

Evidencia de producto: GA10-220501097-AA10-EV01 elabora documentos técnicos y de usuario del Software – *Manual Técnico*

Presentado por:

Hanks Donovan Audivett Valencia

Sebastián Cantillo

Héctor Bohórquez

Ficha: 3000004



**SERVICIO NACIONAL DE APRENDIZAJE SENA
TECNOLOGO EN DESARROLLO DE SOFTWARE
BARRANQUILLA / ATLÁNTICO**

2026



Contenido

Introducción	3
Prerrequisitos de instalación del sistema.....	4
Frameworks y estándares con los que se construyó el sistema	5
Diagrama de casos de uso del sistema.....	6
Modelo entidad relación de la base de datos	7
Diccionario de datos.....	8
Scripts de instalación.....	10
Diagrama de componentes	13
Referencias	14



Introducción

El proyecto ApiFood es una aplicación web de comercio electrónico (e-commerce) desarrollada en Python utilizando el framework Flask. Permite a los usuarios navegar por un catálogo de productos, gestionar un carrito de compras y simulación de pagos. Además, cuenta con un panel de administración para la gestión de usuarios, roles (Administrador/Usuario), inventario de productos.



Prerrequisitos de instalación del sistema

Sistema Operativo: Windows 10/11, Linux o macOS.

Python: Versión 3.8 o superior.

Base de Datos: MySQL Server 5.7+ o MariaDB.

Navegador Web: Chrome, Firefox, Edge, etc.

Cuenta de correo Gmail (para funcionalidad de recuperación de contraseña) con "Contraseña de aplicación" habilitada.

Memoria RAM: 4 GB mínimo (recomendado 8 GB)

Espacio en disco: 500 MB libres

Conexión a internet para descargas

Enlaces del proyecto:

URL del proyecto:

<https://alonso1.pythonanywhere.com/>

Al registrarse su rol será de Usuario.

Para ingresar como Admin puede ingresar con los datos de:

- Correo: matilda@polo.com
- Contraseña: 123

Repositorio del proyecto:

https://github.com/hector-bohorquez/Apifood_Proyecto.git



Frameworks y estándares con los que se construyó el sistema

La aplicación sigue el patrón de diseño Modelo-Vista-Controlador (MVC) adaptado a Flask.

Stack Tecnológico:

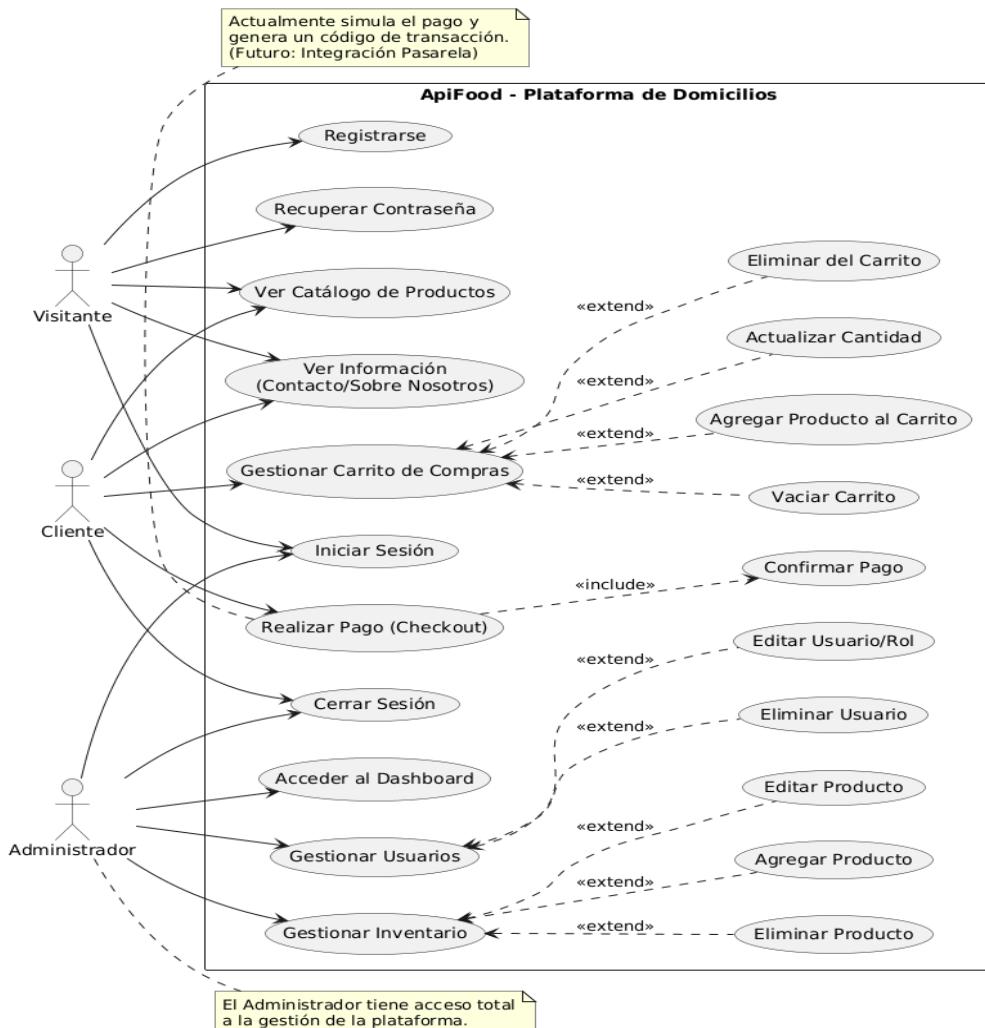
- Backend: Python 3, Flask.
- Base de Datos: MySQL.
- Frontend: HTML5, CSS3, JavaScript (Vanilla).
- Motor de Plantillas: Jinja2.

Librerías Principales:

- Flask: Framework web.
- flask-mysqldb: Conector para MySQL.
- python-dotenv: Gestión de variables de entorno.
- Werkzeug: Utilidades de seguridad (hashing de contraseñas).
- flask-wtf: Protección CSRF para formularios.
- smtplib: Envío de correos electrónicos.

