**ESCOLA S DE SÃO JOSÉ**

**Professor:** Jackson **Unidade curricular:** Teste de Sistemas

**Alunos:** Matheus F. Duarte, Gabriel J. Lemos,

João G. S. Porfirio, Victor B. de Jesus, Arthur R. Greghi

**Turma:** 3A

**DOCUMENTOS PADRÃO IEEE 829**

**SISTEMA DO MARCOLA TESTES VOCACIONAIS**

São José, 3 de outubro de 2022

**PLANO DE TESTES**

Para fazer os testes do software do Marcola Testes Vocacionais, utilizaremos o “Modelo em V”, encontrado na Internet, baseado em duas bases: verificação e validação.

A partir disso, iremos testar o sistema manualmente, procurando saber se o software foi construído da maneira correta e se ele possui pontos de melhoria para atender ainda melhor as expectativas de um cliente em potencial.

Tudo isso será feito no mesmo dia que iniciarmos os testes, a partir pelos desenvolvedores do projeto:

* Matheus F. Duarte;
* João G. S. Porfirio;
* Victor B. de Jesus;
* Gabriel J. Lemos;
* Arthur G. Roecker.

**ESPECIFICAÇÃO DO PROJETO DE TESTE**

**Características a serem testadas**

**Requisitos Funcionais**

1. O usuário deverá inserir seu CPF. Caso o CPF já tenha sido usado, ele será redirecionado para a tela do seu resultado anterior.. Todavia, caso ele nunca tenha sido usado, o usuário vai diretamente para a tela de perguntas.
2. Na tela de perguntas, cada alternativa somará uma pontuação pré-determinada para cada carreira. Ao final, o sistema deverá considerar a carreira com maior pontuação como resultado do usuário.
3. Na tela final, que exibe o resultado, deverá existir um botão para reiniciar o teste.

**Requisitos Não-Funcionais**

1. JAVA (Android Studio), API 27;
2. Firebase Realtime Database;
3. Firebase Authentication;
4. Designer figma.,

**ESPECIFICAÇÃO DE CASOS DE TESTE**

**Itens a serem testados**

1. Área de inserção de CPF(entrada); Botão de Avançar; Botão de Reiniciar; Check box de opções de resposta; Resultado final com a profissão baseada;
2. Especificação de Entrada: 11 dígitos para número de CPF; Especificação de Saída: transição para tela de seleção das checkboxes;
3. Premissas: Após a inserção de somente 11 dígitos para a entrada de CPF, ocorrerá uma transição para a tela de checkbox onde serão selecionadas diversas respostas à pergunta; Botão de avançar ocasiona uma troca de tela para novas telas com respostas nas checkboxes;
4. Restrições do Ambiente: Android;

**ESPECIFICAÇÃO DE PROCEDIMENTO DE TESTE**

1. No Android Studio, com um celular configurado na API 27, rodar o aplicativo;
2. Inserir um CPF que nunca foi cadastrado e checar se ele vai para a tela de perguntas;
3. Ir até o final das perguntas;
4. Na tela de resultado, apertar no botão para reiniciar e refazer o flow, mas com diferentes respostas, a fim de verificar se o resultado irá mudar;
5. Fechar o aplicativo;
6. Inserir o CPF que foi usado no passo 2 e checar se ele o usuário é diretamente levado para a tela de resultado.
7. Analisar o código .java e ver possíveis repetições que poderiam ser substituídas por funções.

**RELATÓRIO DE STATUS DE TESTE**

**Responsável pelo teste: Matheus F. Duarte**

**OK EXISTE ALGO DE ERRADO**

1. No Android Studio, com um celular configurado na API 27, rodar o aplicativo;
2. Inserir um CPF que nunca foi cadastrado e checar se ele vai para a tela de perguntas;
3. Ir até o final das perguntas;
4. Na tela de resultado, apertar no botão para reiniciar e refazer o flow, mas com diferentes respostas, a fim de verificar se o resultado irá mudar;
5. Reiniciar o aplicativo;
6. Inserir o CPF que foi usado no passo 2 e checar se ele o usuário é diretamente levado para a tela de resultado.
7. Analisar o código .java e ver possíveis repetições que poderiam ser substituídas por funções. - o código tem muitas repetições que poderiam ser substituídas por métodos e ir puxando o método.

**Responsável pelo teste: Gabriel J. Lemos**

**OK EXISTE ALGO DE ERRADO**

1. No Android Studio, com um celular configurado na API 27, rodar o aplicativo;
2. Inserir um CPF que nunca foi cadastrado e checar se ele vai para a tela de perguntas;
3. Ir até o final das perguntas;
4. Na tela de resultado, apertar no botão para reiniciar e refazer o flow, mas com diferentes respostas, a fim de verificar se o resultado irá mudar;
5. Reiniciar o aplicativo;
6. Inserir o CPF que foi usado no passo 2 e checar se ele o usuário é diretamente levado para a tela de resultado.
7. Analisar o código .java e ver possíveis repetições que poderiam ser substituídas por funções. - **o código tem muitas repetições que poderiam ser substituídas por métodos e ir puxando o método.**

**LOG DE TESTE**

1. **26/09/2022, 08:36** - No Android Studio, com um celular configurado na API 27, rodar o aplicativo;
2. **26/09/2022, 08:36** - Inserir um CPF que nunca foi cadastrado e checar se ele vai para a tela de perguntas;
3. **26/09/2022, 08:39** - Ir até o final das perguntas;
4. **26/09/2022, 08:43** - Na tela de resultado, apertar no botão para reiniciar e refazer o flow, mas com diferentes respostas, a fim de verificar se o resultado irá mudar;
5. **26/09/2022, 08:44** - Reiniciar o aplicativo;
6. **26/09/2022, 08:45** - Inserir o CPF que foi usado no passo 2 e checar se ele o usuário é diretamente levado para a tela de resultado.
7. **26/09/2022, 08:47** - Analisar o código .java e ver possíveis repetições que poderiam ser substituídas por funções.

O sistema está funcionando com rapidez e pouco consumo de memória do celular.

**RELATÓRIO DE INCIDENTES DE TESTE**

Os computadores e softwares utilizados durante os testes estavam funcionando perfeitamente e não houve incidentes quanto a isso. Um ponto de melhoria seria apenas uma sala mais silenciosa visando maior interação entre a equipe de desenvolvimento do projeto.

**RELATÓRIO SUMÁRIO DE TESTES**

* No Login, o usuário precisa ter uma senha e um botão para “Esqueci minha senha”. No sistema atual, qualquer pessoa com o CPF de outra em mãos consegue violar sua privacidade e obter o acesso ao seu resultado;
* Na tela de resultado, quando há mais de uma carreira como resultado, o balão de fala fica pequeno para o texto, abrindo pouco espaço para margem, tornando o design mais harmônico. Sendo assim, nesse caso em específico, o design deveria ser aprimorado.