**Test Plan Alura Pic**

# Introdução

Allura Pic é um sistema de gestão de fotos, o sistema foi criado com o propósito de dar suporte para a formação em Angular. Conta com funcionalidades simples de Cadastro, Login e Manipulação de fotos.

# Arquitetura

O framework utilizado para a implementação do front-end do AluraPic é o Angular, que possui uma arquitetura orientada a componentes.

Para o armazenamento, consulta e alteração de dados da aplicação, será usada uma API que disponibiliza um conjunto de serviços REST. Funciona como na imagem, em que o 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.

# Funcionalidades

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidades** | **Comportamento Esperado** | **Verificações** | **Critérios de Aceite** |
| Cadastro | 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. | * Senha min 8 caracteres e no máximo 18 * Todos os campos devem ser obrigatórios. * Exibir uma mensagem de confirmação em caso positivo. * Redirecionar o usuario 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. |  |
| Login | 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 | * Login no Sistema com sucesso * Usuario Inválido * Usuário não preencher campo obrigatório * Senha Incorreta * Senha Incorreta 3 vezes | * Ter acessibilidade no sistema * Redimensionar a Tela |
| Lista de Fotos | Ao entrar na tela inicial o usuário irá ver uma lista com todas as fotos.  Deve ser exibido um ícone de coração no canto direito.  Deve o ícone de coração deve exibir uma animação e trocar para a cor vermelha.  Deve ser exibido uma caixa de texto para o usuário escrever o comentário. |  |  |
| Comentario em Fotos | Ao clicar nos detalhes das fotos o usuário poderá fazer um comentário.  Deve ser exibido uma caixa de texto para o usuário escrever o comentário.  Deve ser exibido o nome do usuário  Deve ser exibido o texto “Comentar” na barra de texto para orientar o usuário.  Deve ter limite de 140 por comentário | * Limite de 140 caracteres. * Nome do usuario. * Comentários anteriores. |  |
| Detalhes da fotos | Serão exibidos os detalhes da foto: - Foto - Curtidas - Comentarios  - Opção de remoção da foto  Usuário deve conseguir curtir uma mensagem ao clicar no ícone de “coração”  O ícone de coração deve mudar de cor  Usuário que teve sua mensagem curtida deve receber uma notificação  Deve ser exibido uma caixa de texto para o usuário escrever o comentário. | * Número correto de curtidas e comentários. * Opção de remoção de fotos. * Possibilidade de curtir e comentar nas fotos. |  |
| Remover foto | Ao clicar no item em excluir a foto deverá ser removida  Deve aparecer uma caixa de confirmação perguntando se usuário deseja mesmo excluir o stories  O usuário deve ter a opção de cancelar exclusão  O usuário deve confirmar a exclusão. | * Exclusão da foto * Acessar a foto após a remoção. * Mensagem de confirmação em caso positivo. * Mensagem de cancelamento. |  |
| Pesquisa | Ao digitar um texto na barra de pesquisa deverá aparecer as fotos 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 fotos correspondentes a pesquisa do usuário. | * Pesquisa encontrou os resultados esperados * Pesquisa não encontrou resultados * Sugestões de Pesquisa. |  |

# 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 upload de fotos.

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 20 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 |
| POSTMAN | Qualidade | Ferramenta para realização de testes de API |
| Jasmine | Desenvolvimento | Framework utilizada para testes unitários |
| Selenium | Qualidade | Ferramenta para testes end-to-end |

# Classificação de Bugs

Os Bugs serão classificados com as seguintes severidades:

|  |  |  |
| --- | --- | --- |
| ID | Nível 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 |

# 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.