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

PLATAFORMA ACADÉMICA ONLINE PARA BÚSQUEDA, LOCALIZACIÓN Y ENLACE DE TUTORES EN DISTINTOS SECTORES EDUCATIVOS EN GUATEMALA

Esvin José Estrada Soc
Jeyson Steve Montenegro Alay
Asesorado por el Ing. Marlon Francisco Orellana López

Guatemala, julio de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PLATAFORMA ACADÉMICA ONLINE PARA BÚSQUEDA, LOCALIZACIÓN Y ENLACE DE TUTORES EN DISTINTOS SECTORES EDUCATIVOS EN GUATEMALA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

ESVIN JOSÉ ESTRADA SOC
JEYSON STEVE MONTENEGRO ALAY
ASESORADO POR EL ING. MARLON FRANCISCO ORELLANA LÓPEZ

AL CONFERÍRSELES EL TÍTULO DE

INGENIEROS EN CIENCIAS Y SISTEMAS

GUATEMALA, JULIODE 2017

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. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Ing. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
EXAMINADOR	Ing. William Samuel Guevara Orellana
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
SECRETARIA	Ing. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

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

PLATAFORMA ACADÉMICA ONLINE PARA BÚSQUEDA, LOCALIZACIÓN Y ENLACE DE TUTORES EN DISTINTOS SECTORES EDUCATIVOS EN GUATEMALA

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería Ciencias y Sistemas, con fecha 15 de noviembre de 2016.

Jeyson Steve Montenegro Alay

Esvin José Estrada Soc

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

Guatemala, 02 de Febrero de 2017

Señor
Ing. Carlos Azurdia
Facultad de Ingeniería
Universidad de San Carlos de Guatemala
Guatemala, Ciudad

Respetable Ing. Azurdia:

El motivo de la presente es para informarle que como asesor de los estudiantes Esvin José Estrada Soc y Jeyson Steve Montenegro Alay he procedido a revisar el trabajo de graduación titulado "PLATAFORMA ACADÉMICA ONLINE PARA BÚSQUEDA, LOCALIZACIÓN Y ENLACE DE TUTORES EN DISTINTOS SECTORES EDUCATIVOS EN GUATEMALA" y que de acuerdo a mi criterio el mismo se encuentra concluido.

He tenido comunicación periódica con los estudiantes y luego de haber revisado el trabajo, considero que cumple con los requisitos de calidad y profesionalismo que deben caracterizar a un futuro profesional de la informática.

Sin otro particular me suscribo de usted,

Atentamente,

Ing. Marlon Francisco Orellana López
Colegiado No. 8182

MARLON FRANCISCO ORELLANA LÓPEZ
INGENIERO EN CIENCIAS Y SISTEMAS
COL. 8182

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

Guatemala, 29 de Marzo del 2017

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 de los estudiantes **ESVÍN JOSÉ ESTRADA SOC** con carné **201212774 y CUI 2572 34705 0101,** y **JEYSON STEVE MONTENEGRO ALAY** con carné **201211472 y CUI 2288 07018 0101,** titulado: **"PLATAFORMA ACADÉMICA ONLINE PARA BUSQUEDA, LOCALIZACIÓN Y ENLACE DE TUTORES EN DISTINTOS SECTORES EDUCATIVOS, EN GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

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
INGENIERIA EN CIENCIAS Y SISTEMAS
COORDINACION TESIS

ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24188000 Ext. 1534

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, ***PLATAFORMA ACADÉMICA ONLINE PARA BÚSQUEDA, LOCALIZACIÓN Y ENLACE DE TUTORES EN DISTINTOS SECTORES EDUCATIVOS EN GUATEMALA*** *realizado por los estudiantes, ESVIN JOSÉ ESTRADA SOC y JEYSON STEVE MONTENEGRO ALAY, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

Ing. Marlon Antonio Pérez Türk
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 19 de julio de 2016

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

Ref.DTG.D.313.2017

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: **PLATAFORMA ACADÉMICA ONLINE PARA BÚSQUEDA, LOCALIZACIÓN Y ENLACE DE TUTORES EN DISTINTOS SECTORES EDUCATIVOS EN GUATEMALA,** presentado por los estudiantes universitarios: **Esvin José Estrada Soc y Jeyson Steve Montenegro Alay,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Pedro Antonio Aguilar Polanco
Decano

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

Guatemala, julio 2017

/cc

ACTO QUE DEDICO A:

Dios	Por ser una importante influencia en mi carrera
Mi madre	Jeanina Rosario Alay Ruiz de Montenegro, por ser mi guía, consejera y mi fuente de motivación para no darme por vencido, además de brindarme su apoyo y amor en todo momento.
Mi padre	Oscar Antonio Montenegro Franco, por sus consejos.
Mi tía	Marta Marina Alay Ruiz, por haberme brindado su apoyo y consejos a lo largo de mi vida.
Mis hermanos	Oscar Alexander y Jenniffer Julissa Montenegro Alay, por estar siempre presente en mis éxitos y en mis fracasos además de brindarme su apoyo en todo momento.

Jeyson Steve Montenegro Alay

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi *alma mater* y mi fuente de conocimiento.
Facultad de Ingeniería	Por brindarme la oportunidad de ser un profesional de excelencia.
Escuela de Ingeniería en Ciencias y Sistemas	Por brindarme los conocimientos y metodologías para ser aplicadas como un profesional.
Mi asesor	Ing. Marlon Orellana, por brindarme su tiempo asesorándome en la creación de este trabajo.
Mis compañeros de la Facultad	Con los compañeros que he convivido por haber compartido conocimiento y grandes experiencias.

Jeyson Steve Montenegro Alay

ACTO QUE DEDICO A:

Dios	Por sus bendiciones en mi carrera.
Mi madre	María Elena Soc Mas, mi maestra de la vida, por su apoyo incondicional y por ser el motivo de mi éxito.
Mi padre	Esvin Estrada, por su apoyo brindado durante esta etapa de la vida y por su amor de padre incondicional.
Mi tía	Thelma Estrada, por brindarme su ayuda y apoyo económico y psicológico durante mis estudios.
Mis hermanos	Kely, Sarai, Samuel y Eli, por ser la razón de mis logros.

Esvin José Estrada Soc

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi casa de estudios.
Facultad de Ingeniería	Por brindarme la oportunidad construir una carrera profesional de excelencia.
Escuela de Ingeniería en Ciencias y Sistemas	Por brindarme los conocimientos y metodologías para ser aplicadas como un profesional en la industria tecnológica.
Mi asesor	Ing. Marlon Orellana, por apoyarme con su tiempo y experiencia, en el desarrollo de este trabajo de graduación.
Mis compañeros de la Facultad	Por haber compartido conocimiento y faenas académicas.

Esvin José Estrada Soc

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
CRUD	Acrónimo en Inglés de las cuatro funciones básicas del almacenamiento persistente: *Create, Read, Update, Delete.*
REST	Transferencia de Estado Representacional es un estilo de arquitectura software.
UI	Es la interfaz de usuario.
DOM	*Document Object Model*Modelo de objetos del documento en español, es una interfaz o plataforma que permite a los programas acceder y actualizar contenido, la estructura y el estilo dinámicamente representación de documentos HTML.
API	Conjunto de servicios que exponen una funcionalidad.

GLOSARIO

Arquitectura Estructura lógica y fisca de los componentes de un sistema computacional.

Back-end Se refiere a la aplicación del lado del servidor.

Benchmarking Consiste en tomar como referencia a los mejores competidores y adaptar sus métodos, sus estrategias, dentro de la legalidad.

Callback Una devolución de llamada o retrollamada donde una función usa de argumento otra función.

Comando Instrucción que se da a una computadora.

Componente Es un elemento de un sistema de software que ofrece un conjunto de servicios, o funcionalidades, a través de interfaces definidas.

Debug Traducido es depurar, en el desarrollo es la ejecución y captura de variables y funciones de manera que se puede suspender el flujo del proceso.

DOM Modelo de Objeto de Documento, modelizan tanto la ventana del navegador como el historial, el documento o

página web, y todos los elementos que pueda tener dentro la propia página.

Framework Es un entorno de trabajo para desarrollo que integran herramientas que facilitan el desarrollo de aplicaciones.

Front-end Se refiere a la aplicación del lado del cliente.

Función Una función es un grupo de instrucciones con un objetivo en particular y que se ejecuta al ser llamada desde otra función o procedimiento.

IDE Ambiente de desarrollo integrado.

JSON Es un estándar abierto, que por sus siglas en inglés significa: JavaScript *Object Notation*, se utiliza para transmitir objetos de datos en formato texto legible con una convención atributo – dato.

JSONP Extensión de JSON para aplacar las dificultades de las restricciones de adaptabilidad entre exploradores.

Middleware Es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware y/o sistemas operativos.

Objeto Es una unidad dentro de un programa de computadora que consta de un estado y de un comportamiento, que a su vez constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución.

ORM	Es una técnica de programación para convertir datos entre el sistema de tipos de un lenguaje de programación orientado a objetos a una base de datos relacional.
Protocolo	Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.
Scope	Es el estado de un objeto en la aplicación.
Scroll	Componente del explorador web utilizado para recorrer una página web verticalmente.
Thumbnails	Imagen de tamaño reducido con forma circular.
Renderización	Es el proceso de carga y presentación visual de una página web.
RESTful	*Es la implementación de la arquitectura REST.*
SSL	Es un protocolo diseñado para permitir que las aplicaciones puedan transmitir información de manera segura, a través de internet.
SSH	Protocolo utilizado para acceder a máquinas remotas a través de una red.

RESUMEN

La plataforma de integración de tutores y estudiantes, permite enlazar y sustituir antiguas tareas de adquisición de servicios de educación, brindando un sitio público, intuitivo y de acceso inmediato. Provee un catálogo dinámico de tutores y de oportunidades a través de los estudiantes que son usuarios del sistema.

Se plantea la modelación del fenómeno educativo a través del modelo *task technology fit,* traducido al español es adaptación tecnológica de tareas. Busca relacionar el éxito de un proceso con las tareas y las herramientas tecnológicas empleadas para su ejecución. Al utilizar factores medibles, se presenta el impacto de la tecnología en el proceso que se observa. La plataforma es un sitio web accesible desde cualquier dispositivo con acceso a internet, que utilice tecnologías web. El sitio está basado en una arquitectura de servicios en sus componentes de más alto nivel. Contiene un *back-end* desarrollado en Koa.js y un f*ront-end* en AngularJS.

Koa.js es un *framework* basado en Node.js, que se utiliza para implementar aplicaciones del lado del servidor, que sirven y se integran con cualquier tecnología o *framework* del lado del cliente. Koa.js está desarrollado en Javascript; presenta una base sencilla para iniciar un sitio con múltiples funcionalidades y con dinamismo de utilización de bibliotecas.

AngularJS es un *framework* de aplicaciones web del lado del cliente que presenta facilidad en la construcción de sitios web; provee interacción robusta con los componentes del DOM; permite escribir sitios escalables con alta carga de información, brindando páginas dinámicas a través de su sistema de inyección de dependencias y sus *binding* de dos vías.

OBJETIVOS

General

Proveer una solución tecnológica para mejorar el acceso al aprendizaje de los estudiantes, a través de una plataforma pública en línea, en la cual se pueda establecer interacciones entre tutor y estudiante de todo tipo de ámbito académico o área técnica específicas.

