

DIAGRAMAS DEL DISEÑO DE ARQUITECTURA DEL PRODUCTO



Integrantes :

Alfredo Berrocal Chavez
David Azael Hernandez Escudero
Itzel Mancilla Betanzos
Elias Mayor Gonzalez
Marco Antonio Solis Pardo

Diagrama de contexto del sistema (Nivel 1)	3
Propósito del diagrama	3
Descripción de los componentes	3
Importancia	4
Diagrama del contenedor (Nivel 2)	5
Explicación	5
Interacciones:	6
Diagrama de Componentes (Nivel 3)	7
Capa de Presentación	7
Capa de Lógica de Negocio	7
Capa de Datos	7
Relaciones	8
Diagrama de Código (clase)(Nivel 4)	9
Diagrama de Código más de cerca	9
Explicación de Diagrama de Código	10
Paquetes Principales	10
Clases Principales	10
Usuario	10
Plataforma	11
Servidor	11
BaseDeDatos	11
TestAnsiedad	11
Tutorial	11
Componente del Servidor	12
APIDeAutenticacion	12
ModuloDePruebasDeAnsiedad	12
ModuloDeTutoriales	12
Relaciones y Conexiones	12
Interacción entre Clases	12
Aspectos Importantes	13
Seguridad y Autenticación	13
Almacenamiento y Recuperación	13
Estructuración	13
Diagrama de Entidad-Relación de la base de Datos (Nivel 5)	13
Entidades:	13
1. Usuario	13
2. Test_Ansiedad	13
3. Pregunta	14
4. Respuesta	14
5. Tutorial	14
6. Nivel_Ansiedad	14
Descripción General de la Base de Datos:	15

Diagrama de contexto del sistema (Nivel 1)

El diagrama de contexto del sistema MindPeace tiene como objetivo proporcionar una vista general de las interacciones principales entre el sistema y sus actores externos. Este diagrama es fundamental para comprender cómo el sistema se relaciona con los usuarios y administradores, así como las funcionalidades clave que ofrece.

