



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

**APLICACIÓN COLABORATIVA PARA INCENTIVAR VENTAS Y COMPRAS
EN COMUNIDADES LOCALES “REMATALOGT”**

Luis Renato García López

Asesorado por el Ing. Rene Gustavo Catalan Gudiel

Guatemala, octubre de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**APLICACIÓN COLABORATIVA PARA INCENTIVAR VENTAS Y COMPRAS
EN COMUNIDADES LOCALES “REMATALOGT”**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

LUIS RENATO GARCÍA LÓPEZ

ASESORADO POR EL ING. RENE GUSTAVO CATALÁN GUDIEL

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

| | |
|------------|---------------------------------------|
| DECANA | Inga. Aurelia Anabela Cordova Estrada |
| VOCAL I | Ing. José Francisco Gómez Rivera |
| VOCAL II | Ing. Mario Renato Escobedo Martínez |
| VOCAL III | Ing. José Milton de León Bran |
| VOCAL IV | Br. Christian Moisés de León Bran |
| VOCAL V | Br. Kevin Armando Cruz Lorente |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

| | |
|------------|---------------------------------------|
| DECANA | Inga. Aurelia Anabela Cordova Estrada |
| EXAMINADOR | Ing. César Augusto Fernández |
| EXAMINADOR | Ing. Herman Igor Véliz |
| EXAMINADOR | Ing. Ricardo Alfredo Morales |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez |

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:

APLICACIÓN COLABORATIVA PARA INCENTIVAR VENTAS Y COMPRAS EN COMUNIDADES LOCALES “REMATALOGT”

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 7 de febrero de 2019.

Luis Renato Garcia Lopez

Guatemala 19 de Noviembre de 2019

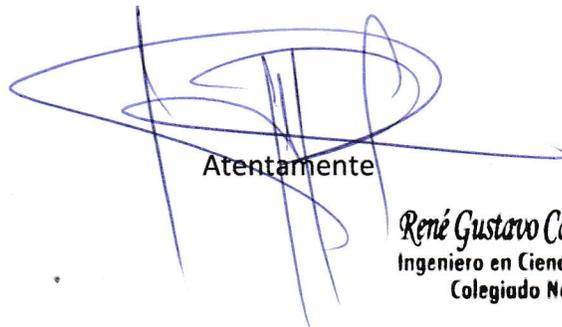
Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Revisor de Trabajos de Graduación
Escuela de Ciencias y Sistemas
Universidad de San Carlos de Guatemala

Ingeniero Azurdia:

Estimado Ingeniero, me dirijo a usted cordialmente para para hacer de su conocimiento que he revisado el trabajo de graduación titulado **"APLICACIÓN COLABORATIVA PARA INCENTIVAR VENTAS Y COMPRAS EN COMUNIDADES LOCALES REMATALOGT"** del estudiante **Luis Renato Garcia Lopez**, quien se identifica con el número de carne **200714251** y CUI **1705633830101**, quien conto con mi asesoría para el desarrollo del mismo.

Considero que el trabajo realizado por el estudiante, cumple con los objetivos bajo los cuales fue planteado y cumple satisfactoriamente cada uno de las actividades planificadas por lo que procedo a aprobarlo.

Agradeciendo la atención dada a la presente.


Atentamente

René Gustavo Catalan Gudiel
Ingeniero en Ciencias y Sistemas
Colegiado No. 10881

Ing. Rene Gustavo Catalan Gudiel
Colegiado No: 10881



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

Guatemala, 28 de noviembre de 2019

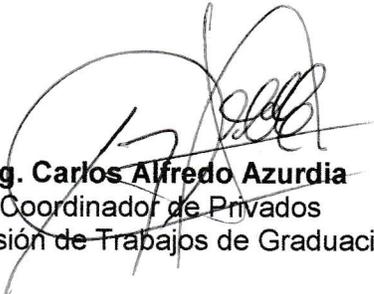
Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **LUIS RENATO GARCIA LOPEZ** con carné **200714251** y **CUI 1705 63383 0101** titulado **“APLICACIÓN COLABORATIVA PARA INCENTIVAR VENTAS Y COMPRAS EN COMUNIDADES LOCALES REMATALOGT”** 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

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 “APLICACIÓN COLABORATIVA PARA INCENTIVAR VENTAS Y COMPRAS EN COMUNIDADES LOCALES “REMATALOGT”, realizado por el estudiante, LUIS RENATO GARCÍA LÓPEZ aprueba el presente trabajo y solicita la autorización del mismo.



Digitally signed by Carlos Gustavo Alonzo
DN: 2.5.4.13=Profesional Titulado, c=GT, l=Guatemala / Guatemala, street=Via 53-65 zona 4 Ed. El Angel 5to nivel of 52, 2.5.4.20=22347420, ou=NA, o=NA, title=Ingeniero en Ciencias y Sistemas Colegiado. 6358, serialNumber=227803167 0101, 2.5.4.45=29020980, 2.5.4.27=06/03/79, email=carlosalonzo@infoutilitygt.com, cn=Carlos Gustavo Alonzo
Date: 2020.10.16 20:34:26 -06'00'

Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 16 de octubre 2020

DTG. 318.2020.

La Decana 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: **APLICACIÓN COLABORATIVA PARA INCENTIVAR VENTAS Y COMPRAS EN COMUNIDADES LOCALES “REMATALOGT”**, presentado por el estudiante universitario: **Luis Renato García López**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana



Guatemala, octubre de 2020

AACE/asga

ACTO QUE DEDICO A:

| | |
|---------------------|--|
| Dios | Por nunca abandonarme y bendecirme grandemente durante toda mi carrera. |
| Mis padres | Paty Lopez y Arturo Garcia, por su amor y apoyo incondicional, sin importar cuantos errores cometí. |
| Mi esposa | María José, por ser el pilar de mi vida y ayudarme a ser mejor cada día. |
| Mis hijos | Sophia y Theo Garcia, por ser mi motivación y la razón de seguir adelante sin importar las dificultades. |
| Mis hermanos | Oscar y Ricardo Garcia Lopez, por formar mi carácter y enseñarme a ser fuerte sin importar la situación. |
| Mis tíos | Ana y Mauricio Cali, por estar siempre para nosotros en las situaciones difíciles. |

AGRADECIMIENTOS A:

| | |
|---|--|
| Universidad de San Carlos de Guatemala | Por formarme como profesional, pero sobre todo por formarme como un ciudadano digno y con principios. |
| Facultad de Ingeniería | Por abrir mi mente y enseñarme a ver soluciones donde nadie más las ve. |
| Jinny y John Laderer | Por darme una oportunidad cuando más la necesité y ser una gran bendición para mi vida. |
| Equipo de trabajo de vcreative | Por hacerme sentir como un miembro de su familia y confiar en mi capacidad como profesional. |
| Mis amigos | Orlando, Gerson, Maiekel, Bryant, Anthony, Fredy, Javier, Edgar, Diego y Chris por el tiempo, la distracción y las risas que siempre estuvieron presentes. |
| Mis familiares | Por estar siempre ahí y apoyarnos mutuamente. |
| Ing. René Catalán | Por su tiempo y apoyo, un excelente profesional siempre dispuesto a ayudar. |
| Ing. Manuel Pinto | Por su tiempo y apoyo incondicional |

ÍNDICE GENERAL

| | |
|--|------|
| ÍNDICE DE ILUSTRACIONES..... | V |
| GLOSARIO | VII |
| RESUMEN..... | XIII |
| OBJETIVOS..... | XV |
| INTRODUCCIÓN | XVII |
| | |
| 1. ANÁLISIS DE LA PROBLEMÁTICA Y PLANTEAMIENTO DE LA SOLUCIÓN..... | 1 |
| 1.1. Teorías que soportan la investigación | 1 |
| 1.1.1. Teoría Económica..... | 1 |
| 1.1.2. Teoría del Aprendizaje..... | 2 |
| 1.1.3. Teoría de la Escuela Sociológica..... | 2 |
| 1.2. Aplicación de las teorías en RemataloGT..... | 2 |
| 1.2.1. Aplicación de la Teoría Económica..... | 3 |
| 1.2.2. Aplicación de la Teoría del Aprendizaje..... | 3 |
| 1.2.3. Aplicación de la Teoría de la Escuela Sociológica | 3 |
| | |
| 2. HERRAMIENTAS Y SERVICIOS A UTILIZAR..... | 5 |
| 2.1. Front End..... | 5 |
| 2.1.1. Apache Cordova | 5 |
| 2.1.2. Modelo Vista Controlador | 6 |
| 2.1.2.1. Modelo..... | 6 |
| 2.1.2.2. Vista..... | 6 |
| 2.1.2.3. Controlador | 7 |

| | | |
|---------|---|----|
| 2.1.3. | Angular | 7 |
| 2.1.4. | Typescript..... | 8 |
| 2.1.5. | Ionic..... | 8 |
| 2.1.6. | API de Geolocalización de Google Maps | 9 |
| 2.1.7. | Geohashing | 9 |
| 2.2. | Back End..... | 10 |
| 2.2.1. | Servicios Web de Amazon | 10 |
| 2.2.2. | Amazon API Gateway | 11 |
| 2.2.3. | Node.JS..... | 12 |
| 2.2.4. | NPM | 12 |
| 2.2.5. | Computación sin servidor | 13 |
| 2.2.6. | AWS Lambda | 14 |
| 2.2.7. | REST..... | 14 |
| 2.2.8. | AWS Servicio de Base de Datos Relacionales..... | 16 |
| 2.2.9. | PostgreSQL..... | 16 |
| 2.2.10. | AWS Servicio de Almacenaje Simple (S3) | 17 |
| 2.2.11. | AWS Elastic Compute Cloud (EC2) | 18 |
| 3. | ANALISIS Y DISEÑO DE LA APLICACIÓN..... | 19 |
| 3.1. | Requerimientos funcionales | 19 |
| 3.1.1. | Generales..... | 19 |
| 3.1.2. | Crear articulo..... | 19 |
| 3.1.3. | Lista de artículos | 20 |
| 3.1.4. | Mensajes | 20 |
| 3.1.5. | Mis artículos | 20 |
| 3.1.6. | Perfil | 21 |
| 3.1.7. | Ver articulo | 21 |
| 3.2. | Requerimientos no funcionales | 21 |
| 3.3. | Diagrama entidad relación | 22 |

| | | |
|-----------|---|----|
| 3.4. | Diagrama general de casos de uso | 24 |
| 3.5. | Topología de la solución..... | 24 |
| 3.6. | Prototipos | 25 |
| 3.7. | Política de seguridad | 31 |
| 3.8. | Política de privacidad | 32 |
| 3.9. | Política de confidencialidad de datos | 33 |
| 3.10. | Leyes de protección y regulación de datos..... | 33 |
| 3.10.1. | Leyes de protección y regulación de datos en Guatemala | 34 |
| 3.10.2. | Consecuencias de la falta de Leyes | 36 |
| 3.10.2.1. | Demandas | 36 |
| 3.10.2.2. | Multas | 36 |
| 3.10.2.3. | Gastos de Compensación y remediación | 37 |
| 3.10.2.4. | Pérdida de Ingresos..... | 37 |
| 3.10.2.5. | Perdida de reputación..... | 37 |
| 4. | DESARROLLO DE LA APLICACIÓN | 39 |
| 4.1. | Herramientas | 39 |
| 4.1.1. | Visual studio code..... | 39 |
| 4.1.2. | Xcode | 39 |
| 4.1.3. | ADT (Android developer tools)..... | 40 |
| 5. | MODELO DE SOSTENIBILIDAD | 41 |
| 5.1. | Benchmarks..... | 41 |
| 5.1.1. | Craigslist..... | 41 |
| 5.1.2. | OfferUp | 41 |
| 5.2. | Benchmarks en Latinoamerica | 42 |
| 5.2.1. | Vendiste.com | 42 |

| | | |
|----------------------|--|----|
| 5.3. | Capas gratuitas de AWS | 43 |
| 5.3.1. | Cuando expira una cuenta de capa gratuita de AWS | 46 |
| 5.3.2. | Servicios sin costo de AWS..... | 46 |
| 5.4. | Costos dentro de AWS..... | 46 |
| 5.5. | Capacidades de crecimiento y posibles fuentes de ingreso | 47 |
| CONCLUSIONES..... | | 49 |
| RECOMENDACIONES..... | | 51 |
| BIBLIOGRAFIA..... | | 53 |

