**UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS**



IS-513: Lenguajes de Programación

Informe - Proyecto Delivery

Sección: 1900

Estudiantes: Carlos Javier Sosa Reyes - 20171030678

Cristian Javier Medina Rodriguez – 20231003400

David Fernando Parada Matute – 20211002496

Eduardo Gabriel Martinez Zelaya - 20121010326

Licenciado: Harold Roberto Coello Matamoros

Fecha de Entrega: 07/04/2025

Índice

[**Introducción 4**](#_jy2lkrmqlj7t)

[**Marco Teórico 5**](#_9ghv2z8ze836)

[API 5](#_qo5u51df46ui)

[API Web 5](#_ltcux3qu0dcz)

[API Rest 5](#_pu87n8rtz6q6)

[REST 5](#_20ru2fqqh3pt)

[Arquitectura de Microservicios 5](#_79h4ft2mkfav)

[Back-End 5](#_qyruvj7z4cay)

[Front-End 6](#_fbhrtz801qw)

[Base de Datos 6](#_dm04zzaafgr6)

[DBMS 6](#_myi49pz9nk3h)

[SQL 6](#_3r0ltrk8ir3j)

[Base de Datos Relacional 7](#_e57ohf2ybmfl)

[MySQL 7](#_dgvqrnhx3ssd)

[Servidor Web 7](#_4uwun1kctpdp)

[Servicio Web 7](#_kae5b6vrktzd)

[Apache 7](#_coj3hvi1pqud)

[Java 8](#_dywo3j2v23s8)

[PHP 8](#_zcqtn1d4pukl)

[PhpMyAdmin 8](#_srcwkqb4xoss)

[Laravel 8](#_in2r0gf7zrf5)

[SpringBoot 8](#_xqqqme9leh1r)

[Spring Framework 8](#_6sxqixlqvlqv)

[BootStrap 5 9](#_dot88oub7xpo)

[ORM 9](#_dnxxevnzsncp)

[Hibernate ORM 9](#_738lhkbc0tec)

[Swagger 10](#_fi7074r5o2j9)

[Editor de Swagger 10](#_iyad8788k8vb)

[IU de Swagger 10](#_d948lsi061er)

[Swagger Codegen 10](#_gw9buwdv0izr)

[Composer 10](#_uc0rwab6k0bx)

[JQuery 10](#_o0l71pfxffce)

[JavaScript 11](#_jn2vhl6ph2kx)

[GitHub 11](#_bx8gm8i3qqdj)

[Git 11](#_crqt3ym6i6db)

[**Explicación de las Funcionalidades 12**](#_qhfsy6mhghf9)

[1-Registro de Clientes 12](#_8klmayeofg9w)

[2-Inicio de Sesion 12](#_g28ng65sful4)

[3-Empresa 12](#_rodgph4isojz)

[4-CrudUsuario 12](#_f3q2ase9i0sf)

[5-Método de Pago 12](#_n92ym81ld86a)

[6-Roles 12](#_bgrlkl2bjm4o)

[7-Creación del Carrito de Compras 12](#_p3caupcnc04v)

[9-Hacer Pedido 13](#_1bbxhr85k08q)

[10-Verificación del estado del Repartidor 13](#_c2p0tr9049ey)

[12-Entrega de Pedido 13](#_t15k9k8t2ac7)

[13-Reporte-Historial de Repartidor 13](#_ci3w1grt57vb)

[14-Reporte-Pedidos y Ganancia 13](#_dh57edtuubq0)

[15-Reporte-Repartidores 13](#_wun9cn5l74td)

[16-Reporte-Empresas 13](#_c5dfl4109uzs)

[17-Reporte-Historial del Cliente 13](#_v6m8ltyf0nkp)

[19-Manejo de Ubicaciones 13](#_lidxokfa9d93)

[**Conclusiones 14**](#_p5phojptmzyj)

# Introducción

Este informe es sobre el Proyecto Delivery para la clase IS-513: Lenguajes de Programación.

Para este proyecto se ha creado una aplicación de Delivery en la cual un cliente puede hacer pedidos a diferentes empresas por una variedad de productos diferentes, simulando aplicaciones ya existentes y reconocidas como Amazon, Alibaba, EBay, PedidosYa, etc. Basándose en una arquitectura de microservicios con JQuery, el proyecto se separa en Front-End (parte visual e interactiva con el cliente, Laravel) y Back-End (parte lógica del sistema, SpringBoot) para poner en práctica el uso de APIs. En cuanto a utilidades para el proyecto, se utilizó MySQL y PhpMyAdmin para base de datos; Laravel y PHP con JQuery para el front-end; Java con SpringBoot para el back-end.

