

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**PLATAFORMA PARA LA ADQUISICIÓN Y OFERTA DE SERVICIOS
TEMPORALES**

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Computación

Presentado por:

Cristian Fernando Peña Espinoza

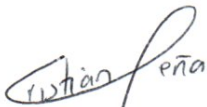
Wilson Israel Plascencia Jordán

GUAYAQUIL - ECUADOR

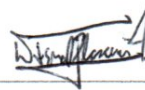
Año: 2019

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Cristian Fernando Peña Espinoza* y *Wilson Israel Plascencia Jordán* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Cristian Fernando Peña
Espinoza



Wilson Israel Plascencia
Jordán

RESUMEN

El presente proyecto integrador tiene como objetivo el desarrollo de un prototipo 100% funcional de una aplicación móvil como solución a la problemática de la adquisición y oferta de servicios temporales. Utilizando el framework Scrum, para el desarrollo ágil se llevó a cabo la implementación del sistema que consta de dos partes: Una plataforma web administrativa donde se muestra el flujo de los datos conforme la aplicación sea usada y así supervisar el contenido de estos que, además, podrán ser reflejados mediante reportes. Una aplicación móvil, desarrollada con "NativeScript", herramienta que posibilitó la obtención de la aplicación en iOS y Android. El fin de la aplicación es permitir la comunicación entre el adquiriente de servicios que en la actualidad no conoce con exactitud dónde hallar a una persona que satisfaga sus necesidades de reparación o elaboración de alguna obra o trabajo, y al oferente que, por otro lado, no tiene un trabajo estable y se ve obligado a salir a las calles en busca de alguien que requiera de su experiencia. Todo esto mediante una búsqueda por palabras relacionadas al servicio requerido en la pantalla principal de la aplicación y, en el caso del oferente, realizando una postulación que será revisada por el administrador del sistema para su aceptación, y de esa manera empiece a brindar sus servicios. Finalmente tenemos que, el trabajo realizado ha despertado gran interés en el público en general debido a la innovación de un método que no ha tenido cambios durante un largo tiempo, brindando una importancia significativa al campo estudiado y a la estructura tecnológica que lo soporta. Por lo que se tiene una buena solución a la problemática y que, con algo de adaptación por parte de personas que no manejan teléfonos inteligentes, les llegará a ser de gran ayuda.

Palabras clave: Framework, Oferentes, Adquirientes, NativeScript

ABSTRACT

This integrating project aims to develop a 100% functional prototype of a mobile application as a solution to the problem of the acquisition and offer of temporary services. Using the Scrum framework, for the agile development the implementation of the system was carried out consisting of two parts: An administrative web platform where the flow of the data is shown as the application is used and that way monitor the content of these that also, may be reflected through reports. A mobile application, developed with "NativeScript", a tool that made it possible to obtain the application on iOS and Android. The purpose of the application is to allow communication between the services acquirer that currently does not know exactly where to find a person who meets their needs for repair or preparation of any work, and the offeror who, on the other hand, He does not have a stable job and is forced to take to the streets in search of someone who needs his experience. All this by means of a search for words related to the service required on the main screen of the application and, in the case of the offeror, making an application that will be reviewed by the system administrator for acceptance, and thus begin to provide their services. Concluding, the work done has attracted great interest in the general public due to the innovation of a method that has not had changes for a long time, giving significant importance to the field studied and the technological structure that supports it. So, you have a good solution to the problem and with some adaptation by people who do not handle smartphones, will be very helpful.

Keywords: Framework, Offerer, Acquirer, NativeScript

ÍNDICE GENERAL

RESUMEN.....	I
<i>ABSTRACT</i>	II
ÍNDICE GENERAL	III
ÍNDICE DE FIGURAS	V
ÍNDICE DE TABLAS.....	VI
CAPÍTULO 1.....	1
1. INTRODUCCIÓN	1
1.1 Descripción del problema.....	2
1.2 Objetivos	2
1.2.1 Objetivo General.....	2
1.2.2 Objetivos Específicos	3
1.3 Marco teórico.....	3
CAPÍTULO 2.....	6
2. METODOLOGÍA	6
2.1 Plan de recolección de datos	6
2.2 Fiabilidad de datos	6
2.3 Análisis de datos, factores y causas	6
2.4 Propuesta de solución.....	8
2.5 Plan de implementación	11
CAPÍTULO 3.....	16
3. RESULTADOS Y ANÁLISIS.....	16
3.1 Requerimientos	16
3.2 Herramientas y Tecnología	17
3.2.1 Arquitectura	17

3.2.2	Servidor Web y Base de datos	17
3.2.3	Aplicación Móvil	18
3.3	Implementación	19
3.3.1	Plataforma Administrativa	19
3.3.2	Aplicación Móvil	20
3.3.3	Aplicación Web	25
3.3.4	Administrador Web	26
CAPÍTULO 4	27
4.	CONCLUSIONES Y RECOMENDACIONES	27
	Conclusiones.....	27
	Recomendaciones.....	28
BIBLIOGRAFÍA	29
APÉNDICES	31
Apéndice A:	31
Lista y detalle de Endpoints	31

ÍNDICE DE FIGURAS

Figura 2.1. Diagrama Entidad Relación	8
Figura 2.2. Diagrama de estado (Adquirir servicio)	9
Figura 2.3. Diagrama de componentes.....	9
Figura 2.4. Diagrama de Actividades (Adquirir servicio).....	10
Figura 2.5. Diagrama de casos de uso	11
Figura 2.6. Patrón de diseño MVC [9].....	13
Figura 2.7. API REST [9].....	14
Figura 3.1. API REST [9]	17
Figura 3.2. Creación de usuario.....	19
Figura 3.3. Pantalla Principal.....	23
Figura 3.4. Notificaciones.....	23
Figura 3.5. Perfil.....	23
Figura 3.6. Inicio de sesión	23
Figura 3.7. Verificación SMS.....	24
Figura 3.8. Enviar Requerimiento	24
Figura 3.9. Calificar servicio.....	24
Figura 3.10. Publicar Requerimiento	24
Figura 3.11. Página administrativa principal.....	26

ÍNDICE DE TABLAS

Tabla 1.1 Estadísticas de Empleo INEC [4].....	4
--	---

CAPÍTULO 1

1. INTRODUCCIÓN

El presente trabajo de grado hace referencia a los servicios temporales que se pueden definir como trabajos que, profesionales o técnicos realizan en un reducido lapso o son eventuales; comprenden actividades de reparación, instalación o elaboración de un producto.

Actualmente, el sistema que se viene aplicando perjudica a los dos agentes importantes de la economía: productor y consumidor. Por un lado, el productor, la persona que ofrece el servicio, a falta de un trabajo estable; debido a que muchas empresas, por amenorar costos en mano de obra y demás obligaciones laborales, no les es posible contratar a este pequeño sector productivo, sin embargo, según el artículo 33 de la constitución ecuatoriana: El trabajo es un deber y un derecho social. Todo aquel que garantice una digna existencia, respeto y un sueldo legítimo al trabajador, y además vele por el bienestar de él y su familia, gozará amparo por parte del estado...” por esta razón es que contamos con estos trabajadores que brindan sus actividades independientemente.

Por otro lado, la ley de defensa a los consumidores les otorga el derecho a estar informados con seriedad y veracidad sobre los bienes y servicios que desean adquirir, enfocándonos en el servicio, el usuario debe contar con un registro tanto del oferente y como del servicio que brinda, incluyendo la posibilidad de la valoración del desempeño de quienes realizan el trabajo contratado.

