2 DE JULIO DE 2024

Universidad Continental

APLICACIÓN WEB ODONTOGRAMA

MANUAL TÉCNICO V1.0

TALLER DE PROYECTOS 2 – INGENIERIA DE SISTEMAS E INFORMÁTICA

ODONTOCONTI
Av. San Carlos 1980 Urb. San Antonio - Huancayo

Contenido

Manual Técnico del Odontograma

I. INTRODUCCIÓN

El Manual Técnico es una guía integral diseñada para proporcionar a desarrolladores, ingenieros y otros profesionales involucrados en el proyecto una referencia detallada sobre la arquitectura, configuración, implementación y mantenimiento del sistema. Este documento es esencial para comprender tanto la estructura técnica del proyecto como los procesos clave necesarios para su desarrollo efectivo y su operación continua.

En este manual, se abordarán todos los aspectos técnicos relevantes del proyecto, desde la configuración inicial del entorno de desarrollo hasta las estrategias avanzadas de despliegue y optimización. Además, se incluirán detalles sobre las tecnologías utilizadas, requisitos de hardware y software, y mejores prácticas recomendadas para garantizar un rendimiento óptimo y una gestión eficiente del sistema.

El objetivo principal de este manual es servir como una herramienta de referencia práctica y completa, facilitando la colaboración entre equipos, la resolución de problemas y el mantenimiento a largo plazo del proyecto. A lo largo del documento, se proporcionarán instrucciones claras y ejemplos concretos para guiar a los usuarios a través de cada etapa del ciclo de vida del desarrollo de software, desde la planificación inicial hasta las actualizaciones y mejoras continuas.

Al seguir las directrices y recomendaciones presentadas en este manual, se espera que los equipos puedan maximizar la eficiencia operativa y asegurar la calidad del producto final, cumpliendo así con los objetivos establecidos y las expectativas del cliente.

II. REQUERIMIENTOS TECNICOS

Para asegurar un entorno de desarrollo y ejecución óptimo para este proyecto en un sistema operativo Windows 10 o 11, se deben cumplir los siguientes requerimientos técnicos:

Hardware Recomendado

- **Procesador:** Intel Core i5 o equivalente, o superior.
- **Memoria RAM:** Mínimo 8 GB de RAM; se recomienda 16 GB o más para un rendimiento óptimo.
- **Almacenamiento:** Disco duro SSD o unidad de estado sólido para una mayor velocidad de acceso a datos.

Sistema Operativo

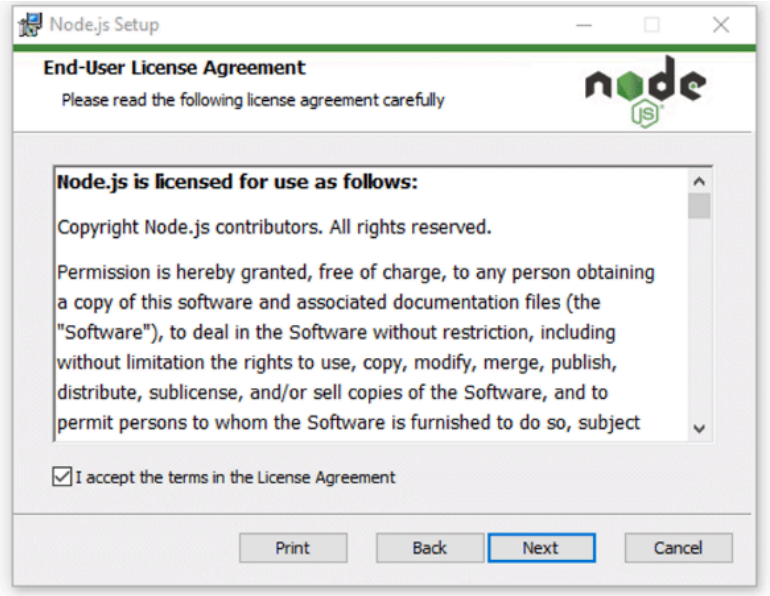
- **Windows 10 o Windows 11:** Sistema operativo actualizado con las últimas actualizaciones de seguridad y funciones.

III. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

Configura tu entorno de desarrollo con los siguientes pasos:

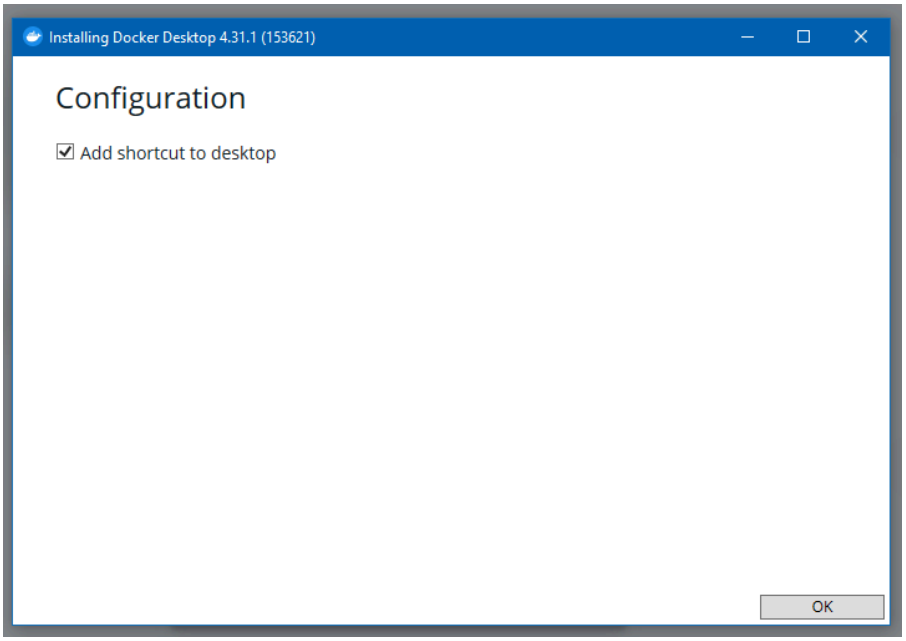
Instalación de Node.js y npm:

- Descarga e instala Node.js desde nodejs.org.
- Verifica la instalación con node -v y npm -v.



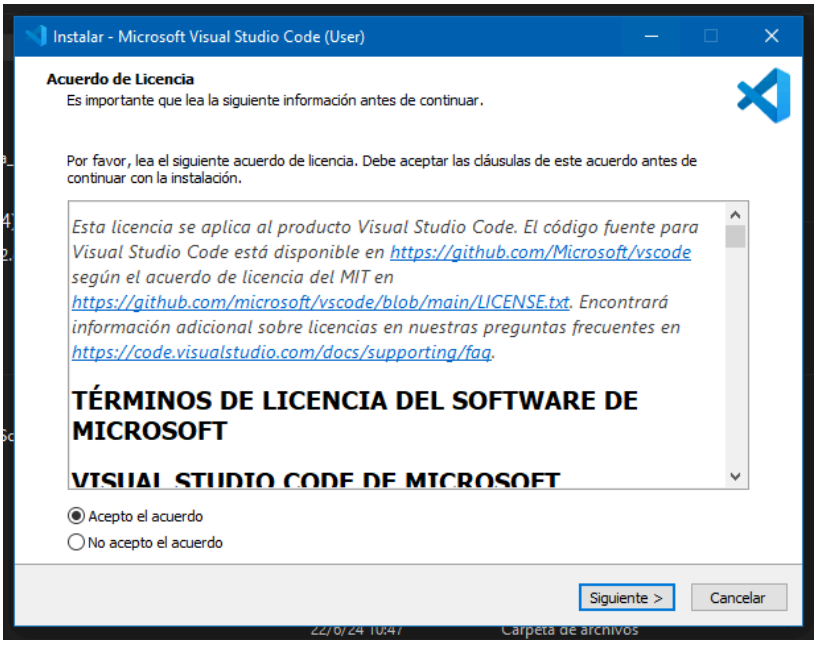
Instalación de Docker:

- Descarga e instala Docker Desktop desde docker.
- Verifica la instalación con docker --version



instalacion de Visual Studio Code:

- Descarga e instala VSCode desde code.visualstudio.com.

