**Documentación del Proyecto: Sistema Inteligente para Óptica con PostgreSQL e IA**

**📌 Nombre del Proyecto:**

Sistema Inteligente de Gestión para Óptica con IA y PostgreSQL

**👨‍💻 Desarrollador:**

Francisco Leonardo Martínez Nicolás

**🧠 Objetivo General:**

Desarrollar una aplicación web integral para una óptica, que permita gestionar pacientes, historiales clínicos, productos y ventas, incorporando inteligencia artificial para facilitar diagnósticos optométricos y sugerencias de armazones basadas en reconocimiento facial.

**1. Justificación del Proyecto**

El avance tecnológico ha permitido que herramientas como la inteligencia artificial y las bases de datos relacionales sean accesibles para pequeñas y medianas empresas. En este proyecto se pretende integrar estas tecnologías para mejorar la experiencia del cliente y facilitar la toma de decisiones clínicas en una óptica. Este sistema servirá tanto como solución real como proyecto de aprendizaje profesional.

**2. Tecnologías a Utilizar**

* **Base de Datos:** PostgreSQL
* **Backend:** Node.js con Express.js (Arquitectura Hexagonal + Vertical Slicing)
* **Frontend:** React.js con TailwindCSS
* **Reconocimiento Facial:** OpenCV + Mediapipe o librerías en Python (Flask API conectada al sistema)
* **IA Clínica:** OpenAI API para sugerencias de diagnóstico
* **Autenticación:** JWT + Roles personalizados
* **Contenedores:** Docker + Docker Compose (en este proceso recordaremos que no se utilizar Docker así que ten mucho cuidado para que me tengas que guiar paso a paso para que no halla ningún tipo de problema)
* **Documentación API:** Swagger
* **Control de Versiones:** Git + GitHub

**3. Arquitectura del Proyecto**

Se trabajará bajo los principios de:

* Arquitectura **Hexagonal** para separar la lógica de negocio del resto del sistema.
* **Vertical Slicing** para dividir el sistema por funcionalidades completas.
* **Microservicios** en los módulos IA si es necesario para escalar.

**4. Módulos del Sistema**

**4.1 Módulo de Pacientes**

* Registro y actualización de pacientes
* Consulta de historial clínico
* Búsqueda avanzada

**4.2 Módulo de Historial Clínico**

* Registro de síntomas, diagnósticos, tratamientos
* Generación de reportes
* Sugerencia automática de diagnósticos (IA)

**4.3 Módulo de Productos y Ventas**

* Gestión de armazones, lentes y accesorios
* Registro de ventas
* Sugerencia de armazón con IA por reconocimiento facial

**4.4 Módulo de Usuarios**

* Autenticación
* Gestión de permisos

**5. Funcionalidades con Inteligencia Artificial**

**a) Diagnóstico Optométrico Asistido**

* El sistema enviará los síntomas del paciente a un microservicio de IA (GPT o modelo entrenado) para sugerir posibles diagnósticos o tratamientos sugeridos.

**b) Sugerencia de Armazones con Reconocimiento Facial**

* El sistema encenderá la camra y revisara las facciones del rostro del cliente y en esa misma cámara ya que haya analizado el rostro, le modelara un armazón en la cara del paciente así viendo y provandoce todos los tipos de lentes que le quedaran dependiendo o haciendo una serie de lentes que puede probarce según sea el caso.
* A través de OpenCV/Mediapipe detectará forma de rostro
* Con base en un catálogo de estilos, recomendará el armazón ideal o recomendara todos los que se acoplan a su rostro del cliente.

**6. Flujo de Trabajo Propuesto**

**Fase 1: Planeación y Requerimientos**

* Definición de objetivos
* Selección de tecnologías
* Diagramas de casos de uso y entidad-relación

**Fase 2: Diseño**

* Diseño de base de datos en PostgreSQL
* Arquitectura de carpetas y rutas
* Diagramas de arquitectura y flujos

**Fase 3: Desarrollo Backend**

* Configuración de Express y conexión a PostgreSQL
* CRUD de pacientes e historial clínico
* API de diagnóstico (GPT)

**Fase 4: Desarrollo Frontend**

* Diseño de UI con TailwindCSS
* Componentes funcionales en React.js
* Integración con backend

