

Relatório do Projeto – Apostas Euromilhões

1. Introdução

Esta aplicação foi desenvolvida para efetuar apostas online num jogo similar ao Euromilhões, integrando dois sistemas independentes: Sistema CrediBank e o Sistema EuroMilRegister, baseados em duas tecnologias distintas API RESTs e gRPCs respetivamente.

2. Objetivos

- Permitir ao utilizador inserir a chave de aposta e a identificação da conta de crédito.
- Obter um cheque digital junto do CrediBank para o valor necessário (10 créditos).
- Enviar a aposta juntamente com o cheque digital para o EuroMilRegister.
- Informar o utilizador sobre o êxito ou falha na operação.

3. Descrição da Implementação

A aplicação foi desenvolvida em **.NET (Windows Forms)** e utiliza chamadas assíncronas para integrar os serviços externos, recorrendo a duas tecnologias distintas: API REST e gRPC.

- O CrediBankAPI é um serviço baseado em API REST, responsável pela emissão dos cheques digitais. A comunicação com este sistema é realizada através de chamadas HTTP no formato RESTful, utilizando o método GET para solicitar um cheque digital de 10 créditos.
- O EuroMilRegister utiliza gRPC, um protocolo de comunicação de alto desempenho que permite a comunicação eficiente entre sistemas distribuídos. Neste projeto, a interação ocorre através do método RegisterEuroMil, que recebe a chave da aposta e o cheque digital como parâmetros.
- Em Program.cs, define-se o ponto de entrada da aplicação, configurando a interface gráfica e inicializando as funcionalidades.
- O projeto inclui os arquivos de configuração padrão do Visual Studio 2022 e as referências a pacotes como **Google.Protobuf**, **Grpc.Net.Client** e **Grpc.Tools** para comunicação gRPC.
- No arquivo **euroMilRegister.proto**, está definido o serviço gRPC EuroMil, que possui o método RegisterEuroMil, responsável por receber uma chave e um cheque digital para efetuar o registo da aposta.
- A classe **EuroMilRegisterClient.cs** implementa a lógica para estabelecer um canal gRPC (não seguro, via HTTP), cria a requisição e processa a resposta recebida do serviço EuroMilRegister.
- No **Form1.cs**, a interface recolhe os dados inseridos pelo utilizador e, em sequência, realiza as seguintes operações:
 - valida a chave da aposta, de acordo com as seguintes regras específicas, 5 números de 1 a 50 + 2 estrelas de 1 a 12 (X,X,X,X,X+Y,Y) e a conta do utilizador (8 dígitos);

- Solicita ao **CrediBank** a emissão de um cheque digital no valor de 10 créditos (através do CrediBankClient).
- Exibe os detalhes do cheque digital obtido e utiliza-o, juntamente com a chave, para registar a aposta via EuroMilRegisterClient.
- Trata e exibe mensagens de erro, caso ocorram exceções durante as operações.

Figura 1: Form1

5. Repositório Git

<https://github.com/lsantos1970/7.-Aplicacao-integradora-de-Sistemas>

6. Conclusão

A aplicação cumpre os requisitos propostos, integrando com sucesso os sistemas CrediBank e EuroMilRegister através de duas tecnologias distintas: API REST para a obtenção de cheques digitais e gRPC para o registo das apostas.

A utilização de chamadas assíncronas permite um fluxo de operações não bloqueante, melhorando a experiência do utilizador. No entanto, devido a problemas externos com o serviço gRPC, a ligação ao EuroMilRegister não pôde ser testada na prática.

7. Referências

Mapeamento entre Protobuf e .NET, <https://docs.microsoft.com/en-us/aspnet/core/grpc/protobuf?view=aspnetcore-3.1>

Call a Web API From a .NET Client (C#), <https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client>

Tutorial: Create a web API with ASP.NET Core, <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-5.0&tabs=visual-studio>

Tutorial: Make HTTP requests in a .NET console app using C#, <https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/console-webapiclient>