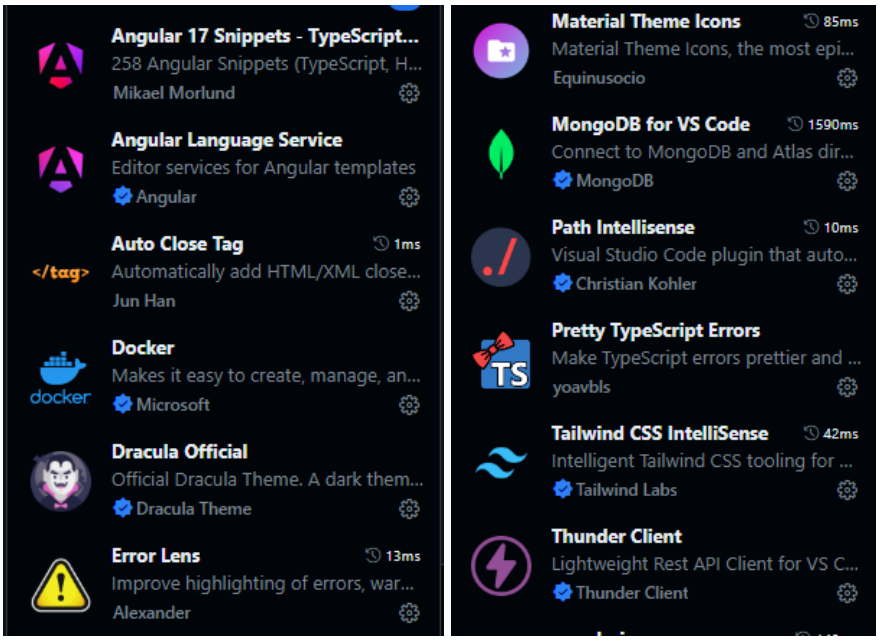


instalación de Git:

- Descarga e instala Git desde git-csm.com.
- Verifica la instalación de git con git --version

Extensiones recomendadas de visual studio code para el proyecto 👍 :

Estas extensiones ayudarán a potenciar Visual Studio Code como un entorno de desarrollo completo y eficiente para trabajar, mejorando la productividad y facilitando el mantenimiento del código en el proyecto del odontograma.



Para comenzar con el desarrollo, clona el repositorio desde tu sistema de control de versiones:

```
git clone https://github.com/JhordanRicaldi/Proyecto-fin-de-curso-taller-de-proyectos-2.git
```

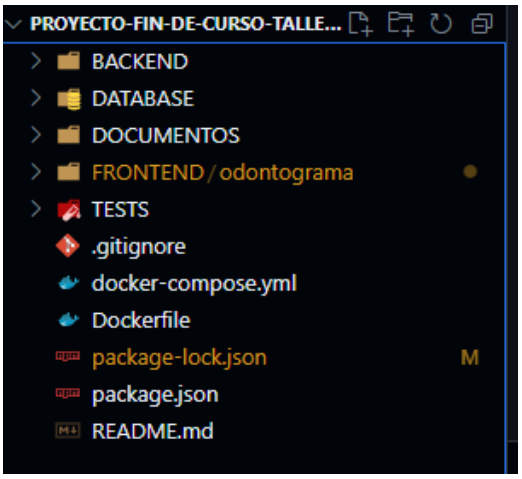
Abre una terminal de git bash o windows en el directorio donde clonaste el proyecto y ejecuta:

```
code .
```

Una vez clonado el proyecto veras una estructura de archivos y directorios como esta, a continuación te brindamos una descripción de la estructura del proyecto

```
├── BACKEND                 # Directorio principal del backend
│   ├── Dockerfile          # Configuración para Dockerizar el backend
│   ├── README.md           # Documentación específica del backend
│   ├── config              # Configuraciones específicas del proyecto
│   ├── helpers             # Utilidades y funciones de ayuda
│   ├── index.js            # Punto de entrada principal del backend
│   ├── models              # Esquemas y modelos de MongoDB
│   ├── node_modules        # Dependencias de Node.js (ignorado por Git)
│   ├── package-lock.json   # Bloqueo de versiones de las dependencias
│   └── package.json        # Archivo de configuración de npm para el backend
├── DATABASE                # Directorio para la base de datos
│   ├── README.md           # Documentación de la base de datos
│   └── odontograma.mongodb  # Archivo para validar el esquema MongoDB
├── DOCUMENTOS              # Directorio para documentos  del proyecto
│   ├── ARTEFACTOS          # Documentos generados o entregables
│   ├── DESARROLLO DE PROYECTO  # Documento de desarrollo del proyecto
```

```
│   └── LISTA DE INTEGRANTES.docx     # Lista de integrantes del equipo
├── Dockerfile              # Dockerfile en el directorio raíz para frontend o herramientas)
├── FRONTEND                # Directorio principal del frontend
│   └── odontograma         # Directorio de la aplicación frontend Angular
├── README.md               # Documentación general del proyecto
├── TESTS                   # Directorio para pruebas automatizadas
│   ├── cypress             # Pruebas automatizadas utilizando Cypress
│   ├── cypress.config.js   # Configuración de Cypress
│   ├── package-lock.json   # Versiones de las dependencias para pruebas
│   └── package.json        # Archivo de configuración de npm para las pruebas
├── docker-compose.yml      # Archivo de configuración para Docker Compose
├── package-lock.json       # Bloqueo de versiones de las dependencias para todo el
proyecto
└── package.json            # Archivo de configuración de npm para todo el proyecto
```



Estructura de directorios

Comentarios Adicionales:

BACKEND: Contiene todos los archivos y directorios relacionados con el desarrollo del backend, incluyendo configuraciones, modelos de datos y utilidades.

DATABASE: Almacena el archivo de base de datos MongoDB y su documentación correspondiente.

DOCUMENTOS: Agrupa todos los documentos relacionados con el proyecto, como artefactos, documentos de desarrollo y evaluación, y la lista de integrantes del equipo.

FRONTEND: Directorio principal para el desarrollo de la aplicación frontend Angular.

TESTS: Incluye pruebas automatizadas utilizando Cypress, con su configuración y archivos de dependencias separados.

Dockerfile y docker-compose.yml: Archivos de configuración para contenerizar y gestionar los contenedores Docker del proyecto.

README.md: Documentación general del proyecto en el directorio raíz, proporcionando una visión general y posiblemente instrucciones de configuración y uso.

