



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**INGENIERÍA DE SOFTWARE PARA EL SISTEMA DE EVALUACIONES FORMATIVAS EN
LA PLATAFORMA DTT, UTILIZADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y
SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA**

Daytón Estuardo Agdally García Montúfar

Asesorado por el Ing. Sergio Arnaldo Méndez Aguilar

Guatemala, abril de 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**INGENIERÍA DE SOFTWARE PARA EL SISTEMA DE EVALUACIONES FORMATIVAS EN
LA PLATAFORMA DTT, UTILIZADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y
SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

DAYTÓN ESTUARDO AGDALLY GARCÍA MONTÚFAR
ASESORADO POR EL ING. SERGIO ARNALDO MÉNDEZ AGUILAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, ABRIL DE 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Oscar Humberto Galicia Nuñez
VOCAL V	Br. Carlos Enrique Gómez Donis
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

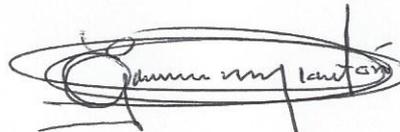
DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

INGENIERÍA DE SOFTWARE PARA EL SISTEMA DE EVALUACIONES FORMATIVAS EN LA PLATAFORMA DTT, UTILIZADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 16 de octubre de 2016.



Daytón Estuardo Agdally García Montúfar



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 17 febrero del 2018

Ing. Carlos Azurdia
Carrera de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala
Guatemala, Ciudad

Respetable Ing. Azurdia:

El motivo de la presente es con objetivo de informarle que en mis labores como asesor del estudiante **Dayton Estuardo Agdally García Montúfar** he procedido a validar el trabajo de graduación titulado como **"INGENIERÍA DE SOFTWARE PARA EL SISTEMA DE EVALUACIONES FORMATIVAS EN LA PLATAFORMA DTT, UTILIZADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"** siendo así a mi criterio este se encuentra completado.

Luego de haber tenido reuniones periódicas con el estudiante y luego de revisar el trabajo, considero que cumple con los requisitos de profesionalismo que deben caracterizar a un futuro profesional de la carrera de ingeniería en ciencias y sistemas.

Sin otro particular me suscribo a usted.

Atentamente,

Sergio Arnaldo Méndez Aguilar

Ingeniero en Ciencias y Sistemas
Colegiado No. 10958

Ing. Sergio Arnaldo Méndez Aguilar



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 14 de marzo de 2018

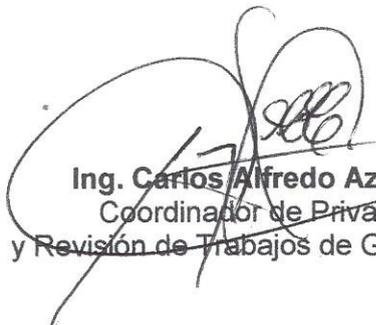
Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **DAYTÓN ESTUARDO AGDALLY GARCÍA MONTÚFAR** con carné **200915754** y CUI **1984 03860 0411**, titulado **"INGENIERÍA DE SOFTWARE PARA EL SISTEMA DE EVALUACIONES FORMATIVAS EN LA PLATAFORMA DTT, UTILIZADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“INGENIERÍA DE SOFTWARE PARA EL SISTEMA DE EVALUACIONES FORMATIVAS EN LA PLATAFORMA DTT, UTILIZADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, realizado por el estudiante, DAYTÓN ESTUARDO AGDALLY GARCÍA MONTÚFAR aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

*Ing. **Martín Antonio Pérez Türk***

Director

Escuela de Ingeniería en Ciencias y Sistemas



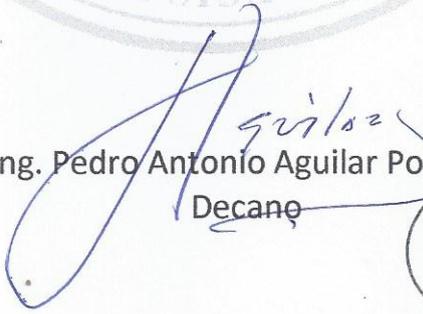
Guatemala, 18 de abril de 2018

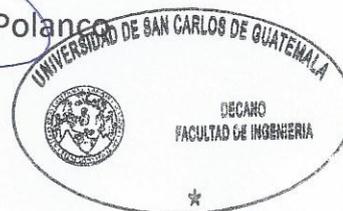


DTG. 137.2018

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **INGENIERÍA DE SOFTWARE PARA EL SISTEMA DE EVALUACIONES FORMATIVAS EN LA PLATAFORMA DTT, UTILIZADO EN LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por el estudiante universitario: **Daytón Estuardo Agdally García Montúfar**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Ing. Pedro Antonio Aguilar Polanco
Decano



Guatemala, abril de 2018

/gdech

ACTO QUE DEDICO A:

- Dios** Por ser una importante influencia en mi carrera, entre otras cosas.
- Mis padres** Rafael y Marcí, por ser el pilar de mi vida, por brindarme su apoyo y amor incondicional y por todo el tiempo que han dedicado.
- Camila González** Por el amor, la paciencia y el apoyo que me ha dado, por estar de alguna forma siempre conmigo y por enseñarme a nunca darme por vencido ante cualquier adversidad.
- Mi ahijado** Santiago, por ser como mi hijo, por darle razón y alegría a mi vida, y por ser parte fundamental de mi existencia.
- Mi abuela** Gorgonia Juárez, por todo el amor y por todas las enseñanzas. Siempre te recordaré donde quiera que estés.
- Mis hermanos** Por estar siempre conmigo, por ser parte fundamental de mi vida.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi casa de estudios y la fuente de mis conocimientos.
Facultad de Ingeniería	Por ser mi Facultad y haber dado todo lo necesario para mi educación durante el transcurso de mi carrera.
Mis padres	Por haberme apoyado durante el transcurso de mi carrera, por todo el amor y los sacrificios que han hecho para hacer posible cumplir este objetivo.
Mi asesor, el ingeniero Sergio Méndez	Por brindarme desinteresadamente sus conocimientos, que fueron la base para este trabajo de graduación.
Mi revisor, la ingeniera Floriza Ávila	Por el tiempo brindado y por haber orientado mi investigación hacia el objetivo deseado.
Ingeniero Miguel Marín	Por la oportunidad que me ha brindado de participar en este proyecto, y por haber orientado mi investigación hacia el objetivo deseado.

**Mis amigos de la
Facultad**

Fernando González, Manuel Tay, Edward Gómez, Erick Suy, Daniel Cos, Lester Tzoc, Leonardo Cohobon, Christian Agustín, por el apoyo y por haber hecho ameno el transcurso de mi carrera, por las verdaderas amistades que hemos cosechado.

Mis catedráticos

Por brindarme desinteresadamente sus conocimientos, por ser parte fundamental en mi formación académica y ser la guía de esta y muchas generaciones.

**Personas involucradas
en el proyecto DTT**

Por el apoyo y por darle un valor agregado a mi investigación con sus conocimientos, tiempo y experiencia.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS	VII
GLOSARIO	IX
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN	XV
1. ANTECEDENTES	1
1.1. Antecedentes del sistema DTT.....	1
1.1.1. Misión	1
1.1.2. Visión.....	2
1.1.3. Objetivos.....	2
1.2. Situación actual	3
1.3. Requerimientos para el módulo de evaluaciones online	3
1.3.1. Definición de requerimientos	3
1.3.2. Actualidad del sistema DTT.....	6
1.3.3. Mercado objetivo	6
2. ASPECTOS GENERALES	7
2.1. Descripción general del proyecto	7
2.2. Análisis y diseño del sistema.....	9
2.3. Sistemas de bases de datos.....	10
2.4. Bases de datos SQL vs bases de datos NO SQL	11
2.4.1. Bases de datos SQL.....	11
2.4.2. Bases de datos NO SQL	11

	2.4.2.1.	Bases de datos documentales	12
	2.4.2.2.	Bases de datos en grafo.....	14
	2.4.2.3.	Bases de datos clave/valor.....	14
	2.4.2.4.	Bases de datos multivalor	15
2.5.		Elección de tecnologías	16
	2.5.1.	Comparativa de bases de datos clave/valor.....	16
	2.5.1.1.	Apache Cassandra.....	16
		2.5.1.1.1. Descripción general	16
		2.5.1.1.2. Ventajas y desventajas	17
	2.5.1.2.	MongoDB	17
		2.5.1.2.1. Descripción general	17
		2.5.1.2.2. Ventajas	18
	2.5.1.3.	Redis	18
		2.5.1.3.1. Descripción general	18
		2.5.1.3.2. Ventajas	19
	2.5.1.4.	Conclusión sobre comparativa	19
	2.5.2.	Justificación.....	20
2.6.		Beneficios.....	21
	2.6.1.	Inversión.....	22
3.		ANÁLISIS DE MÉTODOS PARA EVALUACIONES <i>ONLINE</i>	25
	3.1.	Aspectos generales sobre sistemas de evaluación <i>online</i>	25
	3.1.1.	Preguntas convencionales	25
		3.1.1.1. Preguntas de opción múltiple	26
		• Preguntas de respuesta única:.....	26
		• Preguntas de respuesta múltiple:	26

3.1.1.2.	Preguntas de falso/verdadero.....	26
3.1.2.	Tecnologías disponibles para evaluaciones <i>online</i>	28
3.1.2.1.	Disponibles en la Universidad de San Carlos de Guatemala.....	28
3.1.2.1.1.	Sistema de evaluación para pruebas específicas de computación (SAE/SAP) en la Facultad de Ingeniería..	29
3.1.2.1.2.	Sistema de evaluación para examen de ubicación de idioma técnico en la Facultad de Ingeniería	30
3.1.2.2.	Disponibles en internet	30
3.1.2.2.1.	Socrative	31
3.1.2.2.2.	Quizstar de 4teachers ..	33
3.1.2.2.3.	QuizWorks.....	33
4.	ARQUITECTURA DE LA APLICACIÓN	35
4.1.	Vista lógica	35
4.1.1.	Diagrama de clases	35
4.1.2.	Diagrama de datos	36
4.1.3.	Diagramas de secuencia	37
4.2.	Vista de implementación.....	40
4.2.1.	Diagrama de componentes.....	40