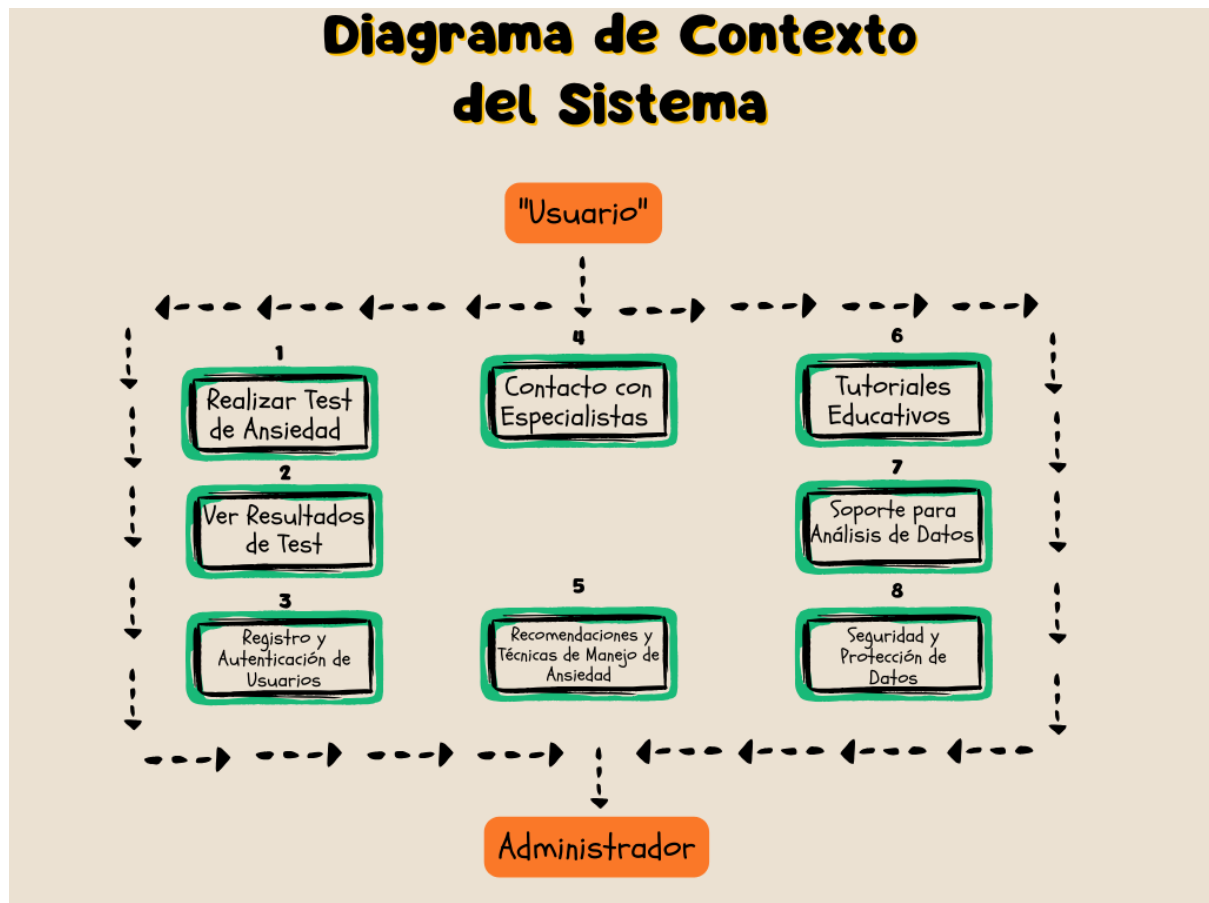


Diagrama 1. Contexto del Sistema

Propósito del diagrama

El propósito de este diagrama de contexto es describir de manera clara y concisa cómo los usuarios y administradores interactúan con el sistema MindPeace y cómo se gestionan las principales funcionalidades de la plataforma. Este diagrama ayuda a visualizar la estructura del sistema, las responsabilidades de cada actor y las interacciones principales entre ellos.

Descripción de los componentes

Usuarios:

- Los usuarios son las personas que utilizan la plataforma MindPeace para evaluar su nivel de ansiedad y acceder a recursos para manejarla.
- **Interacciones:**

- **Realizar Test de Ansiedad:** Los usuarios pueden realizar un test interactivo para evaluar su nivel de ansiedad.
- **Ver Resultados de Test:** Los usuarios pueden ver los resultados de sus tests.
- **Servicio de Autenticación de Usuarios:** Los usuarios pueden registrarse y autenticarse en la plataforma para guardar sus resultados y progreso.
- **Contactar con Especialistas:** Los usuarios pueden ponerse en contacto con especialistas en ansiedad a través de la plataforma.
- **Acceder a Tutoriales Educativos:** Los usuarios pueden acceder a tutoriales educativos sobre técnicas de manejo de la ansiedad.
- **Recibir Recomendaciones y Técnicas de Manejo de Ansiedad:** Basado en los resultados del test, los usuarios reciben recomendaciones personalizadas.
- **Consultar FAQs:** Los usuarios pueden consultar las preguntas frecuentes para resolver dudas comunes.

Administrador:

- Los administradores son los responsables de gestionar y mantener la plataforma MindPeace.
- **Interacciones:**
 - **Ver Resultados de Encuesta:** Los administradores pueden ver y analizar los resultados de las encuestas realizadas.
 - **Actualización de Contenido:** Los administradores pueden actualizar, añadir o eliminar contenido educativo sobre manejo de la ansiedad y técnicas de relajación.
 - **Soporte para Análisis de Datos:** Los administradores tienen acceso a herramientas para recolectar y analizar datos de uso del sistema.
 - **Seguridad y Protección de Datos:** Los administradores se aseguran de implementar medidas de seguridad para proteger los datos personales y de salud de los usuarios.

Importancia

Este diagrama es crucial para entender la estructura del sistema MindPeace, permitiendo una visión clara de las funcionalidades clave y las interacciones entre los actores principales. Ayuda a identificar las responsabilidades de cada componente y a asegurar que todas las partes del sistema funcionen de manera coherente y eficiente.

Diagrama del contenedor (Nivel 2)

El diagrama de contenedor a nivel 2 del modelo C4, para MindPeace su objetivo es la de proporcionar una vista detallada de la arquitectura del sistema, mostrando cómo los diferentes componentes de software se interrelacionan y cómo gestionan las principales funcionalidades de la plataforma. Este diagrama es crucial para entender la estructura del sistema, permitiendo una visión clara de los contenedores de software, sus responsabilidades y las interacciones entre ellos.

