Figura 1.7: Página web para evaluación de examen de integración.

ID	IDExamen	IDPreguntas	IDUsuario	Op1	Op2	Op3	Op4	Op5	Op6	Mc1	Mc2	Mc3	Mc4	Mc5	Mc6	RespuestaCorta	Respuesta	FechaEvaluacion
443	2	103	11-004-0017	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	40702
443	2	104	11-004-0017	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	40702
450	2	105	11-004-0017	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	40702
453	2	106	11-004-0017	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	40702
1657	3	56	11-011-0179	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2	40702
1658	3	57	11-011-0179	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	40702
1659	3	58	11-011-0179	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	40702
1660	3	61	11-011-0179	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		3	40702
1661	3	62	11-011-0179	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		4	40702
1662	3	64	11-011-0179	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1	40702
1663	3	65	11-011-0179	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		4	40702

Figura 1.8: Registro de los datos de un cuestionario.

### 1.3. Justificación

Las limitaciones de los sistemas actuales para editar y calificar evaluaciones expuestas en el punto anterior, crean una oportunidad para desarrollar un nuevo sistema, que solucione la problemática expuesta y facilite el proceso de edición, aplicación y evaluación de cuestionarios y además extenderlo a cualquier área en donde se necesite realizar una evaluación con estas mismas características, haciendo de una evaluación más puntual y flexible, con la disponibilidad de la aplicación a través de dispositivos que tengan un navegador web.

Con este trabajo se busca aportar un beneficio a la Universidad y al mismo tiempo favorecer, ampliar y aplicar conocimientos adquiridos durante la formación profesional en un sentido más real, motivado por el interés en lograr una mejor comprensión del papel que desempeña la ingeniería de software dentro del proceso de desarrollo de nuevos productos. Por tanto el presente proyecto pretenderá abarcar tanto la aplicación de elementos de ingeniería de software como el desarrollo del prototipo de software asociado.

### 1.4. Objetivos

#### 1.4.1. Objetivo general

Realizar el análisis y diseño de un sistema web para crear, aplicar y calificar cuestionarios en línea. Además construir e implementar un primer prototipo con funcionalidades acotadas, es decir, que no contendrá todas las características del sistema final, pero que incluirá elementos suficientes para permitir evaluar el sistema propuesto y determinar

aquellas características que deberán cambiar o añadirse.

### 1.4.2. Objetivo específicos

- Recopilar, analizar y verificar los requerimientos del sistema.
- Utilizar la métrica de software Puntos de Función Lite para la estimación del tamaño y costo del prototipo.
- Aplicar metodologías de desarrollo ágiles para la construcción del prototipo.
- Analizar y diseñar una base de datos relacional utilizando el método CASE para el almacenamiento de los datos en el sistema.
- Desarrollar la versión beta del sistema en línea de cuestionarios con preguntas de un tipo específico, utilizando el framework Laravel, utilizando el lenguaje de programación PHP.
- Aplicación del marco de trabajo de Laravel para el diseño del sistema.

## 1.5. Estado del arte

Por lo que se refiere a dar solución a la problemática planteada previamente se puede pensar inicialmente en dos alternativas.

La primera es el uso de herramientas o software ya desarrollado que proponen una solución genérica para ciertas problemáticas en común, para nuestro caso podemos considerar herramientas como:

- **Moodle.** Es una aplicación web del tipo ambiente educativo virtual para la creación de cursos de E-Learning, es de distribución libre y ayuda a los educadores a crear comunidades de aprendizaje en línea. Moodle cuenta con varias características que la hacen interesante, como módulos de tareas, foros, wiki, entre otras. No obstante en torno al problema que nos concierne Moodle en sus características tiene el módulo de encuesta y el módulo de cuestionario, el primero es solo un conjunto de preguntas que se aplican a los estudiantes sin que este tenga una calificación y el segundo permite realizar cuestionarios que un profesor puede evaluar, con preguntas de diferente tipo.

- **Formularios de Google.** Es una herramienta gratis a través de Internet que ofrece posibilidades de crear formularios personalizados para recopilar información, por ejemplo en encuestas, suscripciones, formar un test o cuestionarios para estudiantes, para introducir preguntas desde del tipo de casilla de verificación hasta en forma de texto (preguntas abiertas), entre otras características.
- **Socrative.** Es una herramienta que puede ser utilizada tanto en dispositivos móviles, así como para computadoras (laptop y de escritorio) de forma gratuita. Permite que un profesor puede subir diversas actividades (preguntas, quizzes, juegos,) y estas ser respondidas por los estudiantes. Es posible hacer cuestionarios pre-diseñados o plantear otros. Admite respuestas cortas. Además los estudiantes al responder un cuestionario podrán visualizar las respuestas generándose un informe con las calificaciones, que se envía por correo o puede consultarse online como hoja de cálculo de Google o descargarse en formato Excel. Las respuestas del estudiante se pueden proyectar de manera instantánea en el dispositivo del profesor y al terminar pueden comentar su experiencia, dejando retroalimentación sobre la actividad.

Si bien dentro del mundo educativo y los sistemas de información existen varias herramientas para la creación y aplicación de cuestionarios, sin embargo en la problemática presente en este proyecto, se necesita realizar un sistema que este basado en requisitos específicos de acuerdo a necesidades muy particulares del cliente, es decir, desarrollar un software a la medida. De este modo se propone la segunda opción que es la creación de un software con características especiales, que Moodle u otros software educativos no cumplen, como por ejemplo, generación automática de actas cualitativas, las cuales se entregan a un estudiante después de revisar los instrumento de evaluación, para que este observe su aprovechamiento escolar durante un semestre. Esta característica es propia del modelo de la UACM y los indicadores son propuestos por los profesores quienes son editores de las evaluaciones, por lo tanto el nuevo software de aplicación de cuestionarios, buscará ajustarse a este requisito específico y otros más.

## 1.6. Alcance y limitaciones

- En el prototipo se delimitará las funciones debido a que este será una versión beta del sistema.
- A pesar de ser una versión beta, será un sistema completo ya que incluirá todos los

requisitos para los cuestionarios de un solo tipo, es decir, solo cuestionarios con un solo tipo de pregunta: las de opción múltiple.

- En términos de tiempo tenemos la limitante de que el producto completo, es decir, la aplicación web terminada, llevaría un total aproximado, según la métrica aplicada, de quince meses, lo que hace replantear este proyecto a un solo prototipo para ajustarlo a un tiempo razonable de entrega para un proyecto de titulación de licenciatura.
- Las limitantes están en función de los resultados obtenidos por la métrica puntos de función lite, la cual nos da una aproximación en tamaño y costo del producto a construir y en donde se tiene dos importantes limitaciones:
  - Tiempo
  - Recursos humanos
- Aunado al punto anterior la métrica también arroja que el recurso humano para desarrollo de este proyecto de software, fuese de tres desarrollares, lo cual que este proyecto no pudo alcanzar, puesto que el autor de esta tesis es el único encargado de tomar los roles necesarios para la construcción del prototipo.

Por lo tanto, debido a las limitantes antes mencionadas, el alcance de este desarrollo de software será llevado a un prototipo que servirá como demostración de lo que el sistema a futuro estaría realizando.

---

## Parte II

### Marco Teórico

---

## Capítulo 2

# El desarrollo de software: modelos y metodologías

La ingeniería de software es una subdisciplina de las ciencias computacionales, su principal objetivo es el desarrollo de sistemas de software de calidad mediante modelos de desarrollo y metodologías. En este sentido nos enfocaremos en conceptos que giran en torno al modelo de prototipos el cual nos dará la pauta para considerar al modelo incremental como una base que nos proporcione el enfoque para la creación de sistemas de prototipos bajo la idea de pequeños incrementos. Posteriormente se hablará de dos metodologías de desarrollo de software en donde buscaremos artefactos que contribuyan al desarrollo de nuestro proyecto para obtener un proceso formalizado y de esta forma contar con los fundamentos para formular y desarrollar el prototipo propuesto para resolver, dentro de la UACM, la problemática relacionada con el proceso de exámenes que se aplica a los estudiantes de nuevo ingreso.

### 2.1. Ciclos de vida

Actualmente, el proceso de desarrollo de software es considerado una tarea compleja, prueba de ello son las diferentes metodologías y modelos que se han propuesto desde que el término de ingeniería de software comenzara a utilizarse a finales de los ochenta. En el área de desarrollo de software para lograr construir un sistema de información o un prototipo, como el que se desarrolló en este trabajo, regularmente se requiere de metodologías y modelos que constan de fases y artefactos que permiten aplicar métodos y técnicas para lograr una aplicación más eficiente. No obstante, el proceso de desarrollo

de software nos llevaría a analizar modelos tales como el modelo en cascada, modelo en espiral o el modelo incremental y metodologías tales como RUP<sup>1</sup> (metodología tradicional) o Scrum (metodología ágil). En este capítulo nos concentraremos en reflexionar propuestas y modelos relacionados con la construcción de prototipos, con la finalidad de concretar el desarrollo de software planteado para la UACM.

Para la ingeniería de software un modelo se puede ver como marcos de referencia que ayudan a la construcción de proyectos, los cuales tienen un ciclo de vida, es decir, fases, actividades y controles que ayudan a desarrollar un producto de software de manera eficiente y eficaz. Aunque no hay un único ciclo de vida, la elección de alguno estará en función de la complejidad del problema, el tiempo con el que disponemos para la entrega, el equipo de desarrollo, entre otros. Es decir, el tener varios ciclos de vida va implicar un sentido de adaptación, ya que como explica Serrano (1996, p. 41) “el orden y el tipo de actividades en cada fase pueden cambiar adaptándose a las necesidades del producto a realizar y a la propia estructura de la organización que lo desarrolla a partir de las posibilidades que ofrece la tecnología de software empleada”. De esta forma podemos darnos cuenta que el ciclo de vida que elijamos podemos ajustarlo de acuerdo con los recursos y necesidades del proyecto y de la organización. Con esto en mente se pretende buscar un modelo de ciclo de vida que de cabida a la construcción del prototipo y para ello se consideran dos importantes perspectivas, el modelo de prototipo rápido y el prototipado incremental. Aunque en la bibliografía es común encontrarse con clasificaciones de modelos de ciclo de vida muy detalladas pueden considerarse como variaciones del modelo de ciclo de vida en cascada o modelo lineal secuencial. Aún hoy se considera que todos los demás modelos no son más que modificaciones de este modelo básico. Sin embargo, para evitar situaciones que puedan crear mala interpretación, se prefiere tratarlos de forma diferenciada.

## 2.2. Construcción de prototipos

El sistema de cuestionarios en línea propuesto para el área de desarrollo de la UACM, se considera como un prototipo que modelará al producto final, permitiendo efectuar un test de usabilidad sobre determinadas funcionalidades, específicamente la creación, edición y evaluación de cuestionarios con preguntas de opción múltiple, esto debido a limitaciones en términos de un recurso humano único, no obstante debe evitarse pensar que la aplicación tendrá una perspectiva de “simulación”, sino que, por el contrario, se

---

<sup>1</sup>Proceso Unificado Racional

busca que este prototipo pueda evolucionar, con el suficiente refinamiento, hasta alcanzar un producto final, en un tiempo futuro.

Antes de continuar, se debe entender por prototipo como “un modelo parcial ejecutable de un sistema de software, lo definimos como parcial porque no es necesario que cubra todo el sistema sino aquellas partes o perspectivas que se pretenden analizar, este debe ser ejecutable para que la validación del sistema pueda hacerse a partir de la experimentación con el prototipo por parte de usuarios y analistas” (Serrano, 1996, p. 57).

En cuanto a prototipos se refiere, existe un modelo de ciclo de vida que contempla dos tipos básicos en el uso de prototipos, este modelo es llamado **Modelo incremental**. La idea detrás del modelo incremental es “tomar elementos del modelo lineal secuencial (cascada) con la filosofía de construcción de prototipos, pero aplicados repetidamente, estas secuencias lineales son de forma escalonada (figura 2.1) mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un «incremento» del software” (Pressman y Troya, 1988, p. 23), es decir, la funcionalidad del sistema es producida y entregada en pequeñas partes. Entonces se puede decir que este modelo es iterativo por naturaleza.

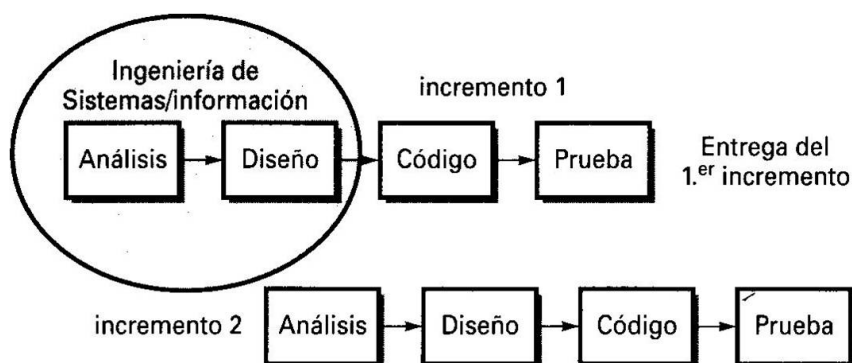


Figura 2.1: Modelo incremental propuesto por Pressman y Troya (1988, p. 63)

Como se mencionó existen dos tipos básicos o clasificaciones que se contemplan en el modelo incremental. En primer lugar, está un modelo basado en **prototipo rápido** y en segundo, el modelo de **prototipado incremental**. Estos dos enfoques exponen al usuario prototipos que le acercan al sistema final. Sin embargo, cabe aclarar sus diferencias con respecto al uso que le podemos dar para completar nuestro objetivo.

El modelo de prototipo rápido, es conocido también como modelo basado en prototipos desechables<sup>2</sup>, es aplicable a proyectos donde sólo se cuente con requisitos generales o un conjunto de objetivos muy genéricos, es decir, aborda el problema de la inestabilidad de los

<sup>2</sup>Desechable está ligada más a limitaciones de la tecnología empleada que a un interés real del usuario en “tirar” el trabajo realizado.

requisitos, generando un prototipo lo antes posible que sirva de base al mejor conocimiento de los requisitos de usuario. Este prototipo es desechado cuando usuarios y desarrolladores acuerdan un documento de requisitos de usuario. Se considera este concepto útil, puesto que inicialmente se cuenta con objetivos y requerimientos que no poseen un refinamiento total y con esto, podemos presentar un modelo preliminar con aspectos del sistema que serán visibles para el cliente y usuario. Por ejemplo, pantallas de entrada y salida de datos, formatos de informes, etcétera.

El modelo basado en prototipado incremental, “se basa en la generación de varios modelos parciales ejecutables del sistema antes de proceder a la implementación (durante la especificación y durante el diseño) con el fin de evaluar sus características y poder obtener al final el sistema implementado. En estos prototipos no se pretende implementar partes del sistema final sino razonar sobre modelos ejecutables no implementados todavía” (Serrano, 1996, p. 60). Con esta idea en mente podemos hacer, para el sistema de cuestionarios, un refinamiento iterativo de las especificaciones del sistema aumentando así el objetivo y/o el alcance. A pesar de lo dicho para el modelo basado en prototipos desechables, en donde desechamos un prototipo cuando se acuerda un documento de requisitos, en la filosofía de modelo iterativo incremental, no se «tira» nada. “Los modelos realizados siguen siendo empleados en el siguiente prototipo o se convierten en un modelo con un nivel de detalle mayor (o incluso código)” (Serrano, 1996, p. 63).

Derivado de lo anterior, se puede observar que el modelo incremental bajo la idea de prototipos, puede satisfacer las necesidades de construcción para el proyecto presente, porque conforme avancemos en la implementación del prototipo este va a ser dotado, poco a poco, con funcionalidad mediante los incrementos, lo que proporcionará al cliente (el área de desarrollo de la UACM) una plataforma de evaluación y de esta forma, aseguran Pressman y Troya (1988, p. 23), se podría desarrollar un plan para un incremento en el futuro. “El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente y la entrega de funciones, y características adicionales. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo” (Figura 2.2).

Ahora pues, si partimos de la premisa de que un prototipo apoya a la fase de análisis de requisitos permitiendo incrementar la confianza en que los requisitos identificados son los que realmente desea el usuario y podemos reducir el riesgo de problemas en la aceptación final del sistema de software. Otras ventajas que podemos tener señala Sommerville y Galipienso (2005, p. 67) son:

- Los clientes pueden aclarar los requisitos que no tengan claros conforme ven las entregas del sistema.
- Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema. Pueden empezar a usarlo desde el primer incremento.
- Las partes más importantes del sistema son entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos.

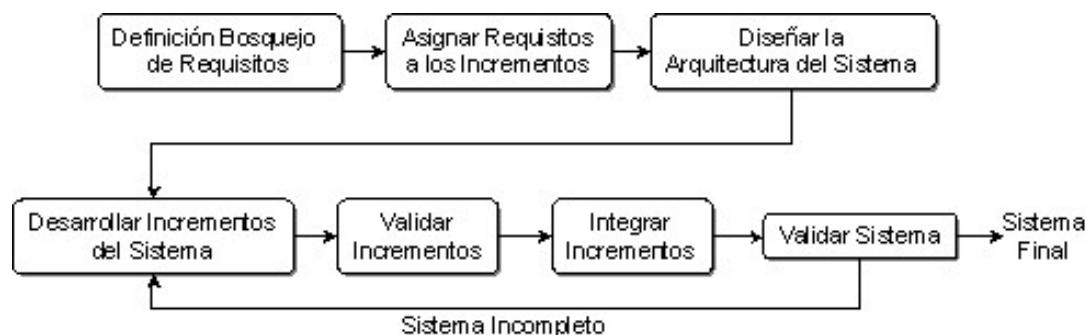


Figura 2.2: Propuesta de Sommerville y Galipienso (2005, p. 67) para la entrega incremental.

Concretamente, esta forma de desarrollo se ha demostrado útil y es un enfoque que está gozando de creciente popularidad dada la disponibilidad de tecnologías de software que lo facilitan, ya que algunas herramientas de software permiten, generar automáticamente interfaces gráficas o plantillas de diseño con las que el usuario puede interactuar, además del uso creciente de herramientas Case las cuales soportan las primeras etapas del desarrollo del prototipo, tampoco hay que olvidar el uso de frameworks los cuales agilizan el proceso de desarrollo.

Finalmente, otra particularidad más del desarrollo incremental y el uso de prototipos, es que puede resultar útil cuando la dotación de personal no está disponible para una implementación completa en la fecha límite que se haya establecido para el proyecto (como es nuestro caso).

Como resultado, se considera que los conceptos revisados hasta el momento ayudaron a delimitar el sistema de cuestionarios web para que este orientado a un prototipo, buscando entregar funcionalidad en términos de creación, edición y evaluación de cuestionarios. Por tanto, se delimita al prototipo de cuestionarios web con una descripción bajo una cierta perspectiva (por ejemplo, arquitectura, datos, comportamiento) empleando notaciones no necesariamente similares al código final; además de definirlo como parcial, porque no es

necesario que cubra todo el sistema sino aquellas partes o perspectivas que se pretenden analizar. Finalmente, será ejecutable para que la validación del sistema pueda hacerse a partir de la experimentación con el prototipo por parte de usuarios (administradores, profesores y estudiantes) y analistas (área de desarrollo de la UACM) y de este modo obtener el flujo de interacción deseado.

### 2.3. Tendencias y metodologías para la construcción de prototipos

Una vez que partimos de la idea de desarrollar un prototipo bajo ciertas características como el inicio de construcción por medio de objetivos y requisitos generales, así como de incrementos en la funcionalidad a través de un proceso iterativo, buscamos obtener una ayuda para administrar el progreso del desarrollo del prototipo, lo que derivó en metodologías o tendencias para el desarrollo de software, por esto resulta indispensable discutir un poco acerca de RUP y Scrum, así como diferentes artefactos<sup>3</sup> que derivan de ellas, ya que estas metodologías se utilizan con regular frecuencia, en el proceso de desarrollo de software.

Entre las metodologías tradicionales más populares se encuentran el Rational Unified Process (RUP) de IBM, “es un marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos. RUP está dirigido por casos de uso, centrado en la arquitectura, y es iterativo e incrementa” (Torossi, 2005, p. 3). En suma, centra su atención en mantener una documentación exhaustiva del proyecto y cumplir con un plan previsto, a pesar de su rigurosidad y alta documentación. Es un metodología altamente usada en el desarrollo de software, en contra parte la disciplina Scrum esta basada en la rapidez y flexibilidad de métodos de desarrollo ágiles ampliamente usados en la industria de productos comerciales. “La diferencia con respecto a RUP es que éste posee unas fases más secuenciales y definidas. En Scrum se le da mucho valor a la persona y se confía en que es capaz de llevar a cabo el proceso para llegar a una solución” (Villanueva y Siachoque, 2014, p. 3). Los principales contrastes entre el desarrollo tradicional (RUP) y el desarrollo ágil (Scrum), que servirán para este trabajo, se pueden observar a continuación.

En el enfoque RUP se establecen una serie de **reglas** para el desarrollo de un proyecto. En la literatura se puede encontrar que RUP es una implementación del desarrollo en

---

<sup>3</sup>**Artefactos.** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

espiral y es así porque sus orígenes se remontan a este modelo, pero más bien consiste en reproducir el ciclo de vida en cascada en menor escala mediante un ciclo de iteraciones hecha en sus cuatro fases o etapas (Inicio, Elaboración, Construcción y Transmisión). Con esto presente autores como Sommerville y Galipienso (2005, p. 78) dicen que “RUP no es un proceso apropiado para todos los tipos de desarrollo sino que representa una nueva generación de procesos genéricos”. Con ello se quiere decir que la metodología RUP es más apropiada para proyectos grandes (al igual que el modelo espiral), dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En pequeños proyectos, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios. A pesar de que RUP es una metodología pensada para proyectos de gran envergadura, llama la atención dos de sus características que pueden resultar útiles a este trabajo, la primera es que esta enfocado a Casos de Uso y la segunda es que divide el esfuerzo de desarrollo en partes mas pequeñas, es decir, es Iterativo e Incremental, estas características resultaran útiles ya que RUP está compartiendo el ideal de incrementar funcionalidad con resultados observables a través de su flujo de trabajo, muy similar a lo que se propone en el modelo de prototipado incremental y la otra cara es la usanza de casos de usos (artefactos) que nos brindan una ayuda para especificar la comunicación y el comportamiento del prototipo a desarrollar, centrándose en las tareas principales de usuario (describiendo los casos de mayor importancia).

En contraste con RUP, los principios de Scrum tienden a ser **líneas guía** para los desarrolladores. Scrum es un marco de trabajo o framework para el manejo de proyectos que tienen como fin el desarrollo de productos complejos y está especialmente dedicado para proyectos con un rápido cambio de requisitos, además dice Pressman y Troya (1988, p. 69) “se incorpora un conjunto de patrones de proceso que ponen énfasis en las prioridades del proyecto, las unidades de trabajo agrupadas, la comunicación y la retroalimentación frecuente con el cliente”, de este modo se pueden corregir problemas y mitigar riesgos de forma temprana. Según Canós, Letelier, y Penadés (2003, p. 7) sus principales características se pueden resumir en dos: “En primer lugar el desarrollo se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración”.

En relación con lo anterior podemos ver que Scrum puede ser una metodología para minimizar los riesgos en la realización de un proyecto, pero de manera colaborativa, lo que nos coloca en un situación donde es complicado llevar dicha metodología en términos

del personal o un equipo. Sin embargo, es posible que existan partes interesantes que, al igual que RUP, pueden aportar elementos útiles para este proyecto de prototipo. En este sentido, se observa que la metodología Scrum está compuesta por roles, reuniones, sprints y el uso de ciertas herramientas (artefactos). Queda entendido que asumir roles y hacer reuniones son conceptos difíciles de llevar a la práctica para este trabajo, pero la idea de los sprints suena atractiva debido a la similitud con el modelo Iterativo incremental, ya que estos señala (Canós y cols., 2003, p. 7) son iteraciones y para estas el autor añade que “deben ser de dos, tres o a lo sumo cuatro semanas. El resultado de cada sprint es un incremento ejecutable (release) que se muestra al cliente”, además de que un sprint está acompañado de reuniones diarias con el equipo. Con esto presente no podemos incluir en su totalidad al concepto de sprint ya que necesitamos recrear la parte de las reuniones diarias con un equipo, pero podemos tomar el concepto de **incremento ejecutable** o release, ya que en términos de artefactos se construye y se entrega un producto con ciertas funcionalidades. Finalmente, el mayor aporte que nos puede dar Scrum es que nos permite la libertad de ser breves con los artefactos, ya que al ser una metodología ágil la documentación así lo es también, es decir, para Scrum existe una propuesta de cuatro artefactos como mínimo, lo que puede resultar conveniente para nuestro prototipo, debido a que en este proyecto no existen las condiciones para un equipo de desarrollo.

A partir de estos conceptos, se quiere explicar en este trabajo, que si bien no existen recetas magistrales para aplicar metodologías a un determinado proyecto, es posible detectar buenas prácticas y otros patrones que hacen posible decidir acerca de los métodos a utilizar. Esto es, tomando lo que se sea necesario de cada metodología y adaptándola al proyecto en cuestión. Coincido con Canós y cols. (2003, p. 8) acerca de que “no existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etcétera). Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes”. Cuando se comienza con un proyecto de software es necesario que el líder de proyecto seleccione la metodología y el modelo más adecuados a los intereses del proyecto. Sin embargo, siempre está latente una cuestión: ¿cuál es el proceso más adecuado para un sistema de software en concreto? A lo que Serrano (1996, p. 71) dice: “La pregunta es inadecuada porque no depende sólo del tipo de sistema a desarrollar sino también de la organización encargada de ello”. Entonces no se trata de elegir de forma permanente, trabajar con una o con otra metodología, sino elegir aquella que mejor se adapte a las necesidades del proyecto. En cuanto a esto debemos

concretar la idea con la mención de un estudio realizado para determinar metodologías para desarrollo de software en México, en donde concluyen (Jiménez y Orantes, 2012, p. 14) que “es recomendable que se diseñe y utilice una metodología híbrida para el desarrollo de software, que integre prácticas de Ingeniería de Software, tomando los elementos que proporcionen mayores ventajas a las empresas, de acuerdo a las características del software a desarrollar, de los desarrolladores, hardware y tiempo requerido para el desarrollo, entre otros elementos”. Este estudio se realiza entrevistando a 86 empresas acerca de sus métodos de desarrollo de software en la que destacaron, las metodologías de tipo ágil, híbrida y tradicional. Se observó, además que, en México, por lo general, las empresas desarrolladoras de software no realizan un estudio de factibilidad y no utilizan un modelo de gestión y aseguramiento de la calidad. Por lo tanto, ellos recomiendan, que seguramente una metodología híbrida bien analizada, discutida y diseñada para tomar de los distintos modelos de desarrollo de software las características más ventajosas, ayudará a estas empresas a elaborar mejores planes de desarrollo del software, que permitan una mejor calidad de sus productos a menor costo y con un tiempo de desarrollo menor.

Retomando la idea y necesidad de analizar y discutir los modelos de desarrollo de software para conocer las ventajas que nos ofrecen se incorpora a continuación un cuadro comparativo (tabla 2.1) entre las dos metodologías mencionadas.

	RUP	Scrum
Enfoque	Iterativo	Iterativo
Iteraciones	Iteraciones de desarrollo. Pocas y largas.	Iteraciones de desarrollo. Muchas y frecuentes.
Ciclo	Ciclo formal se define a través de 4 fases.	Cada sprint (iteración) es un ciclo completo.
Artefactos	24 artefactos para proyectos simples.	Solamente 4 artefactos base.
Filosofía	Reglas para el desarrollo de un proyecto.	Líneas guía para los desarrolladores.
Características	Especialización. Más enfocado en los procesos (flujos de trabajo).	Equipo multidisciplinario. Más enfocado en las personas (prácticas de gestión).
Tipo de proyecto	Proyectos empresariales de largo alcance y tiempo.	Proyectos con requisitos cambiantes o implementación de mejoras rápidas.
Planificación	Plan formal de proyecto asociado a múltiples iteraciones que cubren el mismo de principio a fin.	No existe un plan de proyecto de principio a fin. Además el plan de la siguiente iteración se determina al final de la iteración previa.
Alcance	Es definido antes del comienzo del proyecto y plasmado en el documento de Alcance. El alcance puede revisarse a lo largo de la vida del proyecto pero existe un proceso estricto que controla las revisiones.	Es definido en la Lista de Objetivos (Project Backlog) que es reevaluado al fin de cada iteración (Sprint).

Tabla 2.1: Comparación entre RUP y Scrum.

Teniendo todo lo anterior en cuenta, y pensando que en el futuro el prototipo se pueda completar cumpliendo con las expectativas por parte del área desarrollo de software de la UACM, este proyecto incorporó un esquema híbrido entre RUP y Scrum enfocado en utilizar artefactos que ayuden con el ciclo de vida orientado a prototipos incrementales, ya que queda claro que los artefactos son importantes para el diseño de un sistema, pero se tiene un tiempo límite y un recurso humano único, que acotan mucho el proyecto, por lo que se necesita fluidez para concluir el desarrollo de esta versión “beta” del software de cuestionarios web.

## 2.4. Artefactos de software

Los métodos “tradicionales” están pensados para el uso exhaustivo de documentación durante todo el ciclo de vida del software, mientras que una metodología “ágil” no prohíbe la creación de artefactos o documentos adicionales sino que hace que se juzgue la necesidad, la utilidad y el costo vs. beneficio de cada uno de los artefactos que se decide agregar. Comparando ambas metodologías, el tipo de documentación de RUP hace referencia a las comunicaciones formales con la finalidad de ser más predictivos, mientras que en Scrum se enfoca en las comunicaciones informales continuas y a la adaptación al cambio, con la finalidad de ser más adaptativas. Bajo estas ideas y sumado al hecho de que siempre será necesario realizar ciertas actividades de modelamiento antes de iniciar un desarrollo, nos centramos en dos factores para nuestro prototipo en términos de las metodologías ya mencionadas; por el lado de Scrum se utilizó la idea de las historias de usuario priorizando con ellas las características más importantes del software, y del lado de RUP se realizaron los casos de uso que eran más relevantes para detallar el “core” de la aplicación. Examinemos esto con un poco más de detalle.

“Para RUP en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software” (López, 2013, p. 30). Dada la importancia de esto se usaron artefactos para capturar y llevar la información del proyecto. Cabe mencionar que la metodología RUP consta de veinticuatro artefactos en total divididos en cada una de sus disciplinas. Como dice López Márquez, Villa Palacio, y cols. (2012, p. 29) “... los artefactos se aplican dependiendo del tamaño y alcance de los proyectos”. Esto es importante mencionarlo puesto que para el proyecto actual sólo se piensa usar un limitado número de artefactos. Por esto solo se utilizaron los casos de uso, pero además debido a la naturaleza del proyecto, se contemplan otros artefactos a lo largo del desarrollo

del prototipo, estos serán útiles para el objetivo de este trabajo y serán: el documento de elicitación de requerimientos, el diagrama de entidad relación para el modelado de los datos y una métrica de puntos de función que nos proporciona un medio para la estimación del software, pero de estos se hablará un poco más adelante.

RUP es un proceso que posee tres características: es dirigido por los casos de uso, centrado en la arquitectura y es iterativo e incremental (Jacobson, Rumbaugh, Jacobson, Booch, y Rumbaugh, 2000, p. 4). Como se había dicho dos de estas características son interesantes para este proyecto, el uso del artefacto de casos de uso y la idea de iterar e incrementar funcionalidad. Sin embargo vale la pena revisarlos un poco más a detalle:

- a) **Casos de Uso:** Describe un servicio que el usuario requiere del sistema, incluye la secuencia completa de interacciones entre el usuario y el sistema.
- b) **Centrado en la arquitectura:** La arquitectura es utilizada para conceptualizar, construir, administrar y evolucionar el sistema en desarrollo.
- c) **Iterativo e Incremental:** Significa que la aplicación se divide en pequeños proyectos, los cuales incorporan una parte de las especificaciones, y el desarrollo de la misma es una iteración que va incrementando la funcionalidad del sistema de manera progresiva.

Para el caso en donde el proceso unificado está centrado en la arquitectura, hay que hacer una sencilla aclaración, si bien en ingeniería de software la arquitectura se puede ver como la estructura o “esqueleto” de un proyecto de software, es decir, la selección de los elementos estructurales a partir de los cuales se compone el sistema, entonces para el caso de nuestro prototipo este concepto de arquitectura se contempla, pero no se propone una arquitectura sino que se adopta del marco de trabajo que proporciona Laravel, el cual ya establece una estructura para el desarrollo de sistemas web.

Otro aspecto a considerar es lo relacionado con las iteraciones de desarrollo, que mientras que en RUP tienden a ser pocas y largas, en Scrum tienden a ser muchas pero frecuentes.

Según Palacio y Ruata (2014, p.24) Scrum propone cuatro herramientas o artefactos para mantener organizados nuestros proyectos. Estos artefactos, ayudan a planificar y revisar cada uno de los sprints, aportando medios necesarios para efectuar cada una de los procesos en Scrum. No menciona adrede cualquier otra documentación y/o artefactos. Esto a menudo lleva al malentendido de que los equipos ágiles no necesitan ninguna documentación. Cabe aclarar que no hay guías rígidas acerca de cómo llevar estos artefactos, por lo que cada organización hace lo que cree que es más conveniente. A continuación se definen estos artefactos:

- **Product Backlog** (Pila del producto) es una lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo. Sería el equivalente al documento de requisitos del sistema. Es un listado de características a obtener (historias de usuario), en orden de prioridad.
- **Sprint Backlog.** (Pila del sprint) es una lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto, es decir, es el conjunto de elementos tomados del Product Backlog que fueron priorizados. Estos, en conjunto con sus respectivas historias de usuario, forman oficialmente los requerimientos a elaborar en cada uno de los sprints que tendrá el proyecto. El equipo asume las tareas, y tienen estimados en el tiempo y los recursos necesarios.
- **Sprint**, nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso de avance por tiempos prefijados (time boxing).
- **Incremento** es el resultado de cada sprint. No se admite cambios a mitad de un incremento (sprint, iteración).

Cuando se está utilizando Scrum se suele sugerir que los equipos utilicen solamente aquellos artefactos que son realmente muy valiosos para ellos u otros, a corto y/o largo plazo, por ello a continuación nos centraremos en dos artefactos relevantes para este trabajo.

En primer lugar las Historias de Usuario (User Stories) que son un elemento especial del product Backlog. Son la técnica utilizada para especificar los requisitos del software. Se pueden pensar como tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales (figura 2.3). El tratamiento de las historias de usuario es muy dinámico y flexible. Las historias de usuario son descompuestas en tareas de programación (task card) y asignadas a los programadores para ser implementadas durante una iteración. Una vez priorizado las historias de usuario pasan a formar parte de la pila de producto o product backlog, esto se considera el sprint cero.

Entonces las historias de usuario son útiles puesto que se utilizan como elementos descriptores de las funcionalidades que tiene el software prototipo y son el resultado de la colaboración con el área de desarrollo de la UACM. Es una manera práctica de ir programando lo realmente importante, enfocándonos en definir lo que el usuario necesita hacer.

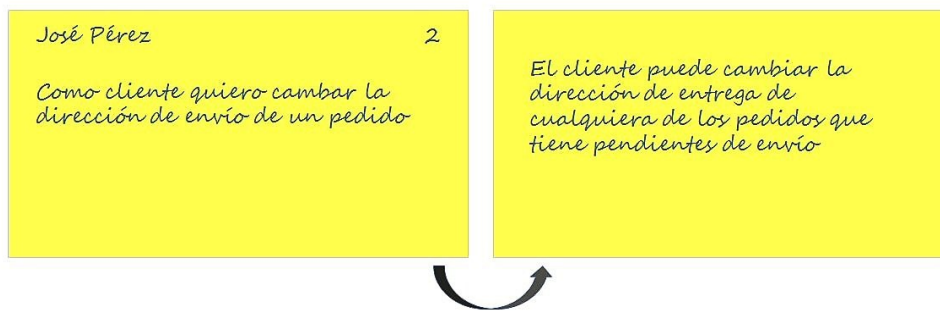


Figura 2.3: Ejemplo de historia usuario.

El segundo artefacto a considerar es el incremento. Palacio y Ruata (2014, p. 27) dicen en su libro que “el incremento es la parte de producto producida en un sprint, y tiene como característica el estar completamente terminada y operativa, en condiciones de ser entregada al cliente”. Con esto presente se considera que nuestro prototipo es un incremento ya que es un producto realizado en un intervalo de tiempo potencialmente entregable: terminado y probado. Aunque en la filosofía Scrum se dice que no se deben considerar como incremento a módulos o sub-módulos, prototipos, ni partes pendientes de pruebas o integración, pero se está refiriendo a que cada elemento de la pila del producto va a implicar a funcionalidades entregables, no a trabajos internos del tipo “diseño de la base de datos”. En el caso del sistema de cuestionarios en línea, si bien es un prototipo, pero se enfatiza que se busca funcionalidad útil para cliente, sin desechar nada, buscando que se tenga un refinamiento posterior a la entrega. Además Scrum añade que es una excepción frecuente que al comenzar con el desarrollo de un software se utilizan trabajos de diseño o desarrollo de prototipos pues es necesario contrastar las expectativas de la plataforma o tecnología que se va a emplear. Teniendo en cuenta esta excepción habitual se considera apropiado este uso de concepto por parte de Scrum.

Y por último, como Scrum menciona que si el proyecto o el sistema requiere documentación adicional, ésta también tienen que estar realizada para considerar que el incremento está “hecho”. Entonces habrá que hablar también, de modo muy breve, de los otros documentos o artefactos realizados durante este trabajo de receptacional.

