

Universidade Federal de Viçosa – Departamento de Informática
INF311 – Programação para dispositivos móveis
Prof. Lucas Francisco da Matta Vegi

- Atividade prática 05 -

Atualmente existem diversos aplicativos móveis que nos permitem fazer check-in em determinados locais e assim compartilhar com amigos lugares que estamos ou visitamos ao decorrer de nossas vidas. O objetivo desta atividade prática será criar uma aplicação que permita ao usuário fazer check-in em determinados locais que ele visitar, armazene-os localmente no banco de dados e exiba-os como pontos em um mapa.

Maiores detalhes e dicas sobre a implementação deste aplicativo serão apresentados a seguir, portanto **LEIA TODO O ROTEIRO ANTES DE COMEÇAR A ATIVIDADE.**

Para o desenvolvimento deste aplicativo, vários conceitos aprendidos em sala de aula serão necessários:

- API de mapas
- API de localização
- Navegação entre telas (com e sem envio de parâmetros)
- Componentes de GUI
- Persistência com SQLite

O banco de dados da aplicação desenvolvida neste roteiro deverá ter o formato e dados apresentados a seguir:

```
CREATE TABLE Checkin (Local TEXT PRIMARY KEY, qtdVisitas INTEGER NOT NULL, cat INTEGER NOT NULL, latitude TEXT NOT NULL, longitude TEXT NOT NULL, CONSTRAINT fkey0 FOREIGN KEY (cat) REFERENCES Categoria (idCategoria));

CREATE TABLE Categoria (idCategoria INTEGER PRIMARY KEY AUTOINCREMENT, nome TEXT NOT NULL);

INSERT INTO Categoria (nome) VALUES ('Restaurante');
INSERT INTO Categoria (nome) VALUES ('Bar');
INSERT INTO Categoria (nome) VALUES ('Cinema');
INSERT INTO Categoria (nome) VALUES ('Universidade');
INSERT INTO Categoria (nome) VALUES ('Estádio');
INSERT INTO Categoria (nome) VALUES ('Parque');
INSERT INTO Categoria (nome) VALUES ('Outros');
```

O aplicativo será composto por 4 telas e deverá ter **API LEVEL mínima 21**:

- Tela Principal:
- Mapa de Check-In:

- Gestão de Check-In:
- Relatório de lugares mais visitados:

Tela Principal

A tela principal da aplicação será responsável por obter as informações do local onde o usuário deseja fazer check-in e persisti-las no banco de dados. Uma formatação possível para a criação desta tela é apresentada na Figura 1.