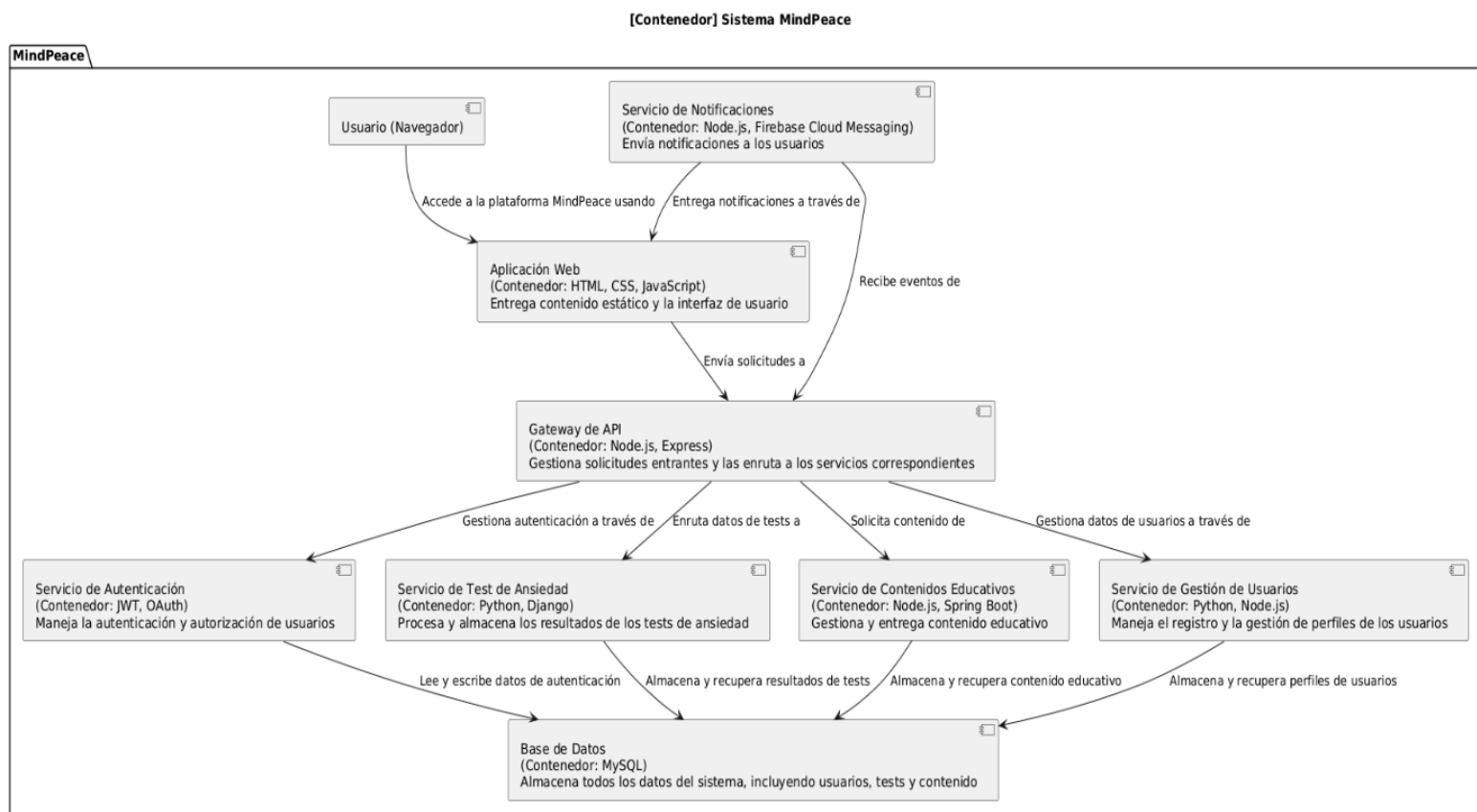


Diagrama 2. Contenedor

Explicación

1. Usuario (Navegador):

- Representa la interfaz del usuario final que interactúa con la aplicación a través de un navegador web.

2. Aplicación Web:

- Contenedor que comprende tecnologías de frontend como HTML, CSS y JavaScript. Se encarga de entregar contenido estático y la interfaz de usuario al navegador del usuario.

3. Servicio de Notificaciones:

- Contenedor que utiliza Node.js y Firebase Cloud Messaging para enviar notificaciones a los usuarios. Está conectado con la aplicación web para entregar dichas notificaciones.

4. **Gateway de API:**
 - Este contenedor, implementado con Node.js y Express, actúa como intermediario para todas las solicitudes entrantes, enrutándolas a los servicios correspondientes dentro de la arquitectura.
5. **Servicio de Autenticación:**
 - Maneja todo lo relacionado con la autenticación y autorización de usuarios utilizando JWT y OAuth. Este servicio valida las credenciales de los usuarios y gestiona las sesiones.
6. **Servicio de Test de Ansiedad:**
 - Implementado con Python y Django, este servicio procesa y almacena los resultados de los tests de ansiedad que los usuarios realizan a través de la aplicación web.
7. **Servicio de Contenidos Educativos:**
 - Utiliza tecnologías como Python y Node.js para gestionar y entregar contenido educativo relacionado con la ansiedad a los usuarios. Se conecta a la base de datos para almacenar y recuperar dicho contenido.
8. **Servicio de Gestión de Usuarios:**
 - Con Node.js y Spring Boot, este servicio maneja el registro y la gestión de perfiles de usuarios, incluyendo la edición de información personal y preferencias.
9. **Base de Datos:**
 - Contenedor MySQL que almacena todos los datos del sistema, incluyendo información de usuarios, resultados de tests y contenido educativo. Los diferentes servicios interactúan con la base de datos para leer y escribir datos.

Interacciones:

- **Aplicación Web a Gateway de API:** La aplicación web envía solicitudes al Gateway de API, que luego las enruta a los servicios correspondientes.
- **Servicio de Notificaciones a Usuario:** El servicio de notificaciones envía alertas y notificaciones a los usuarios a través de la aplicación web.
- **Gateway de API a otros servicios:** Gestiona la autenticación y enruta los datos de tests a los servicios de Test de Ansiedad, solicita contenido del Servicio de Contenidos Educativos, y gestiona datos de usuarios a través del Servicio de Gestión de Usuarios.
- **Servicios a Base de Datos:** Varios servicios interactúan con la base de datos para guardar o recuperar información necesaria para sus operaciones.