## 2.5. Acerca de los requisitos

Si un proyecto de software fuese visto como una pirámide la base de este serían los requerimientos. Así pues que el proceso de elicitación de los requisitos de un sistema de software es fundamental para el éxito de un proyecto y con respecto a esto SCHACH y cols. (2005, p. 63) nos recuerdan que “la tarea principal de los analistas de sistemas de un equipo de requisitos es trabajar con el cliente y con los futuros usuarios del sistema de información para determinar que requieren, lo cual tal vez no sea lo que creen necesitar”, esto último quiere decir que es fácil mal interpretar lo que dice el cliente, lo que provoca que un proyecto necesite cambios rápidos en sus requisitos. Por esto, es importante tener un documento, que formalice las necesidades del cliente. Autores como Chaves (2011, p. 4) dicen que el proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema ayuda a determinar la viabilidad de llevar a cabo el software (si es factible llevarlo a cabo o no), además, añaden Herrera y Lizka (2003) que la importancia de generar un buen documento de requisitos a través de su análisis radica en beneficios como el permitir gestionar las necesidades del proyecto en forma estructurada, disminuir los costos y retrasos del proyecto, así como mejorar la calidad del software.

Una imagen (figura 2.4) que quizás para muchos es conocida, muestra en forma algo pintoresca, que un gran número de proyectos fracasan por no contar con una adecuada definición de los requerimiento y puede pasar porque no se evalúa correctamente lo que desea el cliente, o se conciben ideas distintas a las requeridas originalmente.

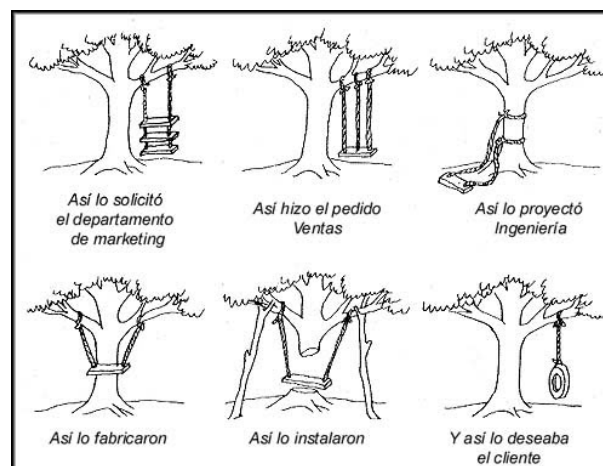


Figura 2.4: Importancia de los requisitos.

Preocupados por estos inconvenientes existe personas dedicadas en la calidad de los productos de software y proponen herramientas y metodologías para hacer del manejo

de los requisitorios una tarea que lleve al éxito de un proyecto, y tal es el caso de Toro y Jiménez (2000, p. 1) que propone una herramienta gratuita de su autoría para la elicitación de requerimientos acompañado de una metodología la cual tiene el objetivo de definir las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de elicitación de la fase de ingeniería de requisitos del desarrollo de software. Para el caso del prototipo de cuestionarios en línea, se usa dicha herramienta llamada REM, con la cual se obtiene un documento de elicitación de requisitos para el proyecto actual.

## 2.6. Modelo entidad relación y la importancia de la información

Dentro de la clasificación de los requerimientos existe un tipo llamado requisitos de información estos especifican qué información del sistema se debe almacenar para poder ofrecer la funcionalidad esperada. Como bien comenta Toro y Jiménez (2000, p. 34) “lo más importante en los sistemas de información es precisamente la información que gestionan”. De estos datos podemos identificar además, tres conceptos básicos: conjuntos de entidades, conjuntos de relaciones y atributos (figura 2.4). Todo esto para lograr un mejor manejo y calidad en la información de la aplicación.

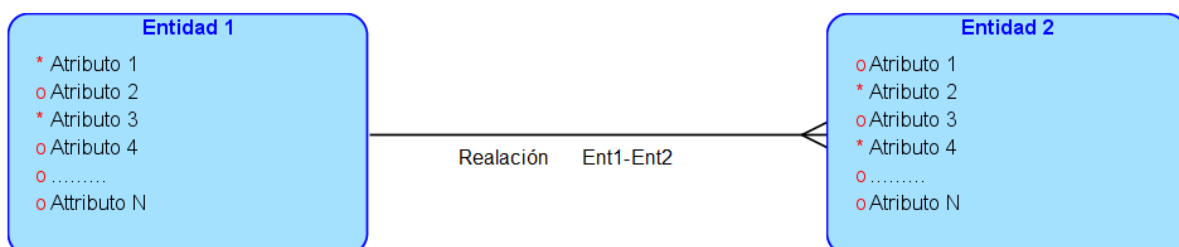


Figura 2.5: Modelado de datos.

Para cubrir el aspecto antes mencionado, para el prototipo se optó por hacer un modelo entidad relación (E-R) para facilitar el diseño de las bases de datos, ya que permite la creación de un esquema que representa la estructura global lógica de la base de datos. Cabe que este modelo considera a todos los requerimientos del cliente y no solo los que se contemplan para el prototipo. Siendo así un modelo semántico porque representa el significado de los datos, apoyado también por el diccionario de datos que es un listado organizado de todos los datos que pertenecen a un sistema. Donde su objetivo es dar precisión sobre los datos que se manejan en el sistema, evitando así malas interpretaciones o ambigüedades. Para poder lograr esto se utilizó la metodología de desarrollo de software

CASE\*METHOD y la herramienta Data Modeler, ambos de la compañía ORACLE. El pensar en esta herramienta sabiendo que existen otras como Navicat, PostgreSQL GUI Tools, SQL Power Architect, Open Systems Architect, entre muchas otras más, fue porque Data Modeler demostró ser una herramienta fácil e intuitiva en uso, con el cual ya había tenido contacto, en el modelado de otros sistemas más pequeños. En segunda instancia se piensa en la metodología mencionada ya que como dice Quiroz (2003, p. 9) “el modelo relacional de bases de datos con sus relaciones normalizadas es una solución simple y elegante para satisfacer las más diversas condiciones de consulta y extracción de datos e información” y el método Case de ORACLE nos proporciona eso en una forma práctica que garantiza la integridad y calidad en el manejo de los datos.

## 2.7. Documento de métrica para estimación del proyecto

Un ingeniero de Software recopila medidas y desarrolla métricas para obtener indicadores, para una mejor planificación para el desarrollo del producto. Por lo cual, en la metodología RUP un plan de desarrollo que se ejecuta al iniciar un proyecto, es aquí donde se define el alcance del mismo a partir de las solicitudes del cliente y en este sentido una métrica nos puede ofrecer elementos en forma de una medida estadística, es decir, nos puede ayudar a conocer la estimación de tiempos y costos del proyecto y preparar un ambiente para establecer un alcance y sus límites, así como el criterio de aceptación y en que consiste el proyecto. Para establecer que este proyecto se convirtiera en un prototipo, se usó una métrica al inicio del desarrollo, cabe decir que una métrica puede ser aplicable en todo el ciclo de vida del desarrollo. De esta forma, para establecer una línea de base para la estimación del software de cuestionarios web, se usó una métrica llamada puntos de función lite. Aunque cabe mencionar que existen otras métricas enfocadas a la usabilidad, mantenimiento del software, complejidad, competencia, desempeño, etcétera. Todas van íntimamente relacionadas con la calidad. Retomando, la elección de usar puntos de función lite se dió porque con ella se mide la aplicación desde una perspectiva del usuario dejando de lado los detalles de codificación. Esta evalúa con fiabilidad, es decir, estima el tamaño y complejidad del software con base en la cantidad de funcionalidad requerida y entregada a los usuarios. Se usó con la intención de conocer los costos con tiempo y recursos humanos necesarios para completar la aplicación en cuestión, y de esta forma determinar cual era el rumbo que debiera tomar el sistema, llevándonos a la conclusión de solo crear un prototipo con funcionalidad limitada.

## 2.8. Aplicaciones web y frameworks

Hoy en día las aplicaciones web han inundado el mundo debido a su practicidad ya que solo se necesita un navegador web y una conexión a internet. Mateu (2010, p. 13) dice: “El éxito espectacular de la web se basa en dos puntales fundamentales: el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones y el otro nos proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente”. La independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones de este estilo sin distribuir e instalar software a miles de usuarios, ha hecho que las aplicaciones desarrolladas para la web tengan ventajas respecto a las aplicaciones de escritorio. Por este motivo se optó por construir un prototipo bajo el concepto de una aplicación web que será un software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador, lo que permitirá una comunicación activa entre el usuario y la información. De modo que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, editar agregando preguntas a un cuestionario y acceder para resolver este mismo.

Por otra parte, al desear obtener un prototipo web con una limitante de tiempo y esfuerzo individual, se pensó en buscar una estructura de soporte, sin pretender “reinventar la rueda” y por tanto se llega a la decisión de usar un framework, ya que estos intentan aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web. Por ejemplo, muchos framework proporcionan bibliotecas para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la reutilización de código.

El hecho de elegir Laravel como framework para nuestro prototipo fue una decisión en términos de tendencias, ya que al hacer una investigación se encontró que Laravel, en voz de muchos desarrolladores, se definía por una legibilidad, sencillez y elegancia del código, lo que lo convierte en un excelente sistema de mapeo para desarrollar aplicaciones web, colocándolo como un marco de trabajo que permite mejorar la manera de poder programar, optimizando el tiempo y el costo en el desarrollo.

---

## Parte III

### Marco metodológico

---

## Capítulo 3

# Ingeniería de Requisitos y Análisis de Software

*“Caminar sobre el agua y desarrollar software con base a una especificación es fácil, si ambos están congelados”.*

- Edward V. Berard

Una de las tareas más importantes en el desarrollo de software es la captura y documentación de requisitos. A través de refinar y añadir conocimiento se llega a un nivel que permita identificar aspectos de la complejidad del sistema, de esta forma se realiza un análisis de alto nivel que permita delimitar el alcance del problema. Con base en la determinación del requerimiento, se deben utilizar métricas de estimación de la complejidad y riesgo que contribuyan a definir el tamaño, complejidad, tiempo y costo del desarrollo. Entonces podemos decir que los requerimientos son la base de las medidas de la calidad y la realización de métricas aportan un valor añadido. Por tanto, un ingeniero de software debe emplear mediciones que permiten evaluar y mejorar al proyecto o sistema, de esta manera inicialmente se puede hacer una estimación de la complejidad del sistema antes de aplicar la ingeniería de software.

En este capítulo se abordan los temas acerca de los requisitos en torno al sistema de cuestionarios web, mediante la introducción de conceptos generales referentes a la ingeniería de requisitos, además se explicará, además la utilización de la métrica llamada puntos de función lite para determinar el tamaño y la complejidad software usando los requisitos iniciales del sistema.

### 3.1. Ingeniería de Requerimientos

Como se vio en el capítulo anterior, el proceso de desarrollo de software está integrado por una serie de etapas que conducen al software bien diseñado; la primera etapa de este proceso es la extracción de requerimientos y la ingeniería de requerimientos es la encargada de este proceso.

En el libro *Classification of Research Efforts in Requirements Engineering* de 1997, citado por [Murga Tapia \(2013, p. 31\)](#) dice que “la ingeniería de requerimientos es la rama de la ingeniería de software interesada en los objetivos del mundo real para funciones y limitaciones en los sistemas de software”.

Otra definición más precisa se desprende de [Sommerville y Galipienso \(2005, p. 108\)](#) y dice: “Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos”.

Ahora pues, también es importante definir que es un requerimiento. De las muchas definiciones que existen se presentan dos tomadas del glosario de la IEEE ([IEEE Standards y cols., 1990, p. 62](#)). Entonces, un requerimiento puede definirse de las siguientes formas:

1. Un requerimiento es una condición o capacidad necesaria por un usuario para resolver un problema o alcanzar un objetivo.
2. Una condición o capacidad que debe conocerse y procesarse por una serie de componentes o sistema para satisfacer un contrato, estándar, especificación, u otra clase de documentación formalmente impuesta.

Los requerimientos de un sistema de software generalmente se clasifican en requerimientos funcionales y no funcionales.

1. **Requerimientos funcionales:** son los servicios que un sistema debe proveer. Cómo el sistema debe actuar a partir de ciertas entradas y cómo debe comportarse en situaciones particulares. En algunos casos los requerimientos funcionales deben definir qué es lo que el sistema no debe hacer.

2. Requerimientos no funcionales: son las restricciones en los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, restricciones en el proceso de desarrollo y estándares, es decir aquellas características del sistema que afectan la calidad del servicio y la forma como el sistema se comporta desde el punto de vista del usuario final. Dentro de este grupo se tienen en cuenta los siguientes factores: Desempeño, Disponibilidad, y Usabilidad.

## 3.2. Características del sistema

De acuerdo con la metodologías RUP y SCRUM, explicadas en el capítulo anterior, antes del análisis y diseño del sistema, se tienen que realizar la captura de los requerimientos, para ello se tuvo que realizar una serie de reuniones con el cliente (que para este sistema el cliente es parte del personal del área de sistemas de la Universidad) y en las reuniones se describieron las necesidades con respecto a la aplicación y a lo largo del proyecto se realizaron varias entrevistas para el refinamiento de los requisitos.

Se decidió utilizar el concepto las características (features) o funcionalidades del software a desarrollar que se tiene en SCRUM, para tener un documento inicial de los requisitos (figura<sup>1</sup> 3.1), formando historias de usuario, que si bien, con el fin de no tener ambigüedades también se trabajaron casos de uso para refinar ciertas historias de usuario.



Figura 3.1: Algunas características capturadas con la aplicación web iceScrum.

<sup>1</sup><https://www.icescrum.com/a/>

Una de las características principales que definen al sistema de cuestionarios, es que sea una aplicación web para facilitar el acceso al software, sin tener que instalarse localmente en cada máquina de un laboratorio, y que además de exámenes para estudiantes de integración, se puedan crear cuestionarios en general para cualquier temática. Otras características o requerimientos<sup>2</sup> funcionales para el sistema de cuestionarios web, se enuncian a continuación:

- Que existan diferentes roles para los usuarios, los roles que se necesitan para el sistema son: Un administrador del sistema, un editor, colaboradores, participantes y un evaluador.
- Creación y edición de un cuestionario.
- Creación de preguntas abiertas y de opción múltiple.
- La posibilidad de agregar preguntas a un cuestionario, las cuales también podrán estar disponibles en un baúl virtual para su reutilización en otros cuestionarios.
- Fecha y hora de aplicación.
- La participación de colaboradores para la publicación de un cuestionario.
- Los cuestionarios podrán relacionarse con un tema.
- La posibilidad de resolver un cuestionario.
- Ver resultados después de resolver un cuestionario.

Sumando a lo anterior, algunos de los requerimientos no funcionales, o también conocidos como atributos de calidad, se listan ahora:

- Interfaz sencilla y amigable.
- Adaptable a cualquier explorador Web.
- Seguridad de la aplicación.

---

<sup>2</sup>Para más detalles acerca de los requerimientos del sistema ver Anexo A.

En conjunto estos requerimientos dan una perspectiva general del sistema que por tanto se pueden ilustrar mediante el uso de un diagrama de contexto<sup>3</sup> que es una vista de alto nivel de un sistema. A continuación el diagrama de contexto en la figura 3.2 que ilustra las entidades externas que tendrán interacción con el sistema de cuestionarios.

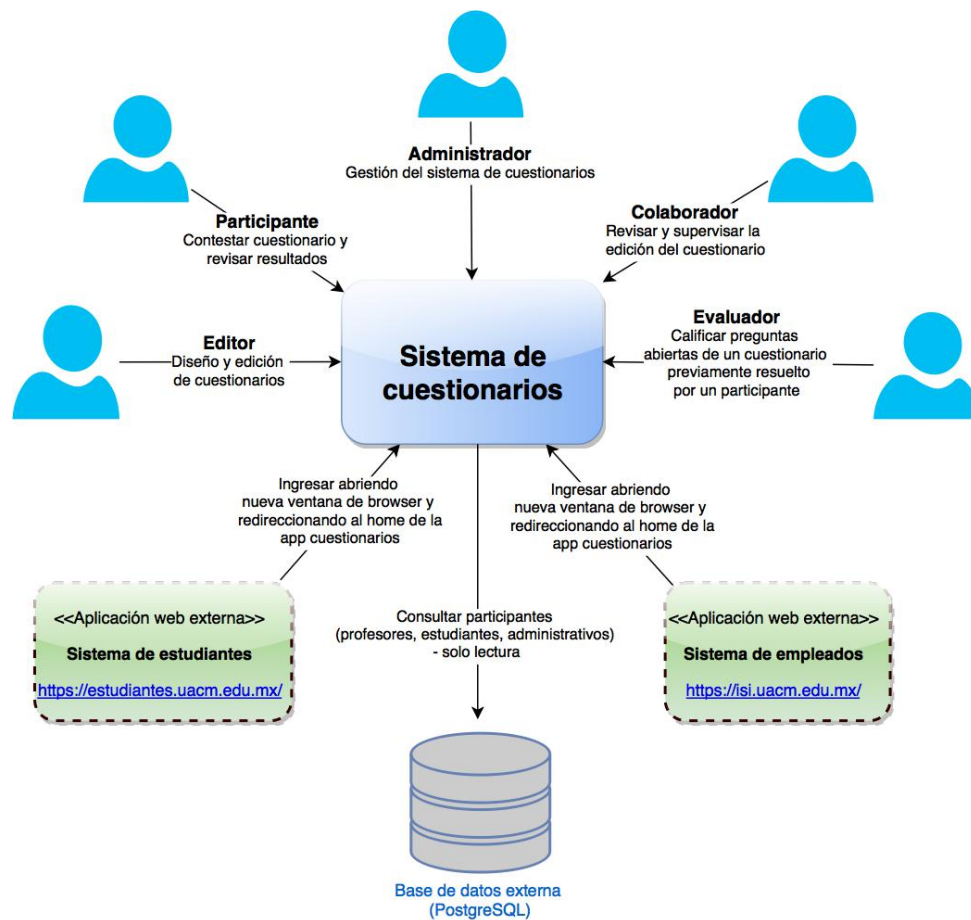


Figura 3.2: Diagrama de contexto inicial para el sistema de cuestionarios.

El diagrama representa todas las entidades externas que podrían interactuar con el sistema de cuestionarios web, retrata el sistema en el centro, sin detalles de su estructura interna, rodeado por todos los sistemas, ambientes y actividades con las que interactúa. De esta forma se logra enfocar la atención en los factores externos y eventos que deberían considerarse en el desarrollo cuando el conjunto completo de requerimientos y restricciones del sistema, se hayan implementado.

<sup>3</sup>Un Diagrama de Contexto de Sistema (DCS) en Ingeniería de software e Ingeniería de sistemas es un diagrama que define los límites entre el sistema, o parte del sistema, y su ambiente, mostrando las entidades que interactúan con él.

### 3.2.1. Historias de usuario

Como se dijo en un principio, para el desarrollo de la aplicación las características principales de requerimientos se especificaron como historias de usuario las cuales describen brevemente lo que el sistema debe poseer o dicho de otra forma las historias de usuario hicieron la forma “ágil” de definir características (features) de la aplicación, teniendo así requisitos funcionales y no funcionales.

Una historia de usuario para el sistema puede verse de la siguiente manera: *Como Administrador, Yo puedo crear un cuestionario, para que un Editor, tenga un nuevo cuestionario para su edición.* Aquí se puede observar que de manera práctica y sencilla se solicita una funcionalidad de crear un cuestionario a petición de un usuario. Entonces si juntamos varias más de estas historias tenemos una pila que podemos priorizar y de esta forma empezar a desarrollarse para ir creando entregables al cliente y generando iteraciones.

### 3.2.2. Casos de uso

Las historias de usuario en ocasiones pueden estar carentes de detalles y para compensar eso, es necesario el uso de instrumentos o artefactos que ayuden a la definición concreta de una parte importante del sistema, en este sentido se utilizaron los casos de uso. A continuación se presenta la descripción de tres casos de uso principales correspondientes a la creación, edición y resolución de un cuestionario<sup>4</sup>.

#### 3.2.2.1. Definición de actores

Antes de iniciar con los casos de uso se presentarán los diferentes actores que se identificaron para el sistema y de los cuales serán actores dentro de los escenarios para cada plantilla de caso de uso mencionado.

---

<sup>4</sup>Otros casos de uso se encuentran en el Anexo A

ACTOR 01	<b>Administrador</b>
Descripción	El administrador es la persona que tendrá la responsabilidad de configurar, mantener, monitorear, documentar y asegurar el correcto funcionamiento del sistema informático.
Comentarios	La principal tarea de este usuario es la creación y actualización de los cuestionarios.

ACTOR 02	<b>Editor</b> (Diseñador)
Descripción	Es el usuario responsable de construir el cuestionario, crear e insertar las preguntas y sus respectivas respuestas.
Comentarios	Un usuario editor, será un profesor perteneciente a la UACM.

ACTOR 03	<b>Colaborador</b> (Revisor)
Descripción	Será la persona que se encargará de revisar la integridad del cuestionario y dará su voto de aprobación, para que el cuestionario pueda ser resuelto por un participante. En caso de no aprobación podrá hacer comentarios con respecto al contenido del cuestionario.
Comentarios	Un usuario colaborador es un profesor que pertenezca a la UACM.

ACTOR 04	<b>Participante</b>
Descripción	Es el usuario que resolverá el cuestionario una vez que este haya sido aprobado.
Comentarios	Un participante es un estudiante de la UACM.

ACTOR 05	<b>Evaluador</b>
Descripción	Es la persona delegada a revisar cuestionarios con preguntas abiertas.
Comentarios	Un usuario evaluador podrá ser un profesor de la UACM.

Cabe aclarar que el usuario evaluador no figura como usuario activo en este proyecto puesto que el sistema es un prototipo acotado a preguntas de opción múltiple, aunque el actor o rol de evaluador fue encontrado en los requisitos del sistema.

### 3.2.2.2. Caso de uso: Crear cuestionario

La tabla 3.1 hace referencia a la especificación de caso de uso crear cuestionario.

<b>Nombre</b>	CU-01. <b>Crear de cuestionario</b>	
<b>Descripción</b>	El sistema debe permitir la creación de cuestionarios, pero sólo un usuario administrador puede hacerlo.	
<b>Actores</b>	<b>Actor</b>	<b>Actor del sistema</b>
	Actor 1	Administrador
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El actor debe haber iniciado sesión.</li> <li>2. El futuro editor y los posibles colaboradores deben estar dados de alta como profesores de la UACM.</li> <li>3. Un profesor hace una petición al administrador para crear un cuestionario.</li> </ol>	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor solicita la creación del cuestionario.
	2	El sistema muestra un formulario para introducir los datos del cuestionario.
	3	El actor introduce los datos de cuestionario al sistema, proporcionados por el profesor y coloca el nombre del profesor solicitante como editor.
	4	El sistema solicita ingresar la información de la fecha de aplicación del cuestionario.
	5	El actor ingresa la fecha.
	6	El administrador da la acción de crear cuestionario.
	7	El sistema comprueba la validez de los datos y los almacena.
<b>Alternativas / erróneas</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el profesor decide introducir usuarios colaboradores.
	a1.	El actor solicita los nombres.
	a2.	El sistema asocia cada uno de los nombres al cuestionario.
	a3.	El sistema continúa con el paso 4.
	3	Si el profesor decide colocar una contraseña.
	a1.	El actor ingresa la contraseña.
	a2.	El sistema comprueba la redundancia de la contraseña.
	a3.	El sistema continúa con el paso 4.
	<b>Postcondición</b>	Se tendrá un cuestionario con título, descripción, una sección por default, una fecha de aplicación y una contraseña opcional, además de contar con o sin colaboradores.
<b>Reglas de negocio</b>	El cuestionario se le podrán agregar un máximo de 5 colaboradores. El profesor que solicita el cuestionario, puede pedir a otro usuario como editor y no ser él exactamente el editor.	

Tabla 3.1: Casos de uso: Crear cuestionario.

### 3.2.2.3. Caso de uso: Editar un cuestionario

La tabla 3.2 muestra el caso de uso editar y publicar un cuestionario, el cual solo contempla la inserción de preguntas de opción múltiple cumpliendo en el objetivo de este trabajo.

<b>Nombre</b>	<b>CU-02. Editar y publicar un cuestionario</b>	
<b>Descripción</b>	Un usuario editará un cuestionario pudiendo insertar y eliminar preguntas y secciones. Cuando la edición del cuestionario termine el usuario podrá publicarlo.	
<b>Actores</b>	<b>Actor</b>	<b>Actor del sistema</b>
	Actor 2	Editor
<b>Precondición</b>	1. El usuario debe haber iniciado sesión. 2. Que el administrador haya creado un cuestionario y que en éste se tenga indicado que el usuario actual es un Editor.	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra los cuestionarios en lista para su posible edición.
	2	El usuario elige el cuestionario.
	3	El sistema presenta una vista con los controles necesarios para inserción de secciones, preguntas de baúl y preguntas nuevas, eligiendo una de estas opciones en el momento que el actor lo decida.
	4	Si el usuario elige agregar una nueva pregunta.
	4.1	Se realiza el caso de uso <i>Insertar nueva pregunta de opción múltiple</i> .
	5	Si el usuario elige agregar una pregunta de baúl.
	5.1	Se realiza el caso de uso <i>Insertar nueva pregunta de baúl</i> .
	6	Si el usuario elige agregar una sección.
	6.1	Se realiza el caso de uso <i>Insertar sección</i> .
	7	El sistema comprueba (para los tres pasos anteriores) la validez de los datos y los almacena.
	8	El usuario elige la opción para publicar el cuestionario.
	9	El sistema solicita confirmación.
10	El usuario decide si aprueba o rechaza la publicación.	
11	Una vez aceptada la solicitud el sistema cambia el estado del cuestionario a publicado.	
12	El caso de caso de uso termina.	
<b>Alternativas / erróneas</b>	<b>Paso</b>	<b>Acción</b>
	11	Si el cuestionario no tiene colaboradores.
	a1.	El sistema colocara el estado del cuestionario en aprobado.
	a2.	El caso de caso de uso termina.
<b>Postcondición</b>	Se tiene un cuestionario editado y con cambio en su estado ya sea publicado o aprobado.	
<b>Reglas de negocio</b>	Un baúl de preguntas será una colección de preguntas ya escritas con anterioridad y que se pueden utilizar posteriormente en un nuevo cuestionario.	

Tabla 3.2: Casos de uso: Editar y publicar un cuestionario.

### 3.2.2.4. Caso de uso: Resolver un cuestionario

La tabla 3.3 hace referencia a la especificación de caso de uso Resolver un cuestionario.

<b>Nombre</b>	CU-03. <b>Resolver un cuestionario</b>	
<b>Descripción</b>	Este caso de uso describe como un usuario participante resuelve un cuestionario que está en estado de aprobado.	
<b>Actores</b>	<b>Actor</b>	<b>Actor del sistema</b>
	Actor 4	Participante
<b>Precondición</b>	1. El usuario debe haber iniciado sesión. 2. Que el cuestionario a resolver esté con estado de aprobado. 3. Que la fecha actual esté dentro del intervalo de la fecha de aplicación.	
<b>Secuencia principal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema muestra la lista de cuestionarios disponibles.
	2	El actor elige el cuestionario a resolver.
	3	El sistema muestra la vista del cuestionario seleccionado.
	4	El actor responde las preguntas, pudiendo dejar alguna sin responder.
	5	El sistema captura las repuestas del usuario.
	6	El actor confirma al sistema que ha concluido el cuestionario.
	7	El sistema informa el número de preguntas respondidas, si usuario decide puede cancelar y corregir o responder alguna pregunta.
	8	El actor confirma la terminación del cuestionario.
9	El sistema muestra la calificación obtenida.	
<b>Alternativas / erróneas</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el cuestionario tiene contraseña.
	a1.	El sistema solicita la contraseña.
	a2.	El actor ingresa la contraseña.
	a3.	El sistema comprueba la contraseña, si esta es correcta continúa con el paso tres, sino se cancela el caso de uso.
<b>Postcondición</b>	Se tiene un cuestionario resuelto, con resultados para usuario y con un estatus de resuelto.	
<b>Reglas de negocio</b>		

Tabla 3.3: Casos de uso: Crear cuestionarios.

### 3.3. Métrica puntos de función lite

La métrica del punto función es un método utilizado en ingeniería de software para medir el tamaño del software. Fue definida por Allan Albrecht (1979), de IBM en “Measuring Application Development Productivity”. Con más detalle la métrica sirve para medir la aplicación desde una perspectiva del usuario, dejando de lado los detalles de codificación, esto es, que es una técnica totalmente independiente de todas las consideraciones de lenguaje de programación, y evalúa fiabilidad, es decir, considera aspectos como: tamaño del proyecto, coste y tiempo de desarrollo. Su objetivo es el medir lo que el usuario pide y lo que el usuario recibe, medir independientemente de la tecnología utilizada en la implantación del sistema, proporcionar una métrica del tamaño, dar un medio para la estimación del software.

Un punto de función (PF) es una unidad de medida para expresar la cantidad de funcionalidad empresarial que un sistema de información proporciona a un usuario. De esta manera un programa que tenga “x” PF’s entrega “x” funciones al usuario final. El mejor modo de trabajo es la interacción analista-usuario.

Los PF miden dos tipos de características:

- Los elementos de función (entradas, salidas, archivos, etcétera.)
- Los factores de complejidad.

El método PF Lite es una derivación del método el IFPUG (International Function Point Users Group), ya que el método tradicional del IFPUG es bastante complejo de implantar<sup>5</sup>. El FP Lite lo que hace es reducir y simplificar las fases del IFPUG.

---

<sup>5</sup>Para conocer más acerca del procedimiento para el calculo de puntos de función ver el instructivo para la cuenta de puntos función en <http://docplayer.es/223556-Instructivo-para-la-cuenta-de-puntos-funcion.html>

## 3.4. Estimación

A continuación se describe el procedimiento para el cálculo con puntos de función (con PF-Lite) para el sistema de cuestionarios web.

### 3.4.1. Tipo de conteo

Para comenzar con la métrica se tiene que establecer un tipo de conteo para el proyecto de acuerdo a sus características. Por ejemplo, se considera si es un nuevo proyecto, de mantenimiento, para escritorio, con un nuevo lenguaje, etcétera. Estas características se toman de las tablas de parámetros de estimación de esfuerzo de ISBSG<sup>6</sup> (International Software Benchmarking Standards Group) del 2005; en nuestro caso aunque se tiene un sistema como antecedente, la construcción no hace modificaciones al sistema anterior, por lo tanto, se consideró como un proyecto nuevo. Esta decisión repercute en los factores de calibrado o de corrección C (complejidad) y E (esfuerzo) que se encuentran presentes en la tabla de parámetros antes mencionada. Los parámetros se utilizarán posteriormente en la fórmula para estimar el esfuerzo.

### 3.4.2. Delimitación de los límites / frontera de la aplicación

Además del tipo de conteo se tiene que acotar los límites/frontera de la aplicación mediante un diagrama de límite. Para ayudar a identificar las funciones de datos conocidos como ILF (por sus siglas en inglés, Internal Logical File) y ELF (del inglés External Interface Files), así como las funciones transaccionales EI (External Input), EQ (External Query) y EO (External Output). Entonces, por un lado estará el sistema de información de la UACM (<https://isi.uacm.edu.mx/>) y por el otro el sistema para los estudiantes (<https://estudiantes.uacm.edu.mx/>). Ambas serán referenciadas por la aplicación de cuestionarios (figura 3.3).

---

<sup>6</sup>Tablas tomadas de: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/40580/4/jsanchez-secoTFC0115memoria.pdf>

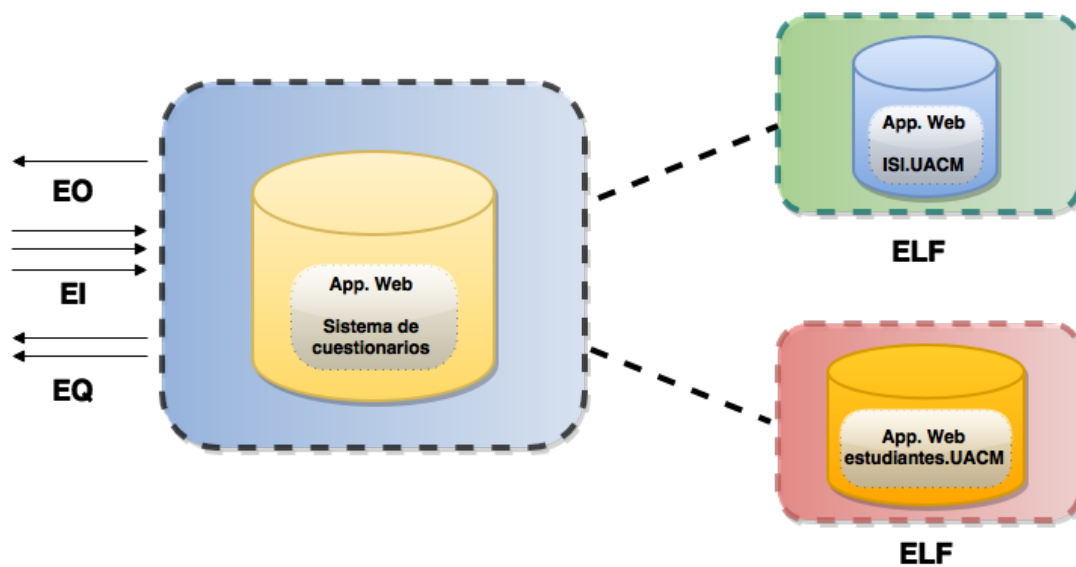


Figura 3.3: Diagrama de límite de la aplicación.

### 3.4.3. Identificación de las funciones de datos

#### 3.4.3.1. Identificación de ILFs

Cabe aclarar que los archivos lógicos internos (ILF), no aparecen en el diagrama porque que estos son reconocidos por el usuario y mantenidos dentro de los límites de la aplicación; los que se consideraron corresponden con las entidades que aparecen en el diagrama entidad relación. Hay que señalar que algunas de las entidades, que a continuación se mencionarán, no están presentes en la versión actual del diagrama de entidad relación, puesto que la métrica se realizó al comienzo del proyecto y posteriormente el diagrama evolucionó.

- Cuestionario
- Aplicación
- Hoja respuesta
- Hoja de respuesta
- Colaborador
- Instrucción
- Tipo discriminante
- Subtipo discriminante
- Respuesta PA
- Resp. POM-Simple
- Sección
- Cuestionario Pregunta
- Pregunta
- Texto
- Respuesta POM-Simple
- Valor escala indicador
- Escala indicador
- Acomodo indicador
- Tema
- Imagen
- Indicador
- Rúbrica

#### 3.4.3.2. Identificación de ELF's

El sistema de cuestionarios, un en futuro, tendrá la responsabilidad de mantener vínculos con las tablas del sistema de estudiantes y el sistema para profesores, para hacer el logeo y obtención de claves para los participantes<sup>7</sup>, ya que cada rol en el sistema tendrá acceso por los sistemas ya mencionados, por lo tanto, los archivos lógicos externos (EIF) identificados por el usuario, pero dentro de los límites de otra aplicación, serian dos, pues son dos sistemas distintos que tienen un almacenamiento referenciado por la aplicación, pero no es mantenido por este (esta mención es observable en el diagrama de límite de la aplicación).

#### 3.4.4. Identificación de las funciones transaccionales

En el caso de identificación de las funciones transaccionales, que representan una funcionalidad suministrada al usuario por el procesamiento de datos en la aplicación, se obtienen tres cosas, las Entradas Externas (EI), Consultas Externas (EQ), y el envío datos hacia afuera de los límites de la aplicación (EO).

---

<sup>7</sup> Cabe decir que la obtención de claves para los participantes es mantenido por una aplicación externa. Sin embargo, la aplicación tendrá acceso a estos datos a modo de solo lectura, aunque es referenciada de manera externa, no se tiene la capacidad de modificar la información de ese grupo de datos y añadiendo, tampoco ha sido identificado por el usuario, sino ha sido una detección por parte del análisis y el diseño de la aplicación. Por lo tanto, no se considera el conteo de ELF.

#### 3.4.4.1. Determinación de los procesos elementales

Estos se determinan como unidades mínimas de actividad significativa para el usuario, que deja al negocio en un estado consistente. Como por ejemplo: agregar preguntas o editar un cuestionario. Los procesos elementales identificados son:

- P1. Agregar preguntas
- P2. Borrar cuestionario
- P3. Agregar pregunta de colección
- P4. Editar cuestionario
- P5. CRUD cuestionario
- P6. Dar de alta una aplicación de cuestionario
- P7. Ceder derechos de edición a otro colaborador
- P8. Cambiar estatus al cuestionario
- P9. Restaurar estatus de edición de cuestionario
- P10. Dar visto bueno a cuestionario
- P11. Habilitar / deshabilitar usuarios
- P12. Escribir comentarios
- P13. Responder cuestionario
- P14. Evaluar pregunta abierta
- P15. Listar cuestionarios en edición
- P16. Listar evaluaciones pendientes
- P17. Consultar cuestionarios
- P18. Consultar top de preguntas
- P19. Consultar resultados

#### 3.4.4.2. Determinación de Entradas Externas (EI)

Las transacciones con el objetivo de actualizar archivos lógicos internos o modificar el comportamiento del sistema que provienen de fuera de los límites de la aplicación identificados son:

- 1. Agregar preguntas
- 2. Borrar cuestionario

3. Agregar pregunta de colección
4. Editar cuestionario
5. CRUD cuestionario
6. Dar de alta una aplicación de cuestionario
7. Ceder derechos de edición a otro colaborador
8. Cambiar estatus al cuestionario
9. Restaurar estatus de edición de cuestionario
10. Dar visto bueno a cuestionario
11. Habilitar / deshabilitar usuarios
12. Escribir comentarios
13. Responder cuestionario
14. Evaluar pregunta abierta

- Total de EI= 14;

Estas operaciones de EI son propuestas ya que los datos se reciben de fuera de la aplicación y/o actualizan uno o más ILFs.

#### **3.4.4.3. Determinación de EOs**

El proceso Consultar resultados, envía datos hacia afuera de los límites de la aplicación mediante un proceso que implica operaciones matemáticas. Debido a que los resultados de los cuestionarios no se almacenan en la aplicación, sino que tienen que ser construidos a partir de la suma y promedio de los aciertos, se toma esta tarea como un proceso lógico EO.

