**Arquitectura de Tres Capas**

Esta arquitectura separa la aplicación en tres capas:

* **Capa de Presentación (Frontend)**:

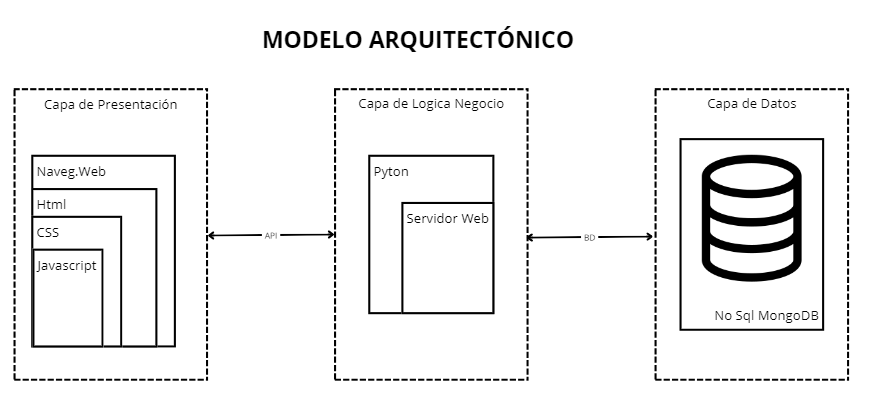
Es la interfaz de usuario diseñada en HTML, CSS y JavaScript. Aquí se maneja toda la lógica de interacción con el usuario, como capturar el nombre de jugador, mostrar las categorías y arrastrar y soltar palabras, también incluye la lógica para cambiar entre pantallas y mostrar los resultados.

* **Capa de Lógica de Negocio (Backend)**:

En una versión más avanzada, se agrega un servidor en Python, donde se maneja la lógica más compleja del juego, como el almacenamiento de puntajes, la validación de los datos y el seguimiento del tiempo de cada partida.

* **Capa de Datos (Database)**:

Aquí se guardará información sobre los jugadores, el tiempo de juego, o estadísticas de partidas, mediante una base de datos NoSql con MongoDB.



**Listado de herramientas**

* **HTML5** = Para estructurar el contenido de la interfaz de usuario.
* **CSS3** = Para el diseño y estilo visual de la interfaz.
* **JavaScript** = Para la lógica de interacción con el usuario, animaciones y efectos de arrastrar y soltar.
* **Python** = Lenguaje de programación para la lógica del servidor.
* **Django** = Frameworks de desarrollo web para gestionar las solicitudes HTTP y las rutas en el servidor.
* **API** **REST** = Para definir endpoints que permitan el intercambio de datos entre el cliente y el servidor.
* **MongoDB** = Base de datos NoSQL para almacenar datos de jugadores, estadísticas de partidas y tiempos de juego.
* **PyMongo** = Librería de Python para interactuar con MongoDB.

**Herramientas de desarrollo**

* **IDE** = Visual Studio Code
* **Git** = Para control de versiones y colaboración en el proyecto.
* **GitHub** = Repositorio remoto para almacenar y colaborar en el código fuente.
* **Postman** = Para probar los endpoints del API.