Diagrama de Componentes (Nivel 3)

El diagrama de componentes de MindPeace sigue una arquitectura de 3 capas, diseñada para separar el sistema en tres niveles distintos: presentación, lógica de negocio y datos.

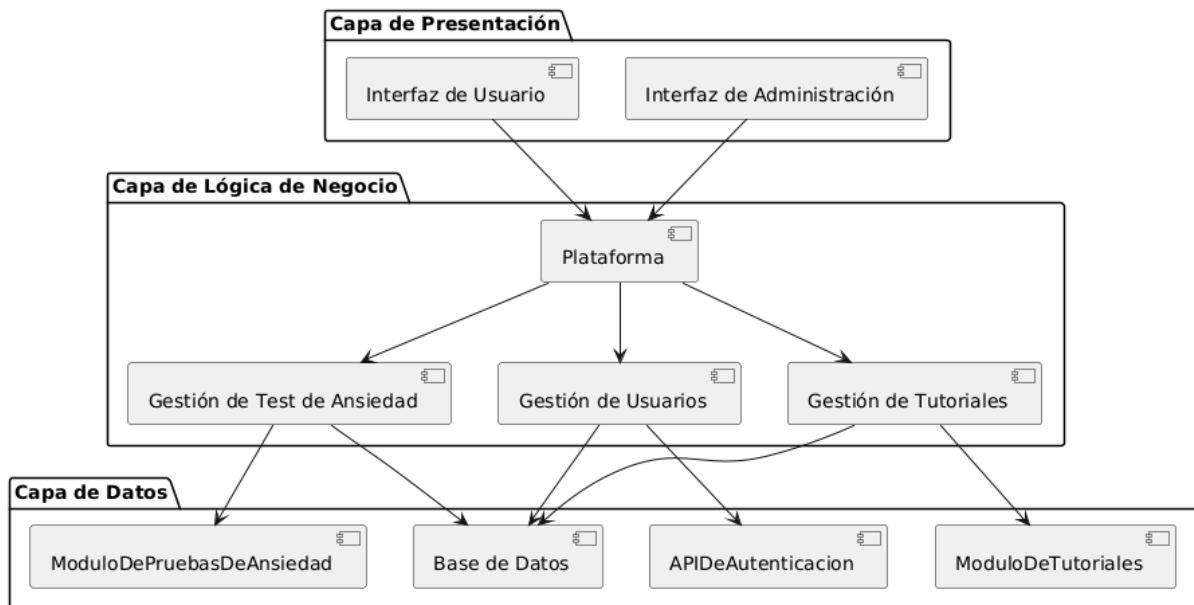


Diagrama 3. Componentes

Capa de Presentación

- **Interfaz de Usuario (InterfazUsuario):** Maneja la interacción con los usuarios.
- **Interfaz de Administración (InterfazAdministracion):** Maneja la interacción con los administradores del sistema.

Capa de Lógica de Negocio

- **Gestión de Test de Ansiedad (GestionTestAnsiedad):** Responsable de la lógica para manejar los tests de ansiedad.
- **Gestión de Usuarios (GestionUsuarios):** Responsable de la lógica para manejar los usuarios.
- **Gestión de Tutoriales (GestionTutoriales):** Responsable de la lógica para manejar los tutoriales educativos.
- **Plataforma:** Centraliza la gestión de las interacciones de la capa de presentación con la lógica de negocio.

Capa de Datos

- **Base de Datos (BaseDeDatos):** Almacena los datos del sistema, incluyendo usuarios, tests de ansiedad y tutoriales.
- **APIDeAutenticacion:** Provee servicios de autenticación para los usuarios.
- **ModuloDePruebasDeAnsiedad:** Maneja la lógica específica de los tests de ansiedad.
- **ModuloDeTutoriales:** Maneja la lógica específica de los tutoriales educativos.

Relaciones

- **Capa de Presentación** se comunica con la **Capa de Lógica de Negocio** a través de Plataforma.
- **Capa de Lógica de Negocio** interactúa con la **Capa de Datos**.
- Componentes específicos de la **Capa de Lógica de Negocio** interactúan con BaseDeDatos, APIDeAutenticacion, ModuloDePruebasDeAnsiedad, y ModuloDeTutoriales según sea necesario.

Diagrama de Código (clase)(Nivel 4)

En esta sección se proporciona una descripción detallada del diagrama de clases de la plataforma MindPeace, diseñada para ayudar a los usuarios a gestionar la ansiedad mediante tests interactivos y tutoriales educativos.