Específicos

1. Facilitar el acceso a ubicación de tutores para un aprendizaje ajustado a las necesidades.

2. Innovar medios sociales para el área académica.

3. Crear medios de beneficio económico para tutores.

4. Centralizar información para uso público.

INTRODUCCIÓN

Se desarrolló la plataforma de enlace de tutores y estudiantes para proveer una solución alterna, eficiente de educación personalizada, que permite aprender o enseñar, cualquier tipo de habilidades o adquirir cualquier tipo de conocimiento. La solución se presenta con los requerimientos funcionales y no funcionales principales de cualquier sitio social. Cada usuario debe crear una cuenta que lo identifica dentro del sistema como tutor o estudiante. Las conexiones entre tutores y estudiantes se realizan a través del concepto de enlace planteado por la plataforma, cada usuario tiene una lista de contactos, cada usuario posee un listado de preferencias en categorías educativas. También existe la posibilidad de calificar las interacciones tutor-estudiante, esto para crear una reputación que se pueda utilizar como indicador de confianza o efectividad.

Se alcanza el éxito de construcción al utilizar tecnologías web, al emplear*frameworks* y herramientas de alta calidad, para brindar una solución creativa a través del software, para validar un modelo que permite evidenciar la efectividad de las tecnologías de la información en educación. Se utilizan certificados para una comunicación segura. Se presenta el ciclo de desarrollo, describiendo los componentes de las herramientas, su funcionalidad, la arquitectura, la estructura, el proceso de implementación y pruebas de la solución.

1. MARCO TEÓRICO

1.1. Identificación de la teoría que soporta la investigación

La investigación se debe basar en una teoría que permita medir, cuantificar e interpretar las necesidades del usuario para determinar, la correcta implementación y ejecución de la solución a través de la tecnología adecuada. Por sus siglas en inglés TTF (*Task technology fit*), que traducido al español es Ajuste tecnológico de tareas, esta teoría describe el impacto positivo que la tecnología tiene en el usuario individual al mejorar la capacidad y rendimiento en las tareas que efectúa. La descripción se realiza a través de un conjunto de factores medibles, que emplean preguntas simples para predecir la efectividad del sistema o tecnología en investigación.

La teoría TTF busca destacar la evidente relación entre el éxito de un sistema o una solución de IT, las tareas y tecnologías empleadas para su realización. Los requerimientos de los usuarios son la base de la construcción del sistema, donde estos permiten definir de manera adecuada la elección de las tecnologías de desarrollo, implementación y manteniendo del sistema. También enfatiza en la identificación temprana de los problemas durante el desarrollo de las tareas, la atención y documentación para una correcta mitigación en el futuro.

Validar la elección de una herramienta como beneficio para el usuario; además, cuantificar el impacto de la tecnología en los requerimientos funcionales y no funciones, son características principales de la teoría TTF.

La teoría TTF establece 8 factores, donde cada uno se mide utilizando entre dos y diez preguntas individualmente, para esto las respuestas se miden en una escala de

uno a siete, donde 1 se interpreta como: muy en desacuerdo y 7 como: muy de acuerdo. Los factores son los siguientes:

1. Calidad
2. Localización
3. Autorización
4. Compatibilidad
5. Facilidad de uso/capacitación
6. Puntualidad de la producción
7. Confiabilidad de los sistemas
8. Relación con los usuarios

La teoría se basa en dos fuentes de información

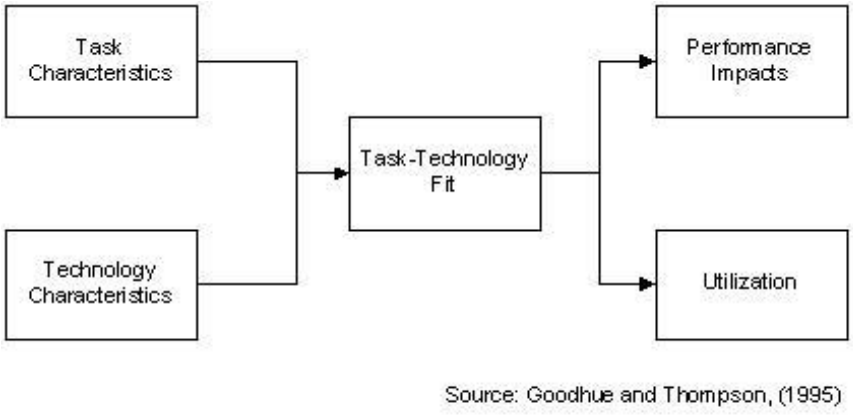
- Características de las tareas
- Características de las tecnologías

Se deben establecer relaciones entre las fuentes para generar una idea y determinar métricas que permiten establecer dos factores calificativos:

- Impacto del rendimiento
- Utilización

El usuario obtendrá beneficios directos, los cuales se interpretan como el impacto por la utilización de la tecnología, mientras que la utilización es el registro de cuántos usuarios utilizan la tecnología, qué tareas realizan con la tecnología y cuál es la finalidad de su utilización. La interpretación conjunta de estos factores resultantes formula la suposición del beneficio del sistema y la ventaja sobre alternativas, si estas existieran.

Figura 1. **Modelo teórico planteado por Goodhue y Thompson**



Fuente: http://istheory.byu.edu/wiki/File:Tf.JPG. Consulta: 21 de marzo de 2016.

El esquema anterior describe la necesidad de conocer las características de las tareas y tecnologías, que se quieren desarrollar y aplicar, para entonces utilizar la teoría, y determinar así una predicción del impacto y la utilización que el sistema en cuestión, obtendrá en el usuario final, esto con la finalidad de producir sistemas a la medida.

Según la teoría TTF, es el comportamiento de emplear tecnología para realizar tareas, esta se compone de factores cuantificables como: la frecuencia de utilización, la cantidad de tecnologías que realicen o soporten las tareas del usuario. El impacto modela el enlace entre la teoría del ajuste y las consecuencias de utilización de un sistema. Esto significa la mejora por el cambio del flujo de trabajo, adopción de la tecnología y las ventajas competitivas adquiridas por el usuario.

1.2. *Relación de la teoría con la tecnología a utilizar*

El proyecto de la plataforma educativa es un sistema de enlace educativo de los tutores y estudiantes en diversos segmentos de aprendizaje, donde un estudiante que necesita desarrollar habilidades y obtener conocimiento se comunica con un tutor que posee los conocimientos y capacidades requeridas. No existe actualmente en el país una plataforma de enlace virtual, esto se interpreta como una limitante en el desarrollo del país, ya que según el evidente desarrollo de países latinoamericanos, la educación permite el crecimiento intelectual de un individuo, y esto permite oportunidades para el futuro. Los usuarios son entidades importantes para el funcionamiento del sistema, ya que este requiere interacción social. Es importante: conocer, medir los requerimientos funcionales y no funcionales para implementar una solución que mejore y simplifique la tarea de enlazar una actividad educativa. Cualquier plataforma social es un producto basado en la utilización del mismo, depende de cuántos usuarios posee para alcanzar su objetivo.

La definición correcta de la tarea de comunicar tutores con estudiantes permitirá identificar barreras sociales, tecnológicas y culturales, que deberán ser solucionadas para satisfacer las necesidades de los tutores y los estudiantes. Las características de las tecnologías disponibles para la media poblacional de Guatemala, son recursos determinantes en el desarrollo de la solución. La propuesta de un sitio web como respuesta al acceso limitado a teléfonos inteligentes, por parte de los tutores, quienes en su mayoría son un segmento de personas mayores de edad, y tecnologías responsivas para satisfacer las necesidades de los estudiantes que por mayoría son jóvenes, además de facilidad de acceso a un sitio, en relación a la descarga de un aplicativo móvil o de escritorio, la mayoría de dispositivos poseen buscador web. Se consideran los puntos siguientes como indicadores de la plataforma:

- Calidad: El sistema obtendrá información de los usuarios a través de un proceso de *rating de la experiencia*, donde un calificativo permitirá evaluar la calidad del mismo, esto de la mano de un sistema de entrega y mejora continua que asegurar la entrega de un producto mejorado en períodos cortos de tiempo

- Localización: Los usuarios proveerán información personal sobre su ubicación, esto producirá un patrón de comportamiento que permitirá dar soluciones personalizadas para un segmento demográfico.

- Autorización: La información personal del usuario será concedida bajo riesgo propio, esto podría interpretarse como una barrera cultural, sin embargo, es un proceso de adaptación.

- Compatibilidad: La solución es una plataforma web, por lo tanto todos los dispositivos con acceso a internet podrán acceder sin ninguna diferencia o limitación.

- Facilidad de utilización/capacitación: El programa de experiencia permite la retroalimentación del usuario, la correcta definición de los procesos, basados en los casos de uso, permitirán que las tareas y el flujo dentro del sistema sean bastante sencillas.

- Puntualidad de la producción: Mediante la implementación de metodologías ágiles de desarrollo, se podrá alcanzar y cumplir los objetivos planteados en el programa de entrega continua.

- Confiabilidad de los sistemas: La información de los usuarios es pública, por lo tanto la confiabilidad se basa en la cultura de la utilización de herramientas sociales, además de proveer la seguridad de la efectividad del sistema, que es proveer estudiantes a los tutores y viceversa.

- Relación con el usuario: Mediante el registro y suscripción a futuro, los usuarios estarán siempre en comunicación constante con el equipo que provee la solución, mediante los programas antes descritos. Todo esto para asegurar la fidelización del cliente.

2. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN PROPUESTA A TRAVÉSDEL APLICATIVO

En la actualidad la educación y capacitación individual es fundamental para el desarrollo de un mercado competitivo, al aumentar la calidad educativa se incrementa la calidad de vida de los empleados y sus familias. Se ha observado que el recurso humano es una de las prioridades actuales de las organizaciones estatales y privadas en Guatemala.

Actualmente en Guatemala la institución responsable de la capacitación es INTECAP (Instituto Técnico de Capacitación y Productividad) que realiza esfuerzo para fomentar distintos temas técnicos, que impulsan el desarrollo en los guatemaltecos. Aun con los esfuerzos de instituciones como INTECAP e instituciones privadas, los guatemaltecos poseen gran dificultad de acceso a estos programas debido a problemas como: los horarios dado que los guatemaltecos poseen jornadas de trabajo extensas y los estudiantes tienden a tener traslapes con su rutina diaria de estudios, los cursos son impartidos en distintos períodos en el año, por lo cual no están siempre disponible, en ocasiones los costos de recibir los cursos o temas específicos son altos para el alcance de los guatemaltecos, entre otra variedad de contratiempos que impiden poder capacitarse en los temas específicos que deseen.

Dada la importancia de recibir capacitación se desarrollan alternativas de enseñanza como lo son las tutorías. Este medio de aprendizaje surge al observar los distintos problemas que conlleva aprender sobre temas de interés específico. Además, que desarrollados por personas con conocimientos sólidos en el tema y puedan brindar soporte a quien tenga la necesidad de aprender con un horario flexible.

Un tutor representa una persona con conocimientos previamente adquiridos y demuestra gran capacidad en el área que se desempeña; es por ello que la idea principal es: proporcionar un espacio de mutuo donde tutores y personas interesadas en estudiar temas específicos se puedan conocer y entablar negociaciones de horario, coste y lugar de la enseñanza.

La seguridad en nuestro país es un mal el cual incrementa la desconfianza en nuestros habitantes; por tal razón, la aplicación que deberá proporcionar un medio para identificar tanto a estudiantes como a tutores, que demuestren ser personas serias y no representen ningún riesgo para la integridad de ambas partes en la interacción educativa.

2.1. Mercado Objetivo

La aplicación TutoSite está dirigida a cualquier persona que desee aprender o enseñar, a nivel académico será enfocado a todos los niveles y tipos de enseñanzas tanto teórica como práctica.

Los dos tipos de usuario en específico que utilizarán la plataforma se representan en los siguientes perfiles:

1. Estudiante: Cualquier estudiante, trabajador, técnico o persona interesada en capacitarse en temas específicas de distintas áreas de conocimiento, que por medio de un costo pueda adquirir tiempo de un persona con experiencia o especializada en el área que el estudiante está buscando.