- Total de EO = 1;

#### **3.4.4.4. Determinación de EQs**

Los procesos siguientes son consultas externas, debido a que solo recuperan datos y no mantienen ningún ILF:

1. Listar cuestionarios en edición
2. Listar evaluaciones pendientes

- 3. Consultar cuestionarios
- 4. Consultar top de preguntas

- Total de EQs=4;

### 3.4.5. Valor final de los PF para proyectos de desarrollo

Para los cálculos del valor de lo puntos de función tenemos:

Elemento	Peso	Cantidad	Total = cantidad * peso
<b>ILF</b>	10	22	220
<b>ELF</b>	7	2	14
<b>EI</b>	4	13	52
<b>EO</b>	5	1	5
<b>EQ</b>	4	4	16
		<b>PF</b>	307
		+20 %	307 + 61.4 = <b>368.4</b>
		-20 %	307 - 61.4 = <b>245.6</b>

Tabla 3.4: Complejidad media para cada elemento funcional.

### 3.4.6. Estimación del esfuerzo

El esfuerzo es una medida abstracta que representa el promedio de trabajo (en horas) necesario por persona. La estimación del esfuerzo se realiza con la siguiente fórmula:

$$\text{Esfuerzo} = C * PF^E$$

Donde PF es la cantidad de puntos de función calculados previamente y C y E (complejidad y esfuerzo respectivamente) son los factores de calibrado o de corrección de las tablas de parámetros de estimación de esfuerzo (ISBSG,2005). Para su utilización en la aplicación del sistema de cuestionarios, se usó la línea 32 de la tabla de estimación

de esfuerzo, ya que es un desarrollo nuevo, en lenguajes 3GL (PHP, Javascript, etcétera.) y se realizará de manera multiplataforma al ser una aplicación web.

Se debe hacer el cálculo teniendo en cuenta el rango  $\{PF_{Max}, PF, PF_{Min}\}$ , calculado previamente. Por lo tanto el cálculo del esfuerzo para la aplicación de cuestionarios será:

$$\text{Esfuerzo } pf_{Max} = 58,16 * 368,4^{(0,664)} = 2942,12 \text{ horas}$$

$$\text{Esfuerzo } pf = 58,16 * 307^{(0,664)} = 2606,65 \text{ horas}$$

$$\text{Esfuerzo } pf_{Min} = 58,16 * 245,6^{(0,664)} = 2302,04 \text{ horas}$$

Los valores obtenidos resultan la cantidad de horas necesarias para llevar a cabo la implantación.

### 3.4.7. Estimación de la duración

La fórmula de la duración es:

$$\text{Duración} = C * PF^E$$

Como la aplicación que estamos analizando no está dentro de ninguna de las categorías de la tabla de duración de ISBSG utilizaremos los valores de la fórmula siguiente:

$$\text{Duración} = 0,411 * \text{Esfuerzo}^{(0,328)}$$

Por lo tanto, la estimación de duración para la aplicación de cuestionarios será la siguiente:

$$\text{Duración } pf_{Max} = 0,411 * 2942,12^{(0,328)} = 5,64 \text{ meses}$$

$$\text{Duración } pf = 0,411 * 2606,65^{(0,328)} = 5,42 \text{ meses}$$

$$\text{Duración } pf_{Min} = 0,411 * 2302,04^{(0,328)} = 5,20 \text{ meses}$$

El valor resultante expresa la cantidad de meses necesarios para realizar el trabajo de implementación de la aplicación.

### 3.4.8. Estimación de cantidad de personal

La cantidad de personas necesarias para el desarrollo de nuestra aplicación teniendo en cuenta que se consideran 20 días laborables al mes y 8 horas por día, con la siguiente fórmula:

$$\text{CantidadPersonas} = \text{Esfuerzo} / (\text{Duracion} * 20 * 8)$$

Aplicando la fórmula anterior obtenemos:

$$\text{Cantidad de Personas}_{max} = 2942,12 / (5,64 * 20 * 8) = 3,26 \text{ personas}$$

$$\text{Cantidad de Personas} = 2606,65 / (5,42 * 20 * 8) = 3,00 \text{ personas}$$

$$\text{Cantidad de Personas}_{min} = 2302,04 / (5,20 * 20 * 8) = 2,76 \text{ personas}$$

El resultado nos indica que se necesitaran tres personas para completar el sistema.

### 3.4.9. Estimación del costo

Según la revista FORBES Ingenieros de Software, Arquitectos de Software, Técnicos con conocimiento del negocio, Tecnologías de la información en general, el Rango salarial<sup>8</sup> es de \$25,000 + bono.

$$\text{Costo por día} = 25000 / 20 = \$1250$$

$$\text{Costo por hora} = 1250 / 8 = \$156,25$$

$$\text{Costo} = \text{Esfuerzo} * [\text{costo promedio por hora}]$$

$$\text{Costo } pf_{Max} = 2942,12 * 156,25 = \$459,706,25$$

$$\text{Costo } pf = 2606,65 * 156,25 = \$407,289,06$$

$$\text{Costo } pf_{Min} = 2302,04 * 156,25 = \$359,693,75$$

---

<sup>8</sup>Dato crudo 2014. Fuente: <http://www.forbes.com.mx/las-10-profesiones-mas-cotizadas-en-el-mercado-laboral/>

La información arrojada por la formula nos muestra la inversión que implicará el desarrollo del sistema, que involucra los sueldo de los tres desarrolladores, basado en su esfuerzo necesario.

### 3.4.10. Estimación de productividad y velocidad de entrega

La productividad de un recurso de software se mide en función de la proporción entre lo que entra y sale de un proceso de producción de software.

$$\textit{Productividad (expresada en horas por PF)} = \textit{Esfuerzo} / \textit{PF}$$

$$\textit{Productividad } pf_{Max} = 2942,12/368,4 = 7,98_{h/PF}$$

$$\textit{Productividad } pf = 2606,65/307 = 9,58_{h/PF}$$

$$\textit{Productividad } pf_{Min} = 2302,04/245,6 = 11,97_{h/PF}$$

Estos resultados nos indican la relación entre la entrada de una actividad y su salida, es decir, nos muestra en el rendimiento del proceso de la ingeniera de software.

$$\textit{Velocidad de entrega (expresada en PF por mes)} = \textit{PF} / \textit{Duración}$$

$$\textit{Vel. Entrega } pf_{Max} = 368,4/5,64 = 65,31$$

$$\textit{Vel. Entrega } pf = 307/5,42 = 67,97$$

$$\textit{Vel. Entrega } pf_{Min} = 245,6/5,20 = 70,84$$

Estos resultados nos indican la velocidad que debe tener para que existan entregables al cabo de un mes, hasta que la entrega del producto final.

### 3.4.11. Estimaciones por fase

A continuación se muestra la estimación por fase de ciclo de vida para proyectos de desarrollo (Tabla:3.5).

Esfuerzo total = 2606.65 horas  $\implies$  100 %

Planificación	Especificación	Diseño	Construcción	Pruebas	Implantación
9 %	11 %	15 %	43 %	16 %	6 %
234.5985h	286.7315h	390.9975h	1120.8595h	417.064h	156.339h

Tabla 3.5: Estimaciones por fase del ciclo de vida.

---

# Capítulo 4

## Modelado de datos

En este capítulo se expondrá el modelo de datos que resultó de estudiar los requerimientos para el proyecto de cuestionarios web. Como se menciona en el [Capítulo 2](#) uno de los objetivos que se tienen para el prototipo es diseñar una base de datos relacional que defina la estructura física de datos que utilizará el sistema a partir de un modelo conceptual. Con relación a esto, dentro de la metodológica RUP se encuentra la disciplina de análisis y diseño, la cual propone artefactos para lograr un resultado en particular, por lo que para nuestro caso se optó por realizar el modelo entidad relación teniendo presente los requisitos establecidos para el sistema, así como las particularidades del entorno tecnológico, de forma que se consiga un acceso eficiente de los datos.

### 4.1. Bases de datos relacionales

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. Entre las principales características de los sistemas de base de datos relacionales podemos mencionar:

- Redundancia mínima.
- Integridad de los datos.
- Respaldo y recuperación.

Para obtener una manipulación de datos eficiente es importante el diseño y uso de

un modelo de datos, el cual es una abstracción que típicamente permite describir las estructuras de los datos, es decir, el tipo de los datos que hay en la base y la forma en que se relacionan. A continuación se hablará del diseño del modelo para la gestión de los datos en el prototipo de cuestionarios en línea.

### 4.1.1. Diseño de la base datos

En la construcción del modelo conceptual se utilizó la herramienta Oracle SQL Developer Data Modeler 4.0.3. Y la notación empleada en este proyecto no corresponde a la convencional de Peter Chen (1976), sino la planteada por Case\*Method de Oracle, que esta basada en la propuesta por Richard Barker del año 1981, ya que es mucho más compacta y rica en su semántica, lo que ayuda a comunicar relaciones complejas en un formato fácil de usar y entender, además de que esta notación sea la más utilizada en el contexto internacional por sus diferentes beneficios en la interpretación y “lectura” del Modelo. Teniendo estos dos elementos en congruencia se obtuvo el modelo para el sistema. Cabe aclarar que aunque se ocupó una herramienta propia de Oracle la base de datos se implementó en PostgreSQL: “PostgreSQL 9.5.0, compiled by Visual C++ build 1800, 64-bit”.

El eje base del sistema es la creación y edición de un cuestionario y para su construcción fue necesario el diseño de un modelo de datos que diera soporte a los procesos involucrados en cada fase de construcción de este mismo, como por ejemplo: la creación de preguntas con textos asociados, el envío de observaciones por parte de un colaborador, la aplicación de un cuestionario, entre otros.

Con la finalidad de facilitar la comprensión del modelo, éste se presentará por partes<sup>1</sup>. Por simplicidad, en los diagramas solo se muestran los atributos de los objetos del negocio que ayudan a comprender el modelo, el resto de elementos como el tipo de dato se omiten.

Además, conforme a los requisitos para el diseño de la base de datos, el modelo conceptual incluye el caso de las preguntas abiertas; aunque esto último no pertenece al alcance u objetivo del proyecto actual, se trabajó en ello para describir los elementos de la realidad que intervienen en todo el contexto del problema y así permitir observar patrones de datos y usos potenciales de los datos, teniendo a una mayor claridad para un trabajo futuro.

---

<sup>1</sup>Para observar el modelo resultante puede remitirse al Anexo B.

### 4.1.2. Entidades y relaciones

Para la construcción del modelo conceptual (modelo entidad/relación) del proyecto, se tomaron aclaraciones que hizo el cliente, como por ejemplo: que un cuestionario puede tener varios colaboradores, que un tema puede tener subtemas o que una aplicación puede estar asociada a varias hojas de respuestas. Teniendo estas y otras consideraciones, se comenzó por encontrar entidades y a definir las relaciones entre ellas. Las entidades encontradas y su semántica se listan y se definen a continuación en la tabla 4.1.

<b>Cuestionario:</b>	Entidad donde se almacenarán los títulos, descripción y estatus de los cuestionarios.
<b>Sección:</b>	Un cuestionario constará de una sección principal y se le podrán añadir más secciones de ser necesario; por ejemplo, un cuestionario de matemáticas podría tener secciones como: aritmética, álgebra, geometría, etcétera.
<b>Colaborador:</b>	Contendrá los identificadores (ids) de los usuarios y su función para un determinado cuestionario; como por ejemplo, la función de editor o de colaborador.
<b>Aplicación:</b>	Entidad donde se colocarán datos como la fecha, hora y una contraseña opcional, para que se aplique un cuestionario determinado.
<b>Pregunta:</b>	Aquí es donde se almacenarán las preguntas que junto a otra entidades (respuestas, textos asociados y temas) forman el baúl del preguntas.
<b>Cuestionario Pregunta:</b>	Entidad que tendrá las relaciones entre una sección y una pregunta para un cuestionario, debido a que una sección puede tener muchas preguntas y una pregunta (al estar en el baúl de preguntas) puede estar en varias secciones, como se tiene una relación muchos a muchos, se “rompió” la relación y surgió dicha entidad.
<b>Texto Asociado:</b>	Textos almacenados para dar información extra a una pregunta.
<b>Respuesta POM:</b>	Entidad que contiene las respuestas de opción múltiple para un cuestionario.
<b>Hoja de respuesta:</b>	Contendrá las respuestas que un participante haya elegido durante la aplicación de un cuestionario.
<b>Tema:</b>	Es donde se guardarán temas y subtemas que puedan tener las preguntas.

Tabla 4.1: Entidades principales del modelo de datos.

En breve se explican las relaciones más relevantes que existen en el diagrama de entidad relación.

1) La primera relación a explicar será la que se tiene entre Sección, Cuestionario-Pregunta y Pregunta. En un principio se tenía una relación de muchos a muchos entre Sección y Pregunta como en la figura 4.1, es decir, *una sección tiene varias preguntas y una pregunta puede estar en varias secciones*, por lo que esta relación se tuvo que descomponer como se observa en la figura 4.2. El modelo resultante contempla ahora tres entidades: dos entidades “fuertes” y 1 entidad “débil”.

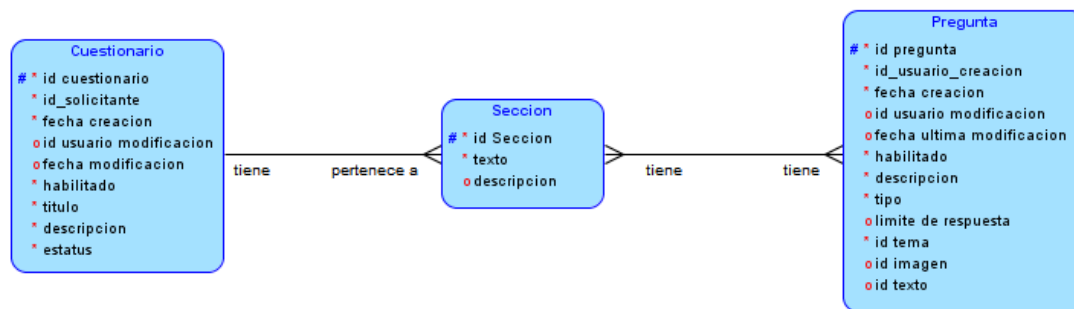


Figura 4.1: Relación muchos a muchos entre sección y pregunta.

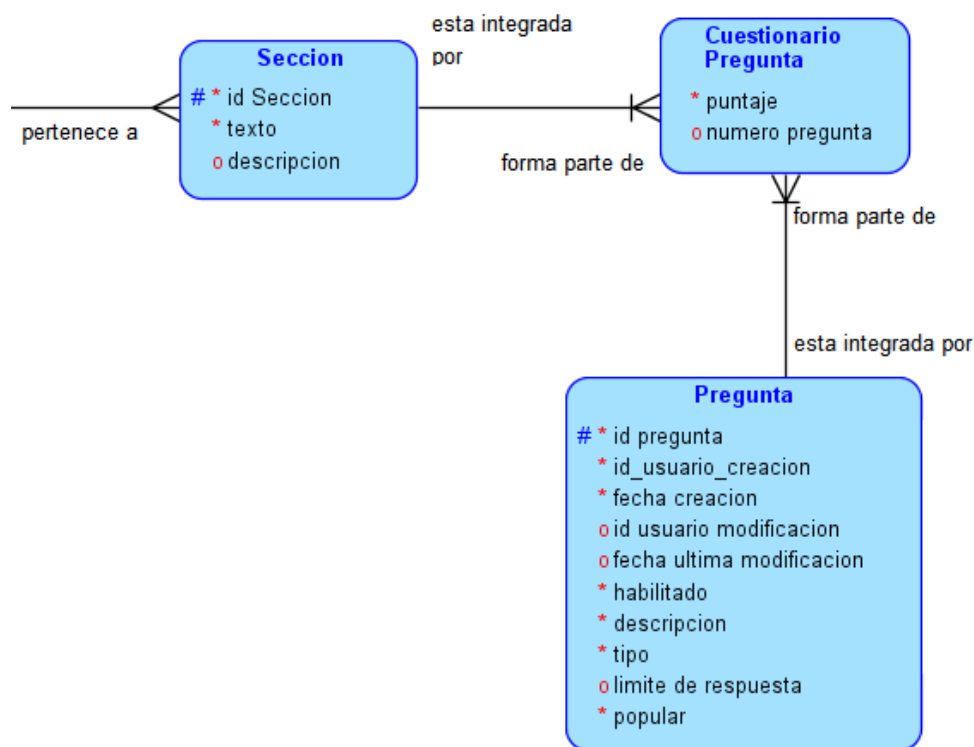


Figura 4.2: Descomposición de entidades sección y pregunta.

De la figura 4.2, se obtiene que *una sección esta integrada por una entidad “débil” llamada Cuestionario-Pregunta, la cual forma parte de una sección y por el otro lado*

tenemos que *una pregunta está integrada por la misma entidad “débil” y que a su vez esta última forma parte de una pregunta*. Ambas entidades (sección y pregunta) forman las entidades “fuertes”.

- 2) En segunda instancia tenemos la relación entre cuestionario y sección, en ella podemos observar que *un cuestionario tiene varias secciones y una sección pertenece a solo un cuestionario* (figura 4.3).

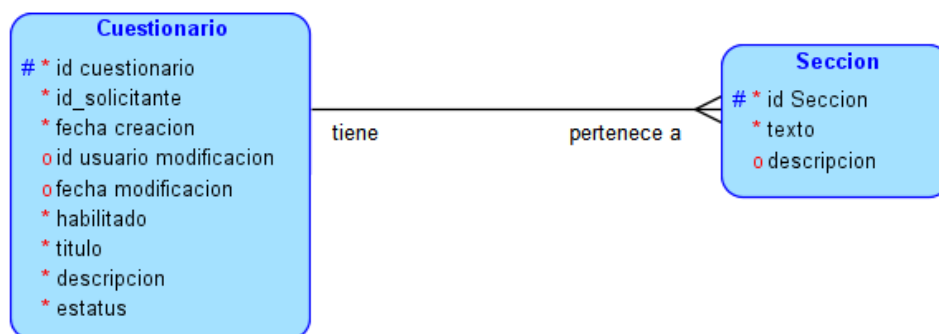


Figura 4.3: Relación entre cuestionario y pregunta.

- 3) La tercera relación a definir es la que se encuentra entre pregunta y tema. Entre estas entidades da la situación de que *para cada pregunta se tiene un tema y cada tema tiene que estar relacionado con cada pregunta* (figura 4.4).

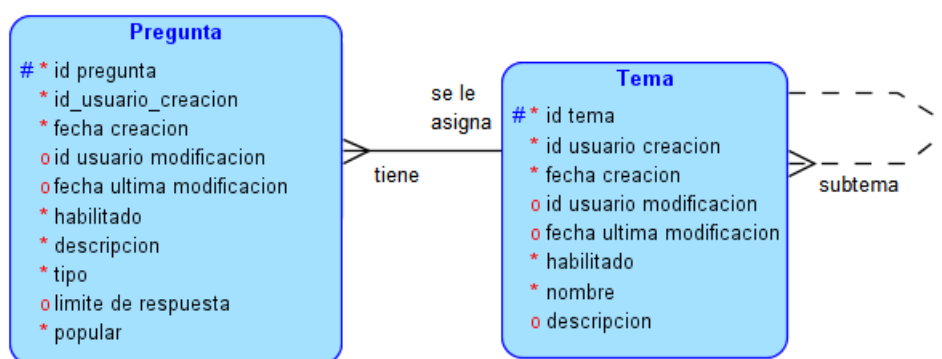


Figura 4.4: Relación entre tema y pregunta.

- 4) Semántica entre pregunta y texto asociado. Estas entidades tienen una relación de opcionalidad ya que *cada pregunta puede tener un texto asociado* y a su vez *cada texto asociado puede estar asignado a una o mas preguntas* (figura 4.5).

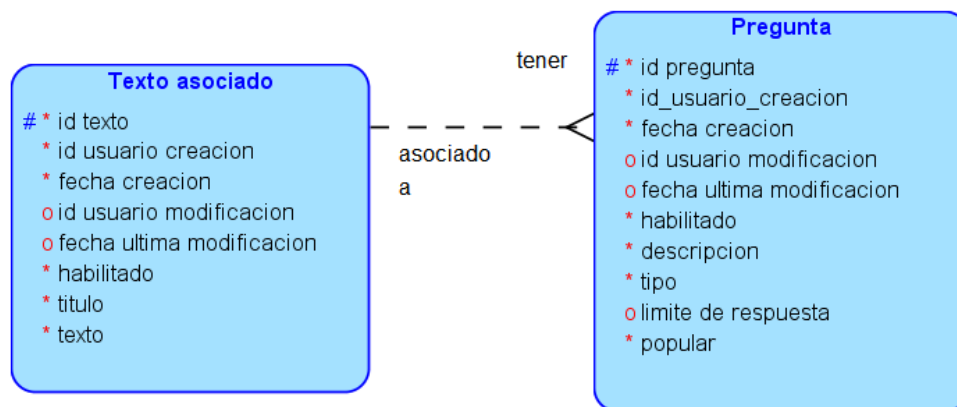


Figura 4.5: Relación entre texto asociado y pregunta.

- 5) A continuación en la figura 4.6 se expone la relación entre pregunta y respuesta POM-simple.

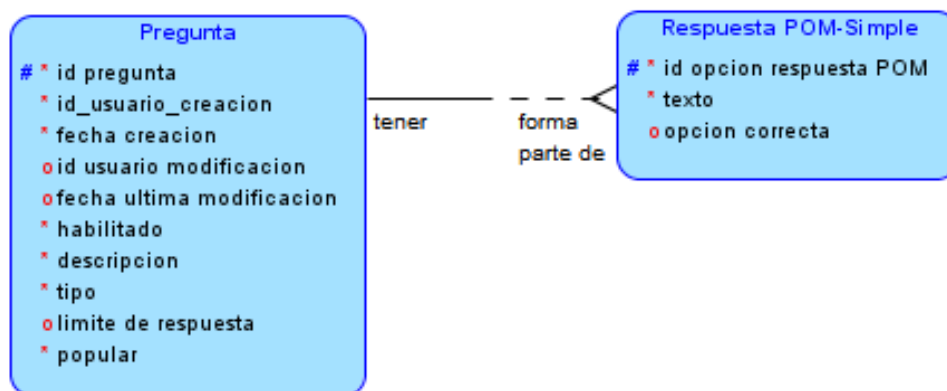


Figura 4.6: Relación entre respuesta y pregunta.

Con este par de entidades tenemos que *una pregunta puede tener varias respuestas* y que *cada respuesta está relacionada con una sola pregunta*.

- 6) La sexta descripción le pertenece al par de entidades llamadas observación y cuestionario 4.7.

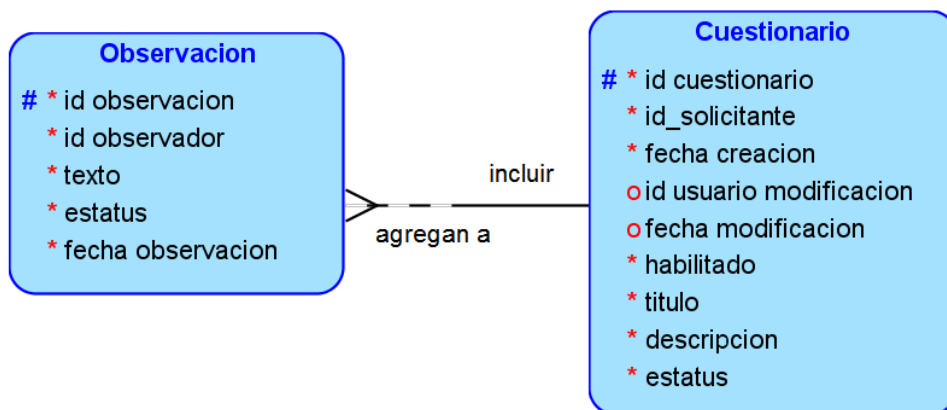


Figura 4.7: Relación entre cuestionario y observación.

De la figura 4.7 se puede decir que *a un cuestionario se le pueden incluir observaciones* y que *una observación se le agrega a un cuestionario*

- 7) La siguiente relación se encuentra entre cuestionario y colaborador (figura 4.8), entre ellas se puede entender lo siguiente: *un cuestionario tiene uno o más colaboradores* y *un colaborador está asociado a un solo cuestionario*.

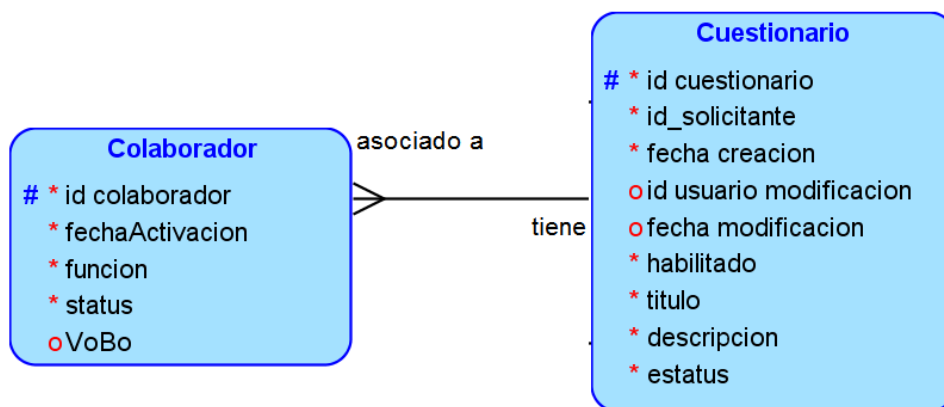


Figura 4.8: Relación entre cuestionario y colaborador.

- 8) Otra relación importante es la que existe entre cuestionario y aplicación que se muestra en figura 4.9. Aquí *un cuestionario puede tener una o varias aplicaciones* y por otro lado *una aplicación solo estará ligada a un cuestionario*.

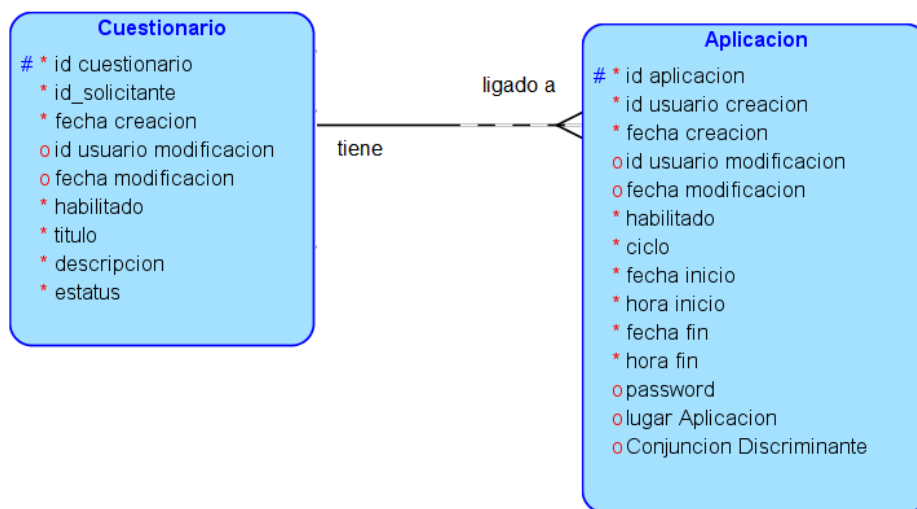


Figura 4.9: Relación entre cuestionario y aplicación.

- 9) La relación entre aplicación y hoja de respuesta dice: *una aplicación podrá tener una o varias hojas de respuestas y una hoja de respuesta pertenece a uno y sólo un cuestionario* (figura 4.10).

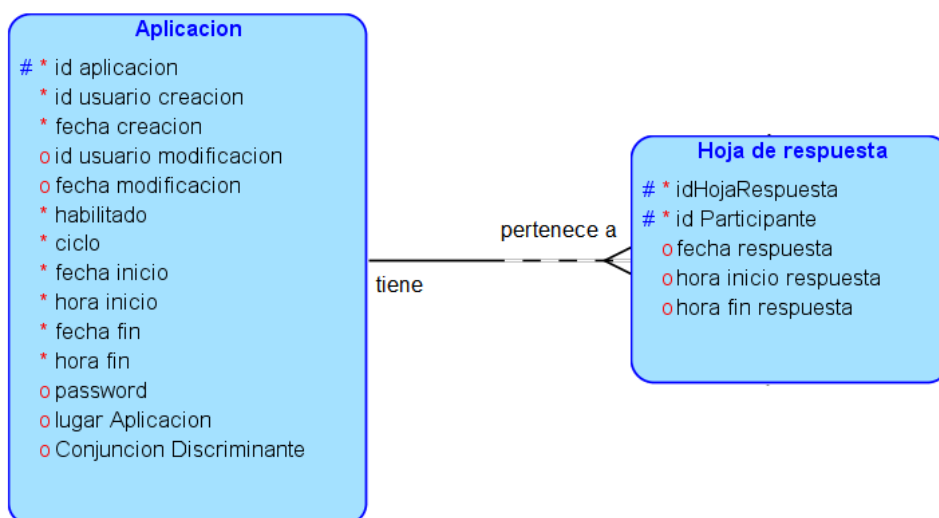


Figura 4.10: Relación entre hoja de respuesta y aplicación.

- 10) La antepenúltima relación se encuentra entre la entidad hoja de respuesta y la entidad donde se almacenaran todas la repuestas que un participante haga. Es decir, el modelo de datos Resp POM simple, (figura 4.11) donde: *una hoja de repuesta puede tener varias respuestas de opción múltiple* y que *una respuesta de opción múltiple puede o no estar en hoja de respuesta*.

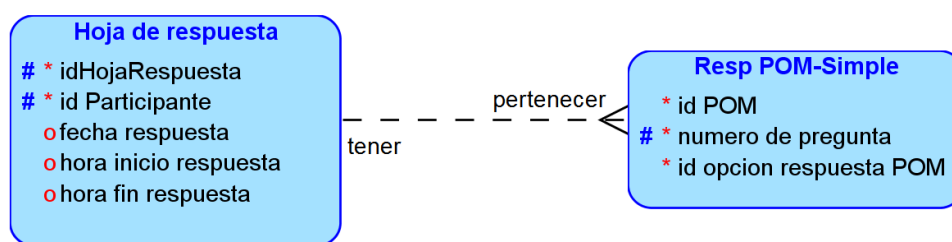


Figura 4.11: Relación entre hoja de respuesta y respuesta POM de usuario.

- 11) Por último, se detallará las relación recursiva o de “cola de cochino” que se tuvo en la entidad tema (figura 4.12).

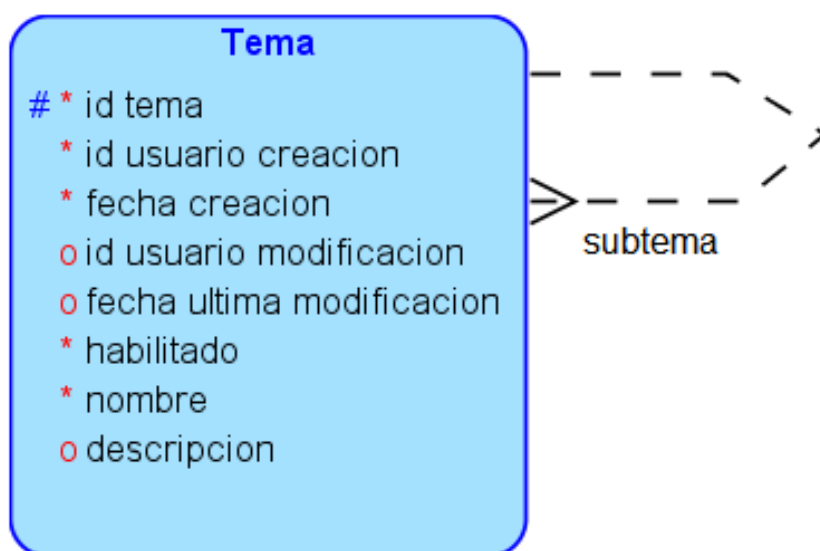


Figura 4.12: Relación entre hoja de respuesta y aplicación.

En los requisitos se observó que un pregunta debería tener un tema, como se observó en la figura 4.4, pero además de los temas, también se requería que existieran subtemas; por lo tanto, se notó que estos dos conceptos son en sí son temas, con la diferencia de que se tenía que recurrir a la misma entidad para generar un tema “padre” y los subtemas serían como los “hijos” del tema padre. Entonces se optó por hacer una relación recursiva o de cola de puerco para resolver este caso, quedando de la siguiente manera:

*“Cada tema puede tener varios subtemas”*

y

*“Cada subtema puede estar asociado a un tema”*

- 12) Para obtener una comparación entre las respuestas de un participante y la respuesta correcta que un editor colocó a una pregunta, se tiene la relación entre las tablas de la figura 4.13, en donde se tiene lo siguiente: “Cada respuesta de usuario puede estar asociada a una respuesta de una pregunta POM.” y “Cada respuesta de una pregunta POM debe existir como una respuesta de un participante”.

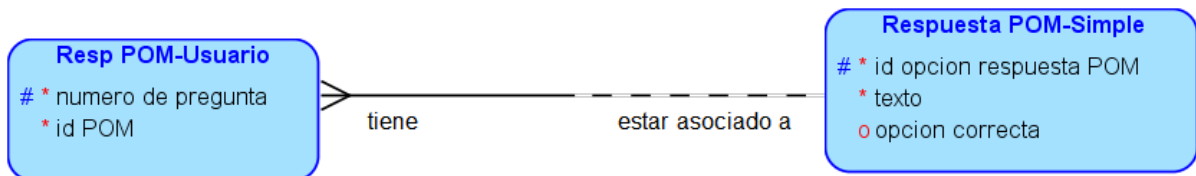


Figura 4.13: Relación entre las respuesta del participante y las respuestas de una pregunta.

Por último, es importante mencionar que la descripción de cada atributo se encuentra en el Anexo C, que es el diccionario de datos para el modelo entidad relación descrito.

---

# Capítulo 5

## Framework Laravel

Como se dijo en el [Capítulo 2](#), para la construcción del prototipo de cuestionarios web se pensó en una estructura de soporte que agilice y facilite el desarrollo del proyecto, además también se menciona que el RUP está centrado en la arquitectura, sin embargo no existe un marco único que cubra todos los aspectos del sistema, es decir, existen múltiples modelos y vistas que pueden definir una arquitectura de software. En este sentido se hará una descripción general, en este capítulo, acerca del framework Laravel, mediante el cual se utiliza la arquitectura propuesta por este marco de trabajo, de esta forma nuestro diseño e implementación del prototipo del sistema de cuestionarios web, estará influenciado por la estructura propuesta por dicho framework.

### 5.1. Introducción

Según [McCool \(2012, p. 3\)](#) Laravel es “un framework MVC<sup>1</sup> para desarrollo web, de código libre, escrito en el lenguaje de programación PHP. Está diseñado para mejorar la calidad del software reduciendo costos de desarrollo inicial y costos de mantenimiento sobre la marcha, mejorando la experiencia de trabajo con las aplicaciones ya que proporciona una sintaxis para expresiones muy limpia y un conjunto básico de funcionalidad que ahorrará horas durante el tiempo de implementación”.

En suma, Laravel esta pensado para desarrollar aplicaciones de una manera rápida con sintaxis elegante y expresiva. La curva de aprendizaje es bastante rápida; además de ser un conjunto de herramientas que facilitan el desarrollo como Eloquent ORM, migraciones,

---

<sup>1</sup>Más adelante se hablará de este patrón de diseño.

manejado por Composer, entre otras.

Actualmente, Laravel tiene una gran comunidad de programadores desde su creación en 2011. Esto se puede observar en una gráfica de *Google tendencias*<sup>2</sup> donde se realizó una búsqueda comparativa desde octubre del 2010 a octubre del año 2016, para encontrar el mejor marco de trabajo en PHP hasta el momento. Como se observa la figura 5.1 desde el 2014 Laravel (en azul) cada vez gana más popularidad y se ha convertido en el framework por excelencia de muchos programadores web en la actualidad.

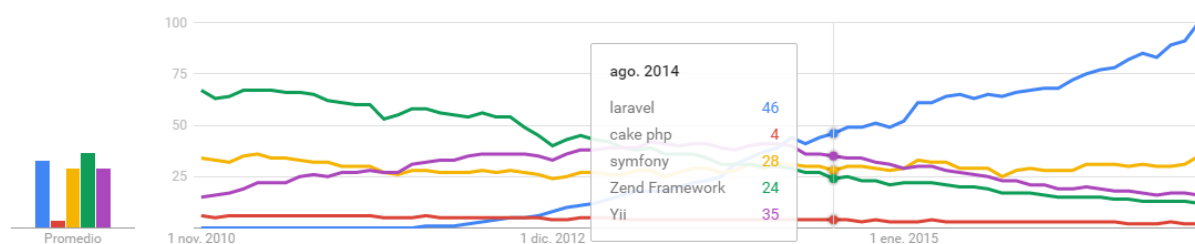


Figura 5.1: Comparativa entre frameworks PHP.

A continuación describiremos de manera general qué es un framework web sus objetivos y sus características, para después ahondar en el diseño y la arquitectura que definen a Laravel. Cabe mencionar que la versión de Laravel para el desarrollo de este proyecto fue la versión 5.1 LTS (Long Term Support o Soporte a largo plazo).

## 5.2. Objetivos de los framework web

Como menciona Saavedra (2008, p. 25) entre los principales objetivos que persiguen los frameworks web están:

- Acelerar el proceso de desarrollo.
- Reutilizar código ya existente.
- Promover buenas prácticas de desarrollo como el uso de patrones de diseño.
- Disminuir el esfuerzo en el desarrollo.
- Tener como aliado a las metodologías de desarrollo Ágiles (XP, Scrum, AD, etcétera).

<sup>2</sup>Tendencias de búsqueda de Google es una herramienta de Google Labs que muestra los términos de búsqueda más populares del pasado reciente.

- Mejorar la organización de carpetas y estructura del proyecto, para promover una mantenibilidad fácil del código.

### 5.3. Características de un framework web

Según Gutiérrez (2014, p. 3 – 4) entre las principales características de un framework son:

- **Abstracción de URLs y sesiones.** No es necesario manejar directamente las URLs ni las sesiones, ya que el framework se encarga de hacerlo.
- **Acceso a Datos.** Los frameworks cuentan con todas las herramientas e interfaces necesarias.
- **Controladores.** Los frameworks son los encargados de crear sus propios controladores estos suelen ser fáciles, adaptables a los proyectos.
- **Autenticación y control.** Mantiene la seguridad tanto para la autenticación a través de un login y un password y para el acceso a determinados usuarios.
- **Separación entre diseño y contenido.** Tienen herramientas e interfaces necesarias para su diseño y contenido.