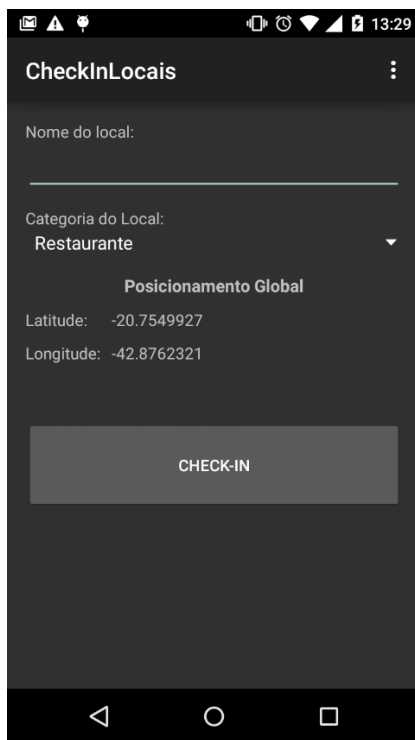


Figura 1 - Tela Principal

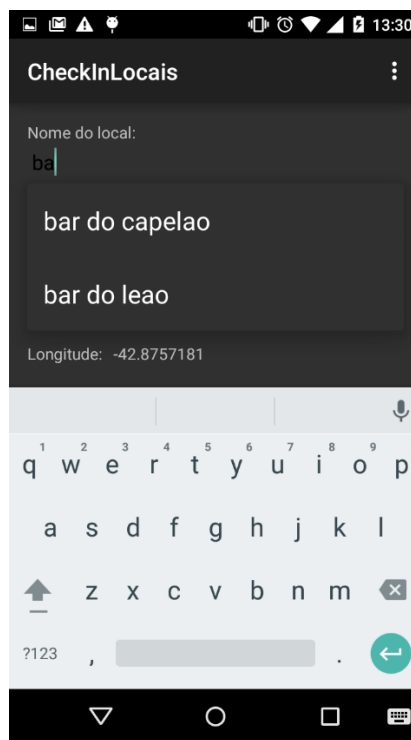


Figura 2 - Tela Principal com AutoComplete

No formulário da “Tela principal”, o campo com o nome do local visitado deverá ser um `AutoCompleteTextView` que pegue no banco de dados os nomes referentes aos locais que o usuário já visitou no passado (Figura 2).

A categoria do local visitado deverá ser um `Spinner` preenchido com os nomes das categorias cadastradas no banco de dados. Para facilitar a implementação, recupere o nome de todas as categorias pela ordem crescente da chave primária desta tabela.

O valores referentes ao posicionamento global do usuário serão recuperados automaticamente pelo dispositivo móvel e atualizados na tela em tempo real sempre que uma nova localização for obtida.

Quando o usuário clicar no botão check-in, duas possíveis operações poderão acontecer (apresentadas a seguir). Após uma das duas operações terem sido realizadas, a Tela Principal deverá ser fechada e aberta automaticamente para dar ao usuário a sensação de atualização da tela:

1- Inserir um novo check-in na tabela Checkin:

Se o nome do local informado for diferente do nome de todos os locais já visitados anteriormente, um novo registro será adicionado à tabela Checkin, tendo este registro a `qtdVisitas = 1`.

OBS: é pré-requisito para esta operação, que o usuário tenha digitado o nome de um local, escolhido uma categoria e o aparelho tenha obtido a posição do usuário. Caso algum destes campos não tenha valores, informe ao usuário.

2- Atualizar um registro da tabela Checkin.

Se o nome do local informado pelo usuário já existir, ao invés inserir um novo registro na tabela Checkin, o registro referente a este local será atualizado, tendo este registro a `qtdVisitas` acrescida em 1.

OBS: Quando for fazer update, desconsiderar a categoria escolhida na interface e o novo posicionamento global do usuário.

A tela principal deverá ter um menu que permita ao usuário três opções (Figura 3):

- 1- ir até um mapa para ver todos os locais onde ele fez check-in;
- 2- ir até a gestão de check-in;
- 3- ir até o relatório de locais mais visitados.

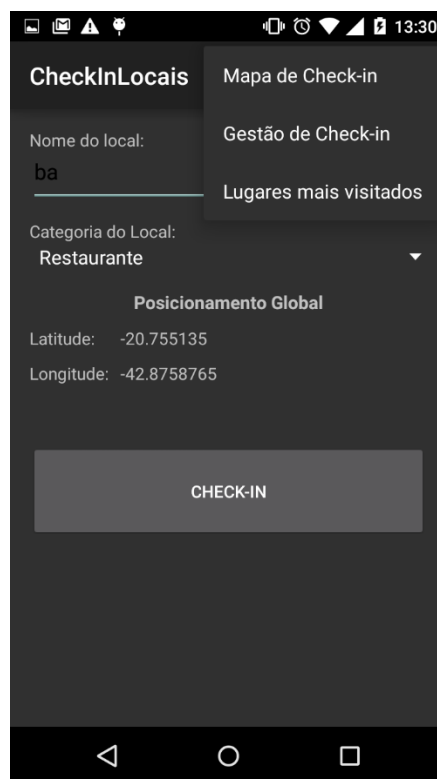


Figura 3 - Menu da Tela Principal

Para o menu aparecer corretamente em sua aplicação, faça todas as telas herdarem de `AppCompatActivity`. Além disso, altere o arquivo `styles.xml` (ou

equivalente), presente em `/res/values/` do projeto da sua aplicação e use um dos seguintes estilos de telas:

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="ThemeOverlay.AppCompat.Dark.ActionBar">
        OU
    <style name="AppTheme" parent="Theme.AppCompat">
        <!-- Customize your theme here. -->
        <item name="windowActionBar">true</item>
        <item name="android:windowNoTitle">false</item>
        <item name="android:showAsAction">always</item>
    </style>
</resources>
```

Mapa de Check-in

A tela de mapa de check-in deverá exibir cada local já visitado pelo usuário como um marcador posicionado na sua respectiva posição global. Estes marcadores deverão ter como *title* no nome do local que ele representa e como *snippet* o nome da categoria do local acompanhado pela quantidade de vezes em que o mesmo foi visitado (Figura 4).

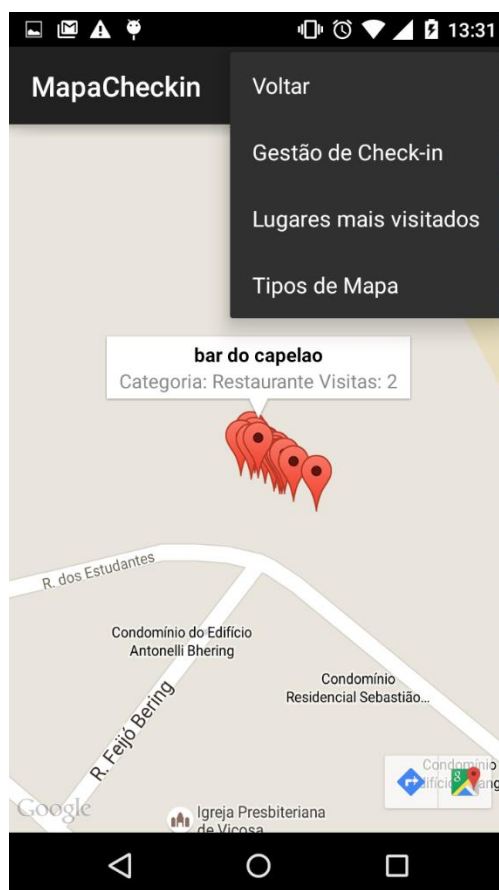


Figura 4 - Marcadores do Mapa de check-in

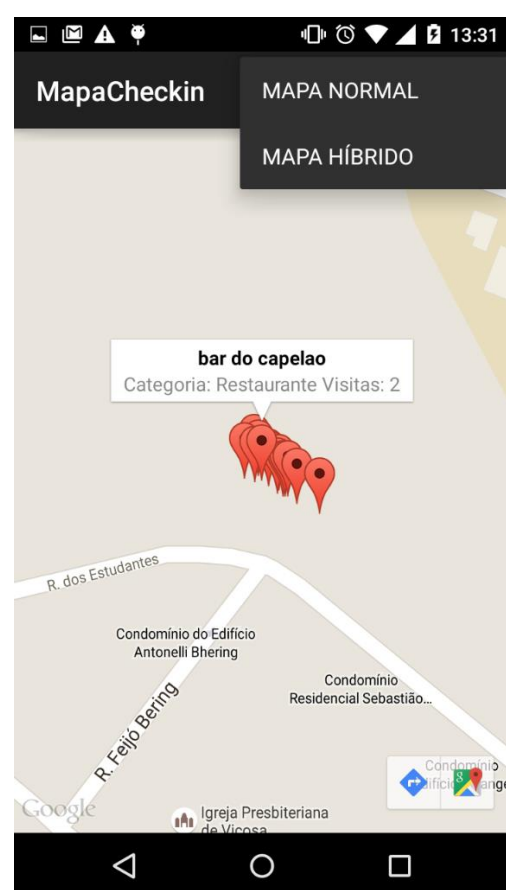


Figura 5 - Submenu do Mapa de check-in

Só deverá ser possível navegar até a tela de “*mapa de Check-in*” por meio do menu da “*Tela principal*”. O “*mapa de check-in*” deve ser aberto centralizado na posição global atual do usuário, logo a aplicação deve garantir que a “*Tela Principal*”

navegue até o mapa levando como parâmetros os valores de latitude e longitude atuais. Caso a “*Tela Principal*” ainda não tenha obtido a posição atual do usuário, a aplicação não deve permitir a navegação até o mapa.