2. Tutor: La real academia española lo define como “Profesor privado que se encargaba de la educación general de los hijos de una familia”, este rol podrá ser aplicado por una persona que posea el estudio adecuado y tenga la habilidad de transmitir fácilmente el conocimiento.

Luego de definir los perfiles de los involucrados, se debe mencionar el contexto donde la experiencia de aprendizaje se desarrollará, por tal razón los participantes deben establecer el lugar y la hora en que se desarrollará la tutoría.

Los recursos necesarios para poder utilizar la plataforma serán una conexión a internet y un medio tecnológico para acceder a la página web, entre los que actualmente se utilizan en el país son: computadora de escritorio o portátil, teléfono inteligente, tableta y otros que dispongan de un navegador web.

2.2. *Benchmark* de la aplicación

Hasta hoy existen aplicaciones con gran similitud a la plataforma, por lo cual se debe mencionar antecedentes nacionales e internacionales para ser utilizadas como referencias importantes el desarrollo de la aplicación.

2.2.1. Mercado Nacional

En el contexto nacional no se encontró ningún sitio que se adapte a todas las características de nuestra aplicación, pero se ha localizado algunas referencias que se encuentran dentro de la industria que pueden compartir ideas relativamente básicas.

Las referencias que se pueden mencionar son las de tipo privado, páginas web que promueven las tutorías por medio de sus propios empleados que brindan las clases.

Algunas de las referencias que se pueden mencionar son:

- Ontutorias: su objetivo es brindar “tutorías a domicilio y su segmento de mercado es desde preescolar hasta nivel universitario”[1].

Figura 2. **Sitio Ontutorias**

Fuente: Ontutorias https://ontutorias.com. Consulta: 3 de abril de 2016.

- Tutorías Guatemala: brindan tutorías por profesionales y se enfocan en “áreas académicas para pruebas de admisión de colegios y universidades”[2].

Figura 3. **Sitio Tutorías Guatemala**

Fuente: Tutorías Guatemala. https://tutoriasguatemala.com. Consulta: 3 de abril de 2016.

[1] http://www.ontutorias.com. Consulta: mayo 2016
[2]http://www.tutoriasguatemala.com. Consulta mayo 2016.

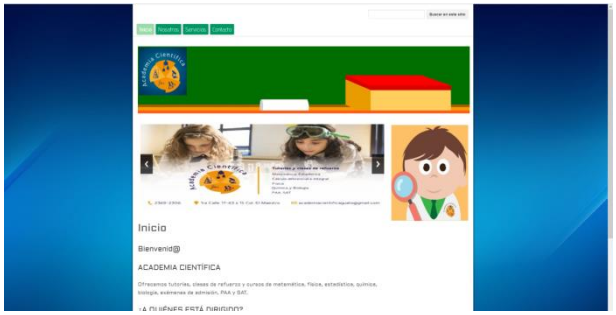
- Clases y Tutores: brindan “tutorías a domicilio y su segmento de mercado es desde preescolar hasta nivel universitario, cuentan con planes para realizar las tutorías”[3].

Figura 4. **Sitio Clases y Tutores**

Fuente: Sitio Clases y Tutores. http://clasesytutores.com. Consulta: 3 de abril de 2016.

- Academia Científica: “profesionales que brindan refuerzos en áreas básicas dirigido a alumnos de nivel escolar hasta universitarios”[4].

Figura 5. **Sitio Academia Científica**



Fuente: Sitio Academia Científica https://www.academiacientificagt.com. Consulta: 3 de abril de 2016.

[3]http://clasesytutores.com/. Consulta: mayo 2016
[4]http://www.academiacientificagt.com. Consulta: mayo 2016

- Tutor Doctor: se enfoca en “educación complementaria, con servicios personalizados y a domicilio ya sea para estudiantes o adultos, y a precios accesibles”[5].

Figura 6. **Sitio Tutor Doctor**

Fuente: Sitio Tutor Doctor https://tutordoctor.gt. Consulta: 3 de abril de 2016.

Este conjunto de ejemplos nacionales únicamente ofrecen servicios propios de tutorías, no fomentan la interacción directa entre cualquier tutor y estudiante, por lo cual se considera que sus servicios son únicamente para brindar sus propios tutores y se limitan a las áreas de conocimiento básico; y quienes lo brindan, son escasos.

[5]http://tutordoctor.gt/ Consulta: mayo 2016

2.2.2. Mercado Internacional

En el contexto internacional encontramos que existen sitios ampliamente desarrollados en el tema de la búsqueda de tutorías en línea por lo que son de referencia precisa a nuestra aplicación.

Algunas páginas que proporcionan la búsqueda de tutores son:

- University Tutor: es una página que proporciona la búsqueda por material, lugar y proporciona las tutorías en línea o presenciales, “cuenta con área de geolocalización donde se busca al tutor más cercano, la disponibilidad del tutor, y las votaciones que reciben los tutores; brinda registro tanto para estudiantes como para tutores, este sitio es utilizado en Estados Unidos”[6].

Figura 7. **Sitio University Tutor**

Fuente: Clases y Tutores de Guatemala. https://www.universitytutor.com
Consulta: 11 de mayo de 2016.

[6]http://www.universitytutor.com Consulta: mayo 2016

- FirstTutors: es una página que su objetivo de “búsqueda primordial es la ubicación y posee la división por tema, posee un filtro de búsqueda por nivel que desee el estudiante que sepa el tutor, es una página de origen inglés que posee un alcance a nivel europeo”[7].

Figura 8. **Sitio WebFirstTutors**

Fuente: FirstTutors. http://www.firstutors.com. Consulta: 3 de abril de 2016.

[7] http://www.firstutors.com/uk/art Consulta: mayo 2016

3. DISEÑO DE LA APLICACIÓN

3.1. Arquitectura del Sistema

3.1.1. API

Las siglas API provienen de la definición en inglés *Application Programming Interface*, que en español se interpreta como Interfaz de Programación de Aplicaciones. Un API es un conjunto de comandos, funciones, protocolos y objetos que los desarrolladores pueden usar para crear software e interactuar con sistemas externos. “Esto provee que el código pueda ser reutilizado y mejorar el rendimiento en operaciones comunes”[8].

3.1.2. REST

Las siglas REST provienen de la definición en inglés *Representational State Transfer*, que en español se interpreta como Transferencia de Estado Representacional. REST se define como un estilo de arquitectura basada en un conjunto de principios que describen como los recursos son definidos y direccionados.

“REST ignora los detalles de implementación de los componentes, para centrarse en las funciones y en la interacción con otros componentes”[9].

[8] CHRISTENSSON, P. *API Definition.* http://techterms.com. Consulta: octubre de 2016.
[9] FIELDING, ROY.*Representational State Transfer (REST).* http://www.ics.uci.edu. Consulta: octubre 2016.

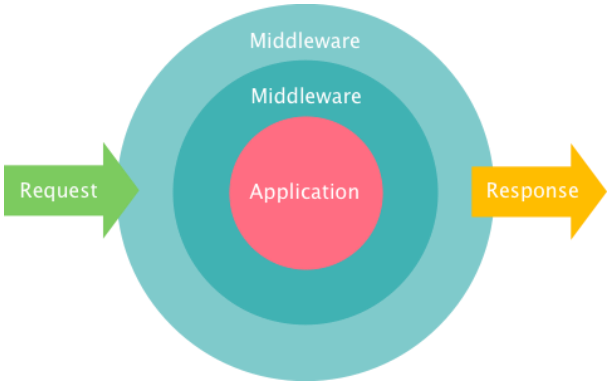
3.1.3. RESTful API

RESTful es una implementación de la arquitectura REST, siguiendo estrictamente las reglas que define REST. Por lo cual una API RESTful se define como una interfaz basada en los estándares de la arquitectura REST, que utiliza el protocolo de comunicación HTTP para transferir la información, permitiendo a los clientes despreocuparse por conocer cómo acceder a los recursos, sin mantener estados en las peticiones que se realicen.

3.1.4. Middleware

Son componentes de software que se encuentran en una capa intermedia, donde funcionan como tuberías hacia la aplicación para manejar solicitudes y respuestas. “Cada componente elige transmitir la información al siguiente componente en la tubería y puede realizar ciertas acciones antes y después del siguiente componente”[10].

Figura 9. **Modelo representación Middleware**



Fuente: Clacklisp. http://clacklisp.org
Consulta: 13 de octubre de 2016.

[10]SMITH, Steve; ANDERSON, Rick. Middleware. https://docs.asp.net/ Consulta: octubre de 2016

3.1.5. Patrón de Arquitectura MVC

Es un patrón de arquitectura de software, también es visto como un patrón de diseño de software. Ambas definiciones son válidas dependiendo el enfoque de la aplicación. Este divide cierta solución de software separadas interconectadas entre sí, las separaciones son representaciones del flujo de la información hasta el usuario.

En la solución de la plataforma de enlace académico se implementó como patrón de diseño para *Front-end* y arquitectura de software para el *Back-end*.

3.1.6. Arquitectura MVW

AngularJS promueve la utilización de la arquitectura *Model - View -Whatever* que traducido es Modelo - Vista - Cualquier Cosa. La vista es la representación en el DOM de los componentes. El modelo es un singleton definido por un servicio, este encapsula los datos, provee un API accesible y portable. Cualquier cosa es la interacción de los servicios y demás componentes que proveen información al $scope, esto decorado por lo que llamamos controlador de AngularJS, entonces, el controlador de AngularJS interactúa con todos; unificando, delegando o transmitiendo simplemente los datos y lógica.

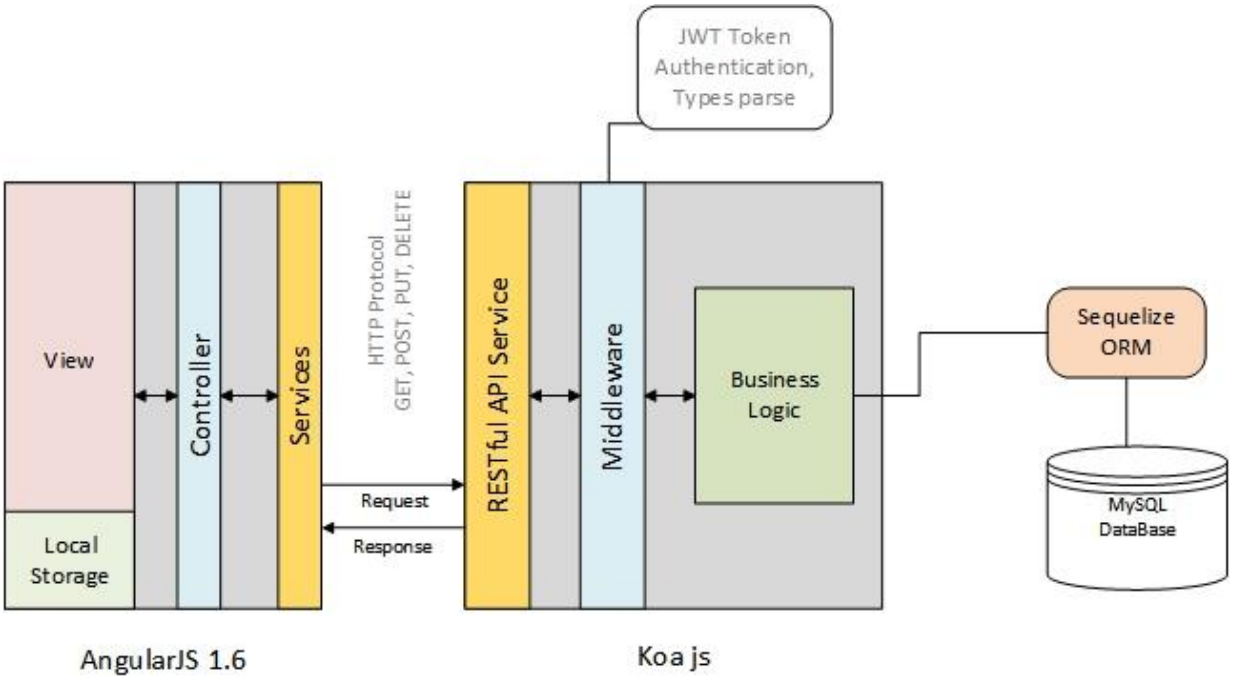
Esta interacción se puede percibir como si se trabajara con una arquitectura MVC, pero la utilización de la convención propuesta por el sistema de inyección de dependencias, delegan muchas funcionalidades propias de un controlador a otros componentes como: las directivas, *providers* (*factory, values*), filtros entre otros, que alejan al controlador de AngularJS del concepto de Controlador.

