

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Desarrollo de una solución multiplataforma para la gestión de servicio  
de entregas a domicilio

**PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**Ingeniero en Ciencias de la Computación**

Presentado por:

Bryan Alexander Puchaicela Buri

Kevin Vicente Briones Fuentes

GUAYAQUIL - ECUADOR

Año: 2022 - 2023

## DEDICATORIA

El presente proyecto lo dedicamos a:

Nuestras familias (Betty Buri y Hugo Puchaicela; Lola Fuentes y Cristina Briones), ya que velaron por nuestro bienestar durante todo este periodo, y nos ofrecieron todo el espacio y equipamiento necesario para culminar el proyecto de manera exitosa y sin contratiempos.

Además, dedicamos este proyecto a nuestros jefes directos de las empresas en que trabajamos, ya que además de ser impulsores de nuestro desarrollo profesional, fueron propulsores de nuestro desarrollo académico, brindándonos todo el tiempo necesario para la culminación de los entregables requeridos para el presente proyecto de materia integradora.

## **AGRADECIMIENTOS**

Agradecemos a Dios, ya que sin él no pudiéramos haber logrado llegar a donde estamos, y que, en compañía de nuestros padres, nos dieron la fortaleza, el apoyo incondicional, y la oportunidad de estudiar la carrera académica en la Institución que anhelamos.

Además, agradecemos a la ESPOL por permitirnos ser partícipes de la institución, la excelencia académica, y todo el aval y reconocimiento internacional que como institución ofrece.

De igual manera, agradecemos a aquellos profesores que nos impartieron clases, y que se convirtieron en nuestros ejemplos de vida, ya que fueron aquellos que nos hicieron crecer de manera ética y profesional, porque nos hicieron ver las dificultades como muros a escalar para llegar a nuestras metas.

## DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *KEVIN VICENTE BRIONES FUENTES* y *BRYAN ALEXANDER PUCHAICELA BURI* 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”



Bryan Alexander  
Puchaicela Buri



Kevin Vicente Briones  
Fuentes

## EVALUADORES

.....  
**Ronald Raul Criollo Bonilla**

PROFESOR DE LA MATERIA



.....  
**Lenin Eduardo Freire Cobo**

PROFESOR TUTOR

## RESUMEN

En el mercado laboral hay un sin número de pequeños emprendimientos, establecimientos o microempresas que no cuentan con infraestructura logística de entrega a domicilio, lo cual limita su capacidad de ventas, a diferencia de empresas que si cuentan con estos servicios y su adquisición económica ha crecido gracias a ello. El presente trabajo tiene como objetivo desarrollar una solución multiplataforma que ayude a esas organizaciones a conseguir un sistema logístico que ayude a cubrir su gestión de entrega a domicilio.

El proyecto fue desarrollado inicialmente gracias al prototipado realizado en FIGMA, prototipado que se realizó para dispositivos móviles y web por la solución multiplataforma generada. El prototipado principalmente se generó en base de requerimientos funcionales y no funcionales, levantados tras sesiones en conjunto con el cliente del proyecto, sacando a su vez los alcances y las limitaciones. Para el desarrollo, se usaron marcos de trabajos para las plataformas web y nativas para la realización de la solución, lo cual fue documentado, añadiendo en ello las pruebas de integración realizadas para corroborar el correcto funcionamiento del sistema.

Finalmente, se concluye que la generación inicial de la arquitectura, y las pruebas de integración fueron cruciales para la culminación del proyecto, y la visualización del comportamiento del sistema ante los usuarios finales.

**Palabras Clave:** Gestión de servicio de entrega a domicilio, solución multiplataforma, requerimientos funcionales y no funcionales, pruebas de integración.

## **ABSTRACT**

*In the industry are a big number of small businesses, establishments or micro enterprises that don't have home delivery logistics, which limits their sales capacities, unlike companies that have these services, and their economic acquisition has grown thanks for that. The objective of this work is to develop a multiplatform solution that helps these organizations to achieve a logistics system that helps cover their home delivery management. The project was initially developed thanks to the prototyping carried out in FIGMA, prototyping that was carried out for mobile devices and the web due to the multiplatform solution generated. The prototyping was mainly generated based on functional and non-functional requirements, raised after sessions together with the client of the project, taking out the scope and limitations at the same time. For the development, frameworks for web and native platforms were used to carry out the solution, which was documented, adding to it the integration tests carried out to corroborate the functions of the system. Finally, it is concluded that the initial generation of the architecture, and the integration tests were crucial for the completion of the project, and the visualization of the behavior of the system before the end users.*

*Keywords: Home delivery service management, multiplatform solution, functional and non-functional requirements, integration tests.*

# ÍNDICE GENERAL

EVALUADORES.....	5
RESUMEN .....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL .....	III
ABREVIATURAS.....	V
ÍNDICE DE FIGURAS .....	VI
ÍNDICE DE TABLAS.....	VII
CAPÍTULO 1 .....	1
1.    Introducción .....	1
1.1    Descripción del problema .....	1
1.2    Justificación del problema .....	1
1.3    Objetivos .....	2
1.3.1    Objetivo General .....	2
1.3.2    Objetivos Específicos.....	2
1.4    Marco teórico.....	2
CAPÍTULO 2 .....	6
2.    Metodología.....	6
2.1    Análisis.....	6
2.1.1    Requerimientos funcionales .....	6
2.1.2    Requerimientos No Funcionales .....	7
2.1.3    Alcance y limitaciones .....	7
2.1.4    Riesgos y beneficios .....	8
2.1.5    Usuarios.....	8
2.2    Prototipado .....	9
2.2.1    Administradores y Organizaciones .....	10



2.2.2	Administradores .....	12
2.2.3	Organizaciones .....	13
2.2.4	Repartidores .....	13
2.3	Evaluación.....	16
CAPÍTULO 3 .....		18
3.	Resultados Y Análisis .....	18
3.1	Análisis .....	18
3.2	Desarrollo de la aplicación.....	18
3.3	Plan de Implementación .....	20
3.4	Pruebas .....	20
3.4.1	Pruebas de integración .....	20
3.5	Análisis de Costo.....	21
3.5.1	Costos de Implementación.....	21
3.5.2	Costos de despliegue.....	22
4.	Conclusiones Y Recomendaciones.....	23
4.1	Conclusiones .....	23
4.2	Recomendaciones.....	23
BIBLIOGRAFÍA .....		24
ANEXOS .....		25

## ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral.
B2B	Business to Business (Tipo de comercio entre Negocio a Negocio).
TIA S.A.	Tiendas Industriales Asociadas S.A.
COE	Comité de Operaciones de Emergencia.
APIs	Application Programming Interface (Software intermediario que permite la comunicación entre 2 aplicaciones).
COE	Comité de Operaciones de Emergencia.
SARS-CoV-2	Virus de transmisión respiratorio de la familia de coronavirus.
Docker	Tecnología que permite contener aplicaciones, y poder desplegarlas de manera más sencilla en distinto ambientes.
Framework	Marco de trabajo que propone criterios y estándares, los cuales se deben seguir para generar un buen código, además que brindan herramientas que facilitan el desarrollo.
PWA	Aplicación web progresiva, la cual permite la posibilidad de crear aplicaciones nativas a partir del código web.
SCRUM	Metodología de trabajo que admite un conjunto de metodologías para trabajar efectivamente en equipo.

## ÍNDICE DE FIGURAS

Figura 1.1 Representación gráfica de evolución del número de empresas (INEC 2020 DIEE) .....	4
Figura 2.1 Arquitectura de la solución de servicio de gestión de entrega a domicilio [Autoría propia]. .....	9
Figura 2.2 Pantalla de inicio para organizaciones y administradores [Autoría propia]. .....	11
Figura 2.3 Pantalla de visualización de apis para organizaciones [Autoría propia]. .....	12
Figura 2.4 Pantalla de inicio de sesión para organizaciones y administradores [Autoría propia]. .....	12
Figura 2.5 Pantalla de visualizaciones de usuarios [Autoría propia]. .....	13
Figura 2.6 Pantalla de visualización de usuarios [Autoría propia]. .....	13
Figura 2.7 Pantalla de inicio de sesión [Autoría propia]. .....	14
Figura 2.8 Pantalla de inicio con muestra de tickets [Autoría propia]. .....	14
Figura 2.9 Pantalla de confirmación de entrega de tickets [Autoría propia]. .....	15
Figura 2.10 Pantalla de tracking de pedido [Autoría propia]. .....	15
Figura 2.11 Pantalla de finalización de entrega de pedido [Autoría propia]. .....	15
Figura 2.12 Pantalla de visualización de pedidos cercanos a motorizados [Autoría propia]. .....	15
Figura 2.13 Visualización del menú de opciones [Autoría propia]. .....	16
Figura 2.14 Pantalla de perfil de usuario [Autoría propia]. .....	16

## ÍNDICE DE TABLAS

Tabla 1.1 Aplicaciones de delivery y más utilizadas en el ecuador .....	3
Tabla 3.1 Tabla de actividades [autoría propia] .....	19
<i>Tabla 3.2 Estimación de horas y costes según las características desarrolladas en la solución [Autoría propia] ..</i>	<i>21</i>

# CAPÍTULO 1

## 1. INTRODUCCIÓN

### 1.1 Descripción del problema

La mayoría de las empresas y establecimientos comerciales no tienen plataformas digitales, sino que tienen una infraestructura física de comercio en el cual el usuario debe acercarse al establecimiento para completar su compra; así mismo, algunos establecimientos comerciales deciden digitalizar su negocio, creando plataformas de comercio electrónico, o, por otro lado, existen establecimientos que ya hacen comercio digital, sin embargo, en la mayoría de estos casos, estas empresas no pueden ofrecer servicios para gestionar la entrega del producto a domicilio del cliente.

