**Manual Tecnico**

Este manual técnico ha sido creado con el propósito de facilitar el entendimiento, uso y mantenimiento de la aplicación “**FoodOrder**”. Su objetivo principal es servir como una guía clara, completa y estructurada destinada a desarrolladores, técnicos y personal encargado de la administración del sistema.

La aplicación está diseñada para optimizar el proceso de pedidos en Food Trucks, ofreciendo una plataforma digital que permite gestionar productos, monitorear pagos, generar reportes inteligentes y brindar soporte y ayuda a través de herramientas en inteligencia artificial. Este documento abarca desde la instalación y configuración inicial, hasta el uso avanzado de las funcionalidades del sistema, contemplando tanto el entorno del usuario como el panel de administración.

Además, este manual proporciona las bases necesarias para la detección y solución de problemas comunes, recomendaciones de mantenimiento, y buenas practicas para asegurar el correcto funcionamiento del sistema a largo plazo. De esta forma, se busca garantizar la eficiencia operativa, la escalabilidad del proyecto y una mejor experiencia tanto para los usuarios finales como para los administradores que trabajan con la aplicación.

**FoodOrder**.

Fecha de creación: 10/1/2025

Version: 1.0

Tecnico Principal y Equipo responsable

Ing. Alejandro

Team Vibe Coding

**Introducción**

La aplicación **"FoodOrder"** es un sistema web desarrollado para digitalizar y automatizar el proceso de pedidos, pagos y administración en negocios de food trucks. Su diseño busca mejorar la experiencia del cliente final y facilitar las tareas operativas y administrativas para los encargados de ventas.

El sistema permite a los usuarios acceder mediante un código QR para explorar el menú, agregar productos al carrito, elegir un método de pago (efectivo, tarjeta o transferencia), y recibir atención personalizada a través de un chatbot. A su vez, ofrece a los administradores una plataforma robusta para la gestión de pedidos, análisis de ventas, generación de reportes inteligentes, y control del rendimiento de cada food truck.

Esta herramienta esta desarrollada para:

* Desarrolladores de software
* Técnicos de soporte
* Personal encargado de mantenimiento

La aplicación ha sido desarrollada utilizando tecnologías modernas como:

* **Python** como lenguaje principal de desarrollo
* **Flask** como el framework back-end para construir la aplicación y el api de gestion de pedidos
* **Jinja2** para renderizar las vistas HTML
* **MySQL** como sistema de bases de datos
* **JavaScript** para el manejo de algunas funcionalidades en el front-end
* **HTML y CSS** para la estructura y el diseño de las interfaces y una buena UX
* **Api Stripe** el api para gestionar los pagos de la aplicación

**Requisitos**

La aplicación **“FoodOrder”** está diseñada para ser ejecutada en servidores locales o en la nube con especificaciones mínimas. También puede ejecutarse en un equipo personal para pruebas o desarrollo.

**Software**

Para el correcto funcionamiento de la aplicación, es necesario contar con los siguientes componentes instalados:

* Python 3.9 o superior
* Flask (Framework Web)
* MySQL (Gestor de bases de datos)
* Stripe
* Acceso a internet
* Navegador moderno

Las dependencias las puede encontrar en el archivo de instalación `requirements.txt` incluido en el repositorio del proyecto.

**Red**

La aplicación necesita conectividad de red para funcionar correctamente en entornos de producción especialmente para:

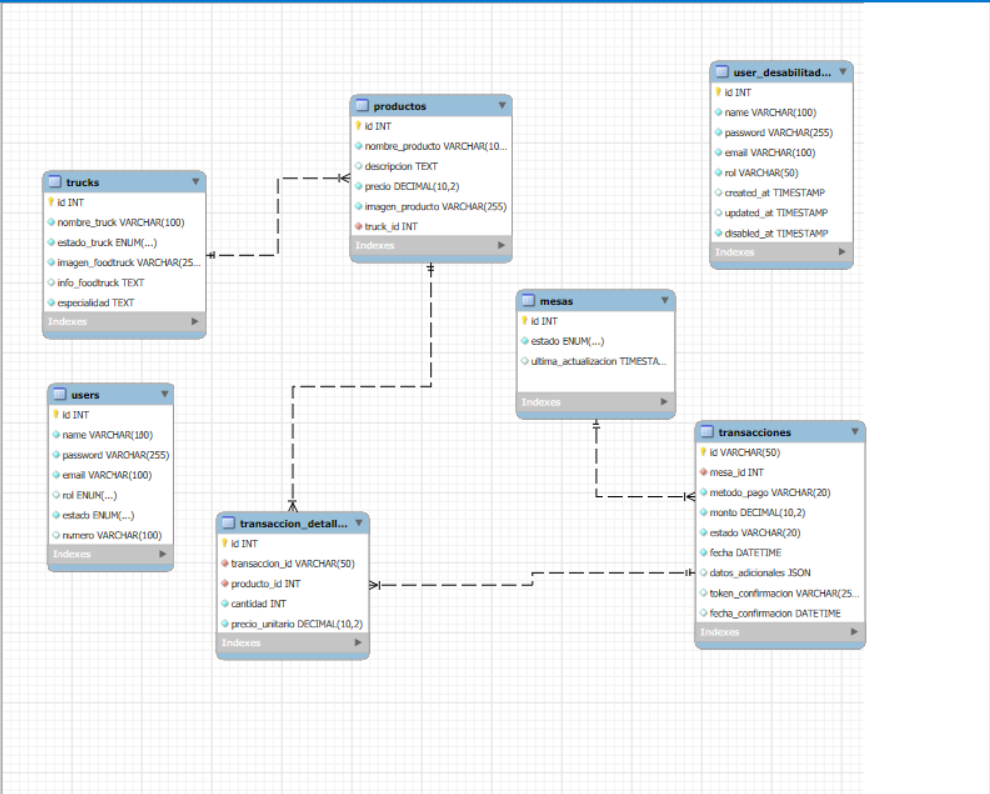
* Procesamiento de pagos en línea (requiere acceso a internet)
* Envió de la factura de compra por Gmail (requiere acceso a internet)
* Configuración de puertos adecuado especialmente los puertos 5000 - puerto Flask de la aplicación y el 139 – puerto SMTP de la aplicación para envíos de correos.

**Instalación y Configuración**

* Pasos para la instalación:
  + Clonar el repositorio: abre una terminal y ejecuta
    - Git clone <https://github.com/Al3jandr0M4p/gestion_pedidos_food_trucks>
    - cd gestion\_pedidos\_food\_trucks
  + Instalar dependencias
    - Asegúrate de tener un entorno virtual (opcional) o no y ejecuta:
      * Python -m venv <nombre\_de\_tu\_entorno>
      * Windows:
        + Venv\Scripts\activate
      * MacOs/Linux:
        + Source venv/bin/actívate
  + Configuración del entorno
    - Crea un archivo `.env` con las siguientes variables:
      * DEBUG=TRUE
      * FLASK\_ENV=development
      * HOST=<host\_de\_la\_db>
      * USER=<usuario\_de\_la\_db>
      * PASSWD=<password\_del\_usuario>
      * DB=<nombre\_de\_la\_db>
      * SECRET\_KEY=<clave\_secreta>
      * PRIVATE\_KEY=<clave\_privada\_de\_la\_api>
      * EMAIL\_USER=[molle0711@gmail.com](mailto:molle0711@gmail.com)
      * EMAIL\_PASSWORD=<password\_de\_app>
      * EMAIL\_DEFAULT\_SENDER=[molle0711@gmail.com](mailto:molle0711@gmail.com)
  + Configuración inicial
    - Crear usuarios admin desde la base de datos o desde una ruta protegida
    - Registrar Food Trucks y productos desde el panel administrador

**Base de Datos**

**Modelo Entidad-relacion (MER)**

****

Dase de datos BackUp Diagrama (MER)



**Tablas Principales**

Tabla users: Esta tabla es donde se guardan los usuarios de la aplicación y su información.

