

2019

**Cinthia Marleny
Huerta Valdivia**

MEJORAMIENTO EN CALIDAD Y USABILIDAD EN PLATAFORMA WEB

REPORTE FINAL PARA ACREDITAR RESIDENCIA PROFESIONAL DE LA CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Asesor interno: Renato Eduardo Reyes González
Asesor externo: Israel Antonio Fonseca

Evolución Digital LTD

Pabellón de Arteaga, Aguascalientes. Junio 201

AGRADECIMIENTOS.

Agradezco a muchas personas que llegara hasta aquí para poder realizar este trabajo y tal vez se me olvide agradecer a alguien así que una disculpa.

En primer lugar, me gustaría agradecerles mucho a mis papás que sin ellos no hubiera llegado hasta aquí ni podría concluir mis estudios, ya que siempre han estado conmigo en las buenas y en las malas apoyándome siempre de todas las maneras que pueden en lo que hago.

A mis amigos y compañeros de clase que a pesar de todo siempre nos apoyábamos y salíamos adelante en todas las materias, pasando muy buenos momentos dentro y fuera de clase.

A mis amigos que siempre me dieron apoyo y motivación por continuar con mi carrera, haciendo que fuera divertido seguir estudiando para que así pudiera llegar a titularme y concluir con mis estudios.

A mis maestros que gracias a ellos tengo los conocimientos necesarios que me hicieron llegar hasta aquí y siempre me apoyaron en todas mis dudas y trabajos que nos pedían realizar.

Agradezco a mis asesores externos Andrés Castaño Cuervo e Israel Fonseca que gracias a ellos logré comenzar y concluir este proyecto, brindándome sus conocimientos, siempre apoyándome en todo y respondiendo mis dudas.

A mi escuela, el Instituto Tecnológico de Pabellón de Arteaga, que me abrieron sus puertas, permitiéndome usar sus instalaciones para aprender ya que cuentan con todo lo necesario en sus instalaciones.

RESUMEN.

Este proyecto explica la manera en que se realizarán las pruebas de calidad y funcionamiento de la nueva plataforma web de la empresa Guepardos, la empresa ya cuenta actualmente con una plataforma web funcional, pero se requiere actualizar toda la interfaz de usuario para simplificar los pedidos que realicen los usuarios. La empresa Evolución Digital ya la creó, pero es necesario que se realicen todas las pruebas funcionales y de calidad necesarias para que no haya fallas al momento de subir la nueva plataforma ya en producción. Se realizarán pruebas de “Caja negra” como ejemplos de todas las pruebas realizadas y cuando hay fallas se realizan pruebas de “Caja blanca” para un mayor detenimiento en el código.

Al mismo tiempo se desarrolló el prototipo de una aplicación móvil nativa para iOS y Android, tomando como base los requerimientos específicos de la nueva aplicación web a la que se le realizaron pruebas. Esta aplicación se creó con las tecnologías más recientes para el desarrollo de aplicaciones móviles nativas sin tener que crear una app por cada plataforma móvil. Estas tecnologías son Flutter y Dart, ya que son las más actuales en desarrollo móvil. Los datos recibidos y enviados de esta app requieren como base de datos la misma que la plataforma web, la cual está basada en NoSQL JSON.



ÍNDICE

AGRADECIMIENTOS.....	2
RESUMEN.....	3
Lista de Tablas.....	6
Lista de Figuras.....	6
GENERALIDADES DEL PROYECTO.....	7
Introducción.....	7
Descripción de la empresa u organización y del puesto o área del trabajo del estudiante.....	8
Visión.....	8
Misión.....	8
Problemas a resolver, priorizándolos.....	8
Objetivos (General y Específicos).....	9
Objetivo general.....	9
Objetivos específicos.....	9
Justificación.....	9
MARCO TEÓRICO.....	10
Objetivos de las pruebas de software.....	10
Tipos de pruebas de software.....	11
Calidad en el Software.....	12
¿Qué son los negocios digitales?.....	13
Plataformas digitales.....	13
¿Qué es una plataforma y su objetivo?.....	13
Áreas que componen una plataforma digital.....	14
¿Qué es un servidor?.....	14
¿Qué es una aplicación móvil?.....	14
Tipos de aplicaciones móviles.....	15
Flutter.....	16
Lenguaje de programación Dart.....	16
Bases de datos NoSQL.....	17
JSON.....	18
DESARROLLO.....	19
Procedimiento y descripción de las actividades realizadas.....	19
Descripción del sistema propuesto.....	19
Perspectiva del producto.....	20
Restricciones y suposiciones.....	21
Requerimientos específicos.....	22



Requerimientos de Interfaz.....	22
Pantallas de la aplicación	24
Pruebas de caja negra y pruebas funcionales	30
Prototipo de aplicación móvil de la plataforma para usuarios	42
Plugins que se necesitaron.....	42
Paquetes importados para el uso de plugins	43
Capturas de pantalla en Android (Pixel 2 XL)	44
Aplicación ejecutada en iOS (iPhone XR)	50
Estructura de la aplicación.....	51
Respuesta del servidor sobre los datos enviados en el pedido	58
Test de la aplicación final.....	60
CRONOGRAMA DE ACTIVIDADES	62
Apoyo en testing y pruebas a plataformas.....	62
Revisión de calidad de productos en desarrollo	62
Capacitación para en lenguajes de desarrollo	62
Apoyo en prototipado de nuevos productos.....	62
Reporte de Issues de la plataforma	62
Gestionar y solucionar Issues de plataformas	62
RESULTADOS	63
CONCLUSIONES.....	64
COMPETENCIAS DESARROLLADAS	65
Competencias desarrolladas y/o aplicadas.....	65
FUENTES DE INFORMACIÓN	66
ANEXOS	67

Lista de Tablas

Tabla 1 Comparativa BD NoSQL y BD Relacionales	17
Tabla 2 Ejemplo de almacén de valores JSON	18
Tabla 3 Requisitos específicos	22
Tabla 4 Procesos del sistema	23
Tabla 5 Cronograma de actividades.....	62
Tabla 6 Resultados de Objetivos.....	63

Lista de Figuras

Ilustración 1 Caja Negra	11
Ilustración 2 Caja Blanca.....	12
Ilustración 3 Áreas o plataformas que conforman una plataforma digital	13
Ilustración 4 Área de trabajo de Flutter	16
Ilustración 5 Ejemplo de diseño en Flutter	Ilustración 6 Ejemplo de función Dart
	17
Ilustración 7 Ejemplo de código para mostrar un objeto JSON	19
Ilustración 8 Plataforma Guepardos actual	20
Ilustración 9 Propuesta de nueva plataforma Guepardos	20
Ilustración 10 Logo JSON	Ilustración 11 Logo Vue.js
Ilustración 12 Logo JavaScript	21
Ilustración 13 Página principal - Parte 1	24
Ilustración 14 Página principal - Parte 2.....	24
Ilustración 15 Página principal - Parte 3.....	25
Ilustración 16 Página principal - Parte 4.....	25
Ilustración 17 Página de acceso.....	25
Ilustración 18 Ventana recuperar contraseña.....	26
Ilustración 19 Ventana Registrarse	26
Ilustración 20 Página Nuevo Servicio - Parte 1	27
Ilustración 21 Página Nuevo Servicio - Parte 2	27
Ilustración 22 Página Mis Servicios - Activos	28
Ilustración 23 Página Mis Servicios - Historial.....	28
Ilustración 24 Página Mis facturas.....	29
Ilustración 25 Página Mi perfil	29
Ilustración 26 Página Actualizar contraseña	29
Ilustración 27 Plugins requeridos en la app.....	42
Ilustración 28 Paquetes importados a los archivos Dart.....	43
Ilustración 29 Pantalla principal en Nexus S	Ilustración 30 Pantalla Pedidos en Nexus S
	60
Ilustración 31 Pantalla principal en Pixel 3 XL	Ilustración 32 Pantalla Pedidos en Pixel 3 XL
	61

GENERALIDADES DEL PROYECTO

Introducción

La comunicación se ha visto como una tarea valiosa la cual era una herramienta para el intercambio comercial, así como la difusión de información o datos en años anteriores a las épocas actuales. La empresa Guepardos no se puede quedar atrás, necesita de estar en constante actualización, por lo cual ya está en proceso de creación su nueva plataforma web con un mejor entendimiento en la interfaz del usuario para realizar los pedidos fácilmente.

El objetivo del presente proyecto es realizar todas las pruebas necesarias en la nueva plataforma web Guepardos antes de que se use en producción. De igual manera se mostrará cómo es que se creó el prototipo de una aplicación móvil para usuarios de manera nativa para iOS y Android, con los mismos requerimientos específicos que la plataforma web, ya que la empresa no contaba con una aplicación nativa, sólo cuenta con la plataforma web y hoy en día, es más fácil realizar un pedido de un mensajero desde el fácil acceso de tu celular.

Descripción de la empresa u organización y del puesto o área del trabajo del estudiante.

Evolución Digital LTD, fundada en octubre de 2004 y se localizada en Medellín Colombia. Es una empresa de servicios dedicada a desarrollar software, prestar asesoría a personas naturales o jurídicas respecto a la compraventa y distribución de productos y servicios nacionales o extranjeros. Asesorías respecto al comportamiento general del mercado, promoción y distribución de los mismos.

Este proyecto surge de uno de sus clientes principales y más importante que es Guepardos.

Guepardos es una plataforma que tiene el respaldo de Mis Domicilios Ltda, empresa de mensajería que lleva más de 10 años en el mercado, ha realizado más de 140,000 servicios de mensajería y ha atendido a más de 15,000 personas o empresas.

Guepardos nació a mitad del año 2015 y partió de la necesidad de servicios Express, sus inicios fueron en conjunto con Mis Domicilios y a finales del 2015 e inicios del 2016 se dividió y comenzó su proceso de mensajería de manera independiente.

Actualmente sólo trabaja en la ciudad de Medellín, y atiende los mercados de e-commerce, Pymes, mediana empresas, restaurantes y personas naturales.

Visión

Para el año 2021 ser el operador logístico líder en servicios de mensajería a nivel nacional, reconocido principalmente por brindar un servicio oportuno, por su confiabilidad, calidad en seguridad en cada servicio, formando equipos de trabajo capacitados y con un alto desempeño que nos permita lograr satisfacer las necesidades de nuestros clientes.

Misión

Brindar a todos nuestros clientes soluciones y apoyo logístico que les permita obtener facilidad en la solicitud y seguimiento de cada servicio, desde el inicio hasta el final de la cadena de valor, de la mano de un personal capacitado, consiente y comprometido con el servicio prestado.

Las actividades desarrolladas en este proyecto se realizaron en el área de Laboratorio I+D, en donde se realizaron pruebas para la plataforma web de pedidos.guepardos.co y se desarrolló software para la empresa.

Problemas a resolver, priorizándolos.

En estos tiempos, es indispensable que las empresas se encuentren completamente comunicadas, contando con tecnologías actualizadas y en un

funcionamiento excelente. Sus plataformas deben tener el menor margen de error para que así, puedan brindar el mejor servicio al cliente y aumentar sus ventas.

La plataforma de Guepardos es utilizada diariamente para realizar servicios de mensajería en la ciudad de Medellín, actualmente ya cuenta con una plataforma web funcional donde se realizan los pedidos. Para mantenerse actualizados, crearon una nueva plataforma actualizada donde es necesario realizar las pruebas suficientes para que ésta no tenga fallas al momento de usarla en producción, por lo cual requieren que todo esté en muy buena calidad y completamente funcional.

Además de realizar las pruebas a la nueva plataforma, la empresa no cuenta con una aplicación móvil para usuarios y se requiere de apoyo en el desarrollo del prototipo.

Objetivos (General y Específicos)

Objetivo general

Realizar una revisión completa de calidad y testing a la nueva plataforma de Guepardos, haciendo todas las pruebas necesarias para no tener ningún inconveniente al usarla en producción.

Desarrollar el prototipo de una aplicación móvil nativa en iOS y Android para la futura aplicación de usuarios de Guepardos tendiendo una conexión directa a la plataforma para realizar los pedidos.

Objetivos específicos

- Mejorar los ciclos de desarrollo mediante el apoyo en testing.
- Realizar pruebas unitarias de los productos de Guepardos.
- Optimizar continuamente en términos funcionales.
- Obtener los conocimientos de los lenguajes de desarrollo para la aplicación.
- Crear prototipos para validar nuevas funcionalidades a incorporar a la plataforma.

Justificación

Este proyecto se crea con la finalidad de analizar la nueva plataforma de Guepardos mediante testing, ya que ésta se pondrá en producción muy pronto y no debe tener fallas de ningún tipo. Son muchos los beneficios de tener una plataforma en producción completamente funcional, como el tener todos los datos en orden, no tener problemas con los pedidos en cuanto a las fechas de entregado y recibido, o en las direcciones de entrega, tener satisfecho al cliente

por la buena comunicación sin problemas en la plataforma para mostrar los datos, entre muchas cosas más.

Se ahorra mucho tiempo y dinero al detectar estos problemas a tiempo, ya que si se pusiera en producción la nueva plataforma con errores, por ejemplo, se tendría que pagar al desarrollador por las horas en que se solucione el error urgentemente o si el pedido se ha perdido por no tener las direcciones correctas, responder al cliente sobre lo que ha pasado obteniendo una mala valoración y de esta manera tal vez perdiendo más clientes.

Las habilidades desarrolladas serán de mucha utilidad para el futuro, se aprenderá a obtener estrategias para el análisis del software y planificaciones de pruebas de software funcionales y habilidades en el razonamiento y solución de errores del sistema. Para el desarrollo del prototipo de la aplicación, se aprenderá a desarrollar aplicaciones nativas para iOS y Android en el lenguaje de Flutter y Dart usando conexiones a la base de datos de la plataforma Guepardos que es Json NoSQL

MARCO TEÓRICO

Todas las empresas de desarrollo de software se ven en constante perfeccionamiento de la calidad de sus plataformas logrando mejores resultados de la producción con la mayor eficiencia. Mostrarán los conceptos detallados del proyecto para su mejor comprensión, para cumplir con los objetivos es necesario realizar pruebas de software y desarrollar el prototipo de una aplicación móvil.

Las pruebas de software consisten en la dinámica de la verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo.

Objetivos de las pruebas de software.

La prueba de software es un elemento crítico para la garantía del correcto funcionamiento del software. Entre sus objetivos están:

- Detectar defectos en el software.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar casos de prueba que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Tipos de pruebas de software

En las **Pruebas Software Funcionales** encontraremos el comportamiento del sistema, subsistema o componente software descrito en especificaciones de requisitos o casos de uso, aunque también puede no estar documentado. Se definen a partir de funciones o características y su interoperabilidad con sistemas específicos, pudiendo ejecutarse en todos los niveles de pruebas.

Se consideran **Pruebas de Caja Negra** (“black-box testing”) puesto que valoramos el comportamiento externo del sistema. Las Pruebas de Seguridad o las Pruebas de Interoperabilidad entre sistemas o componentes son casos especializados de las pruebas funcionales.

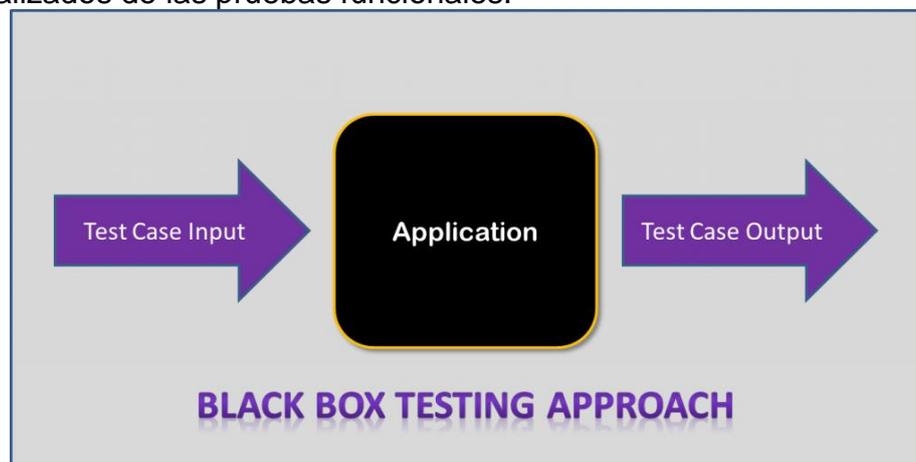


Ilustración 1 Caja Negra

En segundo lugar figuran las **Pruebas Software no Funcionales** que incluyen las pruebas de: Rendimiento, Carga, Estrés, Usabilidad, Mantenibilidad, Fiabilidad o Portabilidad, entre otras. Por tanto, se centran en características del software que establecen “cómo trabaja el sistema”.

Estas pruebas también pueden ejecutarse en todos los niveles de pruebas. Las características no funcionales del software se pueden medir de diversas maneras, por ejemplo, por medio de tiempos de respuesta en el caso de pruebas de rendimiento o por número máximo de sesiones en pruebas de estrés.

En tercer lugar, tenemos las **Pruebas Software Estructurales**. Nuevamente pueden ejecutarse en todos los niveles de pruebas y encajan muy bien si hemos utilizado técnicas de especificación de la estructura o arquitectura del Software. Es posible aplicar técnicas estáticas de análisis de código.

Es especialmente habitual utilizar herramientas de apoyo para calcular la cobertura del código en el caso de Pruebas de Componentes o en Pruebas de Integración de Componentes. Puesto que indagamos en el comportamiento interno, estas pruebas se denominan también **Pruebas de Caja Blanca** (“white-box testing”).

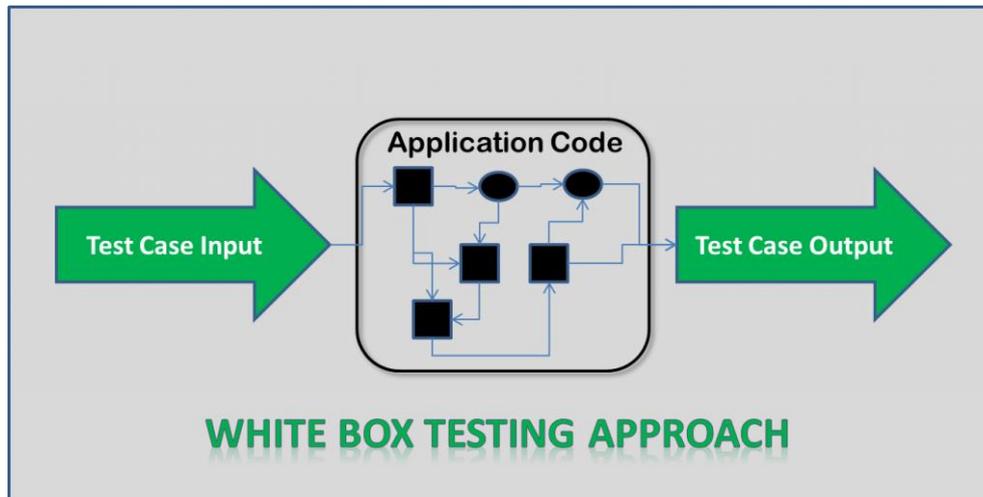


Ilustración 2 Caja Blanca

Finalmente, las **Pruebas de Regresión** consisten en volver a probar un componente, tras haber sido modificado, para descubrir cualquier defecto introducido, o no cubierto previamente, como consecuencia de los cambios. Los defectos pueden encontrarse tanto en el software que se ha cambiado como en algún otro componente. Se ejecutan cuando se cambia el software o su entorno. El criterio para decidir la extensión de estas **Pruebas de Regresión** está basado en el riesgo de no encontrar defectos en el software que anteriormente estaba funcionando correctamente.

Calidad en el Software

La totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas. Se puede decir que el software tiene calidad si cumple o excede las expectativas del usuario en cuanto a:

1. Funcionalidad (que sirva un propósito),
2. Ejecución (que sea práctico),
3. Confiabilidad (que haga lo que debe),
4. Disponibilidad (que funcione bajo cualquier circunstancia)
5. Apoyo, a un costo menor o igual al que el usuario está dispuesto a pagar.