Un ejemplo de establecimientos que deciden crear o adquirir plataformas digitales es TIA S.A., el cual, a raíz de la pandemia, han creado nuevos canales de comercio digital, e incluso adquirido una plataforma móvil de gestión de entregas de pedido a domicilio (Delivereo).

Cabe denotar que TIA, es una empresa que tiene cobertura a nivel nacional, y el optar adquirir o crear una solución como lo es Delivereo no es factible para empresas pequeñas, e incluso medianas.

Para estas últimas, empresas que ya hacen comercio electrónico y quieren optar por crear o adquirir un servicio de gestión de entrega a domicilio, e incluso aquellas que recién se están digitalizando y quieren proveer este servicio, se les convierte en un reto que en la mayoría de los casos desisten por incorporarlos.

### 1.2 Justificación del problema

La mayoría de los establecimientos comerciales se están adaptando a las nuevas tecnologías por lo cual deciden en algunos casos crear o adquirir el servicio de un tercero. Lo que conlleva que deben realizar nuevas formas de mercadeo, seguridad, fiabilidad y entender el gasto que implica en dicho servicio de realizar o adquirirlo. También, deben comprender las necesidades del consumidor debido que está enfocado en ellos.

Para lograr su meta de adaptación a las tecnologías las entidades necesitan realizar una búsqueda exhaustiva del mercado digital para ver que aplicaciones móviles disponen del

servicio, y si su infraestructura si se adapta al negocio de su modelo y rentable. Adicionalmente, el presente proyecto integrador se enfoca en el desarrollo de una solución multiplataforma para la gestión de servicio de entrega a organizaciones, solución afianzada en la versatilidad de integración con las plataformas existentes, en desarrollo, o futuras, establecimientos o empresas que requieran incorporar servicios de gestión de entrega a domicilio.

De manera que, con esta solución, las empresas cuenten con un servicio de gestión de entrega a domicilio fácil de administrar, con costo según la necesidad, e integrable según la evolución digital del negocio.

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

Desarrollar una solución multiplataforma ofertando el servicio de gestión de entrega a domicilio para las organizaciones u establecimientos que no puedan cubrir el sistema logístico.

#### **1.3.2 Objetivos Específicos**

1. Elaboración de la arquitectura de software sobre el servicio de gestión de entrega a elaborar para describir las partes que conformarán al sistema.
2. Desarrollo de una aplicación móvil para la gestión y entrega por parte de repartidores.
3. Desarrollo de una plataforma web con módulos a organizaciones y administración interna para la gestión interna y externa de la solución de servicio de entrega.
4. Ejecución de pruebas exhaustivas del módulo de solicitud y entrega de tickets, para la validación de las APIs ofertadas a las empresas externas.

### **1.4 Marco teórico**

Eventualmente los establecimientos comerciales han optado por una infraestructura virtual dado que permite llegar a diferentes sectores geográficos. Además, ofrecer sus servicios a través del internet. Lo que conlleva que la mayoría de los locales comerciales opten por brindar un servicio a domicilio para su respectiva clientela.

“El servicio de domicilio se entiende como la entrega de servicios o productos realizados por una compañía a sus clientes finales con las especificaciones acordadas de condiciones de productos y tiempo de llegada” (Otero Giraldo, 2017, p.23).

En Ecuador hay diferentes plataformas virtuales que ofrecen el servicio a domicilio. Por ejemplo: Nobles et al. (2022) menciona “En el año 2016 la empresa colombiana Domicilios.com, pionera en los servicios de delivery a nivel local, lo cual ha levantado cerca de 40 millones de dólares americanos en capital y tiene presencia en varios países del continente” (p.10).

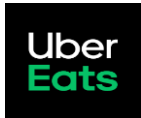
Posteriormente, el uso de las aplicaciones de gestión de entrega a domicilio en el Ecuador, si bien ha sido incorporado antes de la pandemia del SARS-CoV-2 en 2019, comenzó a usarse de manera más frecuente a inicios de esta, debido a la emisión de decretos por el Gobierno ecuatoriano a establecimientos comerciales para el control infeccioso.



Este control infeccioso; generando al inicio pérdidas cuantiosas en ventas por medidas tomadas por el COE Nacional como, suspensión de actividad comercial a excepción de artículos de primera necesidad; crearon una necesidad de búsqueda a los establecimientos para evitar estas pérdidas. Necesidad que algunos establecimientos pudieron encontrar solución en aquellas plataformas que le permitían entregar su producto por medio de un intermediario.

Ejemplos de aquellas plataformas son:

- Uber eats, que permite a establecimientos registrar sus negocios y productos para que los clientes hagan pedidos, y estos puedan ser entregados por un repartidor cercano.
- Rappi, solución similar a la que ofrece uber eats, pero con mayor variedad de locales, y ganancia al repartidor basada en la distancia recorrida.
- Pedidos YA, que permite realizar envíos de productos a domicilio. Antes llamada Glovo.

Tabla 1.1 Aplicaciones de delivery y más utilizadas en el Ecuador

App	Logo	Tiempo de operación	Usuarios activos	Afiliados/ Repartidores	Lugares de funcionamiento
UBER EATS		Empezó su operación en Ecuador en el 2018.	1.5 millones de usuarios.	Tiene un promedio de 25.000 repartidores.	Quito, Guayaquil, Cuenca y Ambato.

PEDIDOS YA		Funcionando anteriormente como Glovo, inició sus operaciones como Pedidos Ya en el 2021.	1 millón de usuarios aproximadamente.	Cuenta con 2.500 repartidores y 7.000 restaurantes registrados.	Quito, Guayaquil y Cuenca.
RAPPI		Inició sus operaciones en Ecuador en septiembre del 2019.	Aproximadamente 500.000 usuarios.	Inició con 350 repartidores, actualmente cuenta con un estimado de 3.500 repartidores.	Guayaquil y Quito

Nota: Las aplicaciones de delivery han sido un factor importante para el Ecuador, siendo estas 3 las más utilizadas, teniendo gran acogida a nivel nacional con gran cantidad de usuarios activos y flotas de repartidores (Román & Huayamave ,2022).

Sin embargo, no todos los establecimientos comerciales tenían las condiciones necesarias para brindar dicho servicio a domicilio, por lo que aquellos que no tenían estas condiciones cerraron. Según los datos realizados por (INEC, 2020) hubo 886,600 empresas registradas en el Ecuador en el periodo 2019. Pero, en el siguiente año el registro de empresas registradas cayó a 846,265 que fue a raíz de la pandemia.

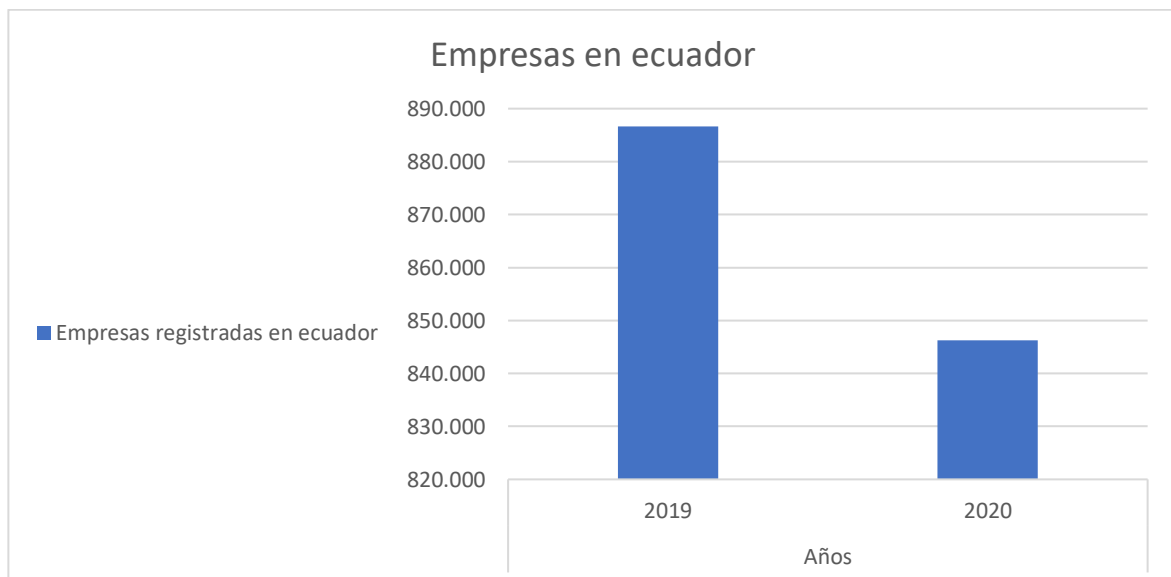


Figura 1.1 Representación gráfica de evolución del número de empresas (INEC 2020 DIEE)

Ahora bien, se conoce que el presidente del Ecuador Guillermo Lasso ha afirmado que la pandemia en Ecuador ha llegado a su fin, sin embargo, esto no significa que la adopción de las plataformas digitales también ha terminado. Por el lado contrario, establecimientos como TIA, apuestan por el crecimiento de estas plataformas, adquiriendo incluso el 60% de acciones de la plataforma de gestión de entrega a



domicilio (Deliveroo). Mas aún, TIA apuesta como accionista extender las operaciones de Deliveroo a países como Uruguay y Argentina.

Deliveroo, desarrolla el negocio B2B, el cual ofrece a las empresas que se alían una plataforma tecnológica bajo el modelo de economía compartida, siendo una forma de negocio útil y entusiasta

# CAPÍTULO 2

## 2. METODOLOGÍA

Para el desarrollo de la solución del servicio de gestión de entrega a domicilio se tuvo que realizar un análisis funcional y no funcional de lo que el servicio debe ofrecer, por ello inicialmente se realizó un levantamiento de requerimientos segmentado por el tipo de usuario, este levantamiento nos permite definir las necesidades y funcionalidades que como servicio debemos ofrecer.