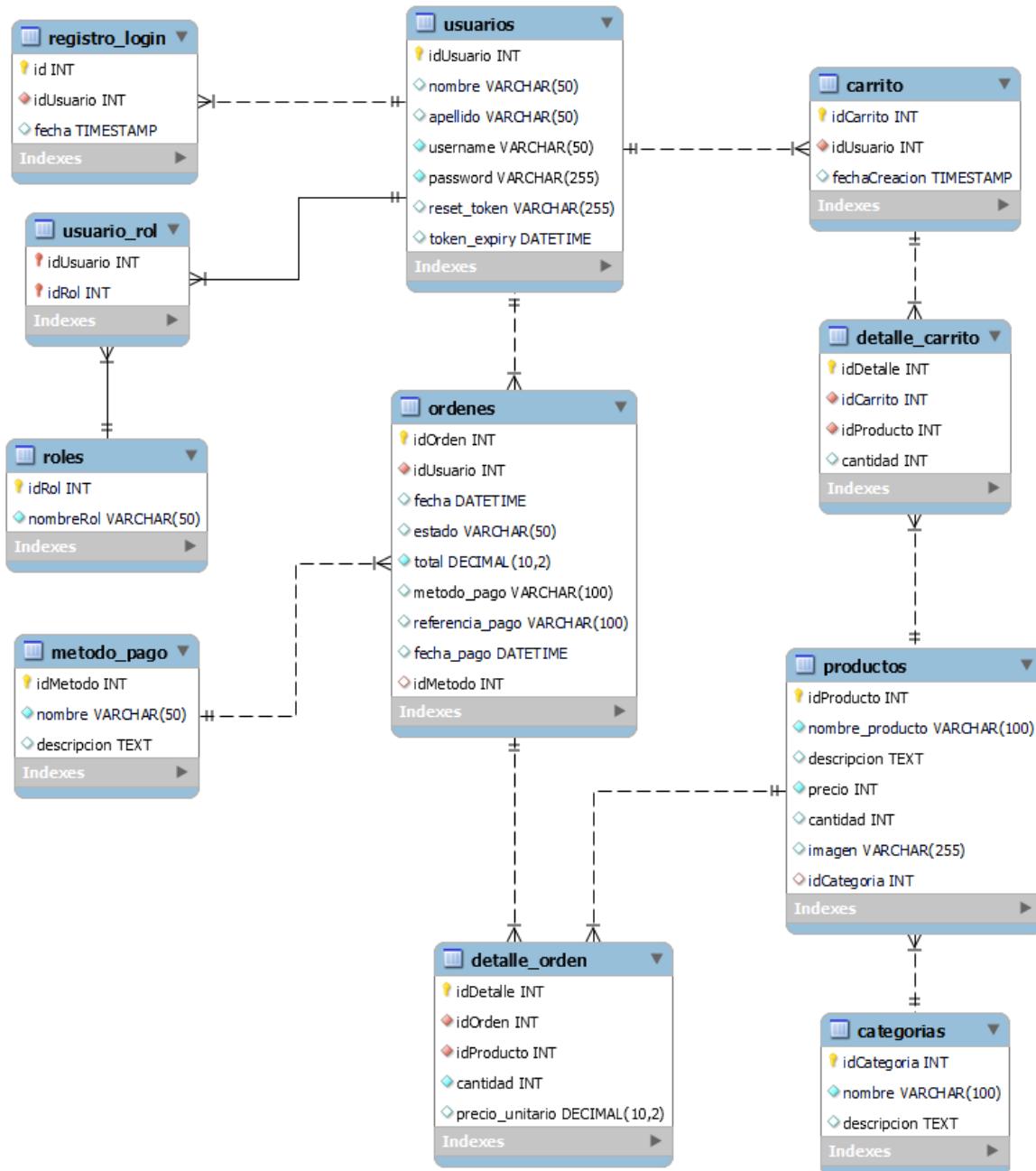


Diagrama de casos de uso del sistema.





Modelo entidad relación de la base de datos





Diccionario de datos

Tabla Usuarios				
Atributos	Tipos de datos	PK	FK	Descripción
idUsuario	INT	X		Identificador único del usuario
nombre	VARCHAR(50)			Nombre del usuario
apellido	VARCHAR(50)			Apellido del usuario
username	VARCHAR(50)			Correo electrónico (único)
password	VARCHAR(255)			Hash de la contraseña
reset_token	VARCHAR(255)			Token para recuperación de contraseña
token_expiry	DATETIME			Fecha de expiración del token

Tabla Productos				
Atributos	Tipos de datos	PK	FK	Descripción
idProducto	INT	X		Identificador del producto
nombre_producto	VARCHAR(100)			Nombre del producto
descripcion	TEXT			Descripción del producto
precio	INT			Precio unitario
cantidad	INT			Stock disponible
imagen	VARCHAR(255)			Ruta de la imagen
idCategoria	INT		X	Categoría del producto

Tabla Ordenes				
Atributos	Tipos de datos	PK	FK	Descripción
idOrden	INT	X		Identificador de la orden
idUsuario	INT		X	Cliente que realizó la orden
fecha	DATETIME			Fecha y hora del pedido
estado	VARCHAR(50)			Estado (Pendiente, En preparación, En camino, Entregado)
total	DECIMAL(10, 2)			Monto total
metodo_pago	VARCHAR(100)			Método de pago
referencia_pago	VARCHAR(100)			Referencia del pago
fecha_pago	DATETIME			Fecha y hora del pago
idMetodo	INT		X	Método de pago seleccionado



Tabla detalle_orden				
Atributos	Tipos de datos	PK	FK	Descripción
idDetalle	INT	X		Identificador del detalle de orden
idOrden	INT		X	Orden asociada
idProducto	INT		X	Producto comprado
cantidad	INT			Cantidad comprada
precio_unitario	DECIMAL(10, 2)			Precio al momento de la compra

