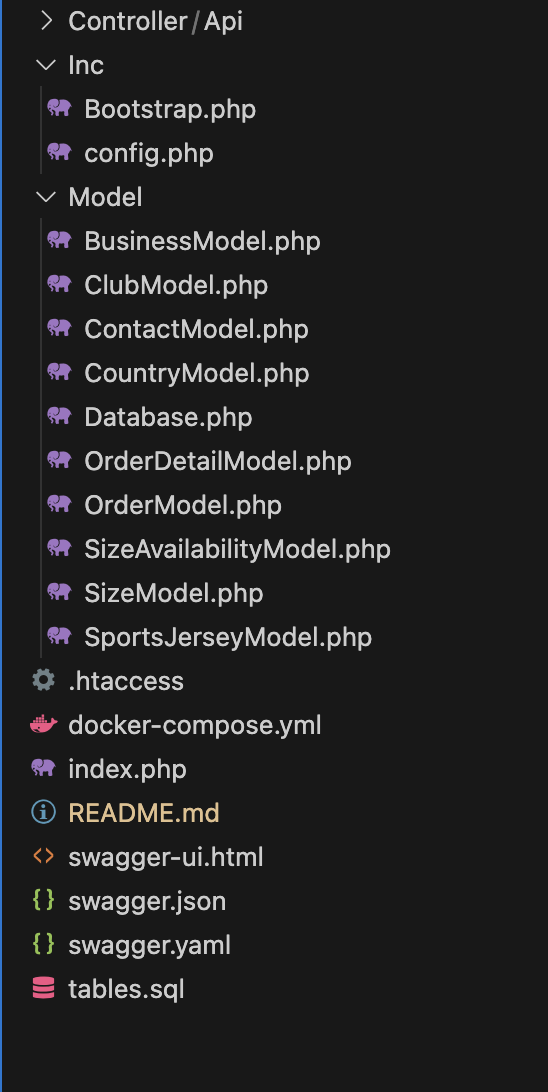
Informe Backend

**Tarea 1 -** Identifica los componentes principales de una arquitectura de backend y su función en

el desarrollo de una aplicación de software.

1. Servidor: Es el ordenador que almacena el código fuente con las reglas del negocio, cada cliente que envía solicitudes son respondidas por este.
2. API (Application Programming Interface): Corresponde a un conjunto de reglas y protocolos que permiten establecer comunicación entre distintas aplicaciones de software.
3. Lógica de aplicación: define la estructura de pensamiento y orden en que se ejecutan las ordenes hacia el ordenador para resolver un problema concreto.
4. Base de datos: Colección organizada de información almacenada de forma electrónica.

Diseño de la arquitectura de archivos. Trace un diagrama de alto nivel (puede ser un screenshot) que muestre la estructura de directorios completa del proyecto (Vistas, controladores, rutas, modelos, etc) y en un breve documento (o mediante comentarios) explique cómo encaja cada componente en el proyecto.



Explicación de componentes del proyecto

`Controller/Api/`: Define los endpoints de la API (ej: `PUT /sports-jersey/:id`). Llama a los modelos para procesar la lógica.

`Model/`: Acceso y manipulación directa de la base de datos. Cada archivo representa una entidad (Club, Order, Jersey, etc).

`Inc/config.php`: Contiene configuraciones globales, como credenciales y constantes de entorno.

`Inc/Bootstrap.php`: Inicializa el proyecto: autocarga de clases, configuración de rutas, etc.

`index.php`: Punto de entrada principal. Actúa como enrutador básico, interpretando la URL y el método HTTP.

`.htaccess`: Archivo de configuración para Apache. Permite URLs limpias y reescritura hacia `index.php`.

`swagger.yaml / swagger.json`: Definiciones OpenAPI para documentar la API. Describe rutas, parámetros y respuestas.

`swagger-ui.html`: Interfaz gráfica que permite probar la API desde el navegador usando Swagger.

`tables.sql`: Script para crear todas las tablas de la base de datos con relaciones e inserciones iniciales.

`docker-compose.yml`: Define los contenedores de desarrollo (ej: PHP + MySQL), útil para levantar el entorno completo.

`README.md`: Documentación principal del proyecto, instrucciones de instalación y uso.

Diseña un modelo de datos sencillo que permita almacenar los datos cubriendo la relación entre ambas entidades.