También fue importante definir hasta dónde vamos a llegar, pues por la duración de implementación del proyecto, necesidades futuras, o tecnologías, se definió los alcances y limitaciones de la solución.

Posteriormente, se definen aspectos importantes, tales como los tipos de usuarios, y los riesgos y beneficios al tomar esta metodología para el desarrollo de la solución.

Luego se establece la arquitectura de la solución, la cual cuenta con la descripción de la arquitectura y prototipos, de los diferentes módulos que conformarán el sistema, tales como; el módulo de repartidores el cual estará disponible para aplicaciones móviles, y el módulo de organizaciones y administradores el cual estará disponible para plataformas Web.

Finalmente, se realizará una evaluación de la arquitectura y los prototipos, obteniendo una interacción y retroalimentación por el usuario; pensando en la experiencia de usuario y el buen diseño de interfaces pensado en el usuario.

### 2.1 Análisis

#### 2.1.1 Requerimientos funcionales

##### Repartidor

- Como repartidor permitir registrarse a través de diferentes plataformas: Desde el registro directo, por Google y Facebook para iniciar sesión con las credenciales (usuario y contraseña) de mi plataforma seleccionada.

- Como repartidor permitir subir una foto de perfil para facilitar la identificación visual por parte del cliente y la organización.
- Como repartidor permitir filtrar tickets por la distancia entre mi ubicación actual y el punto de recogida para seleccionar el ticket más próximo a mí.
- Como repartidor permitir cambiar estado a “inactivo” (no disponible para trabajar) en ciertos horarios del día para tomarme un descanso y atender otras responsabilidades.
- Como repartidor permitir hacer transferencia de dinero a una cuenta bancaria.

### **Organizaciones**

- Como organización quiero visualizar información general de presentación de Colibrí.
- Como organización permitir registrarse en la plataforma de servicio de gestión de entrega a domicilio para iniciar sesión como administrador.
- Como organización permitir visualizar los índices de costos de manera gráfica sobre las entregas de pedidos.
- Como organización quiero monitorear en tiempo real la ubicación del repartidor durante la etapa de recogida para anticiparme a la llegada del repartidor.
- Como organización quiero visualizar documentación para obtener APIs de seguimiento y solicitud de repartidores para utilizarlas en desarrollos propios

### **Administradores**

- Como administrador quiero iniciar sesión para gestionar los usuarios (repartidores y organizaciones)

### **2.1.2 Requerimientos No Funcionales**

- Refactorización de código de servicios de NodeJS existentes para su posterior migración.
- Desarrollo de manual técnico, describiendo levantamientos, ambientes de desarrollo, aseguramiento de calidad, y producción.

### **2.1.3 Alcance y limitaciones**

El servicio de gestión de entrega a domicilio es una solución de software que beneficiará a tres tipos de usuarios, y al ser el público objetivo se puede decir que también se limita

a este; así como se detalla esta limitación, existen otras más las cuales por definiciones, alcances, etc., se detallarán a continuación:

- La solución disponible para dispositivos móviles (Apple y Android) para repartidores; y para organizaciones y administradores en una plataforma web.
- La solución prevendrá el límite de usuarios que soporta sin que exista ninguna caída del servicio, para visualizar la posible demanda del mercado.
- La solución estará implementada de tal forma que, en caso de querer migrarse el software del mismo, permita migrarse fácilmente si el ambiente de producción a subirse permita utilizar contenedores de Docker.
- La administración general del sistema de repartidores se limitará a permitir y bloquear usuarios, así mismo para el bloqueo y habilitación de organizaciones.
- La solución se limitará a ofrecer el servicio exclusivo en lenguaje español.
- Brindar soporte para que en un futuro se pueda incrementar el uso del sistema en idioma inglés.
- La solución se limitará a refactorizar y utilizar aquellos recursos provistos por anteriores desarrollos de este proyecto, siempre y cuando permitan el menor esfuerzo posible; lo demás será implementado desde 0.

#### **2.1.4 Riesgos y beneficios**

Con la metodología propuesta se tuvo los siguientes beneficios:

- Levantamiento de soluciones de manera dinámica, de modo que se pueda migrar de sistema de manera inmediata sin alto nivel de esfuerzo en mantenimiento.
- Actualización de puntos de geolocalización de manera asíncrona, lo cual permite un consumo eficiente de recursos para otras tareas en el módulo de repartidores.

#### **2.1.5 Usuarios**

Se definió que los usuarios involucrados en la solución del servicio de entrega son los siguientes:

- **Repartidores:** Son aquellas personas que permiten la entrega directa de pedidos desde las organizaciones hasta los usuarios finales. Son necesarios para el flujo de la solución, ya que proporcionan el recurso de traslado de pedidos de organizaciones a través de la ciudad.

- **Organizaciones:** Son los clientes de la solución, con estos usuarios se puede medir la tasa de éxito o fracaso del servicio, pues con ellos se genera el flujo de caja de la solución.
- **Administradores:** Son aquellos usuarios que realizan un seguimiento a los demás, y con ello pueden permitir el ingreso de unos usuarios, y así mismo el bloqueo de otros.

## 2.2 Prototipado

Previo al prototipado se presenta el diseño de la arquitectura del servicio de gestión de entrega a domicilio, el cual fue necesario para definir todos los recursos, instancias y plataformas necesarias a utilizar dentro del marco de la solución.

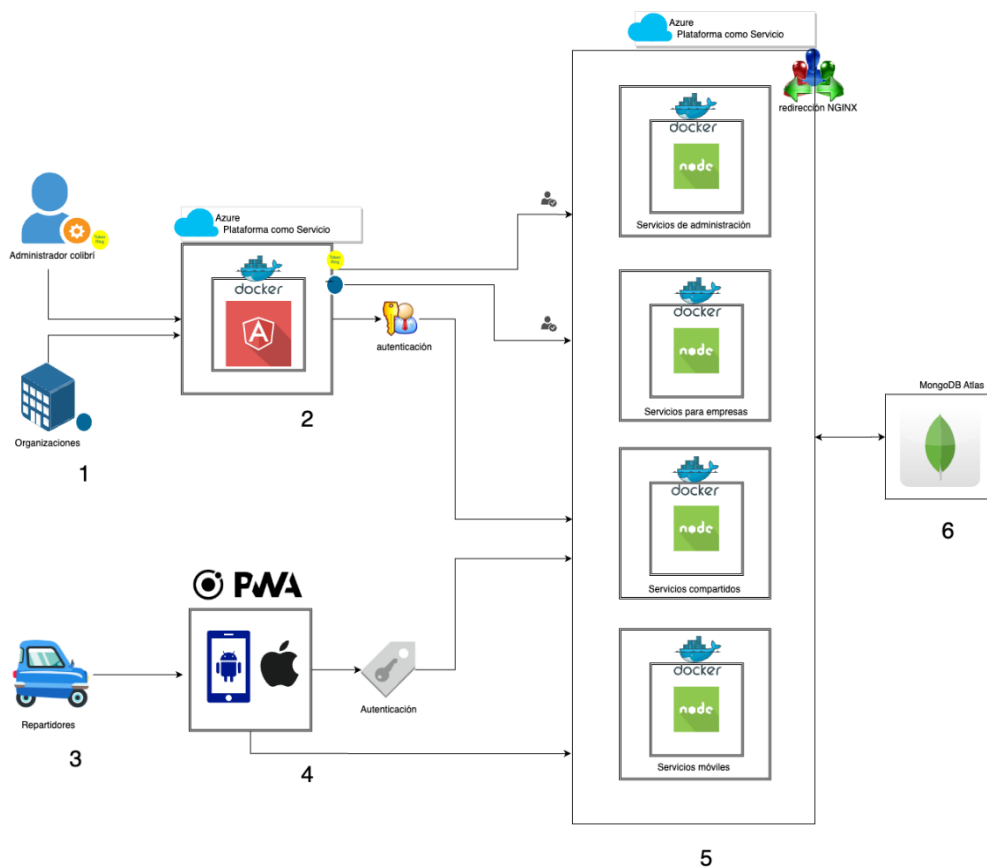


Figura 2.1 Arquitectura de la solución de servicio de gestión de entrega a domicilio [Autoría propia].

La Figura 2.1 muestra la arquitectura de la solución, cuyos puntos importantes están señalados.

- El punto 1 y 3 muestran los tipos de usuarios que utilizarán nuestro servicio, servicio que estará compartido entre 2 tipos de plataformas.

- Las plataformas a utilizar son de dos tipos, en el punto 2 se muestra que la plataforma va a ser WEB, el cual está conformada por el Framework Angular, el cual estará contenido con contenedores Docker y desplegado en Azure.
- En el punto 4 se muestra que esta plataforma es de tipo PWA, el cual estará enfocado para dispositivos móviles Android y iOS.
- En el punto 5 se define el medio de comunicación de las plataformas (web y móvil), donde se establece que la comunicación será por medio de microservicios web realizados en node y estarán contenerizados en Docker. En este punto es importante mencionar que las plataformas tendrán microservicios compartidos y dedicados, el cual con ayuda de la redirección de NGINX, permitirán la alta eficiencia y disponibilidad con ayuda del servicio en la nube de Azure.
- Finalmente, en el punto 6 se muestra la base de datos a utilizar, el cual será una base de datos no relacional desplegada con MongoDB Atlas.

Teniendo la arquitectura a utilizar, se mostrarán los prototipos de la solución, prototipos seccionados según el tipo de usuario.

### **2.2.1 Administradores y Organizaciones**

Los usuarios administradores y organizaciones podrán tener acceso a páginas compartidas, las cuales principalmente muestran el mensaje organizacional, soluciones ofrecidas, contactos, visión y misión. A través del menú de esta pantalla y la opción de login, se permitirá ingresar a la página con diferentes tipos de usuarios.

