Test Plan Avaliaí

1 Introdução

Avaliaí é um sistema de classificação de disciplinas desenvolvido por estudantes e para estudantes, tendo por objetivo principal permitir que alunos consigam visualizar os pontos fortes e fracos das disciplinas que desejam ou irão cursar, tal como o que outras pessoas acharam dessas disciplinas, com base em suas experiências. O sistema conta com funcionalidades como: permitir a avaliação de disciplinas, fazer comentários sobre disciplinas, sugerir a criação de uma nova disciplina no sistema, classificar comentários de outros alunos.

2 Arquitetura

Para o front-end, o Avaliaí utiliza o framework Next.JS, uma estrutura da web de desenvolvimento front-end React, que consequentemente possui uma arquitetura orientada a componentes, contribuindo para a sua consequente reutilização.

O back-end utiliza o Spring-Boot, na linguagem Java, e ferramentas como o Insomnia, que permite fazer requisições de API. Para a documentação dos endpoints, foi utilizada a ferramenta Swagger.

No ambiente mobile, o framework Flutter foi utilizado.

Para que o front-end e o back-end estejam integrados, é utilizada uma API que segue os padrões REST. A aplicação envia uma requisição a um determinado endpoint, dependendo de sua função, para ter acesso aos serviços da API, que retorna um código em JSON.

O grupo optou por utilizar o RabbitMQ como serviço de mensageria, necessário para a requisição de disciplinas e consequente aprovação ou reprovação pelo administrador.

3 Funcionalidades

Funcionalidade	Cadastro
Comportamento Esperado	Ao digitar email, nome completo, usuário e senha irá efetuar um cadastro na plataforma e o usuário deverá ser redirecionado para a tela de login. Deve indicar o campo obrigatório a ser corrigido pelo usuário.
Verificações	● Senha min 8 caracteres e no máximo 18 ● Todos os campos devem ser obrigatórios. ● Exibir uma mensagem de confirmação em caso positivo. ● Redirecionar o usuário para tela de login. ● Exibir a mensagem de falha em caso de usuário existente. ● Exibir mensagem de falha em caso de confirmação de senha não ser igual ● Exibir mensagem de falha no caso de campo obrigatório incompleto.
Critérios de Aceite	● Senha de 8 caracteres, no mínimo. ● Todos os campos devem ser obrigatórios. ● Exibir uma mensagem de confirmação em caso positivo. ● Redirecionar o usuário para tela de login. ● Exibir a mensagem de falha em caso usuário já existente. ● Exibir mensagem de falha caso a confirmação de senha não seja igual a senha digitada anteriormente. ● Exibir mensagem de falha caso o campo obrigatório esteja incompleto.

Funcionalidade	Login

Comportamento Esperado	Ao digitar seu usuário e senha corretamente, o usuário irá logar na plataforma. Ao tentar se logar e falhar 3 vezes consecutivas, o usuário terá que esperar 15 minutos para tentar logar novamente. O sistema deve aceitar usuários que já está logado em outro *device* se logar novamente
Verificações	● Login no Sistema com sucesso ● Usuário Inválido ● Usuário não preencher campo obrigatório ● Senha Incorreta ● Senha Incorreta 3 vezes
Critérios de Aceite	● Ter acessibilidade no sistema ● Redimensionar a Tela

Funcionalidade	Lista de Livros
Comportamento Esperado	Ao entrar na tela inicial e clicar no botão buscar, o usuário irá ver uma lista com todo acervo de livros. Deve ser exibido o ícone de paginação, possibilitando ao usuário navegar entre as telas. Deve ser exibido uma caixa de texto para o usuário realizar nova pesquisa.
Verificações	
Critérios de Aceite	

Funcionalidade	Excluir cadastro
Comportamento Esperado	Ao clicar no item em excluir perfil, o mesmo deverá ser excluído. Deve aparecer uma caixa de confirmação perguntando se usuário deseja mesmo excluir o perfil. O usuário deve ter a opção de cancelar exclusão. O usuário deve confirmar a exclusão.
Verificações	● Exclusão do perfil ● Mensagem de confirmação em caso positivo. ● Mensagem de cancelamento.
Critérios de Aceite	

Funcionalidade	Pesquisa
Comportamento Esperado	Ao digitar um texto na barra de pesquisa deverá aparecer os resultados correspondentes. Devem ser exibidas sugestões ao clicar na caixa de texto e a medida que o usuário digita. Deve ser exibida uma mensagem caso não haja livros correspondentes a pesquisa do usuário.
Verificações	● Pesquisa encontrou os resultados esperados ● Pesquisa não encontrou resultados ● Sugestões de Pesquisa ● Limite de 140 caracteres
Critérios de Aceite	

4 Estratégia de Teste

- **Escopo de Testes**

O plano de testes abrange todas as funcionalidades descritas na tabela acima. Esse plano de testes exclui a funcionalidade de cadastro de livros.

Serão executados testes em todos os níveis conforme a descrição abaixo.

Testes Unitários: o código terá uma cobertura de 60% 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 a documentação de Cenários de teste e destes TestPlan.
Versão Beta: Será lançada uma versão beta para 3 usuários pré-cadastrados antes do release.

- **Ambiente e Ferramentas**

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.

As seguintes ferramentas serão utilizadas no teste:

Ferramenta	Time	Descrição
Insomnia	Qualidade	Ferramenta para realização de testes de API
Jest	Desenvolvimento	Framework utilizada para testes unitários
Cypress	Qualidade	Ferramenta para testes end-to-end
Lighthouse	Desenvolvimento	Avaliação de performance e acessibilidade da aplicação
Gravador de Passos	Desenvolvimento	Prover evidências dos testes

5 Classificação de Bugs

Os Bugs serão classificados com as seguintes severidades:

ID	Nivel de Severidade	Descrição
1	Blocker	● Bug que bloqueia o teste de uma função ou feature causa crash na aplicação. ● Botão não funciona impedindo o uso completo da funcionalidade. ● Bloqueia a entrega.
2	Grave	● Funcionalidade não funciona como o esperado ● Input incomum causa efeitos irreversíveis
3	Moderada	● 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	● Quase nenhum impacto na funcionalidade porém atrapalha a experiência ● Erro ortográfico ● Pequenos erros de UI

6 Definição de Pronto

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