

*Projeto Teste*

*Desenvolvimento de Aplicativos Móveis*

*MODULO 08*

# Introdução

O aplicativo revolucionário *EuVouNaTrip* oferece uma experiência de viagem incomparável, transformando suas jornadas em aventuras emocionantes e personalizadas. Com recursos inovadores como gamificação de pontos turísticos, roteiros sob medida baseados nas suas preferências, check-in simplificado por reconhecimento facial, ofertas exclusivas que adicionam um toque especial à sua jornada, avaliações e comentários úteis de outros viajantes, suporte ao cliente ágil e inteligentes recomendações de destinos autênticos e menos explorados, o *EuVouNaTrip* é o seu companheiro de viagem indispensável, garantindo que cada viagem seja memorável, fluida e enriquecedora.

# Descrição do projeto e tarefas

**API E BIBLIOTECAS**

* [*https://apieuvounatrip.azurewebsites.net/index.html*](https://apieuvounatrip.azurewebsites.net/index.html)
* Todas as bibliotecas permitidas na prova estão no repositório do GIT: https://github.com/Seletiva-Nacional-2023

**LOGO E GUIA DE ESTILO**

* A logo e manual do projeto estão disponíveis no seu computador no caminho: *C:\DataFiles\Guia de Estilos*
* As fontes para utilização no projeto estão disponíveis no seu computador no caminho: *C:\DataFiles\Fontes*

**CRIAÇÃO DO APLICATIVO**

* Você deverá criar o aplicativo conforme as instruções da próxima sessão e seguindo a guia de estilo e o manual de uso da marca prezando pela usabilidade e experiência do usuário.
* Quando necessário as telas poderão conter uma barra de rolagem na vertical. Nenhuma tela poderá possuir uma barra de rolagem na horizontal, logo, todos os itens deverão ser organizados para atender a este requisito;
* Os wireframes disponibilizados no conteúdo desta prova são sugestões, você pode aplicar a organização que julgar necessário, respeitando a guia de estilo e manual da marca além de prezar pela usabilidade e experiência do usuário.

**FORMATO DE ENTREGA E TEMPO DE EXECUÇÃO DA SESSÃO 03**

* Você terá 1:30h para realizar o projeto e efetuar suas entregas.
* A entrega deverá ser feita via GIT considerando as boas práticas de versionamento de código. Logo, você deverá realizar “commits” para cada funcionalidade com uma descrição do que foi desenvolvido até aquele ponto.

# Instruções para o Competidor

Você deverá fazer obrigatoriamente no mínimo um commit por funcionalidade desta sessão.

As principais funcionalidades dessa sessão são:

* Tela de Ofertas
* Lista de Ofertas
* Tela Mapa de Ofertas

**Tela de Ofertas**

A tela de ofertas deverá atender as funcionalidades abaixo, seguindo o layout apresentado na imagem ao lado.

No topo da tela, deve existir o texto “Ofertas” e ao seu lado esquerdo, deve possuir um ícone indicativo para voltar para a tela anterior conforme apresentado na imagem;

Ao clicar no texto ou no ícone, o utilizador ser retornado para a tela anterior;

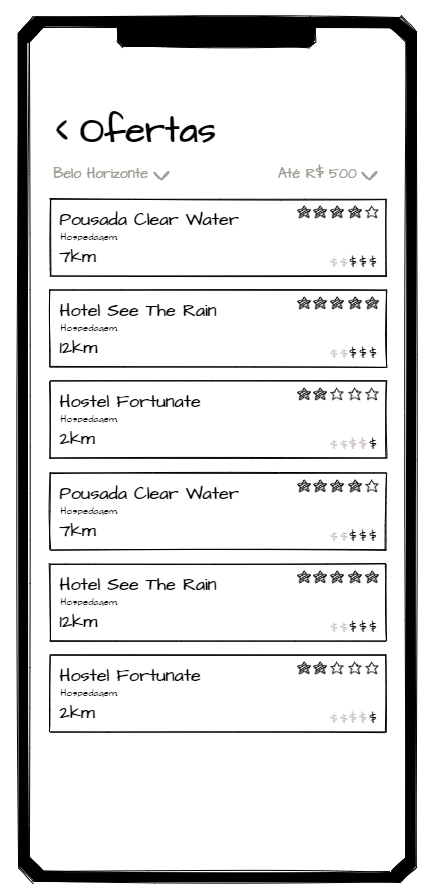
Esta tela deve possuir um mecanismo similar ao de um “carousel”, ou seja, um imagem que após um determinado tempo deve ser substituída por outra imagem, seguindo os seguintes parâmetros:

* O mecanismo deve funcionar com a imagem de 3 ofertas;
* Cada imagem deve permanecer visível por 5 segundos e ao término deste prazo, esta imagem deverá ser substituída pela próxima imagem;
* Quando chegar no fim do prazo da terceira imagem, o mecanismo deverá mostrar a primeira imagem novamente, reiniciando todo o ciclo;
* Deve haver uma transição entre as imagens, da direita para a esquerda;
* Os três pontos abaixo da imagem devem representar qual imagem está sendo mostrada, ou seja, caso a primeira imagem esteja sendo exibida, o primeiro ponto deverá ser totalmente preenchido, caso seja a segunda, o segundo ponto deverá ser totalmente preenchido e caso seja a terceira, o terceiro ponto deverá ser totalmente preenchido;
* Enquanto um ponto está totalmente preenchido, os demais deverão permanecer sem preenchimento, mostrando apenas sua borda.