**Fase 5: Implementación de IA**

* Integración de reconocimiento facial
* Lógica de recomendación de armazones

**Fase 6: Pruebas y Documentación**

* Pruebas unitarias y de integración
* Documentación Swagger
* Manual de usuario

**7. Consideraciones Éticas**

* Protección de datos personales (Ley de protección de datos)
* Consentimiento del paciente para usar IA
* No almacenar imágenes faciales sin cifrado o consentimiento

**8. Metodología de Trabajo**

* **Metodología:** SCRUM (sprints semanales)
* **Gestión de tareas:** Trello o Notion
* **Revisión de avances:** GitHub + Issues + Pull Requests

**9. Futuras Mejoras**

* Entrenamiento de un modelo propio de diagnóstico
* IA para predicción de stock
* Recomendaciones de tratamiento según historial acumulado

**10. Conclusión**

Este proyecto busca ser un ejemplo completo de cómo una aplicación moderna puede combinar múltiples tecnologías para resolver problemas reales del sector salud, al mismo tiempo que sirve como una plataforma de aprendizaje para el desarrollo profesional del autor.

# Sistema Inteligente de Gestión para Óptica con IA y PostgreSQL

## 1. Introducción

Proyecto para la gestión completa de una óptica, integrando IA (diagnóstico asistido y sugerencias de armazones), PostgreSQL y una arquitectura moderna.

\*\*Desarrollador:\*\* Francisco Leonardo Martínez Nicolás

🧠 Objetivo General:

Desarrollar una aplicación web integral para una óptica, que permita gestionar pacientes, historiales clínicos, productos y ventas, incorporando inteligencia artificial para facilitar diagnósticos optométricos y sugerencias de armazones basadas en reconocimiento facial.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Justificación del Proyecto

El avance tecnológico ha permitido que herramientas como la inteligencia artificial y las bases de datos relacionales sean accesibles para pequeñas y medianas empresas. En este proyecto se pretende integrar estas tecnologías para mejorar la experiencia del cliente y facilitar la toma de decisiones clínicas en una óptica. Este sistema servirá tanto como solución real como proyecto de aprendizaje profesional.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Tecnologías a Utilizar

• Base de Datos: PostgreSQL

• Backend: Node.js con Express.js (Arquitectura Hexagonal + Vertical Slicing)

• Frontend: React.js con TailwindCSS

• Reconocimiento Facial: OpenCV + Mediapipe o librerías en Python (Flask API conectada al sistema)

• IA Clínica: OpenAI API para sugerencias de diagnóstico

• Autenticación: JWT + Roles personalizados

• Contenedores: Docker + Docker Compose (en este proceso recordaremos que no se utilizar Docker así que ten mucho cuidado para que me tengas que guiar paso a paso para que no halla ningún tipo de problema)

• Documentación API: Swagger

• Control de Versiones: Git + GitHub

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Arquitectura del Proyecto

Se trabajará bajo los principios de:

• Arquitectura Hexagonal para separar la lógica de negocio del resto del sistema.

• Vertical Slicing para dividir el sistema por funcionalidades completas.

• Microservicios en los módulos IA si es necesario para escalar.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Módulos del Sistema

4.1 Módulo de Pacientes

• Registro y actualización de pacientes

• Consulta de historial clínico

• Búsqueda avanzada

4.2 Módulo de Historial Clínico

• Registro de síntomas, diagnósticos, tratamientos

• Generación de reportes

• Sugerencia automática de diagnósticos (IA)

4.3 Módulo de Productos y Ventas

• Gestión de armazones, lentes y accesorios

• Registro de ventas

• Sugerencia de armazón con IA por reconocimiento facial

4.4 Módulo de Usuarios

• Autenticación

• Gestión de permisos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Funcionalidades con Inteligencia Artificial

a) Diagnóstico Optométrico Asistido

• El sistema enviará los síntomas del paciente a un microservicio de IA (GPT o modelo entrenado) para sugerir posibles diagnósticos o tratamientos sugeridos.

b) Sugerencia de Armazones con Reconocimiento Facial

• El sistema encenderá la camra y revisara las facciones del rostro del cliente y en esa misma cámara ya que haya analizado el rostro, le modelara un armazón en la cara del paciente así viendo y provandoce todos los tipos de lentes que le quedaran dependiendo o haciendo una serie de lentes que puede probarce según sea el caso.