Los prototipos mencionados son los siguientes:

- La Figura 2.2 hace referencia a la página de presentación, que comprende el contenido de estrategia de Mercadotecnia Digital dentro de un sitio web con lo cual la función es atraer futuros clientes potenciales.
- La Figura 2.3 muestra la sección de documentación para el desarrollador, el cual permitirá visualizar las APIs ofrecidas a las organizaciones para su desarrollo.
- La Figura 2.4 muestra la pantalla de Inicio de Sesión para administradores y organizaciones.

# ¿Aún no tiene infraestructura de gestión de entrega?

Adquiera nuestro servicio de delivery y gestione de manera inteligente sus entregas, mejorando los procesos, optimizando costos y midiendo la satisfacción de sus clientes.

Ingrese su correo electrónico

Regístrate

## Resultados

### Resultado 1



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### Resultado 2



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### Resultado 3



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### Resultado 4



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.



## Testimonios

"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eu turpis molestie, dictum est a, mattis tellus. Sed dignissim, metus nec fringilla accumsan, risus sem sollicitudin lacus, ut interdum tellus elit sed risus. Maecenas eget condimentum velit, sit amet feugiat lectus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Praesent auctor purus luctus enim egestas, ac scelerisque ante pulvinar. Donec ut rhoncus ex. Suspendisse ac rhoncus nisi, eu tempor urna. Curabitur vel bibendum lorem. Morbi convallis convallis diam sit amet lacinia. Aliquam in elementum tellus."

-Carlos Romero, Establecimiento Comercial-

Tenga su modelo de gestión de entregas por \$20/mes

¡Aproveche esta oportunidad única!

Ingrese su correo electrónico

Regístrate

## Preguntas Frecuentes

### ¿Lorem ipsum dolor sit amet, consectetur adipiscing elit?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eu turpis molestie, dictum est a, mattis tellus. Sed dignissim, metus nec fringilla accumsan, risus sem sollicitudin lacus, ut interdum tellus elit sed risus. Maecenas eget condimentum velit, sit amet feugiat lectus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Praesent auctor purus luctus enim egestas, ac scelerisque ante pulvinar. Donec ut rhoncus ex. Suspendisse ac rhoncus nisi, eu tempor urna. Curabitur vel bibendum lorem. Morbi convallis convallis diam sit amet lacinia. Aliquam in elementum tellus.



Figura 2.2 Pantalla de inicio para organizaciones y administradores [Autoría propia].

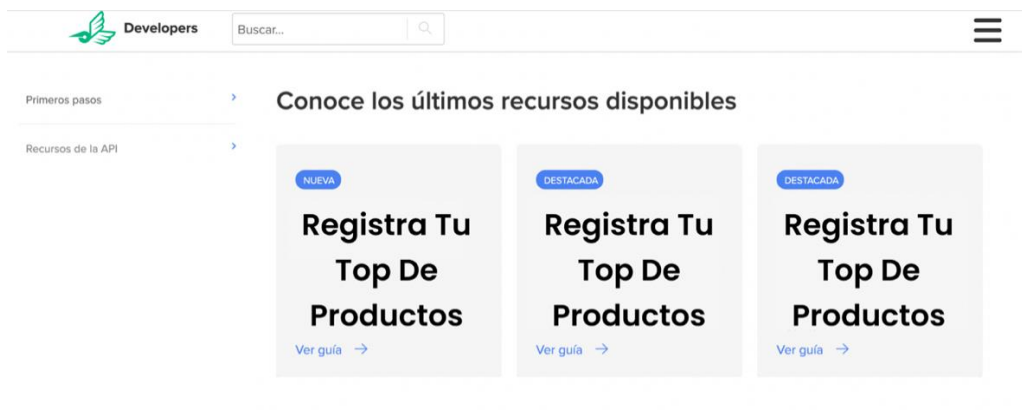


Figura 2.3 Pantalla de visualización de apis para organizaciones [Autoría propia].

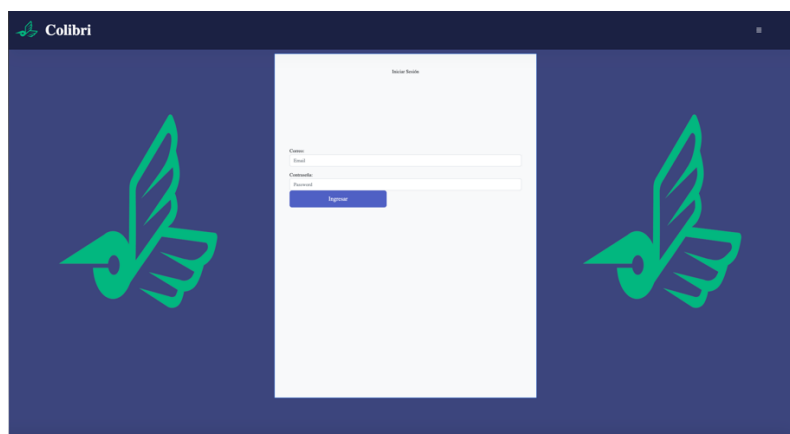


Figura 2.4 Pantalla de inicio de sesión para organizaciones y administradores [Autoría propia].

## 2.2.2 Administradores

Los usuarios administradores tienen la posibilidad de gestionar el acceso a usuarios del aplicativo móvil y web, de tal manera que podrán habilitar, deshabilitar o eliminar a los usuarios, según disponga. Este prototipo se muestra en la figura 2.5.



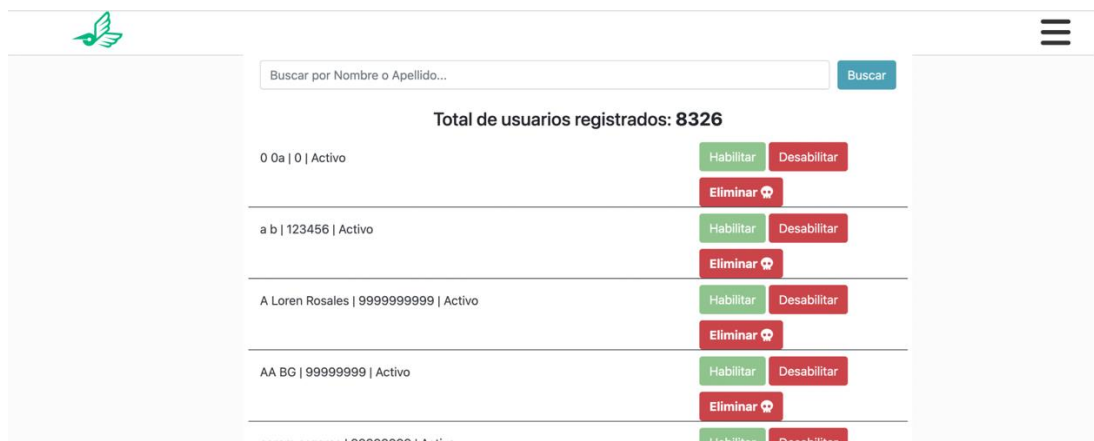


Figura 2.5 Pantalla de visualizaciones de usuarios [Autoría propia].

## 2.2.3 Organizaciones

Los usuarios organizaciones tendrán la capacidad de ver la pantalla de visualización de índices de costos de manera gráfica de su empresa, de tal manera que podrán observar el flujo de pedidos a través del tiempo. Además, podrán generar la Clave de Acceso necesaria para solicitar entrega por medio de repartidores. (figura 2.6).

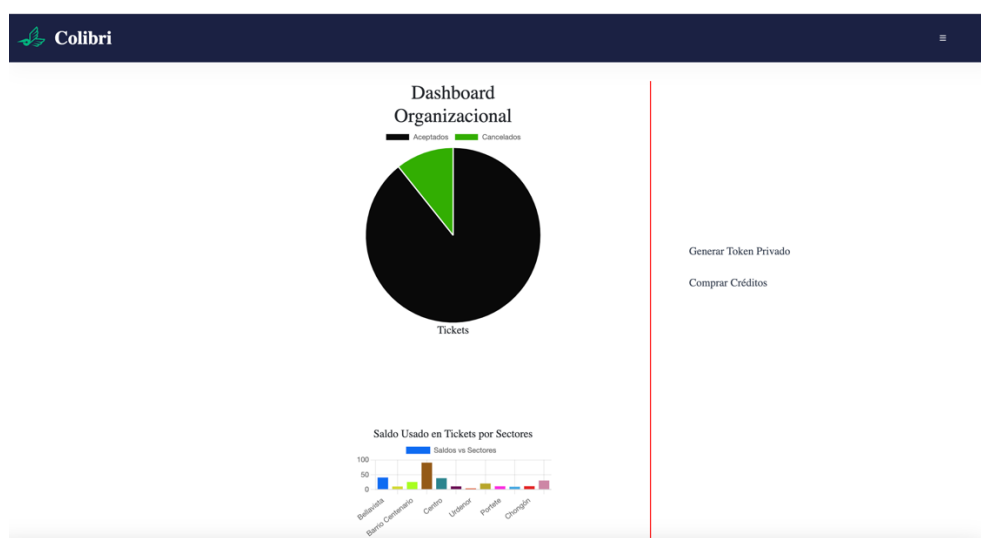


Figura 2.6 Pantalla de visualización de usuarios [Autoría propia].

## 2.2.4 Repartidores

Los usuarios repartidores usarán la plataforma móvil, el cual por medio de esta podrá aceptar entregas a domicilio, hacer seguimientos del destino y origen del pedido, y así

mismo como funciones como perfil, denegación del servicio, entre otras más. El flujo se presentará en las siguientes figuras.