3.1.7. ORM

ORM proviene del significado en inglés *Object-relational mappers*, que significa mapeo de objeto-relacional. “Es una librería que automatiza la transferencia de información almacenada en una base de datos relacional, hacia objetos que son comúnmente utilizados dentro de la aplicación”[11]. El uso primordial de un ORM es proveer una atracción de alto nivel, en el cual permite a un desarrollador manipular acciones con la información tales como: inserciones, eliminaciones, modificaciones y búsquedas, sin necesidad directa de utilizar el lenguaje SQL o procedimientos almacenados.

3.1.8. Diagrama de la solución

Figura 10. **Arquitectura de la solución**



Fuente: elaboración propia.

[11] MAKAI, Matt. ORM https://fullstackpython.com. Consulta: noviembre 2016

3.1.8.1. Descripción de la solución

La solución se plantea con base a las tecnologías de última generación, elegidas para que la aplicación sea robusta y ágil. A continuación se describirá los componentes de la arquitectura.

AngularJS 1.6 es utilizado para representar la parte visual o la vista de la aplicación la cual tendrá una interacción directa con el usuario, AngularJS utilizará una vista y un controlador por cada página web que se dispone, además de utilizar *"Local Storage"* como almacenamiento de información del lado del cliente o navegador web que se utilice. Cada controlador de AngularJS utilizará un *service* que brindará una comunicación con protocolos HTTP hacia los servicios RESTful.

La comunicación se realizará con base en el protocolo HTTP que posee los siguientes métodos de petición.

Tabla I. **Protocolos HTTP**

Protocolo	Funcionalidad
GET	Este método es utilizado para obtener información en base a peticiones con formato JSON hacia el servicio RESTful.
POST	Método utilizado para realizar actualizaciones o modificaciones de datos realizando la petición con formato JSON hacia el servicio RESTful.
PUT	Método utilizado para realizar inserciones de datos realizando la petición con formato JSON hacia el servicio RESTful.
DELETE	Método utilizado para eliminar datos realizando la petición con formato JSON hacia el servicio RESTful.

Fuente: elaboración propia.

Koa.js es un *framework* basado en JavaScript, se implementó debido a su potencial de procesamiento, también permite la incorporación de módulos externos con NPM, Koa.js genera las APIs *RESTful,* las cuales se exponen para su consumo en AngularJS. Koa.js además de funcionar como un motor de servicios, permite el cumplimiento de las reglas del negocio e integra validaciones externas de los datos que se reciben y se envían.

Koa.js permite la interacción con middleware de terceros que permiten integrar seguridad, validaciones, interacciones y otras aplicaciones más que se pueden añadir. El uso de Autenticación por *Token* permite que se realice una comunicación segura con el navegador web y por ende todos los consumos de las *APIs* se realizarán de una manera confiable y evitará que terceros puedan acceder sin autorización a los recursos de los servicios RESTful.

Sequelize es un ORM potente que permite mapear en objetos la base de datos MySQL. Con Sequelize se obtiene una rápida y segura conexión a los datos además de una fácil utilización y permite crear una capa de modelo que pueda utilizar Koa.js para obtener los datos fácilmente.

3.2. Requerimientos no funcionales

3.2.1. Seguridad

3.2.1.1. Autorización por *token* JWT

Autorización por token JWT es una estrategia la cual es utilizada para la validación en todos los servicios que sean consumidos desde el lado del cliente, será necesario un *token* de seguridad, el cual servirá como llave encriptado en base a un algoritmo basado en una llave privada. La gran importancia de esto radica en la seguridad con la cual el servidor deberá responder únicamente a los clientes que

posean su *token* de seguridad, que será conseguido luego de autenticarse con su usuario y contraseña al sistema.

3.2.1.2. Validación de tipos

Es una técnica de seguridad para validar los tipos correctos de los parámetros que son enviados hacia los servicios RESTful, se utilizó la API koa-validate que fue implementada para validar los tipos correctos de los datos y en caso el tipo de dato no es el esperado se devuelve un mensaje de error para evitar que se procese la petición con tipos de datos incorrectos.

3.2.1.3. Certificado SSL

Un certificado SSL sirve para brindar seguridad al visitante de su página web, una manera de decirles a sus clientes que el sitio es auténtico, real y confiable para ingresar datos personales. “Las siglas SSL responden a los términos en inglés *Secure Socket Layer*, el cual es un protocolo de seguridad que hace que sus datos viajen de manera íntegra y segura, es decir, la transmisión de los datos entre un servidor y usuario web, y en retroalimentación, es totalmente cifrada o encriptada”[12].

3.2.2. Mantenimiento

3.2.2.1. Diseño Modular

Se denomina diseño modular a un sistema que está divido en partes diferenciadas donde cada módulo realiza una tarea en específica y se realiza de esta

[12] CertSuperior. Certificado SSL https://certsuperior.com. Consulta: diciembre 2016

manera para observar claridad, reutilizar los módulos y reducir los costos además de facilitar el mantenimiento y la detección de errores.

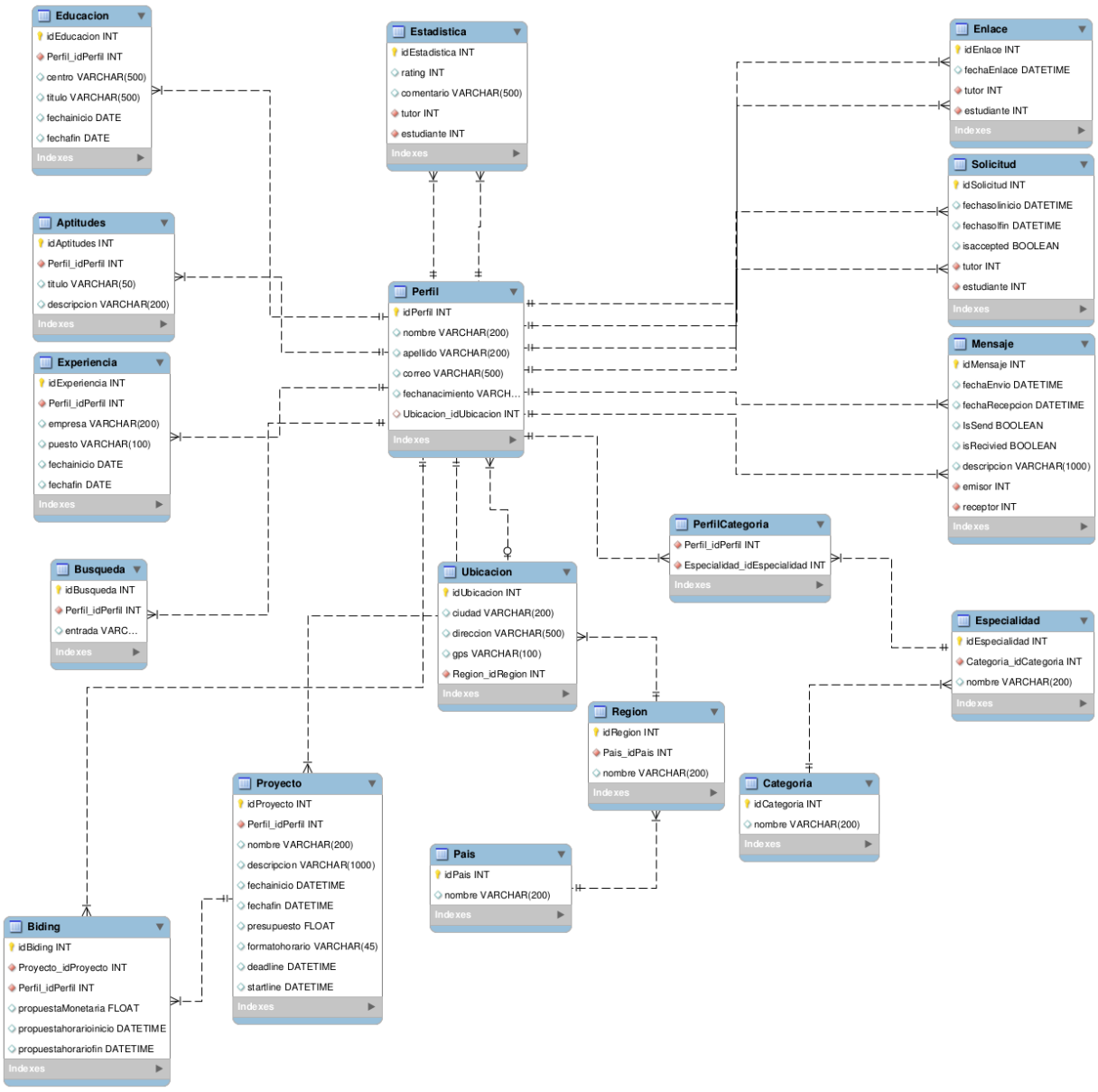
3.2.2.2. Soporte de Componentes

La arquitectura utilizada y el *framework*Koa.js ofrecen gran facilidad y una extensa variedad de componentes gratuitos, que se pueden encontrar los repositorios públicos lo cual hace queKoa.js posea amplia variedad de componentes.

3.3. Diagrama de Datos de la Aplicación

3.3.1. Diagrama entidad-relación

Figura 11. **Diagrama Entidad-Relación**



Fuente: elaboración propia. Utilizando la herramienta MySQL Workbench

3.3.2. Descripción diagrama

El diagrama está enfocado para almacenar de forma adecuada la información siguiendo los estándares de la tercera forma normal para el diseño de diagramas entidad relación.

Se determinó el almacenamiento de la información general del usuario donde se incluye su educación, su experiencia y su lugar de ubicación. Otra información que se almacena son los mensajes, los enlaces con otros usuarios y las solicitudes para obtener tutorías entre los usuarios, también se almacena el costo por hora de los tutores para que los que desean adquirir la tutoría puedan ver el precio de la misma.

3.4. Prototipo

El prototipo de la plataforma se utiliza para comunicar y definir la idea principal del sitio, es una representación limitada del producto, además de clarificar los requisitos de usuario y definir alternativas posibles.

3.4.1. Inicio

El sitio contiene dos versiones de la pantalla de inicio donde ambos tutor y estudiante respectivamente, podrán acceder a su información. Al estudiante la pantalla de la figura 4, le permite ingresar al historial de enlaces también acceso al buscador de tutores.

