

# MAC0209 — EP 2 - 2023

Roberto Marcondes Cesar Jr.

30 de maio de 2023

## Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Contexto . . . . .	2
1.2	Plágio e cópia de EPs . . . . .	2
<b>2</b>	<b>Descrição</b>	<b>2</b>
2.1	Kivy . . . . .	3
2.2	Buildozer . . . . .	5
2.3	Experimentos . . . . .	6
<b>3</b>	<b>Entrega</b>	<b>7</b>
3.1	Relatório de entrega e vídeo . . . . .	7

# 1 Introdução

## 1.1 Contexto

A disciplina de Modelagem e Simulação do curso de Bacharelado em Ciência da Computação tem como objetivo principal que o aluno se familiarize com a modelagem de sistemas físicos reais e seja capaz de simulá-los através da implementação de algoritmos. A capacidade de modelar e simular sistemas usando modelos matemáticos e algoritmos é fundamental para a atuação em ciência de dados e descoberta do conhecimento, permitindo a adoção de abordagens similares em diferentes domínios como informática urbana, saúde, sistemas financeiros e outros.

A disciplina tem uma parte teórica e uma prática e é essa que nos interessa neste documento. A prática é cobrada a partir de exercícios programa (*EPs*) que são feitos por equipes de 4 (**mínimo**) a 6 (**máximo**) estudantes.

## 1.2 Plágio e cópia de EPs

Plágio é a copia/modificação não autorizada e/ou sem o conhecimento do autor original. O plágio é um problema grave que pode levar até a expulsão do aluno da universidade. Leia o Código de Ética da USP (em particular, a seção V): [http://www.mp.usp.br/sites/default/files/arquivosanexos/codigo\\_de\\_etica\\_da\\_usp.pdf](http://www.mp.usp.br/sites/default/files/arquivosanexos/codigo_de_etica_da_usp.pdf).

Além do problema do plágio, cada grupo deve escrever seus próprios programas, realizar seus próprios experimentos e redigir seus respectivos relatórios. A cópia de EPs entre grupos implicará na anulação da nota de todos os envolvidos. O caso poderá ser levado à Comissão de Graduação do Instituto.

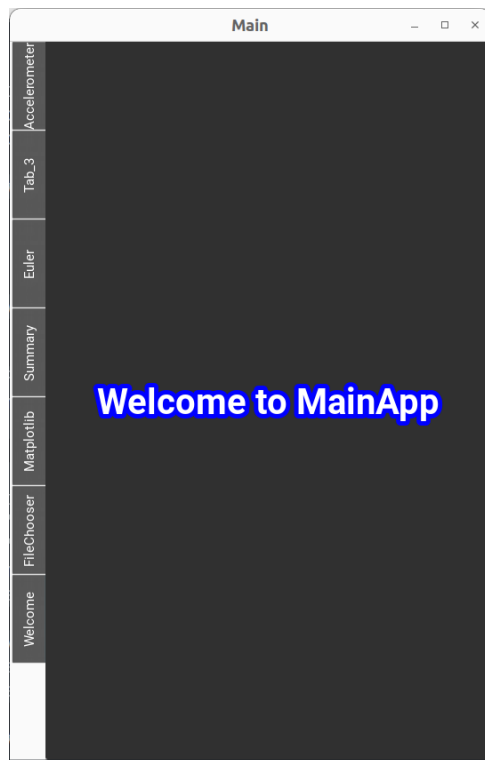
# 2 Descrição

O segundo EP retoma os objetivos do EP1 mas, desta vez, usando programação em dispositivos móveis. No EP1, os dados dos experimentos foram obtidos pelo Physics Toolbox, exportados e analisados em Jupyter Notebooks. O EP2 também consiste em realizar, relatar e simular experimentos de pelo menos dois dos movimentos descritos no primeiro EP. No EP2, cada grupo deve implementar um aplicativo que capture os dados do acelerômetro e realize a mesma análise proposta no EP1. Mas, o processamento deve ser executado no próprio celular. Os resultados também devem ser visualizados no celular. Para tal serão utilizadas as ferramentas **Kivy**<sup>1</sup> e **Buildozer**<sup>2</sup>.