La evolución de las TIC, en cuanto a las comunicaciones móviles, que iniciaron con servicio de voz y mensajería sencilla, teniendo redes de datos en banda ancha más rápidas, han permitido que los teléfonos móviles, a más de ser usados para lo que fueron esencialmente creados, posean aplicaciones, que son básicamente programas que se instalan en los dispositivos móviles para uso personal o de negocios, convirtiendo así el teléfono móvil en un computador pequeño donde se puede realizar casi cualquier tarea diaria. Según la Agencia de Regulación y Control

de las telecomunicaciones “el 46.4% del servicio móvil avanzado” [1] usan teléfonos inteligentes

Tomando en cuenta estos factores, se analiza el planteamiento de una solución tecnológica que facilite la comunicación de oferentes y solicitantes de servicios, que guarde los derechos y necesidades de cada uno, y brinde seguridad en el proceso.

1.1 Descripción del problema

Se define como servicios temporales a aquellos servicios tales como: plomería, electricidad, pintura, etc. En Ecuador se acostumbra a encontrar este tipo de servicios a través de recomendaciones personales, o por el conocimiento de ciertos sectores en donde se concentran trabajadores que ofertan dichos servicios.

El método antes descrito se vuelve difícil a la hora de buscar a alguien que supla las necesidades especificadas. Por otra parte, existe la desconfianza de la calidad del trabajo a realizarse, la inseguridad personal a la hora de contratar, la desinformación acerca de los precios, y otros motivos.

El problema radica en que no se tiene un método específico para encontrar dichos servidores de una manera rápida y cómoda para los clientes, en donde se pueda comparar y discernir a quien se pretende contratar.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollo de un prototipo 100% funcional de una aplicación móvil para la adquisición y oferta de servicios temporales

1.2.2 Objetivos Específicos

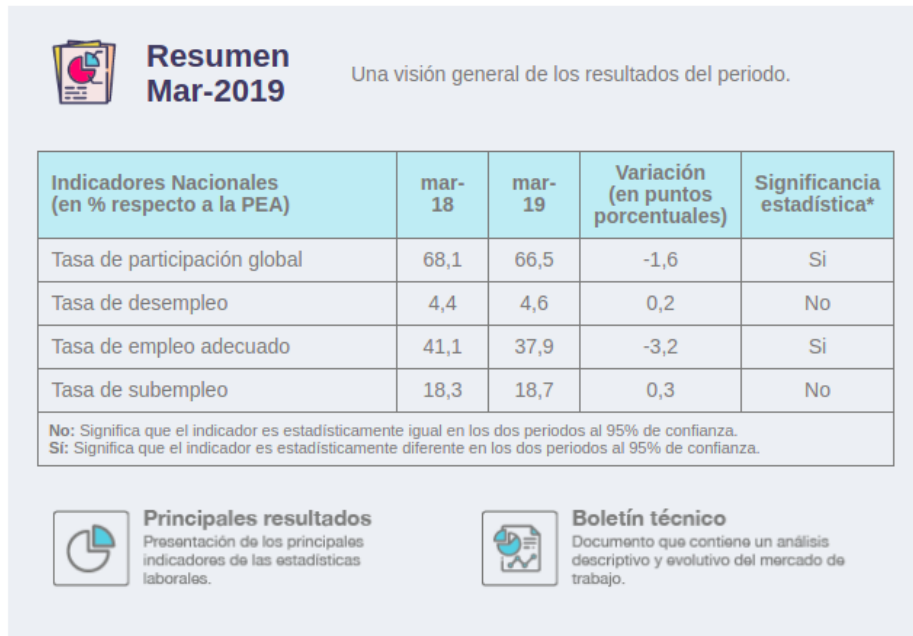
1. Implementar un aplicativo móvil, identificando los requerimientos de un oferente de servicios y de un adquiriente de los mismos.
2. Diseñar una arquitectura escalable para un SaaS.
3. Implementar un BackOffice para la administración de la plataforma.

1.3 Marco teórico

El desempleo es un factor preocupante para la clase media en nuestra sociedad. Una de las causas de desempleo es debido a que las personas tardan en encontrar un trabajo que se ajuste a sus aptitudes. Otra de las causas es la falta de plazas de trabajo adecuado en el mercado laboral añadiendo que el salario que perciben los trabajadores no se ajusta a lo que se necesita [2].

La economía en Ecuador, cuya moneda es el dólar desde hace 19 años, cuenta con una tasa del 4,6% de desempleo para marzo del 2019 (ver Tabla 1.1), según el Instituto Nacional de Estadísticas y Censos (INEC) el cual subdivide los indicadores nacionales de empleo en participación global, desempleo, empleo adecuado y subempleo [3].

Tabla 1.1 Estadísticas de Empleo INEC [4]



Información Estadística
Datos de la operación estadística y metadatos que permitan una interpretación de los resultados.

Existen varias aplicaciones que intentan cubrir esta problemática, pero no son dedicadas a ella en sí, y su interacción se hace compleja al momento de buscar y/o publicar un servicio. Una de ellas es **Olx**, la que te permite publicar anuncios de venta de artículos. Desde teléfonos celulares usados, electrodomésticos, automóviles, hasta departamentos o casas [5].

Como vemos, esta aplicación abarca varios tópicos y tiene un enfoque más basado en ventas. Otra aplicación es de origen mexicano llamada **Ayfix** [3] que muestra a usuarios de la app, los reparadores, prestadores de servicio de todo tipo, que se encuentran más cerca. En la app puedes contactar desde plomeros, personas dedicadas al cuidado del hogar, hasta manicuristas. Pero como lo mencionamos al principio, solo existe en México y no ofrece una buena organización.

Este tipo de aplicaciones generalmente utilizan un modelo de distribución SaaS que, en español, Software como servicio, se refiere a un modelo de distribución

de software en el cual se aplican modelos conceptuales como estructura multi-tenancy o multiempresa, el cual consolida toda la información en una plataforma web donde los clientes puede acceder en cualquier parte del mundo a través de internet y hacer uso de sus servicios [6].

Este modelo generalmente se monetiza por medio de valores mensuales o alquileres, los cuales son pagados por el cliente según su plan de preferencia. Existen otros modelos en los que la suscripción es gratuita y se monetizan por medio de comisiones, tales como Uber, Glovo, Airbnb, entre otros.

CAPÍTULO 2

2. METODOLOGÍA

2.1 Plan de recolección de datos

Se tuvo un acercamiento con potenciales usuarios oferentes (los que ofrecen servicios) de la aplicación, en donde se introdujo la idea de la solución y se recolectaron datos críticos para esta.

Se diseñaron mockups con el fin de realizar las respectivas validaciones con los usuarios nuevamente para conocer factibilidad, usabilidad y flexibilidad.

2.2 Fiabilidad de datos

Se demuestra que los datos son fiables con las entrevistas y validaciones efectuadas directamente con los usuarios principales como lo son los oferentes. Se realizaron también, con ayuda de los mockups, validaciones con los usuarios adquirientes de servicios.