Em seguida, deverá possuir um controle com o texto “Ver todas as ofertas no mapa” que ao ser clicado, enviará o utilizador para a tela de “Mapa das ofertas”;

Na sequência, deverão existir controles para buscar um grupo de ofertas com base em filtros, obedecendo às seguintes regras:

* Deve possuir um título e um controle do tipo DropDown ou similar onde seja possível selecionar um estado entre uma determinada lista de estados;
* A lista de estados deverá ser importada através do arquivo “listaestadoscidades.json” fornecido no datafiles;
* Deve possuir também um título e um controle do tipo DropDown ou similar onde seja possível selecionar uma cidade entre uma determinada lista de cidades;
* O elemento da lista de cidades só deve estar ativo quando a lista de estados tiver um item selecionado;
* A lista de cidades deverá conter apenas cidades correspondentes ao estado selecionado no controle anterior;
* A lista completa de cidades deverá ser importada através de um arquivo “listaestadoscidades.json” fornecido no datafiles, respeitando o tópico anterior;
* Deve possuir um controle do tipo DropDown ou similar onde seja possível selecionar 5 opções de valores, sendo: “R$ 500”, “R$ 300”, “R$ 200”, “R$ 100” e “Sem limite”;
* Na sequência deve existir um botão “Buscar” que deve enviar os dados selecionados para a tela de “Lista de ofertas”.

**Tela de Lista de Ofertas**

A tela de lista de ofertas deverá atender as funcionalidades abaixo, seguindo o layout apresentado na imagem ao lado.

No topo da tela, deve existir o texto “Ofertas” e ao seu lado esquerdo, deve possuir um ícone indicativo para voltar para a tela anterior conforme apresentado na imagem;

Ao clicar no texto ou no ícone, o utilizador ser retornado para a tela anterior;

Deve possuir um controle do tipo DropDown ou similar para realizar a troca da cidade selecionada, este controle deverá mostrar apenas cidades relacionadas ao estado selecionado na tela anterior, utilize as cidades presentes no arquivo “listaestadoscidades.json” fornecido no datafiles. Este controle deverá ser iniciado com o nome da cidade selecionado na tela anterior;

Deve também possuir um controle do tipo DropDown ou similar para realizar a alteração do valor máximo para o filtro. Este controle deverá ser iniciado com o valor selecionado na tela anterior;

Ao realizar a atualização de algum dos dois controle “DropDown”, devem ser atualizados os valores da lista respeitando os novos valores;

Logo em seguida, deve existir uma lista onde cada item respeite as seguintes características:

* Deve possuir um título em texto destacado representando o nome do local;
* Deve possuir um subtítulo em texto menor representando o tipo de local;
* Logo abaixo deve possuir um texto indicando a distância do local atual do dispositivo até o local indicado na lista, onde:
  + Caso seja maior ou igual a um quilômetro, deve ser mostrado em quilômetros (KM), caso seja menor que um quilômetro, deve ser representado em metros;
  + Não deve apresentar números decimais, podendo ser arredondado ou “truncado”.
* A direita do título, deve existir um controle do tipo “rating” ou similar para representar as avaliações do local. Este controle deverá ser preenchido conforme o índice de avaliação fornecido pela API;
* Logo abaixo deve existir um controle do tipo “rating” ou similar para representar a média de custo do local. Este controle deverá ser preenchido conforme o índice fornecido pela API.

**Tela de Mapa das Ofertas**



Imagem 03 - Tela do Mapa de Ofertas

A tela de lista de ofertas deverá atender as funcionalidades abaixo, seguindo o layout apresentado na imagem 03.

No topo da tela, deve existir o texto “Ofertas” e ao seu lado esquerdo, deve possuir um ícone indicativo para voltar para a tela anterior conforme apresentado na imagem;

Ao clicar no texto ou no ícone, o utilizador ser retornado para a tela anterior;

Deve possuir um mapa seguindo as seguintes características:

* Deve mostrar a localização atual do dispositivo ao centro do mapa, para o ícone que representa a posição atual do usuário deve ser utilizado o arquivo “userLocation.svg” presente no datafiles;
* Deve mostrar a localização dos usuários que estão em um raio de alcance de 2km do utilizador, para o ícone que representa a posição destes outros usuários, deve ser utilizado o arquivo “otherUsersLocation.svg” presente no datafiles;
* Deve mostrar a localização dos locais que estão fornecendo a oferta e que estão em um raio de alcance de 2km do utilizador, para o ícone que representa a posição destes outros usuários, deve ser utilizado o arquivo “placeLocation.svg” presente no datafiles;
* No canto inferior direito do mapa, deve existir um controle para centralizar o mapa utilizando como referência o local atual do dispositivo;
* Ao clicar em um ícone no mapa que represente um local, deve ser exibida uma popup com o nome e a avaliação do local conforme a imagem 04;
* Ao clicar em um ícone no mapa que represente outro utilizador do sistema, deve ser exibida um popup com o nome deste utilizador apenas, conforme imagem 05.

Deve possuir uma lista de locais recomendados onde cada item deve:

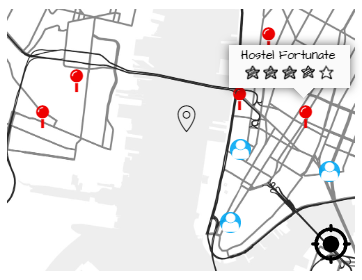


Imagem 04 - Popup do Local no Mapa

* Possuir o nome do local;
* O tipo de local;
* A distância até o local;
* A avaliação do local.

Ao clicar em um item na lista, deve ser exibida um popup em cima do ponto relacionado no mapa conforme imagem 04.

Imagem 04 - Popup do Local no Mapa