ÍNDICE DE ILUSTRACIONES

FIGURAS

| | | |
|----|---------------------------------------|----|
| 1. | Diagrama entidad relación | 23 |
| 2. | Diagrama general de casos de uso..... | 24 |
| 3. | Topología de la solución | 25 |
| 4. | Lista de artículos | 26 |
| 5. | Perfil | 27 |
| 6. | Crear artículo | 28 |
| 7. | Vista artículo | 29 |
| 8. | Mensajes..... | 30 |
| 9. | Mis artículos | 31 |

TABLAS

| | | |
|----|---------------------------|----|
| I. | Costos dentro de AWS..... | 47 |
|----|---------------------------|----|

GLOSARIO

| | |
|----------------|--|
| .Net | Ambiente de trabajo de desarrollo de software creado por Microsoft, que contiene un conjunto de herramientas que facilitan el desarrollo de aplicaciones para sistema operativo Windows. |
| AMI | Imagen de máquina de Amazon, la cual es una base utilizada para crear instancias de servidores en la nube en los servicios web de Amazon. |
| AWS | Servicios web de Amazon, plataforma de servicios web que provee una amplia gama de infraestructura para el desarrollo de software en la nube. |
| ADB | Puente de depuración de Android, utilizado para depurar aplicaciones en dispositivos con sistema operativo Android. |
| ADT | Herramientas de desarrollo de Android, es un conjunto de herramientas que facilitan el desarrollo de aplicaciones para el sistema operativo Android. |
| Android | Sistema operativo basado en una versión modificada de Linux, utilizado en dispositivos móviles. |

| | |
|-----------------------|---|
| Angular | Ambiente de trabajo de código abierto basado en JavaScript, mantenido y actualizado por Google, utilizado mayormente para desarrollo de aplicaciones web. |
| Apache Cordova | Ambiente de trabajo para desarrollo de aplicaciones móviles, específicamente para aplicaciones híbridas. |
| API | Interfaz de programación de aplicación, es un conjunto de servicios y herramientas disponibles para la construcción de software, en la cual se especifica como interactuar con dichos servicios o herramientas. |
| Back End | Parte de un sistema o aplicación la cual no es accedida por el usuario directamente, usualmente encargada del manejo y modificación de la información. |
| CSS | Hoja de estilos en cascada, es un documento utilizado para dar formato y estilo a páginas web. |
| DOM | Objeto modelo del documento, es una representación lógica de todos los elementos pertenecientes a una página web, así como las funciones disponibles para dichos elementos. |

| | |
|---------------------|--|
| DBMS | Sistema de manejo de base de datos, es un software que contiene todas las herramientas para la creación y manejo de una base de datos. |
| Dirección IP | Dirección de protocolo de internet, es una etiqueta numérica separada por puntos que se utiliza para identificar a un dispositivo conectado a una red. |
| Ecommerce | Término para definir la actividad de comercio por medio de internet, ya sea total o parcialmente. |
| Framework | Ambiente de trabajo, el cual está compuesto por un conjunto de herramientas que facilitan las tareas del programador para cierta plataforma. |
| Front End | Parte de un sistema o aplicación la cual es accedida directamente por los usuarios. |
| GB | Gigabyte, unidad de medida de almacenamiento |
| Geohash | Sistema de codificación de posiciones globales, en el cual se codifica cualquier posición geográfica a una cadena de caracteres la cual varía en su longitud, dependiendo de la precisión. |
| Geohashing | Término utilizado para el uso de geohash en cualquier tipo de aplicación. |

| | |
|------------------------|---|
| Geolocalización | Identificación o estimación de la posición real global de un objeto. |
| HTML | Lenguaje de marcado de hipertexto, es un lenguaje utilizado para la creación de páginas web, en el cual se definen los elementos de la misma y sus referencias hacia otras páginas. |
| HTTP | Protocolo de transferencia de hipertexto: protocolo de comunicación utilizado para conectarse al internet, mediante la transferencia de hipertexto. |
| IDE | Ambiente de desarrollo integrado, contiene un conjunto de herramientas y extensiones que facilitan la programación y depuración de aplicaciones de software. |
| IOS | Sistema operativo móvil creado y desarrollado por Apple Inc. exclusivamente para sus dispositivos móviles. |
| JavaScript | Lenguaje de programación de alto nivel utilizado usualmente para el desarrollo de aplicaciones web. |
| Java | Lenguaje de programación de alto nivel desarrollado por Sun Microsystems, el cual se utiliza para desarrollar aplicaciones en múltiples plataformas. |

| | |
|---------------|--|
| Json | Notación de objeto de JavaScript, es un formato ligero basado en texto diseñado para transmitir información estructurada de forma simple. |
| Kotlin | Lenguaje de programación de alto nivel, el cual es interoperable compatible con librerías de Java. Es utilizado principalmente para el desarrollo de aplicaciones para el sistema operativo Android. |
| MVC | Modelo Vista Controlador. |
| MacOS | Sistema operativo creado y desarrollado por Apple Inc. exclusivamente para sus ordenadores de escritorio. |
| Mipyme | Micro, pequeña y mediana empresa. |
| Npm | Manejador de paquetes de node, plataforma en la cual se puede manejar las dependencias de un proyecto de una forma estructurada y concisa. |
| Python | Lenguaje de programación de alto nivel, utilizado principalmente para el manejo de altas cantidades de información |
| Rest | Transferencia representacional de estado, es un estilo de arquitectura que proporciona estándares entre varios sistemas informáticos en la web, lo que facilita la comunicación entre los mismo. |

| | |
|-------------------|--|
| S3 | Servicio de almacenaje simple. Servicio proporcionado por Amazon para el manejo y almacenaje de datos. |
| SPA | Aplicación de página única. Tipo de aplicación web en la cual se carga una única página durante todo el uso de la misma, para reducir la interacción entre cliente y servidor. |
| SSL | Capa de socket seguro, estándar de seguridad para establecer una comunicación encriptada entre navegador y servidor web. |
| SQL | Lenguaje de consulta estructurada, es utilizado para el manejo y acceso de bases de datos de tipo relacional. |
| TIC | Tecnologías de información y comunicación. |
| Typescript | Lenguaje de programación desarrollado y mantenido por Microsoft, el cual es un supe conjunto de funciones de JavaScript. A diferencia de JavaScript, este utiliza tipos de datos nativos y complejos para reducir los errores durante el desarrollo. |
| VM | Máquina virtua, es una emulación de un sistema de computadora dentro de otra. |

RESUMEN

Cada vez estamos más acostumbrados a encontrar lo que necesitamos desde nuestro móvil; así mismo, hemos visto en los últimos años el crecimiento del comercio informal en la sociedad. Ya sea por factores económicos o regulatorios, cada vez se hace más difícil para un individuo tener un local de comercio formal. Por esta misma razón, todos los comerciantes informales pierden oportunidades de venta, ya que no es fácil encontrar sus productos en línea ni tampoco físicamente.

Debido a esto, en el siguiente trabajo de tesis desarrollo la aplicación “RemataloGT”, la cual tiene como objetivo hacer que los comerciantes informales o personas individuales tengan una plataforma para poder ofertar sus productos sin ningún tipo de intermediario, utilizando geolocalización y geo-hashing para poder, de esta forma, presentarlos a compradores que estén buscando sus productos dentro de las comunidades cercanas. Es una plataforma gratuita y de fácil uso para emprendedores y mipymes que desean y necesitan apoyo en sus etapas iniciales para impulsar sus ventas.

OBJETIVOS

General

Desarrollar e implementar una aplicación altamente disponible y escalable aplicando los conocimientos técnicos y teóricos adquiridos durante el desarrollo del ciclo educativo universitario, que incentive la inclusión del sector económico Mypimes dentro del mundo digital. Además, facilitar a los usuarios individuales encontrar artículos en venta de cualquier tipo.

Específicos

1. Brindar una aplicación donde cualquier empresa guatemalteca pueda tener un catálogo de sus artículos disponibles.
2. Permitir a dichas empresas en Guatemala el acceso a un canal digital funcional para publicar fotos de sus artículos, así como categorizarlos e incluir precios y descripciones de los mismos.
3. Facilitar una aplicación donde usuarios individuales pueden consultar ofertas de productos y precios disponibles en su comunidad.
4. Facilitar dentro de la aplicación la comunicación entre comprador y vendedor, por medio de un módulo de mensajes.

INTRODUCCIÓN

Según el último estudio del Ministerio de Economía las *“Mipymes representan un 99.57 % del parque empresarial nacional”*¹. La cantidad de empresas que se dedican al E-Commerce *“se incrementó en un 200 % entre 2016 y 2017”*², por lo cual es de suma importancia fortalecer este tipo de empresas, ya que esto significa fortalecer la economía del país. Así mismo, es esencial continuar desarrollando herramientas que amplíen sus oportunidades de desarrollo y crecimiento en sus operaciones.

Debido al crecimiento de este sector económico, así como su creciente incursión en el e-Commerce, se ha propuesto desarrollar una aplicación que incentiva y apoya a todas las Mypimes nacionales en sus operaciones de ventas y a consumidores, que permita encontrar de forma fácil e intuitiva los productos de dichas empresas, utilizando geo-posicionamiento para filtrar y mostrar productos relevantes en cualquier comunidad del país. Con esta aplicación se mejorará la productividad y competitividad mediante la utilización de Tecnologías de Información y Comunicación (TIC) de última generación.

El siguiente trabajo de graduación consiste en la elaboración de dicha aplicación, que será desarrollada en ambiente híbrido para que esté disponible tanto en los sistemas operativos Android como IOS. Se mostrarán los elementos de tecnología utilizados para implementar en su forma total la arquitectura de software necesaria para cumplir con todos los requerimientos establecidos.

¹ MINECO. <https://www.mineco.gob.gt/guatemala-celebra-por-primera-vez-el-d%C3%ADa-internacional-de-la-micro-peque%C3%B1a-y-mediana-empresa>. Consulta: 7 de febrero de 2019.

² Diario de CA <https://dca.gob.gt/noticias-guatemala-diario-centro-america/mas-empresas-se-suman-al-comercio-electronico-en-el-pais/>. Consulta: 18 de febrero de 2019.

1. ANÁLISIS DE LA PROBLEMÁTICA Y PLANTEAMIENTO DE LA SOLUCIÓN

Este trabajo de investigación está basado en el desarrollo de una aplicación, por lo cual es importante comprender y explicar, desde los diferentes puntos de vista, cuáles son las razones que llevan a los consumidores a utilizar una marca u otra, o utilizar un producto u otro.