Tabla carrito				
Atributos	Tipos de datos	PK	FK	Descripción
idCarrito	INT	X		Identificador del carrito
idUser	INT		X	Usuario dueño del carrito
fechaCreacion	TIMESTAMP			Fecha de creación

Tabla detalle_carrito				
Atributos	Tipos de datos	PK	FK	Descripción
idDetalle	INT	X		Identificador del detalle
idCarrito	INT		X	Carrito asociado
idProducto	INT		X	Producto agregado
cantidad	INT			Cantidad del producto

Tabla registro_login				
Atributos	Tipos de datos	PK	FK	Descripción
id	INT	X		Identificador de auditoría
idUser	INT		X	Usuario que inició sesión
fecha	TIMESTAMP			Fecha del evento

Tabla roles				
Atributos	Tipos de datos	PK	FK	Descripción
idRol	INT	X		Identificador único del rol
nombreRol	VARCHAR(50)			Nombre del rol (Admin, Usuario, Restaurante, Domiciliario)

Tabla usuario_rol				
Atributos	Tipos de datos	PK	FK	Descripción
idUser	INT	X		Relación con la tabla usuarios
idRol	INT			Relación con la tabla roles



Tabla metodo_pago				
Atributos	Tipos de datos	PK	FK	Descripción
idMetodo	INT	X		Identificador del método de pago
nombre	VARCHAR(50)			Nombre (Efectivo, Tarjeta, etc.)
descripcion	TEXT			Descripción del método

Tabla Categoría				
Atributos	Tipos de datos	PK	FK	Descripción
idCategoria	INT	X		Identificador de la categoría
nombre	VARCHAR(100)			Nombre de la categoría
descripcion	TEXT			Descripción de la categoría

Scripts de instalación

Clonar o Descargar el Repositorio

Asegúrese de tener los archivos del proyecto en su directorio de trabajo.

Repositorio del proyecto:

https://github.com/hector-bohorquez/Apifood_Proyecto.git

Configuración del Entorno Virtual

Se recomienda usar un entorno virtual para aislar las dependencias.

- Abrir una terminal o símbolo del sistema en la carpeta del proyecto
- Instalar virtualenv
- Ejecutar:
 pip install virtualenv

Crear entorno virtual:

python -m venv venv

o

virtualenv -p python3 env



Activar entorno (Windows):

venv\Scripts\activate

Activar entorno (Linux/Mac):

source venv/bin/activate

Instalación de dependencias de Python

- Abrir una terminal o símbolo del sistema en la carpeta del proyecto
- Instalar pip si no está instalado:
python -m ensurepip
o
python.exe -m pip install --upgrade pip
- Instalar Flask y otras librerías necesarias:
pip install -r requirements.txt

Configuración (valores por defecto en `app.py`)

- `app.secret_key` se define en `app.py` (cambiar para producción).
- Configuración MySQL por defecto en `app.py`:
 - HOST: `localhost`
 - USER: `root`
 - PASSWORD: `tu_password_mysql`
 - DB: `bd_app`

IMPORTANTE: Cambiar credenciales y secretos antes de publicar. Use variables de entorno en producción.

Configuración de Variables de Entorno

Crear un archivo .env en la raíz del proyecto con el siguiente contenido:

```
SECRET_KEY=tu_clave_secreta_segura
MYSQL_HOST=localhost
MYSQL_USER=root
MYSQL_PASSWORD=tu_password_mysql
MYSQL_DB=bd_app
MAIL_USER=tu_correo@gmail.com
MAIL_PASSWORD=tu_contraseña_de_aplicacion
```



Creación de la base de datos

- Ejecutar el script SQL incluido: scriptDB.sql
- Ejemplo (línea de comandos MySQL):
mysql -u root -p < scriptDB.sql

Pruebas locales y verificación rápida

1. Crear entorno virtual e instalar dependencias.
2. Crear la base de datos con `scriptDB.sql`.
3. Ajustar credenciales en `app.py` o configurar variables de entorno.
4. Ejecutar la app en el bash:
`python app.py`
o
`flask run --port 3000`

La aplicación por defecto expone `host=0.0.0.0` y `port=3000` según `app.py`.