4.2.2.	Diagrama de paquetes	41
4.3.	Vista de proceso	42
4.3.1.	Diagrama de actividad.....	42
4.4.	Vista física.....	44
4.5.	Vista de escenarios.....	44
CONCLUSIONES.....		45
RECOMENDACIONES		47
BIBLIOGRAFÍA.....		49

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Documento de base de datos documental	13
2.	Comparativa entre versiones de Socrative.....	32
3.	Características principales de Quizstar	33
4.	Diagrama de clases.....	36
5.	Diagrama de base de datos	37
6.	Diagrama de secuencia para tutores y catedráticos	38
7.	Diagrama de secuencia para estudiantes	39
8.	Diagrama de componentes	41
9.	Diagrama de paquetes	42
10.	Diagrama de proceso de creación de <i>quiz</i>	43
11.	Diagrama de proceso de tomar un <i>quiz</i>	43
12.	Diagrama de vista física	44

TABLAS

I.	Especificaciones de hardware.....	22
II.	Especificaciones de software	22
III.	Tiempo invertido.....	23

LISTA DE SÍMBOLOS

Símbolo	Significado
BBDD	Bases de datos
DB	Base de datos, por sus siglas en inglés
DTT	Departamento de Transferencia Tecnológica
GB	Gigabyte
GUI	Interfaz gráfica de usuario
C++	Lenguaje de programación C++
SQL	Lenguaje estructurado de consultas, por sus siglas en inglés
MB	Megabyte
MHz	Megahercio
ms	Milisegundos
DBMS	Sistema de administrador de base de datos, por sus siglas en inglés

TB

Terabyte

GLOSARIO

Float	Tipo de variable la cual almacena números decimales de hasta 4 bytes.
JavaScript	Lenguaje de programación interpretado, implementado en un navegador web, permitiendo así mejoras en la interfaz de usuario y páginas web dinámicas.
JSON	Objeto de notación javascript (<i>javascript object notation</i>), es un formato ligero utilizado para el intercambio de datos basado en javascript el cual no requiere del uso de XML.
Join	Es una sentencia de SQL la cual permite realizar la combinación de registros de dos o más tablas en una base de datos relacional.
Código abierto	Código abierto (<i>open source</i>), es el término con el que se conoce al software que se desarrolla y distribuye de forma libre.
Paradigma RAID	Hace referencia a un sistema de almacenamiento, que utiliza múltiples discos duros entre los que se distribuyen o replican los datos.

RAM	Memoria de acceso aleatorio, es la memoria en la cual el procesador recibe las instrucciones y guarda los resultados obtenidos, es temporal y volátil.
Servidor	Físicamente hablando es una computadora o CPU el cual forma parte de una red de computadoras con la finalidad de proveer de diversos servicios a las demás computadoras conocidas como clientes.
Transacción	Es un conjunto de órdenes o instrucciones que se ejecutan de forma atómica o indivisible.
Tupla	Sinónimo de fila en una tabla.
W3C	<i>World wide web consortium</i> , es un consorcio internacional que produce recomendaciones para la red de informática mundial.
XML	Lenguaje de marcas extensibles (<i>eXtensible markup lenguaje</i>) desarrollado por el W3C, desarrollado para permitir la compatibilidad entre sistemas para compartir información de una manera segura, fácil y fiable.

RESUMEN

En este trabajo se abordarán generalidades acerca del desarrollo del módulo de software para el sistema DTT, plataforma que se usa actualmente en los cursos de la Escuela de Ingeniería en Ciencias y Sistemas; este módulo es el que permite la creación de pruebas de conocimiento tipo cuestionario.

Las pruebas se asignarán a un curso y sección específica; no será posible hacer estas pruebas para un porcentaje distinto a la totalidad de estudiantes del curso y la sección seleccionados; tampoco, será posible aplicar la prueba a dos grupos de estudiantes diferentes, por lo que se deberá crear la prueba dos veces para lograr este fin; al finalizar el tiempo de duración de la prueba, las preguntas no contestadas tendrán nota cero; los resultados de las pruebas serán cargados al sistema automáticamente; si se desea la modificación de estos resultados, se podrá realizar únicamente por el tutor o catedrático que creó la prueba. Este módulo de software estará disponible para su uso a partir del primer semestre del año 2017 y al estar incorporado en la plataforma DTT se limita únicamente a cursos pertenecientes a la Escuela de Ingeniería en Ciencias y Sistemas; estudiantes, tutores y catedráticos registrados en dicha plataforma.

OBJETIVOS

General

Contribuir con la modernización de las metodologías actuales de evaluación académica y promover el uso de las tecnologías de información (TI) en los sistemas de educación de Guatemala, en la Escuela de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.

Específicos

1. Optimizar de proceso de evaluaciones.
2. Eliminar los errores humanos en la gestión de notas en pruebas de conocimiento.
3. Motivar a metodologías de trabajo virtuales antes que presenciales para ampliar el alcance de la educación.

INTRODUCCIÓN

La tecnología evoluciona de forma exponencial; la vida cotidiana, educativa y labora deben adaptarse a los cambios tecnológicos adoptando nuevas y mejores metodologías de trabajo.

En la actualidad, todos los cursos y laboratorios de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala adoptan una metodología de trabajo basada en evaluaciones; dichas evaluaciones son realizadas por los alumnos de forma presencial y diseñadas por los tutores académico y/o catedráticos; generalmente, desde una computadora. El proceso de calificación se realiza de forma manual; seguido, las calificaciones se ingresan a la plataforma DTT utilizada en la Escuela de Ingeniería en Ciencias y Sistemas de esta universidad.

El propósito de este trabajo es poner al alcance de los tutores académicos y los catedráticos una herramienta integrada en la plataforma DTT (plataforma que ya utilizan para el control de notas y planificación de cursos) que les permitirá la creación de pruebas de conocimiento de forma digital; estas pruebas podrán diseñarse a través de formularios sencillos que permiten la creación de preguntas de diferentes tipos, la selección del público objetivo (estudiantes de un curso y de una sección específica), la fecha de inicio y el tiempo de duración; así mismo, al concluir la prueba, las notas se cargarán al sistema de notas de la DTT de forma automática lo que evitará la intervención humana y eliminará los errores debido al extenso proceso que actualmente se sigue. También, se pretende que la Escuela de Ingeniería en Ciencias y

Sistemas sea un modelo a seguir para otras escuelas de la facultad en la adopción de nuevas tecnologías y metodologías de trabajo.

1. ANTECEDENTES

En este capítulo se desarrollan los antecedentes del sistema DTT y los requerimientos que se demandan para la inclusión del nuevo módulo de software en este sistema.

1.1. Antecedentes del sistema DTT

En esta sección se hablará de la visión, la misión y los objetivos que han llevado al sistema DTT a su creación y que son los pilares sobre los que se establece el proyecto.

1.1.1. Misión

Otorgar al estudiante las competencias acertadas que garanticen el éxito en la búsqueda del conocimiento por medio de los distintos estilos de aprendizaje y fomentar la investigación de manera permanente que le permita una mejora en su calidad de vida. Toma en cuenta las opciones que el país ofrece a las distintas áreas del mercado actual (logística, administración, información tecnología, finanzas, contabilidad, comercial, etc.), el ámbito internacional debido a la alta competencia que se maneja en estos tiempos. Proporcionar información sobre los diferentes cambios y actualizaciones a nivel mundial para estar enterados de los nuevos sistemas y aplicaciones.

1.1.2. Visión

Reconocer al estudiante de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala como un profesional de alto nivel, con base en los saberes incorporados en el pensum de estudios que permitan formarlo de manera integral para el ejercicio profesional, otorgándole los instrumentos adecuados para su desarrollo ocupacional.

1.1.3. Objetivos

El sistema DTT busca los siguientes objetivos:

- Ser el sistema principal para el control académico en la Escuela de Ingeniería en Ciencias y Sistemas.
- Servir de herramienta para la toma de decisiones de los tutores académicos y catedráticos a través de los reportes, en cada momento, de las notas de los estudiantes.
- Ser una herramienta para el control de las actividades que los tutores académicos realizan como parte de la práctica final; facilita al coordinador de proyecto DTT su seguimiento.
- Ser el repositorio centralizado y unificado donde se pueda encontrar toda la información sobre la Escuela de Ingeniería en Ciencias y Sistemas permitiendo la creación apoye en la toma de decisiones de diferentes áreas, en búsqueda de la mejora continua.

1.2. Situación actual

Actualmente, el sistema desarrollo de transferencia tecnológica no cuenta con el módulo de evaluaciones formativas; por lo cual se ve la necesidad de la implementación, con el fin de reducir el grado de anomalías en los resultados de las evaluaciones y evitar el uso de una herramienta externa ofrecida por terceros. La desventaja principal al usar herramientas de terceros es que no queda ninguna constancia o registro de la evaluación que pueda ser revisada en cualquier momento por las autoridades correspondientes; así mismo, el resultado de las evaluaciones pasaría directamente a la nota que corresponda, y se reduciría altamente el grado de error humano al ingresar manualmente las notas por parte del catedrático o tutor académico a cargo.

1.3. Requerimientos para el módulo de evaluaciones online

A continuación, se definirán los requerimientos de alto nivel del sistema para la creación de evaluaciones en el sistema DTT.

1.3.1. Definición de requerimientos

El sistema debe cumplir como mínimo con las siguientes funcionalidades:

- Creación de evaluaciones: en esta sección se espera poder asignar la evaluación a su actividad correspondiente, por ejemplo, la evaluación pertenecerá a la actividad examen corto 1, de esa forma, se sabrá la ponderación de la evaluación, debido a que las actividades cuentan con ponderación actualmente.