La funcionalidad principal del proyecto es simular *hacer un pedido* para un cliente en la aplicación como si comprara productos de las empresas asociadas, pagara en efectivo o con tarjeta y finalmente su repartidor llegue a entregarle su pedido.

Cada usuario - ya sea de tipo cliente, repartidor, o supervisor de empresa - tiene asignada una dirección que contiene su ubicación geográfica o coordenadas geográficas de las cuales se basa la aplicación para calcular distancias y tiempos de espera (dada la velocidad del vehículo del repartidor). Con esta función, cuando un cliente ingresa a la app, se le muestran las empresas más cercanas a él, y cuando hace un pedido, se busca el repartidor más cercano a la empresa para asignarse al pedido.

Es importante destacar que en este proyecto no se contempla el manejo de fondos ni transacciones, por lo que siempre se asume que el cliente **puede** comprar.

El proyecto Delivery fue hecho por Carlos Javier Sosa Reyes, Cristian Javier Medina Rodriguez, David Fernando Parada Matute y Eduardo Gabriel Martinez Zelaya. El enlace del repositorio de GitHub del proyecto es este: <https://github.com/Edumeka/delivery>

# Marco Teórico

## API

Una API, o interfaz de programación de aplicaciones, es un conjunto de reglas o protocolos que permiten que las aplicaciones de software se comuniquen entre sí para intercambiar datos, características y funcionalidades. (IBM Corporation & Goodwin, 2024)

### API Web

Se utilizan para permitir la transferencia de datos y funcionalidades a través de Internet mediante el protocolo HTTP. (IBM Corporation & Goodwin, 2024)

## API Rest

Una API REST (API RESTful) es una [interfaz de programación de aplicaciones (API)](https://www.ibm.com/mx-es/topics/api) que se ajusta a los principios de diseño del estilo arquitectónico de transferencia de estado representacional. Proporcionan una forma flexible y ligera de integrar aplicaciones y conectar componentes en arquitecturas de [microservicios](https://www.ibm.com/mx-es/topics/microservices). (IBM Corporation, s.f.)

### REST

REST es un conjunto de principios de arquitectura de API web. [Las API de REST](https://www.ibm.com/mx-es/topics/rest-apis), también conocidas como API RESTful, son API que cumplen con ciertas restricciones de la arquitectura de REST. Las API de REST utilizan solicitudes HTTP como GET, PUT, HEAD y DELETE para interactuar con los recursos. REST hace que los datos estén disponibles como recursos, y cada recurso está representado por un URI único. Los clientes solicitan un recurso proporcionando su URI.(IBM Corporation & Goodwin, 2024)

## Arquitectura de Microservicios

[Microservicios](https://www.ibm.com/mx-es/topics/microservices) es un estilo arquitectónico que divide una aplicación en componentes independientes más pequeños, a menudo conectados mediante API de REST. La creación de una aplicación como una colección de servicios independientes permite a los desarrolladores trabajar en un componente de la aplicación independientemente de los demás, y hace que las aplicaciones sean más fáciles de probar, mantener y escalar. (IBM Corporation & Goodwin, 2024)

## Back-End

El backend es la parte invisible pero esencial de un sitio, encargada de manejar la lógica y el procesamiento de datos necesarios para que todo funcione de manera correcta y segura.

El backend se ocupa de tareas como almacenar y recuperar datos de una base de datos, procesar formularios, autenticar usuarios y gestionar la seguridad del sitio. Es responsable de toda la “magia” que sucede detrás de la interfaz visible.

El Backend también accede al servidor, que es una aplicación especializada que entiende la forma en la que el navegador hace solicitudes. (Chapaval, 2025)

## Front-End

Frontend es la parte de un programa, sitio web o dispositivo a la que un usuario puede acceder directamente. En el contexto de diseño web y desarrollo web, se refiere a todas las tecnologías que corren en el navegador y que se encargan de la interactividad con los usuarios.