O mapa deverá ter um menu contendo 4 itens (Figura 4):

- 1- Voltar a Tela Principal
- 2- ir até a gestão de check-in;
- 3- ir até o relatório de locais mais visitados.
- 4- Tipos de Mapa
 - a. Este item na verdade deverá ser um submenu. Neste submenu deverá ser possível alterar o tipo do mapa exibido para híbrido ou normal (Figura 5).

Gestão de Check-In

Na tela de gestão de check-in será possível o usuário visualizar uma lista com todos os lugares que ele já visitou (*TextViews*) acompanhados por botões (*ImageButton*s) que permitem deletar cada um destes lugares (Figura 6).

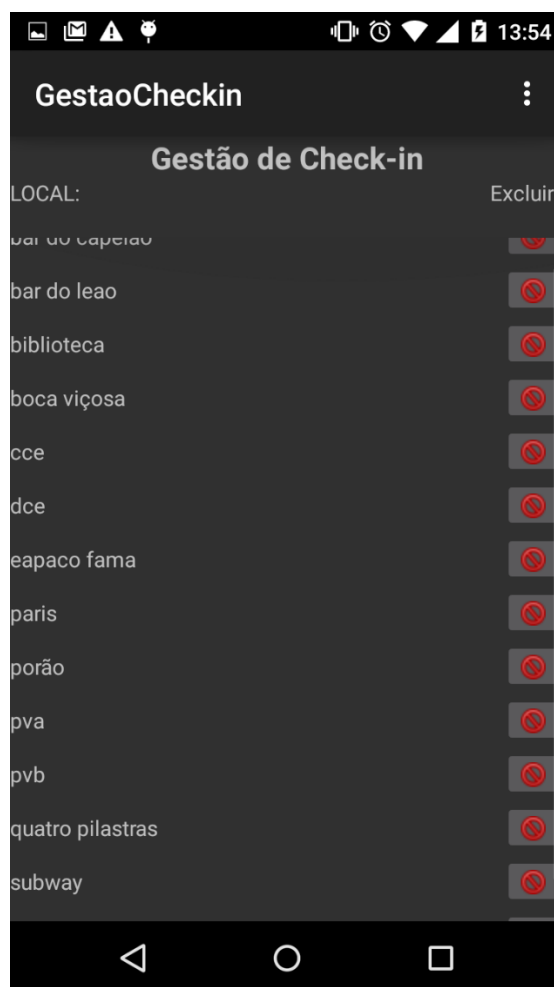


Figura 6 - Tela de Gestão de Check-in

Para conseguir colocar os *ImageButtons* do lado das suas respectivas *TextViews* na lista, a view que desenha esta tela poderá utilizar *LinearLayouts* aninhados, com um *ScrollView* abraçando à todos (Figura 7). O uso do *ScrollView* vertical será importante pois o tamanho desta lista é imprevisível, podendo portanto não caber na tela do dispositivo em que a aplicação for utilizada.

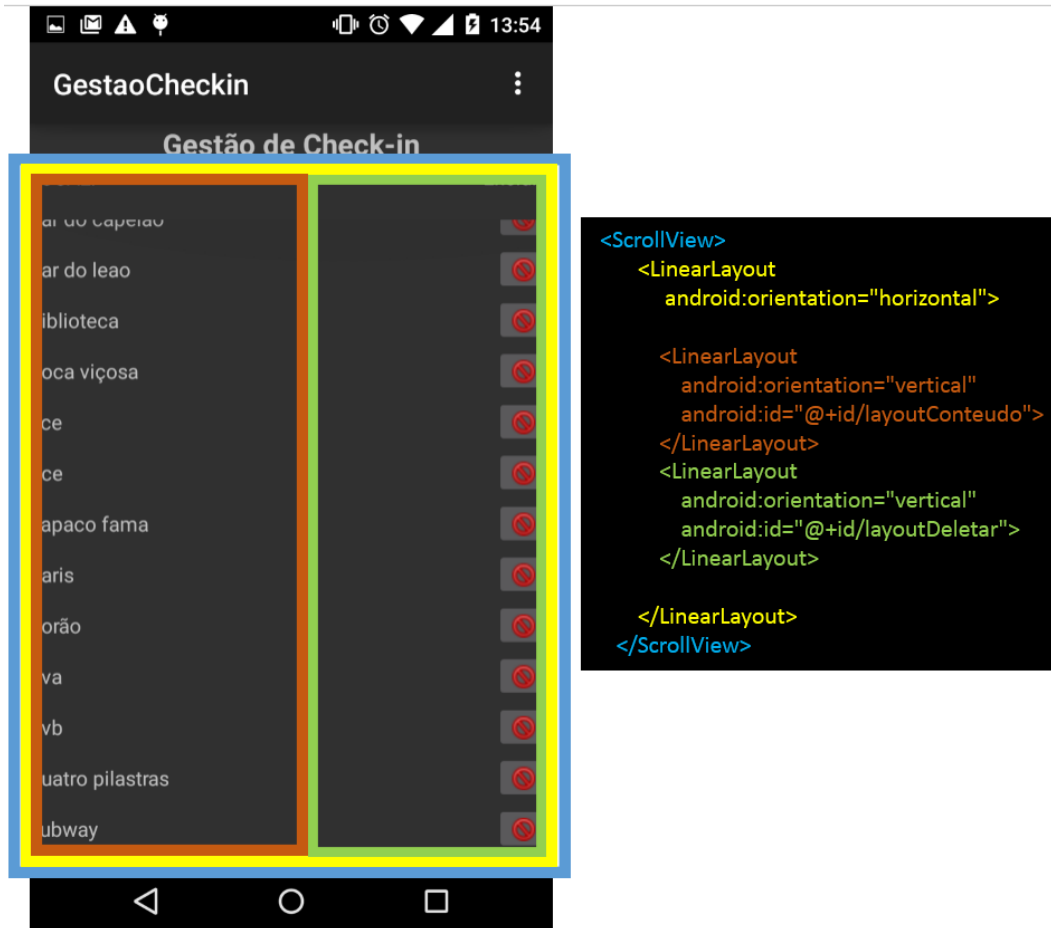


Figura 7 - Esboço da estrutura da lista de Check-in