- Creación de 'n' preguntas: permitir crear 'n' preguntas asignadas a una evaluación; el rol de catedrático o tutor académico podrá crear las preguntas que considere necesarias para cubrir dicha evaluación; él podrá crear los 3 tipos de preguntas que proporciona el módulo.
- Creación de preguntas tipo TF: estas preguntas serán de tipo cerradas ya que la única respuesta posible será *true* (sí) o *false* (no); para ello el rol catedrático y tutor académico solamente tendrá que ingresar el enunciado de la pregunta y colocar si la respuesta correcta es *true* o *false*.
- Creación de preguntas tipo selección: el rol catedrático y tutor académico ingresará el enunciado de la pregunta; así mismo, podrá crear, modificar y eliminar dinámicamente preguntas ya que el cambio se visualiza en tiempo real; al mismo tiempo, en la creación y modificación podrá seleccionar una o varias respuestas como correctas.
- Creación de preguntas directa: una evaluación debe permitir la creación de una o más preguntas; dichas preguntas deben ser de diferentes tipos; uno de los tipos requeridos es de respuesta directa en la que se le proporcionará al estudiante una caja de texto para escribir su respuesta.
- Guardar pregunta: la función permite guardar la pregunta en el cuestionario con los datos ingresados.
- Eliminar pregunta: una pregunta podrá ser removida de la lista de preguntas a través del botón 'eliminar' lo que implicará la eliminación definitiva de la pregunta.

- Modificar pregunta: una pregunta guardada solamente podrá ser modificada desde la opción modificar; esta función requerirá nuevamente que se guarde la pregunta para que los cambios surtan efecto.
- Guardar evaluación: la función permite guardar el cuestionario el cual será asignado al curso desde el cual se ha creado y aparecerá en el banco de evaluaciones del curso.
- Modificar evaluación: una evaluación guardada podrá ser modificada desde esta opción para posteriormente guardar los cambios efectuados.
- Iniciar evaluación para estudiantes: el rol catedrático y tutor académico al terminar de crear la evaluación podrá dar por iniciada la evaluación.
- Ingresar a evaluación: todo usuario identificado como estudiante y perteneciente a un curso donde una evaluación esté disponible podrá tomar la evaluación en la fecha y hora de disponibilidad.
- Asignar notas al estudiante: las notas deben cargarse al sistema de forma automática cuando el estudiante concluya la evaluación.
- Generar reporte de resultados tipo informe: tras finalizada una evaluación, el tutor académico podrá generar reportes estadísticos que deben incluir información relevante: media, moda, mediana, dispersión, porcentaje de alumnos que aprobaron y porcentaje de alumnos que reprobaron.
- Imprimir reportes: los reportes generados en cada evaluación podrán ser impresos si el tutor académico lo desea.

1.3.2. Actualidad del sistema DTT

El sistema DTT se ha convertido en la herramienta primordial de control académico para la Escuela de Ingeniería en Ciencias y Sistemas; desde este sistema se gestionan todos los cursos del área profesional de esta carrera; permite registrar estudiantes, catedráticos y tutores académicos; cada curso puede tener una o más secciones y puede tener tanto la modalidad de clase magistral en la que el responsable del control de notas es un ingeniero catedrático o la modalidad de práctica o laboratorio en la que el encargado de la gestión de notas es un tutor académico.

Los estudiantes pueden pertenecer a más de un curso en un mismo periodo académico; pueden acceder a la planificación del curso en cualquier momento y verificar las notas actuales obtenidas. Desde un punto de vista técnico, el sistema está elaborado bajo una arquitectura de software MVC (modelo, vista, controlador) y posee tecnologías como Web2py y MariaDB. No existe ninguna funcionalidad que automatice la carga de notas o permita la creación de evaluaciones.

1.3.3. Mercado objetivo

Todos los estudiantes, tutores académicos y catedráticos de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala podrán hacer uso de las evaluaciones formativas; en específico, el rol de tutor académico es quien realizará las evaluaciones como exámenes cortos, hojas de trabajo o exámenes finales de laboratorio; el catedrático podrá realizar exámenes cortos, exámenes parciales e inclusive exámenes finales; el rol de estudiantes será el grupo al que se evaluará con dichas pruebas.

2. ASPECTOS GENERALES

En este capítulo se tratan los aspectos generales a considerar para este proyecto y se describe la fase de análisis y diseño; además, se explica de forma breve las consideraciones tomadas para la elección de las tecnologías que se han utilizado, los beneficios que estas dan respecto a otras tecnologías similares y la inversión necesaria que ha permitido la puesta en marcha del proyecto.

2.1. Descripción general del proyecto

El proyecto consiste en la elaboración de un nuevo módulo de software a incluir en el proyecto de transferencia de tecnología, DTT, por sus siglas, utilizado en la Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería, Universidad de San Carlos de Guatemala, que incorpora las funcionalidades necesarias y pertinentes que permiten la implementación de la metodología de evaluación de aprendizaje basada en exámenes.

La metodología está compuesta de las etapas: planificación, implementación y evaluación; esta última es de tipo examen, elaborado como un conjunto de preguntas; dichas preguntas pueden ser de tres tipos: pregunta de opción múltiple, pregunta de falso o verdadero y pregunta de respuesta directa.

La fase de planificación e implementación es realizada por los tutores académicos y/o ingenieros responsables de las cátedras; el sistema DTT una es importante herramienta de apoyo. El sistema solicita a los encargados de

cada curso registrar la planificación en los primeros días de cada periodo académico; se debe cumplir con los requerimientos establecidos por la facultad para el curso específico y se analiza por el usuario administrador del sistema; rol que ocupa el coordinar del proyecto DTT, en la actualidad, el ing. Miguel Marín. Un aspecto importante del sistema DTT es que permite a los tutores e ingenieros seguir la metodología de trabajo que ellos consideren apropiada para el curso que imparten, al establecer las actividades, fechas y ponderaciones según consideren.

Para la fase de evaluación se incorpora este módulo de software en el sistema DTT; permite la creación de exámenes y su publicación para que los estudiantes puedan rendirlos en las fechas establecidas; es necesario que el tutor académico y/o ingeniero de cátedra relacione una evaluación existente en la galería de evaluaciones del curso a una actividad incluida en la planificación; esto provocará que la calificación de la actividad se realice de forma automática y se almacene en los registros del sistema. La estadística estará disponible al finalizar la evaluación; permite ver información como el número de estudiantes aprobados y reprobados (se aprueba con 61 puntos según lo establecido en la universidad).

Se implementa un modelo arquitectónico de tres capas: modelo, vista y controlador; sigue la arquitectura base del sistema DTT, escrito en el lenguaje de programación Python en su plataforma web llamada Web2py. Por motivos de rendimiento y acople de funcionalidad, la base de datos de este módulo se maneja con el motor de base de datos no relacionales llamado Redis a través de estructuras de datos en formato JSON, más adelante, en este capítulo, se presentan la descripción y justificación que han llevado a la elección de esta tecnología y los beneficios que se han obtenido con su implementación.

2.2. Análisis y diseño del sistema

El sistema DTT ha sido diseñado bajo el paradigma arquitectónico de software modelo, vista, controlador (MVC). Se ha implementado en el marco de trabajo Web2py que es un marco de trabajo web para el lenguaje de programación Python.

Web2py es un marco de trabajo que utiliza el lenguaje de programación Python, y un paradigma o arquitectura basada en tres capas: modelo-vista-controlador, MVC. La investigación basada en diseño, IBD, se enfoca principalmente en la migración de procesos que se realizan de forma tradicional, sin uso de sistemas de software, a su rediseño y optimización apoyándose de las tecnología de información, TI.

La elección de la arquitectura de software es sumamente importante en una investigación basada en diseño, ya esta es capaz de restringir la funcionalidad y el desempeño de la lógica del negocio, lo que podría repercutir directamente en la reingeniería del proceso en investigación. Web2py, como casi cualquier marco de trabajo, se adapta perfectamente a las investigaciones basadas en diseño; es muy importante considerar los alcances del proceso a investigar y la lógica de negocio; aunque casi cualquier marco de trabajo se ajusta a una metodología de investigación basada en diseño, el proceso es el que restringe y guía a la elección del marco de trabajo adecuado. En el caso específico de esta investigación, considerando los alcances y el flujo del proceso, Web2 se ajusta de forma adecuada a la reingeniería del proceso, gracias a la división lógica que Web2py realiza de la aplicación, dividiéndola en tres capas.

La capa de modelo de datos se conectará directamente con MariaDB y el sistema de base de datos NO SQL, no relacional, lo que permitirá que cualquier cambio en su data o estructura pueda realizarse interviniendo directamente en esta capa, a menos que el cambio requiera modificar otra.

La capa de vista implementa la tecnología actual de HTML5, un estándar que trabaja con prácticamente todos los navegadores y sistemas operativos; esta capa únicamente se encarga del aspecto visual del sistema, por lo que cualquier cambio en el mismo solo afecta la capa de vista.

La capa de controlador, donde se encuentra la lógica del negocio, se realiza con el lenguaje de programación Python; esta capa envuelve toda la lógica del sistema.

2.3. Sistemas de bases de datos

El sistema DTT implementa actualmente el almacenamiento de datos bajo el sistema MariaDB, antes MySQL; este es un sistema de bases de datos relacional; almacena los datos en forma de tablas; considerada esta implementación existente, se aprovechará dicha tecnología para almacenar los datos estructurados que se generan en este nuevo módulo.

La investigación basada en diseño, en casi todo los procesos a investigar, busca mantener la persistencia de la información; por lo tanto, los sistemas de base de datos son vitales para esta metodología de información, ya que permiten la persistencia de la información estructurada y el control de los procesos de negocio a nivel operativo, donde se utilizan con mayor frecuencia los sistemas relacionales. MariaDB, al ser un sistema de base de datos relacional, tiene mucha importancia y participación vital en la investigación

basada en diseño y se acopla perfectamente al fin de la misma: hacer los procesos existentes más óptimos a través del uso de tecnologías de la información TI.

2.4. Bases de datos SQL vs bases de datos NO SQL

El sistema de base de datos relacional utilizado en el sistema DTT es MariaDB, por lo que en esta sección se hablará de esta tecnología y de los sistemas no relacionales o No SQL a considerar en la implementación de este proyecto.