2.3 Análisis de datos, factores y causas

Para el desarrollo de la solución usamos el framework Scrum, el cual es un marco simple para el desarrollo ágil de software, que tiene como finalidad entregar nuevas funcionalidades al final de cada iteración llamada “sprint” [7]. Comenzamos identificando los actores, los cuales son los siguientes:

- Administrador
- Oferente
- Adquiriente

Una vez identificados estos actores se procedió a realizar las historias de usuario, presentadas a continuación:

- Como adquiriente puedo:
 - Registrarme como usuario en la aplicación móvil.
 - Iniciar sesión en la aplicación móvil.
 - Recuperar Cuenta

- Busca oferentes en una barra de búsqueda
- Listar a los oferentes en la pantalla principal.
- Visualizar calificación de los oferentes.
- Ver detalles del oferente listado en la pantalla principal.
- Enviar detalles de mis necesidades al oferente.
- Recibir notificaciones de respuestas de los oferentes.
- Visualizar respuestas con precio de la oferta del oferente.
- Aceptar propuesta del adquiriente.
- Visualizar datos para el contacto con el adquiriente.
- Enviar mis requerimientos a varios oferentes a la vez.
- Llenar datos restantes para poder postular como oferente.
- Modificar datos de mi perfil.
- Calificar el servicio adquirido.
- Como oferente puedo:
 - Registrarme como usuario en la aplicación móvil.
 - Enviar la foto de mi cédula para que sea aprobada por el administrador.
 - Iniciar sesión en la aplicación móvil.
 - Postular una descripción y portafolio de mis servicios en la aplicación móvil.
 - Recibir notificaciones de requerimientos de servicios del adquiriente.
 - Visualizar los requerimientos del adquiriente.
 - Enviar breve descripción con precio de los requerimientos del adquiriente.
 - Realizar todas las actividades tales como un adquiriente.
- Como administrador puedo:
 - Iniciar sesión en el sitio administrativo web.
 - Registrar usuarios como administradores, adquirientes u oferentes
 - Visualizar cédula de identidad de los oferentes
 - Aceptar la solicitud de registro del oferente

- Visualizar ultimo uso de la aplicación de los usuarios
- Administrar usuarios
- Administrar postulaciones de oferentes
- Administrar ubicaciones como país, provincia y ciudad
- Descargar reportes en Excel

2.4 Propuesta de solución

Para el diseño de la arquitectura se utilizó el modelo 4+1 de Kruchten, en él se definen múltiples vistas concurrentes para las cuales tenemos:

Vista Lógica: se define la estructura y funcionalidad del sistema para el cual se ha escogido dos modelos UML: Diagrama Entidad Relación (ver Figura 2.1) y Diagrama de estado para el sistema (ver Figura 2.2).

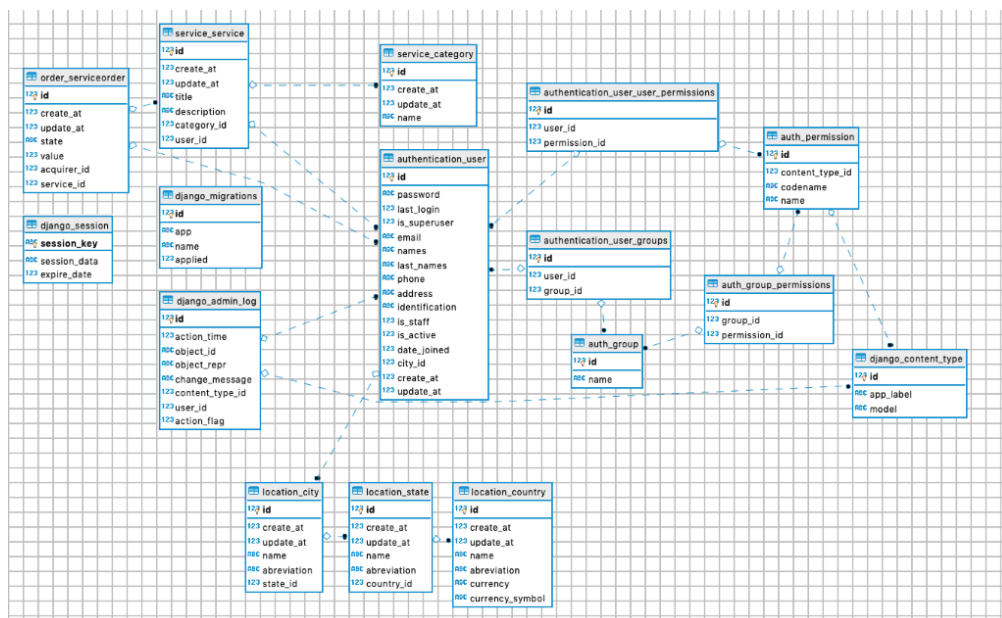


Figura 2.1. Diagrama Entidad Relación

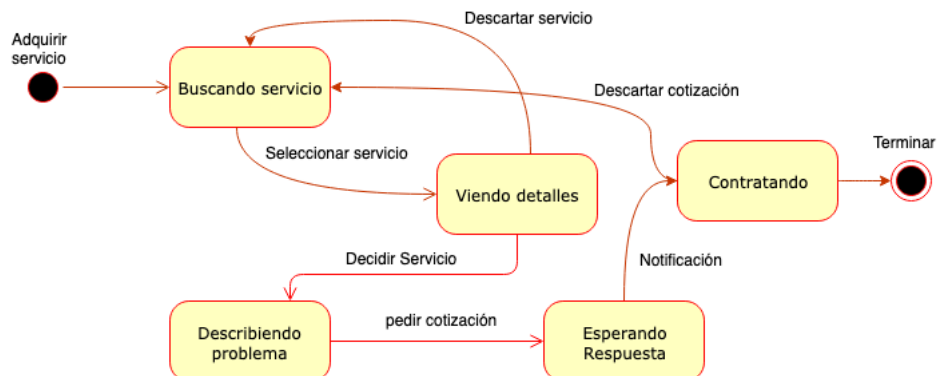


Figura 2.2. Diagrama de estado (Adquirir servicio)

Vista de desarrollo: se describen los elementos del software, se utiliza para dar un enfoque al programador y una perspectiva del sistema a desarrollar. Para este modelo se seleccionó el Diagrama de Componentes de UML. (ver Figura 2.3)

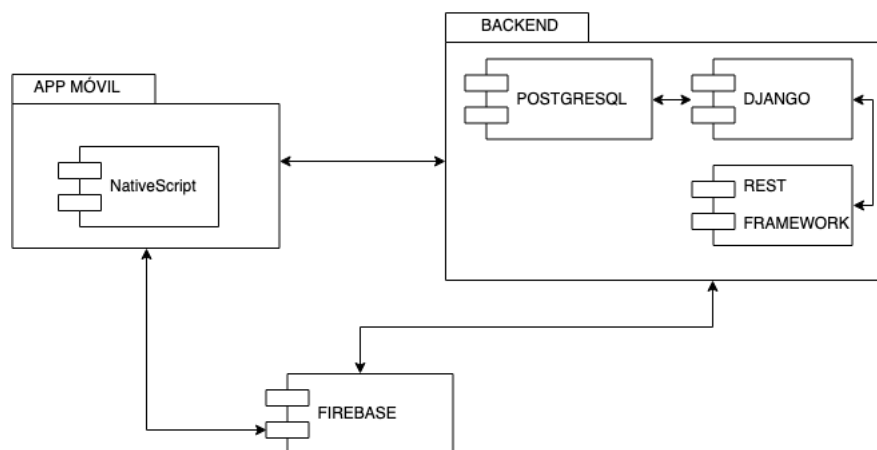


Figura 2.3. Diagrama de componentes

Vista de procesos: Se describen los procesos a llevar a cabo en el sistema, en el cual se muestra la interacción del proceso y como se comunica. Para esta vista se seleccionó el Diagrama de Actividades. (ver Figura 2.4)

