



Curso de Ciência da Computação - UVA

Disciplina: Introdução à Ciência da Computação

Equipe: Luis Guilherme, Rômulo Rocha, Ruan

Projeto: Aplicativo Quiz Acadêmico

1. Introdução

O presente documento descreve o desenvolvimento de um aplicativo de Quiz interativo, concebido como projeto prático para a disciplina. O software visa oferecer uma interface gamificada para testes de conhecimento, unindo lógica de programação, estruturas de dados básicas e design de interface de usuário (UI).

2. Proposta e Objetivos

O aplicativo promete fornecer uma plataforma estável e intuitiva onde usuários podem testar seus conhecimentos em temas específicos.

- **O que ele faz:** Apresenta uma série de perguntas de múltipla escolha, contabiliza a pontuação em tempo real e fornece um feedback final sobre o desempenho do usuário.
- **Utilidade:** Serve como ferramenta de apoio ao estudo (mnemônica) e como base para sistemas mais complexos de avaliação digital.

3. Arquitetura e Desenvolvimento

Com base na estrutura do repositório, o desenvolvimento seguiu o padrão de separação de responsabilidades:

- **Interface (Frontend):** Desenvolvida com foco em usabilidade, utilizando elementos visuais que guiam o usuário através das etapas do quiz (Tela inicial, Pergunta e Resultado).
- **Lógica de Controle:** Implementada para gerenciar o fluxo de dados, validando se a resposta selecionada corresponde à resposta correta armazenada no banco de dados/array de objetos do código.
- **Recursos Visuais:** O projeto utiliza uma pasta dedicada (/images) para armazenar ícones e elementos gráficos que tornam a experiência mais imersiva e profissional.

4. Tecnologias Utilizadas

- **Linguagem de Programação:** HTML, CSS e JavaScript.
- **Versionamento:** Git e GitHub para controle de mudanças e colaboração remota.
- **Recursos Gráficos:** Ativos de imagem customizados para identidade visual do app.

5. Considerações Técnicas e Conclusão

O desenvolvimento deste projeto permitiu à equipe aplicar conceitos fundamentais de Ciência da Computação, como:

1. **Manipulação de Eventos:** Tratar os cliques do usuário.
2. **Lógica Condisional:** Sistema de pontuação e verificação de erros.
3. **Organização de Arquivos:** Estruturação de diretórios no GitHub para facilitar a manutenção do software.

O resultado final é uma aplicação funcional que cumpre os requisitos acadêmicos de interatividade e organização de código.

6. Detalhamento da Lógica de Programação

Para que o quiz funcione de forma dinâmica, a lógica foi estruturada em três pilares principais:

6.1. Estrutura de Dados das Questões

As perguntas não são fixas no layout, mas sim armazenadas em uma estrutura de dados (como um Array de objetos ou uma Lista). Cada objeto contém a string da pergunta, um vetor de opções e o índice da resposta correta. Isso permite que o aplicativo seja escalável, facilitando a adição de novas perguntas sem alterar a interface.

6.2. Algoritmo de Validação e Pontuação

O sistema utiliza uma variável contadora (ex: score) iniciada em zero. O fluxo lógico segue este algoritmo:

1. **Captura de Input:** O sistema aguarda o evento de clique em uma das opções.
2. **Comparação Condisional:** Ao clicar, o programa compara o ID da opção selecionada com o atributo correctAnswer do objeto atual.
3. **Feedback Instantâneo:** Se selecionado == correto, o contador é incrementado (score++). Caso contrário, o sistema pode sinalizar o erro visualmente.
4. **Troca de Estado:** O índice da pergunta atual é atualizado, disparando uma função de "renderização" que limpa a tela e carrega os dados da próxima questão.

6.3. O Ciclo de Vida do Quiz

O fluxo do usuário é controlado por estados (Screens). O diagrama abaixo exemplifica como o código transita entre as telas:

- **Estado Inicial:** Tela de boas-vindas e carregamento dos recursos (imagens da pasta /images).
- **Estado Ativo:** Loop de repetição que percorre a lista de questões enquanto o índice for menor que o tamanho total da lista.
- **Estado Final:** Verificação da condição de término, exibição da pontuação acumulada e opção de reinicialização (reset do estado).

7. Conclusão

A conclusão do desenvolvimento deste aplicativo de Quiz reafirma a importância da aplicação prática dos conceitos teóricos de Ciência da Computação. Através deste projeto, foi possível integrar diferentes pilares da computação, como o desenvolvimento de interfaces, a gestão de ativos digitais e a implementação de lógica condicional estruturada.

Além disso, a elaboração deste trabalho escrito permitiu uma análise crítica sobre a arquitetura do sistema, evidenciando que, embora o aplicativo seja um modelo inicial, sua estrutura é escalável para suportar bancos de dados mais complexos ou novas funcionalidades, como temporizadores ou rankings online. Em suma, o projeto foi fundamental para consolidar o entendimento sobre o ciclo de vida de um software, desde a concepção da ideia até a documentação final.