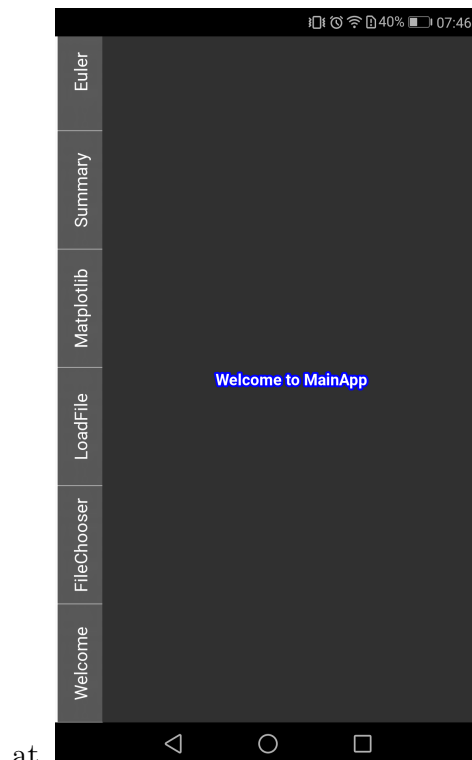
---

<sup>1</sup><https://kivy.org/>

<sup>2</sup><https://buildozer.readthedocs.io/en/latest/>



(a) Execução do Kivy em Python



at

(b) Execução do Kivy no celular

Figura 1: Captura de tela do mesmo aplicativo em duas plataformas diferentes

## 2.1 Kivy

Kivy é uma biblioteca Open Source para Python que permite o desenvolvimento de aplicações multiplataformas (Windows, Linux, iOS, por exemplo). A ideia dessa biblioteca é permitir que o desenvolvedor crie um aplicativo uma vez e o use em todos os dispositivos, tornando o código reutilizável e implantável, permitindo um design de interação rápido e fácil e uma prototipagem rápida. Também oferece implementações em sua "própria linguagem" (arquivo com extensão .kv), permitindo a criação de uma estrutura em árvore do layout da aplicação. Isso auxilia na organização do código, separando sua funcionalidade com o design do aplicativo.

Na construção do EP, será **obrigatório** a implementação dos itens abaixo na aplicação:

### 1. Captura de dados do acelerômetro:

A aplicação deverá ser capaz de ler os dados do acelerômetro e armazená-los. Os dados obtidos serão utilizados como fonte para os experimentos a serem estudados. Além da obtenção dos dados, você deve exibi-los em **tempo real**.