IV. INSTALACIÓN Y USO

Necesitaras instalar las siguientes bibliotecas para la ejecucion del proyecto estas dependencias funcionaran de manera global, abre una terminal en el directorio raiz del proyecto

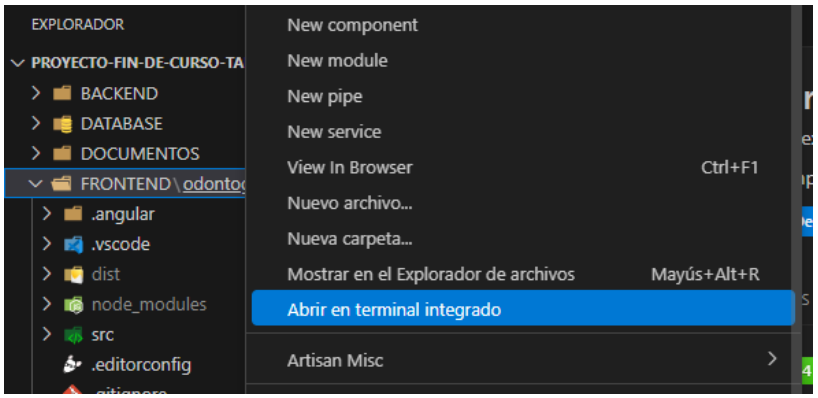
```
npm install -g @angular/cli
```

```
npm install -g vercel
```

```
npm install -g json-server
```

Frontend Odontograma

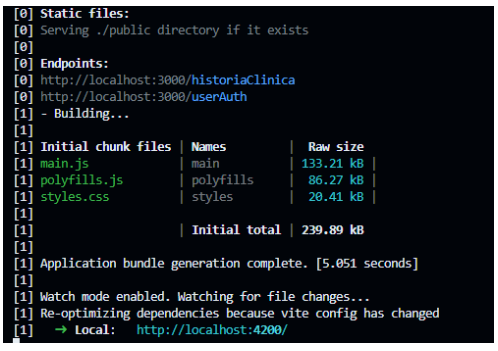
Ubica la carpeta FRONTEND dentro de la estructura de tu proyecto. haz clic derecho sobre la carpeta FRONTEND/odontograma para desplegar el menú contextual y abre un terminal integrado, una vez se abra ejecuta los siguientes comandos

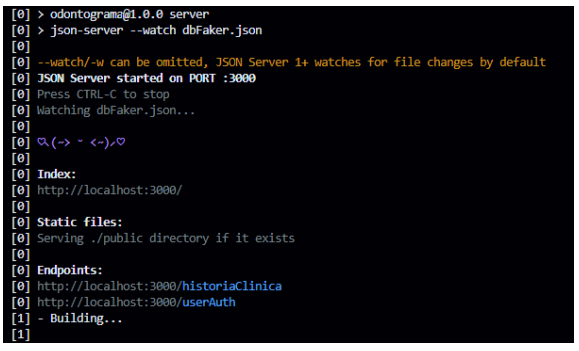



```
npm install
```

```
npm start
```

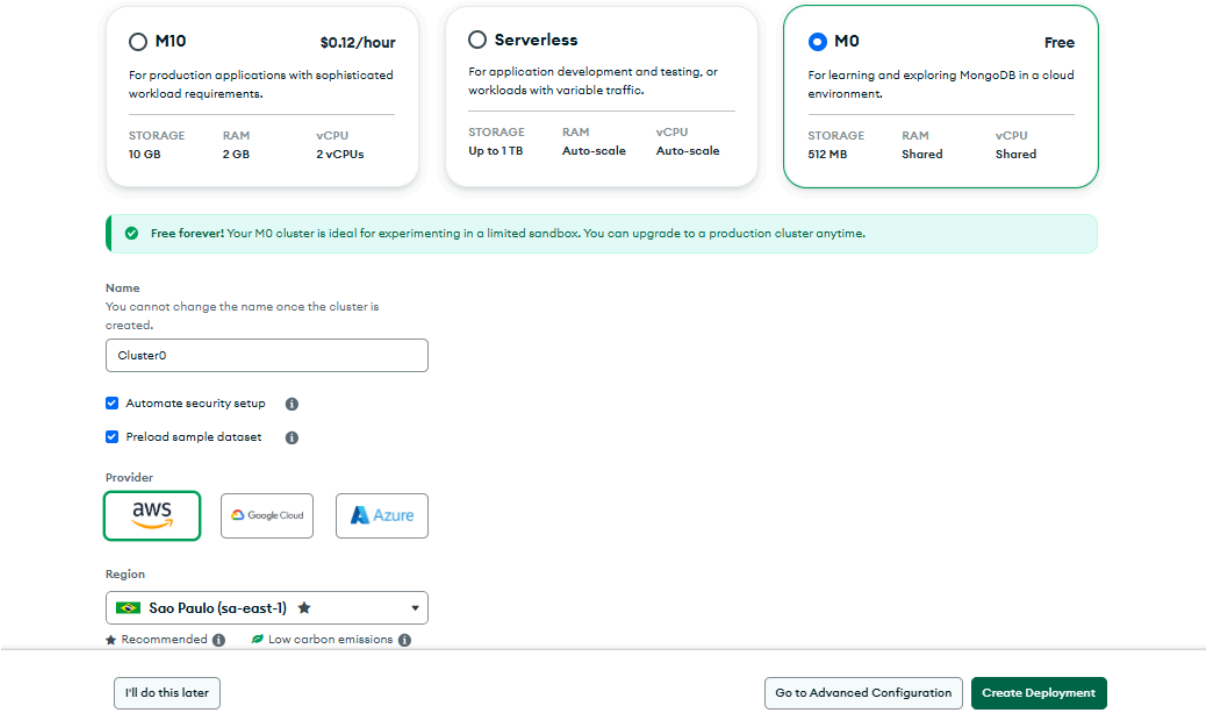
Veras la siguiente salida la url donde esta desplegada el frontend , en seguida veras los puertos disponibles asi que navega a el enlace que te redirigirá a la web de odontograma, en caso de que quieras utilizar datos fake de prueba podrías visitar los enlaces de json server



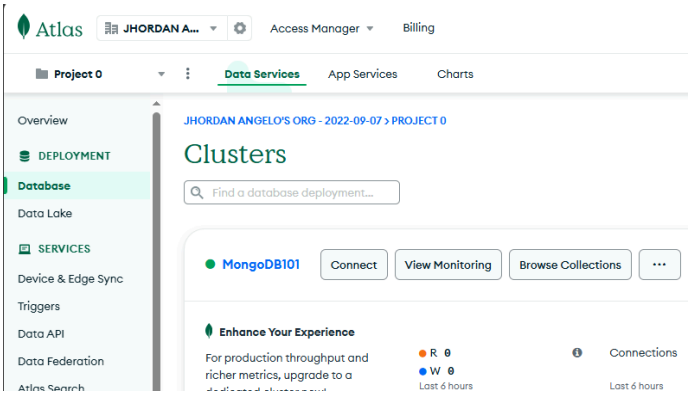


V. Configuración MongoDB

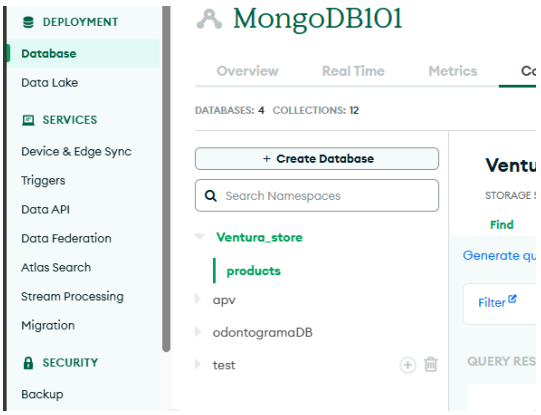
Crea una Cuenta en mongo db atlas mongodb.com ,luego de iniciar sesion con tu cuental crea un nuevo cluster para almacenar tu base de datos, elige el plan gratuito que ofrece mongodb atlas, despues selecciona el proveedor y region de tu preferencia