2.4.1. Bases de datos SQL

Existen una alta gama de sistemas de gestión de bases de datos, DBSM, por sus siglas en inglés. El sistema DTT estructura su modelo de datos en MariaDB, un sistema de gestión de base de datos relacional; en la actualidad, estos sistemas son los más utilizados para el modelado de la información en prácticamente cualquier tipo de negocio; permiten estructurar la data en forma de tablas, semejante al modelo utilizado en las hojas de Microsoft Excel; los humanos suelen percibir dicha relación de esta forma, aunque internamente el sistema de base de datos se encarga de organizar la data para su óptimo funcionamiento.

2.4.2. Bases de datos NO SQL

Los sistemas de bases de datos no relacionales, NoSQL, rompen el paradigma tradicional de almacenar la data en forma de tablas; básicamente, están enfocados al manejo de información no estructurada, lo que hace que sean sumamente rápidos en comparación con los sistemas tradicionales. La

metodología de información basada en diseño puede requerir la implementación de tecnologías; sin embargo, se debe ser muy cuidadoso ya que depende directamente del proceso, la necesidad de aplicar o no estas tecnologías. En esta investigación, por demanda del proceso y los requerimientos del proyecto, el sistema NoSQL se ajusta a la investigación basada en diseño.

La elección de una base de datos NoSQL es más compleja debido a que existen unos 150 sistemas de bases de datos NoSQL. Elegir uno puede ser muy difícil ya que ninguno ha obtenido todavía la fama que sí han conseguido las bases de datos relacionales.

El problema principal es la gran variedad de tipos de sistemas NoSQL. Dependiendo de lo que se necesite, se debe destacar una u otra.

2.4.2.1. Bases de datos documentales

Son aquellas que gestionan datos semiestructurados, es decir, documentos. Estos datos son almacenados en algún formato estándar como puede ser XML, JSON o BSON. Para hacerse una idea un documento suele ser algo parecido a la estructura mostrada en la figura 1.

Figura 1. Documento de base de datos documental

```
{
  Name: "Genbeta Dev",
  Tipo: "Blogging",
  Categorías: [{
    Título: "Desarrollo",
    Artículos: 89
  }, {
    Título: "Formación",
    Artículos: 45
  }]
}
```

Fuente: elaboración propia.

Son las bases de datos NoSQL más versátiles. Se pueden utilizar en gran cantidad de proyectos, incluyendo muchos que tradicionalmente funcionarían sobre bases de datos relacionales.

En esta categoría se encuentran:

- MongoDB: probablemente la base de datos NoSQL más famosa del momento. En octubre del año pasado, MongoDB conseguía 150 millones de dólares en financiación, convirtiéndose en una de las *startups* más prometedoras. Algunas compañías que actualmente utilizan MongoDB son Foursquare o eBay.

- CouchDB: es la base de datos orientada a documentos de Apache. Una de sus interesantes características es que los datos son accesibles a través de una API Rest. Este sistema es utilizado por compañías como Credit Suisse y la BBC.

2.4.2.2. Bases de datos en grafo

Basadas en la teoría de grafos, utilizan nodos y aristas para representar los datos almacenados. Son muy útiles para guardar información en modelos con muchas relaciones, como redes y conexiones sociales.

En esta categoría se encuentran:

- Infinite Graph: escrita en Java y C++ por la compañía Objectivity. Tiene dos modelos de licenciamiento: uno gratuito y otro de pago.
- Neo4j: base de datos de código abierto, escrita en Java por la compañía Neo Technology. Utilizada por compañías como HP, Infojobs o Cisco.

2.4.2.3. Bases de datos clave/valor

Estas son las más sencillas de entender. Simplemente guardan tuplas que contienen una clave y su valor. Cuando se quiere recuperar un dato, simplemente se busca por su clave y se recupera el valor.

En esta categoría se encuentran:

- DynamoDB: desarrollada por Amazon, es una opción de almacenaje que se puede usar desde los Amazon Web Services. La utilizan el Washington Post y Scopely.
- Redis: desarrollada en C y de código abierto, es utilizada por Craigslist y Stack Overflow (a modo de caché).

2.4.2.4. Bases de datos multivalor

Las bases de datos multivaluadas contienen datos arracimados, en el sentido de que pueden almacenar los datos del mismo modo que las bases de datos relacionales; además, permiten un nivel de profundidad al que las relacionales solo se pueden aproximar utilizando subtablas. Esto es prácticamente igual al modo en que XML representa los datos, donde un campo/atributo dado puede contener múltiples valores a la vez. El multivalor se puede considerar una forma de XML comprimida.

Un ejemplo puede ser una factura, la que puede ser vista como:

- Encabezado, una entrada por factura
- Detalle, una entrada por concepto

En el modelo multivaluado se tiene la opción de almacenar los datos como una sola tabla, con tablas imbuidas representando el detalle.

Tiene la ventaja de que la correspondencia entre la factura conceptual y la de la factura como representación de datos es biunívoca. Esto redundante en menor número de lecturas, menos problemas de integridad referencial y una

fuerte disminución del hardware necesario para soportar un volumen de transacciones dado.

2.5. Elección de tecnologías

La capa de datos para este proyecto se basará en un modelo de datos no relacional, específicamente, un modelo de datos clave valor, con el fin de aprovechar la versatilidad de estos modelos y el acople que presentan en la creación de cuestionarios; debido a que diferentes cuestionarios pueden tener diferentes números de preguntas y cada pregunta puede ser de diferente tipo.

2.5.1. Comparativa de bases de datos clave/valor

En esta sección se realizará una comparación de los DBMS más importantes para bases de datos clave valor; se da una breve descripción y se citan sus ventajas ante los demás.

2.5.1.1. Apache Cassandra

A continuación, se describe brevemente este DBSM y las ventajas ante otras bases de datos clave/valor.

2.5.1.1.1. Descripción general

Apache Cassandra, conocida como Cassandra, es un sistema de base de datos distribuido y está diseñado principalmente para almacenar y administrar grandes cantidades de datos entre sus diferentes servidores (distribuidos) básicas. Cassandra está diseñada para utilizarla como clúster de nodos simétricos en vez de la arquitectura maestro-esclavo.

Esto para asegurar que no haya puntos de fallos. Cassandra se encarga de particionar automáticamente los datos de todos los nodos del clúster.

2.5.1.1.2. Ventajas y desventajas

A continuación, se describen las ventajas de esta DBMS:

- El sistema de escritura de Cassandra puede ser utilizado como un almacén de datos operativos en tiempo real para aplicaciones en línea transaccionales.
- Con esto aumentaría la utilidad de la página para ingresar información a la base de datos conforme se vaya utilizando la página.
- Para operaciones de lectura puede ser utilizado como base de datos de lectura intensiva a gran escala o como *business intelligence*.

2.5.1.2. MongoDB

A continuación, se describe brevemente este DBSM y las ventajas ante otras bases de datos clave/valor.

2.5.1.2.1. Descripción general

MongoDB, mejor conocido como Mongo, fue uno de los varios sistemas de base de datos bajo el nombre NoSQL a mediados de los años 2000. Esta arquitectura trabaja de forma de colecciones y documentos. Los documentos se componen de conjuntos de pares clave-valor que son la unidad básica de datos en esta arquitectura.

2.5.1.2.2. Ventajas

A continuación, se describen las ventajas de esta DBMS:

- La arquitectura de Mongo se basa en la colección de conjunto de documentos y funcionan equivalente a tablas de base de datos relacionales.
- La manera de obtener los datos de forma clave-valor puede ser tomada como un reto para una implementación de sistemas escalables.
- El sistema soporta un diseño de esquema dinámico permitiendo que los documentos tengan diferentes campos y estructuras para un mismo tipo de clave-valor.

2.5.1.3. Redis

A continuación, se describe brevemente este DBSM y las ventajas ante otras bases de datos clave/valor.

2.5.1.3.1. Descripción general

Redis es un sistema de almacén de estructuras de datos utilizado como base de datos, cache y aplicaciones de mensajería. Tiene un límite de estructuras de datos con los cuales se pueden hacer operaciones entre los mismos. Las estructuras que maneja Redis son cadenas, hashes, listas, conjuntos, conjuntos ordenados con consultas de rango, mapas de bits, *hiperlogs* e índices geoespaciales con consultas de radio.

La manera de operaciones es de incrementar/decrementar listas o añadir a cadenas, la cual se vuelve un proceso del sistema Redis.

2.5.1.3.2. Ventajas

A continuación, se describen las ventajas de esta DBMS:

- Redis proporciona muchas transacciones y diferentes persistencias en el disco. Proporciona alta disponibilidad a través de su arquitectura y clúster.
- La persistencia del sistema Redis puede estar opcionalmente deshabilitada si solo se necesita una caché de memoria en red.

2.5.1.4. Conclusión sobre comparativa

Cuando se hizo el análisis del sistema obteniendo, ventajas y desventajas, se plantearon los siguientes puntos:

- Alta concurrencia
- Rendimiento de la página
- Memoria caché
- Persistencia en disco
- *Open source*

Tomando en cuenta los puntos anteriores, se llegó a la conclusión de que Redis cumple con los parámetros ingresados:

- Teniendo mejor concurrencia Cassandra, se decidió por Redis ya que se podía manejar de forma caché aumentando el rendimiento de la página sin afectar toda la arquitectura en la que ya se encuentra el sistema.
- Se facilitará la unión de esta tecnología con el sistema existente ya que es *open source*.

2.5.2. Justificación

El uso de Redis permite utilizar los beneficios que ofrece esta base de datos, como la velocidad, debido a que esta base de datos mantiene la información almacenada en la memoria del computador; además de su facilidad de uso y su flexibilidad de procesamiento.

Redis, entre sus grandes beneficios, ofrece asociar valores de tipo cadena a una clave; esto permite utilizar tipos de datos avanzados, los cuales utilizados junto a operaciones asociadas, se pueden resolver muchos casos de negocio. Muchas veces se piensa que es muy complicado resolver cierto tipo de problemas con una base de datos clave-valor. No obstante, Redis también puede ser definido como un servidor de estructuras de datos, el cual era en sus inicios un diccionario remoto, de ahí se genera su nombre, REmote DIctionary Server.