Resumiendo, podemos decir, que la calidad de software se refiere a: "Los factores de un producto de software que contribuyen a la satisfacción completa y total de las necesidades de un usuario u organización".

En el desarrollo del prototipo de la aplicación móvil se necesitaron las siguientes tecnologías para que pueda ser compatible de manera nativa en los diferentes sistemas operativos móviles que son iOS y Android.

¿Qué son los negocios digitales?

Es el desarrollo de una actividad o un servicio a través de canales y plataformas digitales. Y se pueden llamar digital, en la nube u online. El negocio digital es la creación de nuevos diseños de negocios al desvanecer los límites entre los mundos digital y físico debido a la convergencia de personas, negocios y cosas. Para los fabricantes, las empresas digitales pueden significar fábricas conectadas y la Internet industrial.

Plataformas digitales

Gartner en un estudio desarrollo y explica las diferentes plataformas con sus componentes que requiere un negocio, haciendo mención que no siempre se requiere todas las plataformas aquí señaladas, está en función del tipo de negocio y sus necesidades.

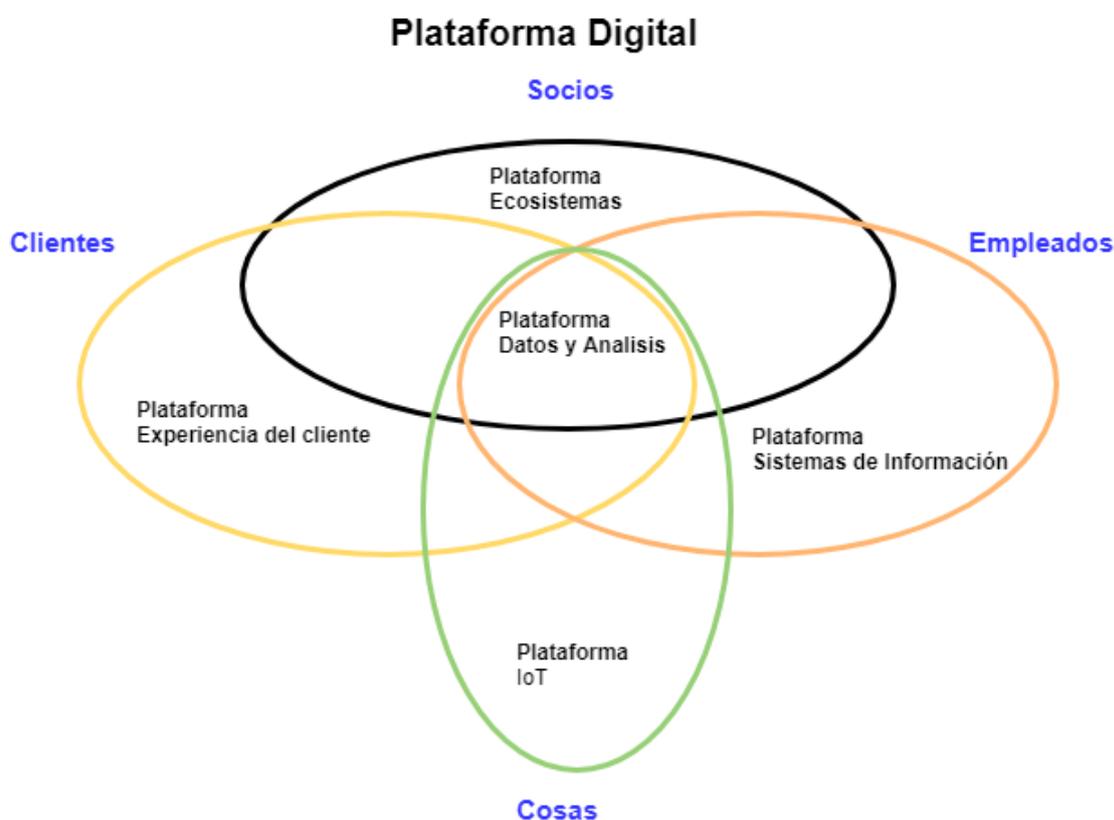


Ilustración 3 Áreas o plataformas que conforman una plataforma digital

¿Qué es una plataforma y su objetivo?

Definición generada en el estudio realizado por Gartner

“La palabra plataforma, significa que el área se basa en principios y arquitecturas basados en servicios.” El objetivo es crear un conjunto interoperable de servicios que se puedan reunir para crear aplicaciones, apps y flujos de trabajo. La plataforma de tecnología empresarial digital se describe a través de la lente de las aplicaciones y los componentes de capacidad empresarial

Áreas que componen una plataforma digital

Un negocio digital cuenta con el respaldo de una plataforma tecnológica en cinco áreas, Ilustración 3.

Plataforma de sistemas de información: Soporta la administración y las operaciones como ERP y sistemas centrales.

Plataforma de experiencia del cliente: contiene los principales elementos del cliente, como los portales de clientes y ciudadanos, el comercio multicanal y las aplicaciones para clientes.

Plataforma de datos y análisis: contiene capacidades analíticas y de gestión de la información. Los programas de administración de datos y las aplicaciones analíticas alimentan la toma de decisiones impulsada por datos y los algoritmos automatizan el descubrimiento y la acción.

Plataforma de Internet de las cosas (IoT): conecta los activos físicos para el monitoreo, la optimización, el control y la monetización. Las capacidades incluyen conectividad, análisis e integración a sistemas de tecnología central y operativa (OT).

Plataforma de ecosistemas: admite la creación de ecosistemas externos, mercados y comunidades, y la conexión a estos. La gestión, el control y la seguridad de la API son elementos principales.

¿Qué es un servidor?

El servidor es el sistema encargado de transmitir la información para el correcto funcionamiento de la organización.

Un servidor es un ordenador u otro tipo de equipo informático encargado de suministrar información a una serie de clientes, que pueden ser tanto personas como otros dispositivos conectados a él.

El servidor actúa como el gran cerebro del sistema informático de la empresa, pues constituye el elemento capaz de albergar la información necesaria para el funcionamiento de cada departamento. El server tiene una gran utilidad en el funcionamiento de la empresa, ya que es capaz de llevar a cabo funciones tanto de carácter físico (funcionamiento de las máquinas y los aparatos necesarios para la actividad empresarial) como a nivel de información, ya que registra, alberga y envía la información que los distintos clientes le van solicitando.

¿Qué es una aplicación móvil?

Es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Las aplicaciones permiten al usuario efectuar un conjunto de tareas de cualquier tipo profesional, de ocio, educativas, de acceso a servicios, etc; facilitando las gestiones o actividades a desarrollar.

El desarrollo de aplicaciones móviles es el conjunto de procesos y procedimientos involucrados en la escritura de software para pequeños dispositivos inalámbricos de cómputo, como teléfonos inteligentes o tabletas.

El desarrollo de aplicaciones móviles es similar al desarrollo de aplicaciones web, y tiene sus raíces en el desarrollo de software más tradicional. Una diferencia fundamental, sin embargo, es que las aplicaciones (apps) móviles a menudo se escriben específicamente para aprovechar las características únicas que ofrece un dispositivo móvil en particular.

Una forma de asegurar que las aplicaciones muestran un rendimiento óptimo en un dispositivo determinado es desarrollar la aplicación (app) de forma nativa en ese dispositivo. Esto significa que, a un nivel muy bajo, el código se escribe específicamente para el procesador de un dispositivo particular. Cuando una app necesita ejecutarse en varios sistemas operativos, sin embargo, hay poco código que puede ser reutilizado desde el desarrollo inicial. La aplicación debe ser esencialmente reescrita para cada dispositivo específico.

Tipos de aplicaciones móviles

Aplicaciones Nativas

Las aplicaciones nativas son aquellas desarrolladas bajo un lenguaje y entorno de desarrollo específico, lo cual permite, que su funcionamiento sea muy fluido y estable para el sistema operativo que fue creada. Utilizan los recursos tanto del sistema como del hardware, permite ser publicada en tiendas para su distribución y en su mayoría, no necesitan estar conectadas a Internet para su funcionamiento.

Aplicaciones Web

Son aquellas desarrolladas usando lenguajes para el desarrollo web como lo son html, css y javascript y un framework para el desarrollo de aplicaciones web. Se podría decir que este tipo de aplicaciones es muy usada para brindar accesibilidad a la información desde cualquier dispositivo, sin importar el sistema operativo, ya que solo se necesita contar con un navegador para acceder a esta. Pueden ser utilizadas desde cualquier dispositivo sin importar el sistema operativo.

Puede que requiera un coste para su desarrollo, pero este puede ser mínimo en comparación con las nativas y no requieren de ninguna aprobación para su publicación.

Aplicaciones Híbridas

Este tipo de aplicaciones se desarrolla utilizando lenguajes de desarrollo web y un framework dedicado para la creación de aplicaciones híbridas. La facilidad que brinda este tipo de desarrollo es que no hay un entorno específico el cual hay que utilizar para su desarrollo y la mayoría de las herramientas son de uso

gratuito, también pudiendo integrarlo con las herramientas de aplicaciones nativas.

Usan de los recursos del dispositivo y del sistema operativo. El costo de desarrollo puede ser menor que el de una nativa, son multiplataforma y permite distribución a través de las tiendas de su respectiva plataforma.

Flutter

Es el marco de la interfaz de usuario móvil de Google para crear interfaces nativas de alta calidad en iOS y Android en un tiempo récord. Flutter trabaja con el código existente, es utilizado por desarrolladores y organizaciones de todo el mundo, y es gratuito y de código abierto. Flutter es un motor de aplicaciones que puede incrustar en una aplicación existente o usar para una aplicación completamente nueva.

Flutter incluye un conjunto completo de widgets que brindan experiencias de píxeles perfectos tanto en iOS como en Android. Y permite la realización definitiva de Material Design, el sistema de diseño abierto de Google para experiencias digitales.

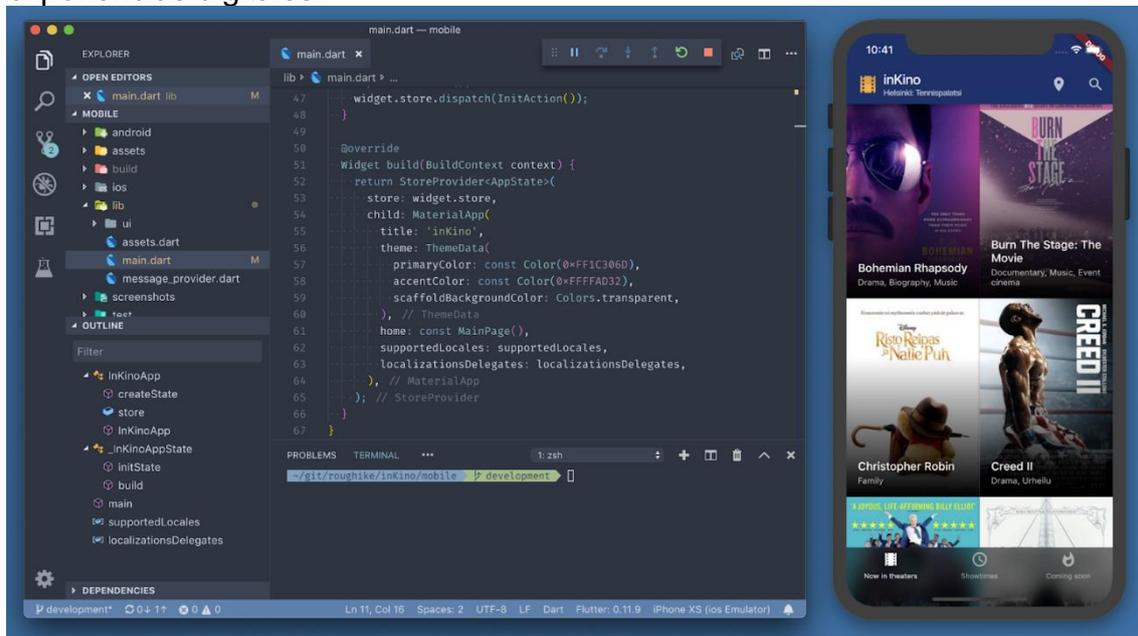


Ilustración 4 Área de trabajo de Flutter

Lenguaje de programación Dart

Dart es un lenguaje optimizado para el cliente, para aplicaciones rápidas en cualquier plataforma. Dart potencia a Flutter, el kit de herramientas de rápido crecimiento para crear hermosas experiencias nativas para iOS y Android desde una base de código único; También es el idioma utilizado por algunos de los proyectos más grandes en Google, como Google Ads.

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ),
    body: Column(
      children: <Widget>[
        HeroImage(),
        ...todaysDiscounts,
        for (var d in items) Item(d),
        ActionBar(),
      ],
    ),
  );
}
```

Ilustración 5 Ejemplo de diseño en Flutter

```
fetchTemperature() async {
  // This call is non-blocking
  // The UI thread will continue
  // to render with no locks
  final response = await
    http.get('https://my/weather');

  if (response.statusCode == 200) {
    temp.setText(response.body);
  } else {
    temp.setText('Unknown temp.');
```

Ilustración 6 Ejemplo de función Dart

Bases de datos NoSQL

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, su funcionalidad y el rendimiento a escala. Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en-memoria y búsqueda.

NoSQL vs. Bases de datos relacionales

La tabla 1, ofrece una breve comparación entre las funcionalidades de NoSQL y las bases de datos relacionales:

Tabla 1 Comparativa BD NoSQL y BD Relacionales

Características	BD NoSQL	BD Relacionales
Rendimiento	Alto	Bajo
Confiabilidad	Pobre	Bueno
Disponibilidad	Bueno	Bueno
Consistencia	Pobre	Bueno
Almacenamiento de datos	Optimizado para grandes datos	Mediano a grande
Escalabilidad	Alta	Alta (Pero más cara)

Cabe señalar que esta tabla muestra una comparación a nivel de la base de datos, no sobre los diversos sistemas de gestión de bases de datos que implementan ambos modelos. Estos sistemas proporcionan sus propias técnicas

patentadas para superar los problemas y deficiencias encontradas en el sistema, además de intentar mejorar significativamente el rendimiento y la fiabilidad.

Key Value Store

En el tipo de almacén Key Value, se utiliza una tabla hash en la que una clave única apunta a un elemento. Las claves pueden ser organizadas por grupos clave lógicos, requiriendo solamente estas claves para ser únicas dentro de su propio grupo. Esto permite tener claves idénticas en diferentes grupos lógicos. La tabla 2, muestra un ejemplo de un almacén de valores clave, en el que la clave es el nombre de la ciudad y el valor es la dirección de Ulster University en esa ciudad.

Tabla 2 Ejemplo de almacén de valores JSON

Key	Value
"Belfast"	{"University of Ulster, Belfast Campus, York Street, Belfast, BT15 1ED"}
"Coleraine"	{"University of Ulster, Coleraine Campus, Cromore Road, Co. Londonderry, BT52 1SA"}

Algunas implementaciones del almacén de valores clave proporcionan mecanismos de almacenamiento en el caché, lo que mejora en gran medida su rendimiento.

Todo lo que se necesita para hacer frente a los elementos almacenados en la base de datos: es la clave. Los datos se almacenan en una forma de una cadena, JSON o BLOB (objeto grande binario). La base de datos NoSQL más famosa que se construye en un almacén de valores clave Key Value es DynamoDB de Amazon.

JSON

Es un formato de texto para la serialización de datos estructurados. Deriva de los Objetos Literales de Javascript y está definido en la tercera edición del estándar de Lenguaje de programación ECMAScript [ECMA]. JSON puede representar cuatro tipos primitivos (cadenas, números, booleanos, valores nulos) y dos tipos estructurados (objetos y arreglos).

- Una Cadena es una secuencia de ceros o más caracteres Unicode.
- Un Objeto es una colección desordenada de cero o más pares nombre:valor, donde un nombre es una cadena y un valor es una cadena, número, booleano, nulo, objeto o arreglo.
- Un Arreglo es una secuencia desordenada de ceros o más valores.

```
var objetoJavascript = {"hora": 11, "dia": 1, "mes": 7, "año": 2014},
    textoJson = JSON.stringify(objetoJavascript , filtro, 4);
console.log(textoJson);

