



Plano de Teste FaceToFace – Sistema para entrevistas do Vem Ser

Time de Desenvolvimento

Back-End

- Bruno Pelliciari Barduco
- Julio Gabriel Moreira Rocha
- Matheus De Souza Gonçalves

Front-End

- Yuri Nunes Atzler
- Gicelly Oliveira

Quality Assurance

- Pietro Luigi Franco Bianchi Silva

Product Owner

- Ályson Campos

ÍNDICE

1 Introdução	4
2 Arquitetura	4
2.1 Referências/Documentações do Projeto	4
3 Requisitos	5
4 Funcionalidades	6
5 Cronograma	7
6 Estratégia de Teste	8
6.1 Escopo de Testes	8
7 Ambiente e Ferramentas	8
7.1 Definições do ambiente de testes	8
7.2 Ferramentas de testes	9
8 Classificação de Bugs	10
9 Cenário de Testes Automatizados	10
10 Cenário de Testes Exploratórios	12
11 Definição de Pronto	12

1 Introdução

FaceToFace é um sistema criado para que o setor responsável da DBC possa gerenciar os candidatos do programa Vem Ser, podendo realizar o cadastro do candidato, adição de nota, gerenciamento de entrevistas, e observações.

2 Arquitetura

O framework utilizado para implementação do Sistema FaceToFace é o React Native, que possui uma arquitetura orientada a componentes e pode ser utilizado tanto para sistemas WEB, Android e/ou IOS.

Para o armazenamento, consulta e alteração de dados da aplicação, será usada uma API que disponibiliza um conjunto de serviços REST. Em que cliente (aplicação) envia uma requisição à uma URL (endpoint) para acessar os serviços da API, e tem como retorno um objeto na forma de um JSON.

2.1 Referências/Documentações do Projeto

A tabela abaixo identifica a documentação utilizada para a elaboração deste Plano de teste.

Documento	Observações
Especificação de Requisitos	Irá definir as funcionalidades e o ambiente que o Sistema deve possuir
Especificação de Casos de Uso	Descreve de forma detalhada as funcionalidades do Sistema
Especificação das Regras de negócio	Definirá as regras e os comportamentos do Sistema

3 Requisitos

Requisitos Funcionais	Requisitos Não Funcionais
RF001 - O sistema deve permitir o login dos usuários cadastrados	NF001 - O sistema deverá ser desenvolvido utilizando o framework React Native
RF002 - O sistema deve permitir o cadastro de candidatos por um usuário logado	NF002 - O banco de dados utilizado deverá ser o OracleDB
RF003 - O sistema deve permitir alteração de informações do usuário logado	NF003 - O sistema deverá ser desenvolvido utilizando o framework Spring Boot
RF004 - O sistema deve permitir alteração de informações dos candidatos por um usuário logado	NF004 - O sistema deve ser navegável apenas com o usuário devidamente logado no sistema
RF005 - O sistema deve permitir uma busca de candidatos	
RF006 - O sistema deve disponibilizar uma funcionalidade de listagem de usuários quando logado por um administrador	
RF007 - O sistema deve permitir o cadastro de colaborador por um usuário administrador logado	
RF008 - O sistema deve permitir o cadastro para marcar data de entrevista com um candidato por um usuário com permissão	
RF009 - O sistema deve permitir	

atualizar o status dos eventos do calendário	
RF010 - O sistema deve permitir desabilitar candidatos por um usuário com permissão	

4 Funcionalidades

Funcionalidades	Comportamento Esperado
Tela Login	<ul style="list-style-type: none"> • Ao digitar email e senha e clicar no botão de Login deve redirecionar para tela de home do usuário • Ao clicar no botão de esqueci minha senha deve redirecionar para tela de Recuperação de senha
Tela Home	<ul style="list-style-type: none"> • Ao clicar no botão de Cadastro usuário deve redirecionar para tela de registrar usuário • Ao clicar no botão de Cadastro candidato deve redirecionar para tela de registrar candidato • Ao clicar no botão de Cadastro entrevista deve redirecionar para tela de registrar entrevista de candidato • Ao clicar no botão de Agenda deve redirecionar para o calendário das entrevistas • Ao clicar no botão de Editar meu perfil deve redirecionar para tela de edição do usuário • Ao clicar no botão de sair deve ser Deslogado • Ao digitar algum nome no campo de pesquisa, ou selecionar filtro de trilha ou edição deve listar os usuário/candidatos
Cadastro Usuário	<ul style="list-style-type: none"> • Ao digitar email, nome completo, cidade, estado, selecionar gênero e tipo irá efetuar um cadastro de usuário na plataforma e o usuário deverá ser redirecionado para a tela de Home • Deve indicar o campo obrigatório a ser corrigido pelo usuário