En este caso, Redis es la herramienta que mejor se adapta a los requisitos del negocio, ya que las tecnologías NoSQL desde su concepción, tienen que ver con la especialización en la gestión de datos; Redis es un caso excelente en cuanto a resolver necesidades concretas de una necesidad con un motor de almacenamiento, por sí mismo, o en combinación con otras bases de datos NoSQL o relacionales.

Redis es usado por diferentes empresas para diferentes funcionalidades, algunos de los usos más usuales de esta base de datos, aparte de su función como base de datos principal, serían: caché de base de datos, contadores y estadísticas, lista de elementos recientes, almacenamiento de sesiones de usuario y caché de páginas web; estos y muchos más usos puede tener la aplicación de una base de datos como Redis.

2.6. Beneficios

Se puede definir que el uso de Redis conlleva muchos beneficios a tomar en cuenta: ser de tipo económico o funcional; uno de los principales beneficios que otorga Redis es ser de código abierto, lo que permite utilizarlo sin costo alguno; es más confiable y seguro como rápido de implementar, más aún que las alternativas propietarias.

Redis permite un escalamiento sencillo; este sustituye la manera usual de escalar de las bases de datos e integra una nueva manera de añadir más servidores para manejar más carga de datos; permite distribuir la carga entre varios hosts a medida que la carga se encuentra en aumento.

La facilidad que tiene una base de datos NoSQL como Redis para funcionar en la nube permite la implementación fácil y rápida de la misma en servidores para cuando exista la necesidad de ampliar el uso de servidores.

Finalmente, se detalla el beneficio principal del uso de Redis como base de datos: la velocidad con la que funcionan ya que estas utilizan la memoria en lugar del disco como principal ubicación de la escritura, esto permite el rendimiento de la aplicación. Los almacenes de datos NoSQL utilizan

generalmente particiones horizontales, por lo que son capaces de tomar ventaja en la nube de la versatilidad del aprovisionamiento.

2.6.1. Inversión

La elaboración de este proyecto requiere inversiones monetarias para la adquisición del equipo de computación para su desarrollo; además, inversión en horas de capacitación y estudio sobre el funcionamiento del sistema DTT en aspectos técnicos y la inversión en horas trabajo para las fases de análisis y diseño, desarrollo y pruebas para la conclusión exitosa del proyecto, las cuales se detallan a continuación:

- Equipo de cómputo: se requiere un ordenado con características similares a las del servidor del sistema DTT; los requerimientos son los siguientes:

Tabla I. **Especificaciones de hardware**

Procesador	1,4 GHz Intel Core i5
Memoria RAM	4 GB 1600 MHz DDR3
Gráficos	Intel HD Graphics 5000 1536 MB
Disco duro	251 GB Flash Storage

Fuente: elaboración propia.

Tabla II. **Especificaciones de software**

Sistema operativo	Ubuntu 14.04
Lenguaje de programación	Python 2.7
Plataforma de desarrollo	Web2py
Sistema de base de datos SQL	MariaDB
Sistema de base de datos NO SQL	Redis
Librerías	Java Script, JQuery, Bootstrap 2.3

Fuente: elaboración propia.

A continuación, se describe el tiempo necesario en horas para cada una de las etapas que se han requerido para la elaboración de este módulo:

Tabla III. **Tiempo invertido**

Capacitación	20 horas
Análisis y diseño	100 horas
Desarrollo	60 horas
Pruebas	40 horas

Fuente: elaboración propia.

Considerando que el equipo de cómputo tiene un costo de Q 8 000,00, el total de horas de trabajo demandadas es de 220 horas y que cada hora de desarrollo, análisis o pruebas en la categoría *senior* equivale a una inversión de Q 75,00 la inversión total del proyecto es de: Q 24 500,00

3. ANALISIS DE MÉTODOS PARA EVALUCIONES *ONLINE*

En esta capítulo se analizarán los diferentes métodos de evaluación existentes a través sistemas de software propiedad de la Universidad de San Carlos de Guatemala y los más importantes de acceso público y disponibles a través de internet.

3.1. Aspectos generales sobre sistemas de evaluación *online*

Los sistemas de evaluación *online* son sistemas que permiten al usuario la creación de evaluaciones conforme a sus necesidades; estos generalmente tienen banco de tipos de preguntas donde predominan los siguientes tipos:

3.1.1. Preguntas convencionales

Existen varias técnicas de examen, una es la modalidad de cuestionario en respuesta directa, también llamado, de preguntas convencionales. En este sentido se identifica como cuestionario de respuesta directa a aquel en el que se debe dar respuesta (se supone que escrita) a la pregunta que se hace. Por ejemplo, ante la pregunta ¿cómo se llama el autor de estas páginas?, deberá escribirse la respuesta: Dayton García.

En la modalidad de evaluaciones *online* se le conoce a este tipo de preguntas como preguntas de respuesta corta. En respuesta a una pregunta (que puede incluir una imagen), el alumno escribe una palabra o frase corta. Puede haber varias respuestas correctas posibles con diferentes puntuaciones. Las respuestas pueden o no tener importancia por MAYÚSCULAS/minúsculas.

3.1.1.1. Preguntas de opción múltiple

Las preguntas de opción múltiple existen con dos modalidades: con respuesta única y con respuesta múltiple. Los sistemas de evaluación *online* típicamente permiten incluir imágenes, sonido y multimedios en la pregunta o incluirlos en la respuesta.

Tipos de preguntas de opción múltiple:

- Preguntas de respuesta única: permite elegir únicamente una respuesta del banco de posibles opciones, es decir, se dan varias opciones pero solo una es la correcta.
- Preguntas de respuesta múltiple: permiten seleccionar una o más respuestas por medio de casillas de verificación. Cada contestación puede tener una puntuación positiva o negativa, lo que implica que seleccionar todas las opciones no necesariamente supone una buena puntuación. Si la puntuación total es negativa, entonces, la puntuación resultante para esta pregunta será cero.

3.1.1.2. Preguntas de falso/verdadero

En respuesta a una pregunta (la cual puede incluir una imagen), el alumno selecciona entre dos opciones: verdadero o falso.

Típicamente, los sistemas de evaluaciones *online* permiten habilitar la retroalimentación; el mensaje de retroalimentación se muestra al alumno después de contestar el examen. Por ejemplo, si la respuesta correcta es 'falso',

pero se contestó 'verdadero' (por lo tanto, incorrecto), entonces, se muestra la retroalimentación 'falso'.

Ante este tipo de preguntas es necesario que se contemplen ciertos aspectos; a continuación, se enumeran algunos:

- Las preguntas de tipo falso / verdadero, al tener solamente dos opciones, tienen un porcentaje de éxito por adivinar aleatoriamente del 50 %; por lo que una pregunta falso / verdadero con un índice de facilidad demostrado del 50 % no sería un buen reactivo para hacer una evaluación objetiva. Los niveles de dificultad deseables usualmente son estimados en un punto medio entre 100 % y el porcentaje de éxito por adivinar, aleatoriamente. Así para preguntas de falso / verdadero, el índice de facilidad deseable sería alrededor de 75 %. Tener presente que solamente será conocido el índice de facilidad de una pregunta hasta después de haberla empleado en un examen.
- Hacer oraciones que sean definitivamente (siempre 100 %) ciertas o definitivamente falsas. Evitar oraciones que podrían ser verdaderas o las que podrían ser falsas en ciertos casos.
- Usar oraciones cortas.
- Evitar contenido extraño que no venga al caso.
- Hacer aproximadamente de la misma longitud las oraciones ciertas y las falsas.
- Evaluar solamente una idea por pregunta.

- Evitar las pistas verbales, determinantes específicos de género y número (un/una, varios...) y evitar oraciones complejas.
- Evitar usar términos absolutos como siempre o nunca, exceptuando los casos donde esto es característico del conocimiento evaluado.
- Evitar copiar oraciones de los libros de texto.
- Siempre escribir la pregunta de forma positiva.

3.1.2. Tecnologías disponibles para evaluaciones *online*

Esta sección describe las tecnologías disponibles actualmente para evaluaciones *online*; se tratarán en un apartado las que están inmediatamente disponibles dentro de la Universidad de San Carlos de Guatemala y a las que se puede acceder de forma pública a través de internet.

3.1.2.1. Disponibles en la Universidad de San Carlos de Guatemala

La Universidad de San Carlos de Guatemala, específicamente en la Facultad de Ingeniería, dispone de dos sistemas de software para evaluaciones; en esta sección se detallan sus principales características.

3.1.2.1.1. Sistema de evaluación para pruebas específicas de computación (SAE/SAP) en la Facultad de Ingeniería

Este sistema de software está desarrollado en el lenguaje de programación web PHP e implementa la base de datos a través del DBSM MySql. Es un sistema que presenta una serie de preguntas que conforman la evolución del examen específico de computación para las personas que desean ingresar a la facultad; una de sus características principales es que evitar la intervención humana para la calificación de dichas pruebas, debido a que compara las respuestas que el usuario ha ingresado con las del banco de preguntas que se ha almacenado en su base de datos; sin embargo, una de las principales limitaciones de este sistema es que únicamente dispone de un pequeño banco de cuestionarios los cuales han sido creado desde la elaboración del sistema de software y no es posible modificarlo o personalizar los cuestionarios; otra limitante es que únicamente se puede acceder a ellos desde los laboratorios SAE/SAP y están diseñados para soporta concurrencia de veinte usuarios, lo que puede considerar un punto débil en caso de que se pretenda expandir los alcances de estas pruebas.

En resumen, este sistema es un sistema de evaluaciones estático y está diseñado específicamente para una función: evaluar las pruebas específicas de computación en la Facultad de Ingeniería, función que cumple a cabalidad pero que no permite la personalización de cuestionarios, como otras herramientas *online*.