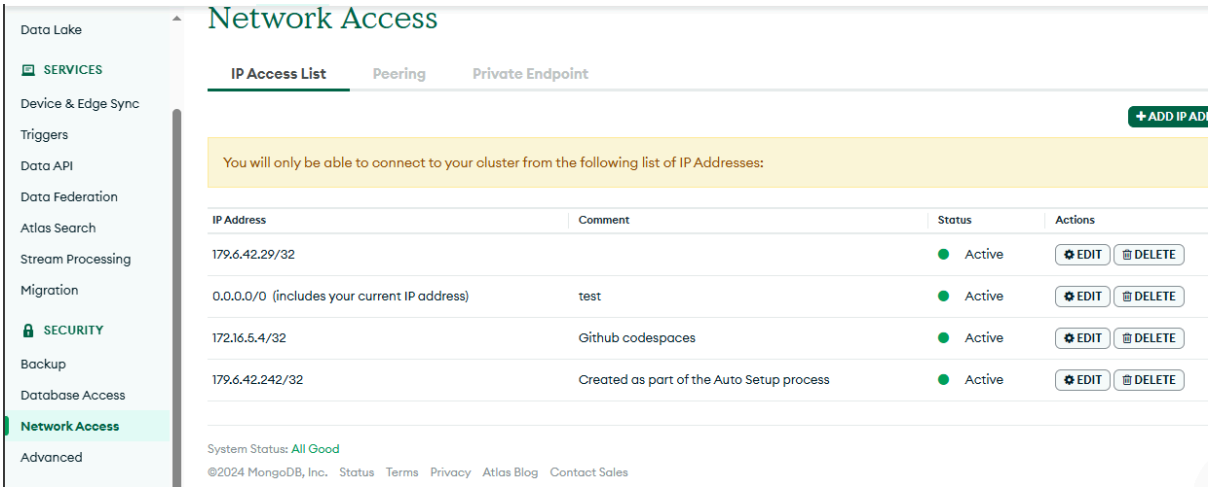
Luego de registrarte veras un panel con distintas configuraciones y opciones, dirígete a database y crea una nueva base de datos, siempre puedes volver a modificar características y configuraciones luego de crear el cluster



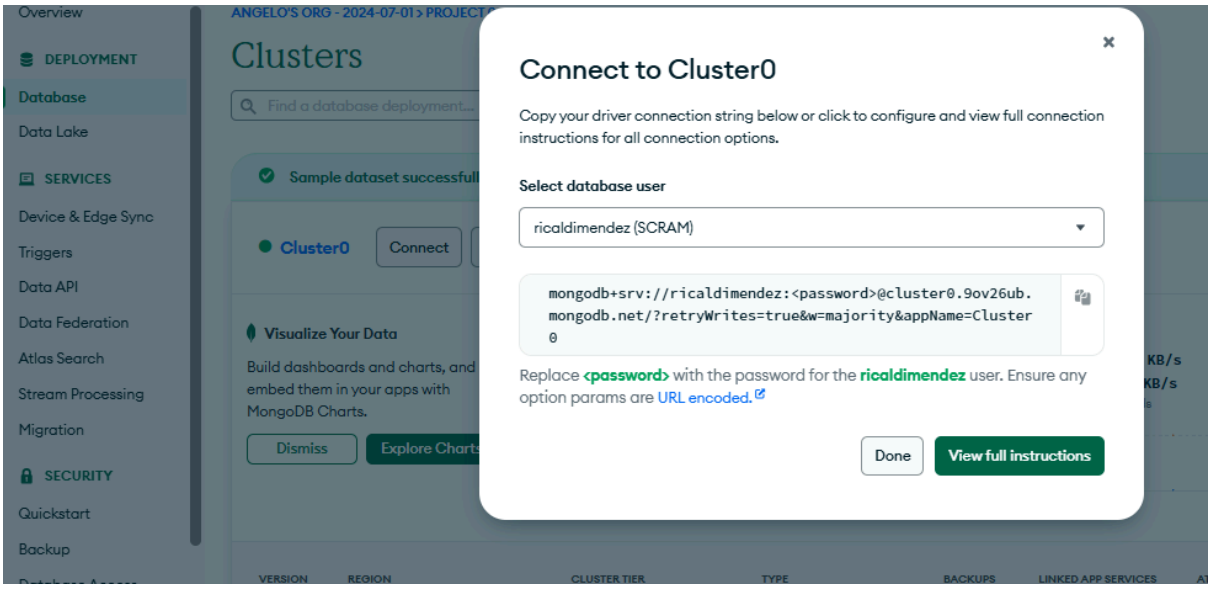
Dirígete a Browse Collections, alli podras crear una nueva base de datos



Tambien puedes configurar las ips de accesos de red para conectarte con distintos equipos, puedes usar el comodin 0.0.0.0 para que acepte cualquier ip en modo de pruebas, es útil cuando utilizas diferentes ordenadores



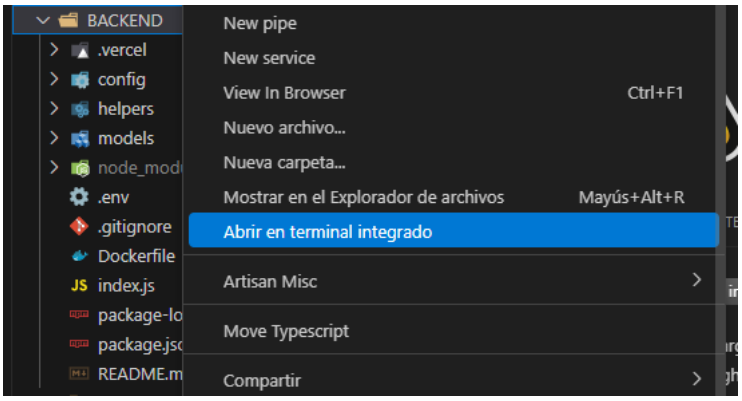
En el menu izquierdo dirigete a Deployment - Database - > Custer# Connect, alli podras obtener tu cadena de conexion para conectarla a la aplicacion



Copia la cadena de conexión en el archivo .env en MONGO_URI=, dentro del directorio backend y proporciona tu nombre de usuario y contraseña de la base de datos para establecer la conexión con la aplicación. Estas credenciales se generaron durante la configuración inicial del usuario en MongoDB Atlas

Backend Odontograma

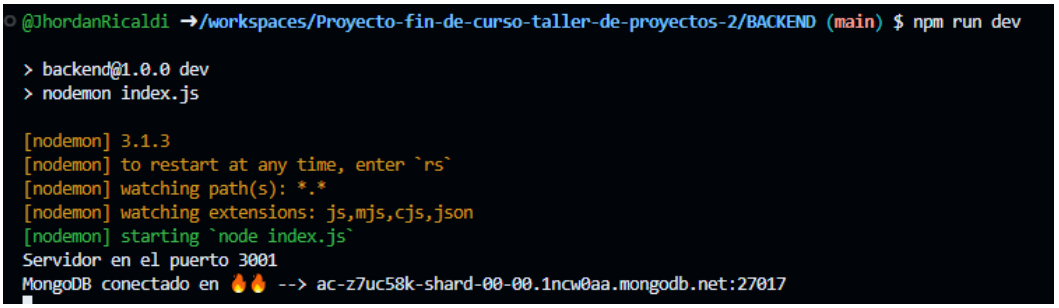
Ubica la carpeta BACKEND dentro de la estructura de tu proyecto. haz clic derecho sobre la carpeta BACKEND para desplegar el menú contextual y abre un terminal integrado, una vez se abra ejecuta los siguientes comandos




```
npm install
```

```
npm run dev
```