La figura 2.7 muestra la pantalla de inicio de sesión, teniendo opciones para iniciar con redes sociales y con el inicio de la aplicación. La figura 2.8 muestra la pantalla principal, estando en el modo principal de pedidos según el filtro por prioridad.

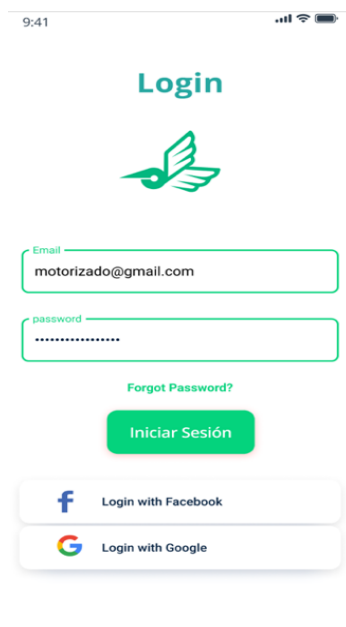


Figura 2.7 Pantalla de inicio de sesión [Autoría propia].

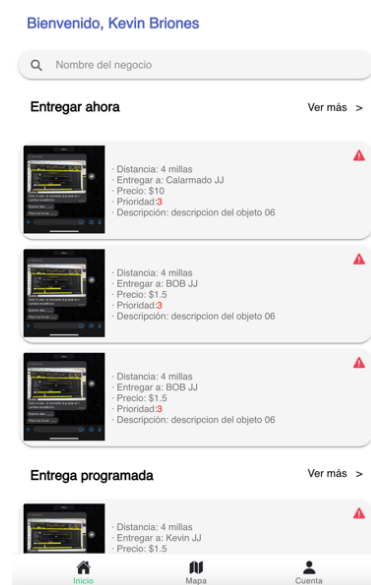


Figura 2.8 Pantalla de inicio con muestra de tickets [Autoría propia].

La figura 2.9 muestra la alerta posterior a seleccionar una solicitud de entrega a domicilio, el cual el repartidor tendría que confirmar si desea seleccionar la solicitud. En la figura 2.10 se muestra el tracking para el motorizado, hasta el punto donde estará el paquete a entregar.

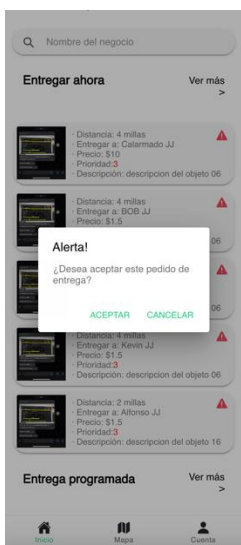


Figura 2.9 Pantalla de confirmación de entrega de tickets [Autoría propia].

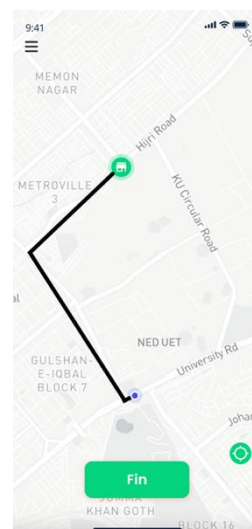


Figura 2.10 Pantalla de tracking de pedido [Autoría propia].

La figura 2.11 muestra el mensaje posterior a la finalización de la entrega del pedido por parte del repartidor, el cual tendrá que darle a finalizar para concluir le entrega. Luego, en la figura 2.12 se muestra el otro modo de búsqueda de solicitudes de pedido a través del mapa.

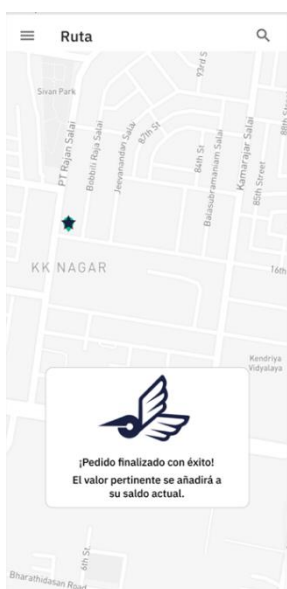


Figura 2.11 Pantalla de finalización de entrega de pedido [Autoría propia].

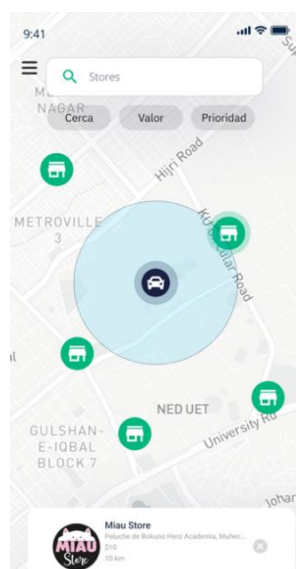


Figura 2.12 Pantalla de visualización de pedidos cercanos a motorizados [Autoría propia].

En la figura 2.13 se muestra el menú de opciones que tendrá la aplicación, y mediante este menú se podrá acceder al perfil, el mapa, y la página de inicio donde saldrán los tickets.



Figura 2.13 Visualización del menú de opciones [Autoría propia].

En la figura 2.14 se muestra la opción de perfil, donde el usuario podrá elegir si estará activo en la aplicación para recibir notificaciones de solicitudes de entrega.

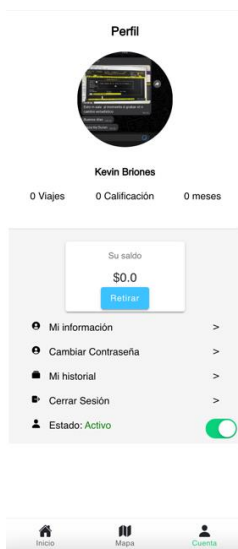


Figura 2.14 Pantalla de perfil de usuario [Autoría propia].

## 2.3 Evaluación

La arquitectura y prototipo fueron evaluados con ayuda del cliente, el cual nos dio recomendaciones que mencionaremos a continuación:

- La página de inicio para organizaciones y administradores (figura 2.4) debería tener una sección para visualizar las organizaciones afiliadas.
- La arquitectura de la solución le parece viable, pues está pensada para el fácil mantenimiento y migración hacia otros servicios en la nube que permitan desplegar aplicaciones web y servicios.

- En la plataforma web, desearía visualizar como quedaría el menú de opciones para los usuarios.
- Además, se dio énfasis en seguir pautas de otras aplicaciones.

Por ello se evaluaron las aplicaciones del mercado que ofrecen soluciones similares a la del presente proyecto, de modo que se va a describir diferentes alternativas a la solución planteada encontradas posterior a la evaluación:

- Soluciones como Uber Eats muestran imágenes para la revisión y selección de pedidos por parte de clientes, de modo que el usuario puede ver lo que le va a llegar a su domicilio previo a la selección y a la entrega, esto es útil incluso para los motorizados, puesto que con ello pueden conocer que será entregado por parte de las organizaciones y poder comparar con lo que el cliente solicitó. Esto es útil para integrarlo en nuestra solución, ya que el motorizado irá con más seguridad, y visualmente en la pantalla de “Tickets” traerá una mejor experiencia de usuario.

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

### 3.1 Análisis

En esta sección se detallan las tareas realizadas durante la ejecución del proyecto. Así mismo, se planteó los requerimientos funcionales y no funcionales que satisfacen al mercado laboral. A continuación, se visualiza el plan de implementación, pruebas y análisis de costo.

### 3.2 Desarrollo de la aplicación

Para el desarrollo de la aplicación se consideró en dividir en cuatro fases: la creación de base de datos, las vistas con respecto a su rol de usuario, la lógica de negocio y la integración de los componentes.

- **En la fase de la creación de base de datos:** el cliente nos mostró un sistema de datos que tenían creado en Django, al parecer no tenían todos los modelos creados, luego se procedió a analizar dicho sistema. Por consiguiente, se conversó con el cliente sobre el sistema y le sugerimos migrar a NodeJs. Por lo cual, al finalizar la migración se dialogó con el cliente y se mostraron los modelos y los servicios realizados.
- **En la fase de las vistas** se crearon prototipos en la plataforma FIGMA a medida que se avanzaba, se realizaba reuniones periódicas con el cliente mostrando los avances del prototipo. Por consiguiente, el cliente acepto los diseños a lo cual empezamos a la ejecución de dichos diseños. Pero, el cliente sugirió ciertos cambios en el diseño e implementación que produjeron ciertos inconvenientes con los tiempos establecidos. En los cambios realizados se mejora la experiencia del usuario mostrando mejoras en visualizaciones para el motorizado.
- **En la fase de la lógica del negocio:** mencionado en la primera fase que el cliente tenía base de datos y un poco de infraestructura de código. A continuación, se realizó un análisis de código. Además, en dicho código no se había considerado algunos puntos que deberían formar parte como requerimientos funcionales de la solución,

por lo cual, se evaluó poner aquellos requerimientos en el documento, e implementarlos en el desarrollo.

- Al requerimiento funcional sobre los tickets se consideró agregar el nivel de prioridad, tiempo, distancia, seguimiento y reporte del ticket por parte de la organización según acorde lo establecido por el cliente.
- Otra funcionalidad importante es el servicio de notificaciones cuando la organización crea un ticket, esto demanda un poco más del tiempo esperado.
- En la fase de las integraciones se realizaron pequeños cambios sobre los mensajes de errores por parte del servidor debido a que la modelo vista usa los mensajes del servidor para visualizar al cliente.

Finalmente se pudo concluir el proyecto con el tiempo estipulado, sin embargo, se considera que aquellas funcionalidades extras que fueron añadidas después del levantamiento de requerimientos forman parte de un tiempo adicional el cual deberá ser tomado para finalizarlos.