Cadastro Candidato	<ul style="list-style-type: none"> • Ao digitar email, nome completo, cidade, estado, selecionar gênero, turma, edição e currículo irá efetuar um cadastro de usuário na plataforma e o usuário deverá ser redirecionado para a tela de Home • Deve indicar o campo obrigatório a ser corrigido pelo usuário
Cadastro Entrevista	<ul style="list-style-type: none"> • Ao digitar email do candidato, deve completar campos relacionados ao candidato e selecionar data e hora da entrevista irá efetuar um cadastro de entrevista na plataforma e o usuário deverá ser redirecionado para a tela de Home • Deve indicar o campo obrigatório a ser corrigido pelo usuário
Agenda	<ul style="list-style-type: none"> • Ao clicar no post de Entrevista marcada deve redirecionar para tela de detalhes da entrevista • Ao clicar no botão de Cadastrar nova entrevista deve redirecionar para o cadastro de entrevista • Ao clicar nos botões de voltar/avançar do calendário deve mudar o mês do calendário
Editar meu perfil	<ul style="list-style-type: none"> • Ao clicar no botão de Enviar deve atualizar as informações do usuário e redirecionar para o cadastro de entrevista • Ao clicar no botão de Mudar senha deve redirecionar para tela de alterar senha

5 Cronograma

Tipo de Teste	Data de início	Data de término
Planejar teste	25/11/2022	28/11/2022
Projetar teste	28/11/2022	02/12/2022
Implementar teste	28/11/2022	02/12/2022
Executar teste	03/12/2022	04/12/2022

Avaliar teste	03/12/2022	04/12/2022
---------------	------------	------------

6 Estratégia de Teste

6.1 Escopo de Testes

O plano de testes abrange todas as funcionalidades descritas na tabela acima. Serão executados testes em todos os níveis conforme a descrição abaixo.

Testes Unitários: O código terá uma cobertura de 100% de testes unitários, que são de responsabilidade dos desenvolvedores.

Testes de Integração: Serão executados testes de integração em todos os endpoints, e esses testes serão de responsabilidade do time de qualidade.

Testes Automatizados: Serão realizados testes end-to-end na funcionalidade de Login.

Testes Manuais: Todas as funcionalidades serão testadas manualmente pelo time de qualidade seguindo o Cenário de Testes deste plano de testes.

7 Ambiente e Ferramentas

7.1 Definições do ambiente de testes

- Teremos testes unitários desenvolvidos individualmente por desenvolvedores com um volume pequeno de dados, faremos testes de integração com desenvolvedores.
- Teremos testes em toda aplicação com testadores com um volume grande de dados, são utilizados dados reais ou criados na hora, também teremos testes de aceitação com toda aplicação.
- Os testes serão feitos do ambiente de homologação, e contém as mesmas configurações do ambiente de produção com uma massa de dados gerada previamente pelo time de qualidade.

7.2 Ferramentas de testes

Ferramenta	Time	Descrição
IntelliJ	Desenvolvimento	Ferramenta de ambiente de desenvolvimento
Visual Studio Code	Desenvolvimento	Ferramenta de ambiente de desenvolvimento
Spring Boot	Desenvolvimento	Framework utilizado para desenvolvimento da aplicação JAVA
Oracle DB	Desenvolvimento	Sistema de gerenciamento de banco de dados
JUnit	Desenvolvimento	Framework utilizada para testes unitários
Jest	Desenvolvimento	Framework utilizada para testes unitários
Cypress	Qualidade	Ferramenta para realização de testes automatizados de Front-End
Cypress	Qualidade	Ferramenta para realização de testes automatizados de API
POSTMAN	Qualidade	Ferramenta para realização de testes exploratórios de API

8 Classificação de Bugs

Os Bugs serão classificados com as seguintes severidades:

ID	Nível de Severidade	Descrição
1	Blocker	<ul style="list-style-type: none">• Bug que bloqueia o teste de uma função ou feature causa crash na aplicação.• O botão não funciona impedindo o uso completo da funcionalidade.• Bloqueia a entrega.

