

Test Plan Alura Pic

1 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.

2 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.

3 Funcionalidades

Funcionalidades	Comportamento Esperado	Verificações	Critérios de Aceite
Cadastro	<p>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.</p> <p>Deve indicar o campo obrigatório a ser corrigido pelo usuário.</p>	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• O sistema deve aceitar senhas com 8 a 18 caracteres e rejeitar senhas que não atendam a esse critério.• O sistema não deve permitir que o usuário prossiga com o cadastro se algum dos campos obrigatórios estiver em branco.• O sistema deve exibir uma mensagem clara de confirmação do cadastro.

		campo obrigatório incompleto.	
Login	<p>Ao digitar seu usuário e senha corretamente o usuário irá logar na plataforma.</p> <p>Ao tentar se logar e falhar 3 vezes consecutivas o usuário terá que esperar 15 minutos para tentar logar novamente.</p> <p>O sistema deve aceitar usuários que já está logado em outro device se logar novamente</p>	<ul style="list-style-type: none"> • Login no Sistema com sucesso • Usuario Inválido • Usuário não preencher campo obrigatório • Senha Incorreta • Senha Incorreta 3 vezes 	<ul style="list-style-type: none"> • Ter acessibilidade no sistema • Redimensionar a Tela •
Lista de Fotos	<p>Ao entrar na tela inicial o usuário irá ver uma lista com todas as fotos.</p> <p>Deve ser exibido um ícone de coração no canto direito.</p> <p>Deve o ícone de coração deve exibir uma animação e trocar para a cor vermelha.</p> <p>Deve ser exibido uma caixa de texto para o usuário escrever o comentário.</p>	<ul style="list-style-type: none"> • Ao acessar a tela inicial, a lista de fotos deve ser exibida. • Abaixo de cada foto, deve haver uma caixa de texto para escrever comentários. 	<ul style="list-style-type: none"> • Todas as fotos devem ser carregadas e exibidas de forma visível • A caixa de texto deve ser visível e estar posicionada abaixo de cada foto.
Comentario em Fotos	<p>Ao clicar nos detalhes das fotos o usuário poderá fazer um comentário.</p> <p>Deve ser exibido uma caixa de texto para o usuário escrever o comentário.</p>	<ul style="list-style-type: none"> • Limite de 140 caracteres. • Nome do usuario. • Comentários anteriores. 	<ul style="list-style-type: none"> • O usuário deve conseguir acessar os detalhes da foto e adicionar um comentário. • A caixa de texto para o comentário

	<p>Deve ser exibido o nome do usuário</p> <p>Deve ser exibido o texto “Comentar” na barra de texto para orientar o usuário.</p> <p>Deve ter limite de 140 por comentário</p>		deve estar visível e funcional.
Detalhes da fotos	<p>Serão exibidos os detalhes da foto:</p> <ul style="list-style-type: none"> - Foto - Curtidas - Comentarios - Opção de remoção da foto <p>Usuário deve conseguir curtir uma mensagem ao clicar no ícone de “coração”</p> <p>O ícone de coração deve mudar de cor</p> <p>Usuário que teve sua mensagem curtida deve receber uma notificação</p> <p>Deve ser exibido uma caixa de texto para o usuário escrever o comentário.</p>	<ul style="list-style-type: none"> • Número correto de curtidas e comentários. • Opção de remoção de fotos. • Possibilidade de curtir e comentar nas fotos. 	<ul style="list-style-type: none"> • Os detalhes da foto devem ser exibidos de forma visível e precisa. • Ao clicar no ícone de "coração", o usuário deve poder curtir a foto.
Remover foto	<p>Ao clicar no item em excluir a foto deverá ser removida</p> <p>Deve aparecer uma caixa de confirmação perguntando se usuário deseja mesmo excluir o stories</p> <p>O usuário deve ter a opção de cancelar exclusão</p>	<ul style="list-style-type: none"> • Exclusão da foto • Acessar a foto após a remoção. • Mensagem de confirmação em caso positivo. • Mensagem de cancelamento. 	<ul style="list-style-type: none"> • A foto deve ser removida com sucesso após a ação de exclusão. • A caixa de confirmação deve aparecer, solicitando ao

	O usuário deve confirmar a exclusão.		usuário que confirme a exclusão. <ul style="list-style-type: none"> • O usuário deve poder cancelar a exclusão e a foto não deve ser removida se a opção "cancelar" for selecionada
Pesquisa	<p>Ao digitar um texto na barra de pesquisa deverá aparecer as fotos correspondentes.</p> <p>Devem ser exibidas sugestões ao clicar na caixa de texto e a medida que o usuário digita.</p> <p>Deve ser exibida uma mensagem caso não haja fotos correspondentes a pesquisa do usuário.</p>	<ul style="list-style-type: none"> • Pesquisa encontrou os resultados esperados • Pesquisa não encontrou resultados • Sugestões de Pesquisa. 	<ul style="list-style-type: none"> • As fotos que correspondem ao texto de pesquisa devem ser exibidas corretamente. • Sugestões relevantes devem aparecer à medida que o usuário digita na caixa de pesquisa, facilitando a seleção de opções de pesquisa.

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 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

5 Classificação de Bugs

ID	Nível de Severidade	Descrição
1	Grave	<ul style="list-style-type: none">• Ao inserir a senha, a quantidade de caracteres não se limita a 18.

2	Baixa	<ul style="list-style-type: none"> • Ao clicar no icone de coração (curtida) ele não se altera para a cor vermelha que é um requisito do sistema
3	Moderada	<ul style="list-style-type: none"> • Após curtir uma imagem não é possível descurtir

6 Definição de Pronto

Uma funcionalidade será considerada pronta quando atender aos seguintes critérios:

- **Implementação Completa:** A funcionalidade foi implementada conforme os requisitos especificados no Test Plan.
- **Testes Unitários:** O código da funcionalidade possui cobertura de testes unitários de pelo menos 60%, com testes que verificam todos os principais cenários de uso e casos de borda.
- **Testes de Integração:** A funcionalidade passou por testes de integração que garantem que os diferentes componentes e serviços interagem corretamente.
- **Testes Automatizados:** A funcionalidade foi testada automaticamente em relação a cenários críticos, como o processo de login.
- **Testes Manuais:** A funcionalidade foi testada manualmente por membros do time de qualidade, seguindo os cenários de teste e critérios de aceitação definidos no Test Plan.
- **Ausência de Bugs Críticos:** A funcionalidade não possui bugs de severidade Grave que impactariam significativamente a usabilidade ou segurança do sistema.

- **Documentação Atualizada:** A documentação, incluindo a documentação de código e a documentação do usuário, foi atualizada para refletir as mudanças na funcionalidade.
- **Revisão de Código:** O código passou por uma revisão de código por outros membros da equipe, garantindo boas práticas de desenvolvimento e conformidade com os padrões estabelecidos.
- **Validação de Produto:** A funcionalidade passou por uma validação de negócio conduzida pelo time de produto para garantir que atenda às necessidades do usuário e ao objetivo do projeto.

Uma funcionalidade só será considerada pronta quando todos esses critérios forem atendidos. Qualquer desvio significativo desses critérios exigirá a correção antes de ser considerada pronta para o lançamento.