

FIAP Cloud Games (FCG) MVP

Introdução

- **Objetivo:**

Nessa primeira etapa do nosso projeto o objetivo é criar uma API RESTful para o cadastro de usuários, cadastro de jogos e todo o gerenciamento que envolve um usuário ter uma biblioteca de jogos.

- **Tecnologias utilizadas:**

.NET 8, Web API, Entity Framework Core, JWT, MySQL, Swagger, xUnit, Moq, Serilog.

Arquitetura

- **Padrão usado:** Clean Architecture + DDD (Domain-Driven Design)

- **Camadas**

- Domain:** Entidades, Enum, Value Objects

- Application:** Casos de uso (Handlers, Requests, Responses)

- Infra:** Repositórios e Contexto EF Core

- API:** Controllers, Middleware, Configuração

Funcionalidades Implementadas

Usuários:

Criar, atualizar, deletar, buscar por ID e email

Login com geração de token JWT

Promoção de usuário para Admin

Jogos:

CRUD de jogos (somente admins)

UserGames:

Comprar jogo, listar jogos do usuário, remover jogo

Autenticação/Autorização

JWT

Policies por Role (Admin / User)

Middleware de exceção global

Swagger UI com autenticação via Bearer Token

Clicar em autenticação e apenas colar o token depois de logar.

Log com Serilog

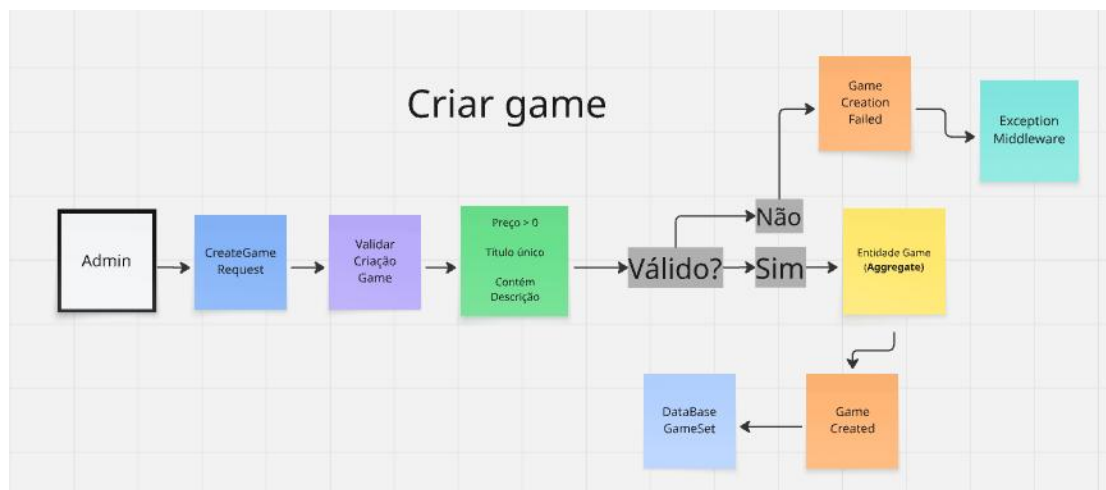
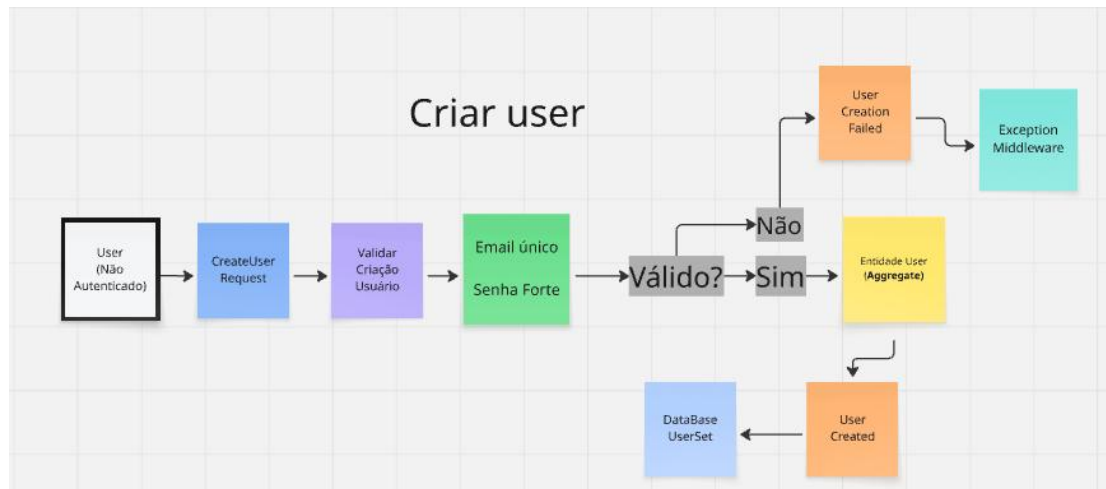
Implementado ainda em poucas funcionalidades para teste.

