



## **Missão Prática | Nível 1 | Mundo 4**

**Italo Augusto Juliano Barbosa - 202303617674**

**Campus Aparecida de Goiânia**

**Nível 1: Vamos Criar um App – 9001 – 4º Semestre**

**Link do repositório no GitHub:** <https://github.com/Anarquia122/trabalho-facul-criandoApp/tree/main>

### **Objetivo da Prática**

1. Configurar o ambiente de desenvolvimento React Native;
2. Implementar a funcionalidade de entrada de texto em um componente React Native;
3. Implementar um Componente de Lista Dinâmica (ScrollView);
4. Implementar componentes React Native para exibir informações de forma dinâmica em listas;
5. Empregar elementos visuais em um aplicativo React Native.

### **Contextualização:**

A empresa "Meeting" busca criar um aplicativo móvel eficaz para o cadastro de fornecedores, com listas e imagens de alta qualidade, economizando recursos e proporcionando uma excelente experiência ao usuário. A escolha da tecnologia React Native é crucial para estabelecer uma presença sólida no mercado móvel. Nesta atividade, você aprenderá os princípios básicos do React Native.

### **Missão Prática:**

De acordo com os requisitos funcionais, foi pedido que o app tivesse:

- 1- Cadastro dos fornecedores;
- 2- Listagem dos fornecedores cadastrados;
- 3- Upload de imagem para os fornecedores;
- 4- Interface intuitiva.

Então vamos analisar cada um desses requisitos no projeto.

## Upload:

Para criar essa funcionalidade utilizei o FireBase Storage para armazenar as imagens e coletar o endereço da mesma.

```
const uploadImage = async () => {
  if (imageUri) {
    setIsUploading(true);
    const fileName = imageUri.substring(imageUri.lastIndexOf('/') + 1);
    const cleanUri = imageUri.startsWith('file:///') ? imageUri.replace('file://', '') : imageUri;
    const storageRef = storage().ref(`images/${fileName}`);

    try {
      const task = storageRef.putFile(cleanUri);
      task.on('state_changed', (snapshot) => {
        console.log(`Progress: ${((snapshot.bytesTransferred / snapshot.totalBytes) * 100)}%`);
      });

      await task;
      const downloadUrl = await storageRef.getDownloadURL();

      setUploadStatus('Upload realizado com sucesso!');
      setIsUploading(false);

      if (onUploadComplete) {
        onUploadComplete(downloadUrl);
      }
    } catch (error) {
      setUploadStatus('Não foi possível fazer o Upload');
      setIsUploading(false);
    }
  }
};
```

Após a imagem ser selecionada ela aparecerá na tela junto com o botão de para que o usuário possa confirmar a imagem para fazer o upload para o FireBase Storage, caso

tenha ocorrido tudo certo no upload, o usuário recebe uma confirmação de que a imagem foi salva.

## Cadastro dos Fornecedores:

Para o cadastro utilizei o Realm para criar o banco de dados com as informações de cada fornecedor (codigo, nome, endereço, contato, categoria, imagem).

A imagem é coletada da seguinte forma, Após realizado o upload o endereço da imagem é coletado automaticamente e salvo na coluna **imagem** do banco de dados. Caso o usuário não insira nenhuma imagem, uma default aparecerá no lugar.

```
const salvar = () => {
  const ultimoCod = realm.objects('Fornecedor').max('codigo');
  const novoCod = (ultimoCod !== null ? ultimoCod : 0) + 1;

  const forneAux = { codigo: novoCod, nome, endereco, contato, categoria, imagem: imagem || imgDefault };
  realm.write(() => {
    realm.create('Fornecedor', forneAux);
  });

  navigation.navigate('Fornecedores');
}
```

## Lista dos Fornecedores:

Ao cadastrar um novo fornecedor, ele é adicionado ao banco de dados. Após isso, a página de lista, coleta essas informações e as apresentam na tela.

Nessa página o usuário tem as seguintes interações:

- 1- Botão para excluir o fornecedor;
- 2- Barra de pesquisa, no topo da tela;
- 3- Botão para filtrar por categoria, na parte inferior direita da tela.

```
const filteredFornecedores = fornecedores.filter(fornecedor => {
  const search = fornecedor.nome.toLowerCase().includes(searchQuery.toLowerCase());

  const filtro = categoryFilter === '' || categoryFilter === 'nenhum' || fornecedor.categoria === categoryFilter;
  return search && filtro;
});
```

## Resultados Finais:

Fornecedores

Search



1  
Dragon Games

Excluir

jogos

dragon@email.com

Casa do Kame



2  
Nemo

Excluir

computadores

nemo@email.com

Oceano pacífico



3  
Dragons Pc

Excluir

computadores

dragons@email.com

Caverna do dragão



4  
Italo

Excluir

jogos

italo@email.com

Rua do lado da outra rua



5  
Esponja Games

Excluir

jogos

Fornecedores

Novo Fornecedor

Novo Fornecedor



Confirme o Upload

Nome

Endereço

Contato (telefone/email)

Categoria:

Jogos

Salvar

Fornecedores

Novo Fornecedor