1.1. Teorías que soportan la investigación

Las teorías de comportamiento económico son las que estudian a través de modelos la forma en la cual las personas como individuos racionales toman sus decisiones económicas, y como logran hacer una decisión cuando se encuentran con varias alternativas para una sola acción económica.

1.1.1. Teoría económica

Alfred Marshal (1842 – 1924) fue quien inicialmente introdujo esta teoría, en la cual considera que el hombre siempre busca maximizar su utilidad. Por tanto, siempre buscará comprar el producto o servicio que le dé más utilidad en función del costo, lo cual maximizará la relación de coste vs beneficio.

Los consumidores tienen completo uso de la razón al tomar una decisión de compra o uso. Esto significa que toman sus decisiones de forma independiente, es decir, que no tienen ninguna influencia externa y que sus preferencias serán constantes a lo largo del tiempo.

1.1.2. Teoría del aprendizaje

Esta teoría se basa originalmente en las teorías de Ivan Pavlov (1849 – 1936) y sostiene que, si bien los primeros actos de compra pueden basarse en factores económicos o de otro tipo, después de haber realizado la primera compra, los consumidores actúan a partir de las experiencias adquiridas en sus compras previas. Pueden realizar cambios en sus decisiones de compra para satisfacer mejor sus necesidades, en cierta forma, aplicando ensayo y error.

Debido a esto, es correcto asumir que el consumidor es infiel por naturaleza cuando no se cumplen sus expectativas, pero si se logra que los consumidores conozcan el producto o servicio, si lo utiliza por primera vez y no se le defrauda, se iniciará una lealtad hacia el mismo.

1.1.3. Teoría de la Escuela sociológica

Desarrollada principalmente por grandes críticos del sistema de consumo moderno, mayormente Thorstein Veblen (1857 - 1929). Fue autor de la *Sociedad del Ocio*, teoría en la cual se sostiene que la razón principal que define el comportamiento de un consumidor es la necesidad de integración en un grupo social de referencia o inspiracional. Quiere decir que los consumidores adoptarán actitudes o costumbres con tal de encajar en la sociedad, imitando a un grupo de referencia deseado y evitando las del no deseado.

1.2. Aplicación de las teorías en RemataloGT

Ya hemos definido tres teorías en las cuales se define por qué y cómo deciden los usuarios o consumidores utilizar un servicio o producto. Con esta investigación se desea aumentar la incursión de las Mypymes dentro del

e-commerce por medio de la implementación de una plataforma digital. Esta no alcanzará su principal objetivo si no es utilizada, por lo cual debemos definir cuáles serán nuestros enfoques para que los usuarios utilicen la plataforma y de esta forma lograr el objetivo planteado.

1.2.1. Aplicación de la teoría económica

Si los consumidores se guían principalmente por los precios o utilidad del producto, la plataforma debe centrarse en transmitir la idea de que los precios más accesibles para comprar productos se encontraran en esta aplicación. Los conceptos de oferta, oportunidad, descuento, etc. deben ser claves en la interfaz, con las cifras y ofertas visibles dentro de la misma.

1.2.2. Aplicación de la teoría del aprendizaje

Basados en esta teoría, podemos asumir que los consumidores o usuarios tomarán la decisión de utilizar o consumir un producto con base en su aprendizaje o experiencia del mismo. Debido a esto, en nuestra plataforma debemos estar seguros de que al momento que un usuario la utilice, debe ser intuitiva y libre de problemas, ya que si este queda insatisfecho es altamente probable que ya no vuelva a utilizarla.

1.2.3. Aplicación de la teoría de la escuela sociológica

Según esta teoría, los consumidores basarán su decisión de utilizar un producto o servicio en lo que algún grupo social dicte. Es por esto que esta plataforma deberá tener presencia en foros de e-commerce dentro del país y dentro de redes sociales, para que los grupos sociales previamente establecidos

y que tienen influencia sobre un grupo más grande de consumidores, puedan redirigir usuarios y consumidores a nuestra aplicación.

2. HERRAMIENTAS Y SERVICIOS POR UTILIZAR

2.1. Front end

Se le llama Front End a la parte visual con la cual un usuario interactúa con una aplicación de software, en la cual están incluidos todos los botones, cuadros de texto e información que se presentara hacia un usuario.

2.1.1. Apache Cordova

Apache Cordova es un framework de desarrollo de aplicaciones móviles que se puede utilizar para crear aplicaciones móviles multiplataforma utilizando HTML5 y JavaScript puro. Por multiplataforma nos referimos a que el código base de la aplicación se puede escribir una vez utilizando HTML5 y JavaScript y se puede ejecutar en múltiples plataformas móviles de destino, como Android o iOS.

Fue creado originalmente por Nitobi. Luego de esto, Adobe Systems compró Nitobi en 2011. Lo renombró como PhoneGap y lanzó una versión de código abierto del software llamado Apache Cordova.

Este **framework** permite a los programadores de software crear aplicaciones para dispositivos móviles utilizando CSS3, HTML5 y JavaScript en lugar de confiar en las API específicas de la plataforma, como las de Android, iOS o Windows Phone. Permite completar el código CSS, HTML y JavaScript dependiendo de la plataforma del dispositivo. Amplía las características de HTML y JavaScript para trabajar con el dispositivo. Las aplicaciones resultantes son híbridas, lo que significa que no son aplicaciones móviles realmente nativas

(porque toda la representación del diseño se realiza a través de vistas web en lugar del marco de la interfaz de usuario nativa de la plataforma) ni puramente basadas en la web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para distribución y tener acceso a las API nativas de dispositivos).

2.1.2. Modelo vista controlador

El modelo Vista Controlador (MVC) es un patrón de arquitectura que separa una aplicación en tres componentes lógicos principales: el modelo, la vista y el controlador. Cada uno está diseñado para manejar aspectos específicos de desarrollo en una aplicación. MVC es uno de los marcos de desarrollo web estándar de la industria más utilizados para crear proyectos escalables y extensibles.

2.1.2.1. Modelo

El componente modelo corresponde a toda la lógica relacionada con los datos con los que trabaja el usuario. Puede representar los datos que se transfieren entre los componentes de Vista y Controlador o cualquier otro dato relacionado con la lógica de negocios. Por ejemplo, un objeto Cliente recuperará la información del cliente de la base de datos, la manipulará y actualizará los datos a la base de datos o la usará para representar los mismos.

2.1.2.2. Vista

El componente Vista se utiliza para toda la lógica de la interfaz de usuario de la aplicación. Por ejemplo, la vista del Cliente incluirá todos los componentes

de la interfaz de usuario, como cuadros de texto, menús desplegables, etc., con los que el usuario final interactúa.

2.1.2.3. Controlador

Los componentes Controlador actúan como una interfaz entre los componentes Modelo y Vista para procesar toda la lógica de negocios y las solicitudes entrantes, manipular los datos utilizando el componente Modelo e interactuar con las Vistas para generar el resultado final. Por ejemplo, el controlador del Cliente manejará todas las interacciones y entradas de la Vista del Cliente y actualizará la base de datos utilizando el Modelo del Cliente. El mismo controlador se utilizará para ver los datos del cliente.

2.1.3. Angular

Angular es un **framework** basado en JavaScript, cuyo objetivo principal es la simplificación. Se destaca en la creación de aplicaciones web dinámicas de una sola página (SPA) y es compatible con la estructura de programación Modelo Vista Controlador (MVC).

Angular simplifica el problema de crear aplicaciones web dinámicas, lo que permite al desarrollador ampliar la funcionalidad de HTML al otorgarles la capacidad de crear nuevos constructores con directivas de Angular. De esta manera, se abstrae la manipulación del DOM que usualmente resulta complicada, y la reduce a elementos simples que se pueden incrustar directamente en una plantilla HTML.

2.1.4. Typescript

TypeScript es un super conjunto de JavaScript, lo que significa que contiene toda la funcionalidad de JavaScript y algo más. Por lo tanto, cualquier programa escrito en JavaScript válido también se ejecutará como se espera en TypeScript. De hecho, cualquier archivo TypeScript al ser compilado resulta en un archivo de JavaScript simple. Entonces, ¿cuál es la diferencia? TypeScript ofrece más control sobre el código a través de definiciones de tipo, interfaces y clases.

TypeScript fue creado por Microsoft y se lanzó en 2012 después de dos años de desarrollo. Fue creado para permitir la verificación de tipos estáticos, que sería particularmente útil al desarrollar aplicaciones a gran escala. Curiosamente, una de las razones por las que Microsoft desarrolló TypeScript fue que sus equipos internos tenían problemas para escribir código de JavaScript que fuera escalable para los propios proyectos de Microsoft, especialmente el equipo que trabaja en Bing Maps.

2.1.5. Ionic

Ionic es un framework de código abierto para el desarrollo de aplicaciones móviles híbridas creado por Max Lynch, Ben Sperry y Adam Bradley de Drifty Co. en 2013. La versión original se lanzó en 2013 y se construyó sobre AngularJS y Apache Cordova. Los lanzamientos más recientes, conocidos como Ionic 3 o simplemente Ionic, se basan en Angular. Ionic proporciona herramientas y servicios para desarrollar aplicaciones móviles híbridas utilizando tecnologías web como CSS, HTML5 y Sass. Las aplicaciones pueden construirse con estas tecnologías web y luego distribuirse a través de las tiendas de aplicaciones nativas para que se instalen en dispositivos aprovechando la utilización de Apache Cordova.

Ionic Framework es una biblioteca de componentes de interfaz de usuario, que son elementos reutilizables que sirven como bloques de construcción para una aplicación. Los componentes de Ionic están contruidos con estándares web que utilizan HTML, CSS y JavaScript. A pesar de que los componentes están preconstruidos, están diseñados desde cero para ser altamente personalizables. Las aplicaciones pueden hacer que cada componente esté personalizado, lo que permite que cada una tenga su propia apariencia. Más específicamente, los componentes de Ionic pueden ser fácilmente tematizados para cambiar globalmente la apariencia en toda la aplicación.

Una aplicación implementada en Ionic está hecha de bloques de construcción de alto nivel llamados componentes. Estos le permiten construir rápidamente una interfaz para su aplicación. Este framework incluye varios componentes, incluidos modales, ventanas emergentes y tarjetas.

2.1.6. API de geolocalización de Google Maps

La API de geolocalización de Google Maps proporciona a los desarrolladores herramientas y documentación para incorporar la geolocalización con tecnologías móviles y WiFi. La API de geolocalización devuelve específicamente un radio de ubicación y precisión basado en la información de las torres de telefonía móvil y los nodos WiFi que el cliente móvil puede detectar.

2.1.7. Geohashing

Geohash es un sistema de geocodificación inventado por Gustavo Niemeyer que nos permite saber en qué área del mapa se encuentra un usuario. El usuario podría estar en cualquier lugar de esta área, no necesariamente en el

centro. Esto es realmente útil, ya que la latitud y la longitud solo representan un punto único en un mapa, mientras que un geohash describe un área fija.

Un geohash es una forma conveniente de expresar una ubicación (en cualquier parte del mundo) mediante una cadena alfanumérica corta, con mayor precisión obtenida con cadenas más largas. Un geohash identifica una celda rectangular en cada nivel; cada carácter adicional identifica una de las 32 sub-celdas de dicha celda.

