

Documentación del Proyecto Quinxo

Grupo 7

2025-12-09

Integrantes del Grupo

- Adrián Cordero - Carné: FI21022475 - Git: Adrianca25 - Correo: acordero10422@ufide.ac.cr
- Johannes Sequeira - Carné: FI22023682 - Git: JohannesSeq - Correo: jsequeira40132@ufide.ac.cr
- Kyran Jose Vilchez Barrantes - Carné: FI17008273 - Git: Kvilchezb - Correo: kyran09.97@gmail.com
- Jose Daniel Araya Arias - Carné: FI22023911 - Git: pgn3 - Correo: jd2002araya@gmail.com

Frameworks y Herramientas

- Frameworks: ASP.NET Core MVC, Entity Framework Core, Bootstrap
- Herramientas: SQLite, Visual Studio, GitHub

Tipo de Aplicación y Arquitectura

- Tipo de aplicación: MPA (Multi-Page Application)
- Arquitectura: MVC (Model-View-Controller)

Diagrama de Base de Datos

Referencias y Prompts IA

- Documentación oficial: ASP.NET Core, Entity Framework y Bootstrap.
- Prompts IA: Conversaciones con ChatGPT para revisión de código, generación de README.tex y sugerencias de mejoras.
- **Error 1:** Problema al actualizar el texto de la etiqueta turnLabel en JavaScript.

- Mensaje generado:

```
game.js:297 Uncaught TypeError: Cannot set properties of null
(setting 'textContent')
    at updateTurnLabel (game.js:297:23)
    at HTMLDocument.<anonymous> (game.js:289:5)
```

- **Causa:** El JavaScript ejecutaba:

```
document.getElementById("turnLabel").textContent = ...
```

pero el elemento con id `turnLabel` no existía en la vista, por lo que `document.getElementById("turnLabel")` devolvía `null` y acceder a `null.textContent` provocaba el error.

- **Solución:** Añadir el siguiente elemento en `Play.cshtml` justo antes del tablero:

```
<h3 id="turnLabel" class="mt-3 mb-3 text-primary" style="font-weight:bold;">
    Turno actual:
</h3>
```

- Ejemplo completo:

```
<div class="container">
    <h2>Partida #@Model.Id</h2>

    <h3 id="turnLabel" class="mt-3 mb-3 text-primary" style="font-weight:bold;">
        Turno actual:
    </h3>

    <div id="gameTimer" class="timer">00:00:00</div>

    <!-- TABLERO -->
    <div id="boardContainer">
        @* Aquí se renderiza el tablero *@
    </div>

    <button id="resetGame" class="btn btn-danger mt-3">Reiniciar</button>
</div>
```

- Resultado: El label se actualiza correctamente tras agregar el elemento en la vista.

- **Error 2:** Referencia a variable inexistente `currentPlayerIndex`.

- Mensaje generado:

```
game.js:294 Uncaught ReferenceError: currentPlayerIndex is not defined
```

- **Diagnóstico:**

- El código original dentro de `updateTurnLabel()` estaba usando `currentPlayerIndex`.
- En el resto del script no existe ninguna variable llamada `currentPlayerIndex`; en su lugar existe `currentTurnIndex` (por ejemplo: `let currentTurnIndex = 0;`).

- **Código con el error (buscar en el archivo):**

```
function updateTurnLabel() {
    const player = PLAYERS[currentPlayerIndex];
    const label = document.getElementById("turnLabel");

    label.textContent = Turno actual: ${player.name}
    (${player.team});
}
```

- **Corrección exacta:** reemplazar `currentPlayerIndex` por `currentTurnIndex`:

```
function updateTurnLabel() {
    const player = PLAYERS[currentTurnIndex];
    const label = document.getElementById("turnLabel");

    label.textContent = `Turno actual: ${player.name}
    (${player.team})`;
}
```

- **Nota importante:**

- Verificá que la variable `currentTurnIndex` esté declarada y actualizada en el scope correcto, por ejemplo:

```
let currentTurnIndex = 0;  
let currentPlayer = null;
```

- Después de la corrección, probá la partida para confirmar que el label muestra el jugador correcto en cada turno.

- **Error 3:**

- `Uncaught ReferenceError: PLAYERS is not defined`
- **Solución:** Este error indica que las variables `PLAYERS`, `MODE` y `GAME_ID` no estaban definidas dentro de `game.js`. Se corrigió asegurando que estas variables fueran enviadas desde la vista mediante un bloque `<script>` que las expone antes de cargar el archivo JavaScript.

Instructivo

Requisitos Previos

- Instalar .NET SDK (versión 6.0 o superior): <https://dotnet.microsoft.com/download>
- Instalar SQLite (incluido en el proyecto, no requiere configuración adicional)
- Instalar Entity Framework Core Tools:

```
dotnet tool install --global dotnet-ef --version 8.0.0
```

Instalación de Dependencias

Ejecutar en la carpeta del proyecto `QuinxoWebApp`:

```
cd QuinxoWebApp  
dotnet restore
```

Compilación

Compilar el proyecto:

```
dotnet build
```

Migraciones y Base de Datos

Si necesita aplicar migraciones:

```
dotnet ef database update
```

Ejecución

Ejecutar la aplicación:

```
dotnet run
```

Luego abrir en el navegador:

```
http://localhost:5212
```

Notas importantes

- Si ejecuta desde la raíz del repositorio, use:

```
dotnet run --project QuinxoWebApp/QuinxoWebApp.csproj
```

- Para detener la aplicación, presione Ctrl + C en la consola.