UNIVERSIDADE NOVA DE LISBOA
FCT - Departamento de Engenharia Electrotécnica e de Computador
Secção de Robótica e Manufatura Integrada

Integração de Sistemas Ciber-Físicos (ISCF)

2022 / 2023

Trabalho Laboratorial 1

Desenvolvimento de serviços RESTful, Cloud e Aplicações Web

Duração: 4 aulas acompanhadas por docente

Entrega: 08 de Abril de 2023

Docente: Ricardo Peres <ra.peres@fct.unl.pt>

1. Apresentação do Problema

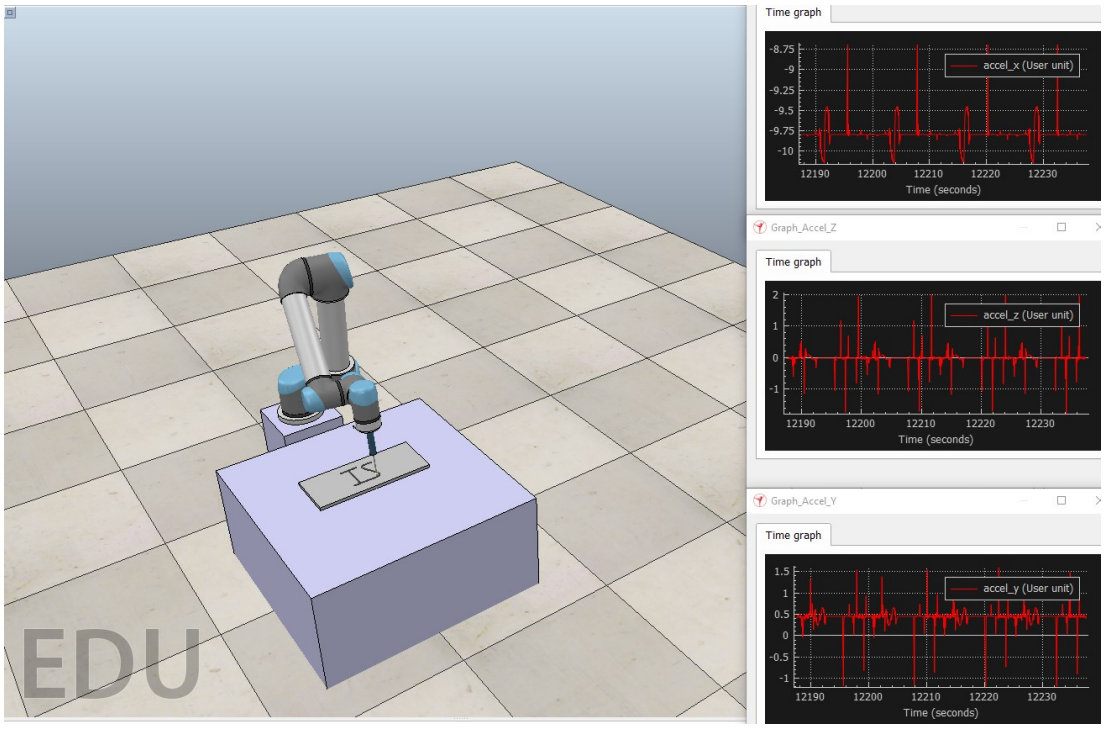

Figura 1 - Ambiente de simulação no CoppeliaSim

Pretende-se desenvolver uma solução capaz de monitorizar dados de um acelerómetro (em x, y e z) instalado num UR5 simulado no CoppeliaSim. Para este efeito, irá ser implementado um script em Python como **ferramenta de integração responsável por extrair os dados e guardá-los numa base de dados na cloud** (*Firebase*).

Posteriormente, deverá ser ainda desenvolvida uma ***web application*** com *Next.js* **que permita a interação com o utilizador** em duas vertentes. Em primeiro lugar, esta aplicação deverá possibilitar a visualização dos dados do acelerómetro guardados no *Firebase*. Adicionalmente, o utilzador deverá ter a possibilidade de customizar o intervalo a que os dados são actualizados.

Infraestrutura

O trabalho encontra-se dividido em duas partes. Durante a primeira aula, cada grupo deverá desenvolver uma solução local, focando-se apenas na ferramenta de integração (módulo *Data Collection* na Figura 2) para a aquisição dos dados, como representado na Figura 2.

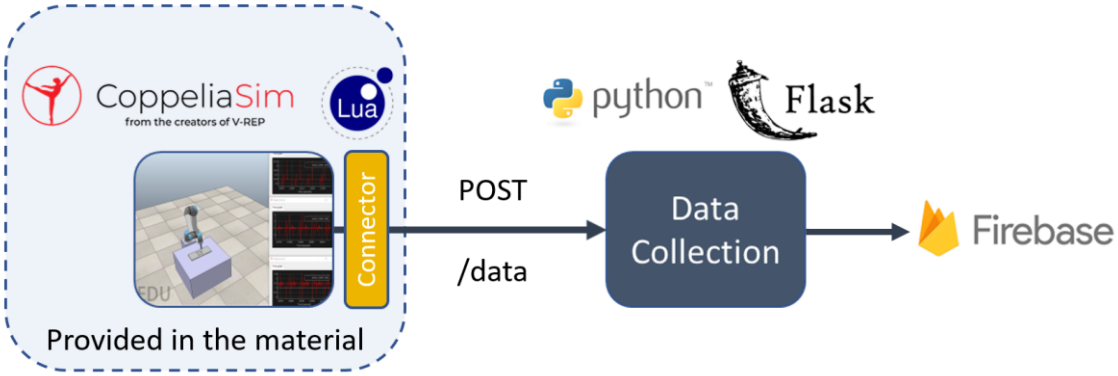

Figura 2 - Esquemático para a primeira fase do trabalho

Nesta fase, pretende-se apenas que a ferramenta de integração consiga extrair os dados da simulação através de uma API RESTful (implementada em Python com o package *flask-restful*). Seguidamente os dados deverão ser guardados na base de dados de tempo-real do Firebase.