/*
Logs:
{
  "dia": 1,
  "mes": 7,
  "año": 2014
}
*/
```

Ilustración 7 Ejemplo de código para mostrar un objeto JSON

DESARROLLO

Procedimiento y descripción de las actividades realizadas.

Se tiene una plataforma web donde se analizarán todos los aspectos necesarios desde la creación del sistema, revisando la descripción del proyecto, los requerimientos funcionales y no funcionales del sistema y las pruebas funcionales realizadas, en este caso de Caja Negra ya que es lo que el usuario puede ver y lo que al cliente le interesa, en caso de encontrar errores se procederá a revisar el funcionamiento del código haciendo pruebas de Caja Blanca.

Descripción del sistema propuesto

La empresa Guepardos necesita una actualización de su plataforma actual, en donde la manera en que se realizan los pedidos de mensajería cambia de estructura para que así sea más fácil de entender y utilizar para el usuario. Actualmente se usa en producción el formulario mostrado en la Ilustración 8 que es muy útil, aunque podría ser algo confuso para algunos usuarios.

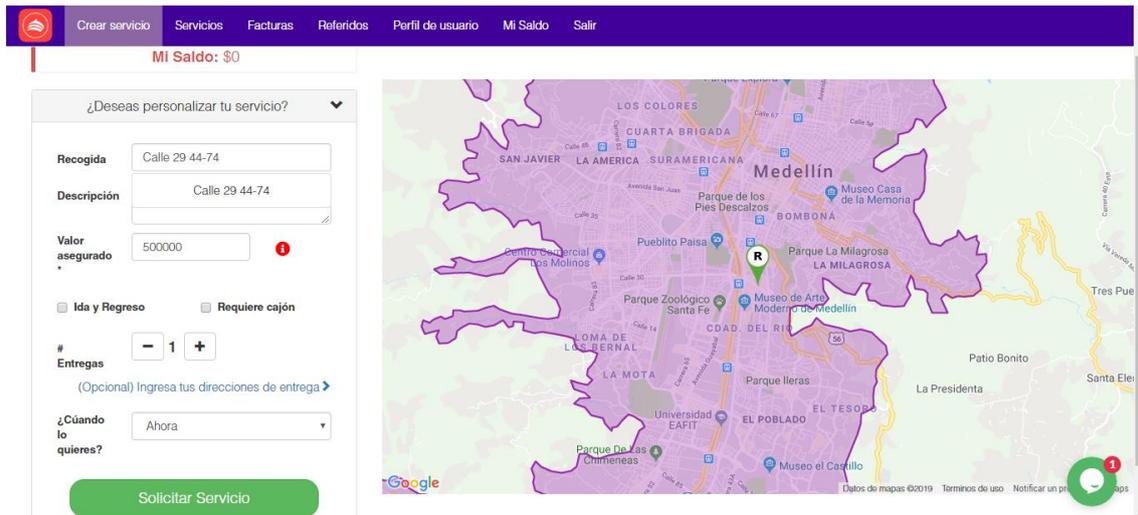


Ilustración 8 Plataforma Guepardos actual

Esta es la pantalla principal y más utilizada por el usuario, así que debe estar en un completo funcionamiento, el cliente requiere que se realice un formulario más sencillo de usar, donde sólo nos pida los datos básicos para realizar el pedido de mensajería. Se ha creado una plataforma con las especificaciones del cliente y resolviendo esos inconvenientes, como vemos en la Ilustración 8, el formulario de pedido es más simple y por lo tanto es menos probable que el usuario no se confunda al momento de realizar sus pedidos.

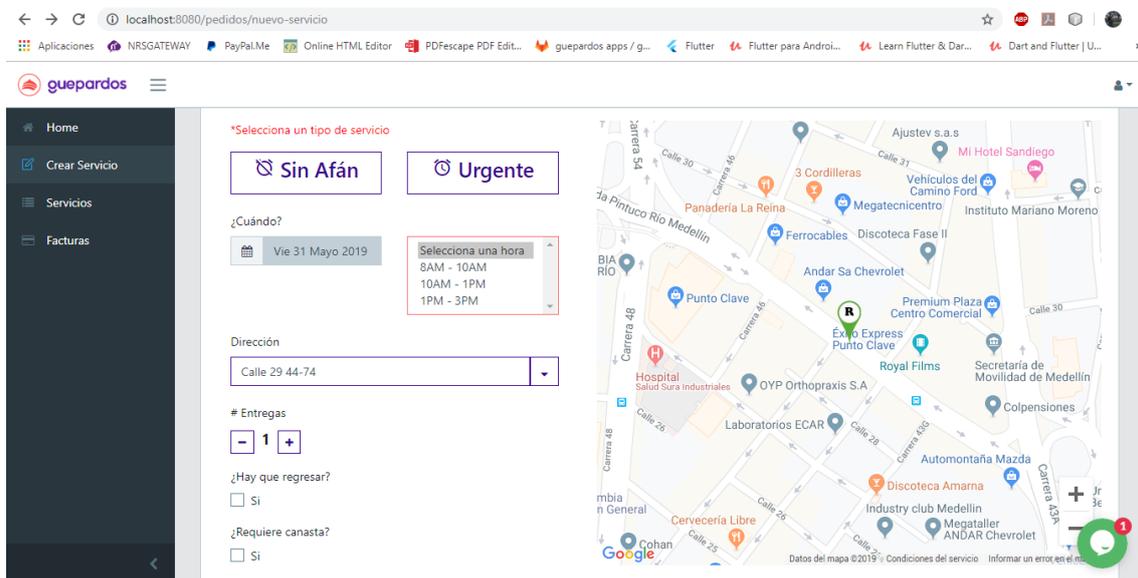


Ilustración 9 Propuesta de nueva plataforma Guepardos

Perspectiva del producto

El sistema se crea en una plataforma web usando el lenguaje de Javascript, usando el framework de Vue.js para crear las interfaces de usuario y usando como base de datos el formato NoSQL Json.

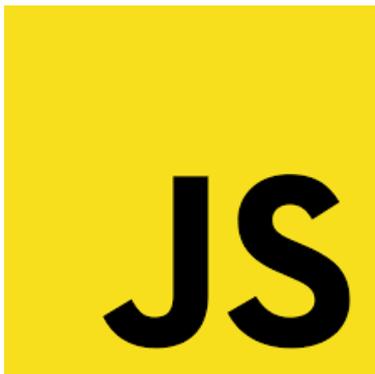


Ilustración 10 Logo JSON



Ilustración 11 Logo Vue.js

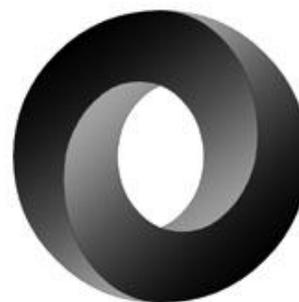


Ilustración 12 Logo JavaScript

El sistema deberá ingresar con en el login con la base de datos de la plataforma actual, el usuario podrá cambiar sus datos y contraseña, realizar pedidos mediante el nuevo formulario y podrá ver sus servicios activos, historial de servicios y facturas dentro de esta nueva plataforma.

Restricciones y suposiciones

- El cliente deberá tener sus datos personales de su perfil completos registrados en la plataforma.
- Los pedidos tienen que llevar mínimo la dirección de recogida y la fecha.
- Los pedidos del tipo Urgente sólo se pueden realizar antes de las 3 pm.
- Los pedidos Sin Afán se pueden realizar para cualquier fecha válida.

Requerimientos específicos

Tabla 3 Requisitos específicos

No. Del requisito	Nombre del requisito	Fuente del requisito	Prioridad del requisito			Clasificación
			Alta	Medio	Baja	
1	El usuario podrá ingresar	Datos de acceso				Funcional
2	El usuario consultará los datos de pedido	Base de datos				Funcional
3	El sistema registrará los pedidos	Datos enviados				No Funcional
4	El sistema enviará los datos a la base de datos	Solicitud al servidor				Funcional
5	El usuario podrá registrarse	Datos del usuario				Funcional
6	El usuario podrá cambiar su contraseña	Contraseña nueva				Funcional
7	El usuario podrá recuperar su contraseña	Correo registrado				Funcional
8	El usuario podrá actualizar sus datos	Datos del usuario				Funcional
9	El usuario podrá consultar pedidos antiguos	Base de datos de plataforma				Funcional
10	El usuario podrá consultar sus pedidos activos	Base de datos de plataforma				Funcional
11	El usuario podrá consultar sus facturas	Base de datos de plataforma				Funcional
12	La base de datos se usará de la plataforma en JSON NoSQL	Diseño de BD				No Funcional
13	El programa se desarrollará en el lenguaje JavaScript y Vue.js	Diseño de Interfaz				No Funcional
14	El sistema validará el correo al registrarse	Validación de errores				Funcional
15	El sistema permitirá elegir una fecha urgente	Fecha válida actual				Funcional
16	El sistema permitirá elegir una fecha sin afán	Fecha válida actual				Funcional
17	El sistema tendrá un menú para acceder a las demás pantallas	Diseño de Interfaz				No Funcional

Requerimientos de Interfaz

Las entradas al sistema serán los datos de usuario que se ingresen en el login, en este momento se comparan las credenciales con los datos del usuario registrado previamente en la base de datos de la plataforma. Ya accediendo los datos del usuario se guardarán localmente para usarlos más tarde al igual que el token se guarda para hacer futuras solicitudes al servidor y obtener los datos solicitados de la base de datos.

La parte más importante es la de crear pedidos, es necesario que el usuario proporcione una dirección de recogida, el tipo de pedido, ya sea "Urgente" o "Sin Afán". Especificar la fecha, hora del pedido, número de entregas y si se desea la

dirección de entrega, entre otras opciones extra como si necesitará canasta y/o si necesitará el cliente que regrese el mensajero.

Tabla 4 Procesos del sistema

Entradas	Proceso	Salidas
<ul style="list-style-type: none"> - Datos de usuario - Credenciales - Tipo de pedido - Fecha del pedido - Dirección - Número de entregas - Descripción (opcional) - Dirección de entrega (opcional) 	<ul style="list-style-type: none"> - Se registran los usuarios - Ingresan con sus credenciales - Se crean los pedidos - Se envían los datos a la base de datos - Se actualizan datos de usuario y contraseña - Se recupera la contraseña 	<ul style="list-style-type: none"> - Se obtiene la información de los pedidos - Se consultan los servicios activo - Se consulta el historial de servicios - Se consultan las facturas

Pantallas de la aplicación

Página principal



Ilustración 13 Página principal - Parte 1

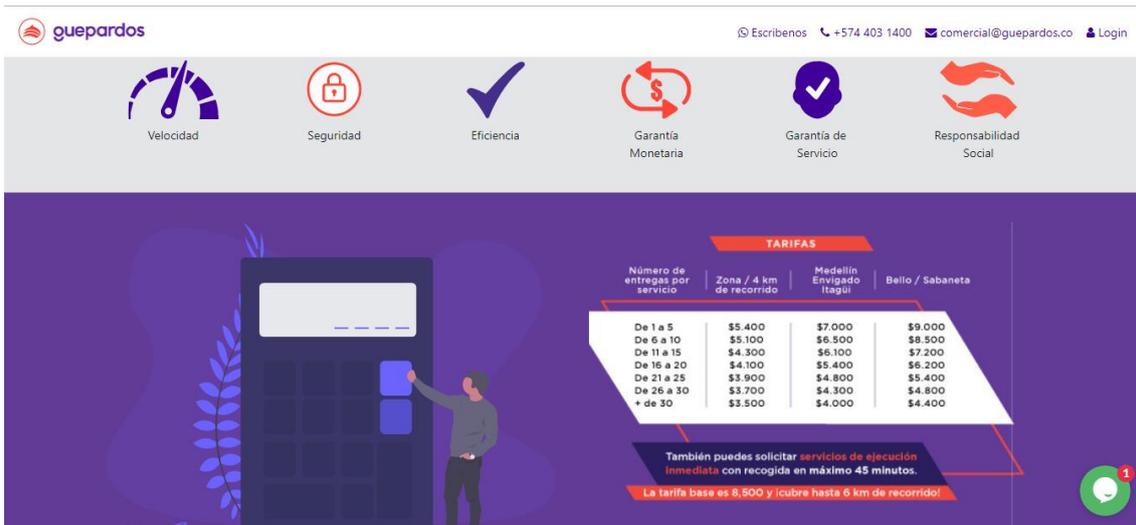


Ilustración 14 Página principal - Parte 2

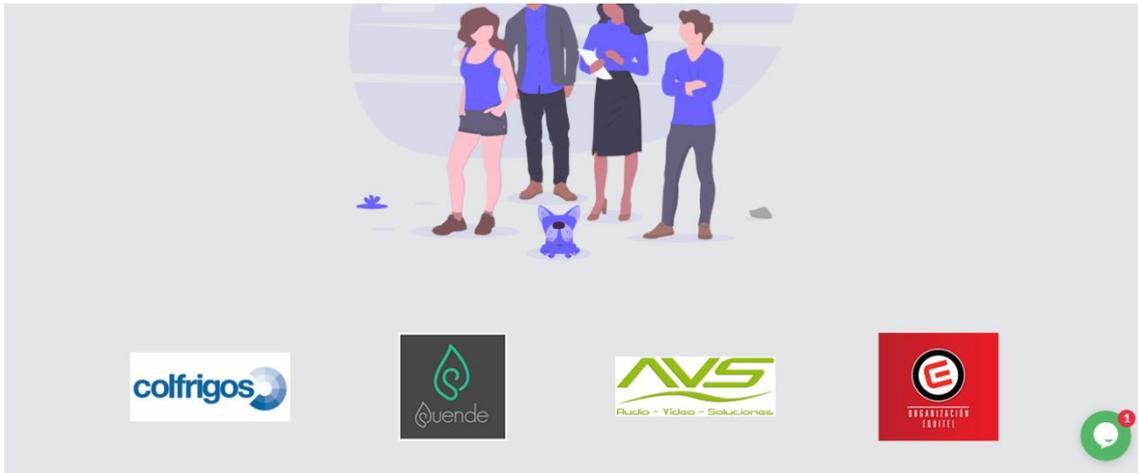


Ilustración 15 Página principal - Parte 3



Comunícate con nosotros

Calle 29 # 41 - 105 Of 501 Edificio Soho Medellín

+57 4 403 14 00
 +57 319 727 5548

comercial@guepardos.co

Lunes-viernes
7am - 8pm
Sábado
8am - 4pm



Ilustración 16 Página principal - Parte 4

Página de Acceso

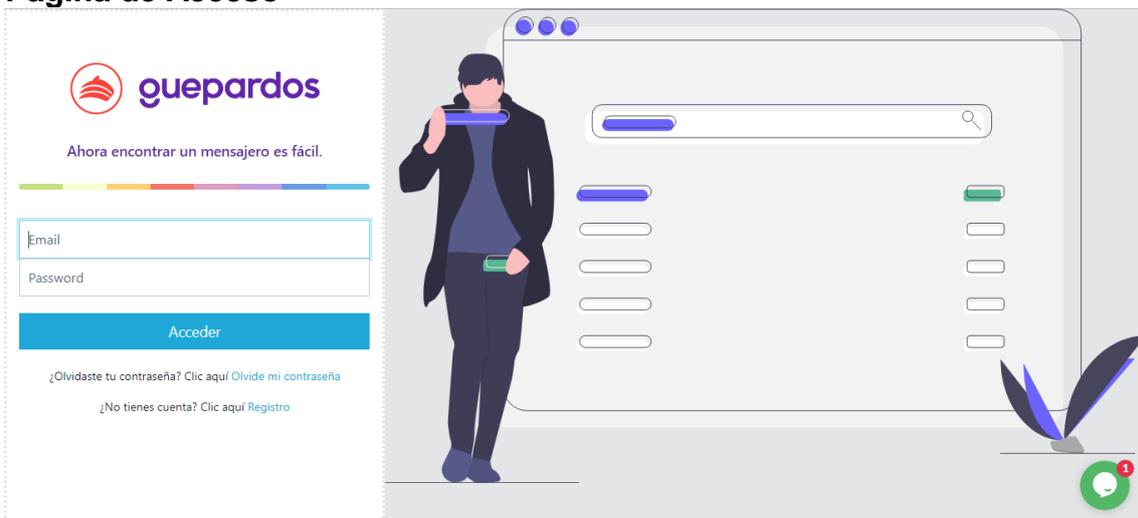


Ilustración 17 Página de acceso

Ventana recuperar contraseña

Recuperar Contraseña



Introduce el email con el que te registraste

example@email.com

Cancelar

Recuperar

Ilustración 18 Ventana recuperar contraseña

Página para registrarse

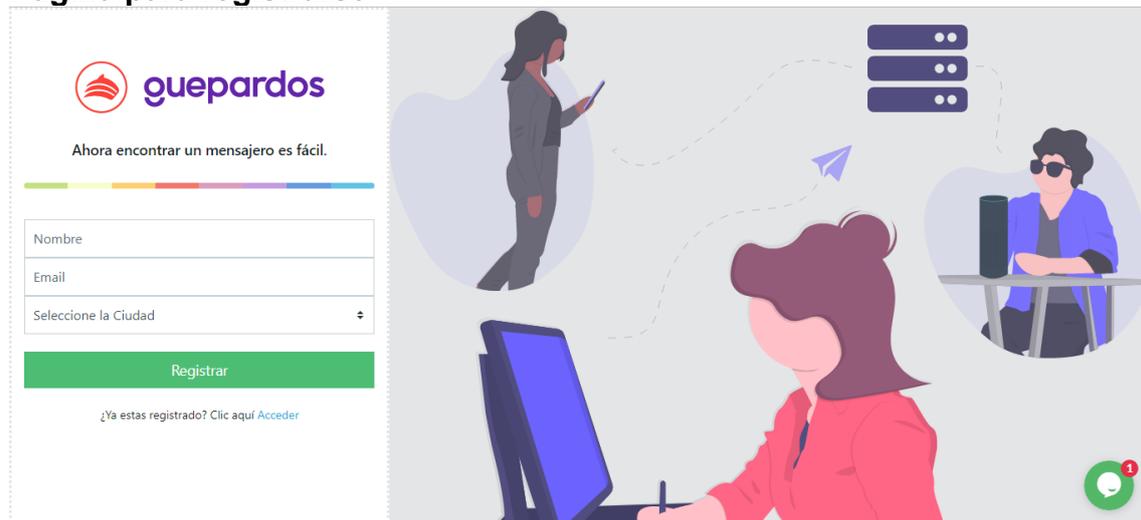
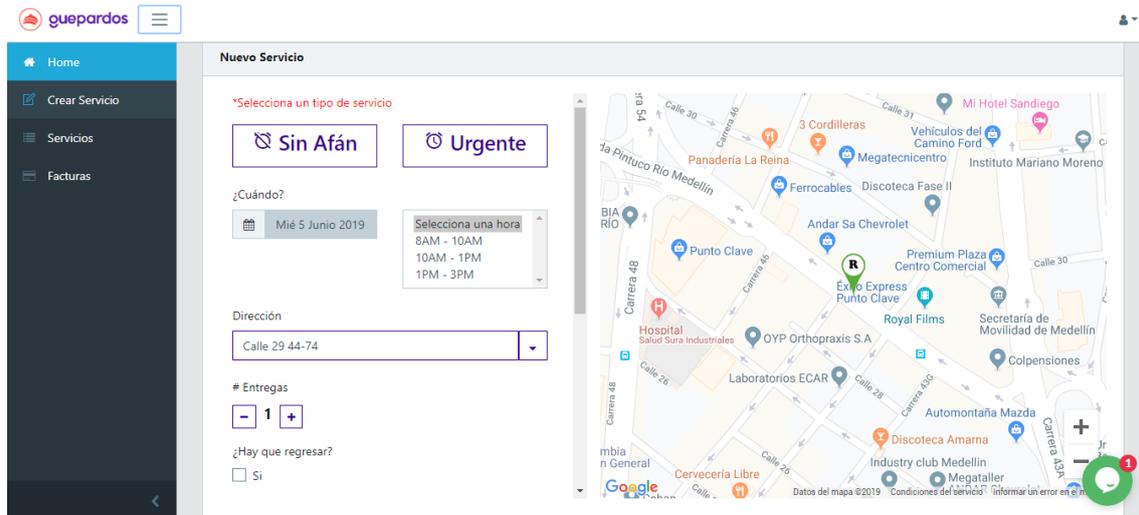


Ilustración 19 Ventana Registrarse

Página de Nuevo Servicio



guepardos

Home

Crear Servicio

Servicios

Facturas

Nuevo Servicio

*Selecciona un tipo de servicio

¿Cuándo?

Mié 5 Junio 2019

Selecciona una hora

8AM - 10AM
10AM - 1PM
1PM - 3PM

Dirección

Calle 29 44-74

Entregas

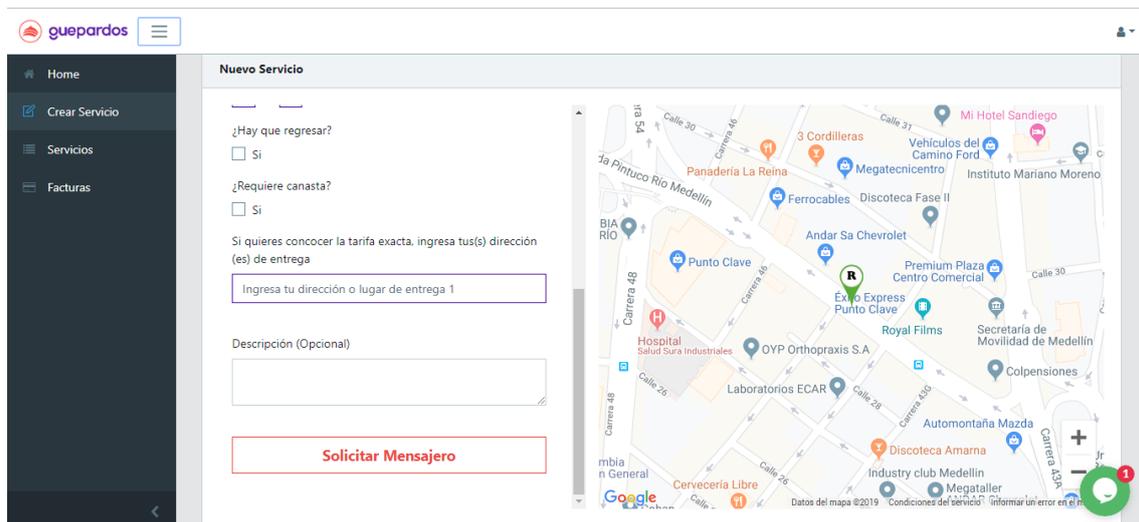
- 1 +

¿Hay que regresar?

Si

Mapa de Google Maps con marcadores de puntos de entrega y recepción.

Ilustración 20 Página Nuevo Servicio - Parte 1



guepardos

Home

Crear Servicio

Servicios

Facturas

Nuevo Servicio

¿Hay que regresar?

Si

¿Requiere canasta?

Si

Si quieres conocer la tarifa exacta, ingresa tus(s) dirección (es) de entrega