### 5.4. Diseño

Laravel está basado en el patrón de diseño que es más utilizado para el desarrollo de aplicaciones web, conocido como modelo vista controlador o MVC (por sus siglas en inglés model view controller). MVC es un patrón de diseño que separa los datos, la lógica de negocios y las interfaces de usuario. Fue utilizado para construir las interfaces de usuario en el lenguaje Smalltalk-80. En la figura 5.2 se ilustra un diagrama que muestra el concepto de este patrón.

A grandes rasgos se podría describir de la siguiente manera:

Entrada → Procesamiento → Salida  
Controlador → Modelo → Vista

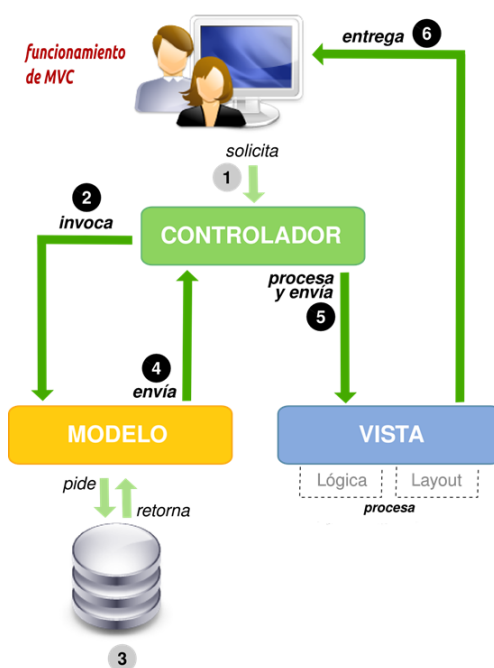


Figura 5.2: Patrón de diseño MVC.

## 5.5. Arquitectura

El concepto de arquitectura en ingeniería de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo.

En arquitectura de software existe un concepto llamado patrón de arquitectura, que en comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor. Como define Tahuiton (2011, p. 43) “un patrón de arquitectura encapsula los elementos y las relaciones que existen entre ellos permitiendo abstraer el comportamiento para que sea posible tener una configuración de componentes que satisfaga ciertas necesidades”. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones.

En términos muy simples la arquitectura de un software puede verse como la estructura interna del proyecto a construir; podemos ver a continuación la estructura de directorios de Laravel que forman parte de la arquitectura propia del framework. La estructura de los directorios de un proyecto de Laravel es la siguiente:

- **app**: donde se encuentran las clases de modelos, controladores, rutas, entre otras.
- **bootstrap**: contiene configuraciones del framework y cache.
- **config**: donde se ubican los archivos de configuración del proyecto.
- **database**: donde creamos las migraciones, seeders y model factories, es decir, lo relacionado a la base de datos.
- **public**: contiene todo lo público de la aplicación al que el navegador tendrá acceso: hojas de estilo, javascript, imágenes, fuentes, favicon. En resumidas cuentas son los conocidos como assets.
- **resources**: se encuentran archivos de idioma (lang), assets y vistas. Cabe aclarar que la carpeta assets se utiliza para almacenar los fuentes de los assets tipo less o sass que se tendrían que compilar para generar las hojas de estilo públicas. No es necesario usar esta carpeta ya que podemos escribir directamente las hojas de estilo dentro de la carpeta public.
- **storage**: archivos temporales del framework.
- **tests**: para ubicar todas las pruebas que tendrá la aplicación.
- **vendor**: donde se alojarán los componentes de terceros manejados por Composer.

## 5.6. Contexto tecnológico

A continuación se describirán las herramientas principales que se utilizaron y que se relacionan con el framework Laravel. Estas herramientas darán un contexto tecnológico previo que forma parte del Backend de la aplicación.

### 5.6.1. Composer

Laravel utiliza Composer<sup>3</sup> para gestionar sus dependencias. Así que antes de usar Laravel, para la construcción del sistema, Composer tuvo que ser instalado en la computadora.

Composer es una herramienta de línea de comandos (CLI por sus siglas en inglés de Command-Line Interface) para la gestión de dependencias en el lenguaje de programación PHP.

---

<sup>3</sup><https://getcomposer.org/>

Con Composer las dependencias siempre se instalan de manera local para cualquier proyecto en un directorio por defecto (normalmente es `/vendor`)<sup>4</sup>.

En el caso del sistema de cuestionarios si deseo agregar o actualizar un componente en el proyecto, por ejemplo un generador de PDFs, esto se puede hacer a través de Composer, mediante el uso de ciertos comandos. Para el caso particular del proyecto de cuestionarios web se instalaron ciertos componentes que ayudan al funcionamiento del sistema, entre los cuales descantan:

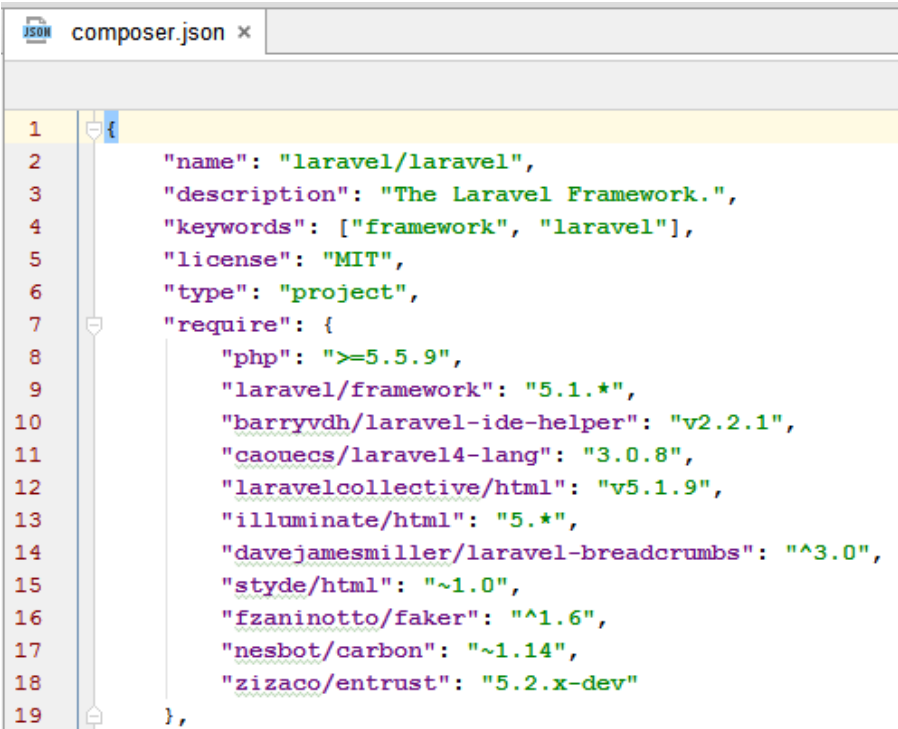
- **nesbot/carbon**: que permite crear y dar formato a fechas con unos sencillos métodos,
- **fzaninotto/faker**: una biblioteca de PHP que genera datos falsos, en la base de datos para hacer pruebas (más adelante se hablará acerca de esto),
- **davejamesmiller/laravel-breadcrumbs**: La “miga de pan” (del inglés breadcrumb), se realiza para tener una guía de la navegación de nuestra web a través de la categorías y diferentes secciones de esta.
- **styde/html**: permite generar campos de formularios y otros elementos HTML desde las plantillas de Blade.

Estos y otros componentes se instalaron en el proyecto de cuestionarios web, la lista completa de biblioteca del sistema se puede observar en la figura 5.3. Esta lista es un archivo en formato JSON que usa Composer para saber que componentes se han instalado y también saber si hay alguna instalación nueva para hacer una actualización.

Con Laravel la mayor parte de la aplicación se encuentra en el directorio de aplicación (**app/**). En este directorio existen clases que contienen espacio de nombres (namespaces) para la aplicación y se cargan automáticamente por Composer utilizando el estándar de carga automática PSR-4. PSR-4 es una especificación para la auto carga de clases desde la ruta de los archivos. La ventaja de usar estándares como el PSR-4 es que permiten a Composer cargar las clases automáticamente, lo que hace ordenar y estructurar mejor las clases.

---

<sup>4</sup>Composer es capaz de instalar y actualizar las bibliotecas que requiere un proyecto con las versiones que se necesiten. Además si las bibliotecas dependen de otras, Composer también es capaz de resolver eso y descargar todo lo necesario.



```
1 {
2     "name": "laravel/laravel",
3     "description": "The Laravel Framework.",
4     "keywords": ["framework", "laravel"],
5     "license": "MIT",
6     "type": "project",
7     "require": {
8         "php": ">=5.5.9",
9         "laravel/framework": "5.1.*",
10        "barryvdh/laravel-ide-helper": "v2.2.1",
11        "caouecs/laravel4-lang": "3.0.8",
12        "laravelcollective/html": "v5.1.9",
13        "illuminate/html": "5.*",
14        "davejamesmiller/laravel-breadcrumbs": "^3.0",
15        "styde/html": "~1.0",
16        "fzaninotto/faker": "^1.6",
17        "nesbot/carbon": "~1.14",
18        "zizaco/entrust": "5.2.x-dev"
19    },
```

Figura 5.3: Lista de componentes instalados para el proyecto web.

## 5.6.2. Artisan

Otra de las herramientas importantes para el proyecto se llama Artisan. Esta es una interfaz de línea de comandos, la cual es un medio para la interacción con la aplicación donde uno mismo puede dar instrucciones en forma de línea de texto simple.

Una de las grandes ventajas de esta herramienta es que tiene un conjunto de comandos dedicado a generadores, es decir, que permiten crear elementos como controladores, middlewares<sup>5</sup>, modelos, migraciones, entre otros. Una muestra del listado completo de los comandos disponibles en Artisan los podemos encontrar tecleando *php artisan list*, desde directorio raíz de un proyecto de Laravel (figura 5.4).

Cada uno de esos generadores tiene sus propias opciones y se pueden revisar con el comando *help*. Por ejemplo, existe un comando que se utilizó repetidas veces en el desarrollo del sistema y es para ver el listado completo de rutas del proyecto, el cual es: *php artisan route:list*. Como resultado nos mostrará un listado de todas las rutas de

---

<sup>5</sup>Un middleware es un mecanismo que se utiliza para filtrar las peticiones HTTP en una aplicación. Están disponibles desde la versión 5 de Laravel y permiten agregar capas adicionales a la lógica de la aplicación. El ejemplo más común de un Middleware es el de autenticación. Este serviría, por ejemplo, en un sitio en donde es necesario iniciar sesión y tener ciertos privilegios para poder ver el contenido de determinadas páginas.

```
$ artisan list
Laravel Framework version 5.1.45 (LTS)

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
                       --ansi                Force ANSI output
                       --no-ansi            Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
                       --env[=ENV]        The environment the command should run under.
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for
more verbose output and 3 for debug

Available commands:
  clear-compiled      Remove the compiled class file
  down               Put the application into maintenance mode
  env                Display the current framework environment
  help              Displays help for a command
  inspire           Display an inspiring quote
  list              Lists commands
  migrate           Run the database migrations
  optimize          Optimize the framework for better performance
  serve            Serve the application on the PHP development server
  tinker           Interact with your application
  up               Bring the application out of maintenance mode
  app
  app:name         Set the application namespace
  auth
  auth:clear-resets Flush expired password reset tokens
  cache
  cache:clear      Flush the application cache
  cache:table      Create a migration for the cache database table
  config
  config:cache     Create a cache file for faster configuration loading
  config:clear     Remove the configuration cache file
  db
  db:seed         Seed the database with records
  entrust
  entrust:migration Creates a migration following the Entrust specifications.
  event
  event:generate   Generate the missing events and listeners based on registration
  handler
  handler:command  Create a new command handler class
  handler:event    Create a new event handler class
  ide-helper
```

Figura 5.4: Algunos comandos de Artisan.

la aplicación con la información sobre el método HTTP, la URI, la acción, el nombre y los middleware definidos para cada ruta, con lo cual, entre otras cosas, podemos verificar que todas las rutas de la aplicación están bien definidas (figura 5.5).

En las secciones siguientes se mencionarán más comandos de Artisan utilizados a lo largo del desarrollo de la aplicación.

### 5.6.3. Migraciones

Las migraciones son una serie de archivos y clases en PHP que actúan como un control de versiones de base de datos, tanto para crear las tablas al inicio del desarrollo de nuestra aplicación, como para realizar cambios como agregar o eliminar una tabla, agregar o

Method	URI	Name	Action
GET   HEAD	/	home	aprendiendo\Http\Controllers\HomeController@index
GET   HEAD	about	nosotros	Closure
POST	auth/login/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\AuthController@postLogin
GET   HEAD	auth/login/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\AuthController@getLogin
GET   HEAD	auth/logout/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\AuthController@getLogout
GET   HEAD	auth/register/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\AuthController@getRegister
POST	auth/register/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\AuthController@postRegister
GET   HEAD   POST   PUT   PATCH   DELETE	auth/{_missing}		aprendiendo\Http\Controllers\Auth\AuthController@missing
GET   HEAD	banco/pregunta	bankq	aprendiendo\Http\Controllers\PreguntaController@index
GET   HEAD	banco/pregunta/before/{id?}/<sc?>	banco_pregunta.before	aprendiendo\Http\Controllers\PreguntaController@before
POST	banco/pregunta/pon	pon	aprendiendo\Http\Controllers\PreguntaController@create
POST	banco/pregunta/store	banco_pregunta.store	aprendiendo\Http\Controllers\PreguntaController@store
GET   HEAD	banco/tema	banco_tema.index	aprendiendo\Http\Controllers\TemaController@index
GET   HEAD	banco/tema/create	banco_tema.create	aprendiendo\Http\Controllers\TemaController@create
GET   HEAD	banco/tema/{tema}	banco_tema.show	aprendiendo\Http\Controllers\TemaController@show
DELETE	banco/tema/{tema}	banco_tema.destroy	aprendiendo\Http\Controllers\TemaController@destroy
PATCH	banco/tema/{tema}		aprendiendo\Http\Controllers\TemaController@update
PUT	banco/tema/{tema}	banco_tema.update	aprendiendo\Http\Controllers\TemaController@update
GET   HEAD	banco/tema/{tema}/edit	banco_tema.edit	aprendiendo\Http\Controllers\TemaController@edit
GET   HEAD	contact	contacto	Closure
GET   HEAD	questionario/crear	crear	aprendiendo\Http\Controllers\QuestionarioController@create
GET   HEAD	questionario/editar/{id}/<idsc?>/<idp?>	editarq	aprendiendo\Http\Controllers\QuestionarioController@edit
POST	questionario/guardar	questionario.save	aprendiendo\Http\Controllers\QuestionarioController@store
GET   HEAD	questionario/mostrar	showq	aprendiendo\Http\Controllers\QuestionarioController@show
PUT	questionario/update/{id}	questionario.generar	aprendiendo\Http\Controllers\QuestionarioController@update
GET   HEAD	home	home	aprendiendo\Http\Controllers\HomeController@index
GET   HEAD	inicio	home	aprendiendo\Http\Controllers\HomeController@index
POST	password/email/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\PasswordController@postEmail
GET   HEAD	password/email/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\PasswordController@getEmail
POST	password/reset/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\PasswordController@postReset
POST	password/reset/{one?}/{two?}/{three?}/{four?}/{five?}		aprendiendo\Http\Controllers\Auth\PasswordController@reset
GET   HEAD   POST   PUT   PATCH   DELETE	password/{_missing}		aprendiendo\Http\Controllers\Auth\PasswordController@missing
GET   HEAD	permisos	aprendiendo\Http\Controllers\PermisosController@index	
GET   HEAD	permisos/asignar		aprendiendo\Http\Controllers\PermisosController@asignar
GET   HEAD	permisos/desasignar		aprendiendo\Http\Controllers\PermisosController@desasignar
POST	roles	roles.store	aprendiendo\Http\Controllers\RolesController@store
GET   HEAD	roles	roles.index	aprendiendo\Http\Controllers\RolesController@index
GET   HEAD	roles/create	roles.create	aprendiendo\Http\Controllers\RolesController@create
GET   HEAD	roles/{roles}	roles.show	aprendiendo\Http\Controllers\RolesController@show
PATCH	roles/{roles}		aprendiendo\Http\Controllers\RolesController@update
DELETE	roles/{roles}	roles.destroy	aprendiendo\Http\Controllers\RolesController@destroy
PUT	roles/{roles}	roles.update	aprendiendo\Http\Controllers\RolesController@update
GET   HEAD	roles/{roles}/edit	roles.edit	aprendiendo\Http\Controllers\RolesController@edit
GET   HEAD	users	users.index	aprendiendo\Http\Controllers\UserController@index
POST	users/create	users.store	aprendiendo\Http\Controllers\UserController@store
GET   HEAD	users/{users}	users.create	aprendiendo\Http\Controllers\UserController@create
PUT	users/{users}	users.show	aprendiendo\Http\Controllers\UserController@show
PATCH	users/{users}	users.update	aprendiendo\Http\Controllers\UserController@update
DELETE	users/{users}	users.destroy	aprendiendo\Http\Controllers\UserController@destroy
GET   HEAD	users/{users}/edit	users.edit	aprendiendo\Http\Controllers\UserController@edit

Figura 5.5: Algunas rutas para el proyecto cuestionarios web.

eliminar una columna y así sucesivamente, a medida que se fue desarrollando el proyecto.

Las migraciones permiten definir las tablas con programación orientada a objetos en vez de SQL, lo cual proporciona portabilidad a los diferentes motores que soporta Laravel: MySQL, Postgres, SQLite, y SQL Server. Para establecer el manejador de datos a usar se configura en el archivo “.env” de la raíz del proyecto. Es importante recordar que las ventajas ofrecidas por las migraciones dentro de Laravel son totalmente opcionales, de esta forma se puede prescindir de ellas y crear las tablas con herramientas como PHPMyAdmin o pgAdmin.

En la figura 5.6 se muestra la creación de una tabla sin usar SQL explícito, desde la línea 16 a la 22 se están definen las columnas y su respectivo tipo de dato para la tabla temas. En la línea 16 se define la columna para el identificador, denotada por id<sup>6</sup>, la cual se irá auto incrementando conforme se vayan agregado datos a la tabla; en la línea 18 se define la columna *nombre* que será de tipo character con tamaño máximo de 128 caracteres; en la línea 19 se definen las llaves foráneas; como se mencionó en la sección 4.1.2 del capítulo 4 la tabla Temas tenía una referencia con sigo mismo, es decir, una relación recursiva, por lo tanto en la línea 22 se observa como se definió la relación escribiéndola fácilmente por medio de lenguaje de programación orientado a objetos a lo que se conoce como mapeo de objeto relacional. En la siguiente sección se hablará un poco más a fondo al respecto. Finalmente este pequeño código escribiría el equivalente en instrucciones SQL para crear

<sup>6</sup>Con Laravel se asumirá que cada tabla tiene una columna de clave principal denominada id.

la tabla temas del sistema. En la figura 5.7, se puede ver el equivalente en líneas de SQL para la creación de la tabla ya mencionada.

```
6 class CreateTemasTable extends Migration
7 {
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13     public function up()
14     {
15         Schema::create('temas', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('nombre', 128);
18             $table->text('descripcion')->nullable();
19             $table->integer('id_padre')->unsigned()->nullable();
20             $table->timestamps();
21             $table->softDeletes();
22             $table->foreign('id_padre')->nullable()
23                 ->references('id')
24                 ->on('temas');
25         });
26     }
27 }
```

Figura 5.6: Creación de la tabla Temas de la aplicación cuestionarios.

```
CREATE TABLE cuestionarios.temas
(
    id serial NOT NULL,
    nombre character varying(128) NOT NULL,
    descripcion text,
    id_padre integer,
    created_at timestamp(0) without time zone NOT NULL,
    updated_at timestamp(0) without time zone NOT NULL,
    deleted_at timestamp(0) without time zone,
    CONSTRAINT temas_pkey PRIMARY KEY (id),
    CONSTRAINT temas_id_padre_foreign FOREIGN KEY (id_padre)
        REFERENCES cuestionarios.temas (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
```

Figura 5.7: SQL equivalente para la creación de la tabla Temas.

### 5.6.4. Eloquent

Eloquent es el mapeo de objeto relacional u ORM<sup>7</sup> (por su siglas en inglés Object Relational Mapping) que incluye Laravel para manejar, de una forma fácil y sencilla, los procesos correspondientes al manejo de bases de datos en el proyecto, gracias a las funciones que provee se puede realizar complejas consultas y peticiones de base de datos sin escribir una sola línea de código SQL. Se puede decir que el ORM proporciona una capa intermedia y abstrae al programador de la base de datos para que este se centre en el desarrollo de la aplicación.

La figura 5.8 da muestra de varias consultas realizadas para crear la vista de un cuestionario. Se puede observar que en la línea 152, se está solicitando el id de el usuario autenticado para identificar posteriormente el tipo de colaborador; más adelante en la línea 153 se extraen la secciones para el cuestionario actual, después la unión entre las tablas de cuestionario, cuestionario-pregunta y pregunta (líneas 155 a la 159), ya que existe una relación entre ellas como se dijo en la sección 4.1.2 del capítulo 4. Finalmente, en la línea 162, se observa que para la unión resultante se extraen la respuestas que pertenecen a cada pregunta.

```
147 * @return \Illuminate\Http\Response
148 */
149 public function show($id)
150 {
151     $answers=[];$textosAsoc=[];
152     $id_user=Auth::user()->id;
153     $sections = Seccion:: where('id_cuestionario', $id)->get();
154     if($sections!=null)
155         $union = DB::table('seccions')
156             ->join('cupreguntas', 'seccions.id', '=', 'cupreguntas.id_seccion')
157             ->join('preguntas', 'preguntas.id', '=', 'cupreguntas.id_pregunta')
158             ->where('seccions.id_cuestionario', '=', $id)
159             ->get();
160     if($union!=null)
161         foreach($union as $fu) {
162             $respuestas = Respuesta:: where('id_pregunta', $fu->id_pregunta)->get();
```

Figura 5.8: Consulta usando ORM para unión de tres tablas del sistema.

Tanto Eloquent como las migraciones se relacionan puesto que Eloquent proporciona la creación de modelos dentro de la aplicación. Estos modelos pueden tener lugar en la persistencia de nuestro sistema, ya que por cuestiones de seguridad Laravel no permite escribir directamente en la tablas, para ello se tiene que crear un modelo y definir

<sup>7</sup> ORM es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador. Es decir, es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos.

los campos en los cuales se podrá escribir. En el sistema de cuestionarios los modelos relacionados con la lógica de negocio se colocaron<sup>8</sup> en el directorio `/app/Yolotl`<sup>9</sup>, por ejemplo, `colaborador` y `pregunta`. Este directorio no pertenece a Laravel, se creó con la idea de concentrar los modelos que estén en el “core” de la aplicación. El resto, como el modelo de usuarios (`User.php`), se encuentran en el directorio por default (en `/app`).

### 5.6.5. Seeders y el componente faker

Los seeders son un recurso que nos permite cargar información a nuestras tablas, es decir, permite poblar la base de datos para no perder tiempo escribiendo de forma manual todos los datos y de esta manera probar, de forma sencilla, el funcionamiento de la aplicación.

Un seeder se ubica en la carpeta `/database/seeds` del proyecto de Laravel y para poder crear un nuevo seeder se usa el comando: `php artisan make:seeder nombre_seeder`

Con esto se crearon los archivos; sin embargo, para generar automáticamente los datos, se usa un componente llamado `faker`<sup>10</sup>.

En el desarrollo del sistema se utilizaron los seeders y el componente `faker` para llenar algunas de las tablas, entre ellas la tabla `Temas`, para lo cual se creó un seeder llamado `temasSeeder`. En la figura 5.9 se muestra el código que se implementó para la inserción automática de temas. Esto se realizó utilizando el método llamado `run()` dentro del archivo seeder (línea 14), aquí se insertaron cincuenta registros. En la línea 17 se hace una instancia de la clase `Faker`, con lo cual se puede hacer la inserción por ORM con texto de relleno en los campos que indicamos. Por ejemplo para el nombre del tema (líneas 22) se indicó que se quiere un dato único de tipo palabra y para la descripción (líneas 23) al objeto `faker` se le pide un texto cualquiera y de esta forma tenemos temas y subtemas “simulados”, cabe mencionar que todos estos temas son del tipo `Lorem ipsum`<sup>11</sup>.

---

<sup>8</sup> Cada modelo se crea mediante el uso de Artisan usando el comando `php artisan make:model Nombre_del_modelo -m`, si se indica a Artisan un parámetro adicional (-m) Artisan creará dos archivos nuevos, uno llamado `Nombre_del_modelo.php` dentro de la carpeta `/app`, el cual es el modelo y otro llamado `create_Nombre_del_modelo_table_000000.php` dentro de `database/migrations` que es la migración para dicha tabla. Para más información ver <https://laravel.com/docs/5.1/eloquent>

<sup>9</sup>Yolotl deriva de una palabra de origen náhuatl y significa corazón.

<sup>10</sup> Faker es un componente externo y necesita instalación con Composer: `composer require fzaninotto/faker`. Esta instrucción lo añade al archivo `composer.json`, en la sección de “require”, como se observó en la figura 5.3. Para más detalles ir a la página de Packagist (<https://packagist.org/packages/fzaninotto/faker>) o al repositorio en Github (<https://github.com/fzaninotto/Faker>).

<sup>11</sup> Lorem ipsum es el texto que se usa habitualmente en diseño gráfico en demostraciones de tipografías o de borradores de diseño para probar el diseño visual antes de insertar el texto final. El más usado está derivado del latín clásico.

```
\temasSeeder
8  /**
9   * Run the database seeds.
10  * Voy a sembrar (seeder) datos falsos (faker) en mi base de datos;
11  * en este caso sembrare temas
12  * @return void
13  */
14  public function run()
15  {
16      //
17      $faker=Faker::create();
18      $j=NULL;
19      $carbon = new \Carbon\Carbon();//Esto creará las fechas
20      for($i=0;$i<50;$i++){
21          $id= \DB::table('temas')->insertGetId([
22              'nombre' => $faker->unique()->word,
23              'descripcion' => $faker->text,
24              'id_padre' => $j,
25              'created_at' => $faker->dateTimeThisDecade($max = 'now'),
26              'updated_at' => $carbon->now()->format('Y-m-d H:i:s')
27          ]);
28          if($i%2==0)
29          { //echo "<br>".$i." es par";
30              $j=$faker->numberBetween($min = 1, $max = $id);
31          }else{
32              //echo "<br>".$i." es impar";
33              $j=NULL;
34          }
35      }
36  }
```

Figura 5.9: Utilizando un seeder y faker para crear temas simulados.

Por último, como datos adicionales en la líneas 25 y 26 se utilizó carbón y un faker para generar fechas falsas. Como se menciona en la sección 5.6.1 carbón genera fechas y aquí se esta indicando que para la columna update\_at se inserte la fecha presente; sin embargo, si queremos fecha aleatorias utilizamos un faker estableciendo la fecha máxima como la fecha actual, es decir, se simulara fechas en el pasado para indicar la columna create\_at, esto es, se simulará que el tema fue creado en determinada fecha hacia atrás.

## 5.7. Presentación web

Así como se hizo un breviarío tecnológico del back end en la sección anterior, la misma idea se plantea a continuación, pero esta vez dirigido a las tecnologías relacionadas con el front end, la aplicación desarrollada y junto al marco de trabajo de Laravel.

### 5.7.1. Bower

Para profesionistas en desarrollo de sistemas de software es raro que se comience todo el trabajo desde cero. Y en el caso de sistemas web es habitual apoyarse en bibliotecas Javascript como jQuery o CSS y HTML como Bootstrap, por tanto al construir el sitio web, se utilizó un manejador de dependencias llamado Bower<sup>12</sup>.

Al igual que Composer, Bower es una herramienta de administración de dependencias para front-end. Puede utilizarse para adquirir las versiones necesarias para cualquier paquete front-end necesario para la aplicación. Con Bower podemos gestionar fácilmente las dependencias del proyecto desde la consola. Bower no impone ninguna restricción en cuanto a la naturaleza de los paquetes, siempre que éstos se alojen en un repositorio Git, como por ejemplo GitHub.

En la figura 5.10 podemos ver el archivo bower.json que nos permite mantener un orden en las dependencias de un proyecto. Estas dependencias son utilizadas para el proyecto de cuestionarios web.

```
1 {
2   "name": "cuestionarios",
3   "description": "proyecto con laravel",
4   "main": "",
5   "authors": [
6     "AC"
7   ],
8   "license": "MIT",
9   "homepage": "",
10  "dependencies": {
11    "bootstrap": "^3.3.7",
12    "startbootstrap-sb-admin-2": "git://github.com/BlackrockDigital/startbootstrap-sb-admin-2.git#^3.3.7",
13    "bootstrap-select": "^1.11.0",
14    "jquery-validation": "^1.15.1",
15    "bootstrap-toggle": "^2.2.2",
16    "eonasdan-bootstrap-datetimepicker": "^4.17.43",
17    "jquery-confirm2": "craftpip/jquery-confirm#^3.0.1",
18    "jquery-ui": "^1.12.1"
19  }
20 }
21
```

Figura 5.10: Archivo JSON con el listado de las dependencias para el front end de la aplicación.

En la figura hay varias líneas en las que se puede observar a Bootstrap y diversos componentes relacionados a este framework. A continuación se hablará de éste y de como se utilizó para dar funcionamiento a la vista del sistema.

<sup>12</sup> <https://bower.io/>.

### 5.7.2. Bootstrap

Bootstrap es un framework o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

Lo más importante de Bootstrap para el proyecto se observa en la línea 12 de la figura 5.10, que fue el uso del template o plantilla **SB Admin 2**<sup>13</sup> un tema de administración de Bootstrap, un panel de control con interfaces, es decir, nos proporciona un “cascaron” profesional y editable sobre el cual se puede construir una aplicación web sin tener que lidiar con el diseño de un sitio. Además ofrece la ventaja del ahorro de tiempo para integrar la información ya que este cuenta con una estructura definida y esta optimizado para ser responsivo. Es importante decir que las plantillas prediseñadas pueden ser una opción válida para el desarrollo de sitios web (sin renunciar a la calidad) cuando existen determinadas circunstancias, como por ejemplo: entregas en extrema urgencia, soluciones temporales ante necesidades puntuales, o simplemente buscar soluciones económicas. En el caso de este proyecto se buscó un desarrollo rápido. Finalmente, un template ofrece un soporte de respaldo porque suele tener una gran comunidad detrás, que ante problemas es fácil conseguir una respuesta de alguna de los cientos de desarrolladores que vienen aportando constantemente novedades sobre el tema.

### 5.7.3. Blade

Las vistas en Laravel son la parte pública que el usuario de nuestro sistema va a poder ver, se escriben en HTML junto con un motor de plantillas llamado Blade.

Blade es básicamente un sub-lenguaje muy sencillo, que antes de ser usado por nuestra aplicación, es compilado a PHP plano. Para usar Blade, simplemente se crearon plantillas en el directorio resources/views con la extensión .blade.php en vez de .php. Por ejemplo: views/cuestionarios/editar.blade.php. Parte del código de este archivo se puede ver en la figura 5.11.

Esta imagen nos muestra varios conceptos que con Blade se manejan, por ejemplo, es común que en los proyectos web, se tenga un “layout” que consiste, por lo general, en un header mas un content y un footer, es decir, muchas páginas llevan dentro una cabecera,

---

<sup>13</sup> **SB Admin 2**. <https://startbootstrap.com/template-overviews/sb-admin-2/>.

```

1  @extends('home')
2  @section('pageTitle')
3      <h1><div class="row"><p> </p></div> </h1>
4  @stop
5  @section('styles')
6      {!! Html::style("bower_components/jquery-ui/themes/base/jquery-ui.min.css") !!}
7  @stop
8  @section('scriptop')
9      {!! Html::script("bower_components/jquery-ui/jquery-ui.min.js") !!}
10 @stop
11 @section('breadcrumb')
12     {!! Breadcrumbs::render() !!}
13 @stop
14 @section('content')
15     <div class="row">
16         <div class="col-lg-12">
17             <div class="chat-panel panel panel-default">
18                 @include('cuestionarios.partials.editar.cabecera')<!-- /.panel-heading -->
19                 @include('cuestionarios.partials.editar.body')<!-- /.panel-body -->
20                 @include('cuestionarios.partials.editar.footer')<!-- /.panel-footer -->
21             </div>
22         </div>
23     </div>
24 @stop
25 @section('modal')
26     @include('cuestionarios.modales.section')
27     @include('cuestionarios.modales.baul')
28 @stop
29 @section('scripts')
30     {!! Html::script("js/cuestionarios/addSection.js") !!}
31     {!! Html::script("js/cuestionarios/modLabel.js") !!}
32     {!! Html::script("js/cuestionarios/baul.js") !!}
33     <!-- DataTables JavaScript -->
34     {!! Html::script("bower_components/datatables/media/js/jquery.dataTables.js") !!}

```

Figura 5.11: Muestra de un archivo de plantilla con Blade.

un logo más el menú, su contenido y el pie de página que por lo regular esta repetido. Ese código HTML que se repite en cada página se llama layout, en el proyecto de cuestionarios web se usó también algo similar que con Blade lo definimos como nuestro “home”, es decir tenemos un archivo llamado views/home.blade.php. En “home” se va insertando las secciones que se necesiten para cada vista, dependiendo lo que queramos mostrar, para no repetir código; por ejemplo, en el código que muestra la figura 5.11 se inserta en home.blade.php. Por eso en la línea 1, se observa que se usa @extend, es decir, “extiende” la plantilla layout (home), como si fuera un objeto hijo que se extiende de un objeto padre. Es decir, que si vamos al archivo, en views/home.blade.php se podrá ver dentro del body, una línea como la siguiente:

Código: @yield('content')

Esa etiqueta de Blade permite definir una “sección” dinámica que puede ser reemplazada en la plantilla “hijo”, en este caso editar.blade.php usando las etiquetas:

Código:

```
@section ('content')  
<!--HTML AQUI -->  
  
@stop
```

De esta manera vamos añadiendo a nuestra página en común (el home) desde elementos a mostrar hasta el uso de scripts que se pueden requerir para una vista en especial (lineas 30-34), de esta forma obtenemos un código HTML más legible<sup>14</sup>.

Blade nos provee de muchas ventajas, además de modularizar nuestras vistas, también permite usar estructuras de control y variables de PHP directamente en ellas, aunque esto es posible usando las etiquetas de php, por ejemplo:

```
<?php echo $var ? >  
  
<?= $var ? >
```

Sin embargo, esto puede hacer que las vistas sean complicadas de entender y “sucias” al mezclar códigos (HTML y PHP).

Entonces con Blade podemos usar el siguiente código a modo de ejemplo: `{{ $pregunta }}`

Esto es el equivalente a `<? = $pregunta ? >`

Además del uso de variables podemos tener estructuras de control usadas en PHP dedicados al Back end (como lo son if, else, elseif, for, foreach, etcétera), pero que podemos usar en el front end si es que necesitamos tomar decisiones con ciertos datos. En la figura 5.12, se ve ejemplo de esto.

En el trozo de código mostrado se observan algunas estructuras de control como el if y el foreach, código que ayuda a construir las secciones en un cuestionario y sus correspondientes preguntas; si bien aquí se usan algunas variables como `$fusión` y `$sección` (lineas 49 y 51) estas variables vienen de un controlador y son el resultado de una consulta en la base de datos, es decir `$fusión` viene de la unión entre las tablas secciones, cuestionario-pregunta y preguntas, por otro lado la variable `$sección` es la consecuencia de solicitar, de la tabla secciones, las secciones para el cuestionario actual. Como se dijo con Blade podemos enviar información adicional hacia la vista desde el controlador, la variable la pasamos a la vista con el método auxiliar `with()`, con ello la vista contará con los datos al momento de generarse. En la sección 5.8, veremos con más detalles la interacción entre modelos-persistencia, controladores y vistas.

---

<sup>14</sup> Para más información acerca del uso de Blade visitar <https://laravel.com/docs/5.1/blade>.



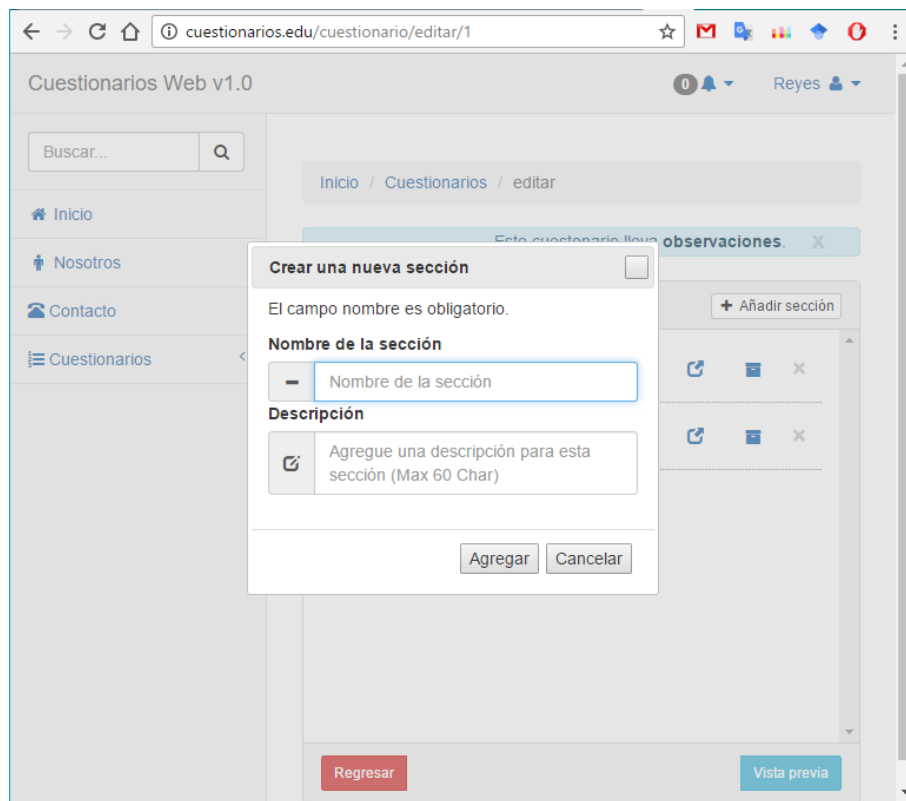


Figura 5.13: Ventana construida con jQuery-UI, para la creación de secciones.