Veras la siguiente salida la donde esta el puerto disponible ,asi que navega a https://localhost:3001 el enlace que te redirigirá al web service del backend,



Endpoints:
A continuación se detallan los endpoints disponibles en la API del proyecto de odontograma, desarrollado con Node.js, Express y MongoDB:

GET /

- **Descripción:** Endpoint de prueba para verificar la disponibilidad del servidor.
- **Método HTTP:** GET
- **Ruta:** /

Respuesta Exitosa:

```
{
    "ok": true,
    "msg": "Respondío del backend"
}
```

Código de Estado: 200 OK

POST /pacientes

- **Descripción:** Crea un nuevo registro de paciente en la base de datos.
- **Método HTTP:** POST
- **Ruta:** /pacientes

Cuerpo de la Petición:

```
{
```

```
    "nombres": "Nombre del paciente",
    "apellidos": "Apellidos del paciente",
    "dni": "Número de DNI del paciente",
    "edad": "Edad del paciente"
    // Otros campos según la estructura del modelo Paciente
}
```

Respuesta Exitosa:

```
{
    "mensaje": "Paciente creado exitosamente"
}
```

- **Código de Estado:** 201 Created

GET /pacientes

- **Descripción:** Obtiene todos los pacientes almacenados en la base de datos.
- **Método HTTP:** GET
- **Ruta:** /pacientes
- **Respuesta Exitosa:** Lista de objetos JSON, cada uno representando un paciente.
- **Código de Estado:** 200 OK

POST /enviar-mail

- **Descripción:** Envía un correo electrónico al paciente con un archivo PDF adjunto conteniendo su odontograma.
- **Método HTTP:** POST
- **Ruta:** /enviar-mail

Cuerpo de la Petición:

```
{
    "nombre": "Nombre del paciente",
    "correo": "Correo electrónico del paciente",
    "DNI": "Número de DNI del paciente"
}
```

Respuesta Exitosa:

```
{
    "mensaje": "Correo enviado exitosamente"
}
```

- **Código de Estado:** 200 OK

POST /buscar-paciente

- **Descripción:** Busca un paciente por su nombre en la base de datos.
- **Método HTTP:** POST
- **Ruta:** /buscar-paciente

Cuerpo de la Petición:

```
{
    "nombre": "Nombre del paciente a buscar"
}
```

- **Respuesta Exitosa:** Objeto JSON que representa al paciente encontrado.
- **Código de Estado:** 200 OK

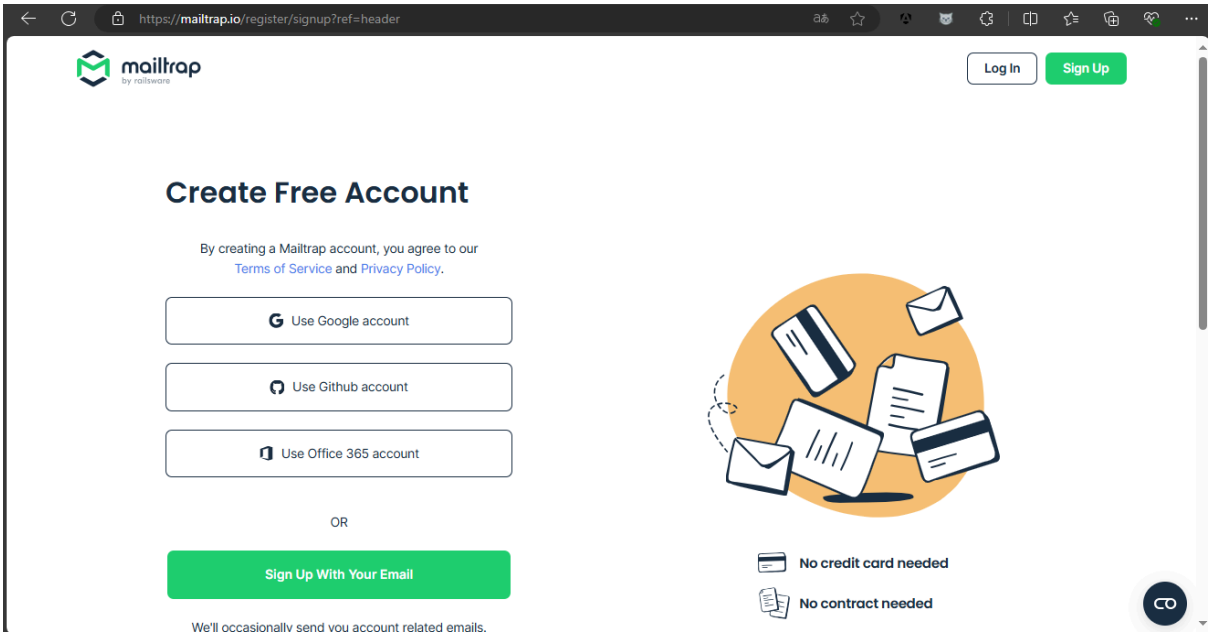
Respuesta en Caso de No Encontrar al Paciente:

```
{
    "error": "Paciente no encontrado"
}
```

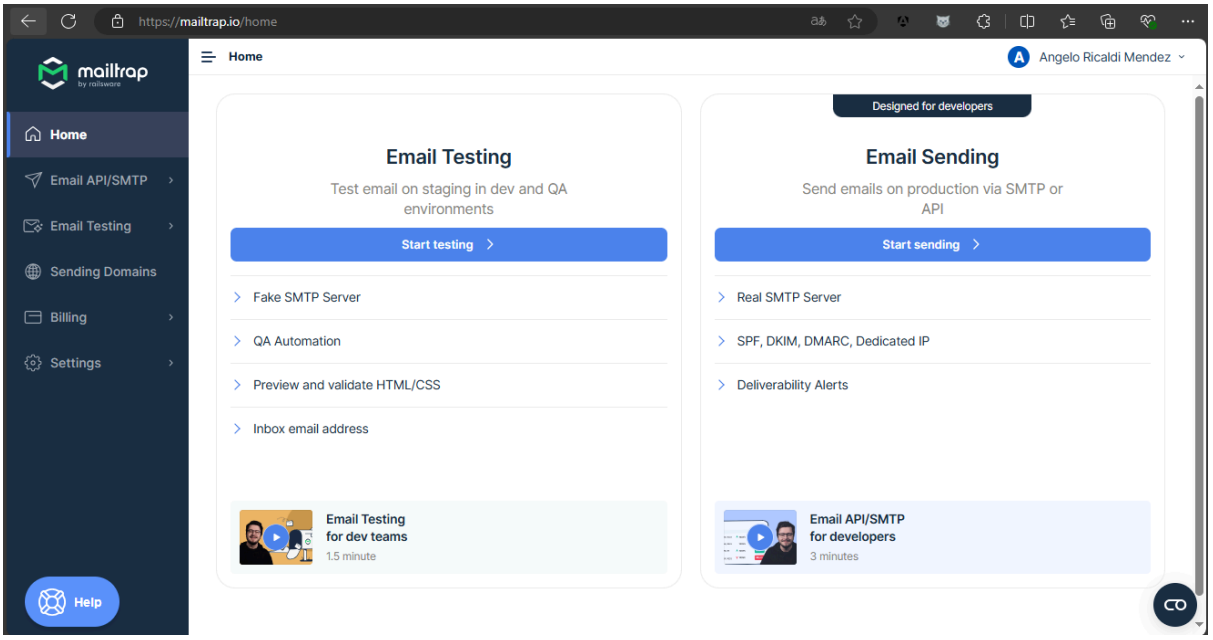
- **Código de Estado:** 404 Not Found en caso de que no se encuentre al paciente.