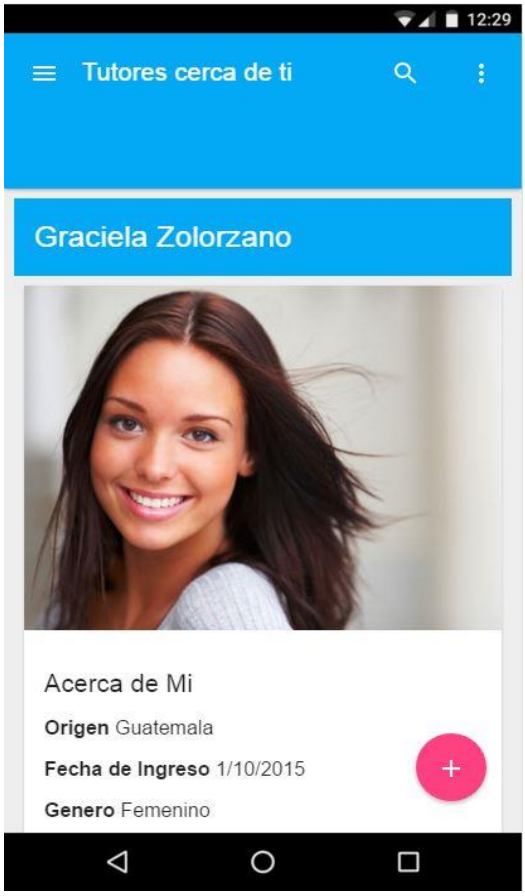
Figura 12. **Pantalla de inicio**

Fuente: elaboración propia.

3.4.2. Perfil

El perfil muestra información relacionada al usuario, información de localización, educación, logros, conocimiento, preferencias, costo de contratación entre otros. Los datos son presentados de forma que se pueda conocer al usuario; así el estudiante elija al tutor o bien el tutor al estudiante.

Figura 13. **Pantalla de perfil**



Fuente: elaboración propia.

3.4.3. Enlace

Registra el enlace entre dos usuarios; tutor y estudiante, mostrando información del contrario, contiene botones que permiten ingresar al perfil o enviar un mensaje, además de contener *thumbnails* de las fotos de perfil de los usuarios.

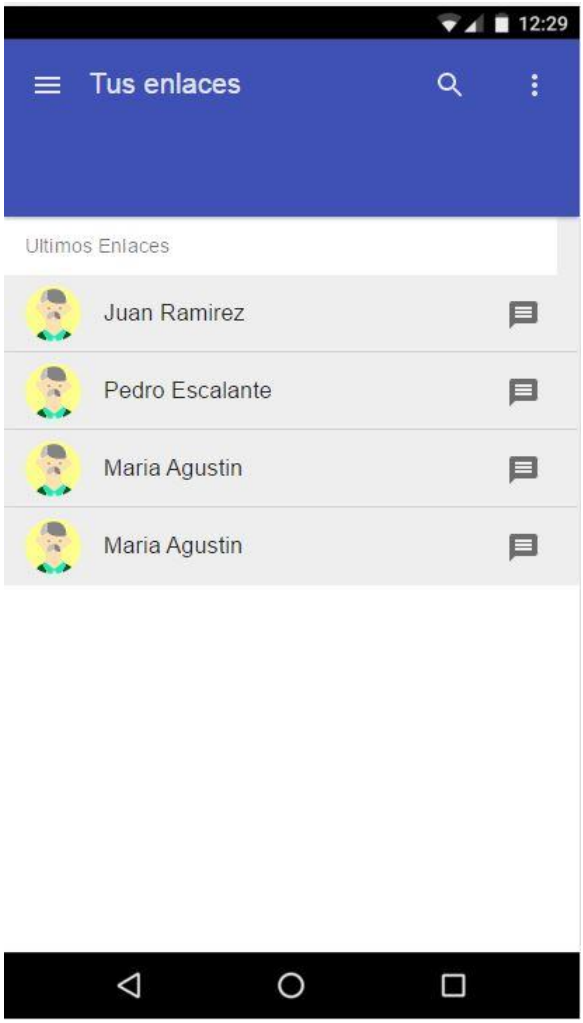
Figura 14. **Pantalla de enlace**

Fuente: elaboración propia.

3.4.4. Tus enlaces

Lista los últimos enlaces del usuario, a partir de fecha más reciente, donde cada ítem muestra un tutor o estudiante, con un botón de acceso rápido para enviar un mensaje.

Figura 15. **Pantalla de enlaces**



Fuente: elaboración propia.

3.4.5. Tutores en el área

Para los estudiantes en esta pantalla se muestran los tutores cercanos según la residencia registrada por los mismos y la posición obtenida por el buscador del dispositivo. Permitiendo clasificarle por categoría de tutor.

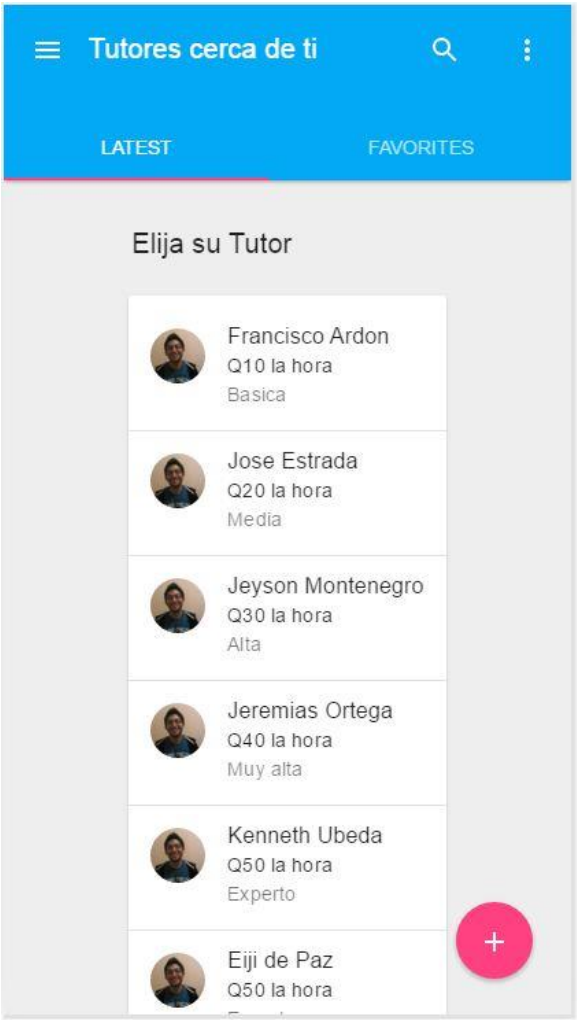
Figura 16. **Pantalla de tutores en el área**

Fuente: elaboración propia.

3.4.6. Tutores

Muestra los tutores según un parámetro de búsqueda seleccionado, donde el listado muestra el nombre del tutor, el precio del servicio que presta y el nivel académico que este posee.

Figura 17. **Pantalla de tutores**



Fuente: elaboración propia.

3.4.7. Categorías

Muestra información de las categorías existentes para encontrar un tutor, información importante relacionada a la categoría y un resumen de algunos de los temas que alcanzan.

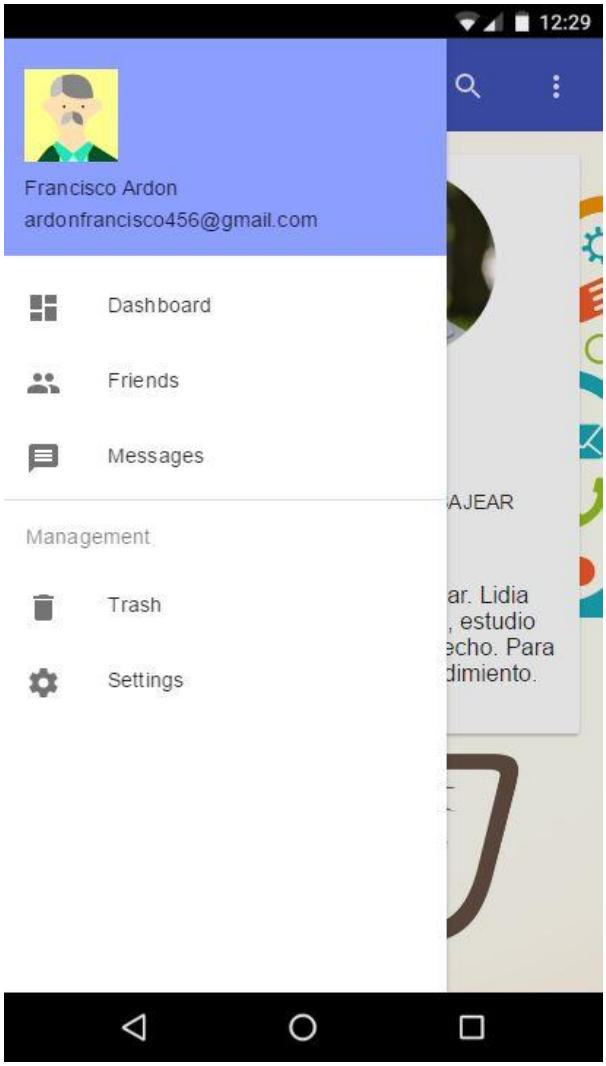
Figura 18. **Pantalla de categorías**

Fuente: elaboración propia.

3.4.8. Navegador

En cada pantalla es posible acceder a la información registrada en el perfil del usuario, según el tipo de usuario que utilice la plataforma, se muestran los enlaces hacia distintas funcionalidades relacionadas con las necesidades del usuario.

Figura 19. **Navegación**



Fuente: elaboración propia.

3.5. Ergonomía del diseño

3.5.1. Material Design

Es un conjunto de prácticas (metáfora) de diseño enfocado a la visualización limpia de los componentes mostrados en pantalla, que utiliza animaciones y transiciones de respuesta. Se basa en manejar los objetos como en la realidad, donde los materiales poseen características físicas de los objetos y la interacción en una pantalla no rompe las leyes de la física. También indica la utilización de la luz, las superficies y el movimiento para formar efectos realistas, los que el usuario final esta adecuado en ver. Las características de *Material* son:

- Sólido
- Ocupa puntos únicos en el espacio
- Impenetrable
- Cambia tamaño respecto al espacio
- Se puede crear y destruir
- Se mueve sobre cualquier eje.

3.5.2. Angular Material

Angular en un *framework* de JavaScript para desarrollo de *Front-end*, su principal características es la implementación de directivas en etiquetas HTML. Angular Material es un conjunto de directivas, servicios y etiquetas orientados a la Guía de diseño material (*Material Design*), que permite construir una página web de forma que esta tenga diseño material, utilizando componentes y planos de diseño con enfoque material. Angular Material provee todas las herramientas necesarias para implementar el diseño material. Estructura básica de una pantalla en Angular Material:

Figura 20. **Aplicación básica**

Fuente: Angular Material-Start (Tutorials). https://github.com/angular/material-start/tree/es5-tutorial. Consulta: 20 de abril de 2016.

4. Documentación y tutorial de la aplicación

4.1. Requisitos mínimos

Se listan y especifican los requisitos mínimos para desarrollar e implementar esta aplicación, herramientas y estructura del ambiente de desarrollo del sitio.

4.2. Herramientas

Las herramientas utilizadas son programas y servicios para implementar un sitio web básico, además de herramientas de desarrollo.

Las herramientas utilizadas se listan a continuación:

- Atom IDE
- DigitalOcean Droplet Manager
- Filezilla Client (Cliente SSH)
- Chrome Developer Tools

4.3. Hardware

Se recomienda la utilización de una computadora Inter core I3, con al menos 4Gb de memoria RAM, 1 GB de espacio disponible en disco duro. También es necesario para publicar el sitio en Internet, poseer una máquina en la Nube de cualquier proveedor; se recomienda el proveedor Digital Ocean.

4.4. Software

4.4.1. JavaScript

Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, originalmente era un lenguaje que se utilizaba del lado del cliente sobre un navegador web, pero al observar el potencial se ha utilizado del lado del servidor implementado en Node.js.

