

# Proyecto Gryffindor - Actividades

## Previas al desarrollo

- ☒ ~~Configurar 2 aprobaciones para hacer merge de PRs en los repositorios de front y back.~~
- ☒ ~~Habilitar protección de la rama `main` para impedir commits directos y forzar PRs.~~
- ☒ ~~Decidir la convención de commits a utilizar (ejemplo: Conventional Commits).~~
- ☒ ~~Instalar gestor de hooks de git para implementar la convención de commits (ejemplo: Husky) [Óscar]~~
- ☒ ~~Definir reglas de linting y formateo (ejemplos: ESLint, Prettier) [Óscar]~~
- ☐ Instalar Radix [Maxi]
- ☒ ~~Definir Herramientas adicionales además de React:~~
  - Vite.
  - pnpm
  - HTML
  - CSS
  - SCSS
- ☐ Instalar SCSS [Alejandro]
- ☒ ~~Configurar bases del proyecto [Óscar]~~
- ☐ Crear README [Pendiente]
- ☐ Crear un diseño básico en Figma (Front) [Ficho]
- ☐ Definir estilo (paleta de colores, íconos, fuentes...) [Alejandro]

## Historia de Usuario 1: Configuración de usuario temporal

### Descripción:

Como jugador, quiero ingresar mi nombre y elegir mi casa de Hogwarts para poder participar en los duelos mágicos.

 Definición de Ready (DoR):

- ☒ Se ha definido la lista de casas de Hogwarts disponibles.
- ☒ Se ha decidido cómo se almacenará temporalmente la información del jugador.

 Criterios de aceptación:

- ☒ El jugador puede ingresar un nombre de usuario.
- ☒ Puede elegir una de las cuatro casas de Hogwarts.
- ☒ La información se mantiene durante la sesión, pero no se guarda en una base de datos permanente.

 Definición de Done (DoD):

- ☒ Código revisado y aprobado por otro desarrollador.
- ☒ Probado en diferentes dispositivos.

- ✅ Integrado con el flujo del juego (el usuario no puede participar sin configurarse).

## Tareas

### Front

- Crear formulario para que el jugador ingrese su nombre (+ validación de datos).(Tarjeta “Pantalla de Inicio” en trello)
- Crear un componente visual para que el jugador seleccione su casa de Hogwarts (puede ser un dropdown o botones). Las opciones deben ser: Gryffindor, Hufflepuff, Ravenclaw y Slytherin..(Tarjeta “componente seleccion casa” en trello)
- Guardar temporalmente el nombre y la casa en el estado de la aplicación. (Tarjeta “Pantalla de Inicio” en trello)
- Mostrar un botón para confirmar la selección y continuar. (Tarjeta “componente seleccion casa” en trello)

## Historia de Usuario 2: Selección de hechizos en el duelo

### Descripción:

combate.

Como jugador, quiero seleccionar un hechizo en mi turno para poder participar en el

📌 Definición de Ready (DoR):

- ✅ La lista de hechizos disponibles ha sido definida.
- ✅ Se han definido reglas de combate sobre qué hechizos pueden contrarrestarse entre sí.

📌 Criterios de aceptación:

- ✅ Se muestra una lista de hechizos disponibles en la UI.
- ✅ Al seleccionar un hechizo, éste se envía al backend para su procesamiento.✅ Una vez seleccionado, el hechizo no puede cambiarse hasta el próximo duelo.
- ✅ Debe haber un indicador visual de selección confirmada.

📌 Definición de Done (DoD):

- ✅ Código revisado y aprobado por otro desarrollador.
- ✅ Probado con diferentes hechizos.
- ✅ Integrado con el backend para la lógica de duelo.

## Tareas

- Definir lista de hechizos (si alguien conoce algunos que sean icónicos de las pelis, mejor).(Tarjeta “Selección de hechizos” en trello)
- Definir reglas de combate sobre qué hechizos pueden contrarrestarse entre sí.

- Definir acuerdo de interfaz: ¿Qué datos le enviaremos al back? (# de sala, identificador de los jugadores, hechizos...).

## Front

- Crear el componente visual que muestre la lista de hechizos disponibles. (Tarjeta “Selección de hechizos” en trello)
- Implementar la selección de un hechizo con confirmación visual.(Tarjeta “Selección de hechizos” en trello)
- Implementar el envío de la información al backend. (Tarjeta “Selección de hechizos” en trello)
- Desactivar el componente de selección tras la confirmación del hechizo.
- Mostrar retroalimentación visual cuando el jugador confirme su selección (ejemplo: resaltado del hechizo elegido). (Tarjeta “Selección de hechizos” en trello)
- Manejar estados de error si falla el envío del hechizo al backend.
- Sincronizar la UI con el estado del duelo (ejemplo: deshabilitar selección si ya es el turno del rival).
- Mostrar cronómetro con tiempo que tiene el usuario para elegir su hechizo (30 segundos) (Tarjeta “Selección de hechizos” en trello)
- Mostrar feedback si el rival ya ha seleccionado su hechizo (opcional)

## Back

# Historia de Usuario 3: Procesamiento del duelo mágico

## Descripción:

Como sistema, quiero procesar las elecciones de hechizos de ambos jugadores y determinar el resultado del duelo.

 Definición de Ready (DoR):

- ✓ Se ha definido la lógica de combate entre hechizos.
- ✓ Se ha acordado cómo se enviarán y recibirán los datos del duelo.

 Criterios de aceptación:

- ✓ Se comparan los hechizos de los jugadores según reglas de combate predefinidas.
- ✓ Si un hechizo es más fuerte que el otro, se determina un ganador.
- ✓ Si ambos hechizos son equivalentes, se declara un empate.
- ✓ El resultado debe enviarse al frontend en menos de 2 segundos.

 Definición de Done (DoD):

- ✓ Código revisado y probado con múltiples combinaciones de hechizos.
- ✓ Integrado con el frontend para mostrar el resultado en la UI.

## Tareas

### Front

- Recibir y procesar el resultado del duelo enviado por el backend.
- Mostrar el resultado en la UI con feedback visual claro (ejemplo: animaciones o cambios de color).
- Deshabilitar la selección de hechizos mientras se muestra el resultado.
- Añadir una transición visual (ejemplo: pantalla intermedia con “Procesando resultado...”). O cuando los dos usuarios eligen sus hechizos, cambiar contador (3 segundos) para mostrar resultado del duelo
- Después de mostrar el resultado permitir volver a la pantalla de inicio, iniciar un nuevo duelo o revancha (si ambos jugadores quieren revancha, sino dar feedback al jugador que quería revancha).

### Back