• A través de OpenCV/Mediapipe detectará forma de rostro

• Con base en un catálogo de estilos, recomendará el armazón ideal o recomendara todos los que se acoplan a su rostro del cliente.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Flujo de Trabajo Propuesto

Fase 1: Planeación y Requerimientos

• Definición de objetivos

• Selección de tecnologías

• Diagramas de casos de uso y entidad-relación

Fase 2: Diseño

• Diseño de base de datos en PostgreSQL

• Arquitectura de carpetas y rutas

• Diagramas de arquitectura y flujos

Fase 3: Desarrollo Backend

• Configuración de Express y conexión a PostgreSQL

• CRUD de pacientes e historial clínico

• API de diagnóstico (GPT)

Fase 4: Desarrollo Frontend

• Diseño de UI con TailwindCSS

• Componentes funcionales en React.js

• Integración con backend

Fase 5: Implementación de IA

• Integración de reconocimiento facial

• Lógica de recomendación de armazones

Fase 6: Pruebas y Documentación

• Pruebas unitarias y de integración

• Documentación Swagger

• Manual de usuario

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Consideraciones Éticas

• Protección de datos personales (Ley de protección de datos)

• Consentimiento del paciente para usar IA

• No almacenar imágenes faciales sin cifrado o consentimiento

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Metodología de Trabajo

• Metodología: SCRUM (sprints semanales)

• Gestión de tareas: Trello o Notion

• Revisión de avances: GitHub + Issues + Pull Requests

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9. Futuras Mejoras

• Entrenamiento de un modelo propio de diagnóstico

• IA para predicción de stock

• Recomendaciones de tratamiento según historial acumulado

---

## 2. Arquitectura y Tecnologías

- \*\*Backend:\*\* Node.js + Express.js (Hexagonal + Vertical Slicing)

- \*\*Base de Datos:\*\* PostgreSQL

- \*\*Frontend:\*\* React.js + TailwindCSS

- \*\*Reconocimiento Facial:\*\* Python (OpenCV + Mediapipe, Flask API)

- \*\*IA Clínica:\*\* OpenAI API (GPT)

- \*\*Autenticación:\*\* JWT + Roles

- \*\*Documentación:\*\* Swagger

- \*\*Control de Versiones:\*\* Git + GitHub

---

## 3. Modelado de Base de Datos

### 3.1 Entidades Principales

- \*\*Paciente:\*\* Datos personales y contacto.

- \*\*Examen Clínico:\*\* Cada consulta/examen con toda la información clínica.

- \*\*Historial Clínico:\*\* Listado de exámenes por paciente.

- \*\*Producto (Armazón/Lente):\*\* Gestión de productos.

- \*\*Venta:\*\* Registro de ventas.

- \*\*Usuario:\*\* Personal y permisos.