4.4.2. Koa.js

Koa.js es un nuevo *framework* que pretende ser una base más pequeña, más expresiva y más sólida para aplicaciones web y API. “A través de los generadores Koa.js permite renunciar a los *callbacks* y aumentar mucho el manejo de errores. Koa.js proporciona un elegante conjunto de métodos que hacen que los servidores de escritura sean rápidos y agradables”.[13]

4.4.3. AngularJS

Es un *framework* estructural de aplicaciones web del lado del cliente desarrollado por Google, es gratuito y de código abierto; permite el desarrollo de aplicaciones de lado del cliente, utiliza la propuesta de arquitectura MVVM. Angular lee código HTML tiene etiquetas especiales llamadas directivas, que proveen información de entrada y salida, hacia un modelo en Javascript. El *framework* adapta y extiende código HTML tradicional para presentar contenido dinámico, que genera sincronización entre modelos y vistas. Existen actualmente 2 versiones de Angular.

[13] Koajs. Koajs definition. https://koajs.com. Consulta: diciembre 2016

Angular 1 es la versión más antigua que contiene una filosofía de Controladores y *Scope.* Utiliza inyección de dependencias, simula los servicios tradicionales del lado de servidor, como vistas dependientes de los controladores, esto permite eliminar la necesidad de escribir código repetitivo.
Angular 2 es la versión más reciente, una reconstrucción total de Angular que cambia la filosofía anterior a componentes y directivas, donde un componente es una directiva unida a una plantilla.

La plataforma es desarrollada en Angular 1, utiliza las propiedades de *framework* para minimizar el contenido estático, permite su dinamismo y provee al buscador una sintaxis nueva de etiquetas HTML. Todo esto para soportar necesidades del software como validaciones, intercambio de información, controles, manejo de elementos y agrupación de componentes del aplicativo. Abstrayendo la vista en elementos del DOM como estructuras dinámicas y totalmente accesibles, se aprovecha el API del DOM.

AngularJs permite separar lo que el usuario visualiza y toma en cuenta todas las capacidades que el cliente de un sistema pueda tener, permite delegar procesos al cliente con la finalidad de mejorar la eficiencia del flujo de trabajo del software.

Técnicamente Angular simplifica el proceso de desarrollo de aplicaciones, ya que permite utilizar el lenguaje JavaScript y aprovechar sus propiedades asíncronas a través de la correcta implementación de patrones de diseño, que permiten monitorear y modificar todo el flujo de datos de la solución. Elimina la necesidad de registrar *callbacks,* también restringe por eficiencia la manipulación del DOM en lugares donde no es correcto, el despliegue de la información proveniente de operaciones CRUD de manera transparente para el usuario según la carga obviamente, elimina la necesidad de código de inicialización. El desarrollo es casi directo a la lógica del negocio.

Características

- Data binding
- Expresiones
- Interpolación
- Directivas
- Vistas y rutas
- Filtros
- Compilador HTML
- Forms

Estructura de la aplicación

- App wiring - Inyección de dependencias
- Exposición de modelos a plantillas
- Bootsrap – Arranque de la aplicación
- Comunicación con el servidor.

Las directivas más comunes son:

- ng-app
- ng-bind
- ng-model
- ng-model-options
- ng-class
- ng-controller
- ng-repeat
- ng-if
- ng-animate

4.4.4. AngularMaterial

Es un *framework* de desarrollo por componentes UI para aplicaciones web que implementa las especificaciones del diseño material de Google, provee un conjunto de componentes como etiquetas web y directivas reutilizables y editables para la presentación de la aplicación. Utiliza todas las especificaciones de AngularJS con el agregado de los componentes materiales de servicios y directivas nuevas. Las directivas más comunes de Angular Material son:

- md-autocomplete
- md-autofocus
- md-button
- md-card
- md-checkbox
- md-chip
- md-dialog
- md-divider
- md-fab-actions

Angular Material provee servicios que permiten el acceso de eventos y el manejo de los mismos durante la ejecución del proceso dinámico de la aplicación web. Los principales servicios utilizados dentro del desarrollo de la plataforma son:

- $mdDialog
- $mdIcon
- $mdIconProvider
- $mdBottomSheet
- $mdSideNav
- $mdToast

4.4.5. HTML5

Es la actualización del lenguaje de representación de contenido web, que también es un estándar. En su quinta versión posee muchas características soportadas en exploradores modernos, el HTML utilizado en el proyecto es de esta versión. Entre las características más importantes se pueden puntualizar: las etiquetas nuevas, la estandarización de soporte multi-exploradores, un cantidad considerable de API´s que simplifican el desarrollo. HTML5 provee soporte para distintos tipos de contenido, además de eliminar la rigurosa dependencia de software de terceros para presentar contenido multimedia. Es importante señalar que no es necesario instalar, documentar o adaptar el código desarrollado, porque la interpretación de las etiquetas y sus propiedades se realizan en el explorador.

4.4.6. MySQL

Es un sistema de administración de bases de datos relacionales RDBMS, por sus siglas en inglés, de código abierto desarrollado por Oracle, actualmente en su versión 5.6.12. MySql se prefiere en aplicaciones web: por su fácil integración, además es compatible con gran variedad de frameworks y accesible mediante varios lenguajes de programación. En el sistema operativo Ubuntu está instalado por defecto, su instalación puede realizarse mediante un gestor de paquetes del sistema operativo o por el instalador que se puede obtener de la página oficial.

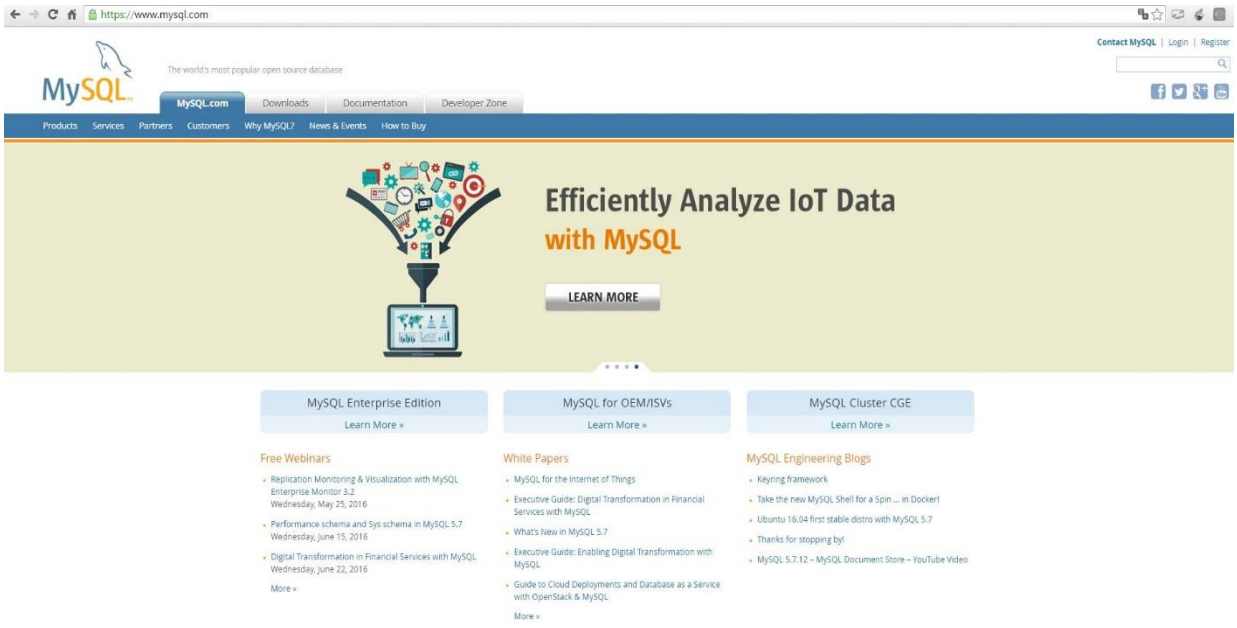
Instalación mediante Ap-get en Ubuntu:

```
$ sudo apt-get install mysql-server
$ sudo mysq_secure_installation
$ sudo mysql_install_db
```

Acceso a MySQL vía consola:

```
$ mysql –u root -p
```

Figura 21. **Sitio descarga MySQL**

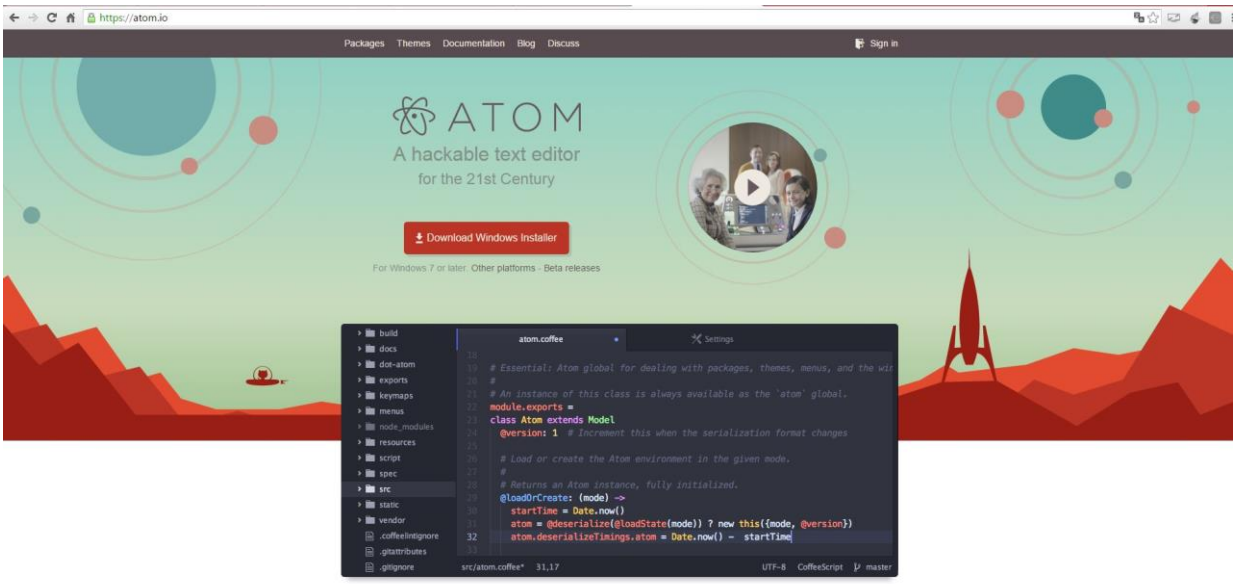


Fuente: elaboración propia. https://mysql.com. Consulta: 20 de abril de 2016.

4.4.7. Atom

Es un editor de texto gratis de código abierto compatible con GNU/Linux, con soporte para *plugins* escrito en Node.js, desarrollado por GitHub, utilizado por un IDE esta optimizado para desarrollo en lenguaje JavaScript. Puede ser obtenido en su sitio oficial.

Figura 22. **Sitio de descarga Atom.**



Fuente: elaboración propia. https://atom.io. Consulta: 20 de abril de 2016.

4.4.8. Node.js

Es un ambiente multiplataforma de tiempo de ejecución para desarrollo de aplicaciones web de lado del servidor, utiliza una arquitectura dirigida por eventos capaz de realizar procesos asíncronos. Es distribuido de manera gratuita y es de código abierto. Puede ser instalado mediante un gestor de paquetes del sistema operativo o en su página oficial.

4.4.9. NPM

Es un gestor de paquetes para Node.js (desarrollo del lado del servidor) que permite la instalación de todo tipo de paquetes de forma pública y por aplicación web. Su instalación está incluida en la instalación de Node.js.