3.1.2.1.2. Sistema de evaluación para examen de ubicación de idioma técnico en la Facultad de Ingeniería

Este sistema es similar al sistema de evaluaciones de SAE/SAP y consiste en un pequeño banco de cuestionarios diseñados de forma estática y que se usan para hacer el examen de ubicación de las personas que desean acceder al curso de idioma técnico; para acceder a este examen es necesario que el usuario esté registrado y que el administrador haya creado una sesión para realizar el examen a las personas que se agreguen; un factor a resaltar es que el examen se programa para una hora específica y se activa automáticamente cuando es la hora indicada, de igual forma la desactivación es programada automáticamente, lo que libera al tutor del trabajo de cronometrar la duración de la evaluación y estar pendiente de los horarios de inicio y final.

En aspectos técnicos, el sistema se ha desarrollado en lenguaje de programación PHP y la bases de datos que utiliza es una base de datos relacional, lo que implica un mayor consumo de recursos y que se penalice en rendimiento ante alta concurrencia; lo que ha llevado a los encargados del Departamento de Idioma Técnico a realizar la evaluación presentado principal interés en el control de usuario que se asignan a determinada sección para garantizar el funcionamiento en condiciones del software.

3.1.2.2. Disponibles en internet

Los sistemas de evaluaciones *online* han tomado importancia en los últimos años, lo que ha generado que varias empresas de desarrollo de software ponga los ojos en este tipo de aplicaciones; muchas han fracasado

debido a la tímida adopción de los usuarios, pero otra tantas han tenido un éxito rotundo y en la actualidad son usadas en muchas universidades y colegios a nivel mundial. En esta sección se hablará de las aplicaciones de software disponibles en internet para la creación de exámenes *online* más exitosas en la actualidad.

3.1.2.2.1. Socrative

El software de creación de evoluciones en línea más utilizado en la actualidad es un software bastante completo y disponible en dos versiones: una versión libre, Socrative Free, que da al usuario un panorama con un banco importante de funciones y una versión de pago, Socrative Pro. En la figura 2 encontrara una comparativa entre ambas versiones extraída de la documentación oficial de Socrative.

Figura 2. **Comparativa entre versiones de Socrative**

Socrative Free	Socrative PRO
<ul style="list-style-type: none">• On-the-fly Questioning• 1 public room for your classes• 50 students per session• Space Race Assessment• Formative Assessments• Visualize Real-time Results• Device Accessibility• Reporting• Share with an SOC code• Help Center Access• State and Common Core Standards	<ul style="list-style-type: none">• Everything with Free, plus all the following:• Up to 10 private or public rooms• 150 student capacity• Space Race countdown timer• Roster Import via CSV or Excel• Restricted access to rooms with students ID# requirement• Personalized header for students• Instant quiz share to colleagues with unique link• Space Race custom icon upload• Silent Student Hand Raise• Shareable links for easy student login• Searchable quiz community• Dedicated Help Center & email support

Fuente: *Socrative pro & free*. www.socrative.com. Consulta: 20 de octubre de 2017.

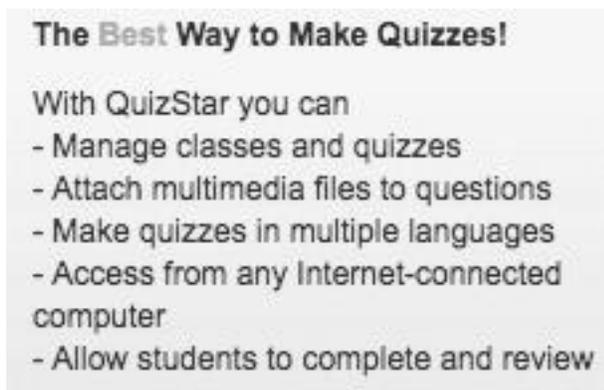
Este software contempla prácticamente todas las características importantes para este tipo sistemas; además de tener un buen rendimiento y garantizar tiempo de respuesta adecuados, lo que lo convierte en un gran sistema para evaluaciones; sin embargo, el factor a considerar para desarrollar un sistemas de evaluaciones propio de la Escuela de Ingeniería en Ciencias y Sistemas es la centralización de módulos de software en el sistema DTT y la carga automática de notas que evita la intervención humana por parte del catedrático y/o tutor académico; además de que se requiere la posibilidad de

almacenar un banco de exámenes que puedan ser reutilizables en cada curso, lo que conlleva la creación de un módulo propio.

3.1.2.2.2. Quizstar de 4teachers

Es un módulo de creación de examen gratuito al igual que otros módulos de 4teachers; todas las cuentas suscritas en este dominio son gratuitas y no requieren renovación. Este sistema permite acceder en dos roles distintos: maestros y estudiantes; los maestros pueden diseñar los cuestionarios personalizados y publicarlos a un listado de usuarios que debe informarse en la publicación. Las características principales de este software se pueden visualizar en la figura 3.

Figura 3. **Características principales de Quizstar**



Fuente: *Quizstar*. <http://quizstar.4teachers.org/>. Consulta: 20 de octubre de 2017.

3.1.2.2.3. QuizWorks

Utiliza el juego y la gamificación de contenidos, el compromiso de las personas y el aprendizaje alternativo. Sus servicios para la comunidad de

recursos humanos incluyen *branding* de empleadores, compromiso de empleados y aprendizaje y desarrollo a través de preguntas y gamificación. Para la comunidad de marketing, sus servicios se centran principalmente en la educación del cliente, el compromiso de socios comerciales y soluciones de marca diseñadas a medida.

4. ARQUITECTURA DE LA APLICACIÓN

En este capítulo se describe de forma técnica a través de diagramas estandarizados a nivel mundial para representación de la arquitectura de software; el modelo arquitectónico del módulo de software a integrar en el sistema DTT.

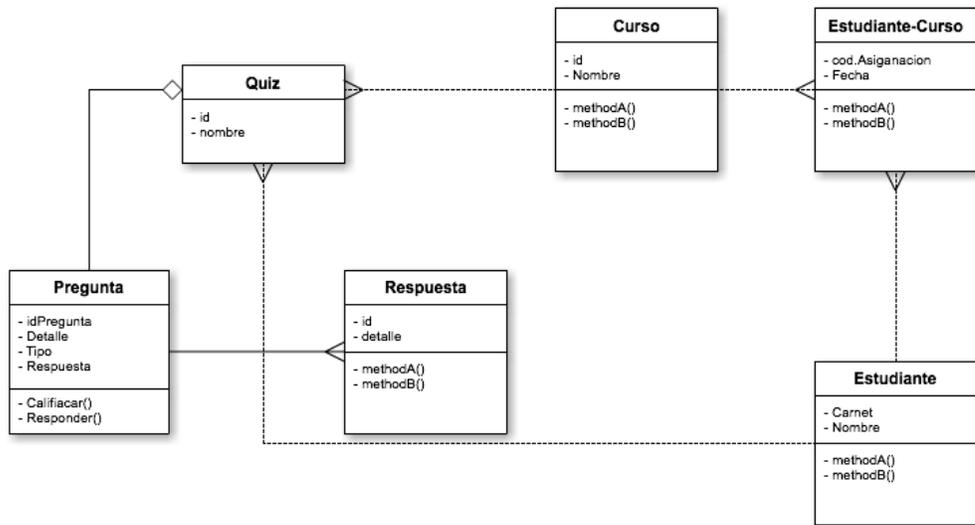
4.1. Vista lógica

A continuación, se describen y muestran los diagramas que corresponden a la vista lógica de la arquitectura software de este proyecto.

4.1.1. Diagrama de clases

En la figura 4 se puede observar que la clase 'Evaluación' está compuesta por un conjunto de preguntas, con una o más respuestas. Los estudiantes pueden acceder a los exámenes que estén relacionados con los cursos a los cuales se hayan asignado, lo que permite a varios estudiantes rendir un mismo Quiz en diferentes instancias.

Figura 4. Diagrama de clases

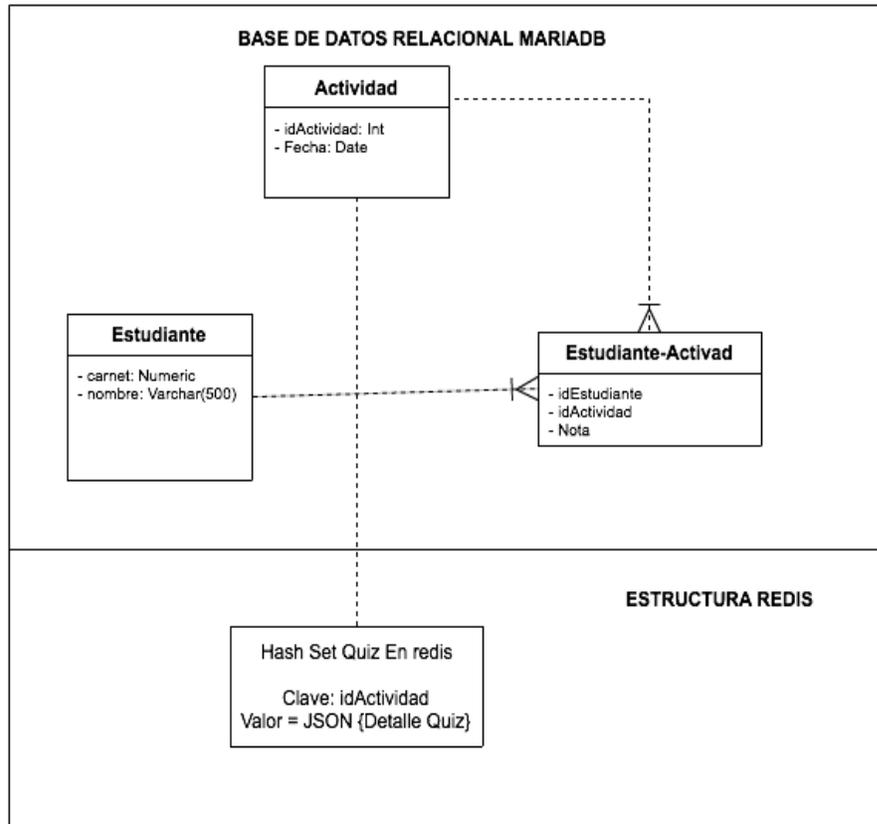


Fuente: elaboración propia.