El frontend es como la cara visible del sitio, donde se muestra:

* El diseño
* Los colores
* Los botones
* Todo lo que puedes ver en la pantalla

También se encarga de hacer que todo funcione correctamente y de responder a tus acciones, como hacer clic en un botón o desplazarte por la página. (Chapaval, 2025)

## Base de Datos

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un [sistema de gestión de bases de datos (DBMS)](https://www.oracle.com/mx/database/what-is-database/#WhatIsDBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos. (Oracle Corporation, 2020)

### DBMS

Un DBMS sirve como interfaz entre la base de datos y sus programas o usuarios finales, lo que permite a los usuarios recuperar, actualizar y gestionar cómo se organiza y se optimiza la información. Un DBMS también facilita la supervisión y el control de las bases de datos, lo que permite una variedad de operaciones administrativas como la supervisión del rendimiento, el ajuste, la copia de seguridad y la recuperación. (Oracle Corporation, 2020)

### SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje de programación estandarizado, específico de un dominio, que destaca en el manejo de relaciones de datos. Se utiliza mucho para almacenar, manipular y recuperar datos en sistemas como MySQL, SQL Server y Oracle.

Cuando es necesario recuperar datos de una base de datos, se utiliza SQL para realizar la petición. SQL es utilizado por administradores de bases de datos, desarrolladores y analistas de datos para tareas como la definición de datos, el control de acceso, el intercambio de datos, la escritura de scripts [de integración de datos](https://www.ibm.com/es-es/topics/data-integration) y la ejecución [de consultas analíticas](https://www.ibm.com/es-es/topics/big-data-analytics). (Mucci, 2024)

### Base de Datos Relacional

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las [bases de datos](https://www.oracle.com/mx/database/what-is-database/) relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos. (Oracle Corporation, s.f.)

## MySQL

MySQL es un RDBMS de código abierto que utiliza SQL para crear y gestionar bases de datos. Como base de datos relacional, MySQL almacena datos en tablas de filas y columnas organizadas en esquemas. Un esquema define cómo se organizan y almacenan los datos y describe la relación entre varias tablas. Con este formato, los desarrolladores pueden almacenar, recuperar y analizar fácilmente muchos tipos de datos, incluidos texto simple, números, fechas, horas y, más recientemente, JSON y [vectores](https://blogs.oracle.com/mysql/post/become-an-ai-developer-with-mysql-heatwave).(Erickson, 2024)

## Servidor Web

Un servidor web, webserver o servidor HTTP es la instancia encargada de mediar entre un [usuario](https://concepto.de/usuario/) de [internet](https://concepto.de/internet/) y el servidor donde se encuentran alojados los datos de la [página web](https://concepto.de/pagina-web/) que desea visitar. Se trata de un [software](https://concepto.de/software/) de comunicaciones que maneja el protocolo de transferencia de [hipertextos](https://concepto.de/hipertexto/) (HTTP). Por lo tanto, permite conexiones bidireccionales o unidireccionales, síncronas o asíncronas, con cualquier aplicación del usuario, como los [navegadores web](https://concepto.de/navegador-web/).

Esta comunicación entre el servidor web y el navegador del usuario se da bajo un lenguaje común, de acuerdo a varios métodos de petición de datos, como son las secuencias de instrucciones GET y POST. A través de ellas, la transmisión de información entre los servidores involucrados y la computadora del usuario ocurre de manera controlada. (Raffino & Equipo Editorial, Etecé, 2025)

### Servicio Web

Un servicio web es un componente de software de Internet que facilita las transferencias de datos a través de una red. Dado que un servicio web expone los datos y la funcionalidad de una aplicación a otras aplicaciones, en efecto, cada servicio web es una API. Sin embargo, no todas las API son un servicio web. (IBM Corporation & Goodwin, 2024)

## Apache

Apache HTTP Server, también conocido simplemente como Apache, es un servidor web de código abierto que permite a los usuarios alojar sitios web. Funciona recibiendo solicitudes de los navegadores y devolviendo archivos como HTML, imágenes o aplicaciones. Es compatible con múltiples sistemas operativos y ofrece una arquitectura modular, lo que permite personalizar sus funciones según las necesidades del proyecto. (Pontia tech, 2024)

## Java

Java es un lenguaje de programación orientado a objetos y una plataforma de software ampliamente empleados que se ejecuta en miles de millones de dispositivos, incluidas computadoras portátiles, dispositivos móviles, consolas de juegos, dispositivos médicos, entre otros. Las reglas y la sintaxis de Java se basan en los lenguajes C y C++. (IBM Corporation, s.f.)