Configuración de correo electrónico

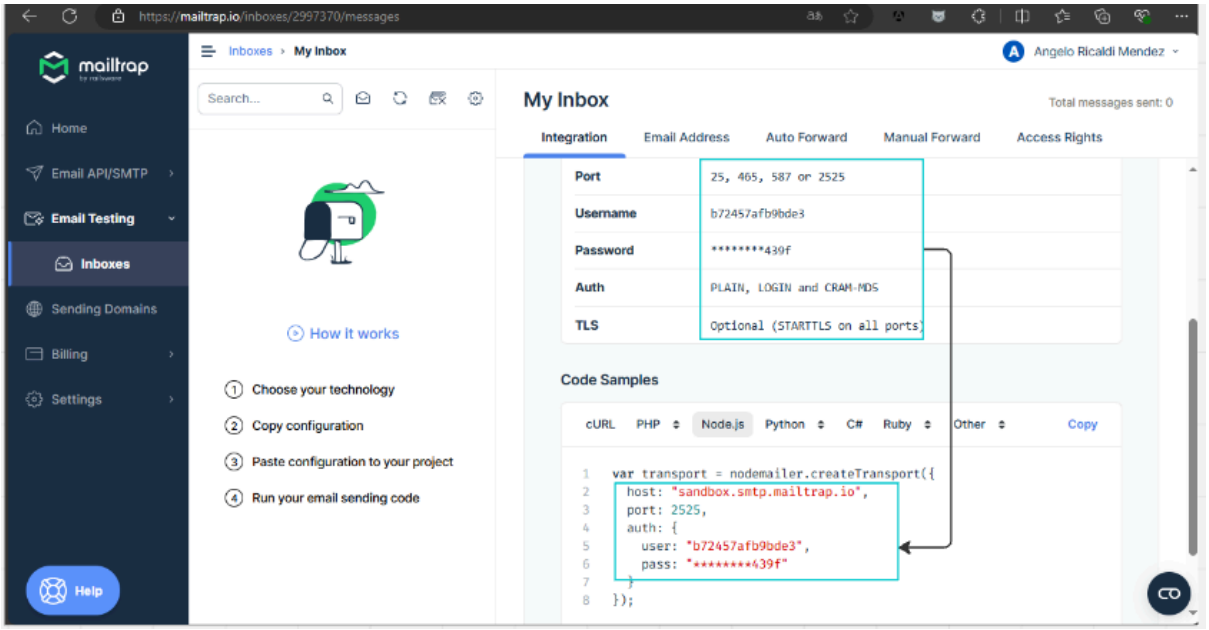
Crea una Cuenta en mailtrap mailtrap.io ,luego de iniciar sesion con tu cuenta elige la opcion start testing para poder crear un



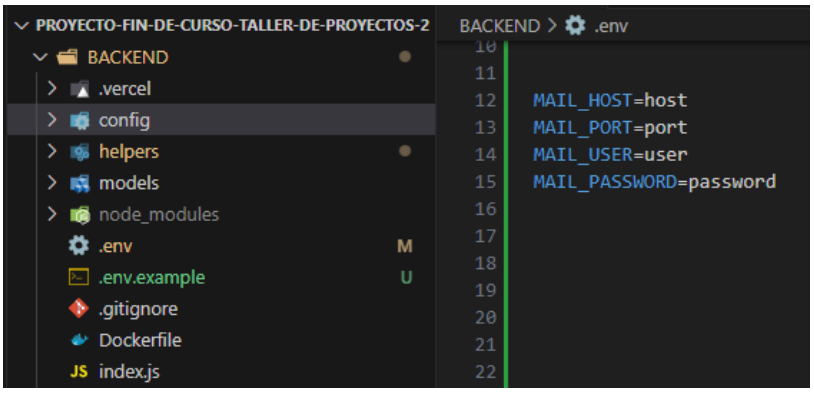
Esto te llevará al entorno de pruebas de Mailtrap, donde podrás crear buzones de correo virtuales para inspeccionar y depurar tus correos electrónicos antes de enviarlos a producción.



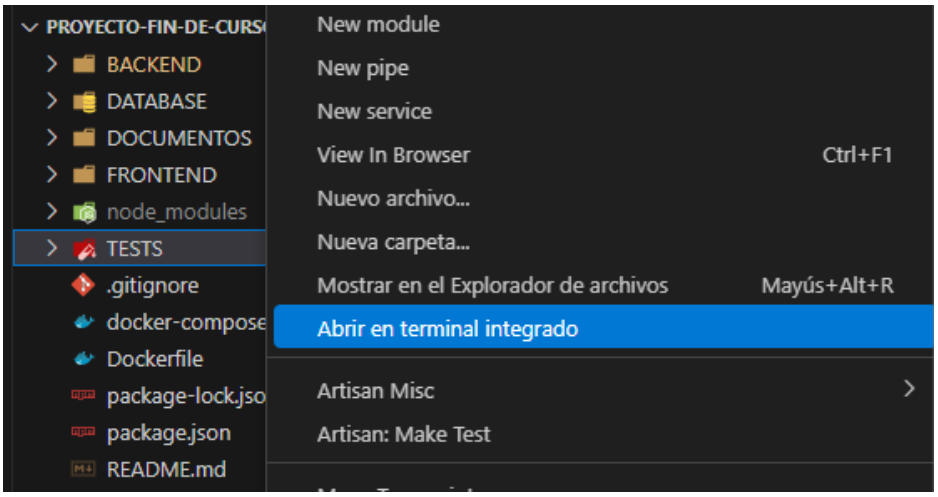
Utiliza las credenciales proporcionadas por Mailtrap (como el host SMTP, el puerto y las credenciales de autenticación) en tu aplicación para que pueda enviar correos electrónicos a través de Mailtrap.