4.1.2. Diagrama de datos

La base de datos principal se encuentra diseñada en MariaDB, para fines de este módulo son de interés las estructuras descritas arriba, tabla 'actividad', tabla 'estudiante' y la tabla que relaciona a estos dos, que en este caso es estudiante-actividad. El sistema de base de datos relacional está diseñado en el sistema DTT por lo que se aprovecha el modelo existente para generar el número de actividad; dicho número de actividad será exportado a Redis (sistema de base de datos no relacional), donde este número de actividad tomará lugar como la clave, de a estructura clave valor de un Quiz, la cual será almacena en un HASH SET proveído por Redis, lo que provee de un mejor desempeño y mejores tiempos de respuesta ante la alta concurrencia. Vale la pena aclarar que el detalle del Quiz en Redis expirará después de 3 meses.

Figura 5. Diagrama de base de datos



Fuente: elaboración propia.

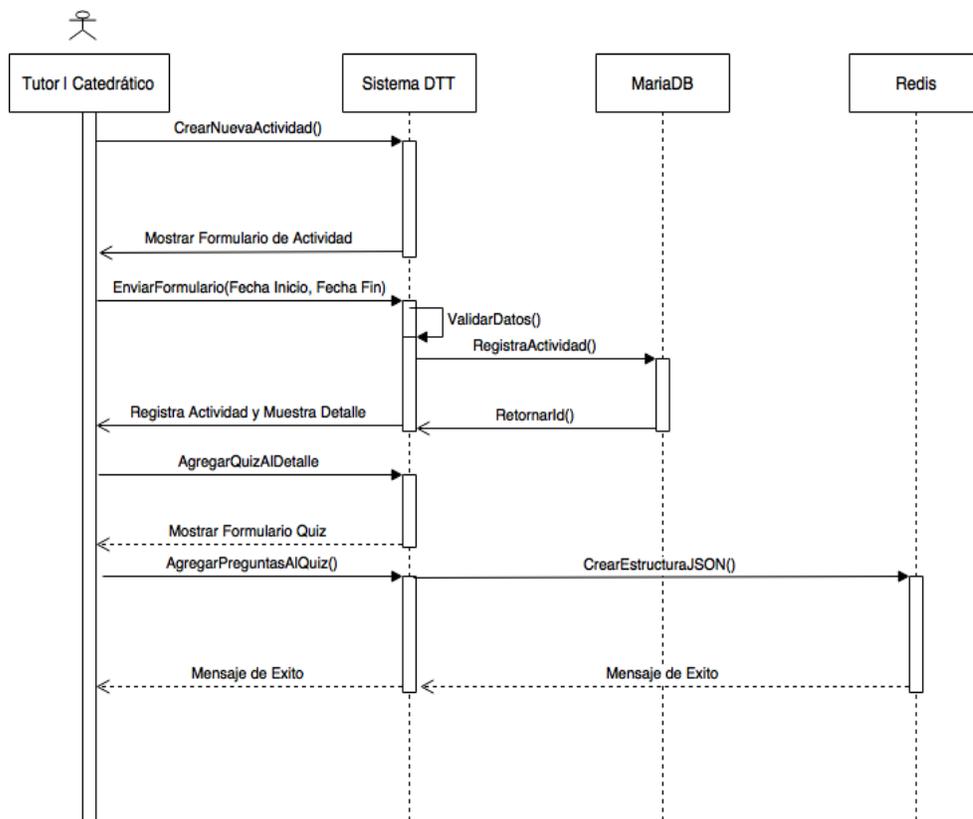
4.1.3. Diagramas de secuencia

En la figura 6 se puede observar la secuencia de interacciones para los casos de uso correspondientes a los tutores académicos y/o catedráticos. Como se observa la primer acción es hacer clic en la opción 'crear nueva actividad', lo que provocará una respuesta por parte del sistema DTT, que muestra el formulario para registrar una actividad; dicho formulario debe ser completado con la información correspondiente a la actividad al concluir se enviar al sistema DTT quien verificará que los datos sean correctos; tras esta

verificación se procede a registrar la actividad en la base de datos MariaDB, donde se genera un código automático que identificará la actividad como única.

En el detalle de la actividad, el tutor y/o catedrático procederá a agregar la lista de preguntas que formarán parte del cuestionario; al concluir con dicha lista se guarda el formulario y el sistema DTT envía Redis (gestor de base de datos no relacional), los datos del cuestionario en forma de una estructura JSON, dicha estructura es la que se persistirá en la base de Redis.

Figura 6. Diagrama de secuencia para tutores y catedráticos

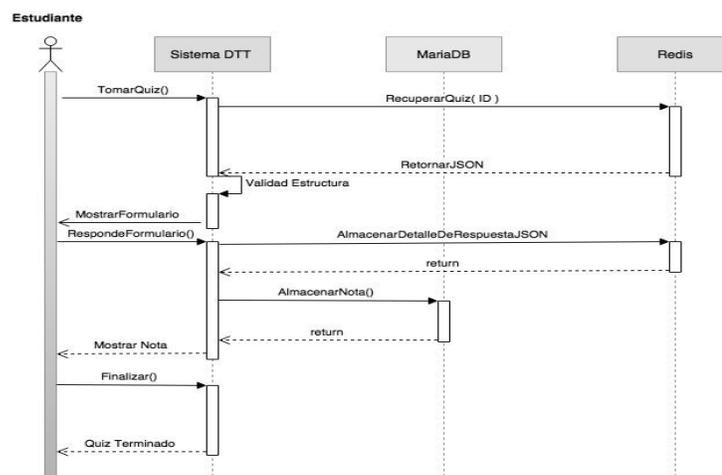


Fuente: elaboración propia.

La figura 7 ilustra la secuencia de los casos de uso para los estudiantes; como se puede observar la actividad inicia cuando el estudiante selecciona la opción 'tomar quiz' desde el sistema DTT, dentro del curso correspondiente (esta opción solo estará activada si hay un Quiz programado para la fecha y hora indicada).

El sistema DTT procede a recuperar el detalle del Quiz desde Redis realizando una consulta por identificador de la actividad a la base de datos; al recuperar esta información, el sistema verifica que la estructura sea la correcta y la adapta a un formulario web para que el estudiante pueda visualizar el contenido y responderlo; al concluir con todas las respuesta, el estudiante envía la información al sistema DTT quien procede a almacenar el detalle de las respuestas en la base de datos Redis; mientras en la base de datos MariaDB se almacena únicamente la nota obtenida por el estudiante. Al finalizar el cuestionario se muestra la nota obtenida y el botón 'Finalizar'; al hacer clic en dicho botón se concluye con la actividad.

Figura 7. Diagrama de secuencia para estudiantes



Fuente: elaboración propia.

4.2. Vista de implementación

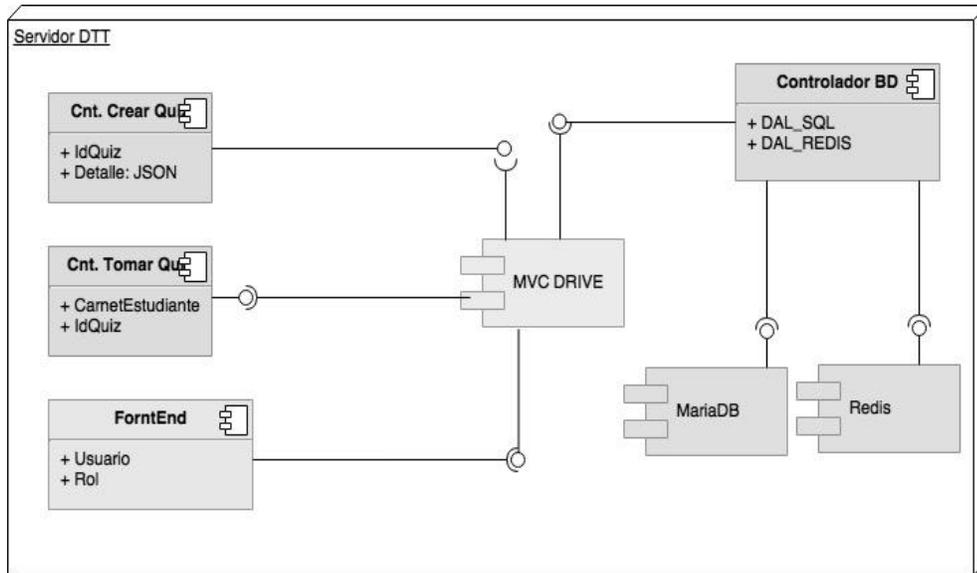
A continuación, se describen los diagramas que corresponden a la vista de implementación de la arquitectura software de este proyecto.

4.2.1. Diagrama de componentes

La figura 8 muestra la interacción de los diferentes componentes de software localizados en el servidor de la DTT; se pueden apreciar los controladores en color verde; la vista o *frontend* en color anaranjado y los modelos en lila. Se localizan dos sistemas de bases de datos diferentes: uno bajo el paradigma SQL y el otro un NOSQL; siendo este último es esencial para esta investigación al ser el servidor de base de datos donde se almacena la estructura de un quiz en formato JSON.

El sistema DTT ha sido construido con una arquitectura MVC (modelo, vista, controlador); esta arquitectura separa la aplicación en tres diferentes capas para un mejor control; la capa de datos (modelo) es la encargada de controlar los accesos a la base de datos y desde esta se configuran dichos accesos; la capa de vista es la encargada de controlar y gestionar las diferentes vista o formularios que se le presentan al usuarios final; para efectos de esta investigación se localiza el *frontend* con las diferentes vistas de la aplicación; por último, pero no menos importante la capa del controlador que es donde se gestiona toda la lógica del negocio y hace posible la ejecución de las acciones en cada uno de los escenarios.