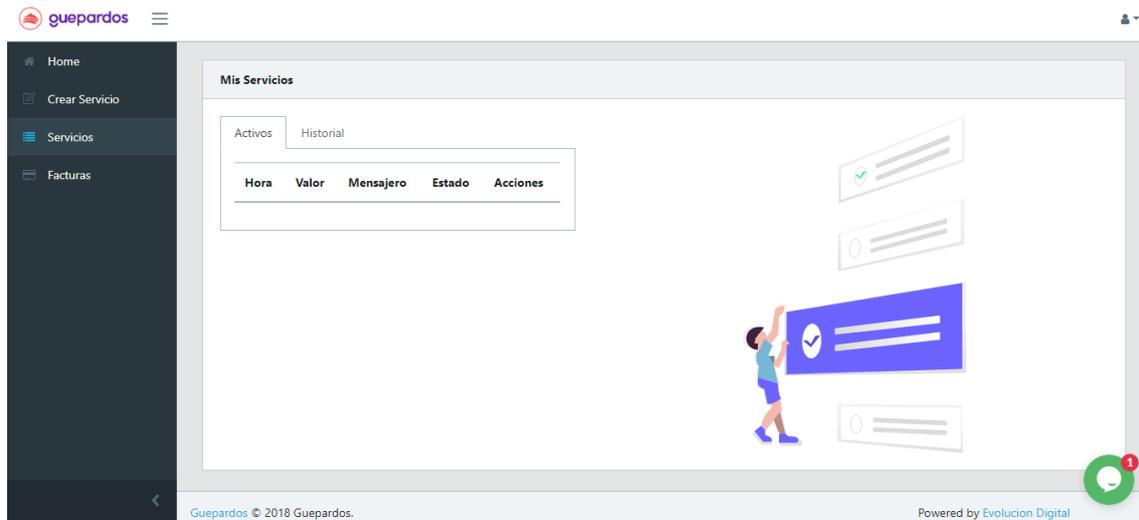
Ingresar tu dirección o lugar de entrega 1

Descripción (Opcional)

Mapa de Google Maps con marcadores de puntos de entrega y recepción.

Ilustración 21 Página Nuevo Servicio - Parte 2

Página de Mis servicios - Activos



Mis Servicios

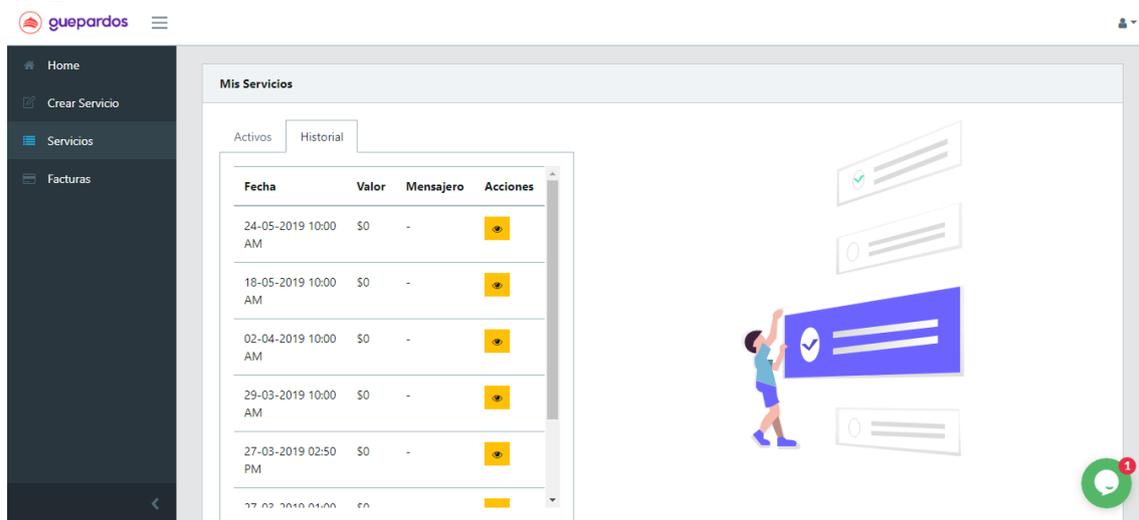
Activos Historial

Hora	Valor	Mensajero	Estado	Acciones
------	-------	-----------	--------	----------

Guepardos © 2018 Guepardos. Powered by Evolucion Digital

Ilustración 22 Página Mis Servicios - Activos

Página de Mis servicios - Historial



Mis Servicios

Activos Historial

Fecha	Valor	Mensajero	Acciones
24-05-2019 10:00 AM	\$0 -	-	[Icono]
18-05-2019 10:00 AM	\$0 -	-	[Icono]
02-04-2019 10:00 AM	\$0 -	-	[Icono]
29-03-2019 10:00 AM	\$0 -	-	[Icono]
27-03-2019 02:50 PM	\$0 -	-	[Icono]
27-03-2019 04:00 PM	\$0 -	-	[Icono]

Guepardos © 2018 Guepardos. Powered by Evolucion Digital

Ilustración 23 Página Mis Servicios – Historial

Página de Mis facturas

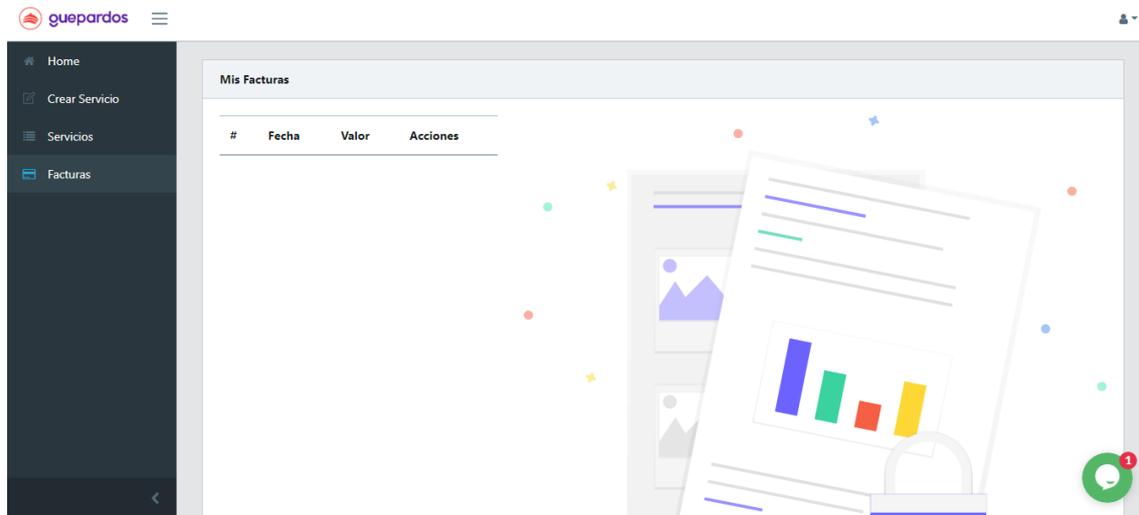


Ilustración 24 Página Mis facturas

Página Mi Perfil

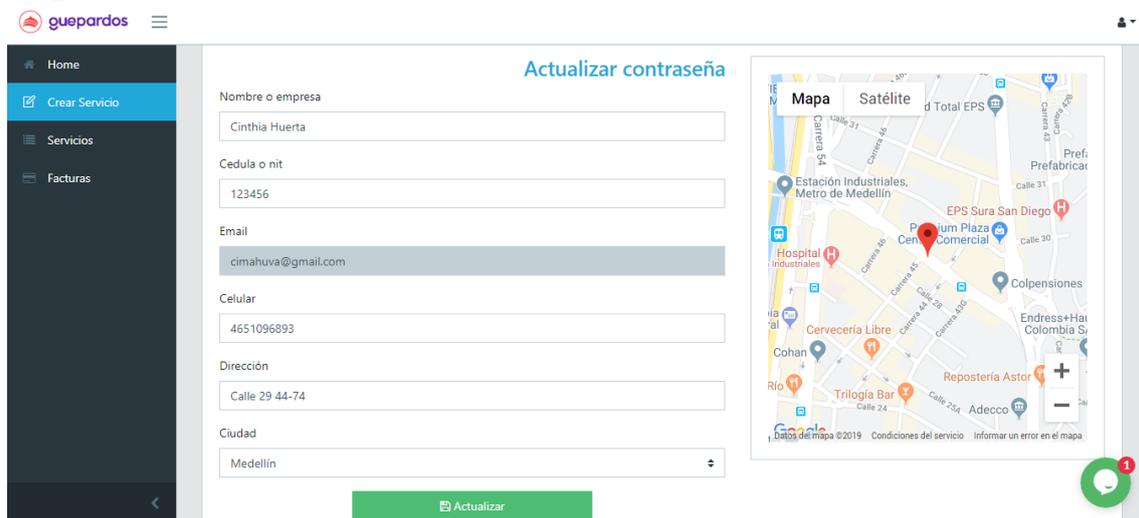
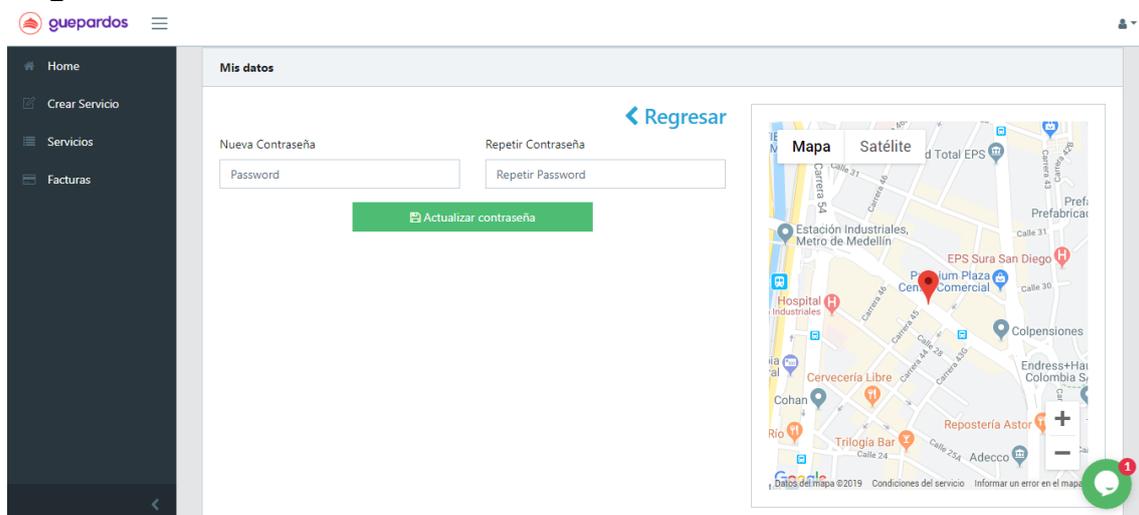


Ilustración 25 Página Mi perfil

Página Actualizar contraseña



Ilustración

26

Página

Actualizar

contraseña

Pruebas de caja negra y pruebas funcionales

Registrarse en la plataforma

El usuario debe ingresar sus datos como Nombre o empresa, email, y seleccionar la ciudad en la que se encuentra.

Registrar datos de usuario → La plataforma envía un correo electrónico al cliente con su contraseña y muestra un aviso de que se ha registrado

Validar formulario de registro

Se tienen que llenar todos los campos, ingresar un correo electrónico válido y seleccionar una ciudad

Ingresar el nombre → El sistema permite el registro del usuario

No ingresar nombre → El Sistema muestra error “El nombre es obligatorio.”

El nombre es obligatorio.

Ingresar el email válido → El sistema permite el registro del usuario

Ingresar un correo no válido o ya registrado---> El sistema muestra el error “El correo ya se ha registrado”

No ingresar el correo → El sistema muestra el mensaje de error “El email es obligatorio y debe ser válido.”

El email es obligatorio y debe ser válido.

Seleccionar una ciudad → El sistema permite registrar el usuario

No seleccionar una ciudad → EL sistema muestra el mensaje de error “Selecione una ciudad”

Seleccione una ciudad.

Recuperar contraseña

Si el usuario olvida su contraseña debe recuperarla en la ventana de Recuperar contraseña ingresando su correo electrónico

Ingresar un correo válido y ya registrado → El sistema envía un correo al cliente para recuperar la contraseña

No ingresar un correo válido o existente → El sistema envía mensaje de error “El email no está registrado en la plataforma”



Email no registrado en nuestra
plataforma.

OK

No ingresar ningún correo y dar clic en recuperar → El sistema manda el mensaje de error “Por favor introduce un email válido”



Por favor introduce un email válido.

OK

Ingresar a la plataforma

El usuario tiene que ingresar sus datos como el correo electrónico y su contraseña tal y como se registró e ingresando la contraseña que se le envió al correo o la actual si ya se cambió anteriormente.

Ingresar el email registrado → El sistema permite ingresar a la plataforma

No ingresar un email registrado → El sistema muestra el mensaje “Error de autenticación” y no permite ingresar



Error de autenticación

Usuario y/o contraseña incorrecta.

OK

No ingresar ningún correo → el campo se pone en color rojo y no permite ingresar

Ingresar la contraseña correcta → La plataforma permite el ingreso al usuario
Ingresar la contraseña incorrecta → El sistema muestra el mensaje de error “Error de autenticación” y no permite ingresar



Error de autenticación

Usuario y/o contraseña incorrecta.

OK

No ingresar ningún correo → El sistema muestra el campo en color rojo y no permite el ingreso a la plataforma

Página de Mi perfil

El usuario debe tener sus datos actualizados y completos en este formulario de manera correcta

Eliminar el nombre o empresa → El campo se pone en rojo y aparece el mensaje “El nombre es obligatorio” y no permite actualizar

Nombre



El nombre es obligatorio.

Ingresar la cédula o NIT → El sistema permite actualizar los datos

Dejar en blanco la cédula o NIT → El campo se pone en rojo y muestra el mensaje de error “La cédula es obligatoria” y no permite actualizar

NIT



La cedula es obligatoria.

El email no se debe cambiar ni actualizar → El sistema no permite que se edite el campo de email

Email

cimahuva@gmail.com

Ingresar el número celular → EL sistema permite actualizar los datos del usuario

Ingresar un número de celular inválido → La plataforma muestra el mensaje de error “El número de celular no es válido.” y no permite actualizar

46510968



El numero de celular no es valido.

No ingresar un número de celular → El sistema muestra el campo del celular en rojo y no permite actualizar los datos

Ingresar la dirección del usuario → El sistema muestra la dirección permite actualizar los datos

Dirección

Calle 29 44 74, Colombia, El Poblado, Medellín, Antioquia

Medellín

Calle 29 44 74, Perpetuo Socorro, La Candelaria, Medellín, Antioquia

No ingresar una dirección → El sistema muestra el campo en color rojo y no permite actualizar

La ciudad no se puede dejar en blanco → El sistema no permite que se elimine la ciudad

Ingresar todos los datos correctamente y completos → La plataforma muestra el mensaje “Datos actualizados”



Datos Actualizados

Datos de usuario actualizados correctamente :)



Pedir un servicio

El usuario puede pedir un servicio en la página de pedidos teniendo los datos básicos completos y de manera correcta

Se debe recordar seleccionar un tipo de servicio → El sistema muestra en color rojo “Selecciona un tipo de servicio”

*Selecciona un tipo de servicio



Seleccionar un tipo de servicio → El sistema procesa el servicio correctamente

No se selecciona un tipo de servicio, el sistema muestra el mensaje de error “Tipo de servicio obligatorio” y no permite realizar el pedido



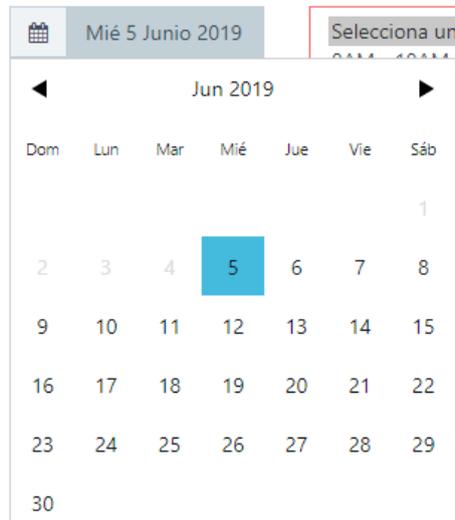
Tipo de servicio obligatorio

Por favor seleccione un tipo de servicio.

OK

La fecha no debe quedar vacía → El sistema tiene la fecha actual seleccionada
Se trata de seleccionar una fecha anterior → Se validó que solo se muestre la fecha actual hasta la de ahora

¿Cuándo?



Dom	Lun	Mar	Mié	Jue	Vie	Sáb
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Seleccionar “Sin afán” y seleccionar un rango de hora → El pedido se envía correctamente

Seleccionar “Sin afán” y no seleccionar un rango de hora → El campo queda en color rojo, aparece el mensaje “Fecha u hora no válidas” y no permite procesar el pedido



Fecha u hora no válidas

Selecciona una fecha y rango de horas válido, para tu recogida.

OK

Seleccionar “Urgente” y debe aparecer la fecha y hora actual → El sistema muestra la hora actual

Tratar de dejar el campo de hora vacío → El sistema no permite eliminar esos datos

Seleccionar una hora anterior a la actual → La plataforma muestra el mensaje de error “Error de hora” y no permite procesar el pedido



Error de hora

Rango horario no disponible.



El sistema debe tomar la dirección registrada actual → El sistema si muestra la dirección actual