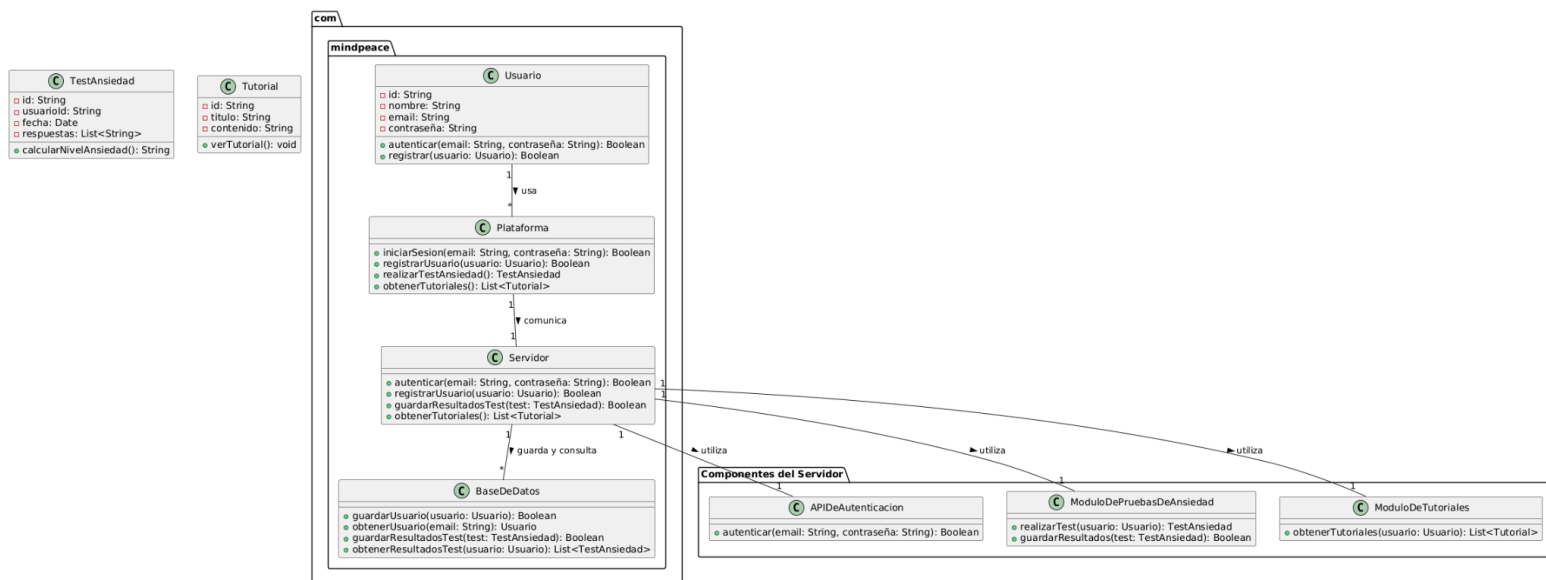


Diagrama 4. Código

Diagrama de Código más de cerca

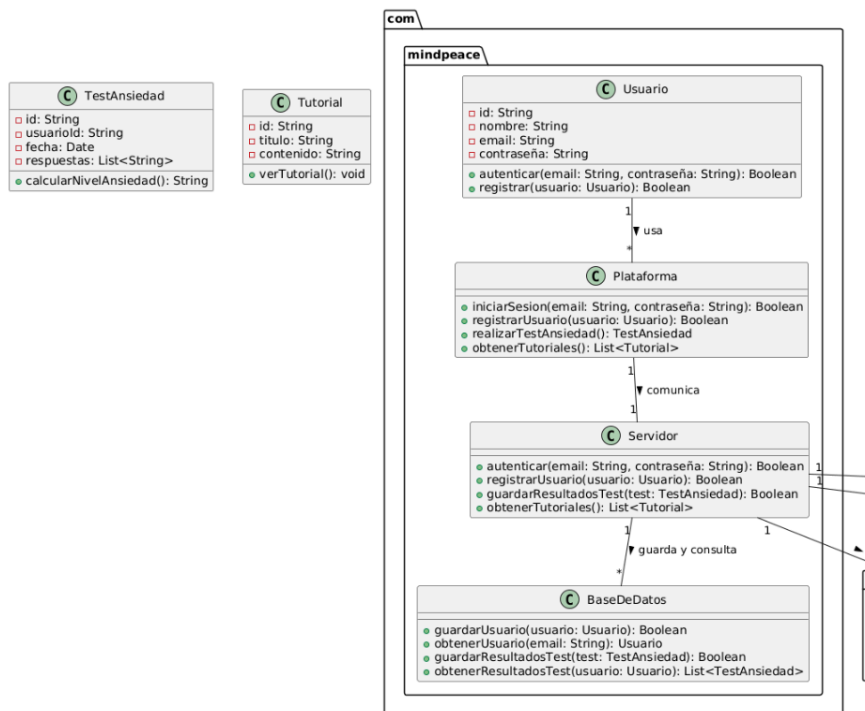


Diagrama 5. Código 2

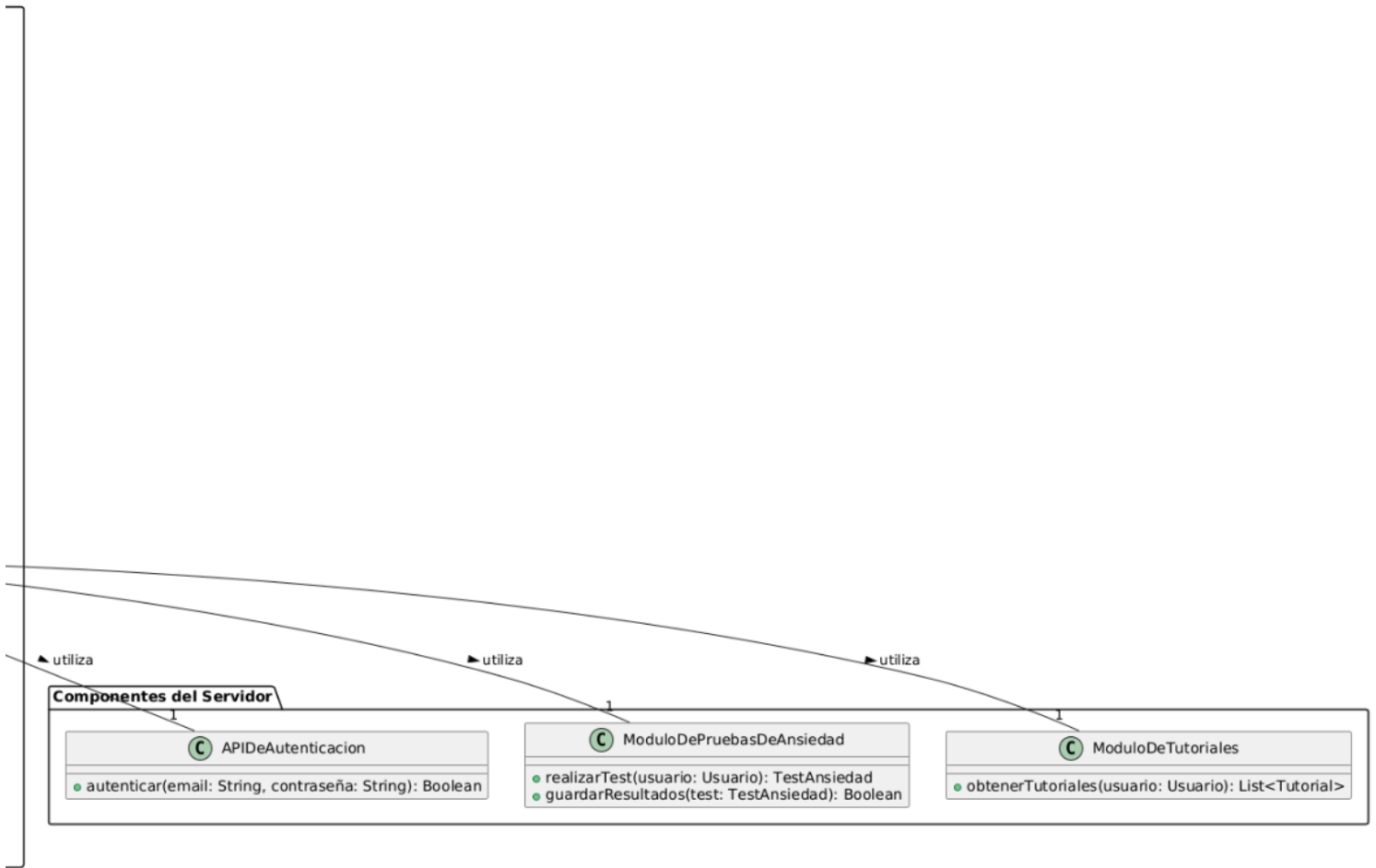


Diagrama 6. Código 3

Explicación de Diagrama de Código

Paquetes Principales

El paquete principal es “com.mindpeace”, que contiene las clases fundamentales para la funcionalidad de la aplicación.

Clases Principales

Usuario

Descripción: Representa a los usuarios de la plataforma, incluyendo métodos para autenticar y registrar.

- Atributos
 - id: String
 - nombre: String
 - email: String
 - contraseña: String

- Métodos
 - + autenticar(email: String, contraseña: String): Boolean
 - + registrar(usuario: Usuario): Boolean

Plataforma

Descripción: Clase que representa la plataforma de MindPeace, accesible desde dispositivos móviles y escritorio. Permite iniciar sesión, registrar usuarios, realizar tests de ansiedad y obtener tutoriales.

- Métodos
 - + iniciarSesion(email: String, contraseña: String): Boolean
 - + registrarUsuario(usuario: Usuario): Boolean
 - + realizarTestAnsiedad(): TestAnsiedad
 - + obtenerTutoriales(): List<Tutorial>

Servidor

Descripción: Clase que representa el backend de la aplicación, encargado de autenticar usuarios, registrar nuevos usuarios, guardar resultados de tests y proporcionar tutoriales.