2	Grave	<ul style="list-style-type: none"> • Funcionalidade não funciona como o esperado • Input incomum causa efeitos irreversíveis
3	Moderada	<ul style="list-style-type: none"> • Funcionalidade não atinge certos critérios de aceitação, mas sua funcionalidade em geral não é afetada • Mensagem de erro ou sucesso não é exibida
4	Pequena	<ul style="list-style-type: none"> • Quase nenhum impacto na funcionalidade porém atrapalha a experiência • Erro ortográfico • Pequenos erros de UI

9 Cenário de Testes Automatizados

Funcionalidades	Cenário de Testes (CT)
Login	ct1: Validar login com username e senha válida ct2: Validar login com username e senha inválida ct3: Validar login sem preencher nenhum campo ct4: Validar login com username correto sem preencher senha ct5: Validar login com username incorreto com senha correta ct6: Validar botão Esqueci minha senha
Esqueci minha Senha	-
Cadastro Candidato	ct1: Validar cadastrar candidato preenchendo todos campos corretamente ct2: Validar cadastrar candidato preenchendo somente campos obrigatórios ct3: Validar cadastrar candidato preenchendo somente campos não obrigatórios ct4: Validar cadastrar candidato sem preencher nenhum campo

Cadastro Usuário	ct1: Validar cadastrar usuário preenchendo todos campos corretamente ct2: Validar cadastrar usuário sem preencher nenhum campo ct3: Validar cadastrar usuário preenchendo somente campos não obrigatórios ct4: Validar cadastrar usuário sem preencher nenhum campo
Cadastro de Entrevista	ct1: Validar cadastrar entrevista preenchendo todos campos corretamente ct2: Validar cadastrar entrevista preenchendo o campo email do candidato e completando o resto das informações diretamente nos campos ct3: Validar cadastrar entrevista preenchendo somente campos obrigatórios ct4: Validar cadastrar entrevista preenchendo somente campos não obrigatórios ct5: Validar cadastrar entrevista sem preencher nenhum campo ct6: Validar cadastrar outra entrevista de um candidato com entrevista já marcada
Filtro	ct1: Validar pesquisa preenchendo campo com nome ct2: Validar pesquisa selecionando campo com edição ct3: Validar pesquisa selecionando campo com trilha ct4: Validar lista candidatos por nota ct5: Validar lista Gestao Pessoas/Instrutor ct6: Validar botão resetar filtros de pesquisa ct7: Validar botão marcar entrevista de candidato ct8: Validar botão detalhes de candidato ct9: Validar botão editar de candidato ct10: Validar botão detalhes de usuário ct11: Validar botão editar de usuário
Agenda	ct1: Validar post de entrevista do candidato marcada ct2: Validar cancelar entrevista do candidato

10 Cenário de Testes Exploratórios

Funcionalidades	Cenário de Testes (CT)
Login	ct1: Validar login com username e nova senha enviado para o email do usuário
Esqueci minha Senha	ct1: Validar envio de nova senha para email do usuário
Cadastro Candidato	ct1: Validar cadastrar candidato com email já registrado
Cadastro Colaborador	ct1: Validar cadastrar usuário com email já registrado
Cadastro de Entrevista	ct1: Validar cadastrar entrevista em data/hora já utilizada
Filtro	ct1: Validar botão deletar de candidato ct2: Validar botão deletar de usuário
Agenda	ct1: Validar funcionamento do calendário ct2: Validar botão cadastrar nova entrevista ct3: Validar mudança de status da entrevista, confirmada pelo email do candidato

11 Definição de Pronto

Será considerada pronta as funcionalidades que passarem pelas verificações e testes descritas neste plano de teste, não apresentarem bugs com a severidade acima de Moderada, e passarem por uma validação de negócio de responsabilidade do time de produto.