2.2. Back End

En el ámbito de desarrollo de software el back end es la parte interna de una aplicación, la cual no tiene una interfaz gráfica y es puramente código que es ejecutado por medio de alguna interacción entre el usuario y la misma, usualmente la interacción es por medio del Front End.

2.2.1. Servicios web de Amazon

Los servicios web de Amazon se componen de una serie de productos y servicios de computación en la nube. Esta división de Amazon altamente rentable proporciona servidores, almacenamiento, redes, informática remota, correo electrónico, desarrollo móvil y seguridad. Se divide en 12 regiones globales, cada una de las cuales tiene múltiples zonas de disponibilidad en las que se encuentran sus servidores. Estas regiones se dividen para permitir a los usuarios establecer límites geográficos en sus servicios (si así lo desean), pero también para brindar seguridad al diversificar las ubicaciones físicas en las que se guardan los datos.

Debido a que el costo de los servicios web de Amazon está basado en función del uso de los clientes, las empresas nuevas y pequeñas pueden ver los beneficios obvios de usar Amazon para sus necesidades tecnológicas. De hecho, es ideal para construir un negocio desde la base, ya que proporciona todas las herramientas necesarias para que las empresas se inicien en la utilización de computación en la nube. De igual forma para las compañías existentes, Amazon proporciona servicios de migración de bajo costo para que su infraestructura existente pueda trasladarse sin problemas.

2.2.2. Amazon API Gateway

Amazon API Gateway es un conjunto de servicios de AWS que permite a un desarrollador conectar aplicaciones que estén dentro de AWS a recursos que se encuentren dentro de AWS, como servidores o código. Amazon API Gateway permite que un cliente de AWS aumente la utilidad general de otros servicios en la nube de Amazon.

Amazon API Gateway proporciona dos medios para manejar el tráfico de llamadas API. La limitación de API está disponible para limitar el número de llamadas API por hora o por día. Esto ayuda a mantener el rendimiento de las llamadas cuando hay un aumento inesperado en las llamadas a la API, como por ejemplo cuando muchos usuarios comienzan a usar una aplicación al mismo tiempo. Dado que muchas llamadas a la API utilizan la misma información y devuelven el mismo resultado, se puede utilizar el almacenamiento en caché para proporcionar respuestas a API comunes en lugar de realizar todo el procesamiento requerido para producir un resultado. El almacenamiento en caché reduce el número de llamadas a la API y mejora aún más el rendimiento de las llamadas.

2.2.3. Node.JS

Node.js (Node) es una plataforma de desarrollo de código abierto para ejecutar código JavaScript en el lado del servidor. Esta plataforma es útil para desarrollar aplicaciones que requieren una conexión persistente desde el navegador al servidor y, a menudo, se usa para aplicaciones en tiempo real, como chat, noticias y notificaciones web.

Node.js está diseñado para ejecutarse en un servidor HTTP dedicado y para emplear un solo hilo con un proceso a la vez. Las aplicaciones implementadas en Node.js están basadas en eventos y se ejecutan de forma asíncrona. El código creado en la plataforma Node no sigue el modelo tradicional de recibir, procesar, enviar, esperar, recibir. En su lugar, Node procesa las solicitudes entrantes en una pila de eventos constantes y envía pequeñas solicitudes una tras otra sin esperar respuestas.

Esto es un alejamiento de los modelos convencionales que ejecutan procesos más grandes y complejos y ejecutan varios subprocesos simultáneamente, con cada subproceso esperando su respuesta adecuada antes de continuar.

2.2.4. NPM

NPM (Administrador de paquetes de node) es el administrador de paquetes predeterminado para Node.js y está escrito completamente en JavaScript. Desarrollado por Isaac Z. Schlueter, se lanzó inicialmente el 12 de enero de 2010. NPM administra todos los paquetes y módulos para Node.js y consta del cliente de línea de comandos npm. Se instala en el sistema con la instalación de Node.js.

Los paquetes y módulos necesarios en el proyecto Node se instalan usando NPM.

Un paquete contiene todos los archivos necesarios para un módulo y los módulos son las bibliotecas de JavaScript que se pueden incluir en el proyecto Node, de acuerdo con los requisitos del proyecto.

NPM puede instalar todas las dependencias de un proyecto a través del archivo package.json. También puede actualizar y desinstalar paquetes. En el archivo package.json, cada dependencia puede especificar un rango de versiones válidas utilizando el esquema de versionamiento semántico, lo que permite a los desarrolladores actualizar automáticamente sus paquetes y, al mismo tiempo, evitar cambios de ruptura no deseados.

2.2.5. Computación sin servidor

La computación sin servidor es una arquitectura donde la ejecución del código es totalmente administrada por un proveedor de la computación en la nube, en lugar del método tradicional de desarrollo de aplicaciones y su implementación en servidores.

Esto significa que los desarrolladores no tienen que preocuparse por administrar, provisionar y mantener servidores al implementar el código. Antes de esta arquitectura, un desarrollador tendría que definir cuánta capacidad de almacenamiento y base de datos sería necesaria antes de la implementación, lo que ralentizaría todo el proceso.

La computación sin servidor se basa en funciones o, más específicamente, en funciones como servicios, donde los desarrolladores dividen sus aplicaciones

en pequeños fragmentos sin estado. Esto significa que pueden ejecutarse sin ningún contexto con respecto al servidor subyacente.

2.2.6. AWS Lambda

AWS Lambda es un servicio de computación en la nube impulsado por eventos de AWS. Permite a los desarrolladores programar funciones de pago por uso sin tener que provisionar almacenamiento o recursos de cómputo para darles soporte.

Uno de los principales beneficios de AWS Lambda es que abstrae la administración del servidor de los requerimientos del profesional de IT dentro de una organización. Con AWS Lambda, Amazon administra los servidores, lo que permite que un desarrollador se centre más en escribir el código de la aplicación.

AWS admite códigos escritos en una variedad de lenguajes de programación. Los lenguajes de AWS Lambda incluyen Node.js, Python, Java y C #. Los desarrolladores también pueden usar herramientas de compilación de código, como Maven o Gradle, y paquetes para construir funciones.

2.2.7. REST

REST (REpresentational State Transfer) es un estilo arquitectura para el desarrollo de servicios web. REST es popular debido a su simplicidad y al hecho de que se basa en los sistemas y características existentes del protocolo de internet HTTP para lograr sus objetivos, en lugar de crear nuevos estándares, marcos y tecnologías.

Una de las principales ventajas de usar REST, tanto desde la perspectiva del cliente como del servidor, es que las interacciones basadas en REST ocurren mediante llamadas o métodos que son familiares para cualquiera que esté acostumbrado a usar el protocolo de transferencia de hipertexto (HTTP) de Internet.

Un ejemplo de esta disposición es que las interacciones basadas en REST comunican su estado utilizando códigos de estado HTTP estándar. Entonces, un 404 significa que no se encontró un recurso solicitado; un código 401 significa que la solicitud no fue autorizada; un código 200 significa que todo está bien; y un 500 significa que hubo un error de aplicación irrecuperable en el servidor.

De manera similar, los detalles como el cifrado y la integridad del transporte de datos se resuelven al no agregar nuevos marcos o tecnologías, sino confiando en el conocido cifrado de capa de sockets seguros (SSL) y la seguridad de la capa de transporte (TLS). Por lo tanto, toda la arquitectura REST se basa en conceptos con los que la mayoría de los desarrolladores ya están familiarizados.

REST es también un estilo de arquitectura independiente del lenguaje. Las aplicaciones basadas en REST se pueden escribir en cualquier idioma, ya sea Java, Kotlin, .NET, Angular o JavaScript. Siempre que un lenguaje de programación pueda realizar solicitudes basadas en la web utilizando HTTP, es posible que ese lenguaje se utilice para invocar una API o servicio web RESTful. De manera similar, los servicios web RESTful se pueden escribir en cualquier idioma, por lo que los desarrolladores encargados de implementar dichos servicios pueden elegir las tecnologías que mejor se adapten a su situación.

2.2.8. AWS Servicio de base de datos relacionales

El Servicio de base de datos relacional de Amazon (Amazon RDS) es un servicio de base de datos SQL proporcionado por Amazon Web Services (AWS). Amazon RDS es compatible con una serie de DBMS para almacenar y organizar datos y ayuda con las tareas de administración de la base de datos, como la migración, el respaldo, la recuperación y la aplicación de parches de seguridad.

Un administrador de la nube utiliza Amazon RDS para configurar, administrar y escalar una instancia de base de datos relacional en la nube. Este servicio también realiza copias de seguridad automáticas de las instancias de la base de datos RDS. Se puede capturar una instantánea diaria de los datos y retiene los registros de transacciones para permitir la recuperación de un punto en el tiempo. RDS también parchea automáticamente el software del motor de base de datos.

Para mejorar la disponibilidad y confiabilidad de las cargas de trabajo, Amazon RDS permite la replicación. Un administrador también puede habilitar el failover automático en múltiples zonas de disponibilidad con replicación de datos síncrona. Un usuario de AWS controla Amazon RDS a través de la consola de administración de AWS, las API de Amazon RDS o la interfaz de línea de comandos de AWS.

2.2.9. PostgreSQL

PostgreSQL es un sistema de administración de base de datos de propósito general y relacional de objetos, es el sistema de base de datos de código abierto más avanzado. PostgreSQL fue desarrollado sobre la base de POSTGRES 4.2

en el Departamento de Ciencias de la Computación de Berkeley en la universidad de California.

Este DBMS admite consultas tanto SQL (relacionales) como JSON (no relacionales). Es un sistema de gestión de bases de datos altamente estable, respaldado por más de 20 años de desarrollo comunitario que ha contribuido a sus altos niveles de resiliencia, integridad y corrección. Se utiliza como almacenamiento de datos principal para muchas aplicaciones web, móviles, geoespaciales y de análisis. La última versión es PostgreSQL 11.

PostgreSQL tiene un rico historial de compatibilidad con tipos de datos avanzados, y admite un nivel de optimización del rendimiento que es común en sus homólogos de bases de datos comerciales, como Oracle y SQL Server. AWS es compatible con PostgreSQL a través de un servicio de base de datos totalmente administrado con el Servicio de base de datos relacional de Amazon (RDS).

2.2.10. AWS Servicio de almacenaje simple (S3)

S3 es un servicio de almacenamiento en la nube escalable, de alta velocidad, basado en la web, diseñado para realizar copias de seguridad y guardar datos y aplicaciones dentro de los servicios web de Amazon. Amazon S3 se diseñó con un conjunto mínimo de características y se creó para facilitar la escalación web para los desarrolladores.

Amazon S3 es un servicio de almacenamiento de objetos, que difiere del almacenamiento en bloque y de archivos en la nube. Cada objeto se almacena como un archivo con sus metadatos incluidos y se le asigna un número de identificación. Las aplicaciones utilizan este número de identificación para

acceder a un objeto. A diferencia del almacenamiento de archivos en la nube y bloques, un desarrollador puede acceder a un objeto a través de una API REST.

Este servicio brinda a los suscriptores acceso a los mismos sistemas que Amazon utiliza para ejecutar sus propios sitios web. S3 permite a los clientes cargar, almacenar y descargar prácticamente cualquier archivo u objeto que tenga un tamaño de hasta cinco terabytes (TB), con la carga más grande con un máximo de cinco gigabytes (GB).