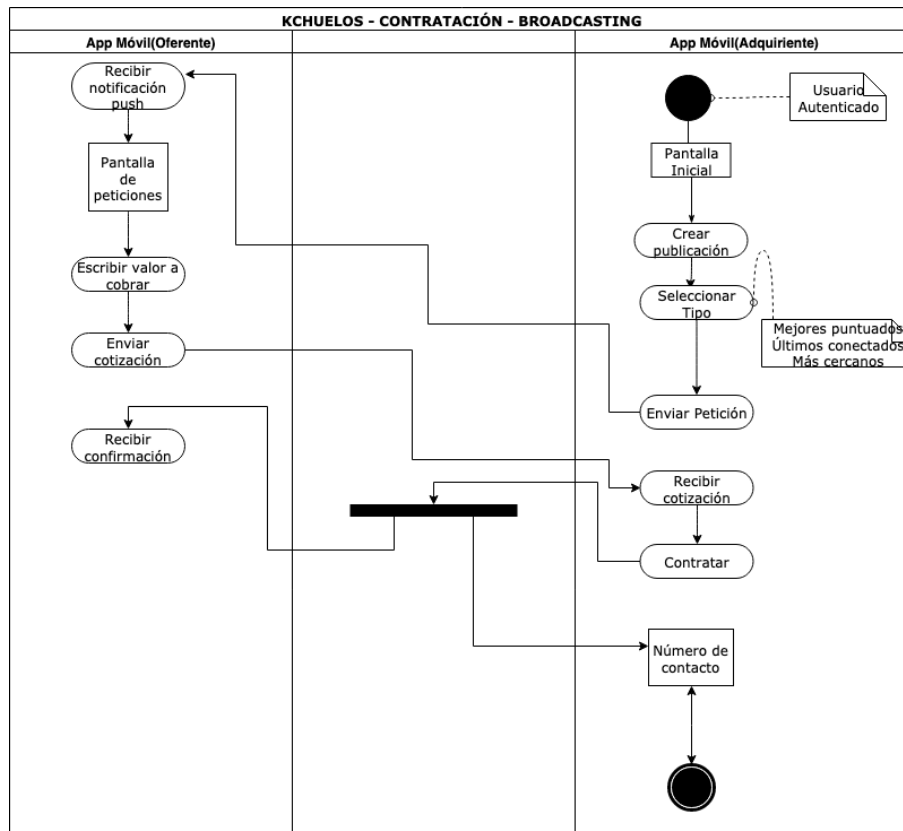


Figura 2.4. Diagrama de Actividades (Adquirir servicio)

Escenarios: se describen los casos de uso para el sistema a desarrollar para tener una vista general de las actividades a realizar. Para está vista se utilizó el Diagrama de Casos de Uso (ver Figura 2.5).

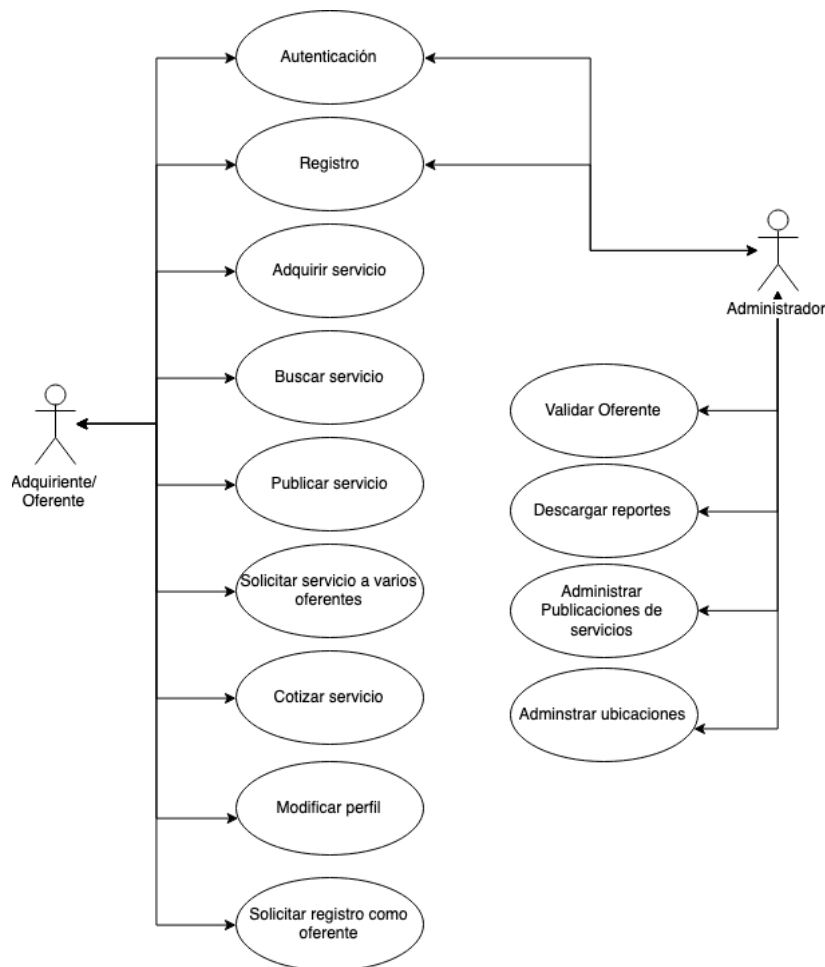


Figura 2.5. Diagrama de casos de uso

2.5 Plan de implementación

Para la implementación de la solución se utilizará el modelo de distribución SaaS, el cual va a ser desarrollado mediante tecnologías Web y móviles en donde los clientes y la parte administrativa pueden interactuar con libertad.

La solución consta de dos partes:

- Aplicación Móvil: la cual va a ser usada tanto por los oferentes de servicios como los adquirientes.
- Aplicación Web: va a ser utilizada especialmente por los administradores del sitio.

Para la arquitectura de la aplicación en general se hará una separación en dos partes:

- BackEnd: responsable del manejo de los datos como almacenamiento, validación, acción, transporte y disponibilidad.
- FrontEnd: responsable de la presentación de los datos tales como la interfaz de usuario Web y Móvil.

Se utilizará el patrón de diseño MVC (Modelo Vista Controlador).

Este patrón de diseño define un modelo en donde cada elemento se encarga representar los datos por separado en donde:

- Modelo: Representa la estructura de los datos
- Vista: Representa la presentación de los datos
- Controlador: Escucha eventos emitidos por la vista y se encarga de responder apropiadamente [9]

Una representación gráfica más detallada utilizada como ejemplo para un sitio desarrollado en JSP se puede observar en la Figura 2.6.