## PHP

PHP es un lenguaje de «scripting» que puede ser embebido en HTML. Gran parte de su sintaxis se toma prestada de C, Java y Perl con un par de características específicas propias de PHP. El objetivo del lenguaje es permitir a los desarrolladores web escribir con rapidez páginas generadas dinámicamente.

PHP significa PHP: Hypertext Preprocessor. Esto confunde a mucha gente porque la primera palabra del acrónimo es el acrónimo mismo. (The PHP Documentation Group, s.f.)

## PhpMyAdmin

PhpMyAdmin is a free software tool written in PHP that is intended to handle the administration of a MySQL or MariaDB database server. You can use phpMyAdmin to perform most administration tasks, including creating a database, running queries, and adding user accounts. (The phpMyAdmin devel team, s.f.)

## Laravel

Laravel es un marco de aplicaciones web con sintaxis expresiva y elegante. Un marco web proporciona una estructura y un punto de partida para crear su aplicación, lo que le permite concentrarse en crear algo sorprendente mientras sudamos los detalles.

Laravel se esfuerza por proporcionar una experiencia de desarrollador sorprendente al tiempo que proporciona características potentes, como inyección de dependencia exhaustiva, una capa de abstracción de base de datos expresiva, colas y trabajos programados, pruebas de unidades e integración, y más. (Laravel, s.f.)

## SpringBoot

Spring Boot es un framework de código abierto cuyo objetivo es facilitar el desarrollo de aplicaciones basadas en el ecosistema de Spring. Ha sido diseñado para simplificar la configuración y el despliegue de aplicaciones, ayudando a los desarrolladores a centrarse menos en las tareas tediosas de configuración. Así, estos pueden dedicar más tiempo a la lógica de negocio, que es donde pueden aportar más valor. (Fernández, 2025)

### Spring Framework

Java Spring Framework (Spring Framework) es un popular marco de trabajo empresarial de código abierto que sirve para crear aplicaciones autónomas de producción que se ejecutan en una máquina virtual Java (JVM).

Java Spring Boot (Spring Boot) es una herramienta que acelera y simplifica el desarrollo de microservicios y aplicaciones web con Spring Framework gracias a tres funciones principales:

* Configuración automática
* Un enfoque de configuración obstinado
* La capacidad de crear aplicaciones autónomas

Estas características, combinadas, conforman una herramienta que le permite configurar una aplicación basada en Spring con el mínimo de instalación y configuración. (IBM Corporation, s.f.)

## BootStrap 5

Bootstrap es un framework CSS de código abierto orientado a la creación de interfaces de usuario para la web. Fue creado inicialmente por Twitter, aunque actualmente se encuentra bajo la dirección de una nutrida comunidad.

Cosechó un gran éxito y muchos seguidores desde su creación, lo que lo han convertido en uno de los frameworks de [front-end](https://www.arsys.es/blog/frontend-que-es-y-para-que-se-utiliza-en-desarrollo-web) más populares desde hace años.

Es un framework basado en componentes, lo que quiere decir que, con Bootstrap podemos conseguir una nutrida biblioteca de interfaces listas para usar, como botones, menús, tarjetas, acordeones, tabs, etc. Estas interfaces se usan con simples clases de CSS, mediante las que se aporta el estilo de una manera cómoda y rápida. Además, ocasionalmente las interfaces pueden requerir algo de Javascript (que ya te aporta el propio framework) para funcionar. Todas las interfaces además son «responsive», es decir, completamente adaptables a ordenadores y dispositivos móviles. (*Bootstrap: Qué Es, Para Qué Sirve Y Cómo Usarlo | Blog De Arsys*, 2024)

## ORM

ORM significa Mapeo de objetos Relacionales (ORM) es una técnica de programación para convertir datos entre bases de datos relacionales y lenguajes de programación orientados a objetos como Java, C#, etc. (Tutorials Point, s.f.)