2.2.11. AWS Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio basado en web que permite a las empresas ejecutar programas de aplicación en la nube pública de AWS. Amazon EC2 permite a un desarrollador activar máquinas virtuales (VM), que proporcionan capacidad informática para proyectos de IT y cargas de trabajo en la nube que se ejecutan con centros de datos de AWS globales.

Un usuario de AWS puede aumentar o disminuir en cuestión de minutos la capacidad de la instancia según sea necesario, utilizando la interfaz web de Amazon EC2 o una interfaz de programación de aplicaciones. De igual forma puede codificar una aplicación para escalar instancias automáticamente.

Para usar EC2, un desarrollador crea una imagen de máquina de Amazon (AMI) que contiene un sistema operativo, programas de aplicación y ajustes de configuración. La AMI se carga en el Servicio de almacenamiento simple de Amazon (Amazon S3) y se registra en Amazon EC2, creando un identificador de AMI. Una vez hecho esto, el suscriptor puede solicitar máquinas virtuales según sea necesario.

3. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

3.1. Requerimientos funcionales

Los requisitos funcionales definen el comportamiento básico del sistema. Esencialmente, son los que definen lo que el sistema hace o no debe hacer, y pueden pensarse en términos de cómo responde el sistema a las entradas. Los requisitos funcionales generalmente definen comportamientos que el sistema debería tener a distintas acciones del usuario.

3.1.1. Generales

- Los usuarios podrán registrarse.
- Los usuarios podrán iniciar sesión.
- Los usuarios podrán cerrar sesión.
- Los usuarios podrán iniciar sesión por medio de Facebook.
- Los usuarios podrán recuperar su contraseña.
- Al recuperar la contraseña se deberá enviar un correo con las instrucciones.
- El sistema deberá restringir el acceso a los componentes que necesitan de una cuenta.

3.1.2. Crear artículo

- El usuario podrá subir fotos desde su galería.
- El usuario podrá subir fotos directamente desde la cámara.

- El usuario podrá definir el título, categoría, descripción, condición y precio del artículo.
- El usuario podrá definir la ubicación del artículo.

3.1.3. Lista de artículos

- El usuario podrá ver la lista de artículos disponibles basado en su geolocalización y filtros.
- Se podrá realizar búsquedas por texto.
- El usuario podrá cambiar la localización de búsqueda.
- Se podrá realizar búsqueda por categoría.
- Se podrá filtrar resultados por precio.
- Se podrá ordenar los resultados por precio y fecha de creación.
- Se podrá filtrar resultados por distancia del artículo.

3.1.4. Mensajes

- El usuario podrá ver la lista de mensajes recibidos.
- El usuario podrá enviar mensajes.

3.1.5. Mis artículos

- El usuario podrá ver la lista de sus artículos activos
- El usuario podrá ver la lista de sus artículos marcados como vendidos.
- El usuario podrá ver la lista de sus artículos borrados.
- El usuario podrá marcar un artículo activo como vendido.
- El usuario podrá editar un artículo activo.
- El usuario podrá borrar un artículo activo.

- El usuario podrá reactivar un artículo borrado.
- El usuario podrá reactivar un artículo marcado como vendido.

3.1.6. Perfil

- Los usuarios podrán subir su foto de perfil.
- Los usuarios podrán cambiar su información.
- Los usuarios podrán cambiar su contraseña.

3.1.7. Ver artículo

- La vista debe mostrar el precio del artículo.
- La vista debe mostrar el usuario que publico el artículo.
- La vista debe mostrar la localización del artículo.
- La vista debe mostrar la descripción del artículo.
- La vista debe mostrar el número de teléfono y correo electrónico del usuario que publico el artículo.
- El usuario podrá marcar un artículo como favorito.

3.2. Requerimientos no funcionales

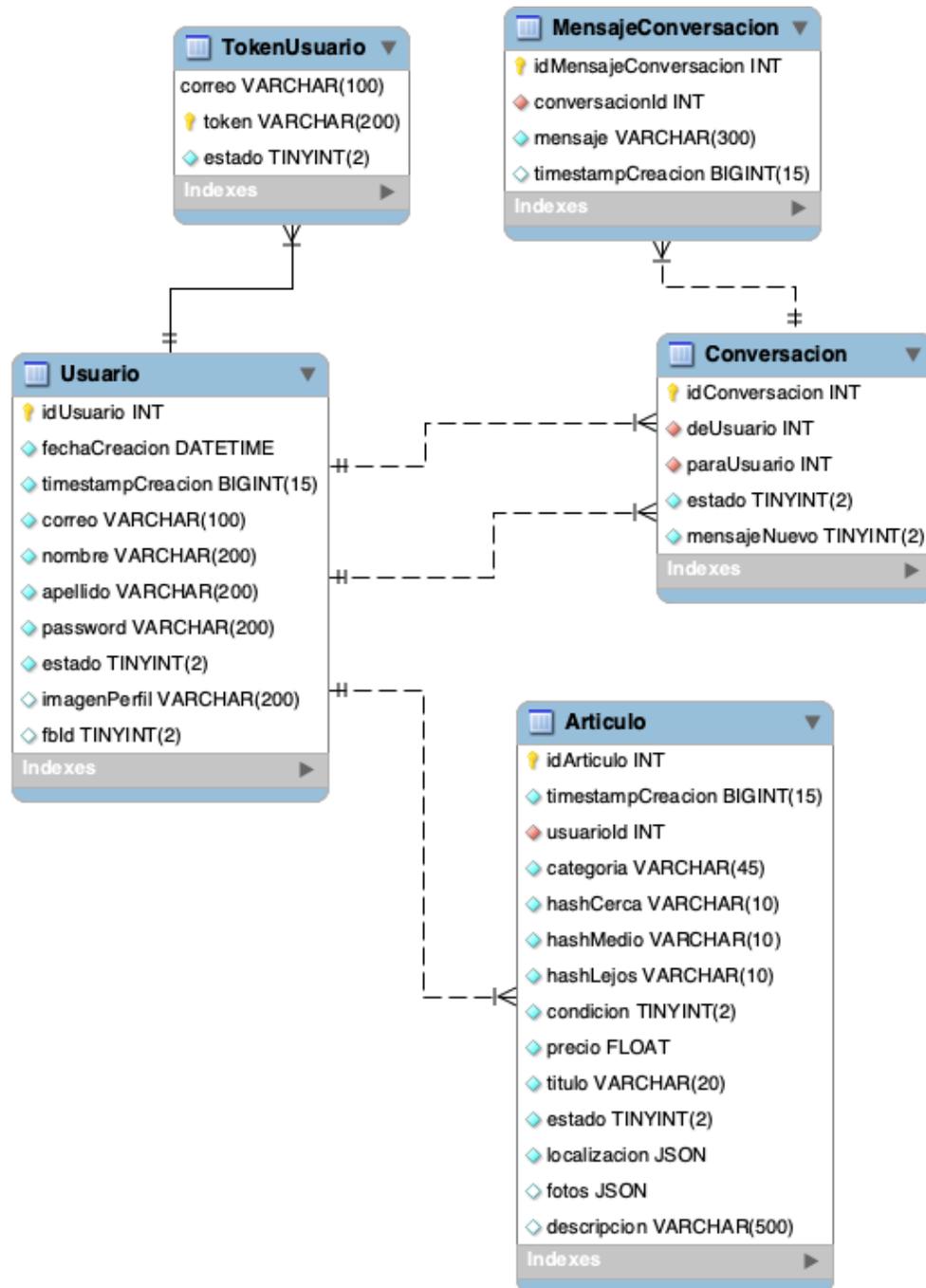
- Los usuarios podrán navegar a las diferentes secciones desde la pantalla inicial.
- La creación de un artículo será por medio de varias pantallas, simulando pasos.
- El usuario podrá definir la ubicación del artículo.
- La lista de artículos deberá adaptar su vista dependiendo del tamaño del dispositivo

- La lista de artículos deberá cargar automáticamente más artículos al llegar al final de la lista actual.
- La lista de mis artículos deberá estar separada por medio de pestañas.
- La vista de artículo deberá poder mostrar las fotografías en tamaño completo.
- La vista de artículo deberá mostrar un botón para enviar mensaje al usuario que publico el artículo.

3.3. Diagrama entidad relación

Este diagrama muestra la relación entre conjuntos de entidades. Un conjunto de entidades es un grupo de entidades similares y estas entidades pueden tener atributos. En términos de una base de datos, una entidad es una tabla en la base de datos, por lo que al mostrar la relación entre las tablas y sus atributos, el diagrama muestra la estructura lógica completa de una base de datos.

Figura 1. Diagrama entidad relación

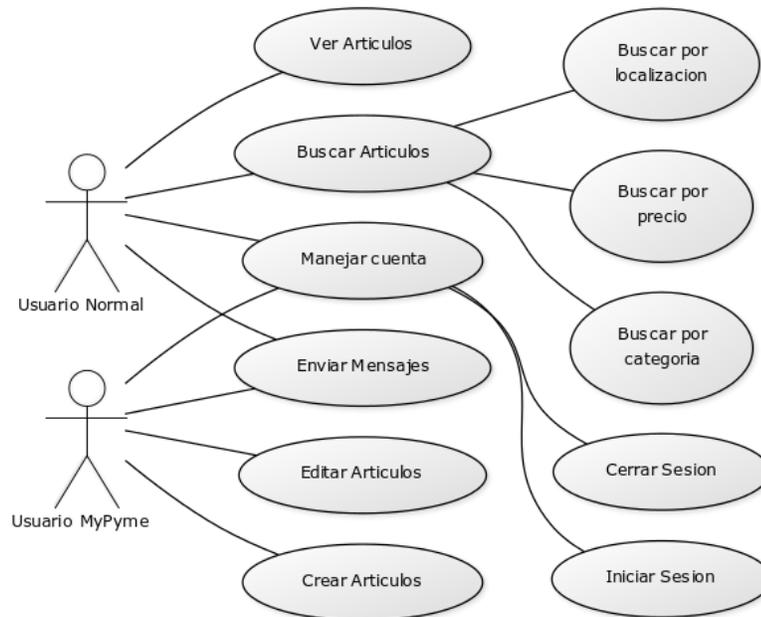


Fuente: elaboración propia.

3.4. Diagrama general de casos de uso

Este diagrama es la forma principal de definir los requerimientos de un sistema o software. Los casos de uso especifican el comportamiento esperado, y no el método exacto para hacerlo realidad es decir solo se muestra que debe hacer el sistema sin especificar como.

