Documentación del Proyecto: Quixo Starter (ASP.NET Core 8 + React Vite)

Integrante	**Carné / Usuario Git / Correo**
Ricardo Patiño Jiménez	FH22011118 / Ricardo-Patino / rickpatinor@gmail.com
Isaac Arias Morera	FI23028657 / IsaacAriasMore / jarias30680@ufide.ac.cr
Alex Monge Arias	FH23014026 / ALE20201 / amonge50242@ufide.ac.cr
Brandon Céspedes	FH22012992 / Bcespedes04 / bcespedes@traarepuestos.com

14 de octubre de 2025

Índice

1. Integrantes del Grupo

- Ricardo Patiño Jiménez — FH22011118 — Usuario Git: Ricardo-Patino — Correo: rickpatinor@gmail.com
- Isaac Arias Morera — FI23028657 — Usuario Git: IsaacAriasMore — Correo: jarias30680@ufide.ac.cr
- Alex Monge Arias — FH23014026 — Usuario Git: ALE20201 — Correo: amonge50242@ufide.ac.cr
- Brandon Céspedes — FH22012992 — Usuario Git: Bcespedes04 — Correo: bcespedes@traarepuestos.com

2. Descripción General del Proyecto

El proyecto **Quixo Starter** es una aplicación web desarrollada utilizando **ASP.NET Core 8 (Minimal API)** en el backend y **React + Vite** en el frontend. Su objetivo es implementar el juego Quixo con una arquitectura moderna basada en separación de responsabilidades, conectividad API REST y despliegue ágil.

3. Estructura del Proyecto

```
QuixoSolution/
  backend/Quixo.Api/        # ASP.NET Core 8 Minimal API + Dapper
  frontend/quixo-web/       # React + Vite SPA
  db/Scripts/quixo_mysql.sql
```

4. Frameworks y Herramientas Utilizadas

- **Backend:** ASP.NET Core 8, Dapper
- **Frontend:** React + Vite
- **Base de datos:** MySQL 8+
- **Entornos de desarrollo:** Visual Studio 2022, VS Code

- **Control de versiones:** Git + GitHub

5. Tipo de Aplicación

El proyecto implementa una **SPA (Single Page Application)**, en la cual el frontend React se comunica con el backend mediante peticiones HTTP a la API REST.

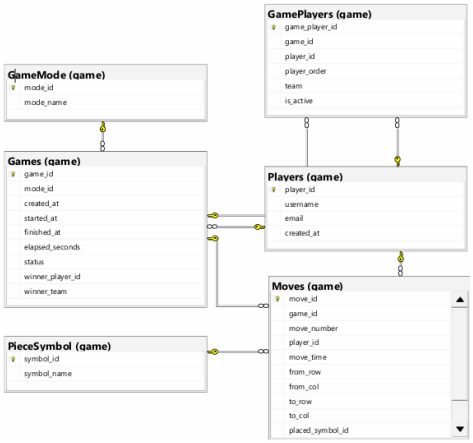
6. Arquitectura Utilizada

Se utiliza una arquitectura **cliente-servidor** basada en **MVC (Modelo–Vista–Controlador)**:

- **Modelo:** Manejado mediante Dapper y las entidades de base de datos en C#.
- **Controlador:** Endpoints Minimal API en ASP.NET Core.
- **Vista:** Componentes de React (frontend).

7. Diagrama de Base de Datos

La base de datos utilizada es **MySQL**, relacional, y consta de tablas para jugadores, partidas y movimientos.



8. Referencias y Prompts de IA

Fuentes Consultadas

- Documentación oficial de .NET 8: `https://learn.microsoft.com/en-us/aspnet/core`
- Documentación de React: `https://react.dev/`
- Tutorial de Vite: `https://vitejs.dev/guide/`
- Manual de Dapper ORM: `https://dapper-tutorial.net/`
- Guía de MySQL 8: `https://dev.mysql.com/doc/`
- Introducción a LaTeX: `https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`

Prompts de Inteligencia Artificial

- **Prompt de entrada:** "Genera un ejemplo de documentación en LaTeX, Además, explica cómo funciona cada sección dentro de la documentación y cuáles son los beneficios de estructurarla de esta manera."
- **Respuesta del agente de IA:** "Se sugiere estructurar la documentación en secciones: Integrantes, Frameworks, Arquitectura, Diagrama, Referencias y Guía de ejecución."

9. Instructivo de Instalación y Ejecución

Requisitos Previos

- MySQL 8+ con base de datos `quixo` creada.
- .NET SDK 8 instalado (Visual Studio 2022 actualizado).
- Node.js versión 18 o superior.

Backend (Visual Studio 2022)

1. Abrir el proyecto `Quixo.Api.csproj` desde Visual Studio 2022.
2. Editar `appsettings.json` con las credenciales de MySQL.
3. Ejecutar el proyecto (F5). La API expone Swagger en `/swagger`.
4. Nota: Si el puerto cambia (por ejemplo de 5199), actualizar la variable `VITE_API_URL` en el frontend.

Frontend (VS Code o similar)

```
cd frontend/quixo-web
npm i
# Si el backend corre en otro puerto/host:
# set VITE_API_URL=http://localhost:5199
npm run dev
```

Abrir en el navegador: `http://localhost:5173`

Datos de Ejemplo

Agregar en la tabla `players` al menos 4 jugadores con IDs del 1 al 4 o ajustar el archivo `NewGame.jsx` según sea necesario.

Exportar XML y Estadísticas

- Exportar partida: `GET /api/games/{id}/export.xml`
- Estadísticas por tipo: `GET /api/stats/duo` y `GET /api/stats/quartet`