Se deja el campo de dirección vacío → El sistema muestra el mensaje “dirección de recogida obligatoria” y no permite realizar el pedido



Dirección de recogida obligatoria

La dirección de recogida es obligatoria, no puede ir vacía o no válida.



Se ingresa una dirección no válida en Colombia → El sistema no lo toma como correcto y aparece el campo en rojo



Input field: calle 54 | [X] [v]

Search results:

- CI. 54, Medellín, Antioquia, Colombia
- CI. 54, Medellín, Antioquia, Colombia

El número de entregas mínimo debe ser 1 → El sistema no permite bajar de ese número

Se agregan entregas y aparecen más campos de direcciones de entrega → La plataforma muestra correctamente un campo de dirección por cada entrega

Entregas

3

¿Hay que regresar?

Si

¿Requiere canasta?

Si

Si quieres conocer la tarifa exacta, ingresa tus(s) dirección(es) de entrega

Ingresa tu dirección o lugar de entrega 1

Ingresa tu dirección o lugar de entrega 2

Ingresa tu dirección o lugar de entrega 3

Se ingresa una dirección válida en Colombia → El sistema no lo toma como correcta y aparece el campo en rojo

Sebastián Baño 

Las direcciones de entrega son opcionales → Si se ingresan o no direcciones válidas el servicio se procesa

Es opcional seleccionar “¿Hay que regresar?” → El sistema toma correctamente el valor en el pedido

¿Hay que regresar?

Si

Es opcional seleccionar “¿Requiere canasta?” → El sistema toma correctamente el valor

¿Requiere canasta?

Si

La descripción es opcional → El pedido se envía si la descripción está o no llena
Se llenan los campos correctamente → La plataforma muestra el mensaje “Tu servicio ha sido creado” redirecciona a la página de Servicios y se envían los datos correctamente

Tu servicio ha sido creado   

El servicio recién creado debe aparecer en la pestaña Activos → El sistema muestra el pedido correctamente

Activos		Historial		
Hora	Valor	Mensajero	Estado	Acciones
08:00 AM	\$0	-	Agendado	 

Dar clic en el botón amarillo para ver los datos del pedido → La plataforma muestra correctamente los datos

Activos		Historial		
Hora	Valor	Mensajero	Estado	Acciones
08:00 AM	\$0	-	Agendado	 

Estado: Agendado

Fecha: 06-06-2019 08:00 AM

Nombre: Cinthia Huerta

Recogida: Calle 29 44-74

Entrega: Multientrega

Ida y Regreso: Si

Entregas: 2

Descripción: NO TOMAR ES UNA PRUEBA! ;)

Valor: \$0

Mensajero: -

[Ver Mapa](#)

Dar clic en el botón rojo para cancelar servicio → La plataforma muestra el mensaje “Cancelar Servicio” y la fecha del servicio a cancelar

Cancelar Servicio ×

* Cancelar servicio programado para : **06-06-2019 08:00 AM**

Salir
Cancelar

Dar clic en el botón salir → El sistema regresa a los Servicios Activos

Dar clic en el botón Cancelar → El sistema cancela el servicio, muestra el mensaje “Servicio cancelado” y ya no aparece el botón de cancelar



Servicio Cancelado

El Servicio ha sido cancelado.

OK

Ver el servicio cancelado en la pestaña Historial → El servicio aparece correctamente

Activos **Historial**

Fecha	Valor	Mensajero	Acciones
06-06-2019 08:00 AM	\$0	-	
24-05-2019 10:00 AM	\$0	-	
18-05-2019 10:00 AM	\$0	-	
02-04-2019 10:00 AM	\$0	-	
29-03-2019 10:00 AM	\$0	-	
27-03-2019 07:50 AM	\$0	-	

Estado: Agendado

Fecha: 06-06-2019 08:00 AM

Nombre: Cinthia Huerta

Recogida: Calle 29 44-74

Entrega: Multientrega

Ida y Regreso: Si

Entregas: 2

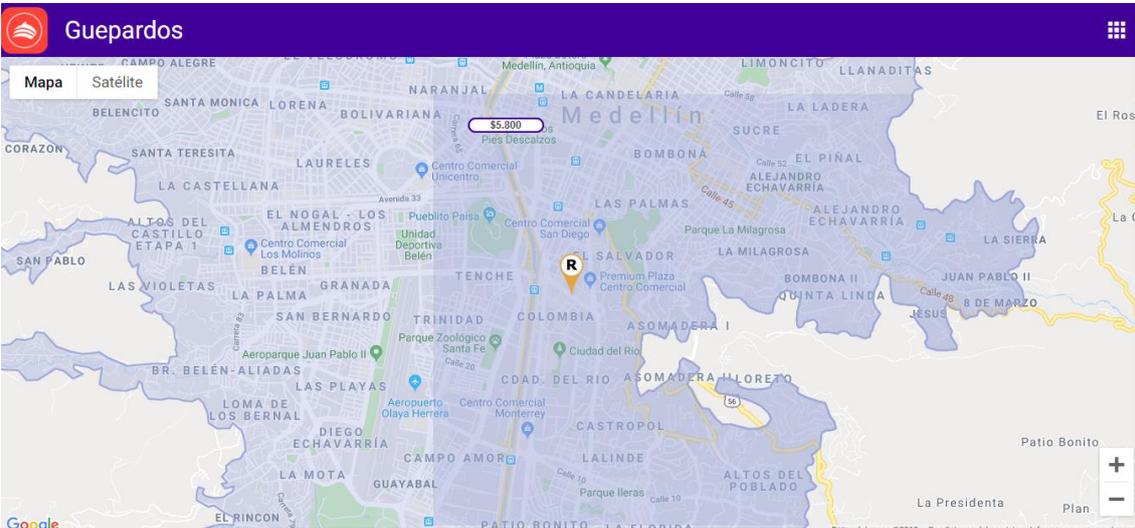
Descripción: NO TOMAR ES UNA PRUEBA! :)

Valor: \$0

Mensajero: -

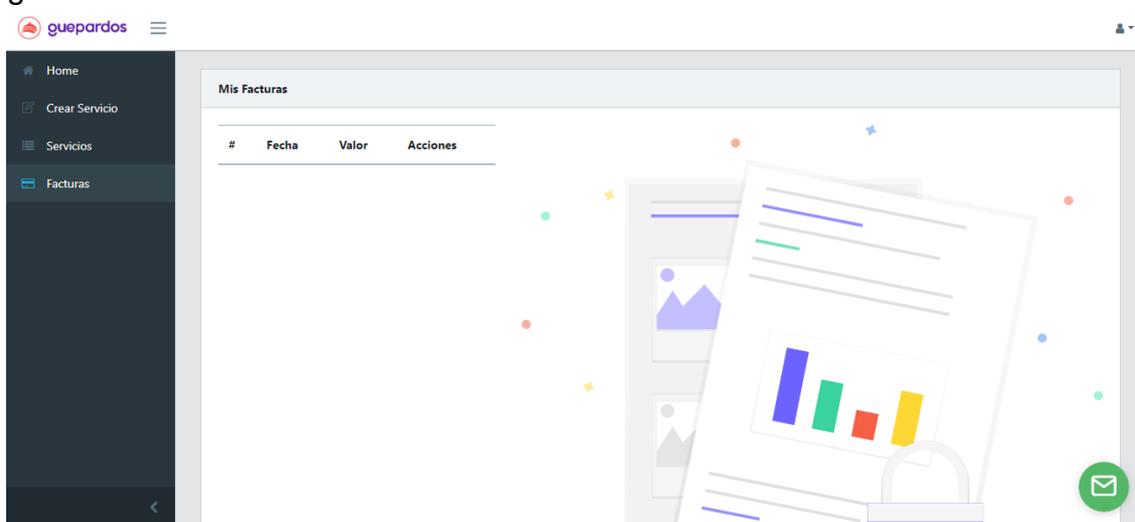
[Ver Mapa](#)

Dar click en ver mapa → El sistema muestra el recorrido del servicio correctamente

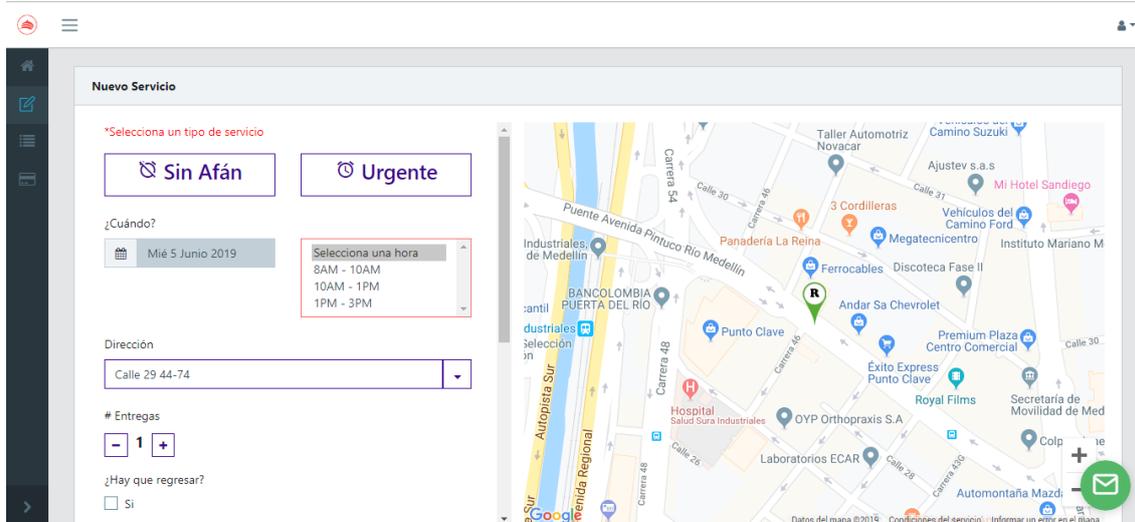


The screenshot shows a Google Maps interface with a purple header for 'Guepardos'. The map displays a yellow route through Medellín, Colombia, with various neighborhoods labeled. A red location pin is placed on the route. The map includes standard Google Maps controls like zoom in (+) and zoom out (-) buttons.

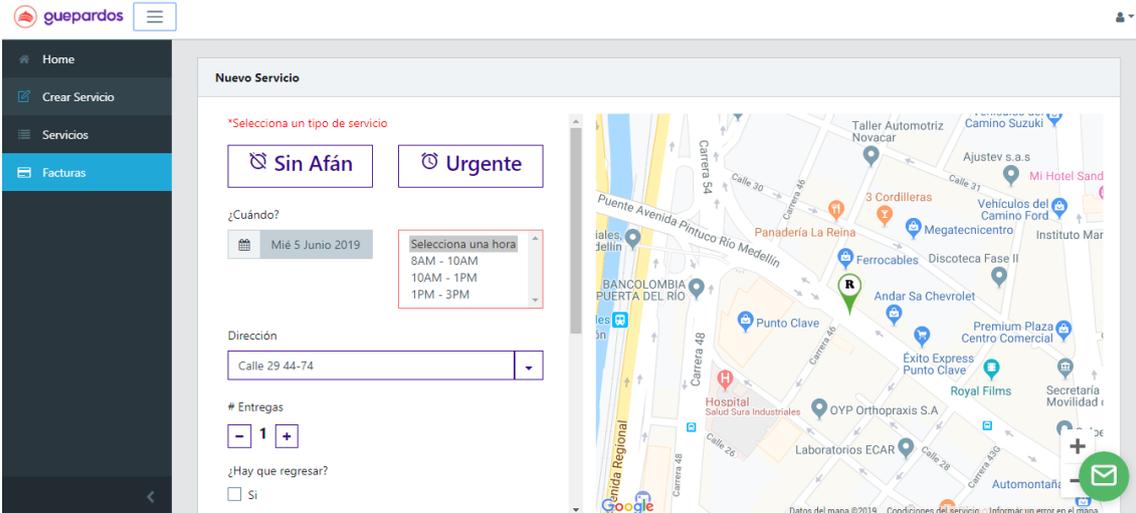
Entrar al apartado Facturas → La plataforma muestra las facturas que se han generado



Encoger o expandir el menú con la flecha de abajo → El sistema muestra correctamente el menú



Encoger o expandir el menú con el ícono de arriba → El sistema muestra correctamente el menú



guepardos

Home
Crear Servicio
Servicios
Facturas

Nuevo Servicio

*Selecciona un tipo de servicio

Sin Afán Urgente

¿Cuándo?

Mié 5 Junio 2019

Selecciona una hora
8AM - 10AM
10AM - 1PM
1PM - 3PM

Dirección
Calle 29 44-74

Entregas
- 1 +

¿Hay que regresar?
 Si

Mapa de Google Maps con marcadores de puntos de entrega y recepción.

Entrar a la página principal en home y seguir autenticado → El sistema muestra en la parte superior derecha la opción App lo cual muestra seguir adentro



Escribenos +574 403 1400 comercial@guepardos.co App

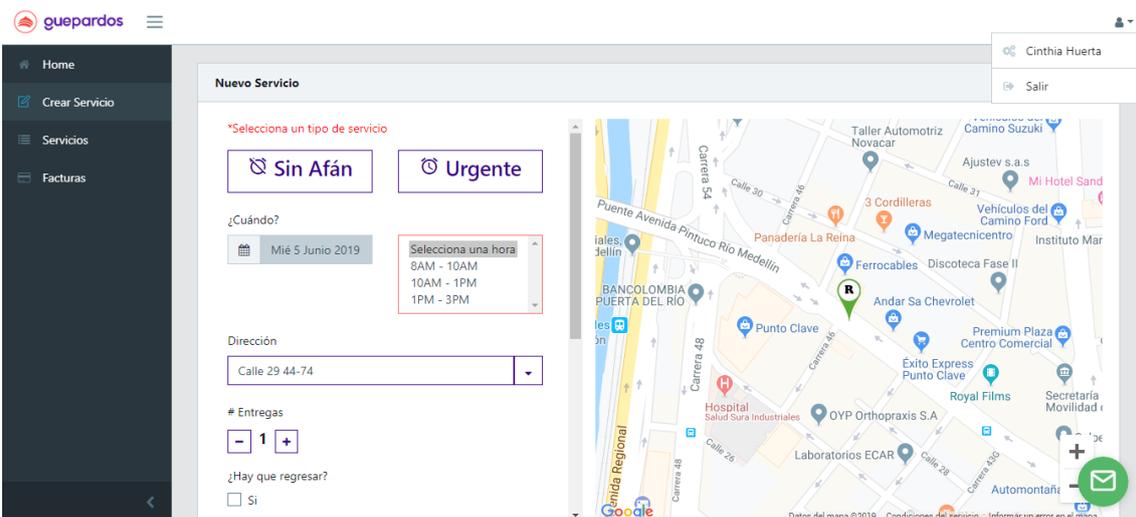


guepardos

¡LA MEJOR OPCIÓN DE MENSAJERÍA PARA TU EMPRESA!

En Guepardos ajustamos nuestros servicios a tus necesidades
¡Somos mensajería de calidad, con una oferta amplia y un gran número de mensajeros listos para atenderte!

Cerrar sesión al lado superior derecho → El sistema cierra sesión correctamente y redirecciona a la pantalla principal



guepardos

Home
Crear Servicio
Servicios
Facturas

Nuevo Servicio

*Selecciona un tipo de servicio

Sin Afán Urgente

¿Cuándo?

Mié 5 Junio 2019

Selecciona una hora
8AM - 10AM
10AM - 1PM
1PM - 3PM

Dirección
Calle 29 44-74

Entregas
- 1 +

¿Hay que regresar?
 Si

Mapa de Google Maps con marcadores de puntos de entrega y recepción.

Cinthia Huerta
Salir

Verificar que ya estamos fuera de la aplicación → El sistema muestra en la parte superior derecha la opción Login, lo cual muestra que ya estamos afuera



The screenshot shows the top section of the Guepardos website. At the top left is the Guepardos logo, which consists of a white circular icon containing a stylized jaguar head profile, followed by the word "guepardos" in a lowercase, white, sans-serif font. To the right of the logo, there is a navigation bar with the text "Escribenos" (with a speech bubble icon), "+574 403 1400" (with a phone icon), "comercial@guepardos.co" (with an email icon), and "Login" (with a person icon). Below the logo and navigation bar is a dark banner with a background of colorful light streaks. The banner contains the text "¡LA MEJOR OPCIÓN DE MENSAJERÍA PARA TU EMPRESA!" in white, uppercase letters. At the bottom of the banner, there is a light gray box with the text "En Guepardos ajustamos nuestros servicios a tus necesidades" and "¡Somos mensajería de calidad, con una oferta amplia y un gran número de mensajeros listos para atenderte!" in blue and black. A green circular icon with a white envelope symbol is located in the bottom right corner of the light gray box.

Prototipo de aplicación móvil de la plataforma para usuarios

Se necesita de crear un prototipo de aplicación móvil con los mismos requerimientos que la aplicación web antes vista. Se necesita que la aplicación sea de manera nativa para iOS y Android, por lo cual, para no hacer trabajo doble se usará Flutter y Dart. De esta manera sólo se realizará una misma aplicación que se ejecute en las dos plataformas móviles, esto se logra mediante widgets creados por Flutter que funcionan para las dos plataformas. Se usará la misma base de datos que usa la plataforma web para mantener en el mismo lugar todos los registros de usuarios y pedidos.

Plugins que se necesitaron

