=======================================================

***Programação Para Dispositivos Moveis***

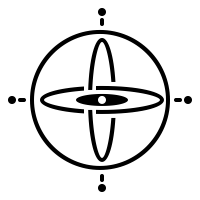
=======================================================

Nessa Prova irá ser abordado os temas Referentes aos Trabalhos apresentados pelos Grupos, o conteúdo desse arquivo é um **COMPILADÃO** dos mesmos.

=======================================================

***Acelerômetro***

=======================================================

O acelerômetro do celular capta movimentos através de um sensor com massas sensíveis à aceleração.

Quando o dispositivo é acelerado, as massas se movem em relação ao sensor, gerando sinais elétricos que serão capturados pelo celular

Os dados coletados permitem que o aparelho celular realize algumas ações quando esses valores de coordenada chegam a certo padrão ou constante.

Os acelerômetros são compostos por:

* **Massa Sísmica:** É uma massa interna suspensa por molas dentro do acelerômetro. Ela se move em resposta à aceleração aplicada ao dispositivo.
* **Molas de Suspensão:** Fornecem a força de restauração para a massa sísmica, permitindo que ela se mova livremente e a mantenha centralizada quando em repouso.
* **Capacitores Variáveis:** Consistem em uma placa fixa e outra ligada à massa sísmica. Com o deslocamento da massa sísmica, a distância entre as placas muda.
* **Piezoelementos:** Com a aceleração, a massa sísmica pressiona os piezoelementos, gerando uma carga elétrica proporcional à força mecânica aplicada.
* **Circuitos Eletrônicos:** Processam os sinais gerados pelos outros componentes, amplificando, filtrando e convertendo os sinais analógicos em um formato digital.

=======================================================

***Persistência com SQLite (CRUD)***

=======================================================

**Aplicação Cliente/Servidor:** São arquiteturas de aplicação distribuídas, possuindo uma rede de fornecedores de recursos (servidores) e requerentes dos recursos oferecidos (clientes), utilizam bancos de dados para armazenar suas informações.

**SQLite:** formado por uma biblioteca em linguagem C, funciona como um servidor próprio e independente, não sendo necessário instalá-lo ou configurá-lo, pois gerencia os dados diretamente do seu sistema de arquivos. Não possui licença, é leve e recomendado para aplicativos desktop/mobile simples, sites leves sem muitos recursos e para sistemas com poucos usuários.

# **SQLite *vs* MySQL**

**SQLite:**

* Opera de forma independente;
* Consultas mais rápidas;
* Estável, multiplataforma e compatível com versões anteriores;
* Código Open Source;
* Não necessita de instalação ou configuração, guarda todo o banco de dados em um só arquivo;

**MySQL:**

* Suporta mais tipos de dados;
* Suporta uma maior quantidade de dados;
* Segurança melhor;
* Possui integração com a nuvem e roda em diversos sistemas operacionais;

**ImageCropper:** Dependência/Biblioteca para Android oferecem funcionalidades de recorte de uma imagem da galeria ou tirar uma nova com a câmera do aparelho.

**CircleImageView:** Dependência/Biblioteca para Android que estende a classe ImageView para exibir imagens em formato circular, adicionando um efeito de máscara à imagem original.

**ScalableJava:** Envolve projetar o sistema de forma que ele possa lidar com contratempos como o aumento na demanda/carga de usuários, dados e transações sem comprometer o desempenho ou a estabilidade.

=======================================================

***Câmera e Galeria de Imagens***

=======================================================

**Picasso:** Dependência/biblioteca de download de imagens que simplifica o processo de carregamento de imagens de URLs externas e as exibe em seu aplicativo. Reduz a quantidade de linhas de código necessárias para realizar tarefas.

**CameraX:** Dependência/biblioteca do Jetpack criada para facilitar o desenvolvimento de apps de câmera, dando suporte à visualização, análise (acesso ao buffer) e captura de imagens e vídeos.