Figura 8. Diagrama de componentes

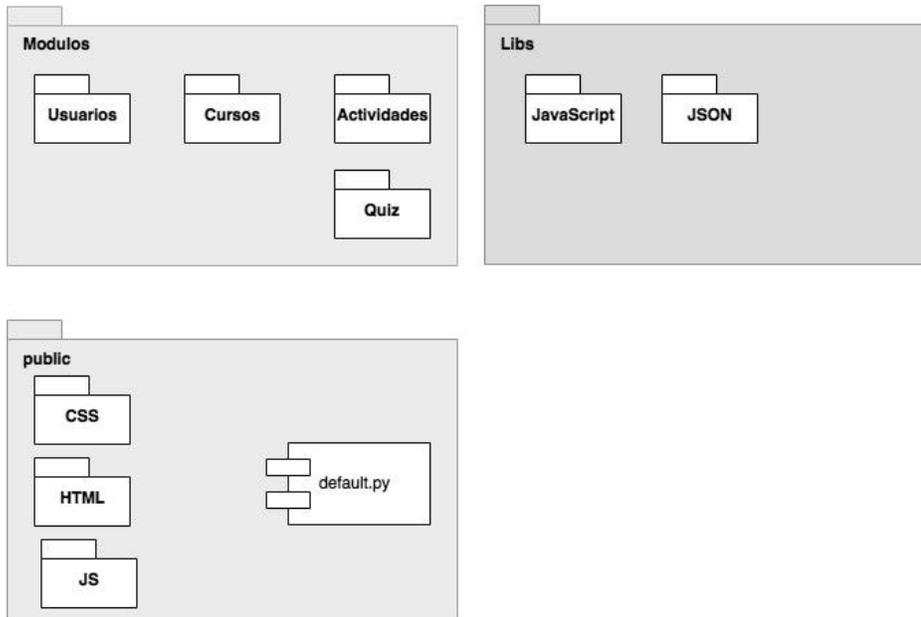


Fuente: elaboración propia

4.2.2. Diagrama de paquetes

La figura 9 muestra los diferentes paquetes que componen la aplicación; como se puede observar existente tres paquetes: el paquete módulos contiene los diferentes bloques individuales de la aplicación; el *public* es el paquete de acceso público donde se localiza todo lo que el usuario podrá percibir tal como vistas y estilos de páginas web, este paquete es controlado por el controlador por defecto de Web2py; el tercer paquete es el de las librerías donde se localizan las librerías principales para el módulo de *quiz* tales como JavaScript y JSO; este último es de suma importancia para controlar la estructura con la que se almacenarán los datos de un *quiz* en la base de datos.

Figura 9. **Diagrama de paquetes**



Fuente: elaboración propia.

4.3. Vista de proceso

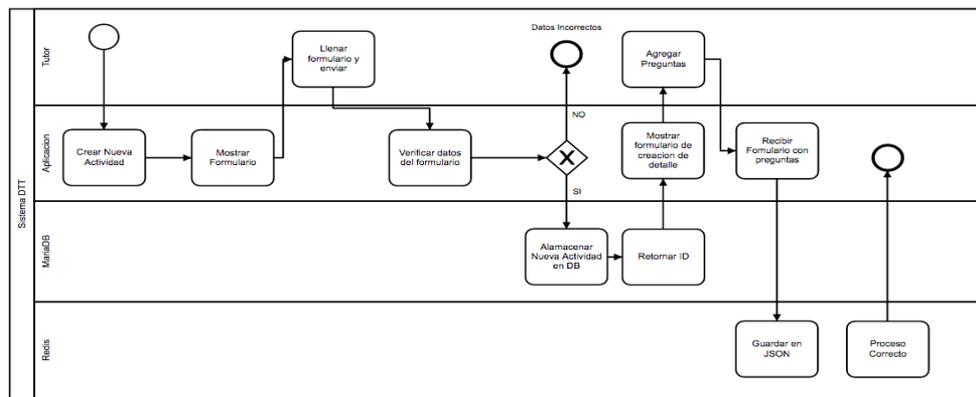
A continuación, se describen los diagramas que corresponden a la vista de proceso de la arquitectura software de este proyecto.

4.3.1. Diagrama de actividad

La figura 10 muestra el diagrama del proceso de creación de *quiz*; dicho proceso es gestionado por un tutor académico y/o un catedrático y consiste en la acción de almacenar en el sistema un nuevo *quiz* el cual consta de una serie de preguntas; está relacionado a un curso y dará inicio en una fecha específica.

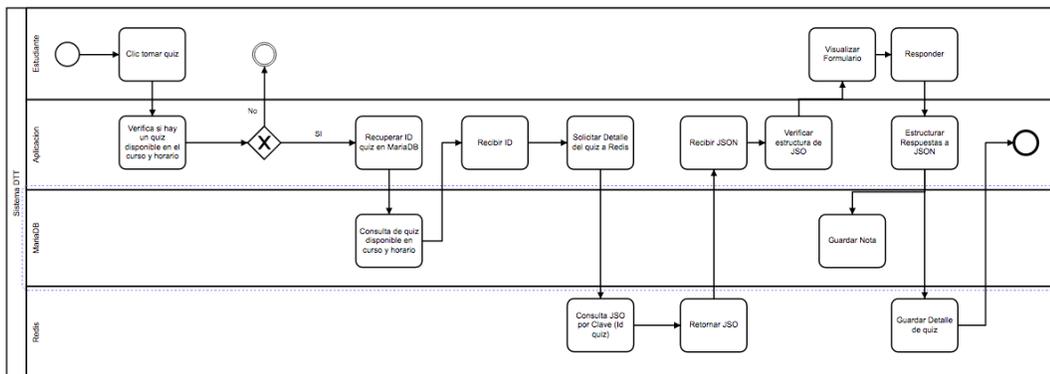
La figura 11 muestra el diagrama del proceso de tomar un *quiz* que consiste en la acción realizada por un estudiante para rendir una prueba en línea; dicha prueba debe estar registrada en el sistema y se habilitará en el horario y día estipulado tras su creación.

Figura 10. Diagrama de proceso de creación de *quiz*



Fuente: elaboración propia.

Figura 11. Diagrama de proceso de tomar un *quiz*

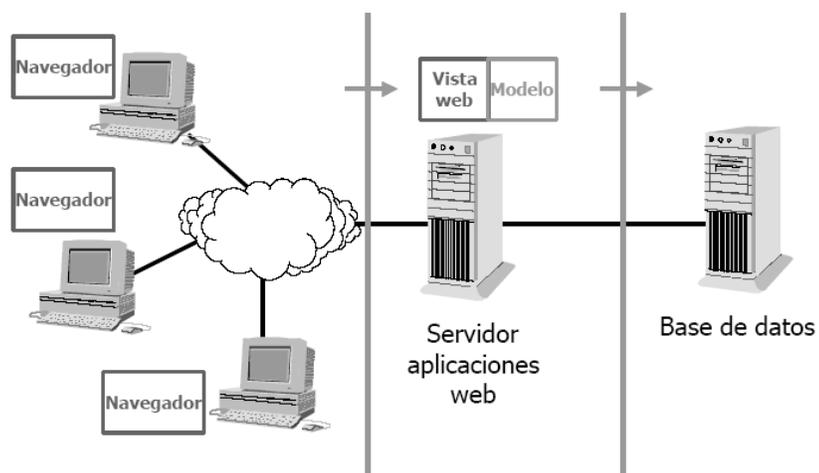


Fuente: elaboración propia.

4.4. Vista física

A continuación, se muestra el diagrama de la vista física correspondiente al sistema DTT.

Figura 12. Diagrama de vista física



Fuente: elaboración propia.

4.5. Vista de escenarios

El sistema presenta dos tipos de escenario para el módulo de exámenes *online*; el primero es el escenario en el que el usuario juega el rol de tutor académico o catedrático, este permite las siguientes funcionalidades: creaciones de exámenes, asignación de exámenes a una actividad, consulta de exámenes y creación de reportes.

Para el escenario en el que el usuario juega el rol de estudiante, el sistema le permite unirse a un examen que este activo en ese momento.

CONCLUSIONES

1. En el desarrollo de software existen diferentes tecnologías que pueden cumplir la misma función con paradigmas distintos para la resolución del problema; sin embargo, es responsabilidad del ingeniero en sistemas la evaluación meticulosa de cada aspecto que el proyecto demande para poder elegir las tecnologías adecuadas considerando que los recursos son limitados.
2. Los métodos de evaluación de conocimiento académico conllevan una responsabilidad de parte del catedrático para que la evaluación sea objetiva y la asignación de las ponderaciones correspondan con la dificultad de cada pregunta.
3. En las evaluaciones tipo cuestionario debe dosificarse el uso de las preguntas de falso y verdadero debido a que estas poseen un grado de incertidumbre del 50 % lo que podría llevar a un alumno a adivinar la mitad de la evaluación que provocaría que la evaluación no fuera objetiva.
4. La adopción de una cultura de evaluaciones en línea en una universidad de Guatemala conlleva la participación y apoyo de parte de diferentes áreas para garantizar que las evaluaciones son objetivas y que los estudiantes siguen los lineamientos establecidos por la institución sin prestarse a actos que provoquen una nota no objetiva.

RECOMENDACIONES

1. Los tutores académico y catedráticos debe evaluar la objetividad de sus evaluaciones antes de trasladarlas a sus estudiantes para evitar resultados que no sean consecuentes a la realidad.
2. Para alcanzar la mejora continua en la educación a nivel superior en la Facultad de Ingeniería; se debe ser todos partícipes de este proceso y apoyar los proyectos que involucran el uso de la tecnología ya que esta evoluciona día a día.

BIBLIOGRAFÍA

1. ARSHAVSKIY, Marina. *Instructional design for ELearning*. Estados Unidos: Springer, 2013. 248 p..
2. BOLAÑOS, Luna; DOUGLAS, Miguel, *Arquitectura de software*. Guatemala: UCML, 2001. 195 p..
3. *Documentación oficial web2py*. [En línea]. <<http://www.latinuxpress.com/books/drafts/web2py/>>. [Consulta: 25 de noviembre de 2017].
4. *Documentación oficial MariaDB*. [En línea]. <<https://mariadb.org/learn/>>. [Consulta: 25 de noviembre de 2017].
5. GORTON, Ian. *Essential software architecture*. Estados Unidos: Springer, 2006. Consulta: 20 de octubre de 2017.