Endpoints

Método	Rota	Descrição	Requer Autenticação?	Perfil
POST	api/users	Cria um novo usuário	Não	Público
POST	api/users/login	Autentica e retorna token	Não	Público
POST	api/games	Cria novo jogo	Sim	Admin
POST	/api/Admin/{userId}/games/{gameId}	Adiciona um Jogo a uma conta	Sim	Admin
POST	/api/UserGames/games/{gameId}	Adiciona um jogo a conta autenticada	Sim	Ambos
GET	/api/Admin/{userId}/games	Lista jogos de um usuário	Sim	Admin
GET	/api/Games	Lista todos os jogos	Sim	Ambos
GET	/api/Games/{id}	Retorna informações do jogo	Sim	Ambos
GET	/api/User/Games/me/games	Mostra a biblioteca do user autenticado	Sim	Ambos
GET	/api/users	Mostra todos usuários	Sim	Admin
GET	/api/users/me	Mostra as informações do usuário autenticado	Sim	Ambos
GET	/api/users/{id}	Mostra informações do usuário buscado	Sim	Admin

Método	Rota	Descrição	Requer Autenticação?	Perfil
GET	/api/Users/email/{email}	Mostra informações do usuário buscado por email	Sim	Admin
PUT	/api/Admin/promote/{id}	Promover user para admin	Sim	Admin
PUT	/api/Admin/users/{id}	Alterar nome e password de usuário por id	Sim	Admin
PUT	/api/Games/{id}	Alterar dados de um jogo por id	Sim	Admin
PUT	/api/Users/me	Atualizar nome e password de usuário autenticado	Sim	Ambos
DELETE	/api/Admin/users/{userId}/games/{gameId}	Deletar um jogo de um usuario	Sim	Admin
DELETE	/api/Admin/{id}	Deletar um usuário	Sim	Admin
DELETE	/api/Games/{id}	Deleta um jogo	Sim	Admin
DELETE	/api/UserGames/games/{gameId}	Deletar jogo da biblioteca do usuário autenticado	Sim	Ambos

Event Storm



Testes Automatizados

Frameworks: xUnit + Moq

Testes implementados:

ValueObjects (Email, Password, Title, Price, Description)

Handlers dos use cases (CreateUser, LoginUser, etc.)

Para rodar os testes abra o terminal na FCG.Tests e de dotnet test

Execução do Projeto

Pré-requisitos:

.NET 8 SDK

MySQL

Passos:

Restaurar pacotes: dotnet restore

Configurar connection string:

"DefaultConnection":

"server=SEU_SERVER;port=SUA_PORTA;database=fcgdb;user=SEU_USER;password=SEU_PASS;"

Criar Migration: dotnet ef migrations "text"

Subir para o Db: dotnet ef database update

Rodar o projeto: dotnet run

Acessar: https://localhost:porta/swagger

Código Fonte:

<https://github.com/devjoaomelo/fiap-cloud-games>

Pontos a serem melhorados no curto prazo:

- Documentação descritiva no swagger
- Interface front-end para melhor visualização
- Adição de saldo para usuário para conseguir adquirir game
- Refatorar código principalmente seção de log/tratamento de erros
- Auditoria
- Melhorar segurança da role admin

Autoria:

João Vitor Gonçalves de Melo