

Desafio - Back-End

Parabéns por sua participação no processo seletivo!

Neste teste, você terá a oportunidade de demonstrar suas habilidades como desenvolvedor(a) e sua capacidade de construir soluções robustas e escaláveis.

Imaginaremos um cenário real de nossa plataforma de apostas esportivas e cassino online, onde você será responsável por desenvolver uma carteira digital simplificada. Essa carteira permitirá que os jogadores realizem apostas, ganhem prêmios e cancelem suas apostas, mantendo um saldo atualizado a cada operação.

Objetivo: Avaliar seus conhecimentos em desenvolvimento Back-End, com foco em C# (.NET 6).

Requisitos Técnicos

Obrigatórios

Os endpoints da API deverão funcionar via Swagger

- **API RESTful:** Crie uma API RESTful utilizando C# (.NET 6) para gerenciar as transações da carteira. Os endpoints devem permitir:
 - **Criar um novo jogador:** criar novo jogador, com uma carteira com saldo inicial e atrelado a uma moeda (R\$ - BRL).
 - **Login do jogador:** fazer o login e retornar informações do jogador e do saldo de sua carteira;
 - **Realizar uma transação de aposta:** deve decrementar o saldo da carteira ao receber a aposta;
 - **Consultar minhas apostas:** deve permitir consultar as apostas do jogador. Deve retornar no mínimo o ID, valor, situação, valor do prêmio (caso aposta ganha) e data/hora da aposta;
 - **Cancelar aposta:** deve permitir cancelar uma aposta do jogador;
 - **Consultar minhas transações:** deve retornar informações das transações realizadas na carteira do jogador, como: ID, Data/Hora, Tipo, Valor;
- **Banco de dados:** Utilize um banco de dados relacional (como MySQL, SQL Server, PostgreSQL ou outro de sua preferência) para armazenar as informações dos jogadores, carteiras e as transações.

- Exportar Modelo Entidade Relacional (MER) e adicionar como recurso dentro do projeto;
- **Validação de dados:** Implemente validações para garantir a integridade dos dados, como por exemplo:
 - E-mail do jogador deve ser único;
 - Verificar se o valor da aposta é válido (maior que zero).
 - Verificar se o valor da aposta é maior que o mínimo (R\$ 1,00).
 - Verificar se o saldo da carteira é suficiente para realizar a aposta ou cancelamento.
 - Não permitir cancelamento de apostas já canceladas;
- **Regras de negócio:** Implemente as regras a seguir:
 - Ao receber a transação de aposta elabore uma lógica em que exista uma chance do jogador ganhar essa aposta. Ao ganhar a aposta gerar uma transação de prêmio do dobro do valor da aposta realizada;
 - Após cada vez que o jogador acumular 5 apostas perdidas, então o jogador deverá receber um bonus de 10% do que foi gasto nessas apostas;
 - Criar estrutura de paginação para os endpoints de **Consultar minhas apostas** e **Consultar minhas transações**. Receber nesses endpoints a página específica e o limite de registros requeridos.

Desejáveis (Não obrigatórios)

- **Testes unitários:** Implementar testes básicos para assegurar as regras de negócio e validações dos dados;
- **Websockets:** Implementar atualização em real-time do saldo da carteira do jogador;
- **Docker do banco de dados:** Configurar ambiente/container Docker para o banco de dados escolhido;
- **Controle de migrations:** Configurar método de controle de migrações de banco de dados e o fluxo para execução;

Entrega

- **Código fonte:** Envie o código fonte completo do seu projeto em um repositório público no GitHub, com o último commit respeitando a data máxima de entrega do desafio.
- **Documentação:** Inclua um README com instruções de como executar o projeto e qualquer outra informação relevante para que seja possível executá-lo e testá-lo na integralidade.

- **Demonstração:** Se possível, grave um vídeo demonstrando o funcionamento da sua aplicação passando por todas as funcionalidades descritas acima.

Lembre-se: *Este é um desafio para avaliar suas habilidades técnicas e criativas. Sinta-se à vontade para adicionar funcionalidades extras que você considerar relevantes, elas serão levadas em conta se existirem. Boa sorte!*