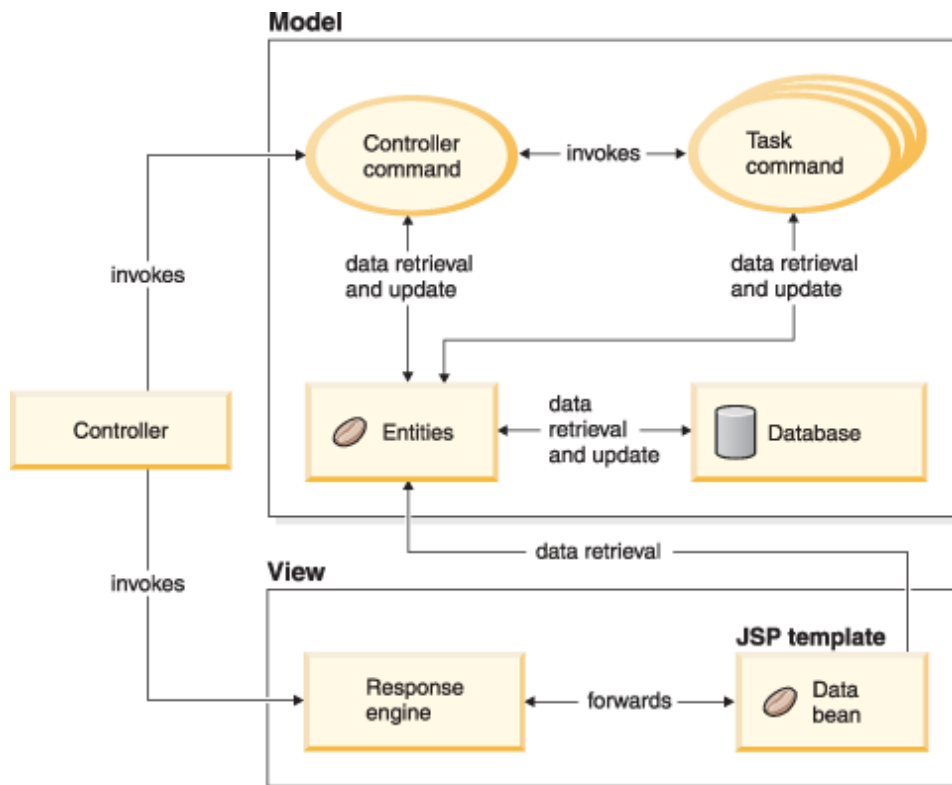


Figura 2.6. Patrón de diseño MVC [9]

En cuanto al almacenamiento de los datos se requiere una base de datos SQL, en donde se representará la información mediante tablas con sus respectivas características.

La comunicación entre los pares FrontEnd y BackEnd a través de internet requiere el uso de un estándar para facilitar el mantenimiento y escalabilidad del sitio. Para esto se utilizará el patrón de diseño API REST.

Este tipo de comunicación utiliza el estándar HTTP en el cual la transferencia de los datos para la comunicación se realiza mediante envío de información en texto plano. Los datos son enviados y recibidos mediante el conjunto de operaciones definidos por HTTP (GET, POST, PUT, PATCH). La estructura de los datos en texto plano generalmente utiliza el formato JSON.

Para el acceso a los datos se utilizan rutas relativas a la dirección web establecida por el BackEnd.

Se puede observar cómo se transfieren los datos en la figura 2.7 [9].

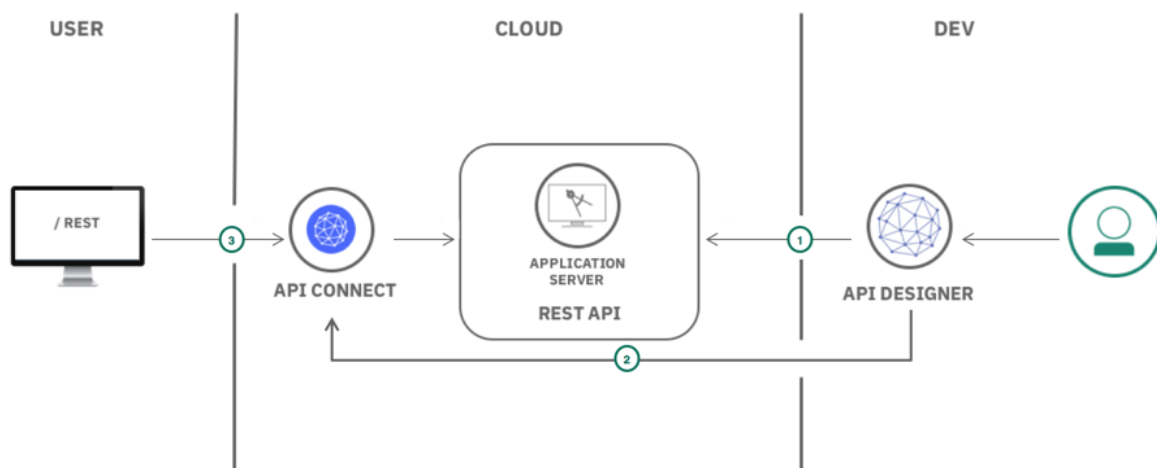


Figura 2.7. API REST [9]

En la actualidad existen diversos sistemas operativos móviles, para el desarrollo de una aplicación para los sistemas operativos en el mercado se requiere la programación en el lenguaje nativo de cada uno de ellos entre los cuales tenemos Android (Java o Kotlin), IOS (Objective C o Swift) como los más populares en el mercado.

La tecnología para el desarrollo de aplicaciones móviles ha avanzado mucho en los últimos años lo que nos permite encontrarnos con soluciones amigables para el programador en donde se puede desarrollar una aplicación general y luego hacer que esta aplicación sea compatible con diversos sistemas operativos. Entre estas soluciones tenemos:

- Aplicaciones Híbridas: desarrollo en tecnologías web las cuales son luego embebidas en un browser dentro del sistema operativo como una aplicación móvil.
- Truly-Native Apps: se utiliza el poder de JavaScript para comunicarse con el API nativo de los sistemas operativos para luego convertirse en una aplicación nativa del sistema operativo. [10]

Uno de los retos en el desarrollo de aplicaciones web y móviles, es la interacción entre el usuario y la aplicación, en donde el usuario debe poder intuir las acciones a poder realizar dentro del aplicativo. Para esto se utilizan técnicas de validación de interfaces graficas entre los más conocidos tenemos: Mockups baja fidelidad, Mockups de alta fidelidad, StoryTelling. En cada una de ellas se representan las pantallas a presentar al usuario con el fin de obtener retroalimentación de la facilidad y uso de la interfaz gráfica. [11]

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

3.1 Requerimientos

En orden de las necesidades descritas en el capítulo anterior, se listan los siguientes requerimientos para la construcción del prototipo funcional.

REQUERIMIENTOS FUNCIONALES:

- Calificar a los oferentes en base a la satisfacción del adquiriente.
- Uso de notificaciones push con firebase.
- Aplicación híbrida para Android y IOS.
- La plataforma debe tener una arquitectura en la nube.
- Publicación de oferta y servicios.
- Contacto del oferente por medio de la aplicación.
- Cotización del oferente por medio de la aplicación.
- Cobro de comisión por cotización al oferente.
- Enviar solicitud de servicio a un oferente.
- Administrar la plataforma.

REQUERIMIENTOS NO FUNCIONALES:

- La aplicación debe ser sencilla de usar.
- La adquisición de un oferente no debe sobrepasar los 4 pasos. palabras de la carrera
- Los oferentes deben validar su cédula de identidad mediante el proceso.
- La aplicación móvil debe contar con diseño de fácil uso siguiendo un patrón como diseño material.
- La plataforma debe ser escalable.

3.2 Herramientas y Tecnología

3.2.1 Arquitectura

La implementación del prototipo funcional requiere de componentes que se integren a través de internet. Entre estos componentes tenemos cuatro más importantes.

- Aplicación móvil (Android/IOS)
- API Rest
- Servidor Web
- Base de datos

La interacción de componentes los podemos observar en el diagrama de la Figura 3.1.