### 3.2 Ejemplo de Tablas (PostgreSQL)

```sql

*-- Pacientes*

CREATE TABLE pacientes (

    id SERIAL PRIMARY KEY,

    nombre VARCHAR(100),

    genero VARCHAR(10),

    edad INT,

    estado\_civil VARCHAR(20),

    escolaridad VARCHAR(50),

    ocupacion VARCHAR(50),

    domicilio VARCHAR(150),

    email VARCHAR(100),

    telefono VARCHAR(20),

    tutor VARCHAR(100)

);

*-- Exámenes Clínicos*

CREATE TABLE examenes (

    id SERIAL PRIMARY KEY,

    paciente\_id INT REFERENCES pacientes(id),

    fecha DATE,

    prediagnostico TEXT,

    diagnostico TEXT,

    plan\_tratamiento TEXT,

    pronostico TEXT,

    proximacita DATE

);

*-- Antecedentes*

CREATE TABLE antecedentes (

    examen\_id INT REFERENCES examenes(id),

    familiares TEXT,

    no\_patologicos TEXT,

    patologicos TEXT,

    padecimiento\_actual TEXT

);

*-- Agudeza Visual, Lensometría, Alineación Ocular, Motilidad, Exploración Física, Refracción, Binocularidad, etc.*

*-- (Ver estructura propuesta en el mensaje anterior)*

1. Análisis de los Módulos/Secciones del Examen

Las imágenes muestran las siguientes secciones clave del proceso optométrico:

Datos personales

Interrogatorio (Antecedentes, síntomas, padecimiento actual, prediagnóstico)

Agudeza visual (lejos y cerca, con/sin corrección)

Lensometría

Alineación ocular

Motilidad ocular

Exploración física y segmento anterior

Refracción (queratometría, estado refractivo, etc.)

Binocularidad y forias

Vergencias

Integración visión binocular, estereoscópica, método gráfico

Grid de Amsler, tonometría, paquimetría

Campimetría

Biomicroscopía

Oftalmoscopia

Diagnóstico y plan de tratamiento.

Recta final (receta óptica)

2. Propuesta de Modelos de Base de Datos (PostgreSQL)

A continuación te propongo una estructura simplificada en formato de tablas, que puedes adaptar a tu proyecto:

SQL

*-- Tabla principal de pacientes*

CREATE TABLE pacientes (

    id SERIAL PRIMARY KEY,

    nombre VARCHAR(100),

    genero VARCHAR(10),

    edad INT,

    estado\_civil VARCHAR(20),

    escolaridad VARCHAR(50),

    ocupacion VARCHAR(50),

    domicilio VARCHAR(150),

    email VARCHAR(100),

    telefono VARCHAR(20),

    tutor VARCHAR(100)

);

*-- Tabla de consultas/exámenes*

CREATE TABLE examenes (

    id SERIAL PRIMARY KEY,

    paciente\_id INT REFERENCES pacientes(id),

    fecha DATE,

    prediagnostico TEXT,

    diagnostico TEXT,

    plan\_tratamiento TEXT,

    pronostico TEXT,

    proximacita DATE

);

*-- Tabla de antecedentes/interrogatorio*

CREATE TABLE antecedentes (

    examen\_id INT REFERENCES examenes(id),

    familiares TEXT,

    no\_patologicos TEXT,

    patologicos TEXT,

    padecimiento\_actual TEXT

);

*-- Tabla de agudeza visual*

CREATE TABLE agudeza\_visual (

    examen\_id INT REFERENCES examenes(id),

    sin\_rx\_lejos\_od VARCHAR(10),

    sin\_rx\_lejos\_oi VARCHAR(10),

    sin\_rx\_cerca\_od VARCHAR(10),

    sin\_rx\_cerca\_oi VARCHAR(10),

    av\_con\_rx\_od VARCHAR(10),

    av\_con\_rx\_oi VARCHAR(10)

);

*-- Tabla de lensometría*

CREATE TABLE lensometria (

    examen\_id INT REFERENCES examenes(id),

    esfera\_od FLOAT,

    cilindro\_od FLOAT,

    eje\_od INT,

    esfera\_oi FLOAT,

    cilindro\_oi FLOAT,

    eje\_oi INT,

    tipo\_bifocal VARCHAR(20),

    material VARCHAR(50)

);

*-- Tabla de alineación ocular*

CREATE TABLE alineacion\_ocular (

    examen\_id INT REFERENCES examenes(id),

    lejos VARCHAR(50),

    cerca VARCHAR(50)

);

*-- Tabla de motilidad ocular*

CREATE TABLE motilidad\_ocular (

    examen\_id INT REFERENCES examenes(id),

    versiones TEXT,

    ducciones TEXT,

    sacadicos TEXT,

    persecucion TEXT

);

*-- Tabla de exploración física y segmento anterior*

CREATE TABLE exploracion\_fisica (

    examen\_id INT REFERENCES examenes(id),

    anexos TEXT,

    segmento\_anterior TEXT,

    via\_pupilar TEXT,

    diametro\_pupilar\_od FLOAT,

    diametro\_pupilar\_oi FLOAT

);

*-- Tabla de refracción*

CREATE TABLE refraccion (

    examen\_id INT REFERENCES examenes(id),

    esfera\_od FLOAT,

    cilindro\_od FLOAT,

    eje\_od INT,

    esfera\_oi FLOAT,

    cilindro\_oi FLOAT,

    eje\_oi INT,

    queratometria\_od TEXT,

    queratometria\_oi TEXT

);

*-- Tabla de binocularidad y forias*

CREATE TABLE binocularidad (

    examen\_id INT REFERENCES examenes(id),

    forias\_lejos TEXT,

    forias\_cerca TEXT,

    vergencias TEXT

);

*-- Productos*

CREATE TABLE productos (

    id SERIAL PRIMARY KEY,

    nombre VARCHAR(100),

    tipo VARCHAR(50), *-- armazón, lente, accesorio*

    descripcion TEXT,

    precio DECIMAL,

    stock INT

);

*-- Ventas*

CREATE TABLE ventas (

    id SERIAL PRIMARY KEY,

    paciente\_id INT REFERENCES pacientes(id),

    producto\_id INT REFERENCES productos(id),

    fecha DATE,

    cantidad INT,

    total DECIMAL

);