```
cupertino_icons: ^0.1.2
http: ^0.11.3+16
shared_preferences: ^0.4.2
uuid: 2.0.0
datetime_picker_formfield: ^0.1.8
```

Ilustración 27 Plugins requeridos en la app

- **Cupertino Icons:** Este plugin viene predeterminado para el uso de íconos en el diseño de la aplicación
- **HTTP:** Se necesitó para hacer las peticiones al servidor de autenticación y enviar los datos a la base de datos
- **Shared Preferences:** Se necesitó de este plugin para guardar datos internamente y de forma segura en la aplicación como el token y los datos de usuario al ingresar en la app
- **UUID:** Se requirió de este plugin sólo para generar ID únicos de manera aleatoria en un formato
- **Date Time Picker Form Field:** Se necesitó de este plugin para elegir la hora y fecha del pedido y mostrándose en el formato correcto

Paquetes importados para el uso de plugins

```
import 'package:flutter/material.dart';  
import 'package:guepardos/screens/menu.dart';  
import 'package:guepardos/apifunctions/saveCurrentLogin.dart';  
import 'package:datetime_picker_formfield/datetime_picker_formfield.dart';  
import 'package:intl/intl.dart';  
import 'package:uuid/uuid.dart';  
import 'package:http/http.dart' as http;  
import 'dart:async';  
import 'dart:convert';
```

Ilustración 28 Paquetes importados a los archivos Dart

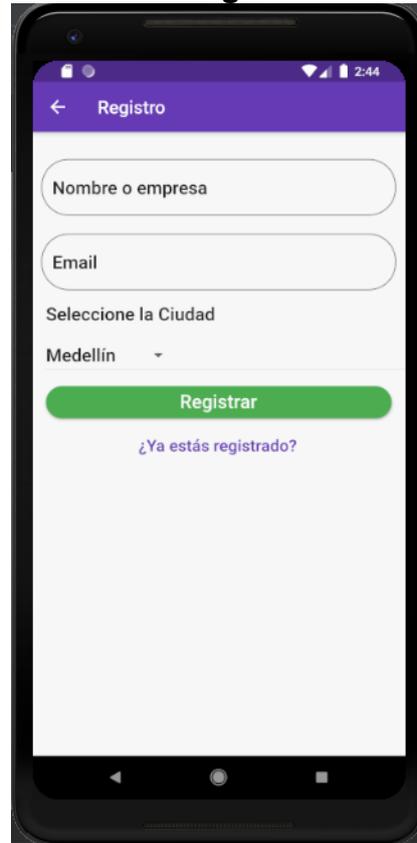
- **Material:** Se necesita de importar este paquete en todas las vistas para usar los widgets de Flutter
- **Screens/menu:** Es el archivo menú que se importó en todas las pantallas
- **ApiFunctions/SaveCurrentLogin:** Se importó este archivo Dart para obtener de ahí los datos guardados del usuario
- **DateTime_Picker_FormField:** Se importó este paquete para el uso del plugin de seleccionar fecha y hora
- **Intl:** Se necesitó de este plugin que viene predeterminado en Dart para el formato y análisis de la fecha y hora
- **UUID:** Este paquete se requiere para usar el plugin de generar ID únicos
- **HTTP:** Se necesita de este plugin para usarlo en peticiones http, autenticación, actualización y mandar y recibir datos del servidor
- **Async:** Es necesario este paquete Dart para generar funciones asíncronas, es decir, que no se termine el programa hasta que la función se complete
- **Convert:** Se necesitó de importar este plugin para la conversión de los datos JSON (codificar y decodificar) y poder usar los datos en nuestra aplicación

Capturas de pantalla en Android (Pixel 2 XL)

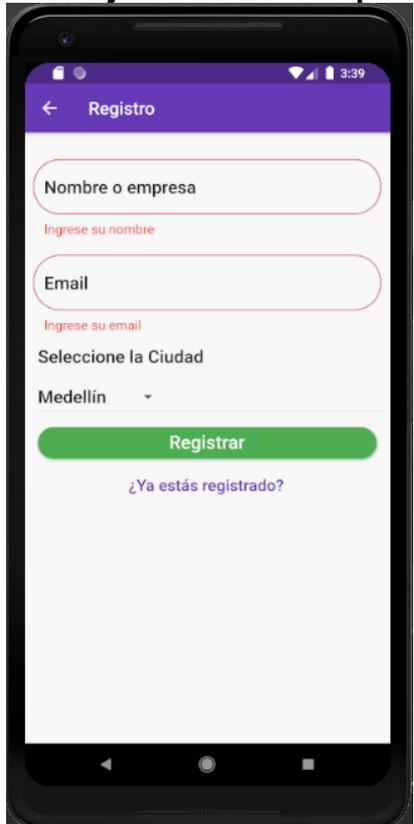
Pantalla de acceso



Pantalla de registro



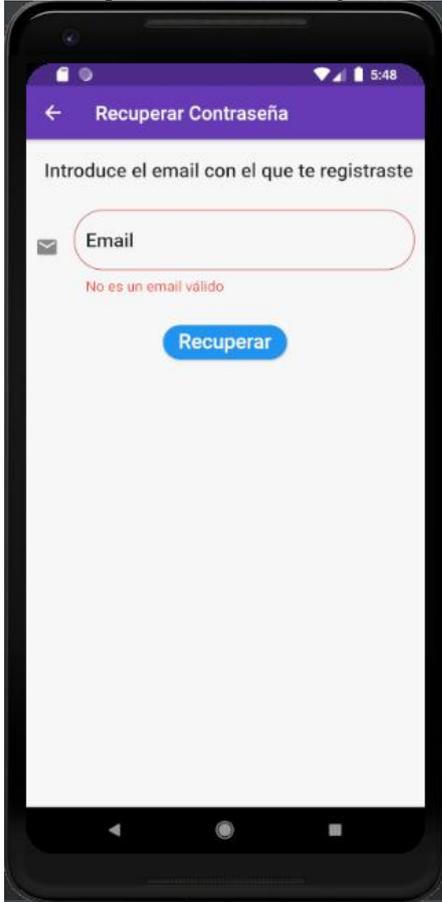
Mensajes de error campos vacíos



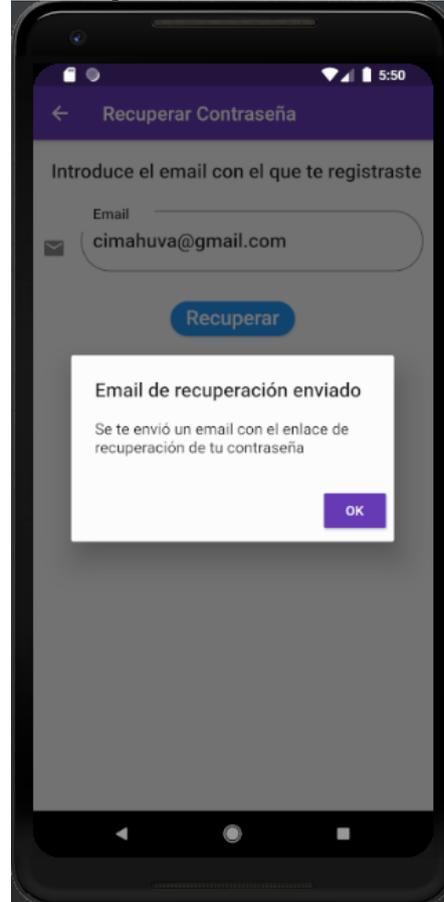
Pantalla de recuperar contraseña



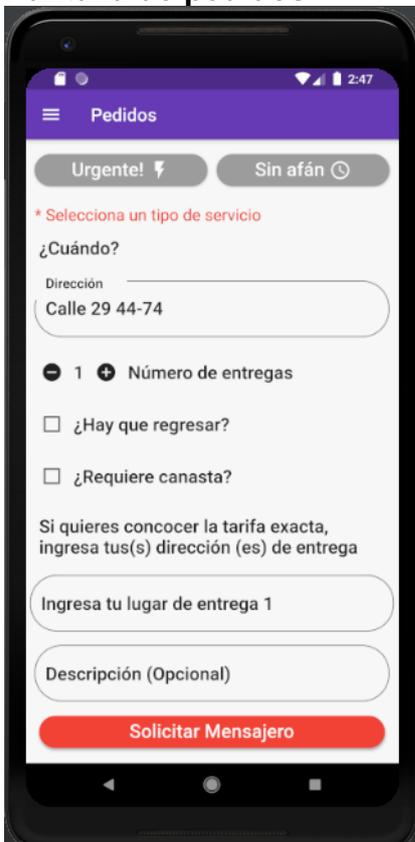
Mensaje de error campo vacío



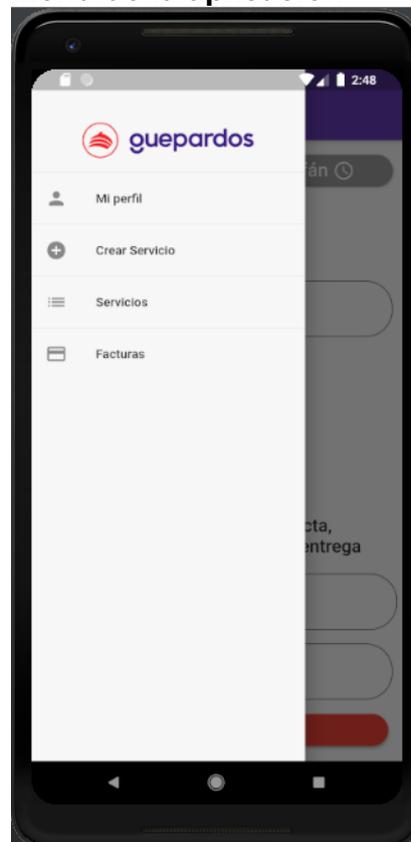
Mensaje de email enviado



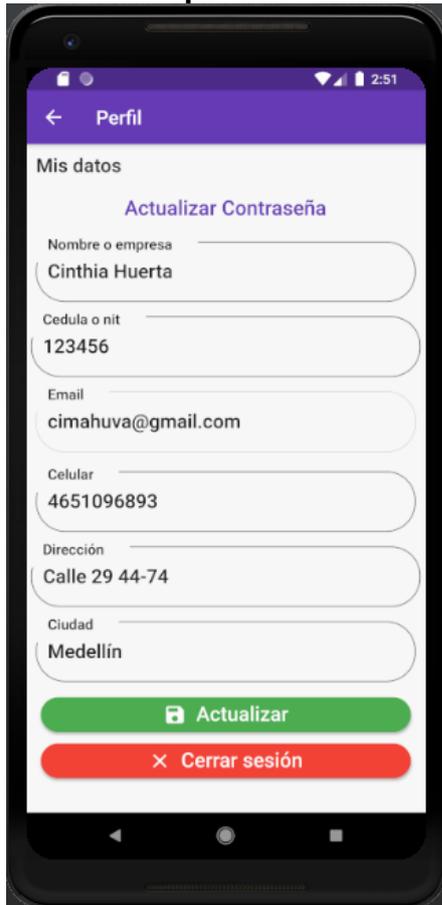
Pantalla de pedidos



Menú de la aplicación



Pantalla Mi perfil



Perfil

Mis datos

Actualizar Contraseña

Nombre o empresa
Cinthia Huerta

Cedula o nit
123456

Email
cimahuva@gmail.com

Celular
4651096893

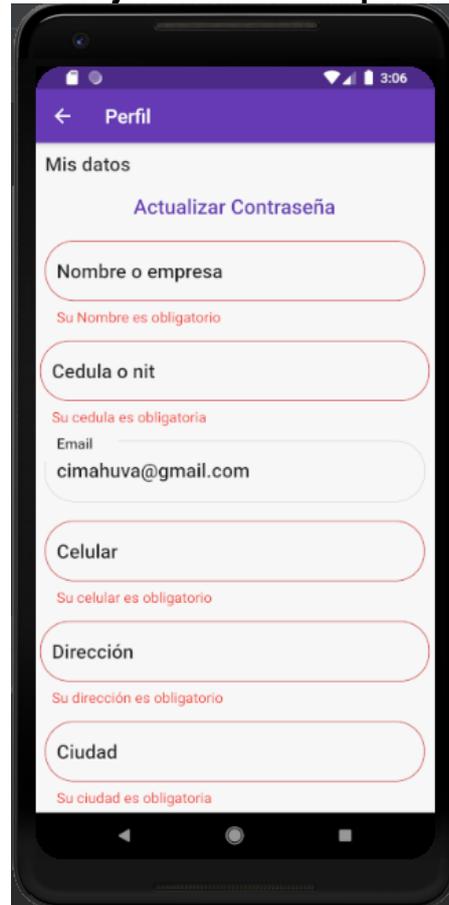
Dirección
Calle 29 44-74

Ciudad
Medellín

Actualizar

Cerrar sesión

Mensajes de error campos vacíos



Perfil

Mis datos

Actualizar Contraseña

Nombre o empresa
Su Nombre es obligatorio

Cedula o nit
Su cedula es obligatoria

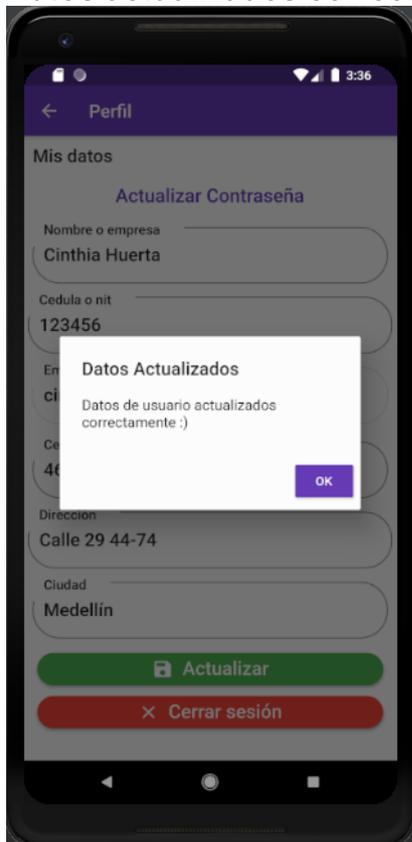
Email
cimahuva@gmail.com

Celular
Su celular es obligatorio

Dirección
Su dirección es obligatorio

Ciudad
Su ciudad es obligatoria

Datos actualizados correctamente



Perfil

Mis datos

Actualizar Contraseña

Nombre o empresa
Cinthia Huerta

Cedula o nit
123456

Datos Actualizados
Datos de usuario actualizados correctamente :)

OK

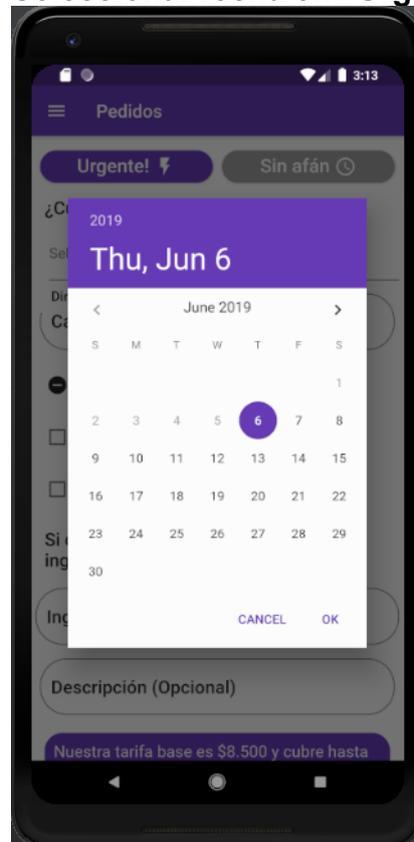
Calle 29 44-74

Ciudad
Medellín

Actualizar

Cerrar sesión

Seleccionar fecha en "Urgente"



Pedidos

Urgente! ⚡ Sin afán ⌚

2019
Thu, Jun 6

June 2019

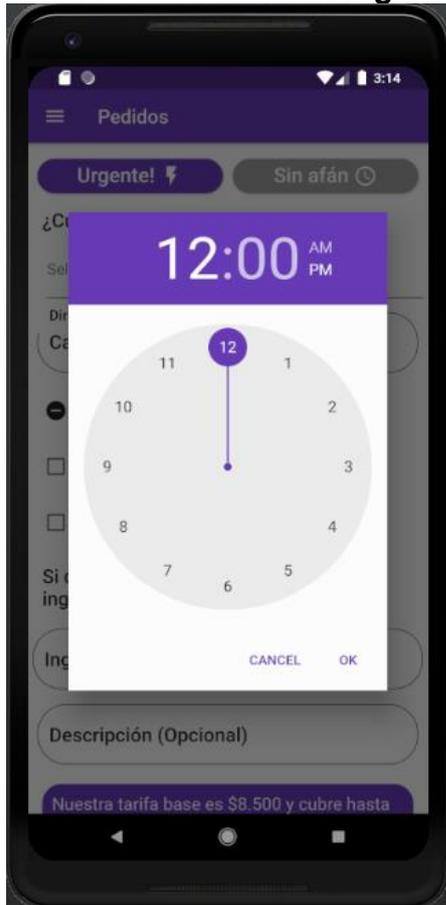
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

CANCEL OK

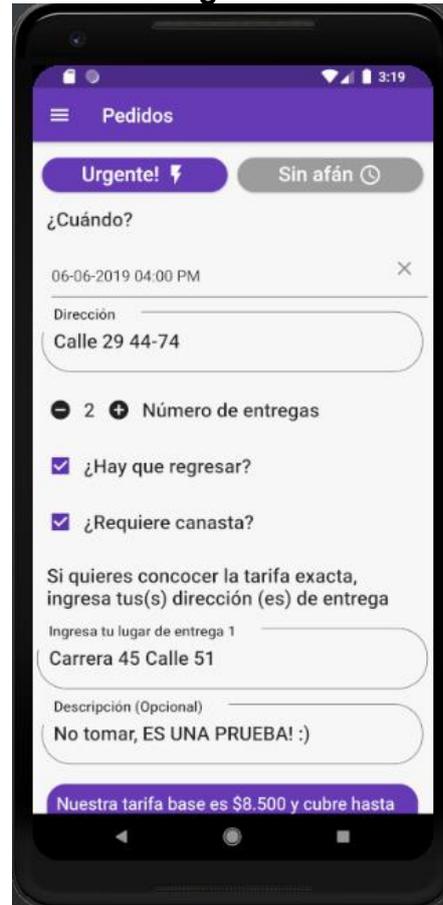
Descripción (Opcional)

Nuestra tarifa base es \$8,500 y cubre hasta

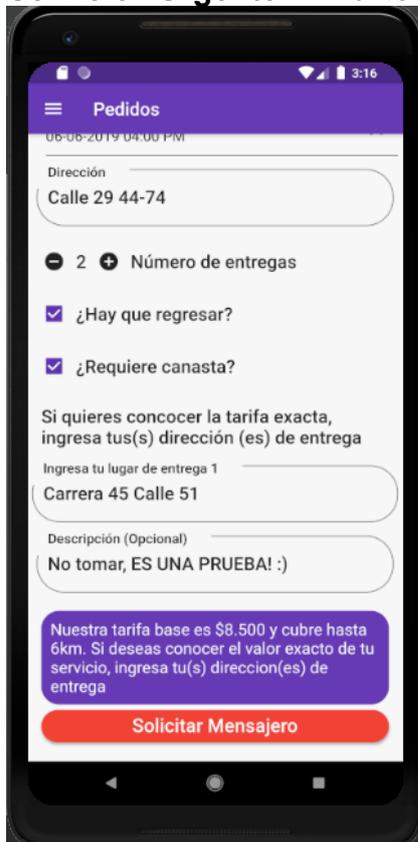
Seleccionar hora en “Urgente”



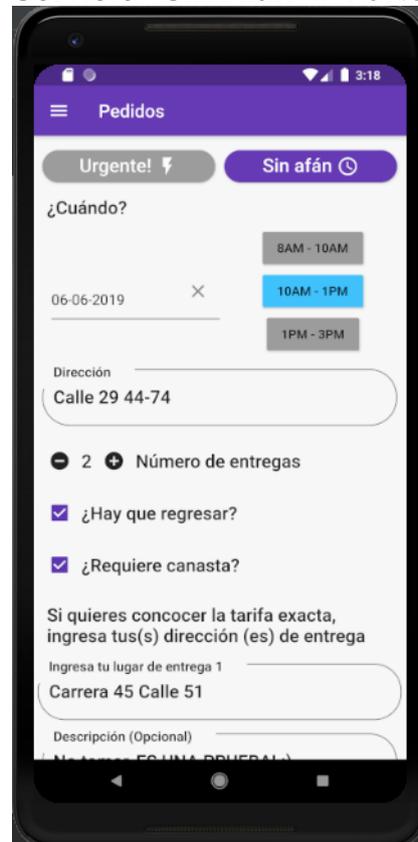
Servicio “Urgente” – Parte 1



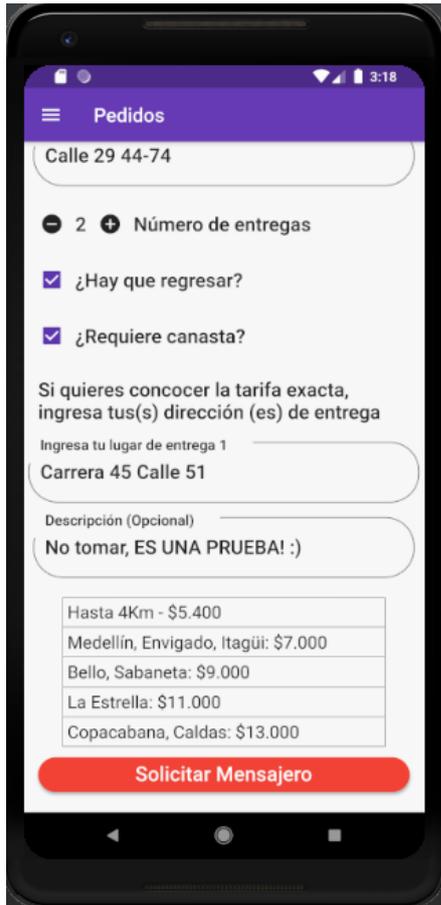
Servicio “Urgente” – Parte 2



Servicio “Sin Afán” – Parte 1



Servicio “Sin Afán” – Parte 2



Pedidos

Calle 29 44-74

2 Número de entregas

¿Hay que regresar?

¿Requiere canasta?

Si quieres conocer la tarifa exacta, ingresa tus(s) dirección (es) de entrega

Ingresar tu lugar de entrega 1

Carrera 45 Calle 51

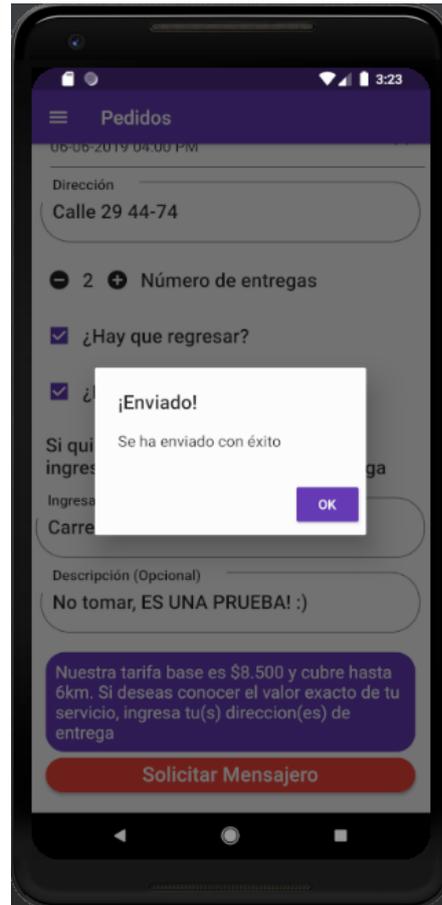
Descripción (Opcional)

No tomar, ES UNA PRUEBA! :)

Hasta 4Km - \$5.400
Medellín, Envigado, Itagüí: \$7.000
Bello, Sabaneta: \$9.000
La Estrella: \$11.000
Copacabana, Caldas: \$13.000

Solicitar Mensajero

Pedido enviado exitosamente



Pedidos

06-06-2019 04:00 PM

Dirección

Calle 29 44-74

2 Número de entregas

¿Hay que regresar?

¿Requiere canasta?

¡Enviado!

Se ha enviado con éxito

OK

Si quieres conocer la tarifa exacta, ingresa tus(s) dirección (es) de entrega

Ingresar tu lugar de entrega 1

Carrera 45 Calle 51

Descripción (Opcional)

No tomar, ES UNA PRUEBA! :)

Nuestra tarifa base es \$8.500 y cubre hasta 6km. Si deseas conocer el valor exacto de tu servicio, ingresa tu(s) dirección(es) de entrega

Solicitar Mensajero

Pantalla Actualizar contraseña



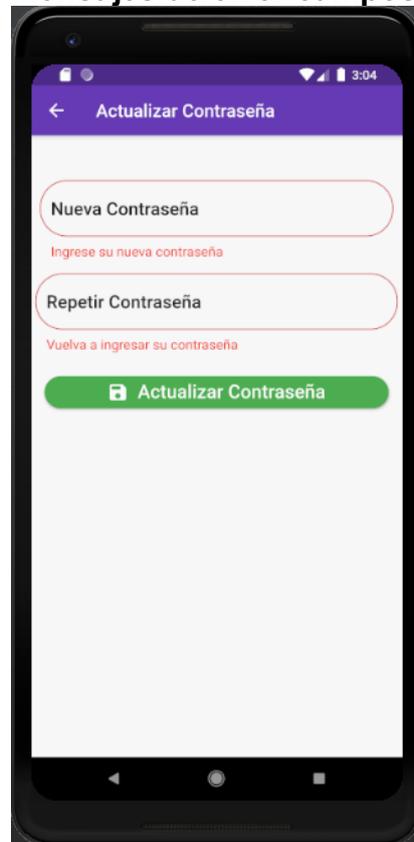
Actualizar Contraseña

Nueva Contraseña

Repetir Contraseña

Actualizar Contraseña

Mensajes de error campos vacíos



Actualizar Contraseña

Nueva Contraseña

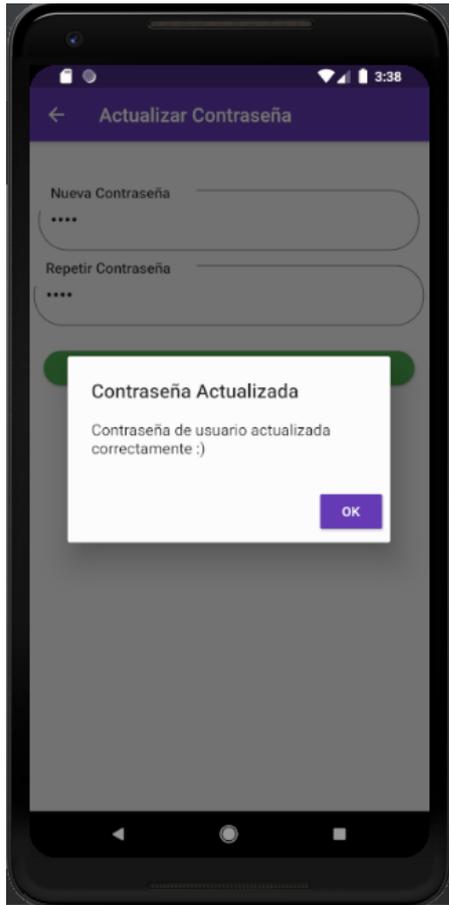
Ingrese su nueva contraseña

Repetir Contraseña

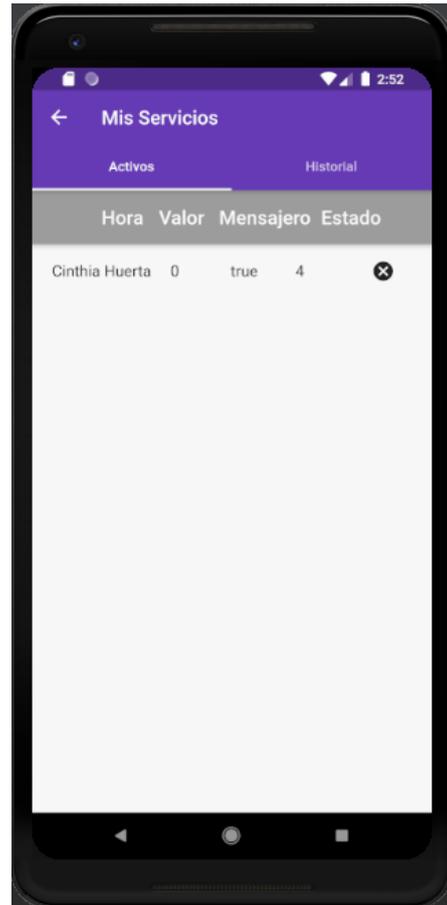
Vuelva a ingresar su contraseña

Actualizar Contraseña

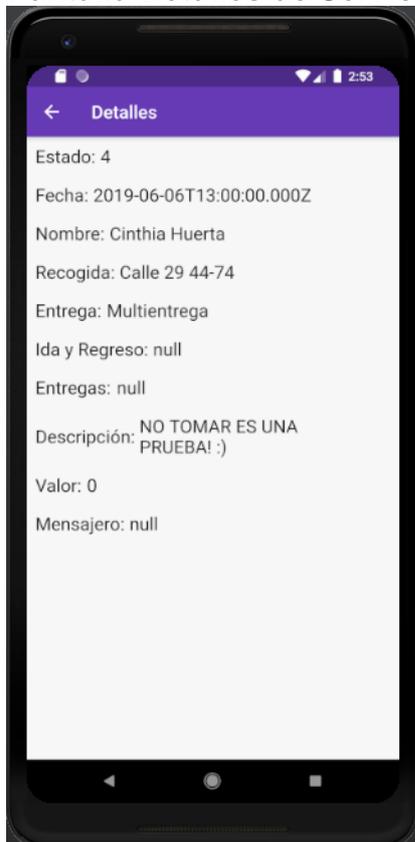
Contraseña actualizada correctamente



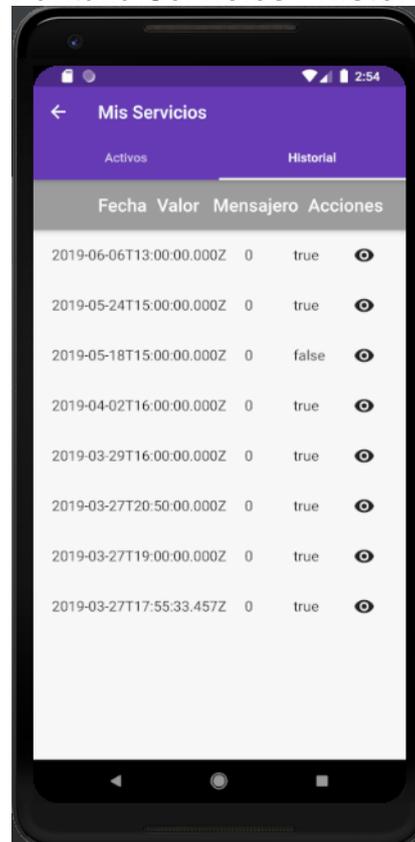
Pantalla Servicios - Activos



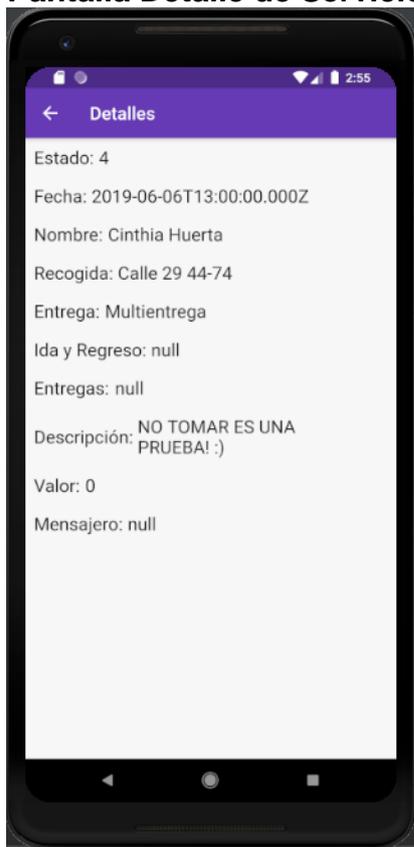
Pantalla Detalles de Servicio Activo



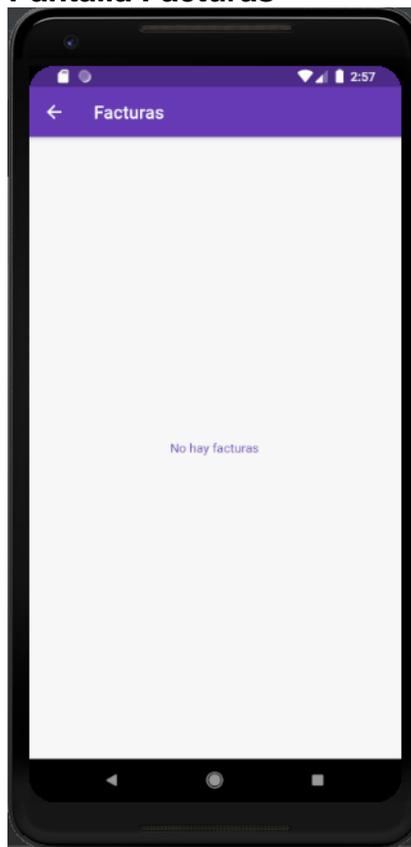
Pantalla Servicios - Historial



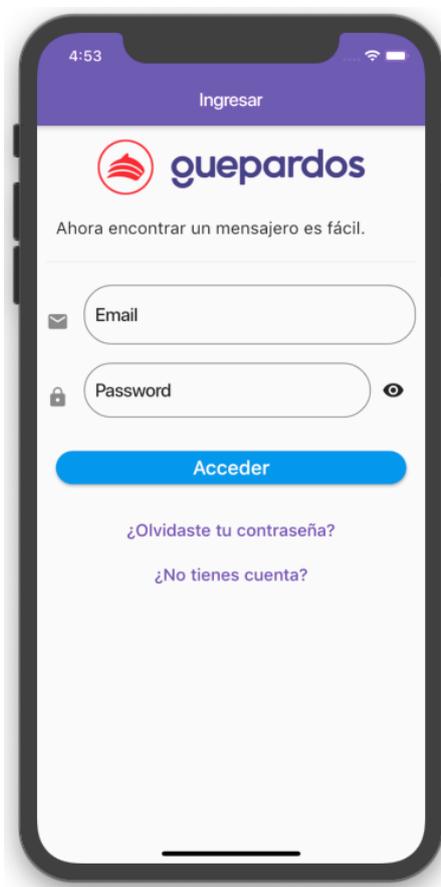
Pantalla Detalle de Servicio Historial



Pantalla Facturas



Aplicación ejecutada en iOS (iPhone XR)



Estructura de la aplicación

Estructura del archivo main.dart