Figura 2. Diagrama general de casos de uso

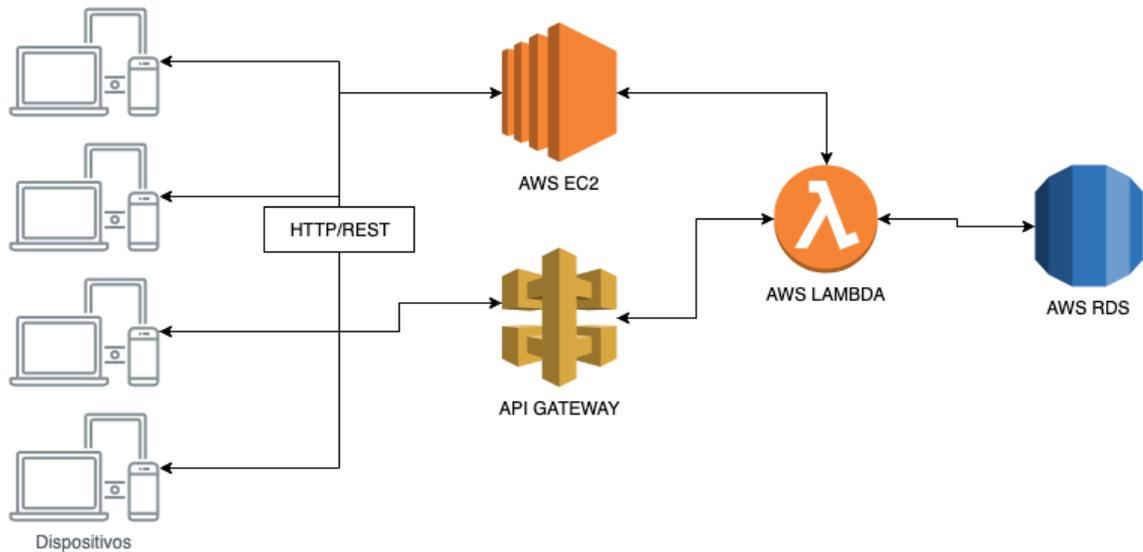


Fuente: elaboración propia.

3.5. Topología de la solución

Una topología es un diagrama de alto nivel en el cual se especifican las capas que un software tendrá, así como también cualquier tipo de servicio externo que se utilizara en la misma para lograr cumplir con los requerimientos.

Figura 3. Topología de la solución

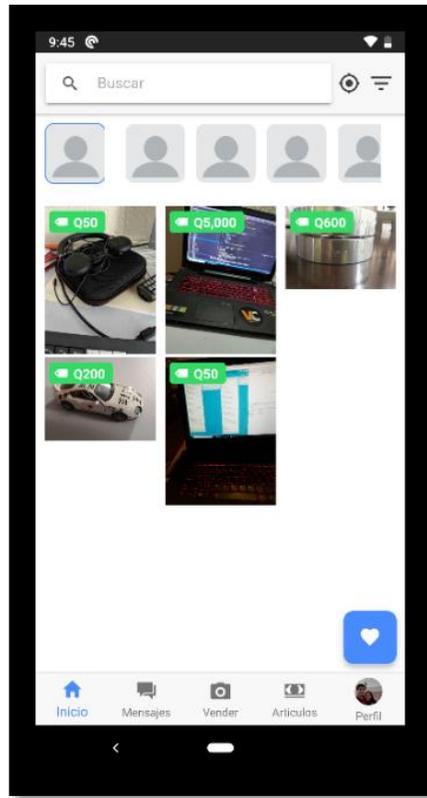


Fuente: elaboración propia.

3.6. Prototipos

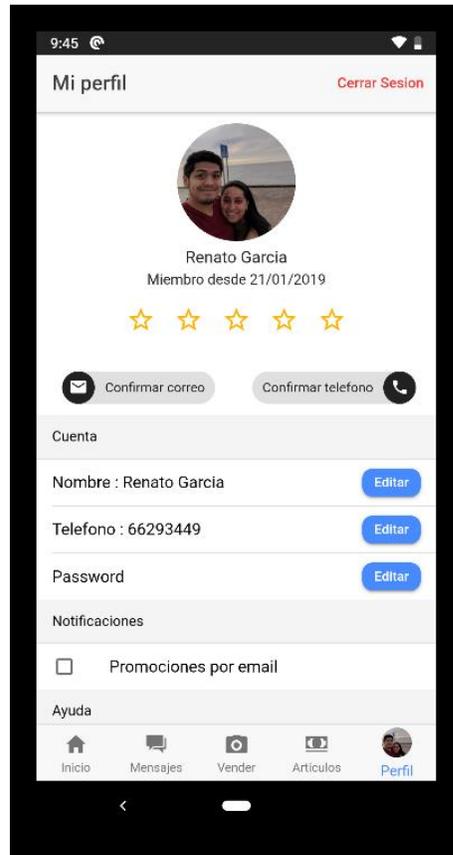
Un prototipo es un modelo experimental simple de una solución propuesta que se utiliza para probar o validar ideas, supuestos de diseño y otros aspectos de su conceptualización de forma rápida y económica, de modo que el diseñador o los diseñadores involucrados puedan realizar los ajustes necesarios o posibles cambios de dirección.

Figura 4. **Lista de artículos**



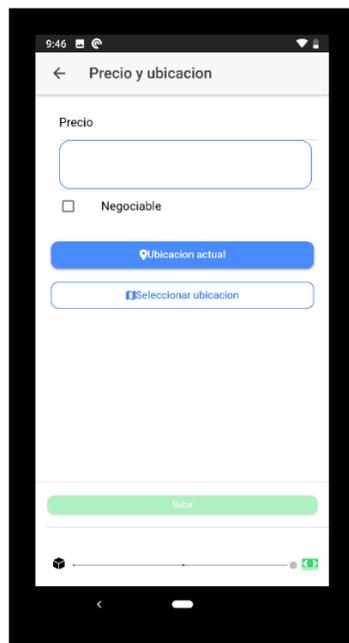
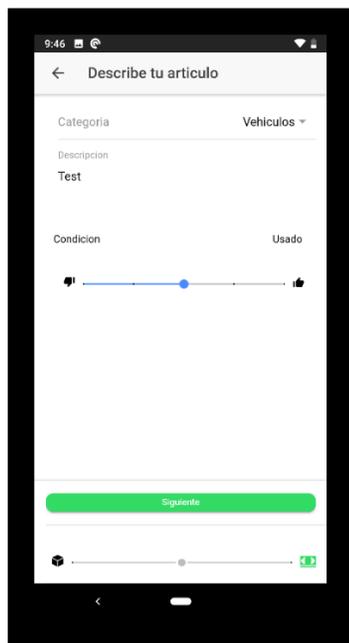
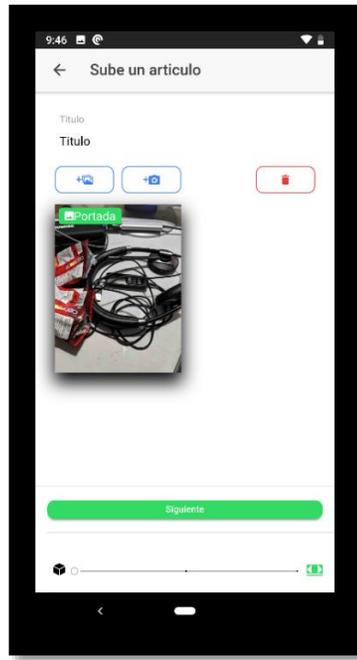
Fuente: elaboración propia, empleando Ionic Sketch Creator.

Figura 5. Perfil



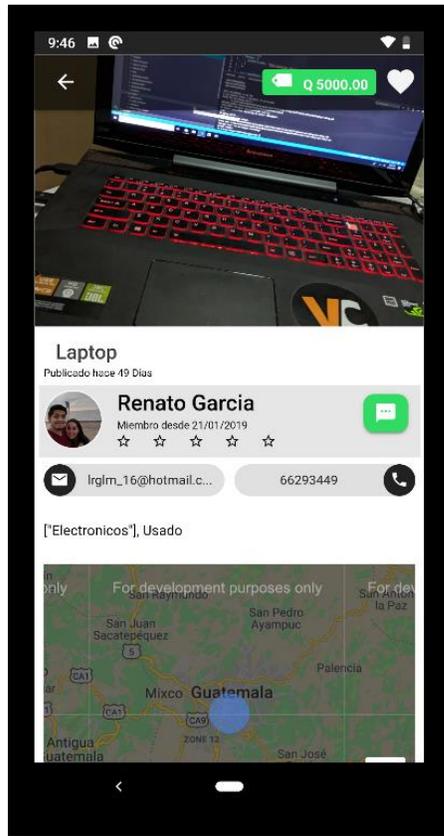
Fuente: elaboración propia, empleando Ionic Sketch Creator.

Figura 6. Crear artículo



Fuente: elaboración propia, empleando Ionic Sketch Creator.

Figura 7. Vista artículo



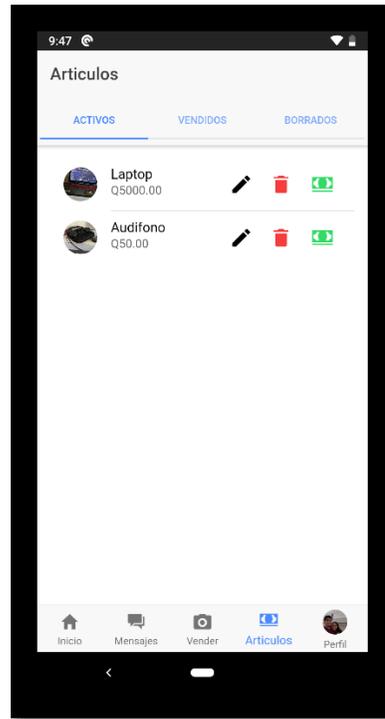
Fuente: elaboración propia, empleando Ionic Sketch Creator.

Figura 8. **Mensajes**



Fuente: elaboración propia, empleando Ionic Sketch Creator.

Figura 9. **Mis artículos**



Fuente: elaboración propia, empleando Ionic Sketch Creator.

3.7. **Política de seguridad**

Esta política detalla cómo se usarán los datos de los usuarios cuando accedan a esta plataforma o cuando interactúa con sus servicios. Si esta política cambia o se actualiza, habrá avisos en la plataforma para informar al usuario y requerir su consentimiento si es necesario.

- Tomamos precauciones para evitar el acceso no autorizado o el uso indebido de sus datos.

- No nos hacemos responsables por los anuncios publicados por nuestros usuarios.
- No compartimos sus datos con terceros con fines de publicidad.
- No nos involucramos en programas de publicidad cruzado o enlaces de referencia.
- No empleamos estrategias de seguimiento de dispositivos para fines de publicidad.
- No le enviamos comunicaciones no solicitadas con fines de publicidad.
- No nos involucramos en publicidad de afiliación

3.8. Política de privacidad

A continuación, se muestra una lista de la información que guardaremos y/o utilizaremos para mejorar la plataforma y proveer los servicios.

- Los datos que se publican o envía a través de RemataloGT, y / o nos envía directamente a través de otros sitios
- Los datos que se envíen o proporcionen en formularios de la plataforma (por ejemplo, nombre, dirección, correo electrónico, teléfono, fax, fotos, identificación fiscal).
- Datos de registro web (por ejemplo, páginas web visitadas, tiempos de acceso, dirección IP, encabezados HTTP).
- Datos recopilados a través de cookies (por ejemplo, datos de búsqueda y listas de "favoritos").
- Datos sobre sus dispositivos (por ejemplo, tamaño de pantalla, almacenamiento local DOM, complementos).
- Datos obtenidos por medio de terceros (por ejemplo, tipo de teléfono, ubicación geográfica a través de la dirección IP).

De la misma forma conservamos los datos según sea necesario para nuestros fines comerciales y hacemos esfuerzos de buena fe para almacenar los datos de forma segura, pero no podemos ofrecer ningún tipo de garantías. Los usuarios de RemataloGT pueden acceder y actualizar ciertos datos a través del inicio de sesión de su cuenta en la sección del perfil.