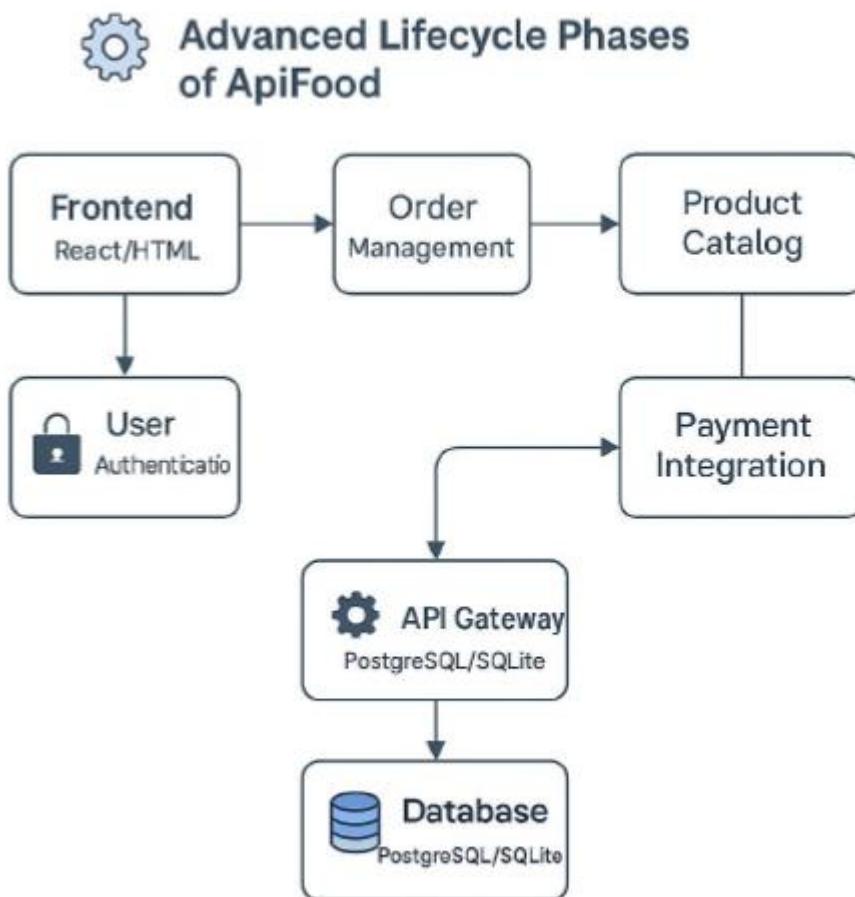
Errores Comunes

1. Error de Conexión MySQL: Verificar que las credenciales en .env sean correctas y el servicio MySQL esté corriendo.
2. Error SMTP (Envío de correos): Asegurarse de que la cuenta de Gmail tenga activada la verificación en dos pasos y se esté usando una "Contraseña de Aplicación", no la contraseña normal.
3. Imágenes no cargan: Verificar que la carpeta static/uploads exista y tenga permisos de escritura.



Diagrama de componentes

Diagrama de Componentes





Referencias

Angela Fonseca. (2025, July 24). *Python Instalacion de python Video 0* [Video]. YouTube.
<https://www.youtube.com/watch?v=Twz9jL89sol>

@josecodetech. (2022, May 1). *13 Publica gratis tu app Flask en la web* [Video]. YouTube.
<https://www.youtube.com/watch?v=TU3UefE8NPo>

homastudio. (2024, January 24). *4 - New database in PythonAnywhere* [Video]. YouTube.
<https://www.youtube.com/watch?v=aM0P5dLDN7w>

Welcome to Flask — Flask Documentation (3.1.x). (s. f.).
<https://flask.palletsprojects.com/en/stable/>

MySQL. (n.d.). <https://www.mysql.com/>

hector-bohorquez/Apifood_Proyecto. (n.d.). GitHub. https://github.com/hector-bohorquez/Apifood_Proyecto

Inicio | APiFood. (n.d.). <https://alonso1.pythonanywhere.com/>