```

▼ main() → void
  MyApp
▼ MyApp
  build(BuildContext context) → Widget
    MaterialApp
      Login
  
```

Estructura ActualizarPass.dart

```

ActualizarPass
  createState() → State<StatefulWidget>
  idUser
▼ ActualizarPassState
  pass → TextEditingController
  newPass → TextEditingController
  _formKey
  pass1 → String
  newPass1 → String
  build(BuildContext context) → Widget
    WillPopScope
      Scaffold
        AppBar
          Text('Actualizar Contraseña')
          IconButton
            Icon(Icons.arrow_back)
        Menu
        Form
          ListView
            Padding
  
```

Facturas.dart

```

Facturas
  createState() → State<StatefulWidget>
  idUser
  jsonData
  total
▼ FacturasState
  postFacturas() → Future<String>
  initState() → void
  build(BuildContext context) → Widget
    Scaffold
      AppBar
        Text('Facturas')
      Container
        Center
          Text("No hay facturas")
      Scaffold
        AppBar
          Text('Facturas')
        Menu
        ListView
          ListTile
            Row
              SecondPage
                SecondPage(this.jsonData)
                  jsonData
                    build(BuildContext context) → Widget
                      Scaffold
                        AppBar
                          Text('Detalles')
                        Center
                          ListView
                            Padding
                              Row
                                Text("Estado: ")
                                Text(jsonData["estad...o"].toString())
                              Padding
                              Padding
                              Padding
                              Padding
                              Padding
  
```

Login.dart

```
▼ Login
  createState() → State<StatefulWidget>
  recuperar → TextEditingController
  ▼ LoginState
    _formKey
    mailController → TextEditingController
    passController → TextEditingController
    _obscureText → bool
    _password → String
    _email → String
    ▼ build(BuildContext context) → Widget
      WillPopScope
        Scaffold
          AppBar
            Text('Ingresar')
          Form
            ListView
              Padding
                Image
              Padding
                Text('Ahora encontra...ero es fácil.')
          ▼ postdata() → Future<String>
            Pedidos
            Text('Ahora encontra...ero es fácil.')
            Padding
            Padding
              TextFormField
                Padding
            Padding
            Padding
              RaisedButton
                Text('Acceder')
            FlatButton
            FlatButton
            moveToLastScreen() → void
            _toggle() → void
            ▼ _showAlertDialog(String title, String message) → void
              AlertDialog
                Text(title)
                Text(message)
                RaisedButton
                  Text('OK')
            postdata() → Future<String>
            Pedidos
```

Menu.dart

```
▼ Menu
  createState() → State<StatefulWidget>
  ▼ MenuState
    ▼ build(BuildContext context) → Widget
      Drawer
        Column
          Padding
            Image
            Divider
          ListTile
            Icon(Icons.person)
            Text('Mi perfil')
            Perfil
            Divider
          ListTile
          Divider
          ListTile
          Divider
          ListTile
```



Pedidos.dart

```

▼ Pedidos
  createState() → State<StatefulWidget>
  PedidosState
    uuid
    _formKey
    direccionR → TextEditingController
    descripcion → TextEditingController
    direccionE → TextEditingController
    urgente → bool
    sinAfan → bool
    H8 → bool
    H10 → bool
    H13 → bool
    seleccionado → bool
    numero → int
    _regreso → bool
    _canasta → bool
    _regresoChanged(bool value) → void
    _canastaChanged(bool value) → void
    dateU → DateTime
    dateSA → DateTime
    date → DateTime
    _dateNow → DateTime
    hora → int
    fecha → String
  build(BuildContext context) → Widget
    WillPopScope
      Scaffold
        AppBar
        Menu
        Form
          ListView
            Padding
              Row
                Expanded
                  RaisedButton
                    Row
                      Text('Urgente!')
                      Container
                      Icon(Icons.flash_on)
                    Container
                  Expanded

```

```

      Expanded
        RaisedButton
          Row
            Text('Sin afán')
            Container
            Icon(Icons.access_time)
        Padding
          Text("* Selecciona u...o de servicio")
          Container
        Padding
          Text("¿Cuándo?")
        Padding
          Column
            Row
              Expanded
                Padding
                  DateTimePickerFormField
            Expanded
            Container
            Padding
            Container
          moveToLastScreen() → void
          _changed(String field) → void
          _changedRango(String field) → void
          _showAlertDialog(String title, String message) → void
            AlertDialog
              Text(title)
              Text(message)
              RaisedButton
                Text('OK')
          postData() → Future<String>

```



Perfil.dart

```

▼ Perfil
  m createState() → State<StatefulWidget>
  v idUser
  ▼ PerfilState
    f _formKey
    f nombre → TextEditingController
    f cedula → TextEditingController
    f email → TextEditingController
    f celular → TextEditingController
    f direccion → TextEditingController
    f ciudad → TextEditingController
  ▼ m build(BuildContext context) → Widget
    ▼ WillPopScope
      ▼ Scaffold
        ▼ AppBar
          Text('Perfil')
        ▼ IconButton
          Icon(Icons.arrow_back)
        Menu
        ▼ Form
          ▼ ListView
            ▼ FlatButton
              Text('Actualizar Contraseña')
              ActualizarPass
            ▼ Padding
              TextFormField
            ▼ Padding
              TextFormField
            ► Padding
            ► Padding
            ► Padding
            ► Padding
            ► Padding
            ► FlatButton
              m moveToLastScreen() → void
            ▼ m _showAlertDialog(String title, String message) → void
              ▼ AlertDialog
                Text(title)
                Text(message)
                ▼ RaisedButton
                  Text('OK')
            m putActualiz() → Future<String>

```

RecuperarPass.dart

```

▼ RecuperarPass
  m createState() → State<StatefulWidget>
  ▼ RecuperarPassState
    f recuperar → TextEditingController
    f _formKey
    f _emailR → String
  ▼ m build(BuildContext context) → Widget
    ▼ WillPopScope
      ▼ Scaffold
        ► AppBar
        ▼ Form
          ▼ Column
            ► Padding
            ▼ Padding
              TextFormField
            ▼ Padding
              RaisedButton
          m moveToLastScreen() → void
        ▼ m _showAlertDialog(String title, String message) → void
          ► AlertDialog
          m postRecuperar() → Future<String>

```



Registro.dart

```

▼ Registro
  createState() → State<StatefulWidget>
▼ RegistroState
  _formKey
  nombre → TextEditingController
  email → TextEditingController
  _opciones
  ciudad
  build(BuildContext context) → Widget
    WillPopScope
      Scaffold
        AppBar
          Text('Registro')
          IconButton
            Icon(Icons.arrow_back)
          Menu
        Form
          ListView
            Padding
              Padding
                TextFormField
          Padding
            Padding
              TextFormField
            RaisedButton
              Text('Registrar')
          FlatButton
            Text('¿Ya estás registrado?')
            Login
        moveToLastScreen() → void
        _showAlertDialog(String title, String message) → void
          AlertDialog
            Text(title)
            Text(message)
          RaisedButton
            Text('OK')
        getUser() → Future<String>
        postUser() → Future<String>
    Padding
      Text("Seleccione la Ciudad")
    Padding
      DropdownButton
        DropdownMenuItem
          Text(dropDownStringItem)
    Padding
      RaisedButton
        Text('Registrar')
    FlatButton
      Text('¿Ya estás registrado?')
      Login
    moveToLastScreen() → void
    _showAlertDialog(String title, String message) → void
      AlertDialog
        Text(title)
        Text(message)
      RaisedButton
        Text('OK')
    getUser() → Future<String>
    postUser() → Future<String>
  
```

Servicios.dart

```

▼ Servicios
  createState() → State<StatefulWidget>
▼ ServiciosState
  controller → TabController
  initState() → void
  dispose() → void
  build(BuildContext context) → Widget
    WillPopScope
      Scaffold
        AppBar
          Text('Mis Servicios')
          IconButton
            Icon(Icons.arrow_back)
          TabBar
            Tab
            Tab
          Menu
        TabBarView
          ServiciosAct
          ServiciosHist
        moveToLastScreen() → void
  
```

ServiciosAct.dart

```
▼ ServiciosAct
  m createState() → State<StatefulWidget>
  v idUser
  v jsonData
  v total
  ▼ ServiciosActState
    m postServices() → Future<String>
    m initState() → void
    ▼ m build(BuildContext context) → Widget
      ▼ Scaffold
        ▼ Container
          ▼ Center
            Text("No hay servicios")
      ▼ Scaffold
        ▼ AppBar
          Container
        ▼ Row
          ▼ Padding
            Text('Hora')
          ▼ Padding
            Text('Valor')
      ▼ Padding
        Padding
      ▼ Padding
        Padding
      ▼ ListView
        ▼ ListTile
          ▼ Row
            ▼ Padding
              Text(jsonData[i]["nombre"].toString())
            ▼ Expanded
              ▼ Padding
                Text(jsonData[i]["valor"].toString())
            ▼ Expanded
            ▼ Expanded
            ▼ Expanded
          SecondPage
        m _showAlertDialog(String title, String message) → void
          ▼ AlertDialog
            Text(title)
            Text(message)
            RaisedButton
            RaisedButton
      SecondPage
```

```
▼ SecondPage
  m SecondPage(this.jsonData)
  i jsonData
  ▼ m build(BuildContext context) → Widget
    ▼ Scaffold
      ► AppBar
      ▼ Center
        ▼ ListView
          ▼ Padding
            ▼ Row
              Text("Estado: ")
              Text(jsonData["estad...o"].toString())
          ▼ Padding
            ▼ Row
              Text("Fecha: ")
              Text(jsonData["fecha"])
          ► Padding
          ► Padding
          ► Padding
          ► Padding
          ► Padding
```



ServiciosHist.dart

```
▼ ServiciosHist
  createState() → State<StatefulWidget>
  idUser
  jsonData
  total
  ServiciosHistState
    url → String
    makeRequest() → Future<String>
    initState() → void
    build(BuildContext context) → Widget
      Scaffold
        Container
          Center
            Text("No hay servicios")
      Scaffold
        AppBar
        ListView
          ListTile
            Row
              Padding
                Text(jsonData[i]["fecha"])
          Expanded
          Expanded
          Expanded
          SecondPage
        SecondPage
          SecondPage(this.jsonData)
          jsonData
          build(BuildContext context) → Widget
            Scaffold
              AppBar
                Text('Detalles')
              Center
                ListView
                  Padding
                    Row
                      Text("Estado: ")
                      Text(jsonData["estad...o"].toString())
                    Padding
                    Padding
                    Padding
                    Padding
```

LoginModel.dart

```
▼ LoginModel
  userName → dynamic
  token → String
  email → String
  userId → int
  LoginModel(this.userName, this.token, this.email, this.userId)
  LoginModel.fromJson(Map<String, dynamic> json)
  toJson() → Map<String, dynamic>
```

RequestLoginApi.dart