Dirígete al directorio BACKEND y ubica el archivo .env, en caso de no tenerlo, crealo y coloca tus credenciales proporcionadas por mail trap en el archivo .env estarán ubicados en la parte superior, podrás ver el código fuente para integrarlo a otros lenguajes de programación en caso de que lo necesites, para este proyecto usaremos solo los valores de las credenciales, con esas configuraciones podrás recibir correo desde la aplicación de odontograma



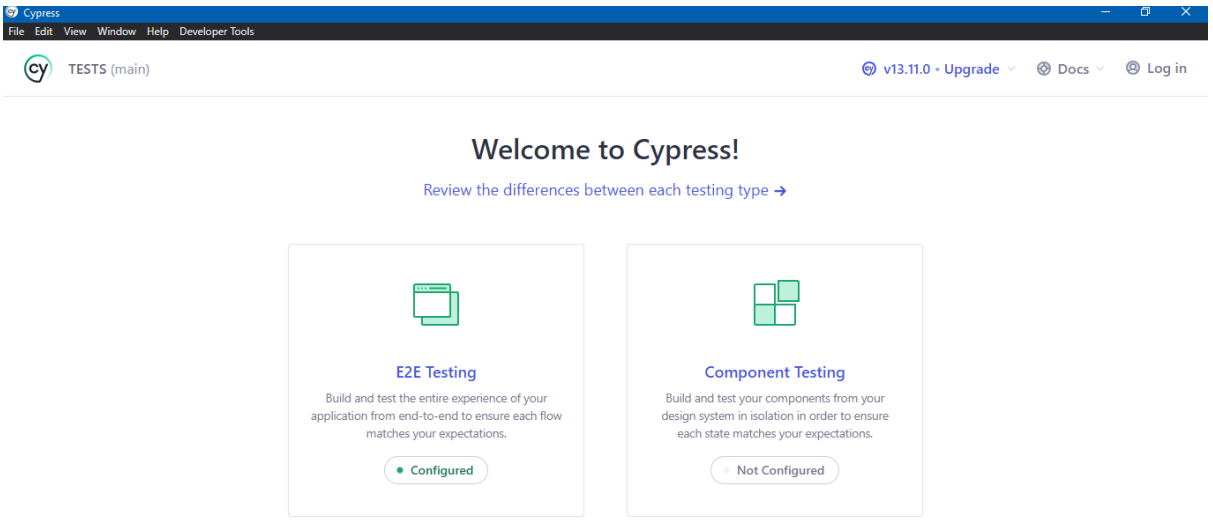
Testing Odontograma
Ubica la carpeta TESTING dentro de la estructura de tu proyecto. haz clic derecho sobre la carpeta TESTING para desplegar el menú contextual y abre un terminal integrado, una vez se abra ejecuta los siguientes comandos



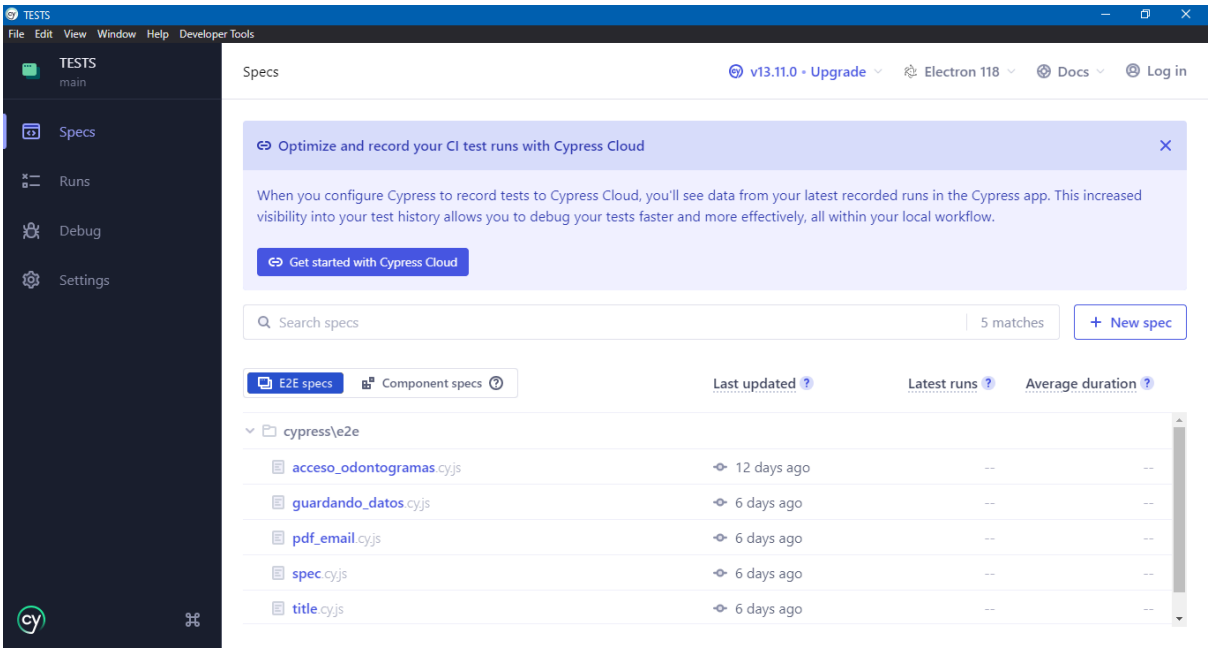

```
npm install
```

```
npm run cy:open
```

Veras la ventana de cypress donde deberas elegir un navegador para realizar los test y tambien veras la opcion para usarlo como aplicacion de escritorio, en seguida selecciona la opcion E2E



El programa detectara el directorio con los test escritos, asi que haz click sobre el test que desees probar, los test tienen la extension cy.js



VI. CONFIGURACIÓN DEL HOSTING

Despliegue del Backend en Vercel

- Crear una cuenta en Vercel vercel.com
- Regístrate en Vercel si aún no tienes una cuenta.
- Inicia sesión con tus credenciales.

Configurar tu Proyecto:

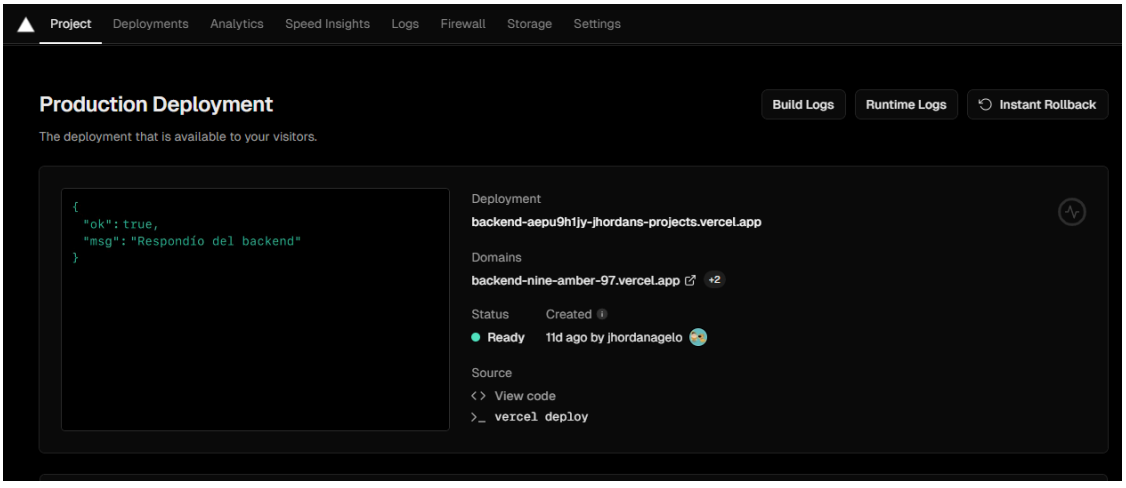
Asegúrate de tener un archivo vercel.json en la raíz de tu proyecto BACKEND. En este archivo, puedes definir configuraciones específicas para el despliegue en Vercel, como las variables de entorno y las rutas de acceso.

```json
{
    "version": 2, // Versión del archivo de configuración (actualmente 2)
    "builds": [
      {
        "src": "index.js", // Archivo de entrada para la construcción
        "use": "@vercel/node" // Tipo de construcción (Node.js en este caso)
      }
    ],
    "routes": [
      {
        "src": "/(.*)", // Expresión regular para todas las rutas
        "dest": "index.js" // Archivo de destino para las rutas
      }
    ]
  }
```