3.9. Política de confidencialidad de datos

A continuación, se muestra una lista de las circunstancias en las que podemos divulgar datos sobre los usuarios de RemataloGT:

- A proveedores y proveedores de servicios (por ejemplo, procesadores de pagos) que trabajan en nuestro nombre.
- Para responder a citaciones, órdenes de registro, órdenes judiciales u otros procesos legales.
- Para proteger nuestros derechos, propiedad o seguridad, o los de los usuarios de RemataloGT o el público en general.
- Cualquier otra situación con su consentimiento (por ejemplo, si nos autoriza a compartir datos con otros usuarios).
- En relación con una fusión, quiebra o venta / transferencia de activos.
- En forma de resumen o reporte, donde no se puede usar información razonablemente para identificarlo.

3.10. Leyes de protección y regulación de datos

Son un conjunto de leyes diseñadas para mejorar y asegurar la privacidad de datos. El énfasis de estas leyes está en proteger a los ciudadanos y sus datos, y brindarles a los usuarios más información y control sobre su uso.

De igual forma buscan proteger la información personal de los ciudadanos regulando cómo se almacenan, administran y procesan estos datos. Los datos personales se relacionan con cualquier información que haga que una persona pueda ser identificar. Esto incluye una amplia gama de variables tales como; nombres, información de contacto, ubicación de la persona, datos bancarios, información médica, imágenes y mucho más. Bajo esta información, la definición de datos personales también incluye identificadores en línea como direcciones IP y cookies.

Así mismo, buscan que los consumidores reciban aún más poder y control sobre estos datos. Tendrán que dar su consentimiento para el uso de estos datos y tener la capacidad de solicitar ver los datos que cualquier organización tiene sobre ellos. Los consumidores también obtendrán un "derecho de borrado", lo que les permite solicitar que se elimine la información sobre ellos.

Para las empresas, la implementación de este tipo de leyes significa mejorar los niveles de seguridad y protección sobre los datos que poseen. Las organizaciones también deberán ser más francas al recopilar esta información personal. Los recolectores deberán asegurarse de que se otorga el consentimiento de manera explícita y divulgar información sobre el propósito de la recopilación de datos.

3.10.1. Leyes de protección y regulación de datos en Guatemala

Actualmente, Guatemala carece de una ley única e integral que regule la recopilación y el uso de la información personal. Únicamente existe la “Iniciativa 4090-2009 Ley de Protección de Datos Personales” en la que se consideran muchos de los aspectos necesarios respecto a la protección de datos personales.

Esta iniciativa de leyes afecta a las organizaciones de todos los niveles Sea cual sea su negocio, si tiene clientes o empleados, tiene datos que requieren protección bajo estas normativas. Dichas normativas están destinadas a garantizar que se tomen las precauciones adecuadas para proteger a las posibles víctimas de delitos digitales, como el fraude o el robo de identidad derivado de actores malintencionados que obtienen acceso a los datos a través de piratería, fallas técnicas o errores humanos.

De igual forma dentro del código penal de Guatemala Decreto No. 17-73 Capítulo VII “De los delitos contra el derecho de autor, la propiedad industrial y delitos informáticos” se tiene contemplado lo siguiente:

MANIPULACIÓN DE INFORMACIÓN

ARTICULO 274 "E".

Se impondrá prisión de uno a cinco años y multa de quinientos a tres mil quetzales, al que utilizare registros informáticos o programas de computación para ocultar, alterar o distorsionar información requerida para una actividad comercial, para el cumplimiento de una obligación respecto al Estado o para ocultar, falsear o alterar los estados contables o la situación patrimonial de una persona física o jurídica.

USO DE INFORMACIÓN

ARTICULO 274 "F".

Se impondrá prisión de seis meses a dos años, y multa de doscientos a mil quetzales al que, sin autorización, utilizare los registros informáticos de otro, o ingresare, por cualquier medio, a su banco de datos o archivos electrónicos.³

Es importante tener en cuenta antes de comenzar cualquier discusión sobre el cumplimiento de estas leyes, que seguir las reglas no garantiza que sus sistemas y datos se mantendrán seguros. Cumplir con los estándares mínimos de la ley debe considerarse como un punto de partida. El lugar a donde llegue

³ Código Penal de Guatemala Decreto No. 17-73 Capítulo VII. *Violación a los derechos de autor y derechos conexos*. <https://www.oas.org/es/sla/ddi/docs/G6%20Codigo%20Penal%20de%20Guatemala.pdf>. Consulta: 8 de febrero de 2019.

las medidas de seguridad dependerá del tipo industria, los tipos de datos que maneje la organización y el apetito por el riesgo.

3.10.2. Consecuencias de la falta de leyes

Debido a la falta de regularización en cuanto a manejo de información personal se refiere, las empresas deben ser muy cuidadosas al implementar sus arquitecturas de software ya que con la falta de regularización y estándares se corre mucho riesgo de incurrir en faltas.

3.10.2.1. Demandas

La violación de datos no solo afecta a la organización violada, sino que también puede poner en riesgo a los empleados, consumidores, clientes, socios y proveedores de servicios asociados, cualquiera de los cuales puede decidir emprender acciones legales en busca de justicia y protección. Gane o pierda, una demanda puede ser una propuesta costosa.

3.10.2.2. Multas

Si la empresa maneja datos de tarjetas de crédito y los datos se ven afectados, los bancos pueden terminar reemitiendo nuevas tarjetas a sus clientes. Cuando eso ocurra y los bancos incurran en costos asociados, probablemente tratarán de recuperar esos costos de la organización cuyo incumplimiento motivó la acción, mediante la imposición de multas o tarifas adicionales.

3.10.2.3. Gastos de compensación y remediación

Entre los muchos costos involucrados con una falla de seguridad están los asociados con las investigaciones forenses para determinar la fuente y la causa de la violación, corregir las brechas que fueron explotadas y abordar cualquier riesgo residual para los consumidores y otros.

3.10.2.4. Pérdida de ingresos

Una violación de datos puede hacer que los clientes pierdan la confianza en la organización afectada. Cuando eso suceda, es muy probable que lleven su negocio a otra parte. Tras una violación que afectó a 70 millones de clientes en 2014, el gigante minorista Target vio caer sus ganancias por 440 millones de dólares en el siguiente trimestre.

3.10.2.5. Pérdida de reputación

Cuando la noticia de una violación de datos sale, pronto se pierde la reputación. Con el fin de reparar las cercas con todas las partes afectadas, las organizaciones incurrirán en costos asociados con el aumento de campañas de marketing, comunicaciones y relaciones públicas.

4. DESARROLLO DE LA APLICACIÓN

4.1. Herramientas

Para el desarrollo de esta aplicación se utilizaron varias herramientas de acceso abierto, las cuales facilitaron y redujeron el tiempo de desarrollo de una forma drástica, para lo cual se investigó de manera profunda las ventajas y desventajas de cada una.

4.1.1. Visual studio code

Visual Studio Code es un editor de código fuente muy ligero, pero con mucho potencial que se ejecuta como una aplicación escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un ecosistema con muchas extensiones para otros lenguajes de programación. Esta herramienta fue anunciada en abril 29 de 2015 por Microsoft y fue lanzada al público en noviembre 18 de 2015 bajo licencia de código abierto.

4.1.2. Xcode

Xcode es IDE que reúne todas las herramientas necesarias para producir una aplicación para macOS e iOS (especialmente un editor de texto, un compilador y un sistema de compilación) en un paquete de software en lugar de dejarlos como un conjunto de herramientas individuales conectadas mediante scripts. Xcode es el IDE oficial de Apple para desarrolladores de Mac y iOS; originalmente se conocía como Project Builder y se le cambió el nombre a Xcode.

En la versión 4, Apple agrego el programa de construcción de interfaces, de modo que solo había un paquete de aplicaciones; el diseño del programa no ha cambiado mucho desde entonces, aunque obviamente las herramientas en este se actualizan regularmente.

4.1.3. ADT (Android developer tools)

ADT es el conjunto de herramientas necesarios para desarrollar aplicaciones para el sistema operativo Android. Es provisto por Google bajo licencia de software libre. Cada herramienta puede pertenecer al SDK (software development kit) o a la plataforma para la que se está desarrollando. Dentro del SDK se encuentra la herramienta Android SDK manager, que es quien maneja las versiones de sistema operativo disponible para el desarrollo. Incluye el manejador de dispositivos virtuales y el emulador, los cuales son sumamente importantes para realizar pruebas conforme se desarrolla la aplicación. Otra herramienta sumamente importante es el ADB (Android debug bridge) la cual permite cargar aplicaciones a dispositivos físicos y depurar el desarrollo y ejecución de las mismas en dispositivos físicos en vez de emulados.

5. MODELO DE SOSTENIBILIDAD

5.1. Benchmarks

En la siguiente sección se mostrará una lista de aplicaciones actualmente disponibles, que cuentan con funcionalidad similar a RemataloGT que han tenido una gran influencia en sus mercados.

5.1.1. Craigslist

Es una plataforma de mercado local en línea para compradores, vendedores, empleadores y personas que buscan empleo. Es enorme y ha tenido un gran crecimiento por su facilidad de uso y alta disponibilidad. Si un usuario vive en (o cerca de) una ciudad con su propia página de Craigslist o su empresa ofrece productos y servicios con alcance nacional, el sitio puede aumentar la presencia en línea de su empresa, la visibilidad y, por supuesto, los ingresos. Cualquier propietario de una empresa puede visitar la plataforma y hacer uso de todas las herramientas de ella de forma gratuita.

- País: Estados Unidos
- Sitio web: <https://www.craigslist.org/about/sites>

5.1.2. OfferUp

La plataforma OfferUp, que se ofrece para iOS, Android y la web, facilita la compra y venta de productos nuevos y/o usados. Esta plataforma, que ha sido optimizada para teléfonos inteligentes, presenta fotos de productos a la venta a

través de una interfaz de desplazamiento infinito visualmente atractiva. Para los compradores, los productos se presentan en función de la ubicación geográfica más cercana al comprador para ayudar a fomentar las interacciones de usuario cara a cara y fomentar las relaciones centradas en la comunidad, pero también pueden filtrarse por categoría, precio y distancia. En el lado del vendedor, los productos se enumeran inmediatamente cargando una foto desde el teléfono inteligente o la computadora de un usuario; la plataforma facilita la clasificación de las consultas entrantes por la oferta más alta antes de contactar a los posibles compradores para acordar el precio y el lugar de reunión.

- País: Estados Unidos
- Sitio web:
https://play.google.com/store/apps/details?id=com.offerup&hl=es_GT

5.2. Benchmarks en Latinoamérica

A continuación, se muestra una aplicación que ha crecido considerablemente desde su lanzamiento, y la cual tiene básicamente el mismo modelo de sostenibilidad que RemataloGT.

5.2.1. Vendiste.com

La app vendiste.com que se ofrece para iOS, Android y la web facilita la compra y venta de productos usados. Esta app, que ha sido optimizada para teléfonos inteligentes, presenta fotos de productos a la venta a través de una interfaz de desplazamiento infinito visualmente atractiva. Para los compradores, los productos se presentan en función de la ubicación geográfica más cercana, esto facilita la búsqueda y la hace mucho más personalizada. Los vendedores pueden enlistar en pocos pasos subiendo una foto del producto desde su

dispositivo móvil. La app tiene ventajas de búsquedas por precios, categorías, ofertas, perímetros, entre otros.

Durante los casi 12 meses de operación de Vendiste se han completado más de 3.000 transacciones, que suman más de 450.000 dólares. El precio promedio de los artículos ofrecidos en esta *app* es de 150 dólares.