### 2. Processamento de dados:

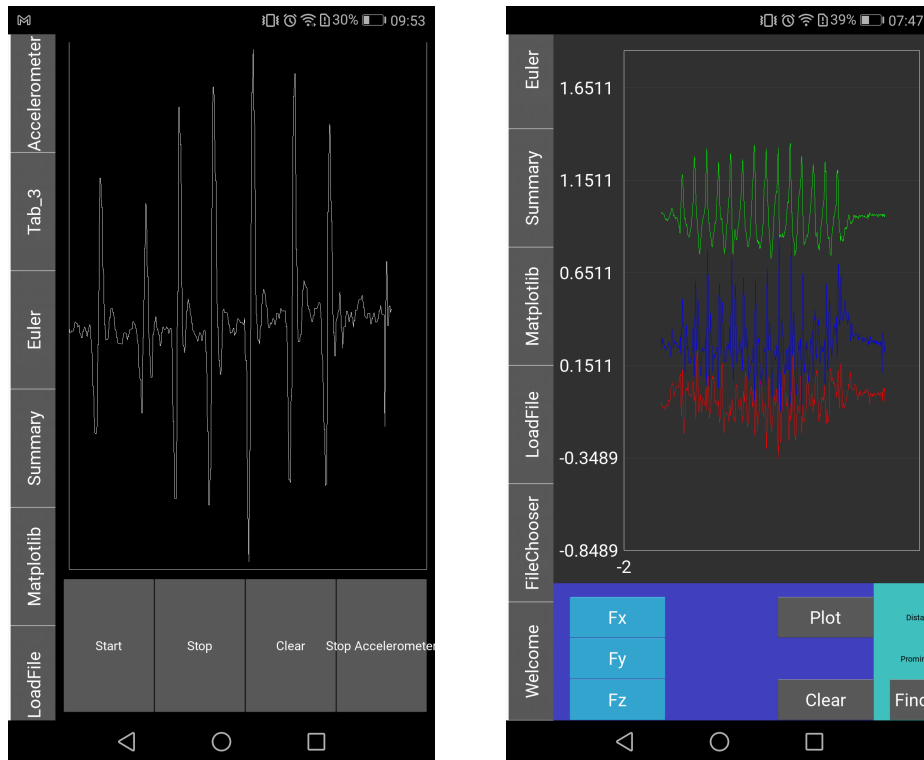


Figura 2: Captura e visualização de dados

Pelo menos dois passos do processamento de dados captados do acelerômetro devem ser ilustrados. Por exemplo a ilustração em forma de gráfico da detecção da quantidade de passos em um movimento uniforme, o registro do ciclo completo em um movimento circular, início e o fim de uma queda, etc.

### 3. Simulação do experimento:

Para cada experimento realizado, deverá ser implementado um modelo de simulação usando os parâmetros obtidos experimentalmente. A aplicação deverá aceitar novos inputs dos parâmetros e ser capaz de atualizar o modelo com a nova entrada dos parâmetros.

Embora tenha de seguir uma estrutura de classes pré-definidas para o seu funcionamento, a implementação do EP deverá ser feita Python dividido em dois arquivos: - um arquivo .py contendo a implementação em Python do EP - um arquivo .kv contendo o 'design' da aplicação

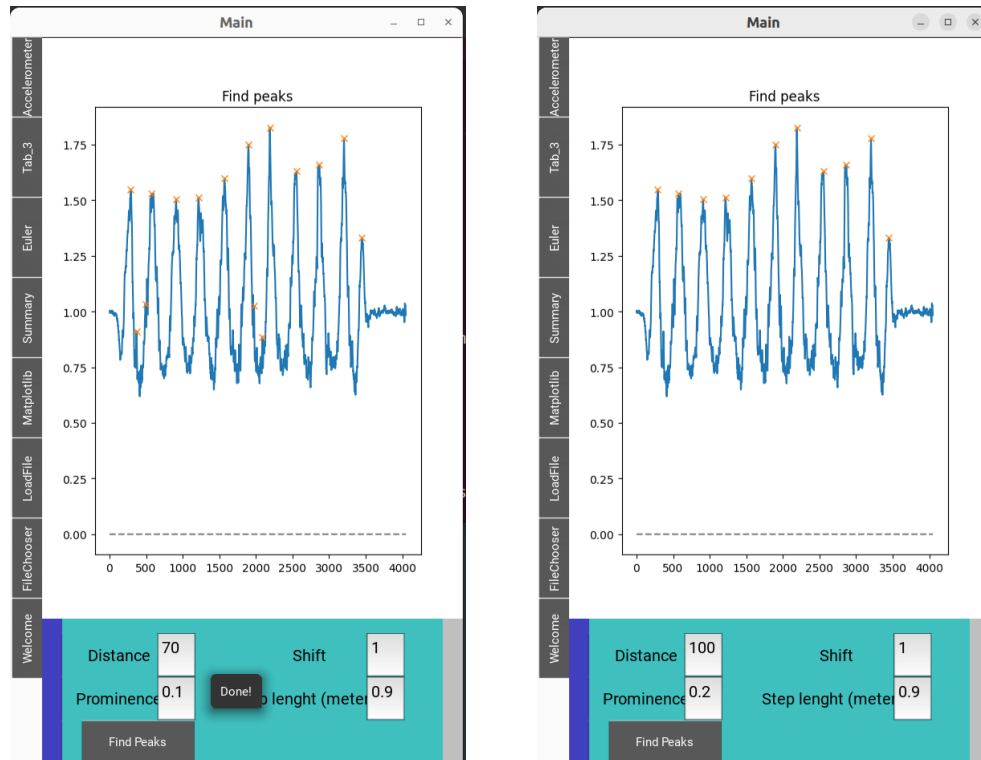


Figura 3: Captura de tela do processamento de dados. Observe que o aplicativo aceita parâmetros (Distance e Prominence) de entrada para modificar o processamento de dados

## 2.2 Buildozer

Implementada a aplicação em Python via Kivy, é necessário convertê-la para um apk. Isto será realizado através do buildozer, uma ferramenta de criação de pacotes para aplicativos. Uma vez instalado <sup>3</sup>, será necessário criar o arquivo *buildozer.spec* através do comando:

```
$ buildozer init
```

O arquivo *buildozer.spec* contém as especificações do aplicativo a ser compilado. Note que os pacotes utilizados para a implementação deverão ser incluídos neste arquivo (scipy, matplotlib, numpy, por exemplo). Atente-se também para a **arquitetura do celular** onde será executada a aplicação (armeabi-v7a, arm64-v8a, x86, x86\_64). Dependendo das funcionalidades utilizadas do celular, será necessário incluir permissões para o aplicativo.