Es una herramienta que realiza un mapeo que nos permite transformar los objetos de la base de datos como tablas y esquemas a clases con atributos en código de programación para poder manipular la información de una forma más fácil sin requerir de SQL. (Moya & Rodriguez, 2022)

## Hibernate ORM

Hibernate es un framework de mapeo objeto-relacional (ORM) para Java que facilita la persistencia de datos. Con Hibernate, los desarrolladores pueden mapear los objetos de Java a las tablas de la base de datos y viceversa, usando metadatos XML o anotaciones en el código Java. Hibernate no solo maneja este mapeo, sino que también proporciona una serie de servicios adicionales como la gestión de transacciones y la manipulación de consultas. (CertiDevs, 2001)

## Swagger

Swagger es un conjunto de herramientas de código abierto construidas alrededor de la especificación de OpenAPI que puede ayudar a diseñar, construir, documentar y consumir API REST. (SmartBear Software, 2025)

Dispone de una serie de herramientas de terceros que puede utilizar con documentos Swagger :

### Editor de Swagger

Le ayuda a crear un documento Swagger desde un navegador web proporcionando una vista en paralelo del documento Swagger y las definiciones de API REST resultantes.

### IU de Swagger

Le permite visualizar y probar una API REST definida con Swagger desde cualquier navegador web. Las funciones de pruebas incorporadas le permiten especificar las entradas de una operación definida en esa API REST, llamar a dicha operación desde el navegador web e inspeccionar los resultados de llamar a dicha operación.

### Swagger Codegen

Genera un kit de desarrollo de software (SDK) en varios lenguajes, incluidos Java™, Objective-C, PHP y Python, a partir de un documento Swagger para una API REST. El SDK resultante se puede utilizar para incorporar llamadas a las operaciones en dicha API REST en un programa de software escrito en uno de los idiomas soportados, sin necesidad de manejar el transporte HTTP subyacente. (IBM Corporation, 2025)

## Composer

Composer es un manejador de paquetes para PHP que proporciona un estándar para administrar, descargar e instalar dependencias y librerías. Similar a NPM en Node.js y Bundler en Ruby, Composer es la solución ideal cuando trabajamos en proyectos complejos que dependen de múltiples fuentes de instalación. En lugar de tener que descargar cada dependencia de forma manual, Composer hace esto de forma automática por nosotros. (Yair, 2019)

## JQuery

JQuery es una biblioteca de JavaScript ligera y potente que facilita la interacción con el HTML de una página web. Su uso permite simplificar tareas complejas en el desarrollo web, como manipulaciones del DOM, o trabajar con eventos y animaciones, utilizando una sintaxis fácil de entender y aplicar.

Esto ha convertido a jQuery en una herramienta valiosa para desarrolladores que buscan mejorar la interactividad y el rendimiento de las páginas web sin sacrificar por ello la velocidad de carga o la experiencia del usuario​. (Martinez & Equipo de Contenidos de GoDaddy, 2024)

### JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas​ y JavaScript del lado del servidor (Fundación Wikimedia, 2025)

## GitHub

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente:

* Presentar o compartir el trabajo.
* Seguir y administrar los cambios en el código a lo largo del tiempo.
* Dejar que otros usuarios revisen el código y realicen sugerencias para mejorarlo.
* Colaborar en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, en el que se basa GitHub. (GitHub, Inc., s.f.)

### Git

Git es un sistema de control de versiones que realiza un seguimiento de los cambios en los archivos. Git es especialmente útil cuando un grupo de personas y tú estáis haciendo cambios en los mismos archivos al mismo tiempo.

Normalmente, para hacerlo en un flujo de trabajo basado en Git, harías lo siguiente:

* Crear una rama a partir de la copia principal de archivos en los que tú (y tus colaboradores) estáis trabajando.
* Realizar modificaciones en los archivos de forma independiente y segura en tu propia rama personal.
* Dejar que Git fusione mediante combinación y de forma inteligente los cambios específicos en la copia principal de archivos, de modo que los cambios no afecten a las actualizaciones de otras personas.
* Dejar que Git realice un seguimiento de tus cambios y los de otras personas, por lo que todos siguen trabajando en la versión más actualizada del proyecto. (GitHub, Inc., s.f.)

# Explicación de las Funcionalidades

## 1-Registro de Clientes

Creación de cuentas para usuarios, estos pueden ser de tipo Cliente, Repartidor, Admin o Supervisor de Empresa. Los clientes pueden crear su propia cuenta desde la vista de Inicio de Sesión mientras que el resto de usuarios son creados por el admin. Sus campos son: nombre, apellido, correo, contraseña y DNI.