Tabla 3.1 Tabla de actividades [Autoría propia]

<b>N</b>	<b>Actividades</b>	<b>Fecha inicio</b>	<b>Fecha final</b>	<b>Participantes</b>
1	Establecer la arquitectura de software de la solución, análisis y definición de requerimientos.	18/10/2022	31/10/2022	Desarrolladores /Cliente
2	Creación de los prototipos del landing page, página web y aplicación móvil.	01/11/2022	07/11/2022	Desarrolladoras
3	Revisión de los prototipos y feedback.	08/11/2022	11/11/2022	Cliente
4	Diseño, migración, creación del modelo de base de datos y servicios.	12/11/2022	21/11/2022	Desarrolladores
5	Crear pantallas de la página web y landing page.	22/11/2022	30/11/2022	Desarrolladores
6	Crear pantallas de la aplicación móvil	01/12/2022	09/12/2022	Desarrolladores

7	Integración de los servicios con la aplicación móvil	09/12/2022	14/12/2022	Desarrolladores
8	Pruebas de integración de los servicios	15/12/2022	17/12/2022	Desarrolladores
9	Revisión de la integración	18/12/2022	21/12/2022	Cliente
10	Integración del módulo de pago en la página web	22/12/2022	10/01/2023	Desarrolladores
11	Plantilla de rol de pago en la página web	11/01/2023	14/01/2023	Desarrolladores
12	Documentación de los servicios	15/01/2023	17/01/2023	Desarrolladores

### 3.3 Plan de Implementación

### 3.4 Pruebas

En la definición del desarrollo del presente proyecto de materia integradora, como tal no se elaboró un plan de pruebas, sin embargo, entorno según la evolución del desarrollo de la solución se elaboraron pruebas para garantizar la viabilidad del proyecto, y corroborar que tanto los requerimientos funcionales y no funcionales cumplen en cabalidad con lo estipulado. Las pruebas que se realizaron son pruebas de integraciones.

#### 3.4.1 Pruebas de integración

El presente proyecto, al tener 3 tipos de usuarios, 3 distintas formas de acceder para estas personas, 2 plataformas, y distintas funcionalidades que como tal deben convivir para la total gestión de la solución brindada, tuvo el planteamiento e implementación de pruebas de integración para cada tipo de caso de uso que podría darse en el sistema. Estas pruebas de integración se realizaron de manera conjunta con ambos desarrolladores y en distintos ambientes para corroborar la funcionalidad total. Estas pruebas se realizaron mediante la herramienta de Interfaz de Programación de Aplicaciones Postman, y con las soluciones desarrolladas probadas en un ambiente en red LAN y en un ambiente de QA en la nube. Los casos de usos con los cuales se realizaron pruebas fueron los siguientes:

- Creación de usuarios para administradores, Organizaciones y repartidores.



- Creación de Tickets por parte de Organizaciones para entrega programada y en el momento.
- Gestión de tickets por parte de repartidores para la recogida y entrega de pedidos.
- Pago prepago por parte de las organizaciones para la compra de saldos con Paganini.
- Seguimiento al desplazamiento del motorizado al recoger y entregar el pedido.

### 3.5 Análisis de Costo

Para el análisis de costo del presente proyecto, habría que considerar el caso de que este proyecto hubiese sido propuesto con un presupuesto definido inicialmente por parte de los desarrolladores y el cliente.

Para ello consideramos que, de manera ficticia, inicialmente hubo un acuerdo entre el cliente y los participantes del desarrollo del presente proyecto, el acuerdo ficticio podría haber definido lo siguiente:

- El desarrollo será efectuado por parte de dos estudiantes politécnicos de la carrera de Ingeniería en Ciencias Computacionales, teniendo cada uno un pago de \$4 dólares por hora.
- El tiempo de duración del proyecto será por todo el tiempo que dura la materia integradora (4 meses o 480 horas laborales).
- No se está considerando el pago de costes por mantenimiento. En caso de que el cliente considere que debiera existir mejoras futuras, o mantenimiento posterior al tiempo pactado se deberá pactar un coste por el tiempo que dure este mantenimiento.

#### 3.5.1 Costos de Implementación

Considerando los puntos anteriores, en la tabla 3.1 las características desarrolladas y las horas utilizadas para realizar cada característica.

Tabla 3.2 Estimación de horas y costes según las características desarrolladas en la solución [Autoría propia]

Descripción de Actividad	Horas
Levantamiento de Requerimientos	40
Evaluación de características iniciales obtenidas de la elaboración previa del proyecto	24
Evaluación de arquitectura de software a utilizar	16
Elaboración de modelo de arquitectura y prototipado	56

<b>Desarrollo de modelos de la base de datos y microservicios</b>	80
<b>Desarrollo de aplicación móvil para repartidores</b>	120
<b>Desarrollo de aplicación web para organizaciones y administradores</b>	96
<b>Integración con Paganini</b>	24
<b>Pruebas de Integración</b>	16
<b>TOTAL</b>	<b>455</b>

La tabla 3.2, muestra que el tiempo total de horas utilizadas para el desarrollo del proyecto fueron de 455 horas, y considerando el total de hora pagada por desarrollador se muestra lo siguiente:

- El costo total del proyecto es de \$3640 dólares americanos.
- El pago para cada desarrollador es de \$1820 dólares americanos.
- Los costes dados son en un caso ficticio, ya que el desarrollo del presente proyecto de forma real no tiene ningún coste.

### **3.5.2 Costos de despliegue**

El coste total de despliegue es de \$200 dólares americanos, coste que se detallará a continuación:

- Al subir la aplicación móvil a las tiendas de Android, se tendría que realizar un pago anual de \$20 dólares americanos.
- Al subir la aplicación móvil a las tiendas de iOS, se tendría que realizar un pago anual de \$100 dólares americanos.
- Al subir la aplicación Web, se desplegarían 2 costes, el coste por el dominio (\$20 dólares americanos anuales), y el alojamiento digital en Digital Ocean (\$72 dólares americanos anuales).

## **4. CONCLUSIONES Y RECOMENDACIONES**

### **4.1 Conclusiones**

- El planteamiento inicial de la solución fue muy importante para generar resultados importantes en el proyecto, ya que al empezar definiendo la arquitectura del software, y de la misma manera la definición del prototipado, se pudo conseguir integralmente las demás características que rodeaban al planteamiento de la solución.
- Al tener definido de manera secuencial los objetivos que se debían realizar también fue una parte fundamental del proyecto de materia integradora, ya que esta secuencia permitió que el desarrollo siga un flujo en el que la solución fuera desarrollada principalmente para los usuarios que generarían una gran propuesta de valor (tales como repartidores y organizaciones).
- Al haber realizado una solución que consiste en la convivencia de 2 plataformas (plataforma Web y Móvil), teniendo como base al framework Angular, dado que ambas plataformas usan el mismo framework hizo que el coste y tiempo de desarrollo sea inferior al haber utilizado dos tecnologías diferentes.
- Las pruebas en el desarrollo de la solución fueron muy importantes, no solo para la generación de solicitudes y tickets si no de forma general; las pruebas de integración fueron cruciales ya que con ello se observaron el comportamiento de las solicitudes realizadas por parte del usuario final.

### **4.2 Recomendaciones**

- El presente proyecto al ser una continuación de un trabajo anterior, y al no cumplir con las condiciones necesarias para tenerlo como base en el presente proyecto, generó que el tiempo del desarrollo sea aún mayor, sin embargo, la evaluación para conocer que el proyecto no cumplía estas condiciones fue crítica, ya que, si no se hubiera hecho, el coste y el tiempo de desarrollo y mantenimiento hubiese sido aún mayor.
- Para la implementación de soluciones que serán ofrecidas en distintas plataformas y para distintos tipos de usuario, es importante evaluar que el modelo de base de datos, y el código del software estén correctamente estructurados, modularizados,

incrementales y adaptables. Esto para permitir el constante crecimiento a futuro de la solución a crear.

- Se debe considerar que cuando se necesite subir la presente solución (específicamente la aplicación móvil para la parte de motorizados) en tiendas para dispositivos iOS, deben considerar aspectos legales y éticos de estas tiendas, ya que esta tienda es más exigente con la legalidad, y por ende se complicaría al subir en producción para esos tipos de dispositivos.

# BIBLIOGRAFÍA

Cómo funciona Uber Eats. Accedido el 8 de diciembre, 2022, desde [https://about.ubereats.com/ec/es/?\\_ga=2.215043607.1849782920.1666725882-1672541555.1666725882](https://about.ubereats.com/ec/es/?_ga=2.215043607.1849782920.1666725882-1672541555.1666725882).