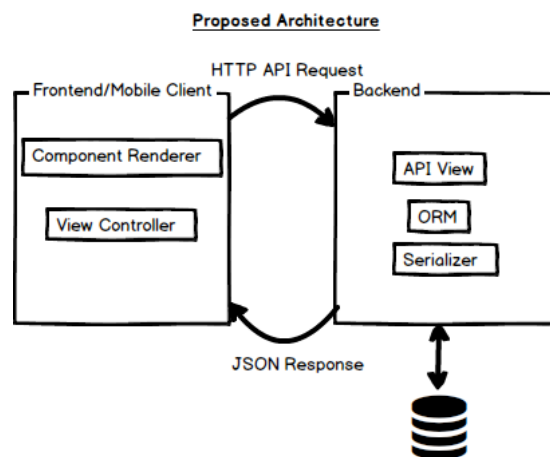


Figura 3.1. API REST [9]

3.2.2 Servidor Web y Base de datos

La aplicación web es fundamental en el desarrollo del prototipo funcional, debido a que procesa toda la información, la válida, almacena y responde.

Este módulo funciona dentro de un servidor conectado a internet, para que pueda ser alcanzado desde cualquier parte del mundo. Debido al alcance

global se necesitan ciertas medidas de seguridad para evitar a los atacantes.

Para los propósitos deseados se necesita una herramienta para el desarrollo de la plataforma, que facilite la comunicación con la base de datos, ofrezca interfaces para la comunicación entre componentes y siga el patrón de diseño MVC. Por esto se ha decidido el framework Django.

[12] Django es un framework creado en 2005 el cual se ha escrito completamente en el lenguaje de programación Python, nos permite la creación de sitios web complejos, escalables, extensibles y además se integra con la mayoría de base de datos relacionales conocidas.

Para almacenar información persistente se necesita el uso de una base de datos, en este caso relacional, debido a la alta consistencia de los datos que se necesita. Para esto se ha elegido la base de datos [13] PostgreSQL, la cual además de contar con toda la funcionalidad de una base de datos relacional, cuenta con tipos de datos no relacionales como JSON, y algoritmos de búsqueda como Full-Text search, la cual se necesita para la búsqueda de servicios en la App Móvil.

3.2.3 Aplicación Móvil

Para las aplicaciones móviles, se usó NativeScript que es un Framework de código abierto cuya ventaja es la construcción de aplicaciones usando un lenguaje de programación independiente del sistema operativo (android, iOS, etc), lo que nos permite utilizar herramientas como XML, CSS y el lenguaje JavaScript. Posee además un poderoso módulo puente el cual traduce las llamadas de las APIs propias de cada plataforma y los retorna a la función que realizó dicha llamada, cada componente que permite NativeScript son traducidos sus respectivos componentes nativos. De esta manera podemos construir aplicaciones nativas para dispositivos Android y iOS. [14]

3.3 Implementación

3.3.1 Plataforma Administrativa

Registro de un nuevo Usuario

Aquí se pueden ver, editar, modificar y borrar los datos de un usuario que se registra mediante la aplicación. Manejamos dos tipos de usuario:

- Al registrarse por primera vez, tenemos el tipo cliente o adquiriente de servicios
- Y para ser oferente de servicios, el usuario debe subir la foto de su cédula, y luego de una verificación por parte del administrador, el cual acepta o declina dicha solicitud mediante el casillero “Es oferente”, puede ser ya un oferente de servicios en la aplicación, o no, dependiendo.

En la Figura 3.2 se muestra la pantalla de Creación de un Usuario mediante la plataforma web.

The screenshot displays a web form for creating a new user. The form includes the following fields and options:

- Numero de telefono:** +593985687938
- Nombres:** Pepito Piguave
- Apellidos:** (empty field)
- Ciudad:** Guayaquil - Guayas - Ecuador (dropdown menu with edit and delete icons)
- Address:** (empty field)
- C.I./R.U.C.:** 0930632596
- Foto de la cédula:** Examinar... No se ha seleccionado ningún archivo.
- Estado del empleado:** ☐ Estado del empleado. Indica si el usuario puede iniciar sesión en el administrador o no.
- Activo:** ☒ Activo. Indica si este usuario debe ser tratado como activo o no. Deseleccione esto en lugar de eliminar cuentas.
- Es oferente:** ☐ Es oferente. Indica si este usuario puede ofrecer servicios o solo pedirlos.
- Fecha en que se unió:**
 - Date:** 2019-08-11 (with Today button)
 - Time:** 00:34:19 (with Now button)
 - Note: You are 5 hours behind server time.

Figura 3.2. Creación de usuario

Reportes

La plataforma del administrador también tiene una opción para exportar los datos en forma de reportes en algunos formatos.

Búsqueda de un servicio

En la pantalla principal de la aplicación se mostrarán los servicios que se ofrece y una barra de búsqueda donde se podrá buscar por categoría el servicio que requiere (Electricidad, Plomería, albañilería, etc) o problema que tiene, por palabras claves por ejemplo: reparación de televisor, computador, refrigerador o construcción de paredes, muebles, etc.

3.3.2 Aplicación Móvil

Características

- **Inicio de sesión con autenticación SMS:**

Al usuario al momento de iniciar sesión le llega un sms al teléfono donde se registró para constatar que el teléfono sea el mismo del registro.

- **Registro de usuario**

El usuario coloca sus datos como numero de teléfono, número de cedula, ciudad, email (para poder recuperar contraseña o algún mensaje enviado por el administrado)

- **Listar servicios**

En la pantalla principal se listan todos los servicios que brindan los usuarios oferentes de la aplicación.

- **Listar categorías**

Una barra debajo de la barra de búsqueda para que el usuario pueda filtrar servicios por categoría.

- **Barra de búsqueda de servicios**

El usuario puede buscar con palabras relacionadas al título, descripción y categoría el servicio que requiere.

- **Publicación de requerimiento múltiple**

La aplicación permite al usuario enviar una solicitud con el problema o falla que tiene a múltiples usuarios (más cercanos, mejores puntuados o aleatoriamente).

- **Publicación de servicio como oferente**

Un usuario de la aplicación puede publicar lo que hace o servicios que brinda una vez que es aceptado como oferente de servicios.

- **Listar ordenes de servicios**

Tanto el usuario adquiriente como el oferente pueden visualizar las peticiones u ordenes de servicios realizadas o adquirida.

- **Detalle de la orden del servicio**

Asimismo, se puede visualizar los pormenores de cada orden de servicio.

- **Enviar cotización de servicio**

Cuando una orden de servicio es recibida, el usuario como oferente puede analizar y luego enviar el precio que cree conveniente.

- **Aceptar o rechazar cotización de servicio**

El usuario analiza el precio que el oferente colocó y puede aceptar para dar a cabo la contratación del servicio o rechazarlo

- **Notificaciones push**

La aplicación notifica al usuario cuando es aceptado como oferente, cuando recibe un orden de servicio, cuando recibe la cotización de su orden y cuando es aceptada o rechazada la cotización.

- **Subir imágenes de requerimiento al servidor**

El adquiriente puede enviar fotos de los daños que quiere que sean reparados o referencias de lo que solicita que haga el oferente.

- **Cerrar Sesión**

Un usuario previamente autenticado en la plataforma puede hacer uso del cierre de sesión para que otros usuarios no tengan acceso a sus datos.

- **Calificación del servicio con estrellas**

Un usuario puede calificar el servicio recibido mediante 5 estrellas.

- **Actualizar datos del usuario**

Los usuarios, oferente y adquirientes pueden editar los datos proporcionados en el registro.