## 2-Inicio de Sesion

Luego de haberse creado usuarios tipo Cliente, pueden ingresar a la aplicación desde la misma vista al ingresar su correo y contraseña. Si las credenciales son correctas, entrarán a la Pantalla de Inicio donde se muestran las empresas disponibles de la app; si las credenciales no son correctas aparecerá un mensaje de error indicando que alguna de ellas no es correcta.

## 3-Empresa

Funcionalidad del lado del admin. Permite realizar 3 operaciones básicas del Crud para empresas, obtener, editar y eliminar empresas de la app. Sus campos son: nombreEmpresa, costoEnvio y RTN.

## 4-CrudUsuario

Funcionalidad del lado del admin. Permite realizar las 4 operaciones básicas del Crud para usuarios, crear/agregar, obtener, editar y eliminar usuarios de la app.

## 5-Método de Pago

Permite al cliente seleccionar su método de pago ya sea en efectivo o tarjeta.

## 6-Roles

Los roles determinan el *tipo* de usuario, este puede ser: Admin, Cliente, Repartidor o Supervisor de Empresa.

## 7-Creación del Carrito de Compras

Antes de crear un pedido, cuando un cliente aún está decidiendo qué productos comprar, se crea un carrito de compras en el cual se guarda la lista de productos escogidos hasta el momento y muestra el subtotal de los precios de los productos en el carrito.

8-Hacer Pago

Una vez decidido el carrito de compras, se registra el pago y se continúa a hacer el pedido.

## 9-Hacer Pedido

Una vez el cliente ha realizado el pago, se pasan los datos del carrito a la nueva instancia *pedido* y se elimina el carrito. El pedido también asigna un repartidor cercano disponible para entregarlo.

## 10-Verificación del estado del Repartidor

Cuando se asigna un repartidor a un pedido, primero se verifica que su estado sea *disponible* de lo contrario dará un mensaje de error para intentarlo de nuevo más tarde.

11-Tiempo de Espera

Se le muestra al cliente el tiempo de espera de acuerdo a la distancia de la empresa al cliente y la velocidad del vehículo del repartidor.

## 12-Entrega de Pedido

Una vez el repartidor llega al cliente y le entrega su pedido, se elimina la instancia de pedido y se modifica el estado del repartidor.

## 13-Reporte-Historial de Repartidor

Funcionalidad del lado del admin. Muestra todos los pedidos entregados por un repartidor particular.

## 14-Reporte-Pedidos y Ganancia

Funcionalidad del lado del admin. Muestra todos los pedidos que han sido completados y la ganancia por cada uno de ellos (suma total de los costos de envío o tarifas de envío de cada pedido).

## 15-Reporte-Repartidores

Funcionalidad del lado del admin. Muestra todos los repartidores de la app.

## 16-Reporte-Empresas

Funcionalidad del lado del admin. Muestra todas las empresas asignadas a la app.

## 17-Reporte-Historial del Cliente

Funcionalidad del lado del admin. Muestra todos los pedidos que ha realizado un cliente.

18-Reporte-Productos Más Vendidos

Funcionalidad del lado del admin. Muestra los productos más vendidos y la empresa que los vende.

## 19-Manejo de Ubicaciones

Funciones que buscan las direcciones asociadas a los usuarios (clientes y repartidores) y las empresas para obtener su ubicación (coordenadas geográficas) y calcular distancias entre el repartidor, la empresa y el cliente así como calcular el tiempo de espera para la entrega del pedido de un cliente luego de obtener la velocidad del vehículo del repartidor.

# Conclusiones

* La funcionalidad principal del cliente relacionada con la realización de pedidos opera de manera adecuada, integrando correctamente todas las funciones asociadas. Estas incluyen la gestión de direcciones tanto de usuarios como de empresas (considerando las ubicaciones más cercanas al cliente), la creación, llenado y eliminación de carritos, la generación y actualización del estado de los pedidos, así como la búsqueda y asignación de repartidores. Además, todas las vistas correspondientes a estos procesos se encuentran implementadas y operativas.
* Las funcionalidades de operaciones CRUD en el módulo administrativo funcionan de manera adecuada, permitiendo al administrador gestionar la información de las empresas y usuarios registrados en la aplicación Delivery CCDE.
* Las funcionalidades de generación de reportes en el módulo administrativo operan de manera correcta. Estas permiten visualizar información relevante sobre las empresas registradas, los repartidores activos, los historiales de pedidos, así como los productos más vendidos dentro de la aplicación Delivery CCDE.