

# Projeto C++: Quiz Dinâmico

## Visão Geral

Este projeto é um jogo de quiz interativo desenvolvido em C++ que permite aos usuários testar seus conhecimentos em diversas categorias e níveis de dificuldade. O diferencial do projeto reside na sua capacidade de dinamismo, onde as perguntas são carregadas de arquivos externos, e os dados dos jogadores (perfis e pontuações) são persistidos, permitindo uma experiência personalizada e a manutenção de rankings.

## Requisitos Funcionais

O sistema deverá permitir:

### 1. Criação/Seleção de Usuário:

- a. O usuário poderá digitar um nome de usuário para iniciar uma sessão.
- b. Se o usuário já existir (verificado em um arquivo de perfis), seus dados (histórico, pontuações) serão carregados.
- c. Se for um novo usuário, um novo perfil será criado e persistido.

### 2. Seleção de Categoria e Dificuldade:

- a. O usuário poderá escolher entre diferentes categorias de perguntas (ex: "História", "Ciência", "Esportes"). As categorias serão definidas por arquivos de perguntas separados.
- b. Dentro de cada categoria, o usuário poderá selecionar um nível de dificuldade (ex: "Fácil", "Médio", "Difícil").

### 3. Jogo de Quiz:

- a. Apresentar perguntas ao usuário, lidas dos arquivos correspondentes à categoria e dificuldade selecionadas.
- b. As perguntas poderão ser de múltipla escolha ou abertas (definir o formato no arquivo).
- c. O programa verificará a resposta do usuário e informará se está correta ou incorreta.
- d. Pontuação será acumulada com base nas respostas corretas e, opcionalmente, na dificuldade da pergunta.
- e. Ao final do quiz (após um número pré-definido de perguntas ou até o usuário desistir), a pontuação final será exibida.

### 4. Atualização e Persistência de Dados do Usuário:

- a. A pontuação e o histórico de jogo de cada usuário (número de jogos, vitórias, pontuação total por categoria, etc.) serão salvos e atualizados em seus respectivos perfis ao final de cada sessão de jogo.
5. **Exibição de Ranking:**
  - a. O sistema exibirá um ranking dos melhores pontuadores (geral ou por categoria/dificuldade), lidos do arquivo de ranking.
  - b. O ranking será atualizado automaticamente caso um novo jogador atinja uma pontuação que o coloque entre os melhores.

## Requisitos Não Funcionais

- **Linguagem de Programação:** C++
- **Manipulação de Arquivos:** Utilização de `fstream` para leitura e escrita de dados (perguntas, perfis de usuário, ranking).
- **Formato dos Arquivos:**
  - **Perguntas:** CSV (Comma Separated Values).
    - Ex: [Categoria]; [Dificuldade]; [Pergunta]; [OpcaoA]; [Opcao B]; [OpcaoC]; [Resposta Correta]; [Pontuacao]
  - **Perfis/Ranking:** CSV.
- **Estrutura de Dados:** Utilização de `struct` ou `class` para representar Pergunta, Usuario, Ranking.
- **Interface:** Linha de comando (console).

## Próximos Passos (Para o Desenvolvimento)

1. **Definição do Formato de Arquivo:** Decidir exatamente como as perguntas, perfis e ranking serão armazenados nos arquivos. Isso é crucial antes de começar a codificar as funções de leitura/escrita.
2. **Criação das Classes/Estruturas:** Implementar as classes Pergunta, Usuario, etc., com seus atributos e métodos básicos.
3. **Leitura e Escrita de Arquivos:** Começar com funções simples para ler e escrever dados em um arquivo de teste.
4. **Implementação do Fluxo Principal:** Construir o menu principal, a seleção de usuário/categoria/dificuldade.
5. **Lógica do Quiz:** Apresentar perguntas, receber respostas, verificar.
6. **Persistência:** Integrar as funções de salvamento/carregamento para perfis e ranking.
7. **Refinamento:** Melhorar a interface do usuário, adicionar tratamento de erros, documentar o código.

8. **Modo Administrador (se for):** Desenvolver a lógica para adicionar/editar perguntas.

## Integrantes

- Edson Ulisses
- Fabrício Nathan
- Matheus Goes
- Edney Bezerra
- Lucas Lima