- **Componentes reactivos para actualización de datos**

Son todos los componentes que no necesitan ser accionados por el usuario para obtener una respuesta, entre estos se encuentran las notificaciones push, inicio y cierre de sesión, envió y recepción de actualizaciones en la orden de servicio.

A continuación, en las Figuras 3.3 a 3.10, se adjuntan las pantallas de la Aplicación móvil.



Figura 3.3. Pantalla Principal

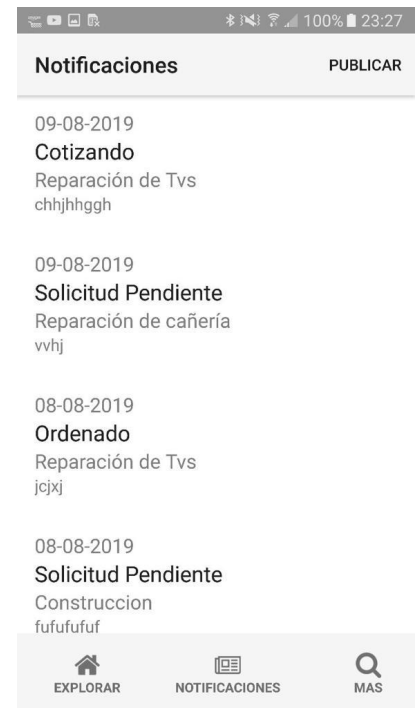


Figura 3.4. Notificaciones



Figura 3.5. Perfil

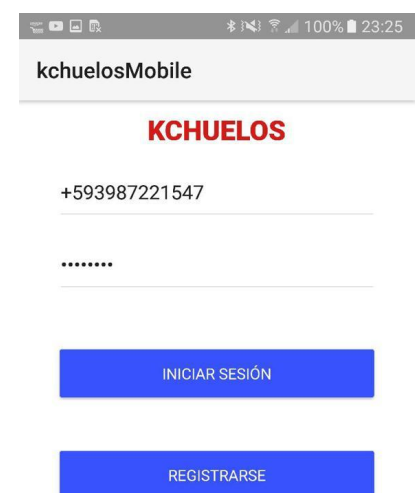


Figura 3.6. Inicio de sesión

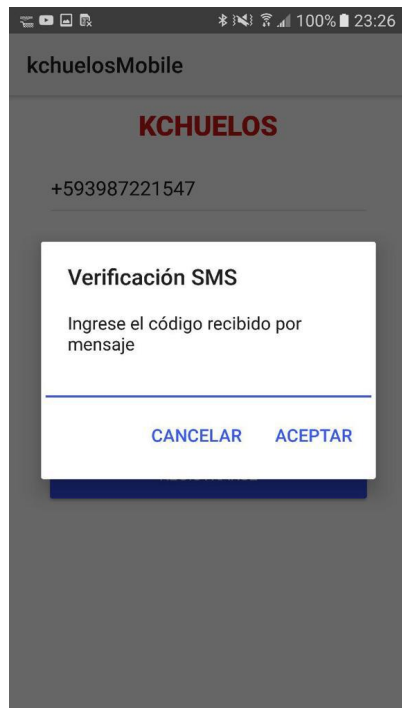


Figura 3.7. Verificación SMS

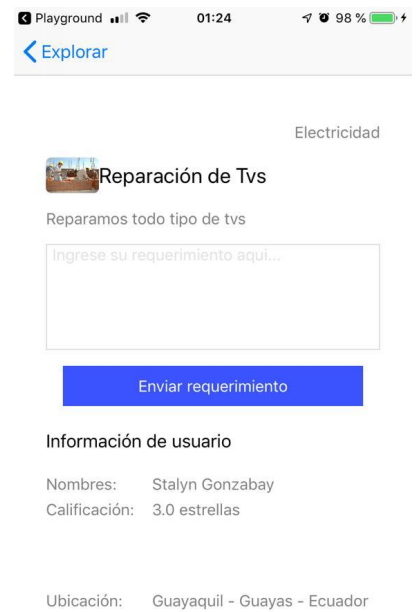


Figura 3.8. Enviar Requerimiento

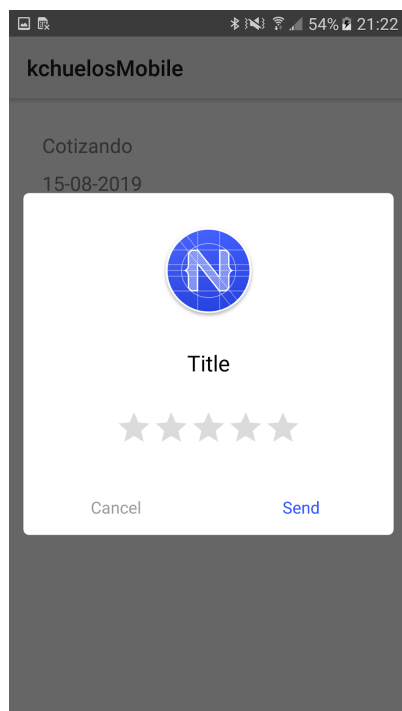


Figura 3.9. Calificar servicio



Figura 3.10. Publicar Requerimiento

3.3.3 Aplicación Web

Características

- **BASE DE DATOS:**
 - Ordenes de Servicio
 - Usuario
 - Servicio
 - Categorías
 - Notificaciones
 - Registro token Firebase
 - Calificaciones

- **API REST**
 - Autenticación
 - Registro de usuario
 - Validación de SMS
 - Publicación de servicio
 - Publicación de orden de servicio
 - Completar registro oferente
 - Aceptar cotización
 - Enviar cotización
 - Registro de token firebase
 - Consumo de servicios
 - Consumo de ordenes de servicios
 - Consumo de perfil

- **SEÑALES**
 - Enviar notificaciones
 - Validar cobro de comisiones
 - Validar autenticación de usuario
 - Validación de estado de la orden del servicio

3.3.4 Administrador Web

Características

- Administración de usuarios
- Administración de categorías
- Administración de ordenes de servicio
- Administración de calificaciones
- Administración de usuarios registrados con firebase
- Administración General
- Reportes de ordenes de servicio
- Reporte de usuarios
- Reporte de servicios