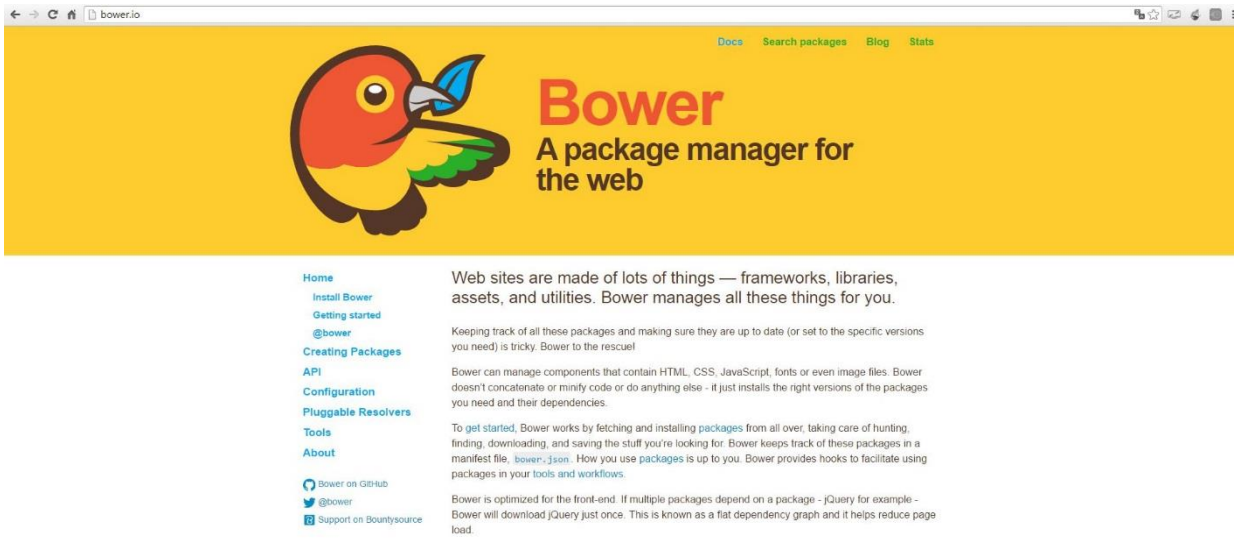
4.4.10. Bower

Es un sistema de administración de paquetes para desarrollo del lado del cliente. Permite administrar las versiones de paquetes de la aplicación web. Es gratuito y es distribuido como software libre. Puede obtenerse a través de *npm* o bien utilizando el instalador en su sitio oficial.

Instalación mediante npm en Ubuntu:

```
$ npm install –g bower
```

Figura 23. **Sitio descarga Bower**



Fuente: elaboración propia. https://bower.io. Consulta: 20 de abril de 2016.

4.5. Documentación y tutorial del sitio

4.5.1. Configuraciones BackEnd

4.5.1.1. Instalación de Servidor Koa.js

Para la instalación de Koa.js necesitamos instalar npm que es un gestor de paquetes con el cual obtendremos Koa.js. Luego instalar npm instalamos Koa.js con el comando *npm install koa.*

Figura 24. **Instalación Koa.js**

Fuente: elaboración propia.

4.5.1.2. Generación de Servicios REST

Se genera un API en la cual el servidor las publicará y se tendrán acceso únicamente con un *token* de seguridad, el cual será proveído cuando el usuario se autentique en el sistema.

En la siguiente imagen, se observa un API llamada categoría, la cual muestra las categorías actuales de los tutores; además, valida los tipos que ingresan desde el consumo del API REST.

Figura 25. **Servicios RESTFUL**

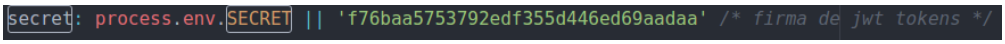
```
route.post('/api/categoria', function*(next) {
    //middleware validation
    this.checkBody('nombre').notEmpty().len(1,199);
    if (this.errors) {
        this.body = this.errors;
        return;
    }
    yield next;
}, insertCategoria);


route.get('/api/categoria', getAllCategory);
route.get('/api/categoria/:id/id', getCategoryById);
route.get('/api/categoria/:nombre/nombre', getCategoryByName);
```

Fuente: elaboración propia.

4.5.1.3. Integración Seguridad por Token a Servicio REST

Para generar y obtener la validación por token, se necesita una firma o una llave privada con la cual se genera automáticamente tokens de autorización para el sistema.

Figura 26. **Firma de token jwt**



```
secret: process.env.SECRET || 'f76baa5753792edf355d446ed69aadaa' /* firma de jwt tokens */
```

Fuente: elaboración propia.

A continuación se necesita incluir el API *koa-jwt* con el cual se genera el token y por último agregar la llave privada al API y agregarlo a la instancia del servicio donde estarán corriendo los servicios REST.

Figura 27. **Adición autenticación al servicio**

Fuente: elaboración propia.

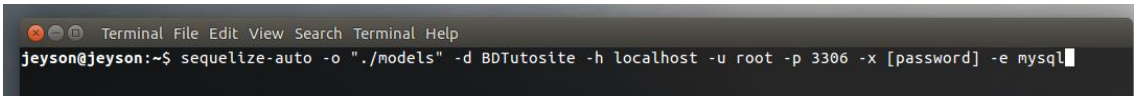
4.5.1.4. Generación de modelos de la base de datos de ORM

Se debe instalar *sequelize-auto* para poder generar los objetos desde la base de datos MySQL, se necesita instalar por medio del siguiente comando:

```
npm install -g sequelize-auto
```

A continuación para poder generar los objetos del modelo se debe ingresar el siguiente comando en consola con los siguientes parámetros: la ubicación de los objetos, el nombre de la base de datos, el nombre del usuario, la contraseña y el puerto del servicio de la base de datos para poder realizar la conexión y generación automática.

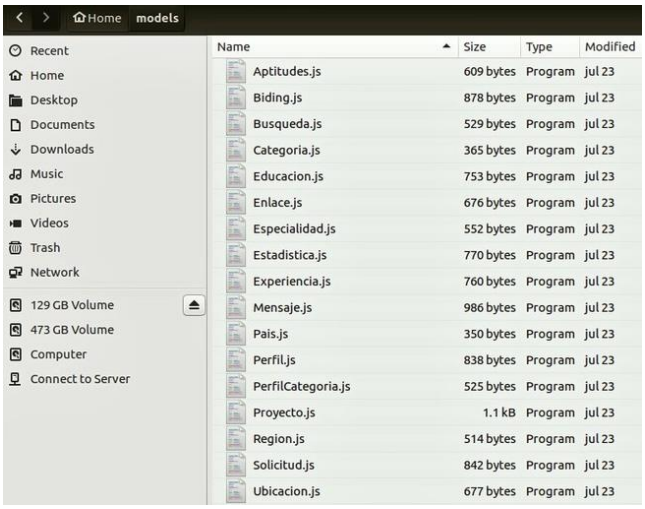
Figura 28. **Generación de modelos desde la base de datos**



Fuente: elaboración propia.

Con lo cual se obtiene los objetos de la base de datos

Figura 29. **Objetos generados de la base de datos**



Fuente: elaboración propia.

4.5.1.5. Utilización de modelos para mapeo en Base de Datos

Se debe agregar una configuración al ORM sequilize, con el cual se especifica la carpeta los modelos y la conexión a la base de datos para poder consumir la información, esta configuración se realiza en el archivo inicial del servidor de Koa.js.

Figura 30. **Conexión de Koa.js a la base de datos MySQL**

```
mysql: {
    modelPath: join(__dirname, 'fixtures/models/'),
    db: process.env.MYSQL_DB || 'DBTutoSite',
    username: process.env.MYSQL_USER || 'root',
    password: process.env.MYSQL_PASS || 'root',
    dialect: 'mysql',
    host: '127.0.0.1',
    port: 3306,
    pool: {
        maxConnections: 10,
        minConnections: 0,
        maxIdleTime: 30000
    },
    logging: false
},
```

Fuente: elaboración propia.

Luego de la configuración, se puede realizar las llamadas a los modelos para la utilización de las sentencias, con las cuales se obtendrán los datos de la base de datos MySQL.

Figura 31. **Utilización de modelos en Koa.js**

```
function* getAllCategory() {
    var foos = yield orm.db.sql.select()
        .from("Categoria")
        .query();
    this.body = foos;
}
function* getCategoryByName(nombre) {
    var foos = yield orm.db.sql.select()
        .from("Categoria")
        .where("nombre = ?", id)
        .query();
    this.body = foos;
}
function* getCategoryById(id) {
    var foos = yield orm.db.sql.select()
        .from("Categoria")
        .where("id = ?", id)
        .query();
    this.body = foos;
}

function* insertCategoria() {
    var categoria = this.request.body;
    var query = yield orm.db.sql
        .insert()
        .into('Categoria')
        .setFields(categoria).query();
    this.body = '{ response: success }';
}
```

Fuente: elaboración propia.

4.5.2. Configuraciones FrontEnd

- Instalación AngularJS

Angular puede ser instalado a través del administrador paquetes Bower o bien descargando las fuentes desde el sitio oficial.

Instalando via Bower:

$ bower install angular

Entonces se debe agregar la etiqueta en el documento HTML.

<script src="/bower_components/angular/angular.js"></script>

Figura 32. **Sitio descarga Angular**

Fuente: elaboración propia. https://angular.org
Consulta: 20 de abril de 2016.

- Instalación Angular Material

  Angular Material puede ser instalado a través de Bower, o bien descargando las fuentes en su sitio oficial.

Instalación deAngular Material desde el administrador de paquetes Bower:

$ bower install angular-material --save

Se debe configurar las dependencias en el documento HTML.

En el encabezado del documento:

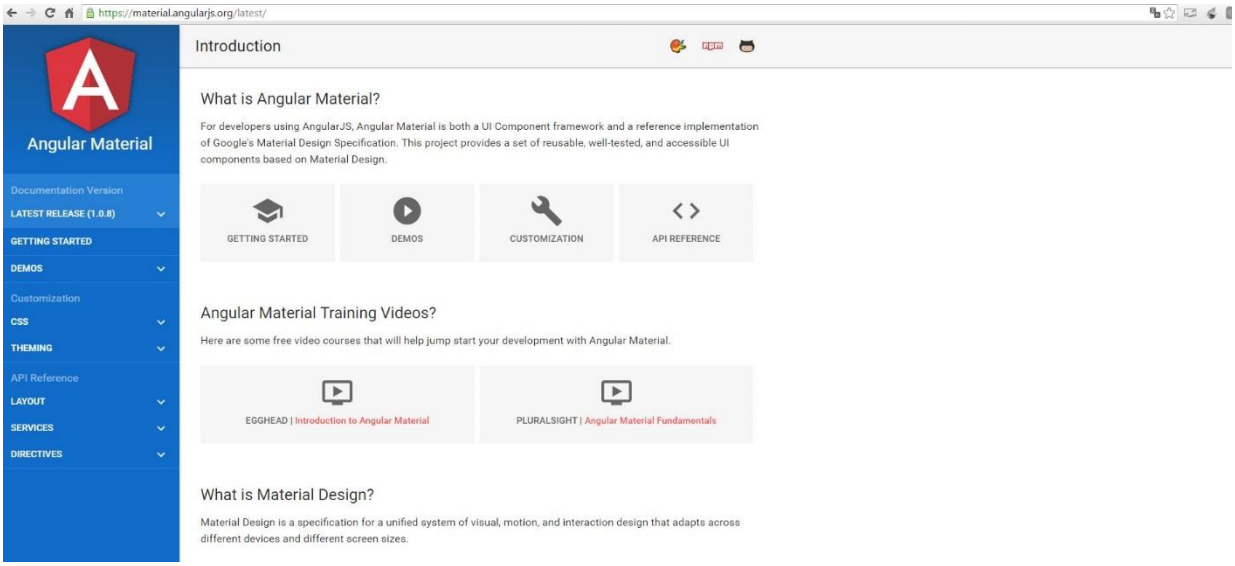
<link rel="stylesheet" href="/bower_components/angular-material/angular-material.css">

En el pie del documento:

<script src="/bower_components/angular-aria/angular-aria.js"></script>

<script src="/bower_components/angular-animate/angular-animate.js"></script>
<script src="/bower_components/angular-material/angular-material.js"></script>

Figura 33. **Sitio descarga Angular Material**



Fuente: elaboración propia. https://material.angularjs.org/latest
Consulta: 20 de abril de 2016.

