

Sua tarefa é desenvolver um aplicativo que atenda aos requisitos abaixo:

- **Requisito Funcional [RF-001]: Autenticar Usuário**
 - O acesso ao aplicativo será restrito a usuários previamente cadastrados na plataforma (Figura 01). Não será permitido realizar novos cadastros pelo aplicativo, pois todos os usuários são registrados exclusivamente na plataforma web. O nível do usuário é definido de acordo com a quantidade de publicações realizadas: Bronze (0 a 5 publicações), Prata (6 a 10 publicações) e Ouro (acima de 10 publicações).
- **Requisito Funcional [RF-002]: Visualizar Menu de Acesso**
 - Após a autenticação, o usuário deve ser direcionado a uma tela contendo as funcionalidades disponíveis: Minhas Fotos, Fotos Públicas, Meus Dados e Sair (Figura 02).
 - A funcionalidade Fotos Públicas deve ser exibida apenas para usuários com nível Prata ou Ouro.
- **Requisito Funcional [RF-003]: Meus Dados**
 - O aplicativo deve permitir que o usuário autenticado visualize suas informações pessoais, como nome e login, além da quantidade de publicações realizadas e do seu nível na plataforma (Figura 03).
- **Requisito Funcional [RF-004]: Fotos**
 - O aplicativo deve permitir que o usuário autenticado armazene suas fotos na nuvem, selecionando um arquivo do dispositivo ou capturando uma nova foto com a câmera do smartphone. Além disso, o usuário deve fornecer uma descrição e indicar se a foto será pública ou privada, enquanto o aplicativo registra automaticamente a localização geográfica no momento do envio. Este requisito garante que o usuário autenticado possa visualizar e excluir apenas as suas próprias fotos (Figura 04).
- **Requisito Funcional [RF-005]: Fotos Públicas**
 - O aplicativo deve permitir que o usuário autenticado, com nível Prata ou Ouro, visualize todas as fotos públicas (Figura 05).

Protótipos

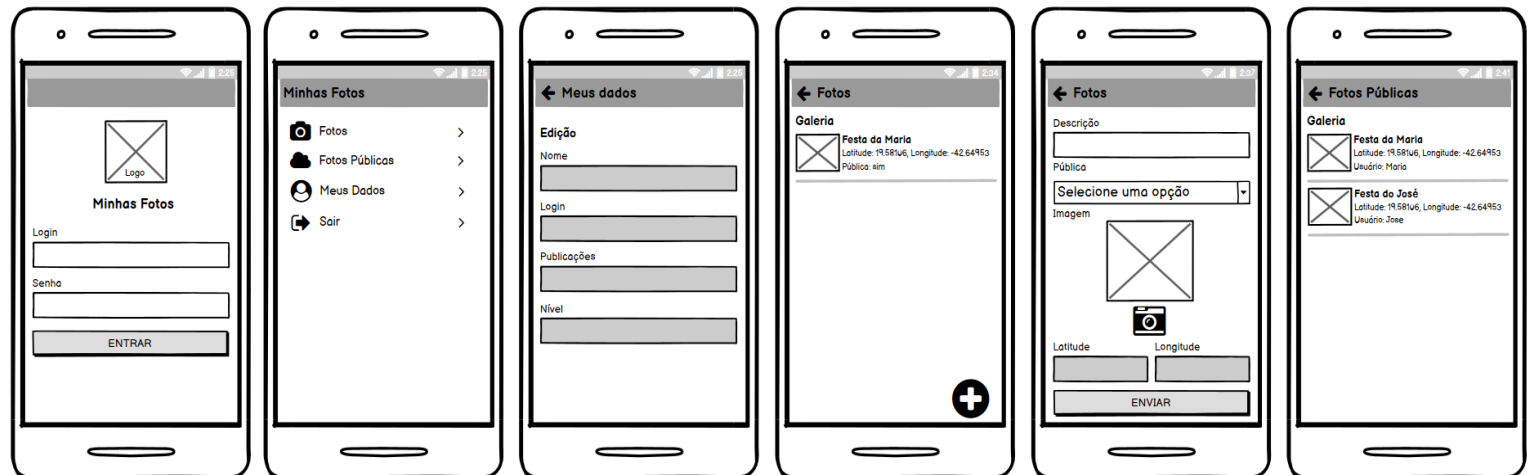


Figura 01 – Autenticação

Figura 02 – Menu de Acesso

Figura 03 – Meus Dados

Figura 04 – Fotos

Figura 05 – Fotos Públicas

- **Requisito Não Funcional [RNF-001]: Aplicação web**
 - Os dados dos usuários devem ser cadastrados por uma aplicação web disponibilizada pelo professor. <http://18.229.132.2/minhasfotos>



Figura 06 – Interface da aplicação web

- **Requisito Não Funcional [RNF-002]:** Consumir API

- O aplicativo deve obter e armazenar os dados na plataforma por meio da API disponibilizada pelo professor.

Autenticar usuário

POST	http://18.229.132.2:9090/usuarios/autenticar
------	--

O login e a senha devem ser enviados no corpo da requisição para autenticação do usuário. Em caso de sucesso, a estrutura JSON ao lado apresenta os dados retornados do usuário autenticado.

Requisição	Resposta
<pre>{ "login": "odilon", "senha": "123456" }</pre>	<pre>{ "id": 1, "nome": "Odilon Corrêa", "login": "odilon", "chave": "42662-1", "publicacoes": 0 "nivel": "Bronze" }</pre>

Observação: A chave é atualizada automaticamente toda vez que o usuário é autenticado.

GET	http://18.229.132.2:9090/usuarios
-----	-----------------------------------

No cabeçalho dessas requisições deve ser enviado:

- **Authorization:** {chave_do_usuario}

Resposta
<pre>{ "id": 1, "nome": "Odilon Corrêa", "login": "odilon", "chave": "42662-1", "publicacoes": 0 "nivel": "Bronze" }</pre>

Upload do arquivo

POST	http://18.229.132.2:9090/upload
------	---------------------------------

No cabeçalho dessas requisições deve ser enviado:

- **Authorization:** {chave_do_usuario}

Requisição	Resposta
<pre>{ "imagem": "data:image/jpeg;base64,iV.." }</pre>	<pre>{ "urlImagem": "http://18.229.132.2/.." }</pre>

Fotos

POST	http://18.229.132.2:9090/fotos
DELETE	http://18.229.132.2:9090/fotos/{id}
GET	http://18.229.132.2:9090/fotos
GET	http://18.229.132.2:9090/fotos/{id}

No cabeçalho dessas requisições deve ser enviado:

- **Authorization:** {chave_do_usuario}

O JSON abaixo ilustra a estrutura utilizada para representar uma foto:

```
{
  "id": "99d9eaa2-8d30-11f0-beb3-641c67d8195e",
  "descricao": "Foto da Maria",
  "latitude": -19.58106,
  "longitude": -42.64953,
  "urlImagem": "http://18.229.132.2:9090/uploads/68bfaa3768.16768396.jpg",
  "publica": 0
}
```

A propriedade **publica** aceita 0 (falso) ou 1 (verdadeiro).