Tabla trucks: En esta tabla esta toda la información de los trucks de la aplicación sus especialidades, y toda información importante para su administración.

Tabla productos: Aquí esta toda la información de los productos relacionados a la tabla de trucks donde se agregan a sus trucks.

Tabla mesas: Aquí esta las mesas que tendrá el usuario en la aplicación.

Tabla transacciones: En esta tabla se guarda la información de los métodos de pago de la aplicación

**Mantenimiento y Actualizaciones**

* Para actualizar las dependencias:
  + pip install –upgrade -r requirements.txt
* Para respaldar la base de datos MySQL:
  + Mysqldump -u usuario -p nombre\_base\_de\_Datos > respaldo.sql
* Para restaurar:
  + Asegúrate de tener creada la db:
    - Créate database nombre\_db
  + Mysql – u usuario -p nombre\_db < respaldo.sql

**API**

Api: Interfaz de programación de aplicaciones

Endpoints: ` /api/pedidos`.

Método: POST

Aquí en este endpoint se envía los datos de los pedidos de los clientes

Para un mayor control de la aplicación y mejora de lo que los clientes más piden y les gusta.

Ejemplo de cómo podría usare con postman:

Usa el método por defecto GET en la barra de URL, escribe la ruta o endpoint completo  
`http://localhost:5000/api/pedidos` - si estas en local

`http://<dominio>:<puerto>/api/pedidos` - si está en un dominio

[

{

"id": 1,

"mesa\_id": 3,

"metodo\_pago": "tarjeta",

"monto": 150.00,

"estado": "confirmado",

"fecha": "2025-05-07 12:35:00",

"fecha\_confirmacion": "2025-05-07 12:40:00",

"detalles": [

{

"id": 1,

"producto\_id": 5,

"nombre\_producto": "Pizza Margarita",

"cantidad": 2,

"precio\_unitario": 50.00

},

]

},

]

**SI NO FUNCIONA**

Asegurate de que tu servidor Flask esta corriendo exitosamente.

Verifica la ruta o endpoint `/api/pedidos` si esta registrada correctamente.

Mira si hay errores en la consola donde corre Flask.

Asegurate de tener datos en las tablas de transacciones, transacciones\_detalles y productos o asegúrate de que el json que mandas desde postman es correcto a la estructura esperada por la api.

**Problemas Comunes y soluciones**

Una lista de errores típicos que podrían pasar y como arreglarlos

* Error de conexión a la base de datos: Verificar las variables de entorno `.env` y revisar que sus valores sean los correctos para conectarte
* No se envían correos: revisar si el puerto SMTP (139) esta bloqueado en tu configuración o si la variable `EMAIL\_PASSWORD` es correcta.
* Stripe no funciona: revisar si la `PRIVATE\_KEY` es la correat o si estas en modo sandbox/live/modo producción y usar las api keys de producción en modo producción y las de developer en developer.

**Conclusión**

La aplicación “**FoodOrder**” representa una solución integral para la digitalización de pedidos en negocios móviles. Su diseño modular, flexible y escalable permite una gestión eficiente tanto para usuarios como para administradores. A través de este manual, se busca asegurar el correcto uso, mantenimiento y evolución continua del sistema.

**Contactos y Soporte**

Correo de soporte: [20230542@ipopsa.edu.do](mailto:20230542@ipopsa.edu.do)

Desarrollador principal: Alejandro Molle Encarnacion – [molle0711@ipopsa.edu.do](mailto:molle0711@ipopsa.edu.do)

Repositorio del proyecto: <https://github.com/Al3jandr0M4p/gestion_pedidos_food_trucks>