## 5.8. Interacción en la aplicación

Hasta este momento se han descrito mecanismos importantes utilizados en el desarrollo de la aplicación. Sin embargo, la forma en cómo actúan en conjunto todos dando paso a la parte funcional de la aplicación, sigue en ausencia. Es por ello que se describirá el cómo marchan estos elementos, para que la aplicación funcione.

A grandes rasgos se puede decir que una aplicación en Laravel lleva en su interior un esquema o patrón MVC como el de la figura 5.14, sin embargo, aquí se observa un elemento extra: las rutas.

### 5.8.1. Rutas

Las rutas en las aplicaciones web corresponden con cadenas de textos fijos y en otras ocasiones cadenas que van a ser variables. En el caso de del sistema tenemos por ejemplo:

- cuestionarios.edu/banco/tema

- cuestionarios.edu/banco/pregunta
- cuestionarios.edu/cuestionario/crear
- cuestionarios.edu/cuestionario/generar/vista/1
- cuestionarios.edu/cuestionario/editar/1

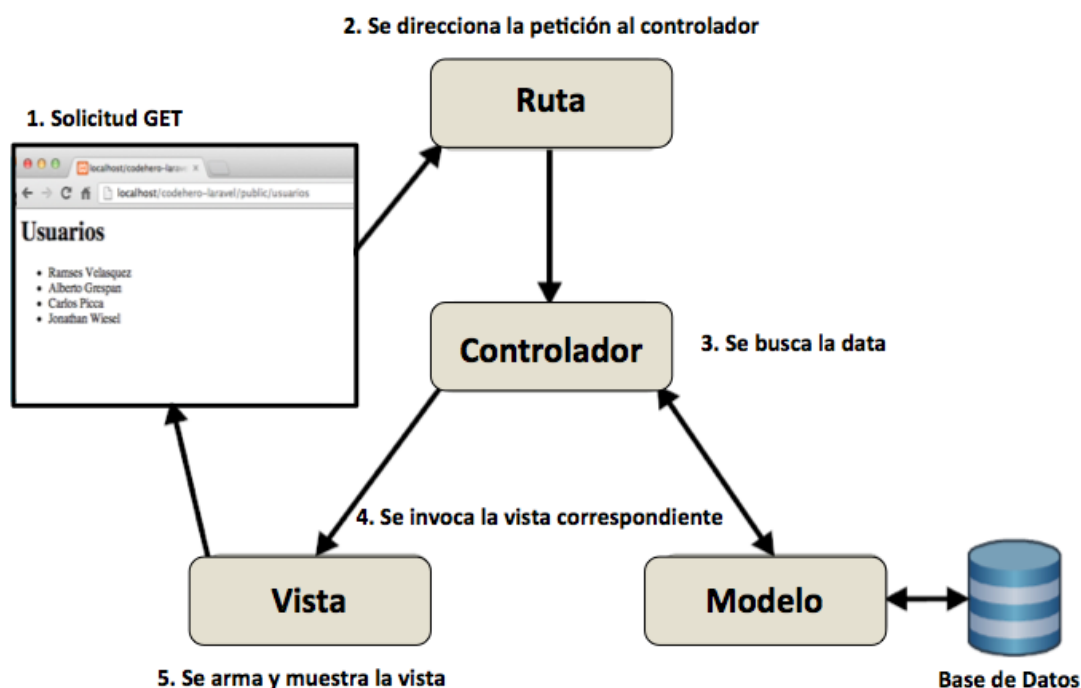


Figura 5.14: Lógica MVC en Laravel, para la construcción del sitio web.

Se puede apreciar que en las rutas<sup>17</sup> hay zonas que son variables, por ejemplo, el tipo de banco que se pretende ver (ya sea banco de temas o preguntas), el identificador del cuestionario para editar, o la acción para crear un cuestionario.

Estas rutas tienen una relación con el controlador ya que a través de ellas se puede ejecutar acciones y como consecuencia el sistema realiza peticiones, es decir, las rutas son para gestionar (mostrar, crear, actualizar y eliminar) los diferentes recursos que conforman a la aplicación (cuestionarios, preguntas, temas, observaciones, etcétera), junto con los controladores necesarios para responder a cada operación sobre los recursos.

En suma las rutas son el “puente” que existe entre las peticiones recibidas a determinada dirección (ruta) del proyecto y los métodos al interior de un controlador.

<sup>17</sup>Cabe decir que para que la ruta base del sistema fuera **cuestionarios.edu**, se tuvo que configurar en el servidor un Host Virtual, para así obtener el dominio local para el sitio.

## 5.8.2. Controladores

Un controlador en Laravel es básicamente una clase al interior del proyecto (extiende de un controlador base nombrado Controller) que posee una serie de métodos que serán ejecutados por el sistema. Usualmente dichos métodos son llamados directamente desde una ruta; sin embargo nada impide que un controlador tenga sus propios métodos, para uso interno y que en cierto modo ni siquiera sean ejecutados por una ruta de manera directa.

Un controlador se puede crear con Artisan, mediante el comando:

```
php artisan make:controller nombreController --resource
```

A continuación, se puede registrar una ruta de recursos para el controlador:

```
Route::resource('nombre', 'nombreController');
```

Esta declaración de ruta única crea múltiples rutas para manejar una variedad de acciones en el recurso. Este tipo especial de controlador<sup>18</sup> es llamado controlador de recurso (resource controller) ideal para módulos CRUDs, que facilita la construcción de controladores tipo RESTful<sup>19</sup>, es decir, el controlador generado ya tendrá métodos para cada una de estas acciones (figura 5.15), incluyendo notas que informan de los verbos HTTP (GET, POST, PUT o DELETE) y URIs que se están manejado. Como se mencionó en la sección 5.6.2, por medio de Artisan podemos ver la lista de rutas de nuestro sistema con el uso de un comando, estas rutas pueden venir acompañas de un verbo HTTP, que se muestra en la primera columna a la izquierda de la consola (puede referirse a la figura 5.5). El controlador usualmente trabaja con las peticiones de los verbos mencionados y están asociados a ciertos métodos de la siguiente forma:

- GET: index, create, show, edit.
- POST: store.
- PUT: update.
- DELETE: destroy.
- PATCH: update.

---

<sup>18</sup><https://laravel.com/docs/5.1/controllers>

<sup>19</sup> La Transferencia de Estado Representacional (en inglés Representational State Transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la world wide web.

```
class HomeController extends Controller
{
    /** Display a listing of the resource. ... */
    public function __construct()
    {
        $this->middleware('auth');
    }
    public function index()
    {
        return view('home');
    }
    /** Show the form for creating a new resource. ... */
    public function create()
    {
        //
    }
    /** Store a newly created resource in storage. ... */
    public function store(Request $request)
    {
        //
    }
    /** Display the specified resource. ... */
    public function show($id)
    {
        //
    }
    /** Show the form for editing the specified resource. ... */
    public function edit($id)
    {

```

Figura 5.15: Controlador RESTful.

Cada método RESTful es donde el programador coloca el código correspondiente a cada lógica del negocio, como se puede ver en la figura 5.16. Aquí se observa el código que está en el método `index` para el controlador de observaciones, el cual accesa al identificador del usuario en sesión actual y genera una consulta para mostrarle las observaciones que se le han hecho a los cuestionarios en los que el usuario a participado como editor

Nada nos obliga a llamar los métodos de una forma u otra; sin embargo, es una muy buena práctica darles un nombre acorde a su funcionalidad. Para el sistema de cuestionarios se estableció lo siguiente acerca del nombre de los métodos en los controladores:

- `index`: Mostrar todos los recursos asociados, respondiendo al método GET.
- `store`: Almacenar un nuevo recurso. Responde al método POST.
- `show`: Mostrar un recurso específico. Responde al método GET.
- `update`: Actualizar un recurso existente. Responde al método PATCH/PUT.
- `delete`: Eliminar un recurso existente. Responde al método DELETE.

```
/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    //
    $id_user=Auth::user()->id;//Usuario actual
    $cu_obs = DB::table('cuestionarios as a')
        ->join('observations as b', 'a.id', '=', 'b.id_cuestionario')
        ->join('users as c', 'b.id_observador', '=', 'c.id')
        ->where('a.id_solicitante', $id_user)
        ->select('b.*', 'a.titulo', 'c.name as autor', 'c.first_name as paterno')
        ->orderBy('b.created_at', 'desc')
        ->get();
    return view('observaciones.mostrar')
        ->with(compact('autor'))
        ->with(compact('cu_obs'));
}
```

Figura 5.16: Código del método index para mostrar las observaciones de un cuestionario.

Como se indicó anteriormente, cuando se crea un controlador, se puede registrar una ruta hacia él, la lógica de una ruta es simple: Verifica si la petición recibida coincide con su valor, si es el caso, entonces llama al método especificado del controlador asignado. Pero para que todo esto pueda funcionar, en el archivo routes.php (de la ruta app/Http/Controllers/), se define la ruta, que obtenga el método HTTP correspondiente y lo pase al controlador que es quien va a definir qué acción tomar. Por ejemplo este es el código de la ruta para un controlador dentro de la aplicación de cuestionarios web:

```
Route::resource('observation','ObservacionController');
```

```
Route::resource('seccion','SeccionController');
```

```
Route::resource('users','UserController');
```

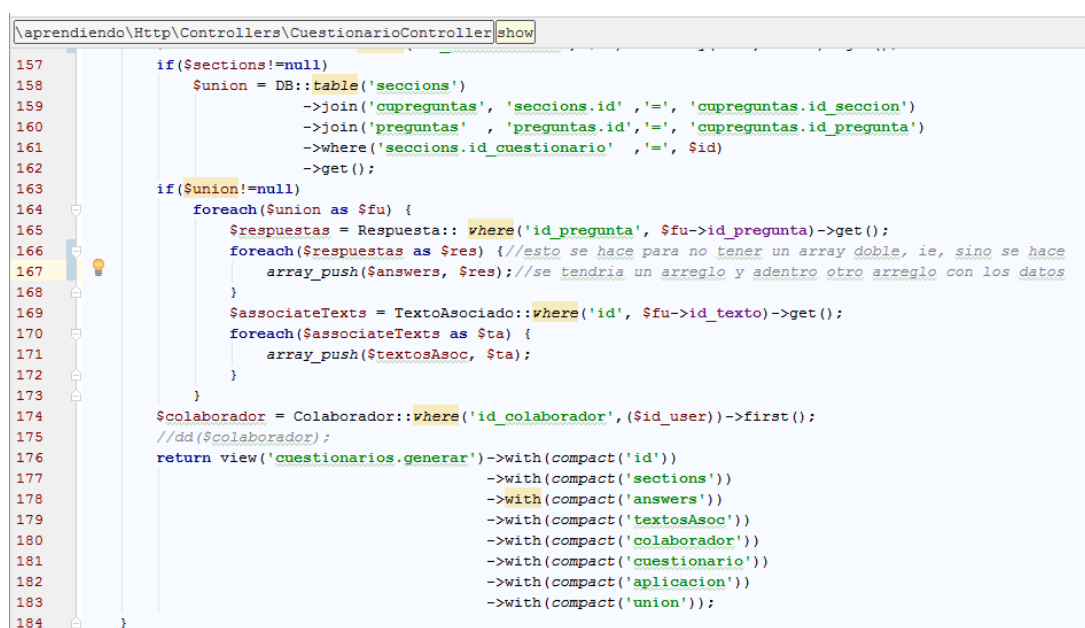
Generalmente es una buena idea tener al menos un controlador por cada recurso del sistema, o por cada conjunto de acciones que se puede realizar sobre un recurso (o recursos). Teniendo tanto las rutas como los controladores, se puede consumir los recursos del sitio web.

Una vez establecido lo anterior, se puede decir que un controlador puede verse como una capa que sirve de enlace entre las vistas y los modelos (figura 5.14), respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo

de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

Hay un concepto que se usa mucho cuando se explica el MVC que es la "lógica de negocio". Se entiende por lógica de negocio el conjunto de reglas que se siguen en el software para reaccionar ante distintas situaciones. En la aplicación de cuestionarios web el usuario se comunica con el sistema por medio de una interfaz, pero cuando acciona esa interfaz para realizar acciones con el programa, se ejecutan una serie de procesos que se conocen como la lógica del negocio.

La interfaz mencionada corresponde a la vista de la aplicación. La vista contiene el código que el usuario verá en su navegador y con la que va a poder interactuar. La interacción entre vista y controlador, ocurre mediante el paso de variables con un método llamado **with()**. Ahora en la figura 5.17 se muestra de la línea 176 a la 183 el paso de parámetros a una vista por medio de un controlador.



```
157 if($sections!=null)
158     $union = DB::table('seccions')
159         ->join('cupreguntas', 'seccions.id', '=', 'cupreguntas.id_seccion')
160         ->join('preguntas', 'preguntas.id', '=', 'cupreguntas.id_pregunta')
161         ->where('seccions.id_cuestionario', '=', $id)
162         ->get();
163
164 if($union!=null)
165     foreach($union as $fu) {
166         $respuestas = Respuesta::where('id_pregunta', $fu->id_pregunta)->get();
167         foreach($respuestas as $res) {//esto se hace para no tener un array doble, ie, sino se hace
168             array_push($answers, $res);//se tendria un arreglo y adentro otro arreglo con los datos
169         }
170         $associateTexts = TextoAsociado::where('id', $fu->id_texto)->get();
171         foreach($associateTexts as $ta) {
172             array_push($textosAsoc, $ta);
173         }
174     }
175
176 $colaborador = Colaborador::where('id_colaborador', ($id_user))->first();
177 //dd($colaborador);
178 return view('cuestionarios.generar')->with(compact('id'))
179     ->with(compact('sections'))
180     ->with(compact('answers'))
181     ->with(compact('textosAsoc'))
182     ->with(compact('colaborador'))
183     ->with(compact('cuestionario'))
184     ->with(compact('aplicacion'))
185     ->with(compact('union'));
```

Figura 5.17: Paso de parámetros de un controlador a una vista

En la imagen se ilustra parte del código para la extracción de información que ayudará a construir o generar un cuestionario: por ejemplo, en la línea 156, se hace la consulta para extraer las secciones y más adelante (línea 158 a 162) se hace la unión entre tablas por ORM para obtener las respuestas de opción múltiple y los textos asociados, si es que existen. De esta forma se tienen variables (en este caso serán arreglos) que contienen la información de las consultas y finalmente la instrucción *return* de la línea 176 indica que se usará la vista que se encuentra en el directorio cuestionarios y su nombre es generar

(`cuestionarios.generar`); luego con el método **with()**, pasamos el nombre de esas variables obtenidas mediante las consultas, a lo que del otro lado (en la vista) se recibirán y se podrá trabajar con ellas a través del sistema de plantillas de Laravel Blade.

Por el contrario, el paso de datos de una vista a controlador se hace por el uso de formularios y los dos métodos de envío en HTML, es decir, usando el método POST o usando el método GET.

---

## Capítulo 6

# Recorrido por el Sistema

Una vez unido todas las piezas teóricas y prácticas, enunciadas en los capítulos anteriores, y haber construido un prototipo bajo un ciclo de vida orientado al incremento en funcionalidad, con una base de datos relacional e implementado en un marco de trabajo como lo fue Laravel, podemos mostrar, en este capítulo, la descripción del prototipo que permite la creación, aplicación y evaluación de cuestionarios para la UACM, resultado del análisis e implementación de los requisitos, describiendo su comportamiento y la interacción con los diferentes tipos de usuario.

### 6.1. Estructura y funcionamiento de la aplicación

De manera general el sitio web esta conformado por cuatro vistas diferentes, una por cada tipo de usuario. Para iniciar la interacción con el sistema, todos los usuarios deben autenticarse, para esto se les presenta una pantalla (como la que se muestra en la figura 6.1) en la cual, se pide al usuario que ingrese su correo<sup>1</sup> y su clave. El sistema cuenta con alerta de mayúsculas activado y las contraseñas están cifradas<sup>2</sup>. Por otro lado, las rutas están protegidas con un middleware, viendo a este como un puesto de vigilancia entre las rutas y los controladores en la que habrá un guardia de seguridad monitoreando todo lo que pasa por allí para ver si cumple con los requerimientos definidos, es decir si alguien intenta ir a una ruta sin haberse autenticado el middleware lo redireccionará de vuelta al login.

---

<sup>1</sup>Es importante decir que se solicita el correo ya que el usuario ingresará con su correo institucional, una vez la aplicación este en modo de producción.

<sup>2</sup>Laravel proporciona la encriptación tipo AES vía la extensión de PHP de Mcrypt. Para más información ver <https://laravel.com/docs/5.1/encryption>

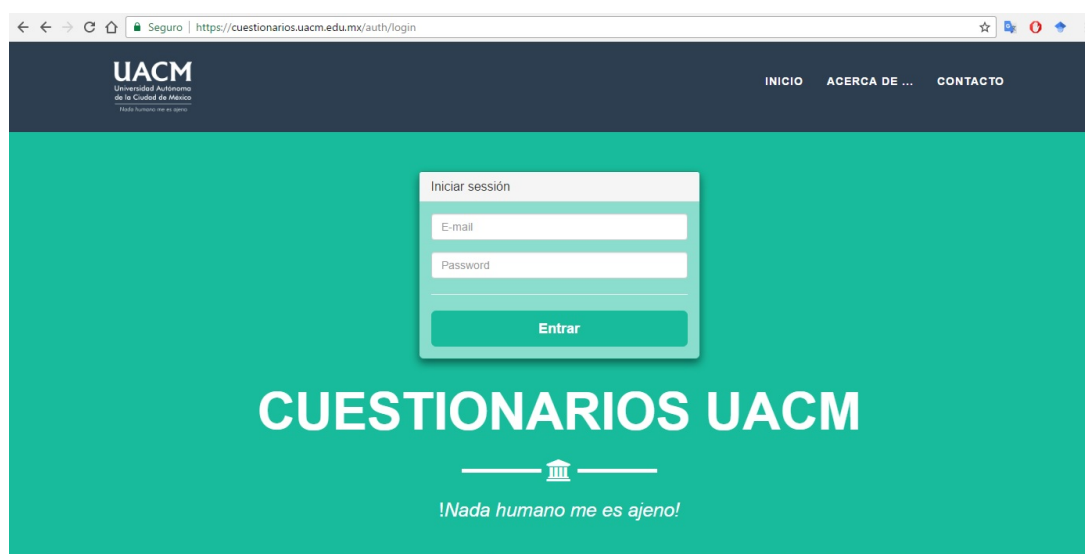


Figura 6.1: Login de entrada para usuarios.

Si los datos introducidos en el login son incorrectos el sistema imprime un mensaje de error en la misma pantalla; en caso contrario el sistema presenta la vista correspondiente al tipo de usuario. Una vez dentro del sistema se redirecciona al listado de cuestionarios actuales y dependiendo de que tipo de usuario sea se muestran diversas funcionalidades.

A continuación se irán describiendo las funcionalidades para cada tipo de usuario, conforme se describe el proceso o ciclo de vida de un cuestionario, es decir, desde su creación, edición, resolución y resultados.

### 6.1.1. Funcionalidades para el Administrador

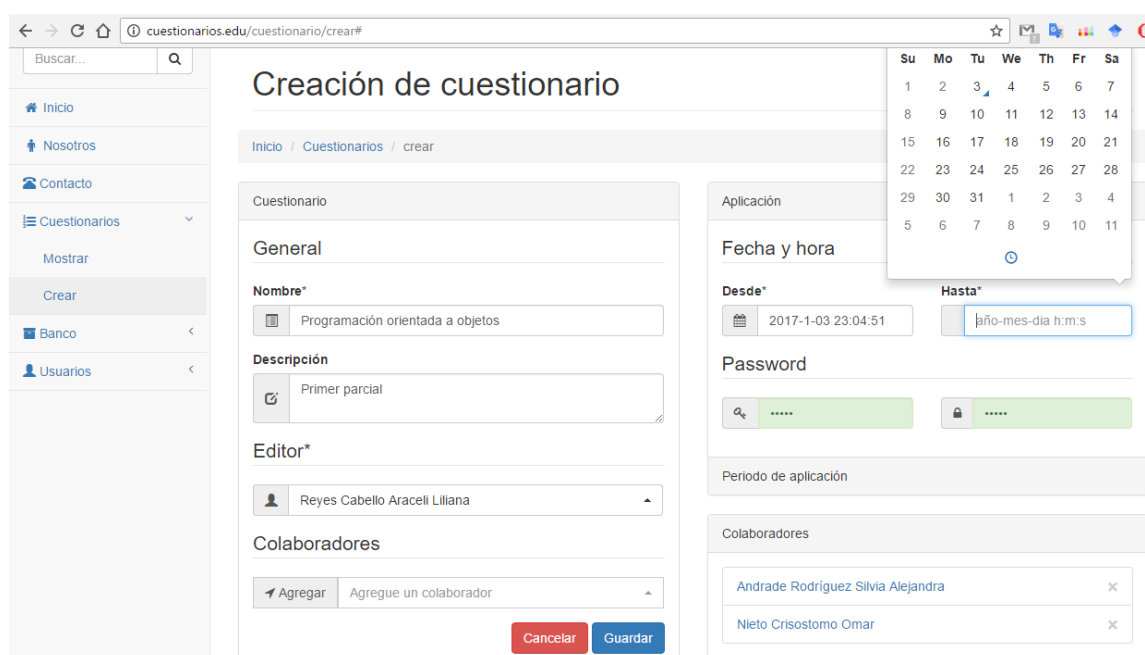
La función más importante para un administrador dentro de la aplicación web es la creación de un cuestionario; según los requisitos un cuestionario sólo podrá ser creado por un administrador (regla de negocio), porque únicamente este puede tener acceso al formulario de creación de cuestionario. Así que si un usuario desea un cuestionario deberá solicitarlo al administrador.

#### 6.1.1.1. Creación de nuevo cuestionario

Al entrar al sistema se mostrará una pantalla con un menú que permitirá crear un cuestionario al usuario administrador. Se debe elegir la opción cuestionario en el menú del lado izquierdo y posteriormente elegir la opción *crear*. Se desplegará un formulario

(figura 6.2) en el cual debe ingresar los datos correspondientes al cuestionario, tales como el título del cuestionario, una descripción opcional, el nombre del editor, así como los colaboradores. Los colaboradores agregados se van colocando en un panel o contenedor en donde se muestra el nombre del colaborador seleccionado, con la posibilidad de poder eliminarlo si es necesario. Tanto el editor como los colaboradores se pueden seleccionar de una lista que cuenta con un control especial de búsqueda<sup>3</sup> de tal manera que basta con ingresar el nombre o apellido de un usuario dentro de la lista y este componente encontrará la coincidencias.

El formulario de creación cuenta además con una sección para ingresar los datos de la aplicación del cuestionario, como la fecha de inicio y final de la aplicación; un campo para ingresar una contraseña opcional para que solo aquellos participantes con la clave puedan resolver el cuestionario. La fecha inicial está configurada para que siempre comienza en el día actual. Cabe señalar que el administrador podrá modificar información del cuestionario como la fecha de aplicación si el usuario editor así lo requiere, siempre y cuando el cuestionario tenga un estado de edición o de publicación.



The screenshot shows a web browser window with the URL `questionarios.edu/cuestionario/crear#`. The page title is "Creación de cuestionario". On the left is a navigation menu with items: Inicio, Nosotros, Contacto, Cuestionarios (expanded), Mostrar, Crear, Banco, and Usuarios. The main content area is titled "Creación de cuestionario" and contains the following sections:

- Cuestionario**
  - General**: Nombre\* (text input: "Programación orientada a objetos")
  - Descripción**: (text area: "Primer parcial")
  - Editor\***: (dropdown menu: "Reyes Cabello Araceli Lilliana")
  - Colaboradores**: (button "Agregar" and input "Agregue un colaborador")
- Aplicación**
  - Fecha y hora**: Desde\* (calendar icon, date: "2017-1-03 23:04:51"), Hasta\* (input: "año-mes-día h:m:s")
  - Password**: (two masked password inputs)
  - Período de aplicación**: (empty field)
- Colaboradores**: (list of names with delete icons: "Andrade Rodríguez Silvia Alejandra", "Niato Crisostomo Omar")

At the bottom right are "Cancelar" and "Guardar" buttons. A calendar pop-up is open over the "Desde\*" date field, showing a grid of dates from Sunday to Saturday.

Figura 6.2: Formulario para crear un cuestionario.

<sup>3</sup>El componente mencionado es llamado *bootstrap-select* y es externo al framework, fue instalado con el manejador de dependencias bower, para más información visitar: <https://silviomoreto.github.io/bootstrap-select/>.

### 6.1.1.2. Inserción de temas

Otra opción importante para el administrador es el ingreso de temas y subtemas para las preguntas. La inserción de temas es muy simple, cuando se ingresa al menú y se da click en opción de *Banco* (que hace referencia al banco de preguntas y temas) se despliega un submenú que muestra la opción *Temas*, al dar click se muestran los temas disponibles (figura 6.3) y en la parte superior derecha de la lista de temas esta un botón **Añadir tema** al dar click en este pasamos a un formulario (figura 6.4) donde se tiene que ingresar el título o nombre del tema y una descripción. Además si ya existen algún tema previo este se puede elegir este como tema padre y de esta forma obtendríamos un subtema para la pregunta.

Tema	Descripción	Tema padre	Fecha Creación	Fecha modificación	Acción
accusantium	Voluptatibus quae sit vel commodi sed nostrum. Suscipit voluptatibus ratione ea omnis rerum. Consectetur eum explicabo rerum placeat vel.	***	13-11-2010	03-01-2017	No tiene acciones asignadas
alias	Impedit libero impedit quos qui quasi facilis. Ratione quam recusandae sit dicta. Necessitatibus ipsa iste dolores.	***	21-06-2011	03-01-2017	No tiene acciones asignadas
animi	Debitis maiores excepturi molestias quo qui asperiores. Eligendi animi minima mollitia explicabo rerum. Odit possimus et voluptatem harum et excepturi corrupti. Voluptate temporibus ea rerum qui.	mollitia	24-06-2015	03-01-2017	No tiene acciones asignadas
culpa	Dolor veniam voluptatem sit tempore aliquam ea. Quos nobis vitae	saepe	27-08-2010	03-01-2017	No tiene

Figura 6.3: Listado de temas disponibles.

### 6.1.1.3. Otras funcionalidades

La aplicación permite modificar un cuestionario siempre y cuando este tenga un estatus de edición, para el caso en que se tenga que cambiar la fecha de aplicación o agregar o quitar un colaborador por ejemplo. El resto de las opciones para el administrador son solo de consulta, de acuerdo con las reglas del negocio. Por ejemplo, la opción de *Usuarios* en el menú principal y la opción de *Preguntas* en el submenú de *Cuestionarios* del menú principal.



The image shows a web application interface. On the left is a sidebar menu with the following items: Inicio, Nosotros, Contacto, Cuestionarios, Banco (with a dropdown arrow), Preguntas, Temas (highlighted), and Usuarios. The main content area is titled 'Agregando un nuevo tema' and has a breadcrumb trail: Inicio / Banco de preguntas / Preguntas / temas / crear. The form contains the following fields: 'Nombre \*' (required), 'Descripción', and 'Tema padre' (a dropdown menu with the text 'Seleccionar un tema ...'). A blue 'Guardar' button is located below the form. At the bottom of the form, there is a red asterisk icon followed by the text '\* Campo requerido'.

Figura 6.4: Formulario para crear un tema o subtema.

### 6.1.2. Funcionalidades para el Editor

El editor tiene varias acciones implementadas en el sistema, primero al igual que el administrador el editor puede ver la lista de cuestionarios activos, en los cuales participa como editor o colaborador. Además de poder editar un cuestionario también puede ver los detalles de este al presionar el botón con el mismo nombre, con ello se despliega en pantalla información como: quiénes son los colaboradores y las fechas de aplicación (figura 6.5).

Comenzaremos con explicar la manera de editar un cuestionario, ya que este es uno de los pilares o ejes del sistema.

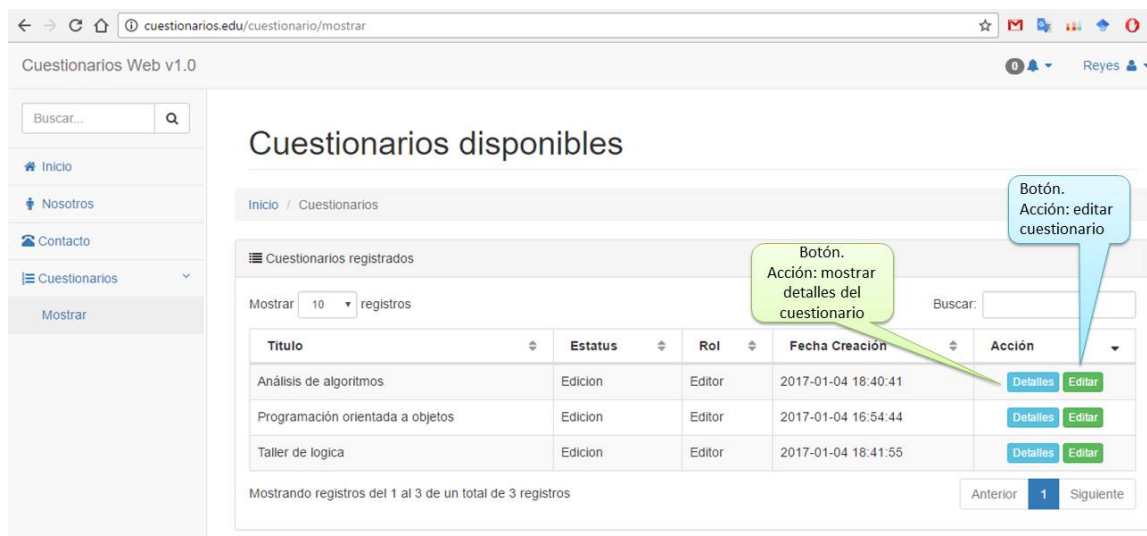


Figura 6.5: Pantalla que muestra los cuestionarios disponibles para un editor.

### 6.1.2.1. Editar un cuestionario

Cuando se da click en el botón **Editar** se envía al usuario a una nueva pantalla que contiene los elementos para la edición del cuestionario (figura 6.6). La descripción de cada componente se detalla a continuación conforma a la enumeración dada en la imagen.

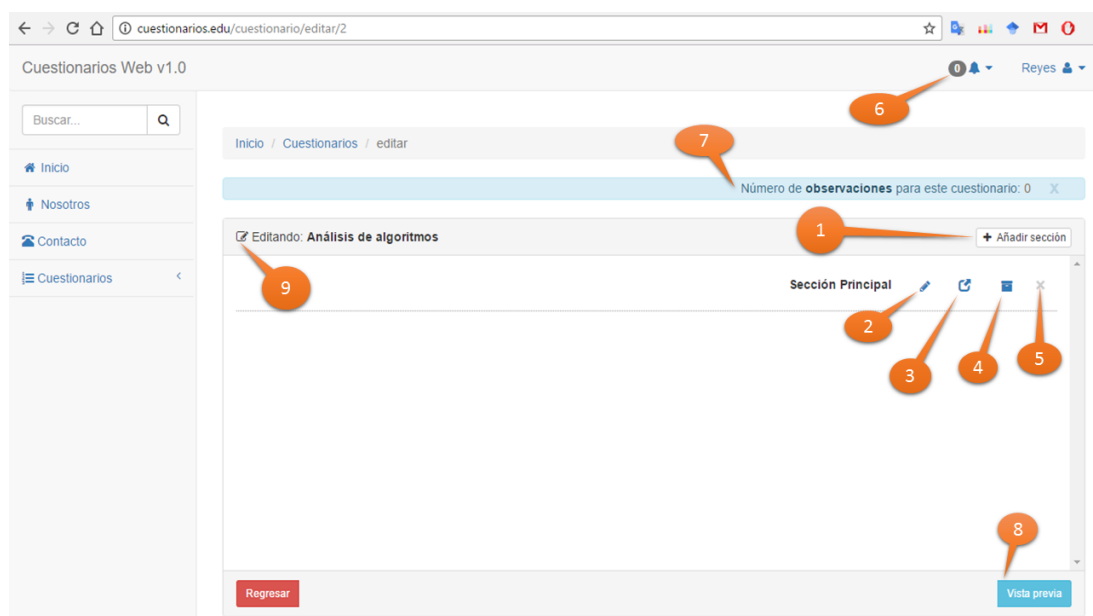


Figura 6.6: Vista para la edición de un cuestionario.

- 1) El número uno señala al botón de inserción de nueva sección. Por default al crear un nuevo cuestionario este siempre tiene una nueva sección con el título “*Sección principal*”, una sección sirve para dividir las preguntas en grupos; por ejemplo si un cuestionario fuese de matemáticas, podría dividirse por secciones, por ejemplo: sección para número enteros, expresiones algebraicas y geometría básica, por nombrar algunas.
- 2) Botón para modificar el texto de la sección. Si se desea cambiar el nombre de una sección ya sea la que viene por default o una que se agrego posteriormente, el botón de edición de texto de sección proporciona un pequeño formulario (ver figura 6.7) para hacer el cambio de nombre y poder agregar o modificar una descripción breve; la descripción se puede ver al colocar el puntero del mouse sobre el texto de la sección.

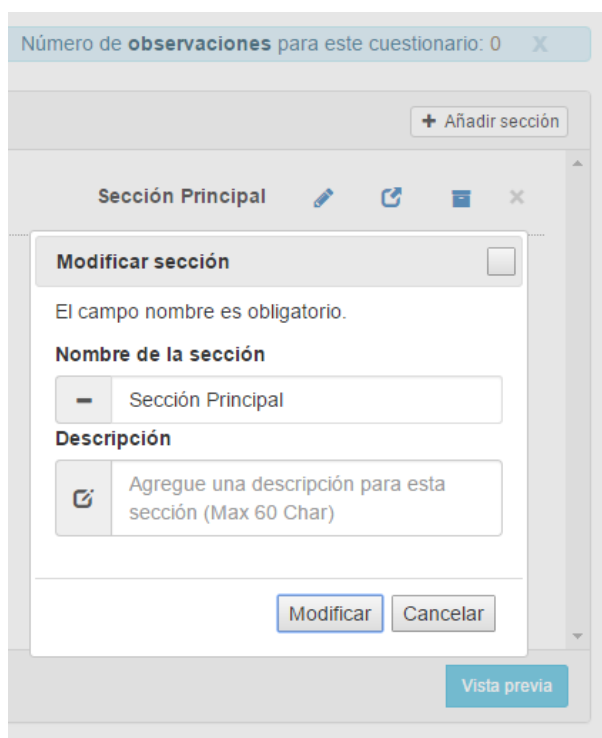


Figura 6.7: Formulario para creación y modificación de nombre y descripción de una sección.

- 3) Botón para introducir una nueva pregunta. Cuando se presiona este botón se pasa a una nueva vista en donde se puede crear la nueva pregunta para el cuestionario actual. Es tema se verá con más detalle en la sección 6.1.2.2.
- 4) Botón de inserción de pregunta de baúl. Si existen preguntas de otros cuestionarios y estos tienen un estatus de aprobado o resuelto, estas preguntas estarán disponibles en el baúl de preguntas y por tanto se pueden anexar al cuestionario. En la sección 6.1.2.3 se hablará al respecto.

- 5) Botón para eliminar sección. Se pueden eliminar una sección si se desea, sólo hay que tener en cuenta que las preguntas que estén dentro de una sección se eliminarán también. Para el caso de las preguntas de baúl no habrá efecto al momento de borrar una sección, estas seguirán siendo disponibles dado que pertenecen a otro cuestionario.
- 6) Icono de notificaciones de observaciones hechas por un colaborador. El icono de notificaciones muestra las observaciones hechas por un colaborador asociado al cuestionario, si el editor ha publicado el cuestionario, los colaboradores pueden realizar comentarios para mejorar o corregir detalles acerca de su contenido. En la sección 6.1.3.1, se hará mención de como generar una observación siendo un usuario colaborador.
- 7) Barra de conteo de observaciones hechas al cuestionario. Como su nombre lo indica la barra de número de observaciones proporciona al usuario editor un pequeño dato acerca de cuántas observaciones lleva el cuestionario que se está editando, este dato es importante para poder estar consciente de que el cuestionario ha sido revisado por parte de los que integrantes del equipo.
- 8) Botón de vista previa de cuestionario. Mientras el editor va agregando las preguntas al cuestionario, se puede hacer una vista preliminar de la aplicación, es decir, el como lo visualizará un usuario participante, esto mediante el uso del botón que esta señalado con el número ocho.
- 9) Botón para modificar el título y descripción de un cuestionario. Si se desea cambiar el título o nombre de un cuestionario basta con dar click sobre el icono cerca del titulo, lo que despliega un pequeño formulario para actualizarlo.

#### 6.1.2.2. Inserción de nueva pregunta

La creación e inserción de una nueva pregunta dentro de un cuestionario en edición, se lleva a cabo por medio de un formulario en una nueva página o vista. A continuación se explicarán los pasos a seguir para la creación de una nueva pregunta.

Una vez presionado el botón de nueva pregunta el sistema muestra un pequeño panel en donde se elige el tema de la pregunta y el tipo de pregunta<sup>4</sup> (figura 6.8).

---

<sup>4</sup>Es importante aclarar que aunque el sistema muestre dos tipos de preguntas en esta versión solo está disponible el tipo de pregunta de opción múltiple.