- Métodos
 - + autenticar(email: String, contraseña: String): Boolean
 - + registrarUsuario(usuario: Usuario): Boolean
 - + guardarResultadosTest(test: TestAnsiedad): Boolean
 - + obtenerTutoriales(): List<Tutorial>

BaseDeDatos

Descripción: Clase que maneja el almacenamiento y la recuperación de datos en la base de datos.

- Métodos
 - + guardarUsuario(usuario: Usuario): Boolean
 - + obtenerUsuario(email: String): Usuario
 - + guardarResultadosTest(test: TestAnsiedad): Boolean
 - + obtenerResultadosTest(usuario: Usuario): List<TestAnsiedad>

TestAnsiedad

Descripción: Clase que representa un test de ansiedad realizado por el usuario.

- Atributos
 - id: String
 - usuarioid: String
 - fecha: Date
 - respuestas: List<String>
- Métodos
 - + calcularNivelAnsiedad(): String

Tutorial

Descripción: Clase que representa un tutorial educativo sobre técnicas de manejo de la ansiedad.

- Atributos

- id: String
 - título: String
 - contenido: String
- Métodos
 - + verTutorial(): void

Componente del Servidor

APIDeAutenticacion

Descripción: Clase encargada de manejar la autenticación de los usuarios.

- Métodos
 - + autenticar(email: String, contraseña: String): Boolean

ModuloDePruebasDeAnsiedad

Descripción: Clase que maneja la realización y el almacenamiento de los resultados de los tests de ansiedad.

- Métodos
 - + realizarTest(usuario: Usuario): TestAnsiedad
 - + guardarResultados(test: TestAnsiedad): Boolean

ModuloDeTutoriales

Descripción: Clase encargada de proporcionar los tutoriales educativos a los usuarios.

- Métodos
 - + obtenerTutoriales(usuario: Usuario): List<Tutorial>

Relaciones y Conexiones

Interacción entre Clases

- **Usuario usa Plataforma:** El usuario interactúa con la plataforma para iniciar sesión, registrar usuarios, realizar tests de ansiedad y obtener tutoriales.
- **Plataforma comunica con Servidor:** La plataforma se comunica con el servidor para procesar las solicitudes de autenticación, registro, almacenamiento de resultados y obtención de tutoriales.
- **Servidor guarda y consulta en BaseDeDatos:** El servidor utiliza la base de datos para almacenar y recuperar información de usuarios y resultados de tests.
- **Servidor utiliza Componentes del Servidor:** El servidor se apoya en los componentes APIDeAutenticacion, ModuloDePruebasDeAnsiedad y ModuloDeTutoriales para manejar la autenticación, las pruebas de ansiedad y los tutoriales, respectivamente.

Aspectos Importantes

Seguridad y Autenticación

La autenticación de usuarios es manejada por la clase `APIDeAutenticacion` para asegurar que solo usuarios registrados puedan acceder a las funcionalidades de la plataforma.

Almacenamiento y Recuperación

La `BaseDeDatos` es crucial para mantener la persistencia de los datos de los usuarios y los resultados de los tests de ansiedad.

Estructuración

La arquitectura del servidor está dividida en módulos para facilitar la escalabilidad y el mantenimiento del sistema.

Diagrama de Entidad-Relación de la base de Datos (Nivel 5)

Entidades:

1. Usuario

- **Campos:**

- `ID_Usuario` (PK): Identificador único del usuario.
- `Nombre`: Nombre del usuario.
- `Email`: Correo electrónico del usuario, único para cada usuario.
- `Contraseña`: Contraseña del usuario.
- `Fecha_de_Nacimiento`: Fecha de nacimiento del usuario.
- `Género`: Género del usuario.
- `Rol`: Rol del usuario (Estudiante, Profesor, Psicólogo).
- `Fecha_de_Registro`: Fecha en que el usuario se registró en el sistema.

- **Relaciones:**

- Un usuario puede realizar múltiples tests de ansiedad (relación 1 con `Test_Ansiedad`).
- Un usuario puede acceder a múltiples tutoriales (relación 1 con `Tutorial`).

2. Test_Ansiedad

- **Campos:**

- `ID_Test` (PK): Identificador único del test de ansiedad.
- `ID_Usuario` (FK): Identificador del usuario que realizó el test.
- `Fecha`: Fecha en que se realizó el test.
- `Resultado`: Resultado del test de ansiedad.

- **ID_Nivel** (FK): Identificador del nivel de ansiedad determinado por el test.
- **Relaciones:**
 - Cada test de ansiedad está asociado a un usuario (relación N:1 con **Usuario**).
 - Cada test de ansiedad puede tener múltiples respuestas (relación 1 con **Respuesta**).
 - Cada test de ansiedad está asociado a un nivel de ansiedad (relación N:1 con **Nivel_Ansiedad**).

3. Pregunta

- **Campos:**
 - **ID_Pregunta** (PK): Identificador único de la pregunta.
 - **Texto_Pregunta**: Texto de la pregunta.
- **Relaciones:**
 - Cada pregunta puede tener múltiples respuestas (relación 1 con **Respuesta**).