Os *TextViews* com o nome de cada local visitado serão adicionados dinamicamente pela classe de controle desta tela no layout interno de id *layoutConteudo* (Figura 7). Já os *ImageButtons* que podem deletar cada um dos locais visitados serão adicionados dinamicamente pela classe de controle desta tela no layout interno de id *layoutDeletar* (Figura 7).

Um único método irá tratar o evento de clique de todos os botões, portanto cada botão adicionado dinamicamente pela classe de controle deverá ter o atributo *tag* diferente.

OBS:

- A tag de cada botão poderá ser a chave primária do registro do banco que ele pode deletar
- Para configurar o método de clique de cada botão, será necessário utilizar a primeira forma de tratar eventos de clique estudada em sala.
- A deleção deverá solicitar a confirmação do usuário (Figura 8)
- Caso um registro seja deletado, a tela de gestão deverá ser fechada e aberta automaticamente para dar ao usuário a sensação de atualização da tela.
- A Tela de Gestão deverá ter um menu contendo só a opção Voltar à tela anterior (Figura 9)

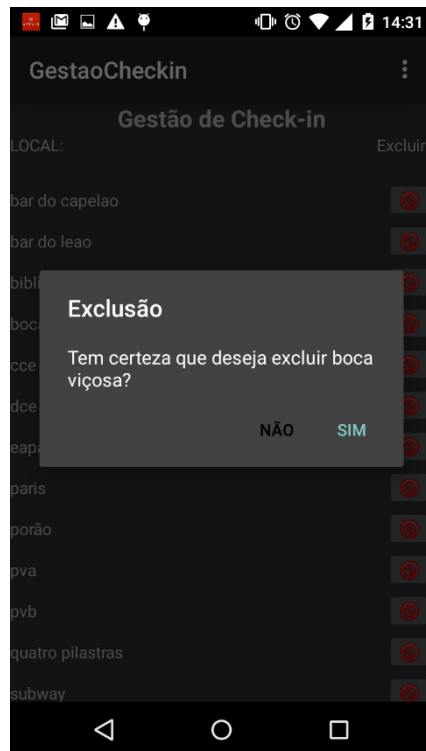


Figura 8 – Confirmação de Exclusão

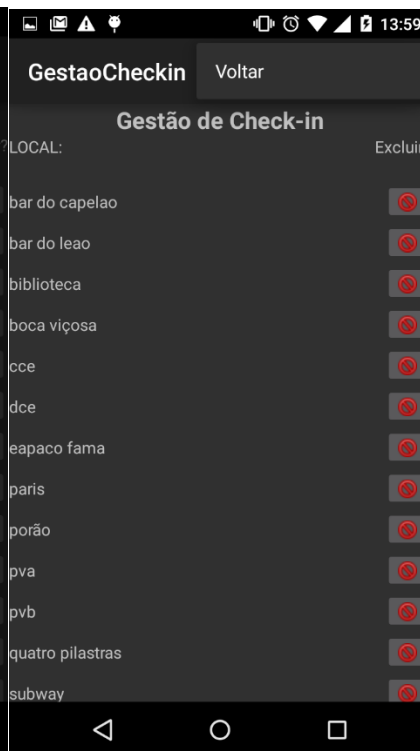
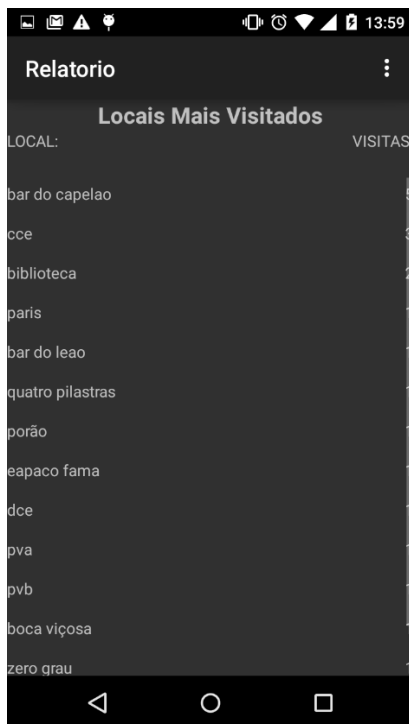


Figura 9 – Menu da Tela de gestão de check-in

Relatório de lugares mais visitados