- Instalación de dependencias

  - ngInfiniteScroll

    Es una directiva para AngularJS, utilizada para evaluar cuando el usuario se aproxima hacia abajo con el *scroll* del explorador, esto para implementar un evento que al ejecutarse permite generar contenido mientras el usuario baja en la página.

- ng-img-scrop

  Es una directiva para AngularJS, utilizada para cortar imágenes en forma circular y cuadrada, también para exportar la imagen resultante, utilizando el componente Canvas de HTML5.

- angular-animate

  Es un servicio que provee funcionalidad de animación para varias directivas nativas del framework y directivas propias, permite emitir animaciones como: transiciones, animaciones de fotogramas, entre otros, durante el ciclo de vida de las directivas.

- angular-messages

  Es una directiva que permite mostrar y ocultar mensajes, utiliza el estado o el valor de la etiqueta a la cual está asignado como una propiedad, diseñada con estados de lenguaje de estilo. Se complementa usualmente con el servicio $error de AngularJS para reportar errores registrados en formularios.

- angular-route

  Es un servicio que permite relacionar y mapear los controladores y las vistas, definiendo una ruta para las mismas. Se implementa en el proyecto con la directiva ngView y el servicio de manejo de parámetros $routeParams.

- Desarrollo de Ruteo

El ruteo o *routing* es la estructuración de la navegación en una aplicación web, también puede ser la estructuración de los estados de una aplicación web. Esto permite registrar y configurar la interacción del usuario con el sistema y validar el acceso, el flujo de acceso, el flujo de la información, las dependencias de funcionalidad, entre otros. El servicio $route de angular se configura a través de estados de la navegación de la plataforma. Por ejemplo: Tutores, Enlaces, Inicio, Registro, son estados del aplicativo y tienen una ruta definida.

- Desarrollo de Módulos

AngularJS permite a los arquitectos de software la facilidad de crear la estructura arquitectónica adecuada a sus necesidades, por lo tanto se utilizará una arquitectura modular, donde un componente modulo es la definición de la lógica, ruteo, plantillas y módulos hijos. Donde los módulos mapean la estructura los directorios en forma de árbol. Los módulos de alto nivel son: Raíz, componentes y común. El módulo de raíz define la configuración y renderización de la aplicación web. El módulo de componentes, registra los componentes de la aplicación, cada uno con sus controladores respectivos. El modulo común registra los servicios, directivas y filtros reutilizables en los componentes. Los módulos de bajo nivel son los componentes individuales de la aplicación web.

- Desarrollo de Controladores

El controlador como su nombre lo indica es la lógica del negocio detrás de la vista, está relacionada al DOM, utilizando la directiva *ng-controller* o el servicio de ruteo.

El controlador debe manejar siempre la lógica del negocio. La dependencia de información y transformación de datos, pueden encapsularse como un servicio, utilizándolos en los controladores a través de la inyección de dependencias, además debe existir un único controlador por vista o componente.

- Desarrollo de directivas

Una directiva es un modelo que se liga a los elementos del DOM, tales como: Etiquetas, Atributos, y clases CSS. Estos marcadores se comunican con el compilador HTML de AngularJS, para crear un comportamiento especifico o modificar algún elemento del DOM. Se desarrollan las siguientes directivas: directiva de lectura y edición de imágenes, directiva de gestión de contraseñas, directiva de gestión de usuarios del sistema.

- Desarrollo de servicios

Un servicio permite agregar funcionalidad y organizar el código en la aplicación. Se implementaron servicios *Singletons*, que son transversales a la aplicación, donde cada componente obtiene una referencia de él mismo. Al utilizar el servicio $provide, se emplea la función Factory, que recibe una propiedad $get para registrarlos; una vez registrados son agregados a los

controladores, directivas o filtros mediante el Sistema de Inyección de dependencias de AngularJS. Se registraron servicios públicos de obtención de data y servicios propios de cada módulo del aplicativo.

- Desarrollo de Servicios REST

Un servicio puede contener desde: gestión de información, modificación y presentación de los datos; pero de forma global se implementaron para consumir servicios REST, siguiendo la arquitectura de servicios. AngularJS provee 2 posibilidades para interactuar con los servicios: La primera es el Factory $resource, que contiene toda funcionalidad para realizar un CRUD pero con métodos de alto nivel. La segunda opción es el servicio $http, utilizado en la aplicación. Funciona de manera que se comunica vía JSONP, este expone un API basado en el concepto de promesas, que retorna éxito o error a la configuración de la conexión y tipo de petición provista al servidor.

- Diseño de páginas Web

La metáfora Diseño Material de Google, se implementó para el diseño de las Páginas Web el framework de *front-end* Angular Material. Este provee componentes como etiquetas HTML, servicios y recursos que siguen las guías de Diseño Material. Los objetos visibles se manejan como objetos reales que poseen propiedades de cualquier material físico; Estos tienen definidos un estilo así como reglas de movimiento y transición. Se implementaron utilizando recursos provistos por el Google como: La suite de iconos, patrones, temas, *layouts* entre otras. El proyecto cuenta con una página de Ingreso aislada y una página master que muestra todas rutas existentes, siguiendo la metáfora de las aplicaciones de una única página (single page application), con tres componentes básicos: barra de estado, gaveta de navegación y contenido. En

el componente de contenido se muestran las diferentes rutas y estados de la aplicación.

- Pruebas unitarias y de fin a fin

Las pruebas unitarias se encargan de probar las partes individuales del código, su fin es el desarrollo concreto de pequeñas porciones de código, que ofrecen funcionalidades básicas o complejas para el sistema. Se trabaja bajo una metodología de desarrollo basada en comportamiento, por lo tanto para asegurar la estabilidad de la aplicación, las pruebas unitarias se realizan con Jasmine y Karma.

Las pruebas de fin a fin, es la percepción del usuario del flujo completo de un proceso específico que realiza la aplicación, ya que toma en cuenta: el funcionamiento, rendimiento y el cumplimiento de los requisitos de la aplicación, para esto se utilizó Jasmine y Protractor.

Jasmine es un framework de pruebas para JavaScript, que no necesita un explorador, ni DOM y es independiente de frameworks de desarrollo.

Karma es un framework de evaluación de ambientes de prueba, que proporciona facilidad de configuración y automatización de las mismas, también, permite realizar un análisis de los resultados de las pruebas entre exploradores.

Protractor es un framework de pruebas de fin a fin para sitios en AngularJS, que simula la interacción del usuario y la ejecución en el explorador, permitiendo recolectar información del comportamiento real de la aplicación.

- Preparación para producción

El ambiente de producción, es donde el producto final es utilizado por el usuario objetivo, esto quiere decir que el producto es funcional, eficiente y estable, entre otras características. Después del desarrollo es necesaria la gestión y preparación del lanzamiento a producción. Existen diferentes prácticas y guías con diferentes enfoques para publicar un sitio. Las actividades básicas y necesarias tomadas en cuenta para la plataforma son: Deshabilitar mensajes de debug, deshabilitar comentarios de desarrollo. Se realiza la segmentación de inclusión de librerías o bibliotecas externas al código de la aplicación, además, se concatena y minifica el código, esto se realiza con la herramienta de administración de tareas Grunt, al usar los plugins: Ng-annotate, Concat y UglifyJS.

Grunt es un administrador de tareas para JavaScript, realiza la automatización de tareas como compilación, minificación, entre otros.

NgAnnotate es un plugin que se utiliza como una tarea para gestionar y reconstruir las anotación de Inyección de dependencias.

Concat es un plugin que se utiliza como tarea para concatenar los archivos de extensión JavaScript del proyecto en un solo archivo.

UglifyJS es un plugin que se utiliza como tarea para minificar el código, concatenado esto permite la portabilidad de la aplicación, además reduce el tamaño de la aplicación para su traslado al cliente de manera rápida.

CONCLUSIONES

1. La plataforma facilita el acceso a información de contacto con tutores, favoreciendo a los estudiantes que desean obtener conocimiento sobre un área específica.

2. Se determinó que la plataforma es pionera en el ámbito social-académico, ya que en la actualidad en Guatemala no existe una plataforma similar que tenga el mismo objetivo.

3. Se observó que en la actualidad los guatemaltecos buscan ingresos extras, por tal razón la facilidad de publicar sus servicios de tutorías para recibir remuneración, es un beneficio.

4. El acceso a información que los tutores proporcionaron a la aplicación, facilitó la búsqueda y centralización de la información, de tal manera que el usuario puede tener al alcance esta información.

RECOMENDACIONES

1. Se insta a los estudiantes de la carrera de Ingeniería en Ciencias y Sistemas a que sigan una línea de investigación en tecnologías de última generación y continúen innovando con ideas para el beneficio social.

2. Apoyar las ideas locales de aplicaciones que tengan como fin ayudar y facilitar la vida diaria de los ciudadanos guatemaltecos, por lo cual se insta a las organizaciones estatales a brindar capital semilla para apoyar a los emprendedores nacionales.

3. Elegir tecnologías que se consideren estables, y también, definir arquitecturas con escalabilidad y alta disponibilidad para que sean fácilmente adaptables en los cambios subsiguientes.

4. Otorgar garantías a los usuarios que ofrecen sus datos personales, de tal forma que únicamente se utilicen para el fin que ellos destinaron y no ser compartidos por terceros para lucrar.

BIBLIOGRAFÍA

1. Academia científica de Guatemala. [en línea]. <http://www.academiacientificagt.com>[Consulta: mayo 2016].

2. CertSuperior. *Certificado* SSL. [en linea]. <https://www.certsuperior.com/>. [Consulta: diciembre 2016].

3. Clases y Tutores de Guatemala. [en línea]. <http://clasesytutores.com/>. [Consulta: mayo 2016].

4. CHRISTENSSON, P. *API Definition.* [en linea]. <http://techterms.com>. [Consulta: octubre 2016].

5. FIELDING, ROY. *Representational State Transfer (REST).* [en linea]. <http://www.ics.uci.edu>. [Consulta: octubre 2016].

6. First Tutors Reino Unido. [en línea]. <http://www.firstutors.com/uk/art>. [Consulta: mayo 2016].

7. Koa.js. Koajs definition. [en línea]. <https://www.koajs.com>. [Consulta: diciembre 2016].

8. MAKAI, M. *Object-relational mappers.* [en línea]. <https://www.fullstackpython.com/>.[Consulta: diciembre 2016].

9. OnTutorias. OnTutoria de Guatemala. [en línea]. <http://www.ontutorias.com>. [Consulta: mayo 2016].

10. SMITH, S.; ANDERSON, R. Middleware. [en línea]. <https://docs.asp.net/>. [Consulta: octubre 2016].

11. Tutordoctor Guatemala. [en línea]. <http://tutordoctor.gt/>. [Consulta: mayo 2016].

12. Tutorías Guatemala. Tutorías Guatemala. [en línea]. <http://www.tutoriasguatemala.com/>. [Consulta: mayo 2016].

13. University *Tutor* Estados Unidos de América. [en línea]. <http://www.universitytutor.com/>. [Consulta: mayo 2016].