```

---

## 4. Módulos del Sistema y Flujo de Trabajo

### 4.1 Módulo de Pacientes

- Registro, actualización y consulta avanzada.

### 4.2 Módulo de Historial Clínico

- Registro de síntomas, diagnósticos, tratamientos.

- Generación de reportes y receta final.

- Sugerencias automáticas con IA.

### 4.3 Módulo de Productos y Ventas

- Catálogo de productos, gestión de stock.

- Registro de ventas, integración con historial.

### 4.4 Módulo de Usuarios

- Autenticación JWT, gestión de roles.

---

## 5. Flujo Clínico Optométrico (Basado en Imágenes 1, 2, 3)

1. \*\*Registro de Datos Personales\*\*

2. \*\*Interrogatorio Clínico (Antecedentes, síntomas)\*\*

3. \*\*Pruebas Visuales y Clínicas:\*\*

   - Agudeza visual (lejos y cerca, con/sin corrección)

   - Lensometría

   - Alineación y motilidad ocular

   - Exploración física y segmento anterior

   - Refracción (queratometría, estado refractivo, subjetivo)

   - Binocularidad y forias

   - Vergencias, integración binocular, visión estereoscópica

   - Campimetría, biomicroscopía, oftalmoscopía, tonometría

4. \*\*Diagnóstico y Plan de Tratamiento\*\*

5. \*\*Receta Final\*\*

6. \*\*Registro de Venta y Sugerencia de Armazón (IA)\*\*

---

## 6. Endpoints/API Sugeridos

- `/api/pacientes` (CRUD)

- `/api/examenes` (CRUD, vinculado a pacientes)

- `/api/productos` (CRUD)

- `/api/ventas` (registro de venta)

- `/api/diagnostico-ia` (enviar síntomas, recibir sugerencia GPT)

- `/api/armazon-ia` (reconocimiento facial y sugerencia)

---

## 7. Implementación de IA

- \*\*Diagnóstico Optométrico Asistido:\*\*

  Microservicio que recibe síntomas y devuelve sugerencia de diagnóstico o tratamiento (OpenAI API).

- \*\*Sugerencia de Armazones:\*\*

  Microservicio que recibe imagen del rostro, detecta forma y recomienda armazones según catálogo.

---

## 8. Seguridad y Ética

- Protección de datos personales.

- Consentimiento explícito para uso de IA y almacenamiento de imágenes.

- Cifrado de datos sensibles.

---

## 9. Metodología de Trabajo

- SCRUM (sprints semanales)

- Gestión en Trello/Notion

- Control de versiones y revisión con GitHub

---

## 10. Futuras Mejoras

- Entrenamiento de modelo propio de diagnóstico.

- Predicción automática de stock.

- IA para recomendaciones de tratamiento.

---

## 11. Conclusión

Proyecto completo, moderno, escalable y ético para la gestión inteligente de una óptica, integrando IA y mejores prácticas de desarrollo profesional.

---

## 12. Referencias Visuales

- Flujo clínico y campos modelados basados en exámenes reales:

  - ![image1](image1)

  - ![image2](image2)

  - ![image3](image3)

---

## 13. Contacto

Para dudas, sugerencias o colaboración:

\*\*Francisco Leonardo Martínez Nicolás\*\*