Comité de Operaciones de Emergencia Nacional. (2020). Informe de Situación COVID-19 Ecuador. Accedido el 10 de diciembre, 2022 desde <https://www.gestionderiesgos.gob.ec/wp-content/uploads/2020/03/Informe-de-Situación-No008-Casos-Coronavirus-Ecuador-16032020-20h00.pdf>

Sobre Nosotros. Accedido el 8 de diciembre, 2022, desde <https://soy-rappi.com.ec/nosotros>

Deliveroo, Página de Inicio. Accedido el 8 de diciembre, 2022, desde <https://soy-deliveroo.com>

¿Qué es mejor Rappi o Uber Eats? - Rappi vs Uber Eats para pedir y para trabajar. Accedido el 8 de diciembre, 2022, desde <https://soy-miracomosehace.com/mejor-rappi-uber-eats-rappi-vs-uber-eats-pedir-trabajar/>

Deliveroo, la plataforma de entregas inteligentes que adquirió Almacenes Tía. Accedido el 8 de diciembre, 2022, desde <https://quenoticias.com/noticias/almacenes-tia-deliveroo/>

# ANEXOS

## Manual de Usuario

### Especificaciones de la aplicación

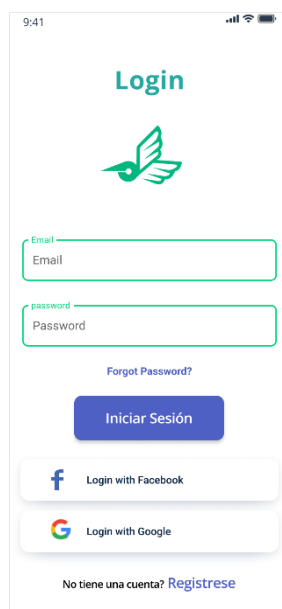
Sistema Operativo	Android   iOS
Aplicación de Desarrollo	Ionic – Angular
Permisos	Ubicación, cámara

### Descripción

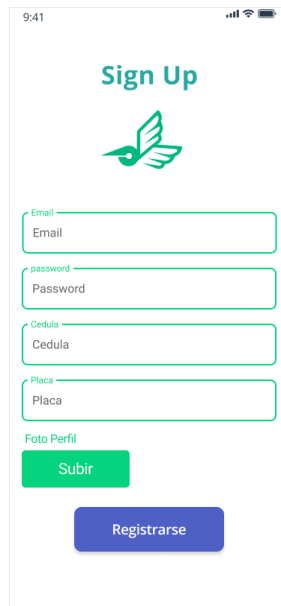
Esta aplicación sirve para las organizaciones y establecimientos que requieran entregar sus productos a domicilio.

### Motorizados

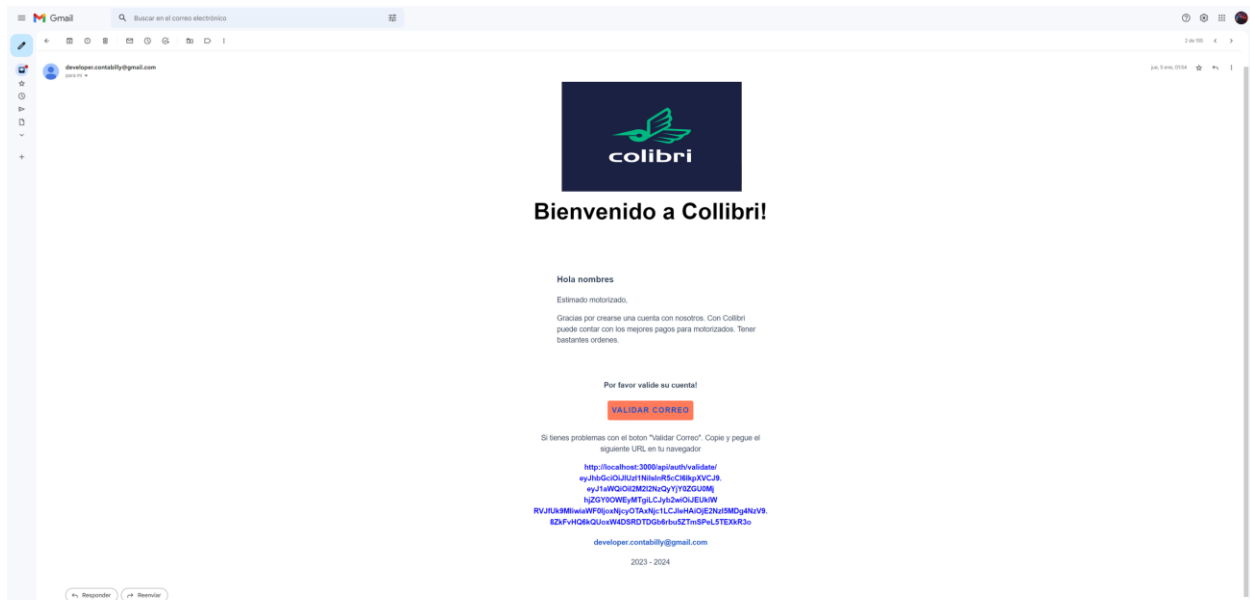
El repartidor presiona la aplicación a lo cual verá la pantalla de inicio. Donde podrá colocar sus datos para poder iniciar sesión dentro del sistema. También, si el repartidor no tiene cuenta podrá registrarse presionando en el botón “Regístrese”.



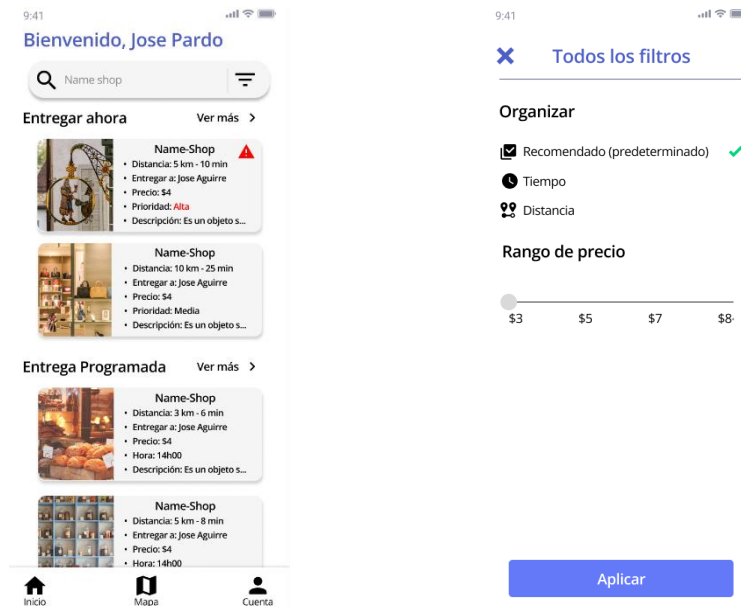
En caso, de que el motorizado se crea una nueva cuenta vera la siguiente pantalla. Por lo cual, debe rellenar los datos. En cuanto, pulse el botón de registrarse el sistema le notificara que se registró con éxito.



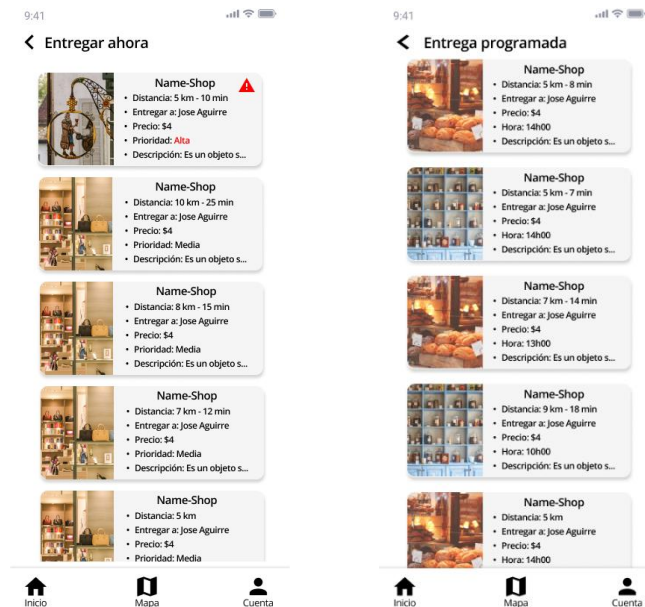
A lo cual, el administrador verificara los datos que sean correctos. En cuanto, el administrador apruebe al motorizado se le notificara a su correo para la validación de correo.



Luego, de validar su correo en el sistema. El motorizado podrá acceder al sistema con los datos registrados. En cuanto, inicie sesión podrá ver la pantalla de inicio. Además, podrá buscar tickets según lo que desea entregar. También, filtrar los tickets según la distancia, tiempo y precio.

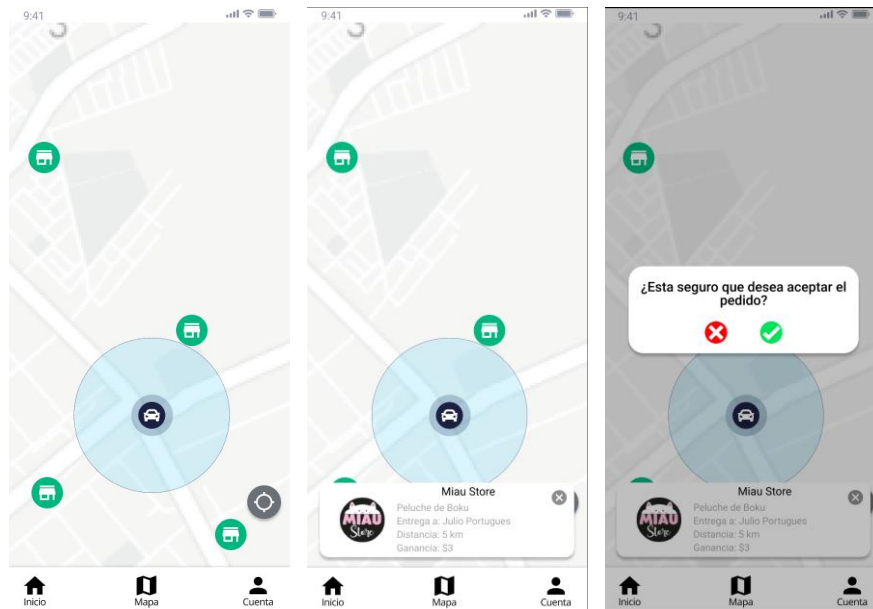


En la pantalla de inicio, podrá ver los tickets según la prioridad, la distancia y si es de entrega inmediata o programada. Cuando presione en la sección “ver más” sea entrega programada o inmediata. Podrá ver todos los tickets que se encuentran en esa sección.