En la Figura 3.11 se muestra pantalla principal de la Aplicación Web

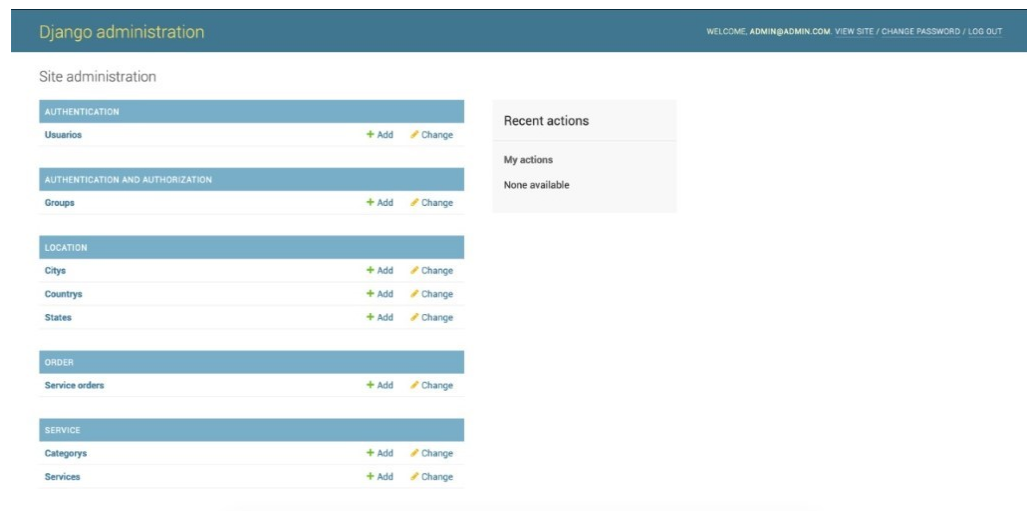


Figura 3.11. Página administrativa principal

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

El trabajo realizado ha despertado gran interés en el público en general debido a la innovación de un servicio que no ha tenido cambios durante un largo tiempo, brindando una importancia significativa al campo estudiado y a su estructura tecnológica que lo soporta. Existen puntos clave en los cuales la aplicación del concepto descrito puede tener problemas, como el desinterés o la falta de habilidad de uso de un smartphone por parte de los oferentes de servicio. Por otra parte, la realización de una plataforma similar a la desarrollada como prototipo, no debería tomar más de 6 meses de implementación con las herramientas y tecnología descrita en el desarrollo de la solución. En un trabajo futuro la implementación del prototipo funcional realizado sería posible sin ningún tipo de problema, ya sea conceptual o tecnológico. Se pretende estudiar más a fondo la interacción humano-computador para que la usabilidad de la aplicación sea lo suficiente buena para que los oferentes no tengan problemas al usarla.

Conclusiones

La implementación de la aplicación móvil, como solución al problema de búsqueda y oferta de servicios, se facilitó gracias a la utilización del framework "NativeScript" cuya ventaja principal es acelerar el desarrollo de una aplicación nativa, funcional tanto en Android como en iOS.

El bosquejo del prototipo realizado en la herramienta "Marvelapp" nos permitió definir las funcionalidades y establecer la navegación correspondiente para reflejar los requerimientos del cliente.

La plataforma web que gestiona la información de la aplicación móvil es vital para el desarrollo de un SaaS debido a que, constituye la parte central del ciclo del negocio.

La interfaz gráfica de la aplicación móvil es intuitiva con el fin de mejorar la experiencia de usuario e incentivar su uso en profesionales y técnicos que ofertan servicios.

Recomendaciones

El diseño de la base de datos en una plataforma tipo SaaS es muy importante que se analice detenidamente al inicio del proyecto a realizar, debido a que cambios en las tablas con datos en producción puede tornarse en un gran trabajo.

El uso de un framework “truly-native” debe ser analizado después de tener una idea de la interfaz gráfica a realizarse, debido a que pueden existir limitaciones en cuanto al diseño que se espera.

Para asegurar la escalabilidad de un sistema, puede usarse una herramienta de contenedores como Docker para su posterior escalamiento.

El despliegue de un sistema que se prevé que tenga una demanda creciente se recomienda se utilice una infraestructura en la nube como Digital Ocean o AWS.

En cuanto al desarrollo del producto, se recomienda obtener retroalimentación del usuario final constantemente para evitar cambios importantes después de tener un avance que toma tiempo.

BIBLIOGRAFÍA

- [1] A. d. R. y. Control, «Arcotel,» Junio 2017. [En línea]. Available: <http://www.arcotel.gob.ec/464-de-usuarios-del-servicio-movil-avanzado-poseen-un-smartphone/?fbclid=IwAR1JnKL35I7KMQXiokZIBjoUv5-GjOKlgZMM1PC40TTZ2B8NDxmceNEUWMg> . [Último acceso: 2019].
- [2] N. G. Mankiw, «La identificación del desempleo,» de *Principios de la Economía*, Cengage Learning, 2012.
- [3] K. Vite, «Milenio,» 27 Enero 2019. [En línea]. Available: <https://www.milenio.com/negocios/ayfix-una-app-que-resuelve-problemas-de-mantenimiento>.
- [4] INEN, «Ecuador en cifras,» 2019. [En línea]. Available: http://www.ecuadorencifras.gob.ec/documentos/web-inec/EMPLEO/2019/Marzo/Boletin_mar2019.pdf.
- [5] Olx, «Microsoft,» 2019. [En línea]. Available: <https://www.microsoft.com/es-es/p/olx/9wzdnrcfj1qx?activetab=pivot:overviewtab>.
- [6] ¿. e. SaaS?, «Microsoft Azure,» 2018. [En línea]. Available: <https://azure.microsoft.com/es-es/overview/what-is-saas/>.
- [7] T. S. F. Poster, «Scrum,» 2017. [En línea]. Available: <https://www.scrum.org/resources/what-is-scrum>.
- [8] P. d. d. d. modelo-vista-controlador, «IBM Knowledge Center,» [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SSZLC2_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm.
- [9] p. y. g. d. I. A. R. Creación, «IBM Cloud,» 07 Marzo 2019. [En línea]. Available: <https://cloud.ibm.com/docs/tutorials?topic=solution-tutorials-create-manage-secure-apis&locale=es>.
- [10] D. A. C. A. I. t. T. N. Apps, «Zolando,» 15 Julio 2016. [En línea]. Available: https://jobs.zalando.com/tech/blog/an-introduction-to-truly-native-apps/?gh_src=4n3gxh1.
- [11] P. Szekely, *User Interface Prototyping: Tools and Techniques*, 2005.
- [12] M. Foundation, «Django Web Framework,» 2019.

- [13] PostgreSQL, «What is Postgres?,» 2013. [En línea]. Available: <https://www.postgresql.org/about/>.
- [14] NativeScript, «How does NativeScript work?,» 2019. [En línea]. Available: <https://www.nativescript.org/faq/how-does-nativescript-work>.

APÉNDICES

APÉNDICE A: LISTA Y DETALLE DE ENDPOINTS

API DE SERVICIOS: "services": [api/services/](#)

```
{  
  "id",  
  "user",  
  "user_name",  
  "user_number",  
  "user_city",  
  "user_address",  
  "user_identification",  
  "user_is_active",  
  "category",  
  "category_name",  
  "photos",  
  "title",  
  "description",  
  "stars"  
}
```

API CATEGORIAS: "categories": [/api/categories/](#)

```
{  
  "id",  
  "create_at",  
  "update_at",  
  "name"  
}
```

API USERS: "users": [/api/users/](#),

```
{
    "id",
    "number",
    "names",
    "last_names",
    "city",
    "address",
    "amount",
    "identification",
    "CI",
    "is_active",
    "is_offerer",
    "date_joined",
}
```

API DE ORDENES DE SERVICIO:

"serviceorders": "</api/serviceorders/>",

```
{
    "state",
    "adquirientes",
    "service",
    "value",
    "requisition",
    "damage_photos ",
}
```

API DE PUTUACIONES: scores": "</api/scores/>"

```
"stars",
"accumulated",
"average",
"service_order",
```



```
}
```

API DE CIUDADES: "cities": "</api/cities/>"

```
{  
  "id",  
  "state",  
  "name",  
  "abbreviation":  
}
```

API DE PAISES: "countries": "</api/countries/>"

```
{  
  "id",  
  "name",  
  "abbreviation",  
  "currency",  
  "currency_symbol"  
}
```

API DE ESTADOS: "states": "</api/states/>"

```
{  
  "id",  
  "country",  
  "name",  
  "abbreviation"  
}
```

API DE DISPOSITIVOS "devices": "</api/devices/>"