

Manual de Programador

Manual de Programador del Sistema ERP

Índice

1. Introducción
 2. Tecnologías Utilizadas
 3. Estructura del Sistema
 4. Instalación y Configuración
 5. Detalles de los Módulos
 6. Mantenimiento y Actualización
 7. Contacto para Soporte Técnico
-

1. Introducción

El sistema ERP de Pollos Copacabana es una solución modular desarrollada con tecnologías modernas para garantizar escalabilidad y mantenimiento eficiente. Este manual está dirigido a programadores que trabajarán en el sistema, detallando su estructura y componentes principales.

2. Tecnologías Utilizadas

- **Frontend:** React.js
 - **Backend:** Node.js con Express
 - **Base de Datos:** MySQL
 - **Servidor:** Nginx
 - **Despliegue:** AWS (EC2 para servidores, RDS para bases de datos).
-

3. Estructura del Sistema

- **Carpeta Principal:**
 - `/frontend`: Contiene el código del cliente.
 - `/backend`: Contiene la lógica del servidor.
 - `/db-scripts`: Scripts para inicializar la base de datos.
- **Base de Datos:**
 - Tabla `ventas`: Registro de pedidos y facturación.

- Tabla **inventarios**: Control de stock.
 - Tabla **sucursales**: Datos de las sucursales.
 - Tabla **finanzas**: Ingresos y egresos.
 - Tabla **clientes**: Registro de clientes.
-

4. Instalación y Configuración

Clonar el repositorio:

```
git clone <https://github.com/tu-repositorio/erp-polloscopacabana.git>
cd erp-polloscopacabana
```

- 1.
2. **Configurar el backend:**

Entra a la carpeta **/backend** y crea un archivo **.env** con las variables de entorno:

```
DB_HOST=localhost
DB_USER=root
DB_PASS=password
DB_NAME=polloscopacabana
```

-
3. **Configurar el frontend:**

- Entra a la carpeta **/frontend** y edita el archivo **config.js** con la URL del backend.

4. **Iniciar el sistema:**

Backend:

```
cd backend
npm install
npm run dev
```

Frontend:

```
cd frontend
npm install
npm start
```

○

5. Detalles de los Módulos

Cada módulo está documentado con sus principales endpoints:

Gestión de Ventas

- Endpoint: **POST** `/api/ventas`
- Función: Crear un pedido.

Ejemplo de Payload:

```
{
  "cliente": "Juan Perez",
  "productos": [
    { "id": 1, "cantidad": 2 },
    { "id": 2, "cantidad": 1 }
  ]
}
```

-

Gestión de Inventarios

- Endpoint: **GET** `/api/inventarios`
- Función: Obtener productos con stock crítico.

6. Mantenimiento y Actualización

- **Actualización del sistema:** Ejecutar **git pull** en el servidor para traer los cambios más recientes.

Base de datos: Realizar backups semanales con el comando:

```
mysqldump -u root -p polloscopacabana > backup.sql
```