**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS.**

(Universidad del Perú**,** DECANA DE AMÉRICA).

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Curso:** GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

**Diagrama de Base de Datos (No relacional)**

**Grupo 1**

Integrantes:

* Cano Vasquez, Juan José
* Huaman Uriarte, Cesar Alberto
* Mallco Laurante, Frans Andre
* Munayco Sirlopu, Ronald
* Rojas Rojas, Darwin Jostein
* Sandoval Falcon, David Brian

**Empresa: FISIDEV\_SOLUTIONS**

**2025-1**

**Indice**

Contenido

[**1.** **Introducción** 3](#_Toc198490941)

[**1.1Tecnologías utilizadas:** 3](#_Toc198490942)

[**2.** **Modelo de Datos** 3](#_Toc198490943)

[**3.** **Estructura de las colecciones** 3](#_Toc198490944)

[**4.** **Relaciones (Diagrama ER para MongoDB)** 5](#_Toc198490945)

[**5.** **API REST (resumen del backend)** 6](#_Toc198490946)

[**6.** **Ejemplo de flujo** 7](#_Toc198490947)

# **Introducción**

Este proyecto consiste en una plataforma digital para la cevichería "Mar Adentro", diseñada específicamente para registrar clientes, gestionar sus pedidos de ceviches y platos marinos, y monitorear el estado de cada pedido desde su creación hasta la entrega.

## **1.1Tecnologías utilizadas:**

 **Backend**: Node.js + Express - Garantiza un servicio rápido y eficiente para procesar pedidos

 **Base de datos**: MongoDB - Ofrece flexibilidad para adaptarse a futuras necesidades del negocio.

El sistema permite a los clientes realizar pedidos online, mientras que el personal de la cevichería puede gestionarlos en tiempo real, asegurando que cada plato llegue fresco y en el tiempo prometido a los comensales.

# **Modelo de Datos**

En nuestro sistema, utilizamos un enfoque basado en MongoDB, por lo que trabajamos con **colecciones** y **documentos** en lugar de tablas relacionales. Esto proporciona mayor flexibilidad y velocidad para gestionar los pedidos de la cevichería.

2.2 **Entidades principales:**

* **Clientes**: Almacena información de contacto de quienes realizan pedidos
* **Pedidos**: Registra los platos solicitados, precio, y estado actual del pedido
* **Productos**: Catálogo de ceviches y platos marinos disponibles
* **Estados de pedido**: Seguimiento del ciclo de vida del pedido

# **Estructura de las colecciones**

* **Cliente (clients):**

Texto

El contenido generado por IA puede ser incorrecto.

* **Producto (productos)**

Texto

El contenido generado por IA puede ser incorrecto.

* **Pedido (pedidos)**

Texto

El contenido generado por IA puede ser incorrecto.

**Estados disponibles para pedidos**

Los estados posibles para un pedido en nuestra cevichería son:

* recibido: Pedido ingresado al sistema
* confirmado: Pedido confirmado por la cevichería
* preparando: Pedido en la cocina, preparándose
* en\_camino: Pedido en ruta de entrega

# **Relaciones (Diagrama ER para MongoDB)**

Diagrama

El contenido generado por IA puede ser incorrecto.

Este diagrama muestra el modelo entidad-relación de un sistema de pedidos. Hay tres entidades principales: **Clientes**, **Pedidos** y **Productos**. Un cliente puede realizar múltiples pedidos (relación 1:N). Cada pedido puede incluir múltiples productos, y un producto puede estar en varios pedidos (relación N:M). Las entidades tienen atributos como ID\_Cliente, ID\_Pedido e ID\_Producto que los identifican. Los pedidos contienen información como total, estado y método de pago. Los productos incluyen detalles como nombre, precio y disponibilidad. Este modelo facilita la gestión de pedidos, clientes y productos en un sistema de ventas.

# **API REST (resumen del backend)**

Nuestro backend implementado en Node.js y Express expone los siguientes endpoints:

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Este cuadro muestra las rutas de una API REST para un sistema de pedidos. Usa métodos HTTP como **GET**, **POST**, **PUT** para acceder y modificar datos. Permite gestionar **clientes**, **productos**, **pedidos** y **estadísticas**. Las rutas /api/clientes permiten listar, registrar y actualizar clientes. Las de /api/productos manejan el catálogo de productos disponibles. Las de /api/pedidos permiten crear, consultar y actualizar pedidos. Además, /api/estadisticas entrega datos analíticos del sistema. Esta API organiza claramente las operaciones CRUD y mejora la gestión del sistema.

# **Ejemplo de flujo**

**Caso de uso: Pedido de ceviche para entrega a domicilio**

1. **Registro/Inicio de sesión de cliente:** Luis Rivera inicia sesión en el sistema de la cevichería "Mar Adentro".
2. **Selección de productos:** Luis añade a su carrito:
   * 2 Ceviches mixtos (S/. 35.00 c/u)
   * 1 Chicharrón de pescado (S/. 30.00)
3. **Confirmación del pedido:**
   * Luis confirma su dirección de entrega y agrega referencias para facilitar la llegada del delivery
   * Selecciona "efectivo" como método de pago
   * Añade instrucciones especiales: "Sin ají por favor"
   * El sistema registra el pedido con estado inicial "recibido"
4. **Procesamiento interno:**
   * El personal de la cevichería recibe notificación del nuevo pedido
   * Confirman disponibilidad de ingredientes y tiempo de entrega (pedido pasa a "confirmado")
   * La cocina comienza la preparación (estado "preparando")
   * Una vez empacado, se asigna al repartidor (estado "en\_camino")
5. **Entrega y cierre:**
   * El repartidor entrega el pedido a Luis y cobra el monto
   * El pedido se marca como "entregado" en el sistema
   * Luis recibe una notificación para valorar su experiencia

El sistema mantiene un registro completo del historial de estados, con marcas de tiempo precisas para cada cambio.