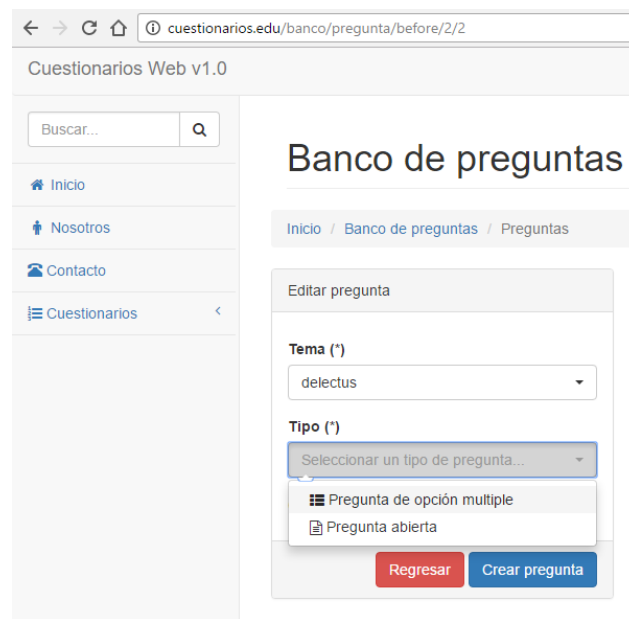


Figura 6.8: Elección de tema y tipo de pregunta.

Cuando se presiona el botón **Crear pregunta**, se despliega una pantalla con los campos para insertar la pregunta, así como el puntaje que tendrá (figura 6.9).

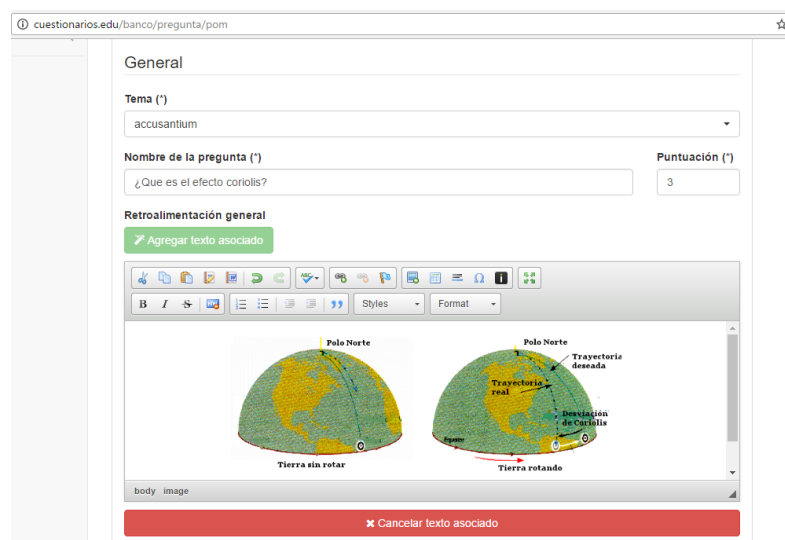


Figura 6.9: Formulario para nueva pregunta.

En la figura 6.9 también se observa que se puede asociar un texto adicional (esto es opcional), el cual puede servir para enriquecer una pregunta, como por ejemplo se puede añadir una explicación que favorezca la comprensión de lo se espera como respuesta. El texto asociado puede ser desde simple texto plano, una fórmula matemática, una imagen

o la combinación de todas estas. Solo hay que presionar el botón de **Agregar texto asociado** y el sistema activará el control para insertar un texto enriquecido.

En el caso de las fórmulas matemáticas existe un control que permite insertar una fórmula mediante el uso de comandos o en formato tipo  $\text{\LaTeX}$  (figura 6.10).

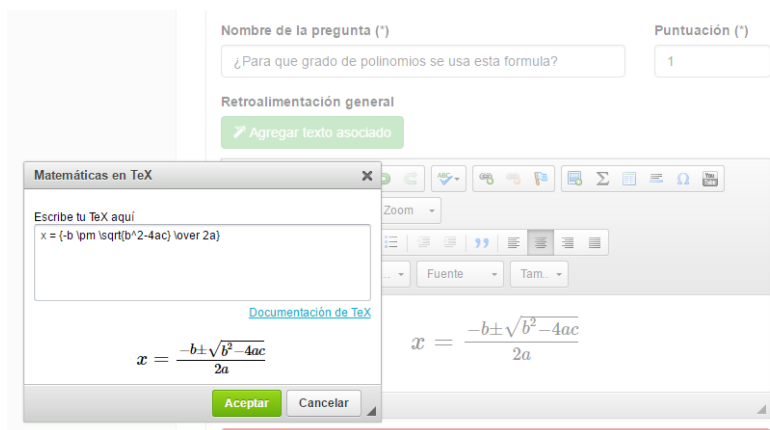


Figura 6.10: Fórmulas matemáticas en formato  $\text{\LaTeX}$ .


Para insertar una imagen se da click en el icono de Imagen que está en el control de texto asociado  el cual lanzará un cuadro de diálogo para insertar una imagen (figura 6.11), desde un servidor de imágenes.



Figura 6.11: Cuadro de diálogo para inserción de imagen.

Desde este control se puede colocar la url de la imagen directamente o subir una imagen desde nuestra computadora, si se desea esta segunda opción se debe presionar el botón que

dice **Browse Server** el cual mostrará una ventana (figura 6.12), desde la cual se puede cargar la imagen almacenada en la PC. Al subir una imagen, estará disponible para el cuestionario y cualquier otro usuario podrá usarla. La imagen aparecerá en la lista del lado derecho (figura 6.13).



Figura 6.12: Frame para subir imágenes.

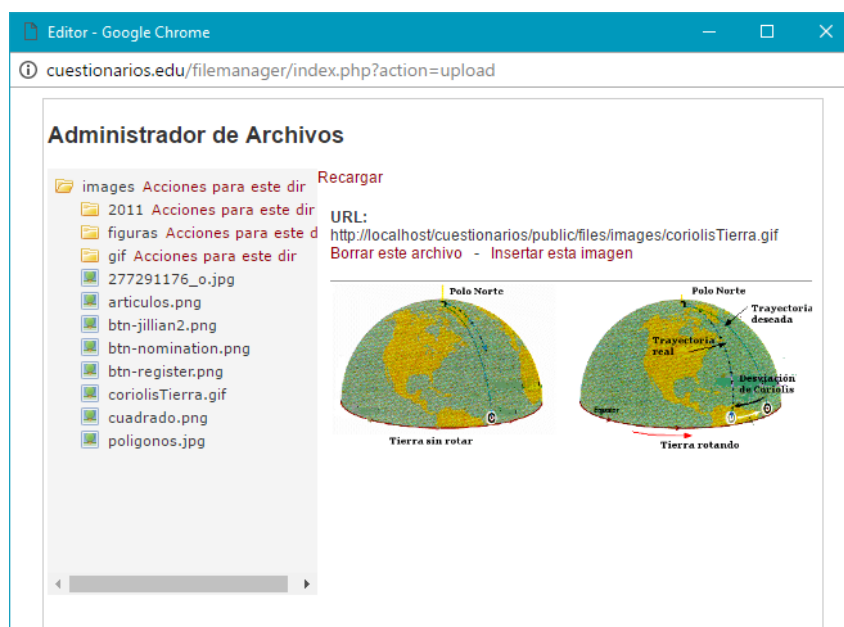
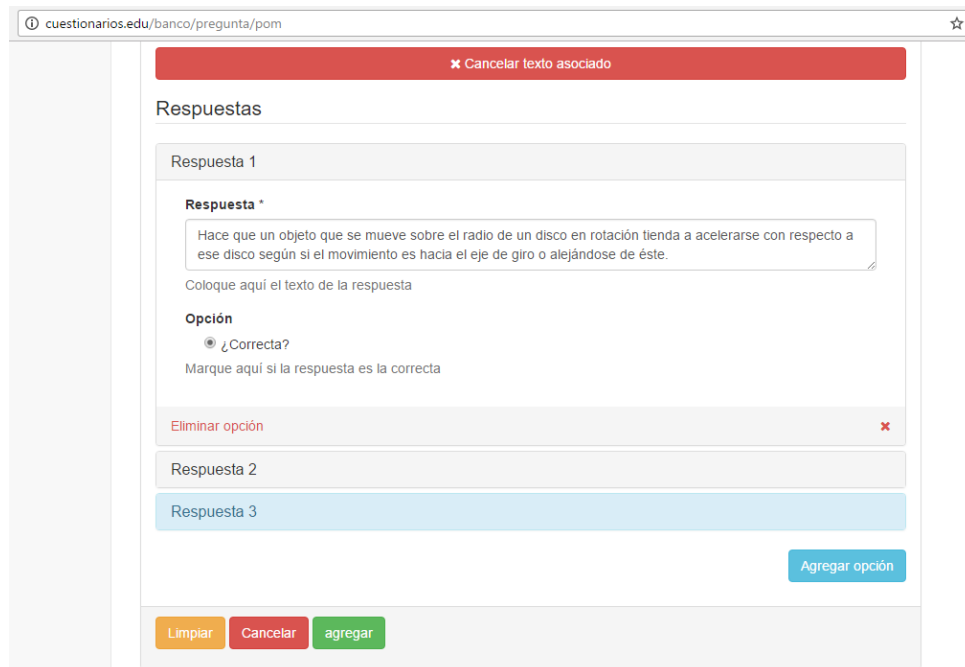


Figura 6.13: Imagen cargada y disponible.

Teniendo lo anterior se puede continuar con la inserción de las respuestas. Para esto el sistema ofrece por default dos respuestas, considerando que son preguntas de opción

múltiple. Si se necesitan más respuestas está el botón **Agregar opción**. Para llenar las opciones de respuesta se ingresa el texto de cada respuesta y se selecciona la opción que sea correcta (figura 6.14).



The screenshot shows a web browser window with the URL `questionarios.edu/banco/pregunta/pom`. At the top, there is a red bar with the text "Cancelar texto asociado". Below this, the section is titled "Respuestas". Under "Respuesta 1", there is a text input field containing the text: "Hace que un objeto que se mueve sobre el radio de un disco en rotación tienda a acelerarse con respecto a ese disco según si el movimiento es hacia el eje de giro o alejándose de éste." Below the input field, there is a label "Coloque aquí el texto de la respuesta". Underneath, there is a section for "Opción" with a radio button selected next to the text "¿Correcta?". Below this, there is a label "Marque aquí si la respuesta es la correcta". At the bottom of the "Respuesta 1" section, there is a red button labeled "Eliminar opción" with a small red 'x' icon. Below "Respuesta 1", there are two more sections labeled "Respuesta 2" and "Respuesta 3", both currently empty. At the bottom right of the "Respuestas" section, there is a blue button labeled "Agregar opción". At the very bottom of the interface, there are three buttons: "Limpiar" (orange), "Cancelar" (red), and "agregar" (green).

Figura 6.14: Insertando respuestas para una pregunta.

El sistema cuenta también con una serie de validaciones para evitar la inserción de datos no válidos. Por ejemplo, que siempre se escriba el texto de una pregunta, que la puntuación siempre sea un número entero, así como la vigilancia de que siempre se escoja una respuesta correcta y que todas las respuestas contengan texto.

Finalmente, cuando las preguntas están dentro del cuestionario estas se pueden modificar, eliminar o cambiar su puntaje según sea necesario. En la figura 6.15 se señalan los iconos respectivos para realizar dichas tareas.

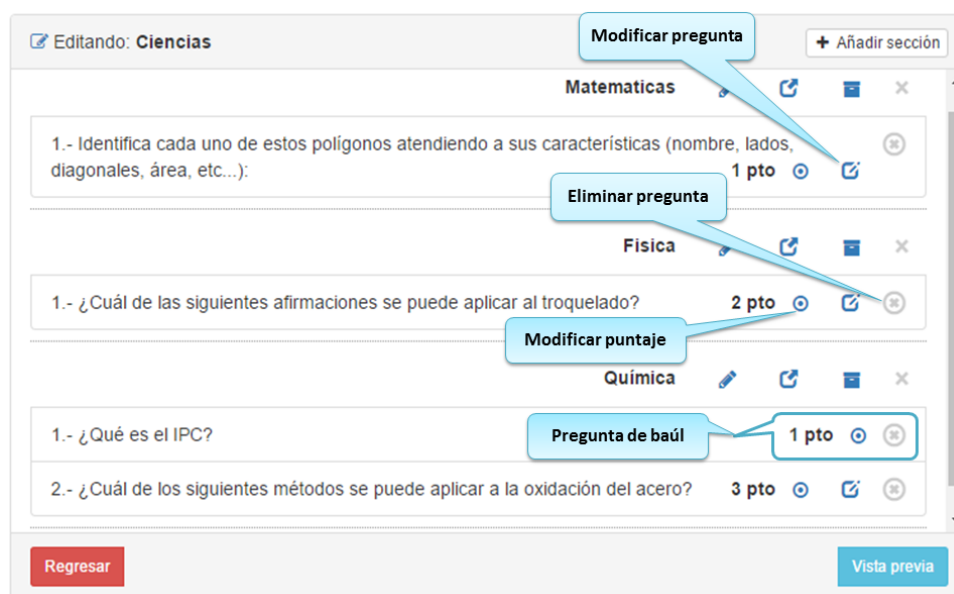


Figura 6.15: Cuestionario con preguntas.

### 6.1.2.3. Inserción de una pregunta de baúl

Cuando se le solicita al sistema una pregunta de baúl este desplegará una ventana donde se muestran las preguntas existentes en una tabla (figura 6.16), en ella podemos hacer búsquedas de una pregunta de acuerdo a una palabra clave que se tenga en mente y se nos mostrarán preguntas relacionadas con esa palabra, ya sea en el texto de la pregunta o en su tema. La adición de una pregunta de baúl es muy simple, sólo se selecciona una o varias preguntas de la tabla y se anexarán al cuestionario. Por obviedad no se pueden tener preguntas repetidas en un cuestionario. Por lo tanto, el sistema notificará si ya existen las preguntas.

Cabe señalar que si una pregunta es insertada del baúl, no puede ser modificada pues pertenece a otro cuestionario. Como se muestra en la figura 6.15 el icono para modificar no se encuentra disponible, lo que nos da también como indicador de que esa pregunta pertenece a otro cuestionario; sin embargo, el puntaje de la pregunta, así como de su estimación son posibles en el cuestionario en edición. Similar al caso del punto 5 de la sección 6.1.2, al eliminarse una pregunta de baúl, solo se eliminará del cuestionario presente y no de la base datos, pues como se ha dicho repetidamente una pregunta de baúl pertenece a uno o varios cuestionarios.



The image shows a dialog box titled "Preguntas de baúl". At the top left, it says "Mostrar 10 registros" with a dropdown arrow. To the right is a search field labeled "Buscar:". Below this is a table with two columns: "Pregunta" and "Tema". The table is currently empty, displaying the message "Ningún dato disponible en esta tabla". Below the table, it says "Mostrando registros del 0 al 0 de un total de 0 registros". There are two buttons: "Anterior" and "Siguiete". At the bottom right, there are two buttons: "Agregar" and "Cancelar".

Figura 6.16: Cuadro de diálogo para inserción de preguntas del baúl.

#### 6.1.2.4. Aprobación de un cuestionario

Cuando el editor ha decidido concluir el cuestionario este puede publicarlo en la sección de vista previa, con el botón **Publicar** (esquina inferior izquierda de la figura 6.17). El sistema preguntará si está de acuerdo en hacer dicha publicación y al aceptar el estatus del cuestionario cambiará. En este punto existen dos vertientes:

- 1.- El estatus del cuestionario pasará a publicado si existe al menos un colaborador, ya que como se mencionó, el cuestionario tendrá que ser aprobado para su aplicación.
- 2.- Si no hay colaboradores el sistema pasaría automáticamente el estatus de cuestionario aprobado, lo que implica que está listo para su aplicación.



Figura 6.17: Cuestionario listo para publicación.

### 6.1.3. Funcionalidades para el Colaborador

La función principal de un colaborador es revisar los cuestionarios que han sido publicados previamente por un editor y dar su voto aprobatorio, o bien, hacer observaciones. Para ello al ingresar al sistema puede ver la lista de cuestionarios activos en los cuales participa (figura 6.18), tanto como colaborador o como editor. Además en la columna de acciones (última columna) se puede ver el botón **Revisar** con el que podrá visualizar el cuestionario, en la misma forma que lo verá un participante, con la diferencia en que un colaborador podrá ver las respuestas correctas.

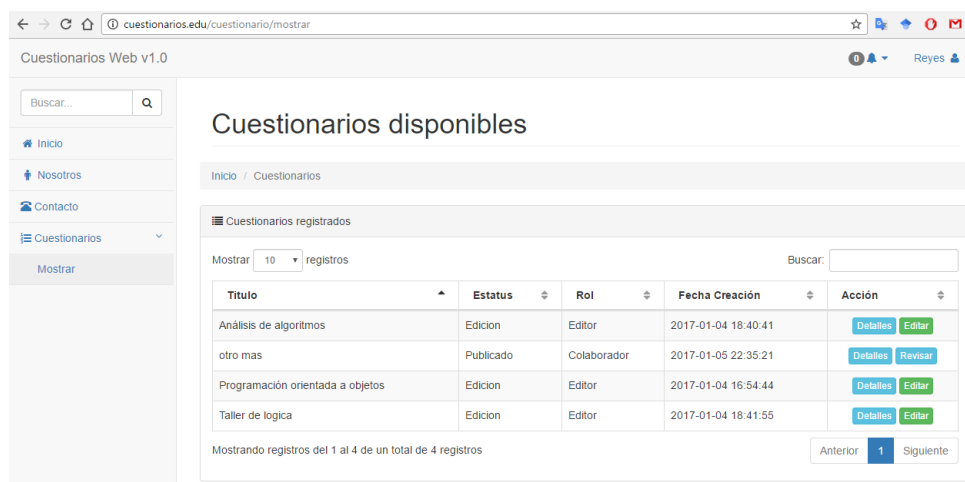


Figura 6.18: Lista de cuestionarios disponibles.

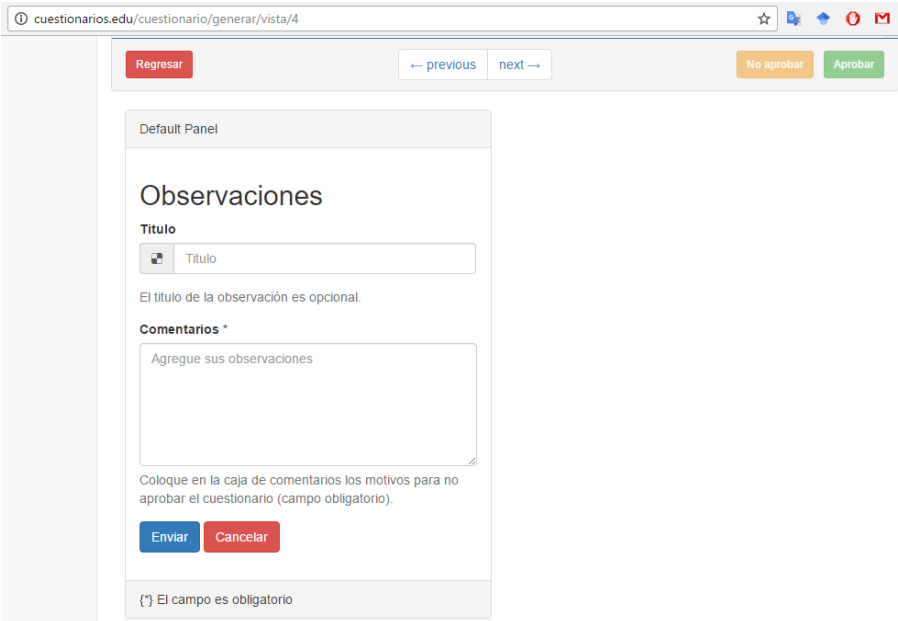
Cuando el colaborador ha completado la revisión del cuestionario puede emitir su voto aprobatorio o dar una observación acerca de este. En la figura 6.19 podemos observar, en la esquina inferior izquierda, los botones de aprobación y no aprobación para el cuestionario. Si el colaborador aprueba el cuestionario el sistema solo confirma la operación y cambia el estatus a *Aprobado*, solo cuando todos los colaboradores lo hayan aprobado, en caso contrario, el sistema despliega en la misma pantalla un formulario para agregar los comentarios.



Figura 6.19: Vista previa para un colaborador.

### 6.1.3.1. Agregar una observación a un cuestionario

Para agregar una observación se tiene que dar click en el botón de “**No aprobar**” como se mencionó en la sección anterior, haciendo esto tenemos la posibilidad de escribir la observación en el formulario de observaciones (figura 6.20), aquí existen dos campos de los cuales solo uno es obligatorio, que es el campo de observaciones o comentarios, el otro campo es opcional y para darle un título general a la observación si se desea. Haciendo la observación se da click en enviar y el sistema regresa a la lista de cuestionarios.



Default Panel

### Observaciones

**Titulo**

El título de la observación es opcional.

**Comentarios \***

Coloque en la caja de comentarios los motivos para no aprobar el cuestionario (campo obligatorio).

**Enviar** **Cancelar**

(\* El campo es obligatorio)

Figura 6.20: Formulario para crear una observación.

Cuando un usuario editor inicie sesión el icono de notificaciones indicará (ver figura 6.21) cuantas observaciones tiene el usuario, si el usuario da click en el icono podrá ser redireccionado a una nueva vista donde se le mostrarán todas las observaciones hechas y a qué cuestionario pertenece. Por ejemplo, esto se ilustra en la figura 6.22, si se presiona el botón **Ver** el sistema despliega el mensaje escrito por el colaborador (figura 6.23).

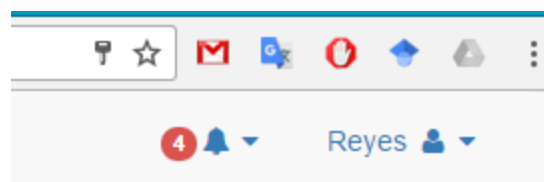


Figura 6.21: Icono indicando que existe una observación.

## Observaciones

Inicio / Observaciones

Observaciones recibidas

Mostrar 10 registros Buscar:

Cuestionario	Estatus	Autor	Fecha Creación	Acción
Base de datos	Visto	Araceli Lillana Reyes	ju. 05 de ene. 2017 22:48 h	<a href="#">Ver</a>

Mostrando registros del 1 al 1 de un total de 1 registros Anterior **1** Siguiente

Figura 6.22: Lista de observaciones.

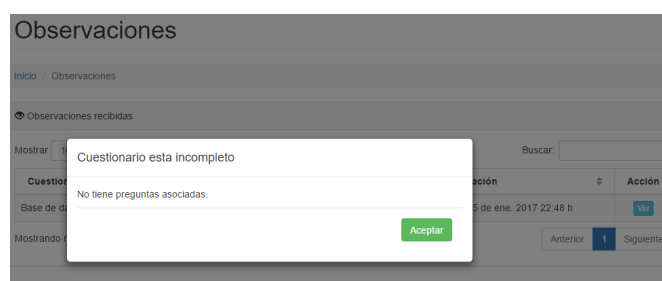


Figura 6.23: Observación hecha por un colaborador vista por un editor.

Finalmente, después de todo el proceso de edición y aprobación el usuario participante puede ver el cuestionario disponible, ya que el cuestionario tendrá ahora un estatus de aprobado.

### 6.1.4. Funcionalidades para el participante

Cuando un participante entre en sesión los cuestionarios disponibles estarán puestos en lista con el más reciente en pila. El participante podrá buscar el cuestionario que desea resolver por medio del campo de búsqueda, en caso de no estar disponible a simple vista.

Existen cuatro escenarios posibles para la aplicación de un cuestionario.

- 1) Que un cuestionario esté caducado. Este caso se presenta cuando un cuestionario a pasado su fecha de aplicación. Para ello el sistema notificará que el cuestionario no puede ser aplicado (figura 6.24).

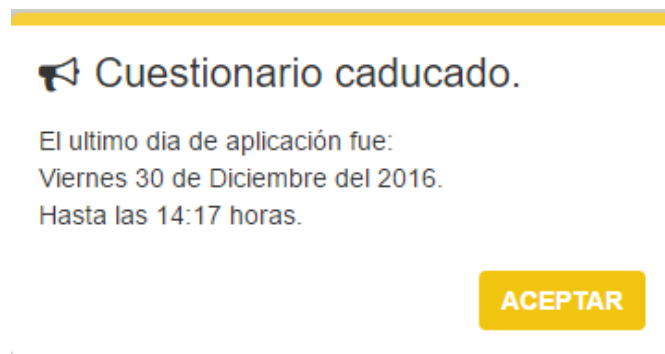


Figura 6.24: Notificación de cuestionar con fecha fuera de aplicación.

- 2) La fecha de aplicación aún no ha llegado. Esto quiere decir que se ha aprobado el cuestionario pero todavía no está en fecha de aplicación por lo que el sistema también notificara al usuario (figura 6.25).

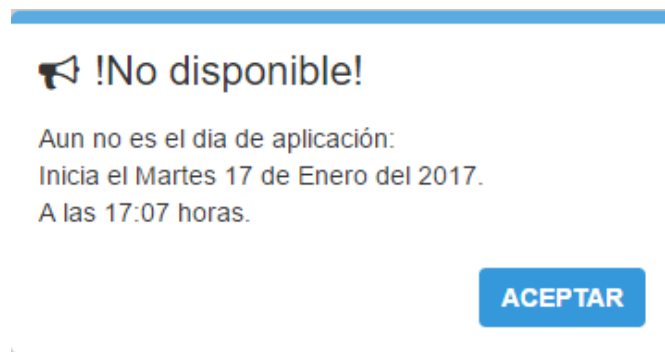


Figura 6.25: Notificación de cuestionario con fecha fuera de aplicación.

- 3) El cuestionario tiene contraseña. El sistema notificará y pedirá la contraseña del cuestionario (figura 6.26), una vez verificada sistema podrá resolver el cuestionario.

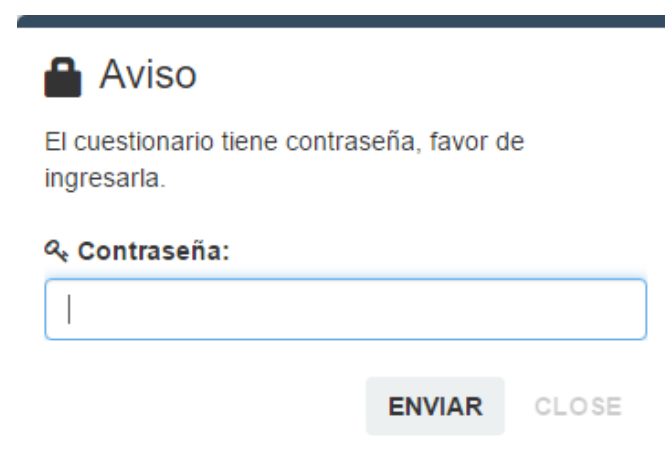


Figura 6.26: Notificación y solicitud de cuestionario.

- 4) Cuestionario libre. Esto ocurrirá cuando ninguna de las anteriores situaciones mencionadas se presenta. En este caso el sistema da paso a la vista donde el usuario participante podrá resolver el cuestionario.

#### 6.1.4.1. Resolver un cuestionario

La vista para resolver el cuestionario es similar a la de un editor o un colaborador, es decir, un participante podrá ver los cuestionarios disponibles en una lista por medio de una tabla, donde existe el control de búsqueda.

En la figura 6.27 se ilustran elementos importantes para resolver un cuestionario, para el siguiente texto refiérase a la figura señalada.

El participante lee la pregunta y selecciona la respuesta que considera correcta. Para moverse entre las preguntas están los botones de Siguiente y Anterior en la parte inferior. Finalmente, cuando se decide dar por terminado el cuestionario se presiona el botón **Terminar**. A continuación el sistema notificará cuántas preguntas han sido respondidas, al mismo tiempo que solicita la confirmación para terminar (figura 6.27). Esto es importante ya que un cuestionario puede quedar con preguntas sin responder y el participante debe estar consciente de ello.



Figura 6.27: Vista para resolver un cuestionario.

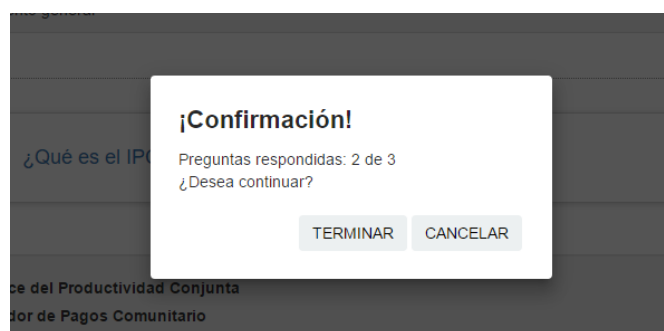


Figura 6.28: Notificación y conficionar de término de cuestionario.

#### 6.1.4.2. Ver los resultados de la aplicación de un cuestionario

Una vez confirmado al sistema que el cuestionario ha sido respondido se muestra en pantalla los resultados obtenidos. Esta información se presenta en un panel con diferentes pestañas. A continuación se habla de cada una de ellas y su contenido correspondiente.

La primer pestaña (figura 6.29) nos muestra la calificación obtenida, si es mayor o igual que siete se toma como aprobado y no aprobado para el caso contrario.



Figura 6.29: Resultados de la aplicación de un cuestionario.

La segunda pestaña proporciona los resultados obtenidos en una tabla con el número de preguntas asociadas a los temas de cada pregunta en el cuestionario, así como los aciertos que se tuvo en ese tema y la suma del puntaje que se obtendría en ese tema y su correspondiente valor en porcentaje, con esto el participante puede inferir el nivel de

manejo de contenidos y habilidades respecto a un tema en particular (figura 6.30).



Figura 6.30: Resultados de la aplicación de un cuestionario.

En tercer lugar tenemos el examen calificado donde se muestran las preguntas y las respuestas que elijo el participante indicando además si fue correcta, incorrecta o si se dejó sin contestar, además se indica el tema de la pregunta, la sección en la que estaba dicha pregunta y el valor que tenía esa pregunta en el cuestionario (figura 6.31).

Información acerca de la aplicación del cuestionario: **cuestionario 1**

Calificación   Resumen temático   Exámen calificado   Información complementaria

Resultados

# ▲	Pregunta	Tema ◆	Tu respuesta	Resultado ◆
1	Identifica cada uno de estos polígonos atendiendo a sus características (nombre, lados, diagonales, área, etc...):	culpa	Trapezio isósceles, Hexágono regular, Octógono irregular	Correcto
2	¿Qué es el IPC?	laborum	Índice de Precios al Consumo	Correcto
3	¿Cuál de los siguientes métodos se puede aplicar a la oxidación del acero?	nam	Químico	Correcto

<< 1 >>




Figura 6.31: Resultados de la aplicación de un cuestionario.

Por último, se muestran los detalles de la aplicación que presentan información como: cuántas preguntas respondió en total, cuántas de éstas fueron acertadas, así como puntos acumulados, el tiempo que le tomo al participante concluir la aplicación del cuestionario y una observación académica que se trata de un mensaje de interpretación generalizado que está en función de la calificación que obtuvo el participante (figura 6.32).

Información acerca de la aplicación del cuestionario: **cuestionario 1**

Calificación    Resumen temático    Exámen calificado    Información complementaria

<b>Detalles</b> <b>Respondiste:</b> 3 de 3 preguntas <b>Aciertos:</b> 3 de 3 preguntas <b>Puntos acumulados:</b> 6 de 6 puntos totales <b>Tiempo ocupado:</b> 0 horas, 0 min, 16 seg	<b>Observación Académica General</b> Obtuviste un puntaje final de 6 puntos de los 6 que podías lograr en este cuestionario, siendo 4.2 el puntaje mínimo para cubrir el cuestionario, demostrando que sabes y sabes hacer lo fundamental de la materia.
--	---

Figura 6.32: Resultados de la aplicación de un cuestionario.

El cualquier momento el usuario participante puede solicitar al sistema su acta cualitativa de resultados, esto haciendo click en el botón que se encuentra en la parte inferior izquierda del panel de resultados (figura 6.33).

Universidad Autónoma de la Ciudad de México **UACM**  
*Nada humano me es ajeno*

**Acta cualitativa de resultados**

Nombre(s) del estudiante: **Virihuana Angeles Perez**      Matrícula: **03-003-0135**  
 Fecha de impresión: **2017-03-10**      Fecha de aplicación: **2017-03-07 13:15:12**

Cuestionario calificado de: CIENCIAS						
#	Pregunta	Tema	Sección	Tu respuesta	Resultado	Puntos
1	Identifica cada uno de estos polígonos atendiendo a sus características (nombre, lados, diagonales, área, etc...):	ad	Matematicas	Trapezio isósceles, Hexágono regular, Octógono irregular	Correcto	1
2	¿Cuál de las siguientes afirmaciones se puede aplicar al troquelado?	defectus	Fisica	Actúa en frío sin cambio significativo de la forma volumétrica de la pieza	Incorrecto	0
3	¿Cuál de los siguientes métodos se puede aplicar a la oxidación del acero?	consequatur	Quimica	Químico	Correcto	3
4	¿Qué es el IPC?	laborum	Quimica	El Índice del Productividad Conjunta	Incorrecto	0
Profesor Diseñador del Cuestionario: <b>Araceli Liliana Reyes Cabello</b>						<b>TOTAL:</b> 4

**CALIFICACIÓN CUANTITATIVA:** 5.71

**Resumen temático**

Temas	# Preguntas asociadas	Aciertos	Puntaje por tema	Puntaje porcentaje
ad	1	1	1	14.29%
defectus	1	0	2	28.57%
consequatur	1	1	3	42.86%
laborum	1	0	1	14.29%
	<b>4</b>	<b>2</b>	<b>7</b>	<b>100%</b>

<b>Detalles</b> <b>Respondiste:</b> 4 de 4 preguntas <b>Aciertos:</b> 2 de 4 preguntas <b>Puntos acumulados:</b> 4 de 7 puntos totales <b>Tiempo ocupado:</b> 0 horas, 0 min, 43 seg	<b>Observación Académica General</b> Obtuviste un puntaje final de 4 puntos de los 7 que podías lograr en el cuestionario, siendo 4.9 el puntaje mínimo para cubrir el cuestionario, demostrando que lo que sabes y sabes hacer de lo fundamental de los temas es insuficiente, así que ahora tu reto será prepararte mejor.
--	---

Página 1 de 1

Figura 6.33: Acta cualitativa de resultados.

---

# Capítulo 7

## Conclusiones

Llegado a este punto podemos decir que se cumplió con el objetivo establecido al inicio de este proyecto puesto que se tiene un prototipo para la creación, aplicación y evaluación de cuestionarios con preguntas de opción múltiple para la UACM, acorde a las especificaciones planteadas por su área de desarrollo de sistemas.

En este apartado hablaré de las experiencias, observaciones y conclusiones obtenidas en cada uno de los pasos que se desarrollaron para alcanzar los objetivos específicos planteados en el [Capítulo 1](#).

El punto de partida fue el levantamiento y análisis de requisitos, sin duda este punto fue el insumo principal en el desarrollo del software prototipo, ya que me permitió delimitar el alcance real del proyecto. En el supuesto de subestimar la gestión de requerimientos, me pudo haber llevado a un re-trabajo de componentes o funcionalidades, porque este proceso es un elemento básico en la construcción de software y, por lo tanto, las tareas implicadas repercuten automáticamente en el resto del desarrollo. Cabe señalar que quien fungió como cliente no fue el usuario final sino el responsable de sistemas, quien previamente había hecho el levantamiento de los requisitos con los profesores (usuarios finales) para implementar los sistemas puestos en marcha hasta el día de hoy, de lo cual no había ninguna documentación. Sin embargo, la información que tenía el responsable de sistemas y su experiencia con el funcionamiento de los sistemas actuales hizo que este paso se acelerara. Por tanto se obtuvo una ventaja para recopilar la información necesaria para establecer la funcionalidad que se quería alcanzar con el sistema.

En cuanto a los cambios en requerimientos que surgían a lo largo del proyecto fueron contemplados en el marco de las funcionalidades principales establecidas en el

documento de requerimientos. Al trabajar con un modelo incremental, los requerimientos se trabajaron al inicio, pero todo pensado con la idea de iteración para posteriormente aumentarlos, corregirlos o redefinirlos. Además la mayoría de los cambios estaban dirigidos a que el sistema fuera más eficiente y/o amigable y totalmente útil para realizar procesos de evaluación de estudiantes.

Por otra parte, se recabó información sobre los dos tipos de preguntas que se necesitaban (abiertas y de opción múltiple) y se hizo el modelado de la base de datos con ambas. Sin embargo, después de calcular la métrica de puntos de función Lite, se determinó realizar un prototipo del sistema que sólo incluyera preguntas de opción múltiple debido a que los resultados obtenidos por la métrica indicaban que se requerían tres personas para completar el sistema en un periodo de cinco meses y medio, considerando un tiempo de trabajo de 20 días laborales y 8 horas por día. Con esto puedo mencionar que el tiempo invertido en el desarrollo del sistema fue aproximadamente de: un mes en el levantamiento de requerimientos, un mes en el modelado de la base de datos y el cálculo de la métrica y seis meses en la implementación (incluyendo la curva de aprendizaje del framework y las pruebas en el proceso). Haciendo una comparación aproximada con la realidad, puedo decir que los resultados de la métrica sobrestimaron el tiempo necesario, ya que si consideramos en la métrica a un solo programador y no a tres como lo establecía el resultado de la métrica, el tiempo de 5.42 meses total de desarrollo se hubiera triplicado a 16.3 meses, suponiendo que el trabajo se repartiera uniformemente entre todo los miembros de desarrollo. Ahora pues, tomando en cuenta los siete meses ocupados en el diseño e implementación del primer prototipo y las preguntas abiertas que no se realizaron, puedo decir el sistema completo estaría listo en aproximadamente 10 meses, si se mantiene el mismo ritmo de trabajo, lo que implica menos tiempo de lo esperado. En mi opinión, esto fue resultado de utilizar un framework ya que este agiliza la implementación. Sin embargo, pienso que para ahondar más en el tópico y no vislumbrar, sería adecuado generar una última métrica para tener una comparativa, entre lo estimado y lo que se obtuvo. Aun con esto, puedo decir que me lleno con experiencia y ahora puedo empezar a construir un registro histórico de proyectos que me pueden ayudar a estimar con mayor exactitud el esfuerzo, tiempo de desarrollo, costo, posibles errores, recursos y tamaño para los nuevos proyectos.