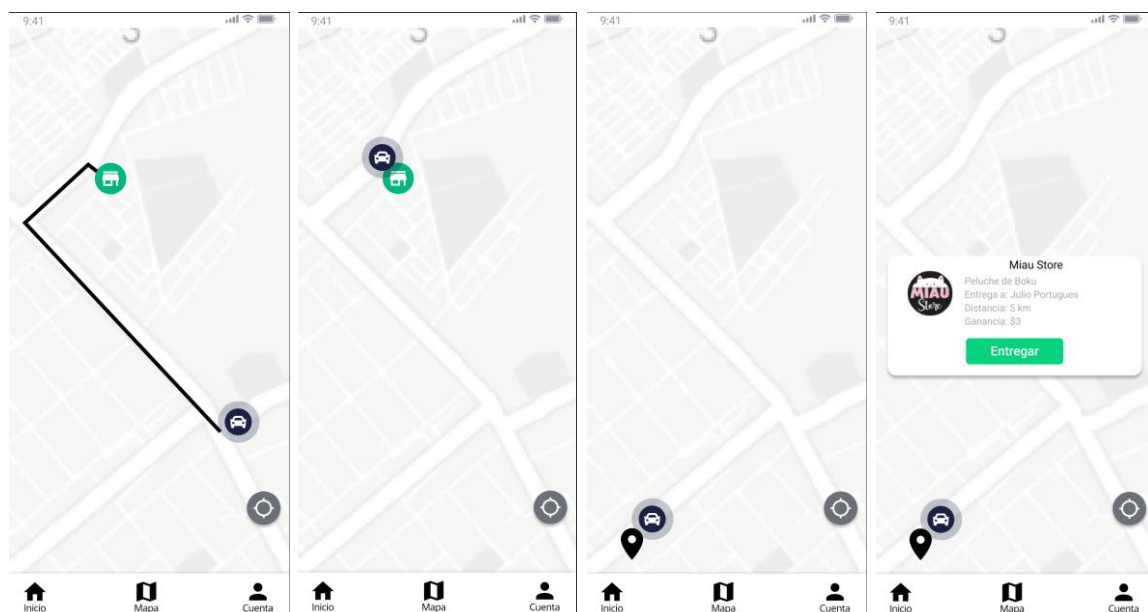


También, se puede dirigir a la pantalla de mapa donde podrá ver los tickets que se encuentra en su zona. Al presionar en los iconos de su mapa le saldrá una notificación en la parte inferior detallando brevemente el producto a realizar la entrega. Por lo cual, cuando presione sobre la notificación para tomar el pedido le saldrá un dialogo sobre si esta seguro de aceptar el pedido. En caso contrario, le pulsa en el icono de “X” para cancelar.

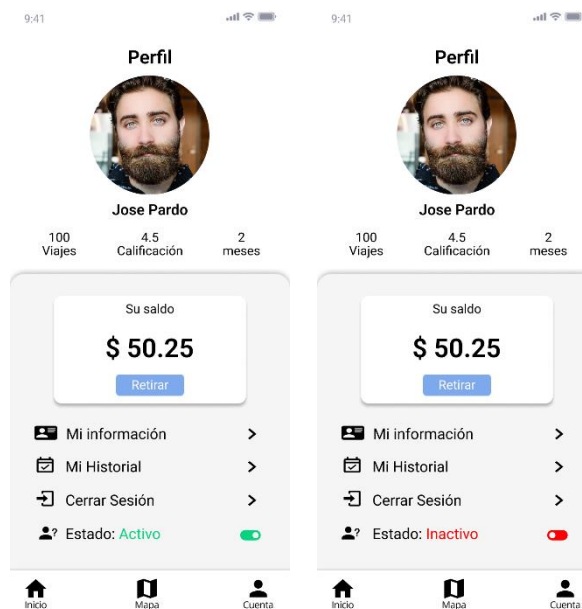




En cuanto, el repartidor acepte el pedido. Automáticamente, el sistema le trazara una ruta para que se dirija a recoger el pedido. En cuanto, haya llegado a la ubicación de recogida. La organización valida que se entregó con éxito al motorizado, por lo cual, el motorizado en el sistema le saldrá la ruta a donde debe entregar el pedido. Cuando el repartido llegue al destino. El sistema le notificara que esta cerca y le saldrá un dialogo donde entrego con éxito su pedido.



Por último, el motorizado podrá acceder a su perfil donde vera la cantidad de viajes realizados, su calificación y el tiempo que ha trabajado en la empresa. Además, puede habilitar/inhabilitar las notificaciones de cuando le lleguen nuevos pedidos.



## Manual de Implementación

### Descripción

Solución para la empresa Contabilly S.A., para generar una propuesta de valor afianzada en la acogida de empresas externas, las cuales desean contratar servicios de gestión de entrega a domicilio, ya sea para solicitar, al tiempo, motorizados para su entrega, o para integrar servicios de Colibrí de gestión de entrega, a sus aplicaciones empresariales.

### CONTROL DE CAMBIOS

Versión	Desarrolladores	Solicitante	Detalle
1.0	Estudiantes de Ingeniería de Software I y II - [Kevin Vásconez y Alfredo Calle]	Leonardo Andre Castro	Creación parcial de solución Colibrí
2.0	Estudiantes de MI - [Kevin Briones y Bryan Puchaicela]	Leonardo Andre Castro	Refactorización, migración, e implementación de módulos faltantes para la solución Colibrí

### ENTORNOS DE DESARROLLO

- Dashboard para organizaciones, página de visualización de APIs ofrecidos a organizaciones

### FRONTEND

Lenguaje: TypeScript, Sass

Librería: Angular

Framework: Ionic, Capacitor

Instalador: npm

Url: <https://gitlab.com/contabilly-frontend/ng-colibri.git>

## **BACKEND**

Lenguaje: Javascript

Framework: Node Js

Instalador: npm

Versión: 16.17.1

URL: <https://gitlab.com/contabilly-backend/nodejs-colibri.git>

## **Base de datos**

Lenguaje: NoSQL

Framework: MongoDB

## **DOCUMENTACIÓN BASE PARA NUEVOS DESARROLLOS**

Servicios API REST - Node js: <https://nodejs.org/en/docs/>

Aplicación Web - Angular: <https://angular.io/docs>

Aplicación móvil -Ionic: <https://ionicframework.com/docs>

## **LEVANTAMIENTO DE AMBIENTES DE DESARROLLO**

### **BACKEND**

Instalación de paquetes necesarios para compilar

- npm install o npm i

Levantamiento del back-end

- npm start

Variables de entorno

- MONGOOSE\_CONNECTION\_COLIBRI\_ADMIN (ruta de conexión de la base de datos)
- PRIVATE\_KEY\_TOKEN (clave para el token)
- KEY\_RESET (clave para el resetear password)
- SMTP\_HOST (smtp.gmail.com) por defecto
- MAIL\_FROM ([developer.contabilly@gmail.com](mailto:developer.contabilly@gmail.com)) correo empresa
- URL\_RESET\_PASSWORD (IP+/reset-password/)(sitio web de resetear contraseña)
- URL\_VALIDATE(IP+/api/auth/validate/)(sitio web de validación de usuario)
- ENDPOINT\_FILES (IP+/?filename)

NOTAS:

- Se recomienda instalar las dependencias del repositorio y configurar las variables de entorno.
- La IP mencionada con anterioridad si se está corriendo localmente o esta subido a la nube.
- Si desea visualizar los datos del back-end puede descargar la plataforma MongoDB Compass

Recomendación: Para subir a un ambiente de producción debe quitar el paquete de nodemon que se encuentra ubicado en dependencia. Además, configurar el script start de "nodemon" a "node".

## **APLICATIVO MÓVIL**

Requisitos para ejecución en ambiente local:

- NodeJS 18.12.1

- Ionic 6.20.4
- Capacitor 4.6.1

Instalación de dependencias obligatorias para ejecución en ambiente local:

- npm install -g @ionic/cli@6.20.4
- npm install @capacitor/cli@4.6.1 @capacitor/core@4.6.1
- npm install @capacitor/ios@4.6.1 @capacitor/android@4.6.1
- npm install (dentro de la ruta del repositorio del Proyecto, para instalar las dependencias del package.json).

NOTA: Se recomienda instalar las dependencias dentro de la ruta del repositorio clonado, y utilizar las versiones descritas, si se usan otras versiones podrían haber conflictos entre librerías.

Requisitos para ejecución en ambiente local, pero para ambientes nativos para iOS y Android.

- Android SDK
- Android Build Tools
- Java 8
- Xcode

Comandos para levantamiento en ambientes nativos:

- Comandos indicados previamente.
- Ionic cap sync (con flag --prod en caso de levantar con las variables de entorno del environment.prod.ts).
- Ionic cap open (android | ios)

NOTA: Para probar nativamente en iOS, se debe agregar en el Info.plist dentro de la ruta ./ios/App/App las recomendaciones lanzadas con el comando ionic cap sync, ya que estos son permisos puntuales que se deben solicitar al usuario para el correcto funcionamiento de la App.

Para Android dentro de la ruta ./android/app/src/main/res/values/strings.xml lo siguiente para el funcionamiento de servicios asociados a google, facebook y notificaciones:

```
<string name="fb_app_id">533259048860849</string>
<string name="fb_app_name">Colibri</string>
<string name="server_client_id">473678408921-
a3rta30c6ajau9bfnc2fjr2b7hmk.apps.googleusercontent.com</string>
<string name="app_client_id">473678408921-
a3rta30c6ajau9bfnc2fjr2b7hmk.apps.googleusercontent.com</string>
<bool name="fb_auto_log_app_events_enabled">>false</bool>
<bool name="fb_advertiser_id_collection_enabled">>false</bool>
```

En la ruta ./android/app/ se debe agregar el archivo google-services.json, el cual esta ubicado en la ruta del proyecto.

## PLATAFORMA WEB

Requisitos para ejecución en ambiente local:

- NodeJS 18.12.1
- Angular 14.2.8

Para la instalación de dependencias, solo se debe usar el comando npm install, dentro del directorio del proyecto.