Na tela de relatório de lugares mais visitados será possível o usuário visualizar uma lista com todos os lugares que ele já visitou (*TextViews*) acompanhados pela quantidade de vezes que o usuário visitou cada lugar (*TextViews*). A lista deverá estar ordenada de forma decrescente em relação à quantidade de visitas a cada local, tendo portanto o lugar mais visitado no topo e o menos visitado por último (Figura 10).



Relatorio

Locais Mais Visitados

LOCAL:	VISITAS:
bar do capelao	5
cce	3
biblioteca	2
paris	1
bar do leao	1
quatro pilastras	1
porão	1
eapaco fama	1
dce	1
pva	1
pvb	1
boca viçosa	1
zero grau	1

Figura 10 – Relatório de Lugares mais visitados



Relatorio **Voltar**

Locais Mais Visitados

LOCAL:	VISITAS:
bar do capelao	5
cce	3
biblioteca	2
paris	1
bar do leao	1
quatro pilastras	1
porão	1
eapaco fama	1
dce	1
pva	1
pvb	1
boca viçosa	1
zero grau	1

Figura 11 – Menu da Tela de Relatório

OBS:

- A Tela de Relatório deverá ter um menu contendo só a opção Voltar à tela anterior (Figura 11)
- A estrutura de organização das informações na tela de relatório poderá utilizar da mesma ideia apresentada para a tela de gestão de check-in (Figura 7)
- Para um melhor entendimento sobre o comportamento do aplicativo, instale o arquivo .apk disponibilizado pelo PVAnet em seu dispositivo móvel ou emulador Android.

ENTREGA DAS ATIVIDADES:

Ao final da atividade, suba o seu projeto para um repositório público do GitHub e informe a URL desse repositório pelo PVAnet Moodle.