De acuerdo a lo anterior pienso que fue un reto constructivo el desarrollar el prototipo de cuestionarios, primero porque no es lo mismo un equipo de trabajo que desarrolla software, que una persona para todo un proyecto, por lo que me implicó un esfuerzo mayor. En segundo lugar, tomar las teorías vistas a lo largo de esta tesis y aplicarlas

considerando las características específicas del contexto del proyecto, implicó comprender con más énfasis que un conjunto de requerimientos de un sistema computacional es complejo, pero obtenemos la posibilidad de especificar sistemas complejos al documentar especificaciones simples y concisas para el sistema. Y esto fue así con el uso de las historias de usuario y también con el uso de herramientas automatizadas para la administración de requerimientos, como lo fue REM, que sirvió de apoyo en capturar requerimientos, administrarlos y producir una especificación de requisitos.

Del lado al ciclo de vida de prototipos, pude darme cuenta que el enfoque de construir un poco y probar un poco, antes de construir el sistema final, sirve como un mecanismo para la definición de requerimientos ya que después de esto se podrá desarrollar el software real con un enfoque hacia la calidad.

Por otro lado, al inicio del proyecto la elección y aplicación de una metodología no fue fácil porque había que elegir alguna que no fuera demasiado exhaustiva, en cuanto a la documentación que se generara, ya que se requerían ver avances en la programación. Y al mismo tiempo que su adopción implicara un aprendizaje rápido. Además de que sus artefactos y/o herramientas se integraran con facilidad a las tareas a realizar dentro del ciclo de vida del proyecto. En este sentido la opción óptima para el desarrollo fue utilizar la combinación de las metodologías RUP y SCRUM con la idea de desarrollar la aplicación de un modo práctico, tomando de RUP los casos de uso de las funciones principales y de SCRUM se adoptó la forma que plantea de ir implementando las historias de usuario con prioridades. Aunque esta adaptación permitió llevar a buen término el prototipo, al final me di cuenta debido a las características del sistema (casi todas eran funciones básicas del tipo CRUD) y del equipo de desarrollo (un solo integrante) este pudo haberse desarrollado de inicio a fin solo con una metodología ágil, ya que no se requiere una investigación exhaustiva previa, puesto que los procedimientos y/o actividades en la elaboración de un cuestionario son de conocimiento común y se puede plasmar la información en una simple historia de usuario. Lo que me lleva a concluir que la elección de una metodología estará en función de los tamaños y complejidad de lo que se necesita construir y que este resultado puede ser un parámetro más en la elección de una metodología para proyectos futuros.

Por lo que se refiere a roles en una metodología de desarrollo, pienso que hubiera sido interesante contar con un grupo de trabajo y ver como se aplicaban las metodologías presentadas bajo el mismo proyecto. Puesto que los roles sirven para definir quién hace que y son una forma de asignar y definir responsabilidades a personas, un solo individuo no podría entregar una retroalimentación íntegra en este sentido. Sin embargo, queda

claro que cada uno de los roles dentro de una metodología, aportaría al proyecto parte del total necesario para tener éxito en el desarrollo. Además al asignar roles, se definen objetivos y actividades para cada uno. Por tanto, los roles son necesarios para cubrir todas las especificaciones necesarias para cumplir un proceso, pero en mi caso jugar los roles que existen en un proyecto (analista, arquitecto, tester, jefe de proyecto, etcétera) y llevar la metodología, me hubiese implicado más tiempo del contemplado, por ello se habla de un equipo para obtener dicha retroalimentación de una forma más certera.

Otro punto a destacar es mi experiencia en el diseño de la base de datos con el método Case\*Method de Oracle, ya que al modelar los datos con esta técnica resulto más sencillo y ordenado, debido a que los procedimientos garantizan llegar a la tercera forma normal de una forma más transparente, lo que nos lleva a evitar duplicidad de registros, tener una integridad referencial y que el modelo sea más comprensible, lo que permitirá su uso de forma óptima en un futuro. Definitivamente, usar una herramienta como la mencionada es algo que recomiendo, porque es mucho mejor que hacer todo un modelado por partes, es decir, ir de la primera hasta la tercera forma normal paso a paso, para mi contar con el método Case es como tener un método de automatización y ordenamiento de los datos.

Desde la perspectiva del marco de trabajo, Laravel fue pieza clave para la construcción del sistema. Aunque en el área de desarrollo de software de la UACM se cuenta con un marco de trabajo para realizar aplicaciones web, este tiene una arquitectura ad-hoc que no ofrece mucha flexibilidad y documentación para nuevos desarrollos. Al contrario, Laravel fue una excelente herramienta de software para el desarrollo porque además de contar con una documentación bastante completa en la red, dispone o se apoya de elementos como: composer el manejador de dependencias que me ayudó a instalar componentes como los breadcrumbs, el faker para generar datos, dompdf para los reportes, carbon para manejo sencillo de fechas, etcétera, elementos que hicieron el desarrollo más ágil. También está el ORM de Laravel llamado Eloquent (que facilito el manejo de los datos) y el sembrador de datos (seeder) que junto al componente faker se pudieron insertar registros en la base datos para simular varios usuarios en el sistema sin tener que crearlos manualmente y por último el sistema de plantillas blade, el cual me permitió generar HTML dinámico con una sintaxis mucho más limpia. Estos elementos hicieron que el proyecto fluyera con éxito, además me permitieron centrarme en programar la aplicación y abstraer el problema más fácilmente, enfocándome en resolver o programar los requisitos relacionadas con el prototipo, evitando así cuestiones o aspectos técnicos, lo que se convirtió en ahorro de tiempo y trabajo. En conclusión, Laravel hizo el proceso de desarrollo más amigable sin sacrificar la funcionalidad de la aplicación y permitió llevar a cabo el proyecto sin tener

que reescribir código, evitando así “reinventar la rueda”, consiguiendo de esta forma que el trabajo fuese eficiente en tiempo y esfuerzo. Y por otro lado, establece una arquitectura del sistema, la cual será más clara para otros desarrolladores, facilitando el mantenimiento del sistema.

Finalmente el prototipo cumple el objetivo de crear un sistema en línea y hace que se resuelva la problemática de tener que instalar máquina por máquina en los laboratorios del plantel, pero además crea una plataforma de mejoramiento y crecimiento del sistema de cuestionarios, ya que en el futuro el prototipo se convertirá en un sistema cada vez más robusto y flexible, para futuras características que se irán añadiendo o mejorando a través del tiempo.

## 7.1. Trabajos futuros y perspectivas del producto

Como hemos mencionado anteriormente el resultado obtenido es un prototipo con funcionalidad limitada, lo que significa darle continuidad a la aplicación para que esta vaya evolucionando conforme a las necesidades institucionales de la UACM. En este apartado se presentan posibles percepciones para el futuro de este trabajo, las cuales tendrán que ser estudiadas y planteadas para desarrollarse. Algunas de las posibles extensiones de este desarrollo, para incluir un mayor número de funcionalidades, las constituyen:

- Permitir que se tenga un enfoque dirigido hacia los dispositivos móviles, manteniendo una sesión de usuario con el servidor de la universidad.
- El manejo y uso de información que tendrá el sistema, al relacionarse con la base de datos institucional, esto con el fin práctico de dar alcance a toda la comunidad universitaria. Los usuarios profesores y personal administrativo se conectarán a la aplicación por medio del sistema de información de la UACM (<https://isi.uacm.edu.mx/>) y los estudiantes a través del sistema para estudiantes (<https://estudiantes.uacm.edu.mx/>).
- Otra cosa a considerar es que además de los dos tipos de preguntas ya mencionados (preguntas abiertas y preguntas de opción múltiple) se pueden incluir otros tipos como:

1. Preguntas de relación de columnas.

- Las preguntas de relacionar columnas (emparejamiento) tienen un área de contenido y una lista de nombres o de oraciones que deben de hacerse coincidir correctamente contra otra lista de nombres o de oraciones.
  - 2. Preguntas de rellenar el espacio en blanco:
    - Se deja un espacio para anotar la respuesta que conteste la pregunta.
  - 3. Preguntas de ordenación.
    - Se utilizan para dar un orden lógico o cronológico a una serie de pasos o eventos.
- Como parte de las preguntas abiertas también es indispensable el formato de indicadores para entrega de actas cualitativas relacionadas con las preguntas abiertas.
  - También se tienen pensado el uso de un concepto llamado “Pregunta popular”, ya que con el tiempo las preguntas contenidas en el baúl se irán acumulando lo que producirá que algunas preguntas sólo se usen una vez y otras por el contrario se usen constantemente, esto provocaría que un cuestionario contenga preguntas que un participante conozca por repetición y el cuestionario pierda valor en términos de conocer el verdadero avance académico del participante. Por lo que se propone dar a conocer el top de las preguntas más usadas o populares.
  - Con los datos almacenados se podrá hacer análisis estadísticos sobre los resultados obtenidos y el tiempo de respuesta de los estudiantes. El sistema podrá proporcionar evidencia del aprovechamiento del estudiante, es decir, datos que posteriormente se analicen y se puedan encontrar, por ejemplo, patrones de desempeño en ciertos temas, y saber fortalezas y debilidades que un estudiante tiene recurrentemente y con esto mejorar los contenidos temáticos impartidos.
  - El objetivo a la hora de programar cada una de la funcionalidades fue buscar la usuabilidad para darle al usuario un sistema fácil de entender y una vez que se familiarice con el software habría que realizar pruebas de aceptación para el usuario final ya que, familiarizándose, este tendrá más libertad sobre la ejecución y puede evaluarlo en relación con sus propias expectativas y necesidades.
  - Finalmente, cuando el sistema esté totalmente completo, se deberá realizar capacitación al personal involucrado en el proceso de creación y edición de un cuestionario.

---

# Anexos

---

## Anexo A

### Requisitos y casos de uso

A continuación se enumeran los requerimientos para el sistema de cuestionarios, en el que se incluyen los casos de uso más relevantes para el sistema. Los requisitos aquí expuestos fueron obtenidos a partir de entrevistas con el responsable del área de desarrollo de software del plantel San Lorenzo Tezonco de la UACM y capturados con la herramienta de software libre REM. Los casos de uso describen los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso dentro de la aplicación cuestionarios web. Según un estándar como el IEEE 830, un documento de requerimientos se compone de un propósito, alcance, perspectiva del producto entre otros elementos, pero varios de estos fueron ya descritos a lo largo de este trabajo. Por simplicidad se hace una versión resumida del listado de los requerimientos del sistema.

Proyecto Cuestionarios Web

# Documento de Requisitos del Sistema

Versión 0.8

Fecha 05/05/2016

Preparado para:

[Área de desarrollo de sistemas](#)

[Universidad Autónoma de la Ciudad de México](#)

Preparado por:

[Alberto Carlos Pérez Cadena](#)

---

## Índice

- 1 [Introducción](#)
  - 2 [Participaciones del Proyecto](#)
  - 3 [Objetivos del sistema](#)
  - 4 [Catálogo de requisitos del sistema](#)
    - 4.1 [Requisitos generales](#)
    - 4.2 [Requisitos específicos](#)
      - 4.2.1 [Requisitos funcionales](#)
      - 4.2.2 [Requisitos no funcionales](#)
  - 5 [Glosario de términos](#)
    - 5.1 [Estatus](#)
    - 5.2 [Usuarios](#)
-

## 1 Introducción

La presente especificación de requisitos de software (SRS) es un documento que recoge toda la documentación correspondiente al análisis de una aplicación, es decir, contiene información para el desarrollador de software y constituye un informe útil para otros involucrados en el producto final, describiendo lo que realmente se desea obtener del sistema y de esta manera tener un documento cuya información servirá para establecer un diseño que se ajuste a las necesidades expuestas del usuario.

## 2 Participaciones del Proyecto

<b>Organización</b>	<b>Universidad Autónoma de la Ciudad de México</b>
<b>Dirección</b>	Calle Prolongación San Isidro No. 151, Colonia San Lorenzo Tezonco, Delegación Iztapalapa, Ciudad de México, C.P. 09790
<b>Teléfono</b>	Tel. 5850-1901
<b>Fax</b>	N/A
<b>Comentarios</b>	¡Nada humano me es ajeno!

<b>Organización</b>	<b>Área de desarrollo de sistemas</b>
<b>Dirección</b>	San Lorenzo Tezonco
<b>Teléfono</b>	ext. 13206
<b>Fax</b>	N/A
<b>Comentarios</b>	Ninguno

<b>Organización</b>	<b>Alberto Carlos Pérez Cadena</b>
<b>Dirección</b>	humanoajeno@gmail.com
<b>Teléfono</b>	55-20308003
<b>Fax</b>	N/A
<b>Comentarios</b>	Ninguno

<b>Participante</b>	<b>Octavio León López</b>
<b>Organización</b>	<a href="#">Área de desarrollo de sistemas</a>
<b>Rol</b>	Product Owner, Stakeholders, Manager
<b>Es desarrollador</b>	No
<b>Es cliente</b>	Sí
<b>Es usuario</b>	Sí
<b>Comentarios</b>	Ninguno

<b>Participante</b>	Alberto Carlos Pérez Cadena
<b>Organización</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Rol</b>	Team
<b>Es desarrollador</b>	Sí
<b>Es cliente</b>	No
<b>Es usuario</b>	No
<b>Comentarios</b>	Ninguno

<b>Participante</b>	Liliana Reyes Cabello
<b>Organización</b>	<a href="#">Universidad Autónoma de la Ciudad de México</a>
<b>Rol</b>	Directora
<b>Es desarrollador</b>	No
<b>Es cliente</b>	No
<b>Es usuario</b>	Sí
<b>Comentarios</b>	Maestra en Ciencias de la computación

### 3 Objetivos del sistema

<b>OBJ-0001</b>	<b>Objetivo principal</b>
<b>Versión</b>	1.0 ( 06/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Liliana Reyes Cabello</a>
<b>Descripción</b>	El sistema deberá <i>permitir crear, aplicar y calificar cuestionarios en línea.</i>
<b>Subobjetivos</b>	<b>[OBJ-0002] Editar:</b> El sistema deberá <i>permitir editar un cuestionario.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	en construcción
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

## 4 Catálogo de requisitos del sistema

### 4.1 Requisitos generales

<b>FRQ-0013</b>	<b>Aplicaciones</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	<ul style="list-style-type: none"> <li><a href="#">[IRQ-0001] Información del cuestionario</a></li> </ul>
<b>Descripción</b>	El sistema deberá <i>permitir que un cuestionario tenga un lugar de aplicación</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	en construcción
<b>Estabilidad</b>	media
<b>Comentarios</b>	El lugar de aplicación tendrá dos opciones: aula o externo. La opción de aula se utilizará dentro de los planteles y se hará reconocimiento de la dirección IP de la computadora así como el nombre de la máquina de donde se esté ejecutando el cuestionario, con la intención de evitar que se ejecute en otras máquinas y evitar duplicidad.

<b>FRQ-0016</b>	<b>Aplicado por discriminación</b>		
<b>Versión</b>	1.0 ( 07/05/2016 )		
<b>Autores</b>	<a href="#">Octavio León López</a>		
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>		
<b>Dependencias</b>	Ninguno		
<b>Descripción</b>	<p>El sistema deberá <i>permitir a que un cuestionario pueda ser aplicado por discriminación a:</i></p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> <i>o Estudiantes</i>   <i>" Periodo de inscripción</i>  <i>" Colegio</i>  <i>" Academia</i>  <i>" Curso</i>  <i>" Grupo</i>  <i>" Carrera</i>  <i>" Plantel</i> </td> <td style="width: 50%; vertical-align: top;"> <i>o Administrativos</i>   <i>" Por área</i>  <i>" Nivel</i>  <i>" Plantel</i> </td> </tr> </table>	<i>o Estudiantes</i>  <i>" Periodo de inscripción</i> <i>" Colegio</i> <i>" Academia</i> <i>" Curso</i> <i>" Grupo</i> <i>" Carrera</i> <i>" Plantel</i>	<i>o Administrativos</i>  <i>" Por área</i> <i>" Nivel</i> <i>" Plantel</i>
<i>o Estudiantes</i>  <i>" Periodo de inscripción</i> <i>" Colegio</i> <i>" Academia</i> <i>" Curso</i> <i>" Grupo</i> <i>" Carrera</i> <i>" Plantel</i>	<i>o Administrativos</i>  <i>" Por área</i> <i>" Nivel</i> <i>" Plantel</i>		
<b>Importancia</b>	importante		
<b>Urgencia</b>	inmediatamente		
<b>Estado</b>	pendiente de verificación		
<b>Estabilidad</b>	baja		
<b>Comentarios</b>	Ninguno		

<b>FRQ-0021</b>	<b>Bitácora de modificaciones</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>contar con una bitácora de modificaciones de preguntas y cuestionarios.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	en construcción
<b>Estabilidad</b>	media
<b>Comentarios</b>	<p>Los datos a guardar deben dar a conocer la información siguiente:</p> <p>" ¿Quién hizo la modificación?  " ¿A qué fecha y hora?  "¿Qué acción realizó (borrar, modificar, crear)?</p> <p>Por ejemplo: El sistema debe permitir registrar la última modificación que se realizó a un cuestionario y guardar el registro de qué usuario (<a href="#">administrador</a> o <a href="#">editor</a>) hizo la modificación.</p>

<b>FRQ-0015</b>	<b><a href="#">Borrado</a> de cuestionarios</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>
<b>Dependencias</b>	• <a href="#">[CRQ-0002] Borrado lógico de cuestionario</a>
<b>Descripción</b>	El sistema deberá <i>permitir borrar cuestionarios.</i>
<b>Importancia</b>	quedaría bien
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Recordar que el <a href="#">borrado</a> será lógico.

<b>FRQ-0020</b>	<b>Colección de cuestionarios</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>

<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>contener una colección de cuestionarios. De tal manera que permitirá tomar un cuestionario ya existente y "clonarlo". Cualquier usuario, con derecho de <a href="#">edición</a>, puede hacerlo.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	en construcción
<b>Estabilidad</b>	media
<b>Comentarios</b>	Este requisito atiende la necesidad de no crear un examen desde cero, es decir, si un <a href="#">editor</a> requiere tener un examen ya hecho, lo puede tomar de la colección de cuestionarios.

<b>FRQ-0003</b>	<b>Derechos <a href="#">administrador</a></b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• <a href="#">[FRQ-0002] Perfiles y roles de usuario</a></li> </ul>
<b>Descripción</b>	<p>El sistema deberá <i>permitir a los usuarios con derechos de <a href="#">administrador</a> poder acceder a las funciones administrativas y además contará con privilegios para hacer las siguientes acciones:</i></p> <p>" <i>CRUD de cuestionarios.</i>  " <i>CRUD de preguntas.</i>  " <i>CRUD de usuarios.</i></p>
<b>Importancia</b>	quedaría bien
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	El <a href="#">administrador</a> , no tendrá la posibilidad de responder un cuestionario bajo este rol, sólo podrá responder si lo hace como un administrativo de la comunidad de la UACM.

<b>FRQ-0031</b>	<b>Evaluar <a href="#">PA</a></b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	<p>El sistema deberá <i>permitir que un usuario <a href="#">evaluador</a> pueda leer las respuestas de las preguntas abiertas (<a href="#">PA</a>) que existan en un cuestionario y evaluará la respuesta mediante:</i></p> <p><i>" Una escala de evaluación, con n indicadores.</i></p> <p><i>" Un valor.</i></p> <p><i>" Y podrá o no hacer un comentario.</i></p>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	<p>Los indicadores serán definidos por los colaboradores y el <a href="#">editor</a>, así como el valor de cada uno.</p> <p>Ejemplo: Indicador. Ortografía: destacado igual a 10 puntos; suficiente igual a ocho; insuficiente igual a cero.</p>

<b>FRQ-0010</b>	<b>Habilitación de usuarios</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Octavio León López</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir al <a href="#">administrador</a> habilitar o deshabilitar usuarios.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Ninguno

<b>FRQ-0023</b>	<b>Modificar estatus</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• <a href="#">[FRQ-0014] Estatus de un cuestionario</a></li> </ul>
<b>Descripción</b>	El sistema deberá <i>permitir cambiar el estatus de un cuestionario, cuando su estatus sea <a href="#">aprobado</a> y se solicite una corrección.</i>
<b>Importancia</b>	quedaría bien
<b>Urgencia</b>	puede esperar
<b>Estado</b>	en construcción
<b>Estabilidad</b>	media
<b>Comentarios</b>	Sólo el <a href="#">administrador</a> , podrá realizar esta operación. El cambio de estatus, no podrá ser realizado, si se ha contestado una o más preguntas de dicho cuestionario.

## 4.2 Requisitos específicos

### 4.2.1 Requisitos funcionales

#### 4.2.1.1 Modelado de casos de uso

##### 4.2.1.1.1 Definición de actores

<b>ACT-0001</b>	<b><a href="#">Administrador</a></b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>
<b>Descripción</b>	Este actor representa <i>la persona que tendrá la responsabilidad de configurar, mantener, monitorizar, documentar y asegurar el correcto funcionamiento del sistema informático, así como crear cuestionarios y agregarles colaboradores, según sea necesario.</i>
<b>Comentarios</b>	Ninguno

<b>ACT-0002</b>	<b><a href="#">Editor</a> (Diseñador)</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Liliana Reyes Cabello</a>
<b>Descripción</b>	Este actor representa <i>la persona que construirá el cuestionario insertado las preguntas con sus respectivas respuestas.</i>
<b>Comentarios</b>	Un usuario <a href="#">editor</a> , podrá ser un profesor o personal administrativo que pertenezca a la UACM.

<b>ACT-0003</b>	<b><a href="#">Colaborador</a> (revisor)</b>
<b>Versión</b>	1.0 ( 08/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Liliana Reyes Cabello</a>
<b>Descripción</b>	Este actor representa <i>la persona que se encargara de revisar la integridad del cuestionario y podrá hacer comentarios acerca de este, además podrá dar voto de aprobación, para que el cuestionario pueda pasar a un estado de <a href="#">aprobado</a>.</i>
<b>Comentarios</b>	Un usuario <a href="#">colaborador</a> , podrá ser un profesor o personal administrativo que pertenezca a la UACM.

<b>ACT-0004</b>	<b><a href="#">Participante</a></b>
<b>Versión</b>	1.0 ( 08/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Liliana Reyes Cabello</a>
<b>Descripción</b>	Este actor representa <i>es el actor al que se le aplican los cuestionarios.</i>
<b>Comentarios</b>	Un <a href="#">participante</a> puede ser un estudiante o algún empleado de la UACM.

<b>ACT-0005</b>	<a href="#">Evaluador</a>
<b>Versión</b>	1.0 ( 08/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Liliana Reyes Cabello</a>
<b>Descripción</b>	Este actor representa <i>la persona delegada a revisar cuestionarios con preguntas abiertas.</i>
<b>Comentarios</b>	Un usuario <a href="#">Evaluador</a> , podrá ser un profesor o personal administrativo que pertenezca a la UACM.

#### 4.2.1.1.2 Casos de uso del sistema

##### 4.2.1.1.2.1 Administrador

<b>UC-0001</b>	<b>Creación de cuestionario</b>	
<b>Versión</b>	1.0 ( 15/04/2017 )	
<b>Autores</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El sistema debe permitir la creación de cuestionarios permitiendo sólo a un usuario <a href="#">administrador</a> crearlos.</i>	
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El actor debe haber iniciado sesión.</li> <li>2. El futuro <a href="#">editor</a> y los posibles colaboradores deben estar dados de alta como profesores de la UACM.</li> <li>3. Un profesor hace una petición al <a href="#">administrador</a> para crear un cuestionario.</li> </ol>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Administrador (ACT-0001)</a> <i>solicita la creación del cuestionario.</i>
	2	El sistema <i>muestra un formulario para introducir los datos del cuestionario.</i>
	3	El actor <a href="#">Administrador (ACT-0001)</a> <i>introduce los datos de cuestionario al sistema, proporcionados por el profesor y coloca el nombre del profesor solicitante como <a href="#">editor</a>.</i>
	4	El sistema <i>solicita ingresar la información de la fecha de aplicación del cuestionario.</i>
	5	El actor <a href="#">Administrador (ACT-0001)</a> <i>ingresa la fecha.</i>
	6	El actor <a href="#">Administrador (ACT-0001)</a> <i>da la acción de crear cuestionario.</i>
	7	El sistema <i>comprueba la validez de los datos y los almacena.</i>
<b>Postcondición</b>	Se tendrá un cuestionario con título, descripción, una sección por default, una fecha de aplicación y una contraseña opcional, además de contar con o sin colaboradores.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si <i>el profesor decide introducir usuarios colaboradores</i> , el actor <a href="#">Administrador (ACT-0001)</a> <i>solicita los nombres. El sistema asocia cada uno de los nombres al cuestionario, a continuación este caso de uso continúa</i>
	3	Si <i>el profesor decide colocar una contraseña</i> , el actor <a href="#">Administrador (ACT-0001)</a> <i>actor ingresa la contraseña. El sistema comprueba la redundancia de la contraseña, a continuación este caso de uso continúa</i>
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	7	17 segundo(s)
<b>Frecuencia esperada</b>	2 veces por día(s)	
<b>Importancia</b>	vital	
<b>Urgencia</b>	inmediatamente	
<b>Estado</b>	pendiente de verificación	
<b>Estabilidad</b>	media	
<b>Comentarios</b>	Al cuestionario se le podrán agregar un máximo de 5 colaboradores. El profesor que solicita el cuestionario, puede pedir a otro usuario como <a href="#">editor</a> y no ser él exactamente el <a href="#">editor</a> .	

#### 4.2.1.1.2.2 Editor

<b>UC-0002</b>	<b>Editar y publicar un cuestionario</b>	
<b>Versión</b>	1.0 ( 15/04/2017 )	
<b>Autores</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando un usuario requiera editar un cuestionario, para lo cual podrá insertar preguntas y secciones. Cuando la <a href="#">edición</a> del cuestionario termine el usuario podrá publicar el cuestionario.</i>	
<b>Precondición</b>	1. El usuario debe haber iniciado sesión. 2. Que el <a href="#">administrador</a> haya creado un cuestionario y que en este se tenga indicado que el usuario actual es un <a href="#">Editor</a> .	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Administrador (ACT-0001)</a> muestra los cuestionarios en lista para su posible <a href="#">edición</a> .
	2	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> elige el cuestionario.
	3	El sistema presenta una vista con los controles necesarios para inserción de secciones, preguntas de baúl y preguntas nuevas, eligiendo una de estas opciones en el momento que el actor lo decida.
	4	Si el usuario elige agregar una nueva pregunta, se realiza el caso de uso <a href="#">Insertar nueva pregunta de baúl (UC-0003)</a>
	5	Si el usuario elige agregar una pregunta de baúl, se realiza el caso de uso <a href="#">Insertar nueva pregunta de baúl (UC-0003)</a>
	6	Si el usuario elige agregar una pregunta de baúl, se realiza el caso de uso <a href="#">Insertar una sección a un cuestionario (UC-0005)</a>
	7	El sistema comprueba (para los tres pasos anteriores) la validez de los datos y los almacena.
	8	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> elige la opción para publicar el cuestionario.
	9	El sistema solicita confirmación.
	10	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> decide si aprueba o rechaza la publicación.
	11	El sistema cambia el estado del cuestionario a <a href="#">publicado</a> .
<b>Postcondición</b>	Se tiene un cuestionario editado y con cambio en su estado ya sea <a href="#">publicado</a> o <a href="#">aprobado</a> .	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	11	Si el cuestionario no tiene colaboradores, el sistema colocara el estado del cuestionario en <a href="#">aprobado</a> ., a continuación este caso de uso continúa
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	3	17 segundo(s)

<b>Frecuencia esperada</b>	1 veces por día(s)
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

<b>UC-0003</b>	<b>Insertar nueva pregunta de baúl</b>	
<b>Versión</b>	1.0 ( 15/04/2017 )	
<b>Autores</b>	<a href="#">Octavio León López</a>	
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>Este caso de uso describe como un usuario <a href="#">editor</a> agrega preguntas del baúl de preguntas.</i> o durante la realización de los siguientes casos de uso: <a href="#">[UC-0002] Editar y publicar un cuestionario</a>	
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario <a href="#">editor</a> debe estar dado de alta en el sistema.</li> <li>2. El usuario debe haber iniciado sesión.</li> <li>3. El usuario <a href="#">editor</a> se encuentre editando un cuestionario y ha solicitado al sistema agregar pregunta de baúl.</li> </ol>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> <i>solicita una nueva pregunta de baúl, en función de una sección existente.</i>
	2	El sistema <i>muestra una lista con las preguntas disponibles.</i>
	3	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> <i>selecciona una o más preguntas de la lista y elige "Agregar" al cuestionario.</i>
	4	El sistema <i>asocia la(s) pregunta(s) al cuestionario guardando la asociación junto a la sección correspondiente y muestra la pregunta en el cuestionario.</i>
<b>Postcondición</b>	Una pregunta de baúl insertada en el cuestionario en <a href="#">edición</a> .	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	<i>Si la actividad seleccionada es "Cancelar" , el sistema deja de mostrar la lista y no hay efecto en el cuestionario., a continuación este caso de uso queda sin efecto</i>
	3	<i>Si una pregunta del baúl ya se encuentra en el cuestionario., el sistema informa al usuario de que la pregunta ya existe en el cuestionario y no agrega dicha(s) pregunta(s)., a continuación este caso de uso continúa</i>
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	2	10 segundo(s)
<b>Frecuencia esperada</b>	10 veces por día(s)	
<b>Importancia</b>	importante	
<b>Urgencia</b>	inmediatamente	
<b>Estado</b>	pendiente de verificación	
<b>Estabilidad</b>	media	
<b>Comentarios</b>	Todas las preguntas añadidas de baúl no tendrán la opción de eliminar pregunta.	

<b>UC-0004</b>	<b>Insertar nueva pregunta de opción múltiple</b>	
<b>Versión</b>	1.0 ( 15/04/2017 )	
<b>Autores</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>Este caso de uso describe como un usuario <a href="#">editor</a> inserta una nueva pregunta de opción múltiple en un cuestionario que está en <a href="#">edición</a>.</i>	
<b>Precondición</b>	1.El usuario debe tener privilegios de <a href="#">editor</a> para el cuestionario. 2.El cuestionario en la sesión vigente debe tener un estatus de <a href="#">edición</a> . 3.El usuario <a href="#">editor</a> ha solicitado al sistema agregar nueva pregunta de opción múltiple.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> <i>solicita una nueva pregunta, en función de una sección existente.</i>
	2	El sistema <i>muestra el formulario para agregar una nueva pregunta, con los campos: pregunta, el puntaje, texto asociado y las respuestas.</i>
	3	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> <i>llena los campos.</i>
	4	El sistema <i>comprueba la validez de los datos y los almacena, mostrando posteriormente la pregunta en el cuestionario, según la sección que se haya indicado.</i>
<b>Postcondición</b>	Se tiene una nueva pregunta de opción múltiple asociada a una sección y al cuestionario en <a href="#">edición</a> .	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si <i>el usuario desea agregar más respuestas.</i> , el sistema <i>podrá agregar hasta cinco respuestas más.</i> , a continuación este caso de uso <i>continúa</i>
	2	Si <i>el usuario selecciona eliminar una respuesta</i> , el actor <a href="#">Editor (Diseñador) (ACT-0002)</a> <i>confirma la eliminación de la respuesta.</i> , a continuación este caso de uso <i>continúa</i>
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	2	17 segundo(s)
<b>Frecuencia esperada</b>	10 veces por día(s)	
<b>Importancia</b>	importante	
<b>Urgencia</b>	inmediatamente	
<b>Estado</b>	pendiente de validación	
<b>Estabilidad</b>	media	
<b>Comentarios</b>	1.- El texto asociado será opcional. 2.- Todas las preguntas añadidas tendrán la opción de eliminar dicha pregunta.	

<b>UC-0005</b>	<b>Insertar una sección a un cuestionario</b>	
<b>Versión</b>	1.0 ( 15/04/2017 )	
<b>Autores</b>	<a href="#">Liliana Reyes Cabello</a>	
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Liliana Reyes Cabello</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>Este caso de uso describe como un usuario <a href="#">editor</a> inserta una sección en un cuestionario con estado de <a href="#">edición</a>.</i> o durante la realización de los siguientes casos de uso: <a href="#">[UC-0002] Editar y publicar un cuestionario</a>	
<b>Precondición</b>	1. El usuario <a href="#">editor</a> debe estar dado de alta en el sistema. 2. El usuario debe haber iniciado sesión. 3. El usuario <a href="#">editor</a> se encuentre editando un cuestionario y ha solicitado al sistema agregar una nueva sección.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> solicita al sistema agregar nueva sección.
	2	El sistema muestra en el formulario que contiene información de registro que debe ser llenada por el usuario, lo cual incluye nombre de la sección y una descripción opcional.
	3	El actor <a href="#">Editor (Diseñador) (ACT-0002)</a> introduce los datos y solicita agregar sección.
	4	El sistema comprueba la validez de los datos y asigna la sección al cuestionario en <a href="#">edición</a> .
<b>Postcondición</b>	Se tiene una nueva sección asociada al cuestionario con opciones para continuar la <a href="#">edición</a> .	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si la actividad seleccionada es "Salir" ., el sistema Se descarta el formulario para el cuestionario, a continuación este caso de uso queda sin efecto
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	2	7 segundo(s)
	4	5 segundo(s)
<b>Frecuencia esperada</b>	10 veces por día(s)	
<b>Importancia</b>	importante	
<b>Urgencia</b>	inmediatamente	
<b>Estado</b>	pendiente de verificación	
<b>Estabilidad</b>	media	
<b>Comentarios</b>	Cada sección agregada al cuestionario contendrá las opciones de modificar sección, agregar nueva pregunta, agregar pregunta de baúl, así como la opción de eliminar sección.	

#### 4.2.1.1.2.3 Participante

<b>UC-0006</b>	<b>Resolver un cuestionario</b>	
<b>Versión</b>	1.0 ( 15/04/2017 )	
<b>Autores</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>Este caso de uso describe como un usuario <a href="#">participante</a> resuelve un cuestionario que está en estado de <a href="#">aprobado</a>.</i>	
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario debe haber iniciado sesión.</li> <li>2. Que el cuestionario a resolver este con estado de <a href="#">aprobado</a>.</li> <li>3. Que la fecha actual este dentro del intervalo de la fecha de aplicación.</li> </ol>	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El sistema <i>muestra la lista de cuestionarios disponibles</i>
	2	El actor <a href="#">Participante (ACT-0004)</a> <i>elige el cuestionario a resolver</i>
	3	El sistema <i>muestra la vista del cuestionario seleccionado</i>
	4	El actor <a href="#">Participante (ACT-0004)</a> <i>responde las preguntas, pudiendo dejar alguna sin responder</i>
	5	El sistema <i>captura las repuestas del usuario</i>
	6	El actor <a href="#">Participante (ACT-0004)</a> <i>confirma al sistema que ha a concluido el cuestionario</i>
	7	El sistema <i>informa el número de preguntas respondidas, si usuario decide puede cancelar y corregir o responder alguna pregunta</i>
	8	El actor <a href="#">Participante (ACT-0004)</a> <i>confirma la terminación del cuestionario</i>
	9	El sistema <i>muestra la calificación obtenida</i>
<b>Postcondición</b>	Se tiene un cuestionario <a href="#">resuelto</a> , con resultados para usuario y con un estatus de <a href="#">resuelto</a> .	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	<i>Si el cuestionario tiene contraseña, el sistema solicita la contraseña. El actor ingresa la contraseña. El sistema comprueba la contraseña, si esta es correcta continua con el paso tres, si no se cancela el caso de uso, a continuación este caso de uso continúa</i>
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	3	15 segundo(s)
<b>Frecuencia esperada</b>	30 veces por mes(es)	
<b>Importancia</b>	importante	
<b>Urgencia</b>	inmediatamente	
<b>Estado</b>	pendiente de validación	
<b>Estabilidad</b>	media	
<b>Comentarios</b>	Ninguno	

#### 4.2.1.2 Cuestionarios

<b>FRQ-0012</b>	<b>Control de estudiantes o grupos</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• <a href="#">[FRQ-0004] Crear cuestionario</a></li> </ul>
<b>Descripción</b>	El sistema deberá <i>permitir colocar una contraseña opcional.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Esta contraseña se colocará para que, si se desea, sólo un grupo de participantes en particular pueda resolver el cuestionario, ya que a estos se les otorgara la contraseña para resolverlo.

<b>FRQ-0014</b>	<b>Estatus de un cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Liliana Reyes Cabello</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Liliana Reyes Cabello</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir colocar un estatus a los cuestionarios.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	validado
<b>Estabilidad</b>	media
<b>Comentarios</b>	<p>El cuestionario tendrá los siguientes estatus:</p> <ul style="list-style-type: none"> <li>* <a href="#">Edición.</a></li> <li>* <a href="#">Publicado.</a></li> <li>* <a href="#">Aprobado.</a></li> <li>* <a href="#">Resuelto</a></li> <li>* <a href="#">Borrado.</a></li> </ul>

<b>FRQ-0017</b>	<b>Caducidad de cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir, al <a href="#">administrador</a> como al <a href="#">editor</a>, agregar una caducidad o vigencia a los cuestionarios</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

<b>FRQ-0018</b>	<b>Edición de cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Liliana Reyes Cabello</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>contener un módulo para la creación de preguntas de diferente tipo, que incluyan un mini <a href="#">editor</a> de texto para tener un control sobre el formato del texto.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	hay presión
<b>Estado</b>	validado
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

<b>FRQ-0019</b>	<b>Agregar preguntas de la colección</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir a un <a href="#">editor</a> agregar preguntas existentes, de la base de</i>

	<i>preguntas, al cuestionario.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

<b>FRQ-0024</b>	<b>Modificación de un cuestionario en estado <a href="#">resuelto</a></b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>restringir que un cuestionario sea modificado por ningún usuario si este ya fue contestado.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Excepción.  El <a href="#">administrador</a> podrá modificar el cuestionario <a href="#">resuelto</a> , si y sólo si la modificación no afecte la calificación del estudiante.