4. Respuesta

- **Campos:**
 - **ID_Respuesta** (PK): Identificador único de la respuesta.
 - **ID_Test** (FK): Identificador del test de ansiedad asociado.
 - **ID_Pregunta** (FK): Identificador de la pregunta asociada.
 - **Texto_Respuesta**: Texto de la respuesta.
- **Relaciones:**
 - Cada respuesta está asociada a un test de ansiedad (relación N:1 con **Test_Ansiedad**).
 - Cada respuesta está asociada a una pregunta (relación N:1 con **Pregunta**).

5. Tutorial

- **Campos:**
 - **ID_Tutorial** (PK): Identificador único del tutorial.
 - **Título**: Título del tutorial.
 - **Descripción**: Descripción del contenido del tutorial.
 - **Enlace_Video**: Enlace al video del tutorial.
 - **Fecha_Publicación**: Fecha de publicación del tutorial.
 - **ID_Nivel** (FK): Identificador del nivel de ansiedad para el cual el tutorial es adecuado.
 - **ID_Usuario** (FK): Identificador del usuario que creó el tutorial.
- **Relaciones:**
 - Cada tutorial está asociado a un nivel de ansiedad (relación N:1 con **Nivel_Ansiedad**).
 - Cada tutorial está asociado a un usuario (relación N:1 con **Usuario**).

6. Nivel_Ansiedad

- **Campos:**

- **ID_Nivel** (PK): Identificador único del nivel de ansiedad.
- **Descripción_Nivel**: Descripción del nivel de ansiedad (Bajo, Medio, Alto).
- **Relaciones:**
 - Cada nivel de ansiedad puede estar asociado a múltiples tests de ansiedad (relación 1 con **Test_Ansiedad**).
 - Cada nivel de ansiedad puede estar asociado a múltiples tutoriales (relación 1 con **Tutorial**).

Descripción General de la Base de Datos:

La base de datos de MindPeace está diseñada para gestionar de manera eficiente la información relacionada con los usuarios, los tests de ansiedad, las preguntas y respuestas asociadas a estos tests, y los tutoriales recomendados según los niveles de ansiedad. Cada usuario puede realizar múltiples tests de ansiedad, y cada test puede generar múltiples respuestas a preguntas específicas. Además, los tutoriales están categorizados según los niveles de ansiedad, permitiendo a los usuarios acceder a recursos específicos basados en sus necesidades.

Las relaciones entre las tablas están cuidadosamente estructuradas para evitar redundancias y asegurar la integridad referencial, facilitando así la mantenibilidad y escalabilidad del sistema. Este diseño asegura que los datos sean consistentes y que las operaciones sobre la base de datos sean eficientes.

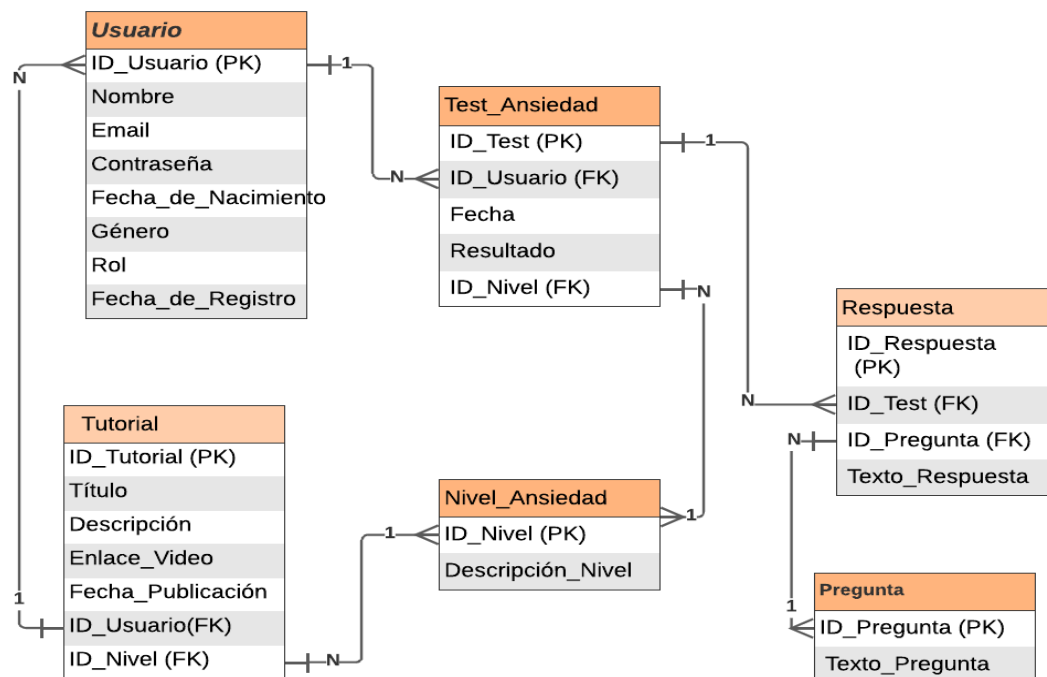


Diagrama 7. Base de datos