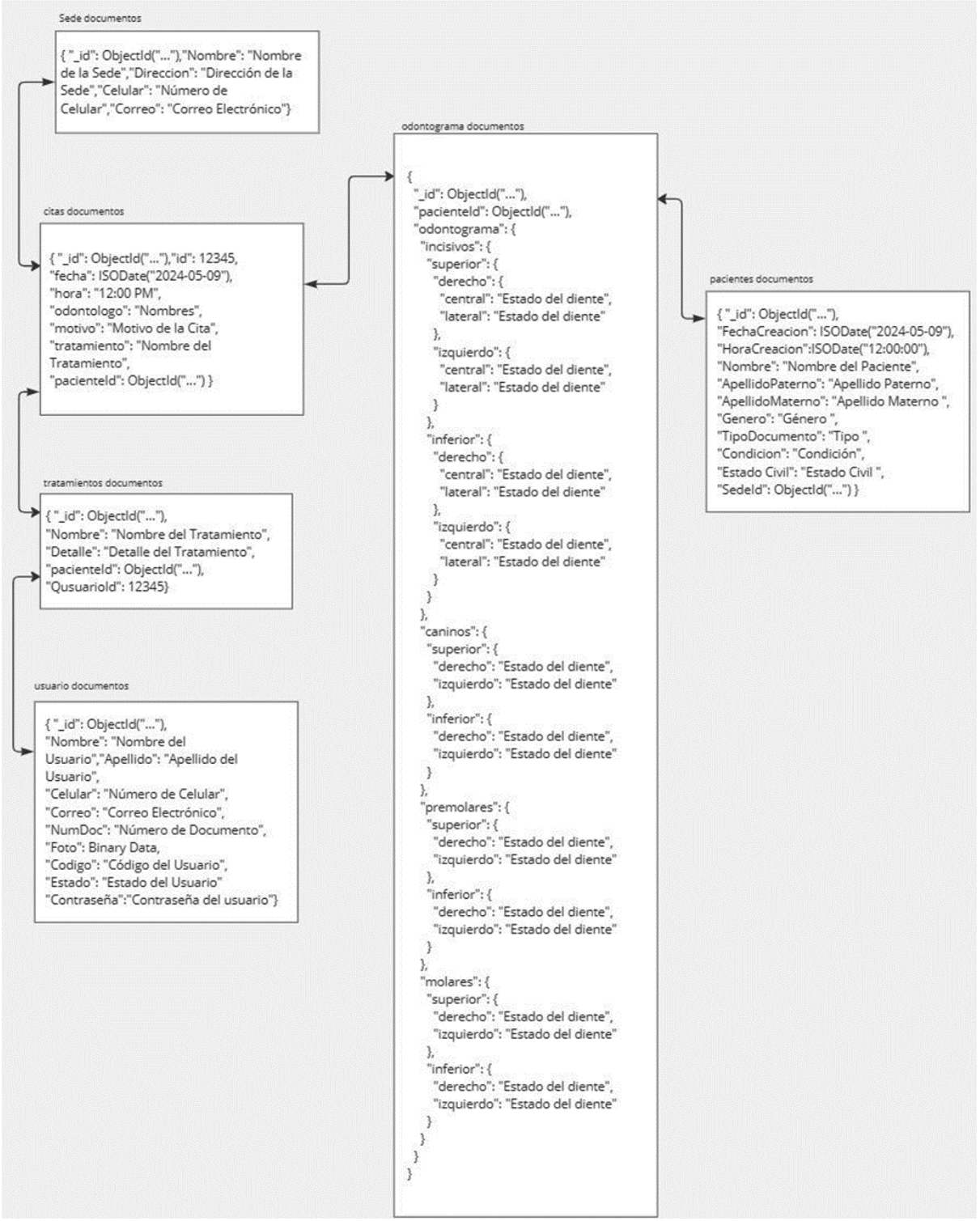
Desplegar tu Aplicación:
Dirigete a backend y abre una terminal, ejecuta el siguiente comando para desplegar tu backend:

```
vercel --prod   o también: vercel
```

Vercel construirá y desplegará automáticamente tu backend en producción.
Obtendrás una URL única para acceder a tu backend en línea.
Verifica el Despliegue:
Visita la URL proporcionada por Vercel para asegurarte de que tu backend esté funcionando correctamente.



VII. DIAGRAMA DE BASE DE DATOS



VIII. PROCESOS DEL SISTEMA WEB

Visión General del Sistema

El odontograma esta desarrollado con el siguiente stack: Angular, Node.js, Express y MongoDB.

Facilita significativamente la gestión y visualización de registros odontológicos mediante varias funcionalidades clave:

Guía de Usuario

Interfaz de Usuario

A. Permite visualizar una lista completa de todos los pacientes registrados en el sistema. Desde aquí, los usuarios pueden realizar búsquedas rápidas por nombre, número de identificación o cualquier otro criterio relevante.
B. Navegación básica por las diferentes secciones (listado de pacientes, ficha odontológica, etc.).

Gestión de Pacientes

C. Disponemos de una interfaz para añadir información del odontograma de pacientes.
D. Asignación de historial odontológico a pacientes mediante el sistema de citas.

Odontograma

E. Funcionalidades específicas del odontograma digital.
F. Cómo registrar procedimientos dentales, diagnósticos, tratamientos y notas relevantes.

Búsqueda y Filtrado

G. Métodos para buscar pacientes específicos o registros odontológicos dentro del sistema.
H. Uso de filtros para acceder rápidamente a información relevante.

Exportación e Informes

I. Tenemos la opcion de generar el informe en formato PDF
J. Generación de informes odontológicos para impresión o envío digital por correo electrónico.

IX. FUTURAS EXPANSIONES

Servicios de Imágenes Dentales:

Integración con sistemas de imágenes dentales para permitir la visualización directa de radiografías, tomografías y fotografías intraorales dentro de la plataforma.

Funcionalidades avanzadas de visualización y análisis de imágenes para apoyar el diagnóstico y tratamiento odontológico.

Inteligencia Artificial y Análisis Predictivo:

Implementación de algoritmos de inteligencia artificial para análisis predictivo de patrones dentales y diagnósticos.

Capacidades de aprendizaje automático para identificar tendencias en la salud bucodental de los pacientes y sugerir tratamientos personalizados.

Teleodontología y Consultas Virtuales:

Desarrollo de módulos para soportar consultas virtuales y teleodontología, permitiendo a los pacientes y profesionales de la salud dental interactuar a distancia de manera efectiva.

Integración de herramientas de videoconferencia y mensajería segura para consultas en tiempo real.

Integración con Sistemas de Gestión Hospitalaria o Clínica:

Conexión con sistemas de gestión hospitalaria o clínica para compartir datos de manera segura y mantener la coherencia en los registros médicos del paciente.

Sincronización de calendarios, gestión de citas y coordinación de servicios médicos complementarios.

Seguridad y Cumplimiento Normativo Mejorado:

Mejora continua de las medidas de seguridad y cumplimiento normativo para proteger la integridad y privacidad de los datos de los pacientes.

Implementación de estándares avanzados de encriptación, autenticación y auditoría de acceso.

Personalización y Adaptabilidad:

Ofrecer opciones de personalización adicionales para adaptarse a diferentes modelos de práctica dental y necesidades específicas del usuario.

Desarrollo de funcionalidades personalizadas según los comentarios y requerimientos de los usuarios y profesionales del sector odontológico.