Vendiste.com ya hace presencia en países como Uruguay, Argentina, Paraguay, Brasil, Costa Rica, y Puerto Rico, pero es en el mercado local en donde quieren consolidarse.

- País: Colombia
- Sitio web:
<https://play.google.com/store/apps/details?id=com.vendiste.app>

Colombia, al igual que Guatemala, ha tenido un crecimiento enorme en cuanto al acceso a internet y el uso de teléfonos inteligentes, por lo cual es un buen ejemplo acerca del modelo de sostenibilidad y la capacidad de crecimiento de una plataforma como RemataloGT.

5.3. Capas gratuitas de AWS

Un aspecto importante de esta investigación es demostrar cómo utilizando servicios en la nube se puede reducir grandemente el tiempo que toma llevar una idea a su implementación completa. Al utilizar servicios en la nube se remueve la restricción de necesitar una infraestructura física para implementar las topologías, ya que únicamente nos preocupamos en configurar y manejar los servicios disponibles dentro de nuestro proveedor de servicios en la nube y, de esta forma, aprovechar todas las facilidades que existen en el mercado. Un

ejemplo son las capas gratuitas de AWS que permite utilizar muchos de sus servicios sin ningún costo. Como parte de la capa gratuita de AWS, se obtienen los siguientes servicios / productos de AWS por mes sin costos:

- 750 horas de uso de la instancia t2.micro de Amazon EC2 Linux o RHEL o SLES (1 GiB de memoria y soporte de plataforma de 32 y 64 bits): suficientes horas para ejecutar continuamente cada mes.
- 750 horas de uso de la instancia t2.micro del servidor Microsoft Windows de Amazon EC2 (1 GiB de memoria y compatibilidad con plataformas de 32 y 64 bits): suficientes horas para ejecutar continuamente cada mes.
- 750 horas de un Elastic Load Balancer más 15 GB de procesamiento de datos.
- 750 horas de instancias de base de datos de Amazon RDS Single-AZ Micro, ejecutando MySQL, PostgreSQL, Oracle BYOL o SQL Server Express Edition: suficientes horas para ejecutar una instancia de base de datos continuamente cada mes. También obtiene 20 GB de almacenamiento de base de datos, 10 millones de Entrada y Salida y 20 GB de almacenamiento de respaldo.
- 750 horas de uso del Nodo de Micro Caché de Amazon ElastiCache: suficientes horas para ejecutar continuamente cada mes.
- 30 GB de almacenamiento de bloque elástico de Amazon en cualquier combinación de uso general (SSD) o magnético, más 2 millones de Escritura y Lectura (con EBS magnético) y 1 GB de almacenamiento de instantáneas.

- 5 GB de almacenamiento estándar de Amazon S3, 20,000 solicitudes de lectura y 2,000 solicitudes de creación.
- 25 GB de almacenamiento, 25 unidades de capacidad de lectura y 25 unidades de capacidad de escritura, suficientes para manejar hasta 200 millones de solicitudes por mes con Amazon DynamoDB.
- 25 horas de máquina de Amazon SimpleDB y 1 GB de almacenamiento.
- 1.000 ejecuciones de flujo de trabajo de Amazon SWF se pueden iniciar de forma gratuita. Un total de 10,000 tareas de actividad, señales, temporizadores y marcadores, y 30,000 días de flujo de trabajo también se pueden utilizar de forma gratuita.
- 100,000 solicitudes de Amazon Simple Queue Service.
- 100,000 solicitudes, 100,000 notificaciones HTTP y 1,000 notificaciones por correo electrónico para Amazon Simple Notification Service.
- 10 métricas de Amazon Cloudwatch, 10 alarmas y 1,000,000 solicitudes de API.
- Transferencia de datos de 50 GB, 2,000,000 de solicitudes HTTP y HTTPS para Amazon CloudFront.
- 15 GB de ancho de banda agregados en todos los servicios de AWS.

5.3.1. Cuando expira una cuenta de capa gratuita de AWS

Las capas gratuitas de AWS vencerán 12 meses después de la fecha de registro. Cuando vence su uso gratuito o si el uso de su aplicación excede los niveles de uso gratuito, simplemente se pagan las tarifas de servicio estándar de pago por uso. Hay ofertas que no tienen vencimiento al final de su período de Nivel Gratuito de AWS de 12 meses y están disponibles para todos los clientes de AWS.

5.3.2. Servicios sin costo de AWS

Hay varios servicios de AWS que no tienen un cargo adicional. No hay tarifas mínimas ni pagos por adelantado. Estos servicios de AWS son:

- Nube privada virtual (VPC)
- Escalado automático
- Elastic Beanstalk
- CloudFormation
- Gestión de Identidad y Acceso (IAM)
-

5.4. Costos dentro de AWS

A continuación, se muestra una lista de los costos fijos que se incurrirán para mantener la plataforma en funcionamiento, en el cual se define la capa gratuita que ofrece AWS y el costo normal.

Tabla I. **Costos dentro de AWS**

| Servicio | Capacidad capa gratuita | Capa Gratuita(Q) | Costo Fuera de Capa Gratuita |
|-----------------------|--------------------------------|-------------------------|--|
| EC2 | 750 Horas | 0,00 | 65,00 Q al mes |
| API Gateway | 1,000,000 solicitudes | de 0,00 | 28,00 Q por cada millón de solicitudes |
| Lambda | 1,000,000 solicitudes | de 0,00 | 1,60 Q por cada millón de solicitudes |
| RDS PostgreSQL | 750 Horas | 0,00 | 110,00 Q al mes |
| S3 | 5Gb Almacenamiento | de 0,00 | 4,00 Q por GB almacenado |
| Total | | 0,00 | Será directamente proporcional al uso |

Fuente: elaboración propia.

De esta forma, nos damos cuenta de que es posible lanzar un producto inicial sin necesidad de incurrir en riesgos de inversión, esto sin considerar los costos de desarrollo. No obstante, como bien se mencionó, esta topología está enfocada en ser viable para emprendedores, los cuales muy probablemente cuentan ya con todas las herramientas necesarias para desarrollar sus aplicaciones.

5.5. Capacidades de crecimiento y posibles fuentes de ingreso

Hasta este momento no se ha definido cómo RemataloGT generará ingresos, lo cual es muy difícil de definir sin saber cuánto uso se le dará a la plataforma. En teoría, RemataloGT no cobrará ninguna tarifa y esa visión es casi correcta. El sitio cobra podrá cobrar tarifas solo por algunas secciones clasificadas, mientras que las publicaciones en la mayoría de las secciones

seguirán siendo gratuitas. De igual forma, si la base de usuarios aumenta, RemataloGT podría aumentar sus ingresos al agregar elementos como anuncios publicitarios. Sin embargo, en un inicio no es necesario implementar esto debido a que los usuarios deberán usar la plataforma antes de agregar dichos elementos.

CONCLUSIONES

1. Las hipótesis de las 3 teorías económicas expuestas en este documento (teoría económica, del aprendizaje y de la escuela sociológica) son determinantes al momento de la construcción y diseño de una aplicación, ya se debe definir las razones por las cuales los usuarios harán uso del sistema, sobre todo en la época actual cuando las aplicaciones móviles se utilizan en todos los campos de la vida diaria.
2. Es importante identificar los problemas y necesidades de los sectores económicos actuales del país, para brindar soluciones adaptables y que procuren resolver total o parcialmente dichas necesidades. RemataloGT se enfoca en resolver las necesidades del sector económico de las Mypymes y apoyar al crecimiento de las mismas.
3. RemataloGT ofrecerá un canal colaborativo gratuito para cualquier tipo de empresa y persona individual, lo cual incentiva y ayuda a el crecimiento del e-commerce en el país.
4. Las capas gratuitas de los servicios web de Amazon hacen que cualquier emprendedor pueda llevar una idea al mercado fácilmente, sin incurrir en grandes gastos y sin altos factores de riesgo.
5. El uso de software de código abierto ofrece varias ventajas, como la posibilidad de reducir los costos y el tiempo de desarrollo, o de evitar depender de un solo proveedor. Por lo tanto, es de esperar que cada vez

más empresas e instituciones comiencen a utilizar software de código abierto.

RECOMENDACIONES

1. Al momento de desarrollar una aplicación móvil, si se utilizara computación en la nube, se recomienda la utilización de los servicios web de Amazon, debido a su flexibilidad en los cargos por servicio, así como también las capas gratuitas que permitirán llevar una idea al mercado de forma rápida y sin riesgos altos.
2. Debido a la dificultad de mantener varias versiones de código para una misma aplicación, se recomienda desarrollar aplicaciones híbridas para mantener y desarrollar una única versión de código. Esto facilitará la depuración e implementación de los requerimientos, con cualquiera de los diferentes *frameworks* disponibles actualmente.
3. Adquirir un dominio público para crear una página corporativa de la aplicación para hacer disponibles los términos y condiciones de la misma, así como para tener presencia en la web.
4. Contactar al gremial de comercio electrónico de Guatemala para realizar un lanzamiento y presentación de la aplicación en algún evento de dicho gremial.

BIBLIOGRAFÍA

1. Diario de Centro América. [En línea]. <<https://dca.gob.gt/noticias-guatemala-diario-centro-america/>>. [Consulta: 12 de febrero de 2019].
2. Google. [En línea]. <<https://www.thinkwithgoogle.com/intl/es-419/insights/guatemala-las-claves-de-la-digitalizacion/>>. [Consulta: 12 de febrero de 2019].
3. Ministerio de Economía de Guatemala. [En línea] <<https://www.mineco.gob.gt/>>. [Consulta: 7 de febrero de 2019].
4. Ionic Community. Ionic Framework. [En línea] <<https://ionicframework.com/docs>>. [Consulta: 12 de febrero de 2019].
5. Typescript. [En línea]. <<https://www.typescriptlang.org/docs/home.html>>. [Consulta: 12 de febrero de 2019].
6. Documentación de Angular. [En línea]. <<https://angular.io/docs>>. [Consulta: 12 de febrero de 2019].
7. Documentación de NPM. [En línea]. <<https://www.npmjs.com/>>. [Consulta: 12 de febrero de 2019].

8. Documentación de NodeJS. [En línea]. <<https://nodejs.org/es/>>. [Consulta: 12 de febrero de 2019].
9. Documentación de PostgreSQL. [En línea]. <<https://www.postgresql.org/docs/>>. [Consulta: 12 de febrero de 2019].
10. Documentación de AWS Lambda. [En línea]. <https://docs.aws.amazon.com/es_es/lambda/latest/dg/welcome.html>. [Consulta: 12 de febrero de 2019].
11. Documentación de AWS RDS. [En línea]. <https://docs.aws.amazon.com/rds/index.html#lang/es_es>. [Consulta: 12 de febrero de 2019].
12. The Apache Software Foundation. Apache Cordova. [En línea]. <<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>. [Consulta: 10 de septiembre de 2017].
13. Documentación de Google Maps Platform. [En línea]. <<https://developers.google.com/maps/documentation/?hl=es>>. [Consulta: 12 de febrero de 2019].
14. Teorías del Comportamiento del Consumidor Completo. [En línea]. <http://www.academia.edu/6109474/TEORIAS_DEL_COMPORTAMIENTO_DEL_CONSUMIDOR_COMPLETO>. [Consulta: 12 de febrero de 2019].