**Permissões:**

* CAMERA (Acessar a Câmera do Dispositivo)
* WRITE\_EXTERNAL\_STORAGE (Escrever no Armazenamento Interno do Dispositivo)
* READ\_EXTERNAL\_STORAGE (Ler o Armazenamento Interno do Dispositivo)

**Feature:**

* CAMERA (Indica que o Aplicativo vai usar a Câmera, e define ela como Obrigatória para o funcionamento do mesmo)
* Atributos MAIN:
* REQUEST\_CAMERA\_PERMISSION (Indica a Permissão de uso da Câmera)
* REQUEST\_IMAGE\_CAPTURE (Indica a Captura da Imagem)
* REQUEST\_IMAGE\_SELECT (Indica a Imagem Selecionada)
* REQUEST\_TAKE\_PHOTO (Indica a Foto Tirada)
* caminhoDaFoto (Indica o Caminho da Foto Tirada)
* imageView (Variável da Interface usada para Mostrar a Imagem)
* imagemCapturada (Mapa de Pixels, variável que irá guardar a imagem)

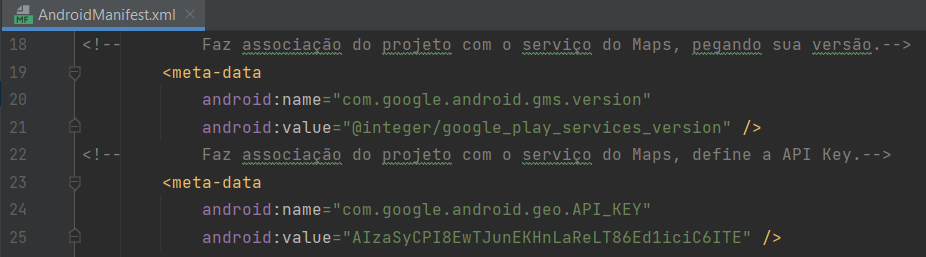
**Provider:** Declarar um provedor para que outras aplicações tenham acesso aos dados da aplicação, ou o sistema não saberá da existência deles e eles não serão executados.

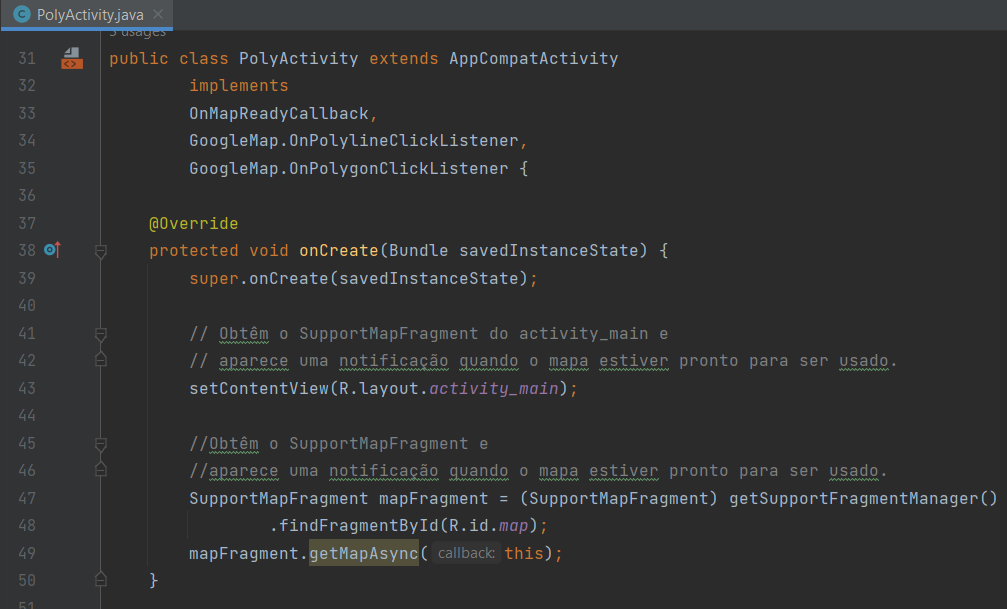
=======================================================

***GPS e Maps***

=======================================================

Foi necessário realizar um cadastro na plataforma Google Maps para conseguir uma key.

No AndroidManifest.xml, foi feita associação do projeto com o serviço do Maps, pegando a versão do Android e definindo a API Key.

No método onCreate, foi Setado um ContentView para trazer o Fragmento do mapa.

Em seguida, o método que adiciona as Polilinhas ao Mapa.

