

# 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</p> <p>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"><li>• Senha mín. 8 caracteres</li><li>• Todos os campos devem ser obrigatórios.</li><li>• Exibir uma mensagem de confirmação em caso positivo.</li><li>• Redirecionar o usuário para tela de login.</li><li>• Exibir a mensagem de</li></ul>	

		falha em caso de usuário existente. <ul style="list-style-type: none"> <li>• Exibir mensagem de falha em caso de confirmação de senha não for correspondente.</li> <li>• Exibir mensagem de falha no caso de campo obrigatório incompleto.</li> </ul>	
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 usuário que já está logado em outro device se logar novamente.</p>	<ul style="list-style-type: none"> <li>• Login no Sistema</li> <li>• Usuário Invalido</li> <li>• Campo Obrigatório</li> </ul>	<ul style="list-style-type: none"> <li>• O nome e a senha devem ser obrigatórios</li> <li>• O sistema deve funcionar em todos os navegadores.</li> </ul>
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>	<ul style="list-style-type: none"> <li>• Número de fotos.</li> <li>• Número de curtidas e comentários correspondentes.</li> <li>• Animação do ícone.</li> </ul>	

	<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>		
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> <p>Deve ser exibido o texto “Comentar” na barra de texto para orientar o usuário.</p> <p>Deve ser exibido o nome do usuário e o comentário</p> <p>Deve ter limite de 140 caracteres por comentário</p>	<ul style="list-style-type: none"> <li>• Limite de 140 caracteres.</li> <li>• Nome do usuário.</li> <li>• Comentários anteriores.</li> </ul>	
Detalhes da fotos	<p>Serão exibidos os detalhes da foto:</p> <ul style="list-style-type: none"> <li>- Foto</li> <li>- Curtidas</li> <li>- Comentarios</li> </ul>	<ul style="list-style-type: none"> <li>• Número correto de curtidas e comentários.</li> <li>• Opção de remoção de fotos.</li> </ul>	

	<p>- Opção de remoção da foto</p> <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"> <li>• Possibilidade de curtir e comentar nas fotos.</li> </ul>	
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> <p>O usuário deve confirmar a exclusão.</p>	<ul style="list-style-type: none"> <li>• Exclusão da foto</li> <li>• Acessar a foto após a remoção.</li> <li>• Mensagem de confirmação em caso positivo.</li> <li>• Mensagem de cancelamento.</li> </ul>	
Pesquisa	<p>Ao digitar um texto na barra de pesquisa</p>	<ul style="list-style-type: none"> <li>• Pesquisa encontrou os</li> </ul>	

	<p>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>	<p>resultados esperados</p> <ul style="list-style-type: none"> <li>• Pesquisa não encontrou resultados</li> <li>• Sugestões de Pesquisa.</li> </ul>	
--	---	---	--

## 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 unitarios
Selenium	Qualidade	Ferramenta para testes end-to-end

## **5 Classificação de Bugs**

Os Bugs serão classificados com as seguintes severidades:

ID	Nível de Severidade	Descrição
1	Blocker	<ul style="list-style-type: none"><li>● Bug que bloqueia o teste de uma função ou feature causa crash na aplicação.</li><li>● Botão não funciona impedindo o uso completo da funcionalidade.</li><li>● Bloqueia release.</li></ul>
2	Critical	Feature funciona pobremente <ul style="list-style-type: none"><li>● Input esperados causam crash ou efeitos indesejados</li><li>● Input incomum causa efeitos irreversíveis</li></ul>
3	Major	Feature não atinge certos critérios de aceitação, mas sua funcionalidade em geral não é afetada <ul style="list-style-type: none"><li>● Mensagem de erro ou sucesso não é exibida</li><li>● Input incomum causar efeitos indesejados, mas contornáveis</li></ul>

4	Minor	Quase nenhum impacto na funcionalidade, mas ainda é um erro válido <ul style="list-style-type: none"> <li>• Erro ortográfico</li> <li>• Pequenos erros de UI</li> </ul>

## 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.