#### 4.2.1.3 Preguntas

<b>FRQ-0027</b>	<b>Baúl de preguntas</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir que todas las preguntas nuevas se agregarán a la base general de preguntas.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Ninguno

<b>FRQ-0028</b>	<b><a href="#">Borrado</a> de pregunta</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir, el <a href="#">borrado</a> físico de las preguntas que sólo pertenezcan a un cuestionario, es decir, que no pertenezcan al baúl de preguntas, sólo al cuestionario actual.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Entiéndase como <a href="#">borrado</a> físico como el <a href="#">borrado</a> directo en la base de datos.

<b>FRQ-0037</b>	<b><a href="#">Borrado</a> lógico de pregunta</b>
<b>Versión</b>	1.0 ( 16/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir eliminar una pregunta de manera lógica, cuando esta pregunta este en un cuestionario y se haya insertado desde el baúl de preguntas.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

<b>FRQ-0030</b>	<b>Modificar pregunta</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Liliana Reyes Cabello</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir a un <a href="#">editor</a> modificar una pregunta, si y sólo si, el <a href="#">editor</a></i>

	<i>sea dueño de la pregunta.</i>
<b>Importancia</b>	quedaría bien
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Se debe tener en cuenta que sólo se podrá hacer la modificación si la pregunta no está en otro cuestionario. O pertenece ya al baúl de preguntas.

<b>FRQ-0038</b>	<b>Pregunta <a href="#">PA</a></b>
<b>Versión</b>	1.0 ( 16/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Liliana Reyes Cabello</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir insertar preguntas con respuesta escrita (abiertas) [PA], en un cuestionario.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

<b>FRQ-0029</b>	<b>Ranking</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>tener un ranking de las preguntas más usadas, es decir, un contador de preguntas.</i>
<b>Importancia</b>	quedaría bien
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	validado
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Cada vez que una pregunta sea usada en un cuestionario que estuvo en aplicación, el sistema aumentará el contador de la pregunta, esto con el objeto de saber la frecuencia de uso de la preguntas.

<b>FRQ-0026</b>	<b>Pregunta <a href="#">POM</a></b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	• <a href="#">[IRQ-0002] Datos de pregunta</a>
<b>Descripción</b>	El sistema deberá <i>permitir al <a href="#">editor</a> crear, en un cuestionario, con preguntas de opción múltiple (<a href="#">POM</a>).</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	en construcción
<b>Estabilidad</b>	alta
<b>Comentarios</b>	<p><a href="#">POM</a></p> <p>" El número de opciones de respuesta puede variar.  "Sólo se elige una opción.  " Tiene la posibilidad de ser contestada o dejar sin contestar, pero esto únicamente lo puede decidir el <a href="#">editor</a>, al momento de estar editando el cuestionario.  "Deben de tener descripción de las opciones, cuyo llenado es opcional.</p> <p><a href="#">PA</a></p> <p>" Estas deben incluir un espacio para responder libremente con palabras. El <a href="#">editor</a> podrá acotar el número de caracteres a escribir.</p>

#### 4.2.1.4 Reglas de negocio

<b>CRQ-0001</b>	<b>Usuario creacion de cuestionario</b>
<b>Versión</b>	1.0 ( 15/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Octavio León López</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>sólo un usuario <a href="#">administrador</a> puede crear cuestionarios.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

<b>CRQ-0002</b>	<b><a href="#">Borrado</a> lógico de cuestionario</b>
<b>Versión</b>	1.0 ( 16/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	La información almacenada por el sistema deberá satisfacer la siguiente restricción: "Sólo un usuario <a href="#">administrador</a> podrá borrar un cuestionario, bajo la solicitud de un usuario. Sólo se podrá borrar un cuestionario si el usuario es creador del cuestionario, es decir, sólo un <a href="#">editor</a> podrá solicitar el borrarlo".
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Nota. Las preguntas contenidas en el cuestionario, no se verán afectadas, estas permanecerán en el banco de preguntas para un uso posterior, siempre y cuando el cuestionario este en estatus de <a href="#">aprobado</a> .

<b>CRQ-0003</b>	<b><a href="#">Borrado</a> cuestionario y preguntas</b>
<b>Versión</b>	1.0 ( 16/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>El cuestionario podrá ser eliminado si este no ha sido contestado, es decir, es decir, puede estar en un estatus de <a href="#">aprobado</a>, <a href="#">publicado</a>, <a href="#">edición</a>.</i>
<b>Importancia</b>	quedaría bien
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Ninguno

<b>CRQ-0004</b>	<b><a href="#">Borrado</a> de preguntas</b>
<b>Versión</b>	1.0 ( 16/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a> <a href="#">Liliana Reyes Cabello</a>

<b>Dependencias</b>	Ninguno
<b>Descripción</b>	<p>La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>El <a href="#">borrado</a> de una pregunta está sujeto a:</i></p> <p><i>*Sólo se podrá borrar si el usuario es creador de la pregunta.</i></p> <p><i>*Si la pregunta no está usada por otro cuestionario, entonces se podrá borrar.</i></p> <p><i>*Si alguien ha respondido a esa pregunta, no se podrá borrar o modificar.</i></p>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

#### 4.2.1.5 Requisitos de conducta

##### 4.2.1.5.1 Reporte de resultados

<b>FRQ-0033</b>	<b>Reporte de resultados</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Liliana Reyes Cabello</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	<p>El sistema deberá <i>generar, a petición del usuario, un reporte de calificaciones o resultados cuantitativo.</i></p> <p><i>El reporte de calificaciones debe contener:</i></p> <p><i>" Nombre</i></p> <p><i>" Matrícula</i></p> <p><i>" Carrera</i></p> <p><i>" Nombre del cuestionario</i></p> <p><i>" Fecha de aplicación</i></p> <p><i>" Conteo de preguntas acertadas</i></p> <p><i>" Resultado cuantitativo</i></p> <p><i>" Total</i></p> <p><i>" Un desglose por tema</i></p> <p><i>En el caso de las <a href="#">PA</a> se incluirán:</i></p> <p><i>" Los indicadores obtenidos</i></p> <p><i>" Los comentarios de los profesores</i></p>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar

<b>Estado</b>	validado
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Para el desglose por tema. Si un cuestionario contenía varios temas, se puede entregar información de los puntos obtenidos en cada tema. Ejemplo: matemáticas 3 de 5, física 1 de 4, comprensión lectura 2 de 2, etcétera.

<b>FRQ-0034</b>	<b>Reporte estadístico</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• <a href="#">[FRQ-0033] Reporte de resultados</a></li> </ul>
<b>Descripción</b>	El sistema deberá <i>Generar reportes estadísticos, no asociadas, es decir, que sólo contendrán los datos básicos de un <a href="#">participante</a>, que no involucren datos personales.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Desde la cuenta de correo institucional, un estudiante, ingresará a la sección de sistemas de cuestionarios. Si su cuestionario está <a href="#">resuelto</a> , el sistema le entrega el reporte, en caso contrario, se muestra una leyenda que informará que no hay evaluaciones disponibles.

<b>FRQ-0035</b>	<b>Reportes no asociados</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• <a href="#">[FRQ-0034] Reporte estadístico</a></li> <li>• <a href="#">[FRQ-0033] Reporte de resultados</a></li> </ul>
<b>Descripción</b>	El sistema deberá <i>entregar los reportes sin almacenarlos en una base datos, es decir, no hay persistencia para ningún reporte, ya sea reporte de resultados o reporte estadístico.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	<p>Por ejemplo que al generar un reporte estadístico el sistema no guarde la matrícula del <a href="#">participante</a>.</p> <p>Si se hace un reporte de resultado, este se generara al vuelo.</p>

<b>FRQ-0032</b>	<b>Evaluación pregunta en general</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>calculara el resultado del cuestionario como la suma de todas las preguntas acertadas entre el número total de preguntas. Siempre y cuando no existen preguntas abiertas en un cuestionario.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	baja
<b>Comentarios</b>	Ninguno

#### 4.2.1.6 Requisitos de información

<b>IRQ-0001</b>	<b>Información del cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a <i>un cuestionario</i> . En concreto:
<b>Datos específicos</b>	Título Descripción ( <i>Aquí se puede escribir el tema</i> ) Número de preguntas <a href="#">Editor</a> Colaboradores Lista de preguntas Fecha de última modificación Fecha creación Habilitado y <a href="#">deshabilitado</a> Estatus Contraseña opcional Valor de cada pregunta Solicitante del cuestionario Lugar de aplicación Observaciones del <a href="#">colaborador</a> Observaciones del <a href="#">administrador</a>

	Discriminante	
Tiempo de vida	Medio	Máximo
	PD	PD
Ocurrencias simultáneas	Medio	Máximo
	1	1
Importancia	importante	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	baja	
Comentarios	Un cuestionario será una colección de preguntas.	

IRQ-0002	Datos de pregunta	
Versión	1.0 ( 07/05/2016 )	
Autores	<a href="#">Octavio León López</a>	
Fuentes	<a href="#">Alberto Carlos Pérez Cadena</a>	
Dependencias	• <a href="#">[FRQ-0026] Pregunta POM</a>	
Descripción	El sistema deberá almacenar la información correspondiente a <i>una pregunta</i> . En concreto:	
Datos específicos	<p>Texto de la pregunta</p> <p>Descripción de la pregunta ( <i>opcional</i> )</p> <p>Un valor numérico ( <i>el valor cuantitativo de esta pregunta</i> )</p> <p>Fecha creación</p> <p>Identificador de la creador</p> <p>Tema</p> <p>Subtema</p> <p>Tipo</p> <p>Respuesta ( <a href="#">POM</a> )</p> <p>Opción correcta ( <a href="#">POM</a> )</p> <p>Indicadores ( <a href="#">PA</a> )</p>	
Tiempo de vida	Medio	Máximo
	PD	PD
Ocurrencias simultáneas	Medio	Máximo
	1	1
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Para el caso del valor numérico, se puede tener, por ejemplo, valores como: 1, 0.75, 85, 5% etcétera. Estos valores servirán, al <a href="#">editor</a> del cuestionario, para darle	

un valor a cada pregunta y este mismo será responsable de la forma en que calificará cuantitativamente. El sistema sólo devolverá la suma de los puntos obtenidos en cada pregunta.

<b>IRQ-0003</b>	<b>Temas y subtemas</b>	
<b>Versión</b>	1.0 ( 15/04/2017 )	
<b>Autores</b>	<a href="#">Octavio León López</a>	
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a <i>integrar una base de preguntas clasificada por temas y opcionalmente por subtemas</i> . En concreto:	
<b>Datos específicos</b>	id tema ( <i>identificador recursivo</i> ) nombre ( <i>Título del tema</i> ) descripción	
<b>Tiempo de vida</b>	<b>Medio</b>	<b>Máximo</b>
	<b>PD</b>	<b>PD</b>
<b>Ocurrencias simultáneas</b>	<b>Medio</b>	<b>Máximo</b>
	1	1
<b>Importancia</b>	importante	
<b>Urgencia</b>	inmediatamente	
<b>Estado</b>	pendiente de verificación	
<b>Estabilidad</b>	baja	
<b>Comentarios</b>	Ninguno	

#### 4.2.1.7 Usuarios: tipos y funciones

<b>FRQ-0036</b>	<b>Cambiar <a href="#">Editor</a></b>	
<b>Versión</b>	1.0 ( 16/04/2017 )	
<b>Autores</b>	<a href="#">Octavio León López</a>	
<b>Fuentes</b>	<a href="#">Octavio León López</a>	
<b>Dependencias</b>	Ninguno	
<b>Descripción</b>	El sistema deberá <i>permitir</i> , a un <a href="#">administrador</a> , <i>cambiar de <a href="#">editor</a> en cualquier momento</i> .	
<b>Importancia</b>	importante	
<b>Urgencia</b>	inmediatamente	
<b>Estado</b>	en construcción	
<b>Estabilidad</b>	media	
<b>Comentarios</b>	Ninguno	

<b>FRQ-0004</b>	<b>Crear cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir sólo a un usuario <a href="#">administrador</a> crear cuestionarios.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	validado
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

<b>FRQ-0008</b>	<b>Derechos de <a href="#">colaborador</a></b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>únicamente dar derechos de sólo lectura para un usuario <a href="#">colaborador</a> y este podrá visualizar el cuestionario en estatus de <a href="#">publicado</a>, <a href="#">aprobado</a> y <a href="#">resuelto</a>. Además este puede hacer observaciones, únicamente si está en fase de publicación.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

<b>FRQ-0005</b>	<b>Editar un cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir a un usuario <a href="#">editor</a>, únicamente editar los cuestionarios. Además podrá cambiar el estatus de los cuestionarios.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	validado
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

<b>FRQ-0011</b>	<b>Editores para cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir exclusivamente un <a href="#">editor</a> por cada cuestionario.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	validado
<b>Estabilidad</b>	alta
<b>Comentarios</b>	El cambio de <a href="#">editor</a> se solicitará al <a href="#">administrador</a> del sistema.

<b>FRQ-0009</b>	<b>Función para el <a href="#">participante</a></b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>ceder derechos a un <a href="#">participante</a> para resolver un cuestionario, mismo que debe de estar con estatus de <a href="#">aprobado</a>.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	Ninguno

<b>FRQ-0007</b>	<b>Modificación de un cuestionario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir al <a href="#">administrador</a> modificar un cuestionario, si y sólo si, no haya respuestas o bien si se autoriza que se borren todas las respuestas.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	media
<b>Comentarios</b>	El cuestionario NO debe estar con estatus de <a href="#">resuelto</a> . Para que se pueda hacer esta acción.

<b>FRQ-0002</b>	<b>Perfiles y roles de usuario</b>
<b>Versión</b>	1.0 ( 07/05/2016 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Octavio León López</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir la creación de diferentes perfiles o roles de usuario para asignarles permisos de acuerdo a sus tareas.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	puede esperar
<b>Estado</b>	en construcción
<b>Estabilidad</b>	media
<b>Comentarios</b>	Los usuarios estarán en el sistema bajo las siguientes categorías: <ul style="list-style-type: none"> <li>" <a href="#">Administrador</a></li> <li>" <a href="#">Editor</a> (diseñador)</li> <li>" <a href="#">Colaborador</a> (revisor)</li> <li>" <a href="#">Participante</a></li> <li>" <a href="#">Evaluador</a></li> </ul>

## 4.2.2 Requisitos no funcionales

### 4.2.2.1 Usabilidad

<b>NFR-0001</b>	<b>Interfaz</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>tener una interfaz gráfica sencilla y amigable, basada en menús, ventanas, listas desplegadas y botones de acción.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

<b>NFR-0003</b>	<b>Mensajes</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>proporcionar mensajes de error que sean informativos y orientados a usuario final.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de verificación
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

<b>NFR-0004</b>	<b>Diseño responsivo</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno

<b>Descripción</b>	El sistema deberá <i>poseer un diseño "Responsivo" a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

#### 4.2.2.2 Interfaces de Software

<b>NFR-0002</b>	<b>Requerimiento de producto</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir que la interfaz de usuario será implementada para navegadores web y utilizando el lenguaje de programación PHP, JavaScript y utilizará el estándar HTML para el diseño de las páginas web del sistema.</i>
<b>Importancia</b>	vital
<b>Urgencia</b>	hay presión
<b>Estado</b>	validado
<b>Estabilidad</b>	alta
<b>Comentarios</b>	De esta forma se garantizaría que el código HTML generado pueda ser interpretado por cualquier de los navegadores comerciales existentes en el mercado.

<b>NFR-0005</b>	<b>Base de datos</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>utilizar el motor de bases de datos PostgreSQL. El sistema no garantiza su funcionalidad sobre otros motores de Base de Datos.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	validado
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

### 4.2.2.3 Seguridad

<b>NFR-0006</b>	<b>Seguridad de acceso al sistema</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>permitir a los usuarios registrados en el Sistema el ingreso hacia las diversas funcionalidades, permitiendo el filtrado de datos de acuerdo al rol o perfil del usuario.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	alta
<b>Comentarios</b>	<p>Existirán dos formas de ingresar al sistema para los usuarios.</p> <p>a) El modo de acceso al sistema será a través de la página del correo institucional correspondiente al estudiante y profesor. El nombre de usuario para el profesor y el estudiante será su correo institucional, la contraseña será la misma que se utiliza para acceder al correo institucional.</p> <p>b) Para un <a href="#">administrador</a>, el acceso será a través de un link especial que redireccionará a un login o forma. Se pedirá su usuario y contraseña.</p>

<b>NFR-0007</b>	<b>Manejo de Contraseñas</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	<p>El sistema deberá <i>proveer las funcionalidades de autenticación. Para tal fin debe proveer las interfaces de usuario necesarias para permitir las siguientes funcionalidades:</i></p> <p><i>* Pantalla para autenticación de usuarios en el sistema (login y password)</i></p>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

#### 4.2.2.4 Robustez

<b>NFR-0008</b>	<b>Verificación de campos obligatorios</b>
<b>Versión</b>	1.0 ( 14/04/2017 )
<b>Autores</b>	<a href="#">Octavio León López</a>
<b>Fuentes</b>	<a href="#">Alberto Carlos Pérez Cadena</a>
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema deberá <i>verificar que los datos correspondientes a los campos obligatorios de los formularios fueron ingresados por el usuario final.</i>
<b>Importancia</b>	importante
<b>Urgencia</b>	inmediatamente
<b>Estado</b>	pendiente de validación
<b>Estabilidad</b>	alta
<b>Comentarios</b>	Ninguno

#### Principales tecnologías utilizadas

<b>Tecnología</b>	<b>Utilidad</b>
<b>Artisan</b>	Interfaz de línea de comandos de Laravel que provee de comandos útiles, como por ejemplo: la creación de controladores, el uso de un ORM (Eloquent) o visualizar todas las rutas del proyecto, entre otras.
<b>Bootstrap</b>	Es marco de trabajo para el desarrollo responsivo, para proyectos en la web, basado en HTML, CSS y JavaScript.
<b>Bower</b>	Programa basado en NodeJS para el manejo de paquetes en el frontend.
<b>Carbón</b>	Un API simple que sirve de extensión de PHP para DateTime (manejo de fechas).
<b>Composer</b>	Manejador de dependencias de PHP.
<b>Datetimepicker</b>	Proporciona un calendario en el estilo Bootstrap.
<b>Dompdf</b>	Generación de reportes en pdf.
<b>Especificación PSR-4</b>	Laravel sigue el estándar PSR-4 el cual permite a Composer cargar las clases del proyecto automáticamente.
<b>Faker</b>	Es una biblioteca de PHP que genera datos falsos. Ya sea que necesite inicializar su base de datos, rellenar la persistencia para probarla con estrés, etcétera.
<b>jquery-confirm2</b>	Un plugin de jQuery que proporciona un gran conjunto de características para elementos emergentes como alertas o confirmación de acciones.
<b>jquery-ui</b>	Es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plug-ins, widgets y efectos visuales para la creación de aplicaciones web.
<b>Laravel</b>	Framework de desarrollo.
<b>MathJax</b>	Es una biblioteca javascript que permite visualizar fórmulas matemáticas en navegadores web, utilizando los lenguajes de marcado de LaTeX.
<b>REM</b>	Es una herramienta experimental gratuita de gestión de requisitos.

## 5 Glosario de términos

**Deshabilitado:** En todo momento se puede deshabilitar cuestionarios, preguntas o usuarios. Esto significa que para un usuario [editor](#) será o no visible el dicho elemento, pero estará todavía en la base de datos ([borrado](#) lógico). La visibilidad del elemento se refiere a que un cuestionario se vuelve visible o no para el usuario en particular.

**PA:** Pregunta de opción

**POM:** Pregunta de opción múltiple

### 5.1 Estatus

**Aprobado:** Será un cuestionario que ha sido [aprobado](#) entre los colaboradores y el [editor](#), en consecuencia el cuestionario se puede poner en marcha para su aplicación. Una vez [aprobado](#), el cuestionario no podrá ser editado, sólo en casos, muy necesarios, una [edición](#) llegada a este estado se realizará por medio de permisos por parte del [administrador](#).

**Borrado:** Por reglas de negocio de la UACM, no siempre es factible eliminar información, por lo que se entenderá por [borrado](#) como un cambio de estatus a [deshabilitado](#) ([borrado](#) lógico) y sólo en ciertos casos muy especiales existirá un [borrado](#) real, es decir un [borrado](#) directo en a la base de datos.

**Edición:** En este estatus el cuestionario se encuentra en [edición](#) por el [editor](#), que junto con el [administrador](#) tienen acceso a él.

**Publicado:** En esta fase el cuestionario ya fue terminado por el [editor](#) y se espera la aprobación de los colaboradores.

**Resuelto:** Cuando un [participante](#) ha [resuelto](#) un cuestionario [aprobado](#).

### 5.2 Usuarios

**Administrador:** es la persona que tendrá la responsabilidad de configurar, mantener, monitorizar, documentar y asegurar el correcto funcionamiento del sistema informático.

**Colaborador:** es el usuario responsable de dar su voto aprobatorio para el cuestionario y de esta forma el este pase a un estatus de [aprobado](#).

**Editor:** este usuario podrá ser un profesor que pertenezca a la UACM. Será el usuario encargado de la creación e inserción de preguntas.

**Evaluador:** este usuario podrá ser un profesor que pertenezca a la UACM. Es la persona responsable de evaluar o calificar las preguntas abiertas.

**Participante:** podrá es un estudiante que pertenezca a la comunidad de la UACM.

---

## Anexo B

### Modelo entidad relación

El Modelo de Entidad Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementándose en forma gráfica a través del Diagrama Entidad Relación. A continuación se muestra el diagrama que describe las entidades y sus relaciones para el prototipo del sistema de cuestionarios.



---

# Anexo C

## Diccionario de datos

El diccionario de datos es una lista organizada de todos los datos pertinentes a un sistema, con un conjunto de definiciones precisas, para que tanto el analista de la base de datos como el usuario mantengan un entendimiento mutuo.

En el paso del modelo E/R a tablas relacionales (mapeo), para producir un diseño completo de la base de datos, se toman en cuenta cada atributo y su significado, se sabe que las relaciones se convierten en llaves foráneas con una integridad referencial, esto agrega atributos adicionales a cada tabla, que también son importantes definir en su semántica. A continuación se detallara los significados de cada uno de los atributos perteneciente a las entidades representadas en el Modelo de Datos.

Notación en el Diccionario de Datos:

PK	Clave primaria
FK	Clave foránea
°	Opcional
*	Obligatorio

Las entidades encontradas para el sistema de cuestionario y su semántica se listan a continuación:

Nombre de la tabla: Cuestionarios			
Descripción: Tabla que contendrá a los cuestionarios de la aplicación.			
Nombre	Propiedad	Tipo	Descripción
id	PK, *	numérico	Clave del cuestionario.
fecha creación	*	fecha	Fecha de creación del cuestionario.
fecha modificación	°	fecha	Fecha de la última modificación al cuestionario.
habilitado	*	booleano	Valor para habilitar o deshabilitar cuestionarios (con esto se logra un borrado lógico).
id solicitante	*	numérico	Identificador de la persona que solicita el cuestionario.
título	*	carácter	Título o nombre del cuestionario.
descripción	*	texto	Ofrece una idea más detallada del cuestionario.
estatus	*	restricción	Estatus del cuestionario. 1. Edición. 2. Publicado. 3. Aprobado. 4. Resuelto 5. Borrado (este será el valor habilitado).

<b>Nombre de la tabla:</b> Colaboradores			
<b>Descripción:</b> Aquí se estarán tanto editores como colaboradores para un cuestionario.			
<b>Nombre</b>	<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
id colaborador	PK,*	numérico	Clave del colaborador.
id cuestionario	FK,*	numérico	Clave foránea del cuestionario.
función	*	carácter	Determinará el tipo de persona que participa en el cuestionario. Puede ser un editor o un colaborador.
estatus	*	restricción	Determina si está activo o no un usuario para el cuestionario presente. Por default estará activo.
VoBo	o	booleano	Visto bueno para saber que ya ha sido aprobado el cuestionario.
fecha de activación	o	date	Fecha de cuando un colaborador fue activado, para participar en la revisión del cuestionario.

<b>Nombre de la tabla:</b> Observaciones			
<b>Descripción:</b> Tabla que tendrá las observaciones del cuestionario.			
<b>Nombre</b>	<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
id observador	PK,*	numérico	Clave del usuario que hace la observación
id cuestionario	FK,*	numérico	Clave foránea del cuestionario
descripción	*	texto	Contenido de la instrucción
fecha observación	*	fecha	Fecha en la que se hizo la observación
estatus	*	restricción	El estatus dependerá si ha visto o no un editor la observación hacia el cuestionario. Los estatus son los siguientes: Pendiente y Visto. Pendiente cuando se manda la observación. Visto cuando se ha abierto la observación.

<b>Nombre de la tabla:</b> Aplicaciones			
<b>Descripción:</b> Tabla que contendrá los datos para la aplicación de un cuestionario.			
<b>Nombre</b>	<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
id	PK,*	numérico	Clave de la aplicación
id cuestionario	FK,*	numérico	Clave de cuestionario
ciclo	*	carácter	Periodo de la aplicación. Ej. 2016-I
fecha inicio	*	date	Fecha del día de la aplicación
fecha fin	*	date	Caducidad del cuestionario
hora inicio	*	date	Hora en la que se inició el cuestionario
hora fin	*	date	Hora en la que se terminó el cuestionario
password	o	carácter	Contraseña opcional del cuestionario
lugar aplicación	o	carácter	Lugar donde se lleva a cabo el cuestionario

<b>Nombre de la tabla:</b> Hoja de respuestas			
<b>Descripción:</b> Tabla que servirá como la hoja de respuesta para los cuestionarios. Contendrá la información acerca del participante así como la fecha y hora en la que respondió un cuestionario.			
Nombre	Propiedad	Tipo	Descripción
id	PK,*	numérico	Clave de la hoja de respuesta
id participante	PK,*	numérico	clave o número de la persona que resolvió el cuestionario
id aplicación	FK,*	numérico	clave foránea de la aplicación
fecha respuesta	°	date	fecha en la que se respondió el cuestionario
hora inicio respuesta	°	date	Momento en que inició a responder un participante el cuestionario.
hora fin respuesta	°	date	Momento en el que un participante termino la aplicación del cuestionario.

<b>Nombre de la tabla:</b> Resp. POM-Simples			
<b>Descripción:</b> Aquí se guardará las respuestas del participante.			
Nombre	Propiedad	Tipo	Descripción
núm. de pregunta	PK,*	numérico	Número de la pregunta en una cierta aplicación y cuestionario.
id POM	FK,*	numérico	Clave de la POM
Id opción respuesta POM	FK,*	numérico	Clave de la Respuesta POM-Simple

<b>Nombre de la tabla:</b> Respuesta POM-Simple			
<b>Descripción:</b> Tabla que guardara las respuestas de un pregunta de opción múltiple.			
Nombre	Propiedad	Tipo	Descripción
id	PK,*	numérico	Clave de la opción de respuesta POM
id pregunta	FK,*	numérico	Clave de la pregunta
texto	*	carácter	El texto que compone a la respuesta. Ej. a)Texto1 b)Texto2, etc.
opción correcta	°	numérico	En este atributo se guarda la opción correcta a la pregunta de opción múltiple. Es opcional porque puede ser que sea la opción correcta de dicha pregunta o tal vez no lo sea.

<b>Nombre de la tabla:</b> Preguntas			
<b>Descripción:</b> Tabla que contendrá las preguntas del cuestionario.			
<b>Nombre</b>	<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
id pregunta	PK,*	numérico	Clave de la pregunta
fecha creación	*	fecha	Fecha de creación de la pregunta
fecha modificación	°	fecha	Fecha de la última modificación de la pregunta
habilitado	*	booleano	Valor para habilitar o deshabilitar una pregunta (con esto se logra un borrado lógico). Una pregunta por default está en habilitada.
descripción	*	longitud variable	Es atributo contendrá el texto de la pregunta
tipo	*	restricción	Tipo de pregunta que puede ser la pregunta, POM o PA
límite de respuesta	°	numérico	Atributo que limita el tamaño máximo de caracteres que un participante puede colocar a la hora de responder la pregunta presente.
popular	°	numérico	Servirá como un contador de preguntas. Cada vez que una pregunta sea usada en un cuestionario que estuvo en aplicación, el sistema aumentará el contador de la pregunta, esto con el objeto de saber la frecuencia de uso de las preguntas.

<b>Nombre de la tabla:</b> Temas			
<b>Descripción:</b> Tabla que contendrá los temas que podrán ser asociados a las preguntas			
<b>Nombre</b>	<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
id tema	PK,*	numérico	Clave del tema
id usuario creación	*	numérico	Identificador del usuario que crea el tema
fecha creación	*	fecha	Fecha de creación del tema
id usuario modificación	°	numérico	Identificador de la última persona que modificó un tema en específico
fecha modificación	°	fecha	Fecha de la última modificación del tema
habilitado	*	booleano	Valor para habilitar o deshabilitar un tema (con esto se logra un borrado lógico)
nombre	*	carácter	Nombre del tema
descripción	°	texto	Descripción del tema
id padre	FK, °	numérico	Clave del subtema

<b>Nombre de la tabla:</b> Textos asociados			
<b>Descripción:</b> El texto asociado es un texto que enriquece una pregunta. Por ejemplo: se quiere complementar una pregunta con un cuento o una historia, pero este texto podrá ser independiente de la pregunta actual, por tanto varias preguntas podrían usar ese mismo texto.			
Nombre	Propiedad	Tipo	Descripción
id texto	PK,*	numérico	Clave del texto
id usuario creación	*	numérico	Identificador del usuario que crea el texto asociado
fecha creación	*	fecha	Fecha de creación del texto asociado
id usuario modificación	°	numérico	Identificador de la última persona que modifico un texto asociado en específico
fecha modificación	°	fecha	Fecha de la última modificación del texto asociado
habilitado	*	booleano	Valor para habilitar o deshabilitar un texto asociado (con esto se logra un borrado lógico)
nombre (titulo)	*	caracteres	Título del texto asociado
descripción (texto)	*	texto	Texto que estará relacionado con la pregunta

<b>Nombre de la tabla:</b> Imágenes			
<b>Descripción:</b> Tabla que guardará las URLs de las imágenes que se podrían colocar en una pregunta			
Nombre	Propiedad	Tipo	Descripción
id imagen	PK, *	numérico	Clave primaria para cada imagen
url	*	carácter	Dirección o ruta de la imagen para poderse colocar en la pregunta
descripción	°	texto	Descripción de la imagen

<b>Nombre de la tabla:</b> Sección			
<b>Descripción:</b> EL cuestionario se podrá dividir en una o más secciones, que separan contenidos entre preguntas, por tanto esta tabla contendrá las secciona para cada cuestionario.			
Nombre	Propiedad	Tipo	Descripción
id	PK,*	numérico	Clave de la sección
id cuestionario	FK,*	numérico	Clave del cuestionario
texto	*	carácter	Título de la sección
descripción	°	texto	Breve idea acerca de la sección

<b>Nombre de la tabla:</b> Cuestionario pregunta			
<b>Descripción:</b> esta tabla contendrá los datos del número y valor de una pregunta, en relación a un cuestionario en específico, es decir, crea la unión entre una pregunta y un cuestionario, por medio de una sección.			
Nombre	Propiedad	Tipo	Descripción
id cuestionario	FK, *	numérico	Clave del cuestionario
id pregunta	FK, *	numérico	Clave de la pregunta
id sección	FK, *	numérico	Clave de la sección
puntaje	*	numérico	Punto o valor numérico que tiene esta pregunta dentro de un cuestionario en una sección, es decir, el puntaje de la pregunta.
numero pregunta	°	numérico	Número de la pregunta para ese cuestionario; Este valor será dado hasta el momento de su aplicación.

<b>Nombre de la tabla:</b> Instrucciones			
<b>Descripción:</b> Entidad que guardara las explicación que describe los que se debe tomar en cuenta en la aplicación del cuestionario.			
Nombre	Propiedad	Tipo	Descripción
id	PK, *	numérico	Clave de la instrucción
id cuestionario	FK, *	numérico	Clave foránea del cuestionario
texto	*	texto	Contenido de la instrucción

<b>Nombre de la tabla:</b> Indicadores			
<b>Descripción:</b> Tabla que guardara datos o información que sirve para conocer o valorar las características de una pregunta y determinar su evolución futura			
Nombre	Propiedad	Tipo	Descripción
id indicador	PK,	numérico	Clave primaria del indicador
id usuario creación	*	numérico	Clave identificador del creador de la pregunta, ya sea un administrador o un profesor
fecha creación	*	fecha	Fecha de creación de la pregunta
id usuario modificación	°	numérico	Identificador de la última persona que modifico una pregunta
fecha modificación	°	fecha	Fecha de la última modificación de la pregunta
habilitado	*	booleano	Valor para habilitar o deshabilitar una pregunta (con esto se logra un borrado lógico). Una pregunta por default está en habilitada.
nombre	*	carácter	Título del indicador
porcentaje	°	texto	Descripción del indicador

---

<b>Nombre de la tabla:</b> Respuestas PA			
<b>Descripción:</b> Tabla que guardará las respuestas de la preguntas abiertas			
<b>Nombre</b>	<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
num pregunta	PK,*	numérico	Número de la pregunta que servirá como parte de la clave primaria
id hoja respuesta	FK,*	numérico	Clave de la hoja de respuesta
id participante	FK,*	numérico	Clave o número de la persona que resolvió el cuestionario. Este id se tomara de la base de datos de los estudiantes.
texto pregunta	o	texto	Texto de la pregunta, este será opcional pues el participante puede dejar sin contestar una pregunta
id pregunta	*	numérico	Clave de la pregunta abierta

---

# Bibliografía

- Albrecht, A. J. (1979). *Measuring application development productivity*.
- Álvarez, D. M. (2010). Plataformas de enseñanza virtual libres y sus características de extensión: desarrollo de un bloque para la gestión de tutorías en moodle. *Ingeniería informática*; <http://www2.uah.es/libretics/files/Tutorias.pdf>.
- Canós, J. H., Letelier, P., y Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. *Metodologías Ágiles en el Desarrollo de Software*, 1.
- Chaves, M. A. (2011). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes*, 6(10).
- Gutiérrez, J. J. (2014). ¿ qué es un framework web? Available in: [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf) Accessed May, 12.
- Herrera, L. J., y Lizka, J. (2003). *Ingeniería de requerimientos ingeniería de software*. <http://www.monografias.com/trabajos6/resof/resof.shtml>.
- IEEE Standards, A., y cols. (1990). Standard glossary of software engineering terminology. *IEEE Software Engineering Standards & Collection*. *IEEE*, 610–12.
- Jacobson, I., Rumbaugh, G., Jacobson, J., Booch, G., y Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software/the unified software development process*. (n.º 004.41). Pearson Educación,.
- Jiménez, E., y Orantes, S. (2012). Metodologías híbridas para desarrollo de software: una opción factible para México. *Revista Digital Universitaria*, 13(1).
- López, F. J. T. (2013). *Administración de proyectos de informática*. Ecoe Ediciones.
- López Márquez, M. D., Villa Palacio, J. C., y cols. (2012). Plantillas y artefactos personalización de rup para proyectos académicos de desarrollo de software.
- Mateu, C. (2010). Desarrollo de aplicaciones web, setiembre 2006.
- McCool, S. (2012). *Laravel starter*. Packt Publishing Ltd.
- Murga Tapia, R. (2013). Diseño y construcción de una plataforma para implementar una línea de productos de software.
- Palacio, J., y Ruata, C. (2014). Gestión de proyectos scrum manager. *Scrum Manager*.

- 
- Palacios, D. (2016). *Cómo crear rutas en laravel 5.\**. <https://styde.net/como-crear-rutas-en-laravel-5/>. (Online; accesado 2016-Octubre-23)
- Pressman, R. S., y Troya, J. M. (1988). *Ingeniería del software* (n.º 001.64 P74s.). McGraw Hill.
- Quiroz, J. (2003). El modelo relacional de bases de datos. *Boletín de Política Informática*, 6, 53–61.
- Saavedra, L. E. (2008). Symfony: Framework para el desarrollo de aplicaciones web (parte i). *SofLix, revista de Linux y software libre*, Available in: <http://www.mclubre.org/descargar/docs/revista-soflix/soflix-03-200809.pdf>, 3.
- SCHACH, S. R., y cols. (2005). *Análisis y diseño orientado a objetos con uml y el proceso unificado/stephen r. schach; tr. lorena peralta rosales*. (n.º QA76. 9. S24 2006.).
- Serrano, G. L. (1996). *Ingeniería de sistemas de software*. Isdefe.
- Sommerville, I., y Galipienso, M. I. A. (2005). *Ingeniería del software*. Pearson Educación.
- Tahuiton, J. (2011). *Arquitectura de software para aplicaciones web*. Centro de investigación y de estudios avanzados del politécnico Nacional. México DF.
- Toro, A. D., y Jiménez, B. B. (2000). Metodología para la elicitación de requisitos de sistemas software. *Informe Técnico LSI-2000-10*. Facultad de Informática y Estadística Universidad de Sevilla.
- Torossi, G. (2005). El proceso unificado de desarrollo de software. *Universidad Tecnológica, Argentina*.
- Villanueva, J. S., y Siachoque, M. M. (2014). Comparando scrum y rup. *Tecnología Investigación y Academia*, 2(1), 39–48.

---

## Referencias para imágenes

- Figura 2.3. Historia usuario ejemplo. (2013). Recuperado de [http://www.scrummanager.net/bok/index.php?title=File:Historia\\_usuario\\_ejemplo\\_2.jpg](http://www.scrummanager.net/bok/index.php?title=File:Historia_usuario_ejemplo_2.jpg)
- Figura 2.4. [Imagen sin título]. (2010). Recuperado de <http://asprotech.blogspot.mx/2010/11/resumen-de-desarrollo-de-requerimientos.html>
- Figura 5.1. Google tendencias. (2016). Interés a lo largo del tiempo. Recuperado de <https://trends.google.com.mx/trends/explore?date=2010-10-16%202016-10-17&q=%2Fm%2F0jwy148,cake%20php,%2Fm%2F09cjc1,%2Fm%2F0cdvjh,yii>
- Figura 5.2. [Imagen sin título]. (2013). Recuperado de <http://imgur.com/eCuptyU>
- Figura 5.14. [Imagen sin título]. Recuperado de <https://richos.gitbooks.io/laravel-5/content/images/arquitectura.png>