```
▼ requestLoginAPI(BuildContext context, String username, String password) → Future<LoginModel>
  Pedidos
  showDialogSingleButton(BuildContext context, String title, String message, String buttonLabel) → void
  AlertDialog
    Text(title)
    Text(message)
  FlatButton
    Text(buttonLabel)
```

SaveCurrentLogin.dart

```
token
user
saveCurrentLogin(Map responseJson)
```

SaveLogout.dart

```
saveLogout()
```

Pubspec.yaml

```
pubspec.yaml C:\Users\cimah\AndroidStudioProjects\gps_app\guepardos
  < > YAML document
    < > name guepardos
    < > description A new Flutter application.
    < > version 1.0.0+1
    < > environment
      < > sdk >=2.1.0 <3.0.0
    < > dependencies
      < > flutter
      < > cupertino_icons ^0.1.2
      < > http ^0.11.3+16
      < > sembast ^1.15.3+1
      < > path_provider ^0.5.0+1
      < > shared_preferences ^0.4.2
      < > uuid 2.0.0
      < > datetime_picker_formfield ^0.1.8
    < > dev_dependencies
      < > flutter_test
    < > flutter
      < > uses-material-design true
      < > assets
```

Respuesta del servidor sobre los datos enviados en el pedido

```
{
  uuid: 3fc57cad-2374-427a-8626-975039443759,
  nombre: Cinthia Huerta,
  celular: 4651096893,
  email: cimahuva@gmail.com,
  mensajeroEmpresarial: true,
  recogida: Calle 29 44-74,
  idaVuelta: true,
  tipoTramite: Mensajería Express,
  valorDeclarado: 0,
  descripcion: No tomar, ES UNA PRUEBA! :),
  detalleE: ,
  detallere: ,
  ciudadR: Medellín,
  paqueteDocumento: false,
  fecha: 2019-06-06T21:00:00.000Z,
  valor: 0,
  latR: 6.184729879164248,
  lonR: -75.57695404149412,
  idServicioCliente: 5920d77d89b5da11007abb4d,
```



```
geoR: {lat: 6.184729879164248,  
      lng: -75.57695404149412},  
geoE: {lat: 0, lng: 0},  
excluidos: ,  
distancia: 0,  
estadoServicio: 4,  
pendiente: true,  
futuro: false,  
ciudadE: ,  
entrega: multientrega,  
fechaHoraEstados: [],  
fechaHoraEntregas: [],  
mensajeroId: ,  
tarifa: {fechaHora: 2019-03-05T09:00:02.963Z,  
        distancia: 0,  
        idaRegreso: true,  
        tiempoMinutos: 36,  
        tiempoMaximo: 2,  
        tipoServicio: Premium,  
        origen: Medellín,  
        destino: Carrera 45 Calle 51,  
        numEntregas: 2,  
        valorServicio: 0,  
        valorxEntrega: 0,  
        servicioId: 000000,  
        id: 140,  
        canasta: true,  
        },  
"documento": "123456",  
"tiempoEspera": "720",  
"estadoArcgis": 1,  
"plazo": "",  
"credito": true,  
"medioPago": false,  
"consolidado": false,  
"facturado": false,  
"liquidado": false,  
"tipoCobroAdicional": "",  
"valorCobroAdicional": 0,  
"compartido": false,  
"modificaciones": {  
  "creadoPor": {"user": "Cinthia Huerta",  
    "fecha": "2019-03-05T22:23:02.963Z"}  
},  
"enviarSms": false,  
"recogidaFormateada": "",  
"restaurante": false,  
"exitoso": true,  
"entregaFormateada": "",  
"pubsub": true  
}
```

Test de la aplicación final

La aplicación se ha probado mediante dispositivos Android y puede ejecutarse desde la mínima versión api 16 (Jelly Bean Versión 4.1) hasta la más actualizada que es la versión 28 (Pie Versión 9).

Se puede visualizar en dispositivos de tamaño de 4.0 pulgadas como el Nexus S como vemos en las Ilustraciones 29 y 30, hasta 6.3 pulgadas como el Pixel 3 XL en las ilustraciones 31 y 32



Ilustración 29 Pantalla principal en Nexus S

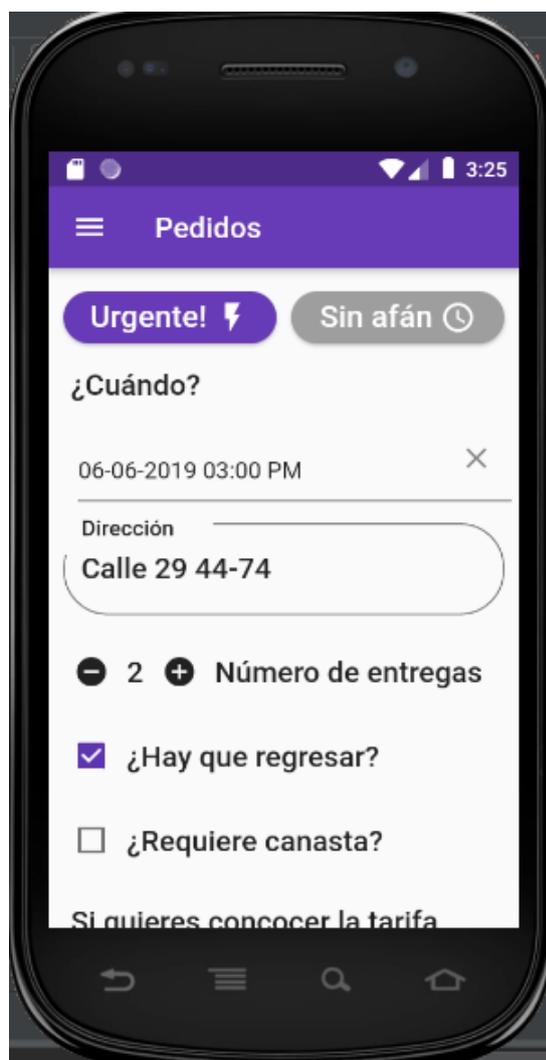


Ilustración 30 Pantalla Pedidos en Nexus S

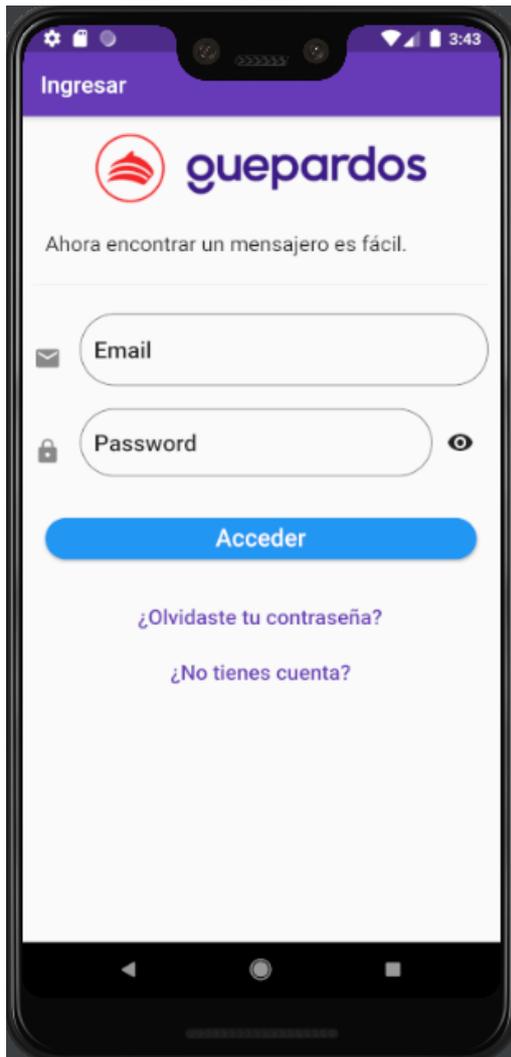


Ilustración 31 Pantalla principal en Pixel 3 XL

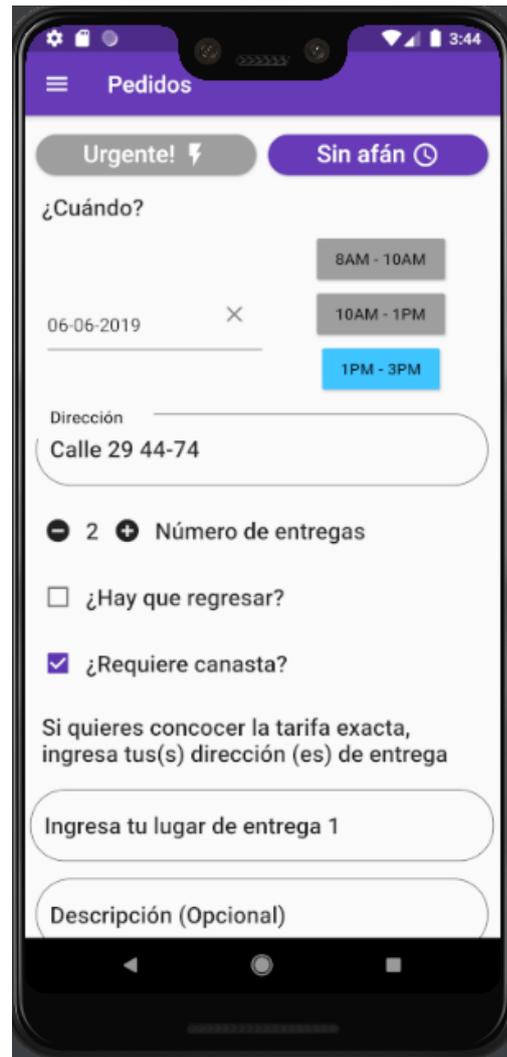


Ilustración 32 Pantalla Pedidos en Pixel 3 XL

CRONOGRAMA DE ACTIVIDADES

Tabla 5 Cronograma de actividades

Actividades por Quincena	Ene-1a	Ene-2a	Feb-1a	Feb-2a	Mar-1a	Mar-2a	Abr-1a	Abr-2a	May-1a	May-2a	Jun-1a
Apoyo en testing y pruebas a plataformas											
Revisión de calidad de productos en desarrollo											
Capacitación para en lenguajes de desarrollo											
Apoyo en prototipado de nuevos productos.											
Reporte de Issues de la plataforma											
Gestionar y solucionar Issues de plataformas											

RESULTADOS

Se han obtenido los resultados esperados propuestos en los objetivos, ya que se ha finalizado con todas las actividades propuestas en el cronograma y se ha entregado el producto final del prototipo. Durante mis prácticas profesionales en la empresa Evolución Digital LTD y trabajando para la empresa Guepardos he obtenido muchos conocimientos y experiencia.

Se lograron todos los objetivos propuestos desde el reporte preliminar los cuales explicaremos con la siguiente tabla:

Tabla 6 Resultados de Objetivos

Objetivo Propuesto	Resultado Esperado
Mejorar los ciclos de desarrollo mediante el apoyo en testing.	Se realizaron todas las pruebas a la plataforma web de usuarios
Realizar pruebas unitarias de los productos de Guepardos.	Se cumplió probando todas las funciones del sistema revisando los envíos en la plataforma de administradores
Optimizar continuamente en términos funcionales.	Se reportaron los errores en la plataforma git.guepardos para la solución y volver a probar
Obtener los conocimientos de los lenguajes de desarrollo para la aplicación.	Se realizaron cursos en Udemy obteniendo los certificados de finalización
Crear prototipos para validar nuevas funcionalidades a incorporar a la plataforma.	Prototipo creado de aplicación móvil para usuarios de manera nativa para iOS y Android

Después de realizar todas las pruebas necesarias de la plataforma web y verificar que no tuviera errores de ningún tipo se logró publicarla ya en producción, pueden encontrarla en este link: <https://guepardos.co/>

El prototipo de la aplicación móvil cumplió con todas las características necesarias para realizar un pedido, actualizar los datos de un usuario, mostrar sus servicios y facturas, ingresar a la aplicación, recuperar la contraseña, registrarse y actualizar la contraseña. Sólo es cuestión de agregarle algunos detalles para mejorarla y que esté completamente funcional.

CONCLUSIONES

En el proyecto realizado se ha logrado cumplir con los requisitos iniciales que se planteaban, tras el desarrollo del proyecto se han cumplido con todos los objetivos propuestos si son revisados uno a uno, lo que se puede considerar un éxito. Gracias a este proyecto se obtendrá un mayor alcance con los clientes en la plataforma web y un mayor uso, ya que es más simple de usar para todo tipo de usuarios ya hora que ya está en producción sin errores de funcionamiento, la empresa crecerá y podrá expandirse.

El prototipo de la aplicación ya quedó realizado cumpliendo sus características mas importantes en su uso, de esta manera, cuando la empresa ya necesite usar la app, sólo será necesario agregar algunas mejoras como el uso de coordenadas en las direcciones o como el seguimiento del pedido en tiempo real para que la aplicación esté en sus mejores condiciones.

Se logró todo con mucha dedicación y aprendizaje a lo largo de mis prácticas en Evolución Digital trabajando en este proyecto para Guepardos. Se obtuvo mucha experiencia y conocimiento que será de mucha ayuda al momento de trabajar en alguna empresa de desarrollo de software, ya que ya tengo lo necesario para realizar una aplicación móvil en las tecnologías más actuales hasta el momento y realizar todas las pruebas necesarias en una plataforma web.

COMPETENCIAS DESARROLLADAS

Competencias desarrolladas y/o aplicadas.

1. Gestiona los problemas analizándolos para la toma de decisiones y así reportándose a la persona indicada para su solución
2. Aplica métodos cuantitativos y cualitativos en el análisis e interpretación de datos y modelado de sistemas en los procesos organizacionales, realizando reportes sobre los clientes de Guepardos.
3. Realiza pruebas funcionales y de calidad del tipo “Caja Negra” en el uso y mantenimiento de la plataforma web Guepardos al igual que pruebas de “Caja Blanca” en el prototipo de la aplicación móvil
4. Participa en la mejora de calidad, diseño, correcciones y funcionamiento en los procesos de proyectos realizados y en la empresa
5. Aprende sobre temas de desarrollo utilizados en el proceso de creación, prueba y diseño del prototipo propuesto, solucionando los problemas de compatibilidad o errores que se presenten
6. Diseña, crea e innova estructuras de proyectos de desarrollo aplicándose a prototipos de aplicaciones móviles nativas para iOS y Android
7. Maneja plataformas de desarrollo para la creación de prototipos, los cuales están diseñados para la utilización de los usuarios
8. Aplica habilidades directivas y de ingeniería en el diseño, gestión, fortalecimiento e innovación de las organizaciones para la toma de decisiones en forma efectiva
9. Utiliza las nuevas tecnologías de información y comunicación en la organización, para optimizar los procesos y la eficaz toma de decisiones.
10. Investiga sobre los procedimientos necesarios para hacer buenas prácticas en programación e implementarlas al proyecto
11. Organiza las actividades a realizar teniendo un buen control del tiempo que se toma al desarrollar una tarea y de esta manera optimizando el trabajo que se realiza por horas productivas
12. Trabaja en equipo con sus compañeros teniendo una comunicación asertiva, organizando todas las tareas a realizar y desarrollando todas las actividades que se les asignan durante el horario de trabajo

FUENTES DE INFORMACIÓN

- Altarade, M. (20 de Abril de 2017). *Pandora fms*. Obtenido de Pandora fms:
<https://blog.pandorafms.org/es/bases-de-datos-nosql/>
- Carlos, V. L. (Febrero de 2008). *Eumed*. Obtenido de Eumed:
<http://www.eumed.net/libros-gratis/2008a/351/Calidad%20de%20Software.htm>
- Developers, G. (04 de Diciembre de 2018). *Developers Google Blog*. Obtenido de Developers Google Blog:
<https://developers.googleblog.com/2018/12/flutter-10-googles-portable-ui-toolkit.html>
- E, D. L. (23 de Agosto de 2018). *Mokware*. Obtenido de Mokware:
<http://mokware.com.mx/blog/una-plataforma-digital/>
- Excelencia, P. T.-C. (11 de Febrero de 2015). *Panel*. Obtenido de Panel:
<https://www.panel.es/blog/software-qa-cuales-son-los-tipos-de-pruebas-software/>
- JC, P. (29 de Mayo de 2013). *Ecured*. Obtenido de
https://www.ecured.cu/Pruebas_de_software
- Medrano, E. L. (29 de Septiembre de 2016). *Infortelecom*. Obtenido de Infortelecom: <https://infortelecom.es/blog/que-es-un-servidor-y-para-que-sirve/>
- Pimienta, P. (05 de 05 de 2014). *De Idea a App*. Obtenido de De Idea a App:
<https://deideaaapp.org/tipos-de-aplicaciones-moviles-y-sus-caracteristicas/>
- Rouse, M. (Julio de 2017). *Search Data Center*. Obtenido de Search Data Center:
<https://searchdatacenter.techtarget.com/es/definicion/Desarrollo-de-aplicaciones-moviles>
- Sánchez, J. (24 de Julio de 2014). *Frontend Labs*. Obtenido de Frontend Labs:
<https://frontendlabs.io/1490--json-que-es-json-parse-json-stringify>
- Thomson, M. (14 de Noviembre de 2018). *Medium*. Obtenido de Medium:
https://medium.com/dartlang/announcing-dart-2-1-improved-performance-usability-9f55fca6f31a?source=collection_home---4-----5---

ANEXOS

Creación de reporte de clientes de Guepardos

Se creó un reporte sobre los últimos 3 meses, creado en JavaScript donde el programa recorre toda la base de datos de la plataforma, los resultados se muestran a continuación:

- Total de clientes Guepardos

3 Meses	
Total de Clientes:	5253
Promedio entregas:	2.427374833
Mínimo 1 pedido:	1674

- El servicio más solicitado:

3 Meses	
	Total Premium: 5121
	Total Domicilios: 2116
Servicio más solicitado -->	Total Diligencias: 5509

- Total de clientes que realizan sólo un pedido

3 Meses	
Clientes 1 pedido:	732

Código utilizado que imprime los valores en un CSV

```
const axios = require('axios')
const moment = require('moment')
const lodash = require('lodash')
const fs = require('fs')

const writeDataResponse = (data) => {
  fs.appendFile('reporte.csv', data, 'utf8', function (err) {
    if (err) {
      return console.log('Some error occurred - file either not saved or corrupted file saved.');
    } else {
      return console.log('It\'s saved!');
    }
  });
}

const getServices = id => {
  const startDate = moment().subtract(6, 'months').startOf('day').utc().toISOString()
  const endDate = moment().endOf('day').utc().toISOString()
  const objquery = {
    fields: {
      id: true,
      tarifa: true,
      calificacion: true
      //campos que necesites del servicio
    },
    where: {
      fecha: { between: [startDate, endDate] },
      idServicioCliente: id
    }
  }
  const query = JSON.stringify(objquery)
  return axios.get(`http://backend.guepardos.co:3000/api/Servicios?filter=${query}`)
    .then(response => response.data)
    .catch(error => error.response)
}

const main = () => {
  const objQuery = {
    fields: {
      id: true,
```



```
id: true,
nombre: true,
documento: true,
email: true,
calificacion: true
//tipoServicio: true
},
where: {
  realm: 'user'
}
}
const query = JSON.stringify(objQuery)
axios.get(`http://backend.guepardos.co:3000/api/users?filter=${query}&access_token=UDCjPRDAN9PjCgNrmNtLN7k6GAose9BBFeoY
.then(async response => {
  let users = response.data
  for (let index = 0; index < users.length; index++) {
    let user = users[index];
    const services = await getServices(user.id) // en es6 await convierte una funcion en promesa, pero hay que convertir
    user.totalServices = services.length
    console.log(user.nombre)
    //console.log(`${user.id}, ${user.nombre}, ${services.calificacion} \n`)
    // const save = writeDataResponse(`${user.id}, ${user.nombre}, ${services.premium}, ${services.domicilios}, ${service
  }
  // aqui al finalizar guardas en un csv cada usuario y en totalServices esta el numero de servicios que pidio en los us
  //console.log(users)
})
.catch(err => console.log(err))
}
main()
```