**Desarrollo de aplicaciones Web**

**Proyecto final: Segunda entrega**

El proyecto consiste en desarrollar una versión web del juego de dibujo y adivinanza Pinturillo. Pinturillo es un juego en el que los jugadores deben dibujar una palabra y los demás intentan adivinar mientras se muestra el dibujo en tiempo real. Objetivos Desarrollar una aplicación web interactiva que permita a los usuarios jugar al juego Pinturillo. Implementar funcionalidades clave del juego, como dibujar en tiempo real, adivinar palabras, llevar un registro de puntos y salas de juego. Utilizar tecnologías modernas de desarrollo web para crear una experiencia fluida y atractiva para los usuarios.

Objetivo: Desarrollar el backend para el juego estilo Pinturillo.

**Tablas de base de datos:**

**Palabra**

| **Campo** | **Descripción** |
| --- | --- |
| IdPalabra | Identificador único de la palabra. |
| Texto | El texto de la palabra en sí. |

**Categoría**

| **Campo** | **Descripción** |
| --- | --- |
| IdCategoria | Identificador único de la categoría. |
| Nombre | Nombre de la categoría. |

**PalabrasPorCategoria (Tabla de unión para la relación muchos-a-muchos)**

| **Campo** | **Descripción** |
| --- | --- |
| IdPalabra | Identificador de la palabra. |
| IdCategoria | Identificador de la categoría. |

**Sala de Juego**

| **Campo** | **Descripción** |
| --- | --- |
| IdSala | Identificador único de la sala de juego. |
| Nombre | Nombre opcional de la sala de juego. |
| IdCategoria | Identificador de la categoría utilizada en la sala (suponiendo que cada sala se base en una categoría específica para seleccionar palabras). |
| Estado | Estado de la sala de juego: Sin iniciar, en curso, finalizado |

**Requisitos del Proyecto:**

1. Crear la base de datos y el modelo entidad relación teniendo en cuenta las tablas indicadas anteriormente, si necesitan agregar entidades o campos pueden hacerlo.
2. Implementar los siguientes casos de uso en el backend:

| **Caso de Uso** | **Descripción** | **Precondiciones** | **Postcondiciones** |
| --- | --- | --- | --- |
| **Agregar Nueva Palabra** | Permite a un usuario agregar una nueva palabra al sistema. | La palabra no debe existir | La palabra se agrega a la base de datos. |
| **Editar Palabra** | Permite editar el texto de una palabra existente. | La palabra debe existir | Los cambios se reflejan en la base de datos. |
| **Eliminar Palabra** | Permite eliminar una palabra del sistema. | La palabra debe existir y | La palabra se elimina de la base de datos. |
| **Agregar Nueva Categoría** | Permite agregar una nueva categoría al sistema. |  | La categoría se agrega a la base de datos. |
| **Editar Categoría** | Permite editar el nombre de una categoría existente. | La categoría debe existir y | Los cambios se reflejan en la base de datos. |
| **Eliminar Categoría** | Permite eliminar una categoría del sistema. | La categoría debe existir | La categoría se elimina de la base de datos. |
| **Asociar palabras a categoría** | Permite asociar una o más palabras a una categoría | La categoría y las palabras deben existir | Los cambios se reflejan en base de datos |
| **Desvincular palabras de categoría** | Permite desvincular una palabra de una categoría | La categoría y las palabras deben existir | Los cambios se reflejan en base de datos |
| **Crear Sala de Juego** | Permite crear una nueva sala de juego. |  | La sala se crea y está lista para ser utilizada. |
| **Eliminar Sala de Juego** | Permite eliminar una sala de juego. | La sala debe existir | La sala se elimina de la base de datos. |
| **Ver Salas de Juego** | Permite ver una lista de todas las salas de juego disponibles, Añadir filtro por estado y categoría | Ninguna. | El sistema muestra una lista de salas de juego. |
| **Obtener las palabras asignadas a una sala de juego** | Permite ver una lista de todas las palabras asignadas a una sala de juego según su categoría | La sala debe existir | Ninguna |
| **Enviar mensaje por sala de juego** | Permite enviar mensajes por medio de sockets por cada sala de juego | La sala debe existir | El jugador envía el mensaje y le notifica a los demás jugadores de la sala |
| **Unirse a una sala de juego** | Permite a un jugador unirse a una sala de juego existente, indicando su nombre y su avatar (utilizando sockets) | La sala debe existir | El jugador se une y le envía la notificación a los demás jugadores de que se ha unido |
| **Asignar turno de juego** | El sistema debe controlar e indicar qué jugador está en juego, así como qué palabra está siendo utilizada | La sala debe existir | Se le notifica al jugador que está en juego cual es la palabra que debe dibujar y a los demás jugadores se les notifica que dicho jugador está dibujando; |
| **Adivinar palabra** | El sistema debe reconocer cuando un jugador adivinó correctamente la palabra y asignar un puntaje a dicho jugador, este puntaje se calcula teniendo en cuenta el tiempo restante y el orden en el que los jugadores van adivinando | Un jugador debe estar en turno | Se le notifica a los jugadores que el usuario adivinó la palabra |
| **Finalizar turno** | El sistema debe reconocer cuando todos los jugadores han adivinado y asignar el turno al siguiente jugador | Un jugador debe estar en turno | Se le notifica a los jugadores que el turno finalizó |
| **Finalizar juego** | El sistema debe reconocer cuando han transcurrido n rondas y de esta forma mostrar la tabla de resultados | Deben haber transcurrido n rondas | Se notifica a los jugadores que el juego ha finalizado y muestra los resultados |

Se valorarán buenas prácticas de desarrollo, que se utilice la estructura acordada en clase para el proyecto y un manejo adecuado de los errores y corner cases, así como un adecuado manejo del repositorio de github.

**La fecha de entrega del proyecto será el 30 de mayo de 2024 de forma presencial.**