Após ajustar as especificações, a compilação pode ser feita com o comando abaixo. Normalmente a primeira vez pode demorar alguns minutos (dependendo da máquina pode chegar a mais de 10 minutos). Se não ocorrer nenhum problema, o apk será gerado na pasta *bin*.

```
$ buildozer -v android debug
```

<sup>3</sup>Mais informações em: <https://buildozer.readthedocs.io/en/latest/installation.html>

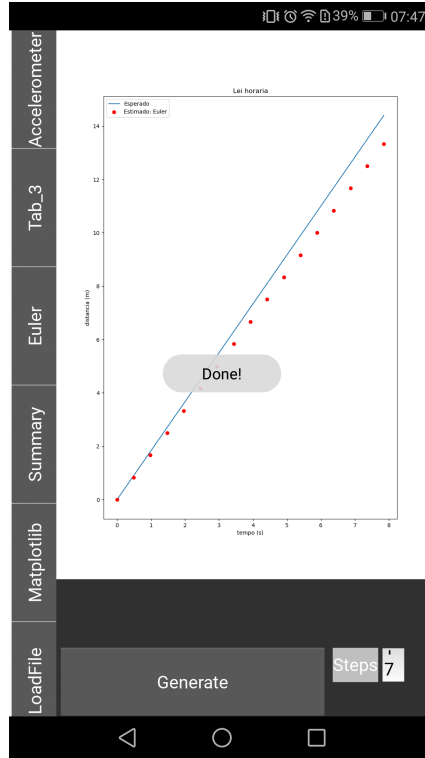


Figura 4: Exemplo da simulação do experimento visualizada no aplicativo.

## 2.3 Experimentos

Para cada um dos experimentos, você deve:

- Projetar o experimento.
- Realizar o experimento **pelo menos 3 vezes**. Mostre o gráfico dos dados para as 3 repetições, de maneira a ilustrar a variação dos dados.
- Obter os dados do acelerômetro com o celular. Mostrar os dados no celular.
- Simular o movimento usando os modelos do EP1 para obter os dados simulados  $x_s(t)$  ou  $\theta_s(t)$ , velocidade  $v_s(t)$  e aceleração  $a_s(t)$ . Mostre outros gráficos para ajudar na interpretação, como o gráfico das posições e dos dados do acelerômetro e de outros sensores que porventura tiver usado.
- Use a notação descrita neste enunciado no seu relatório (i.e.,  $x_e(t)$ ,  $\theta_e(t)$ ,  $v_e(t)$ ,  $a_e(t)$ ,  $x_s(t)$ ,  $\theta_s(t)$ ,  $v_s(t)$ ,  $a_s(t)$  etc).

## 3 Entrega

### 3.1 Relatório de entrega e vídeo

A entrega do EP consistirá no envio de um arquivo zip contendo:

- Arquivo .py e .kv da implementação e o arquivo de configuração buildozer.spec para gerar o aplicativo.
- PDF do relatório do EP usando o modelo disponível em <https://www.overleaf.com/read/xcztkhpdxpwc>. O arquivo deve ser enviado no link do EP no eDisciplinas da USP (Moodle da disciplina).
- O grupo deve criar um vídeo com uma demonstração do uso do aplicativo com, no máximo, 10 minutos. O vídeo deverá mostrar todas as funcionalidades implementadas como se fosse um manual do usuário, mostrando os dados do acelerômetro exibidos em tempo real, o input de outros parâmetros, cálculo do modelo com os novos dados, etc (por exemplo: "na aba X apertando o botão Y será inicializada a captura de dados do experimento Z pelo acelerômetro). No final deverá ser postado no Youtube e o link deverá ser incluído no início do relatório.
- Todos os membros do grupo devem entregar individualmente uma cópia de seu EP no Moodle.