

DOCUMENTO “MEETING”

Desenvolvedor : André Luis Gonçalves Carvalhal

Matrícula: 202203185403

BRIEFING:

Cadastro de Fornecedores: O aplicativo deve permitir o cadastro de fornecedores, incluindo informações detalhadas, como nome, endereço, contato e categorias de produtos fornecidos. Essas informações serão exibidas utilizando componentes como `<Text>`, `<TextInput>`, e `<Image>`.

Listagem de Fornecedores: Deve ser possível visualizar uma lista de fornecedores cadastrados, com opções de pesquisa e filtragem com base em critérios como categoria ou localização. A lista de fornecedores será exibida utilizando componentes como `<Text>` e `<Image>`.

Associação de Imagens: O aplicativo deve permitir a associação de imagens aos perfis dos fornecedores. Os usuários devem poder fazer o upload de logotipos ou fotos relacionadas ao fornecedor, utilizando o componente `<Image>`.

Experiência de Usuário Intuitiva: A interface do aplicativo deve ser intuitiva e fácil de usar, garantindo que os usuários possam navegar, adicionar e editar informações de forma eficiente. Isso será alcançado usando componentes como `<Text>`, `<TextInput>`, e `<Image>`.

DEPENDÊNCIAS:

Node JS.

INICIALIZAÇÃO:

Acessar o diretório meeting e executar os seguintes comandos:

Para instalar os pacotes utilizados:

```
npm install
```

Para iniciar o projeto:

```
npx expo start
```

SOURCE:

\index.js - Inicia o aplicativo cria o navegador do projeto.

ENTIDADES

\app\entities\fornecedor.ts - Cria o objeto Fornecedor e define as propriedades do mesmo, implementa automaticamente uma chave única para a id caso não tenha sido fornecida na chamada do constructor.

REPOSITÓRIOS

\app\repositories\InMemoryFornecedor.ts - Cria um banco de dados em memória para o desenvolvimento e teste do aplicativo, implementa funções para criar novo fornecedor, atualizar fornecedor, localizar index dos fornecedores e deletar fornecedores.

\app\repositories\Cloudinary.js - Envia imagens para um bucket da cloudinary e retorna a url da imagem hospedada.

COMPONENTES

\app\components\cardFornecedor.jsx - recebe o id, nome e souce de um fornecedor e retorna o card Touchable de um fornecenedor, recebendo a linkando para a edição do fornecedor.

\app\components\galeria.jsx - recebe a lista de imagens de um fornecedor e retorna um array de Touchables para visualização das imagens.

\app\components\imageView.jsx - Visualiza as imagens e permite a remoção das mesma da lista de imagens.

\app\components\lista-fornecedores.jsx - Retorna um array de cardFornecedores, passando o nome, id e url da logo do fornecedor

\app\components\menu-button.jsx - Recebe o link de uma screen, um ícone e o nome e retorna um Touchable para fazer os botões da navegação.

\app\components\navigation.jsx - retorna o menu de navegação do app, com componentes do tipo menu-button.

\app\components\profilePhoto.jsx - Recebe o link da imagem, o tamanho e retorna a Image, caso não receba retorna um ícone de usuário.

TELAS

\app\screens\adicionar.jsx - *Retorna uma tela com campos para a adição de um fornecedor, permite a seleção de imagens da galeria do dispositivo para ser colocado no logo e/ou em uma galeria de imagens do fornecedor.*

\app\screens\visualizar.jsx - *Retorna uma tela com textos dos dados de um fornecedor, com sua logo caso exista ou um ícone de perfil e sua galeria de imagens caso existam.*

\app\screens\editar.jsx - *Recebe a id de um fornecedor e retorna uma tela com campos para a edição de um fornecedor, permite a seleção de imagens da galeria do dispositivo para ser colocado no logo e/ou em uma galeria de imagens do fornecedor.*

\app\screens\home.jsx - *Retorna à tela inicial do aplicativo, contendo um menu de navegação e uma lista com todos os fornecedores.*

\app\screens\buscaCategoria.jsx - *Retorna uma tela com um input para a digitação de uma categoria, apresenta uma lista com os fornecedores, mesmo com a digitação parcial da categoria.*

`\app\screens\buscaLocal.jsx` - Retorna uma tela com um input para a digitação da cidade ou estado, apresenta uma lista com os fornecedores que possuírem a cidade ou estado com os caracteres já digitados.

SCREENSHOTS