Numa segunda fase, será desenvolvida uma aplicação web (*dashboard*) que permita ao utilizador não só visualizar os dados que vão sendo guardados no Firebase, como modificar o intervalo de tempo a que os mesmos são actualizados, como ilustrado na Figura 3.

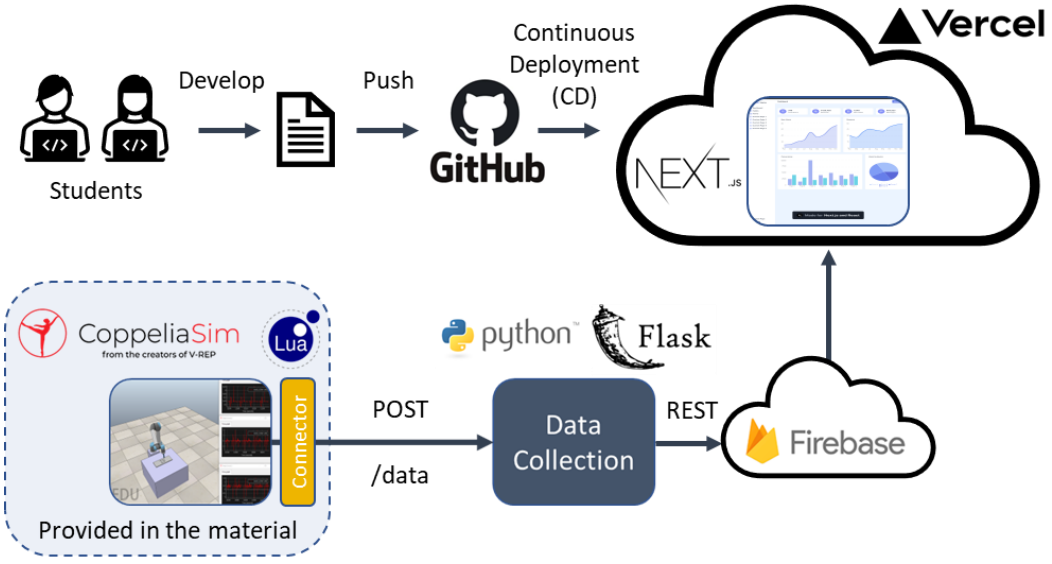

Figura 3 - Esquemático para a segunda fase do trabalho

A aplicação web, desenvolvida com Next.js, deverá ser alojada na plataforma *Vercel*. Desta forma, o código desenvolvido por cada grupo poderá ser *pushed* para um repositório *Github* (facilitando colaboração e controlo de versões), sendo que o *Vercel* possibilita o *deployment* automático da aplicação web ao detectar alterações no repositório do código.

Base de Dados (Realtime Database)

A base de dados (Figura 4), será implementada na plataforma Firebase, sendo esta uma base de dados NoSQL na *cloud* em que os dados são guardados como JSON e sincronizados em tempo real entre os clientes. A Figura 4 mostra uma possível estrutura para ser usada no presente trabalho.

Figura 4 - Estrutura sugerida para a base de dados de tempo real no Firebase

Para configurar o acesso à base de dados será necessário primeiro que tudo criar um novo projeto na plataforma do Firebase, fazendo o login por exemplo com o email da faculdade (`@campus.fct.unl.pt`).

A documentação para a criação da base de dados e interação (leitura e escrita) utilizando uma API REST pode ser encontrada no link abaixo:

https://firebase.google.com/docs/database/rest/start

Um exemplo de um POST em Python para adicionar um novo valor no node accel da DB no Firebase é apresentado de seguida:

```python
from requests import post
# Here the address could be a remote server, or in this case your Firebase real-time DB
post(address, json={"data": 123})
```

2. Implementação

Na implementação pedida será utilizado o seguinte material:

- Linguagem Python 3.7 (recomenda-se a instalação através do Anaconda); https://www.anaconda.com/products/individual
- Liguagem javascript;
- VSCode ou semelhante;
- CoppeliaSim Edu version 4.0.0; https://coppeliarobotics.com/files/CoppeliaSim_Edu_V4_0_0_Setup.exe
- Código fornecido pelo corpo docente;

Packages de Python necessários:

- Flask: https://anaconda.org/anaconda/flask
- Flask Restful: https://anaconda.org/conda-forge/flask-restful
- Requests: https://anaconda.org/anaconda/requests

Para instalar as dependências necessárias basta utilizar o ficheiro *requirements.txt* fornecido juntamento com o código base disponibilizado no CLIP, correndo o comando seguinte na directoria do projecto:

```
pip install -r requirements.txt
```

3. Planeamento das Aulas

Aula 1 – Configuração inicial e Desenvolvimento da API de Integração

1. Siga as instruções fornecidas pelos docentes e instale o Anaconda Environment com Python 3.7. Instale também o IDE VS Code ou equivalente.
2. Estude o código fornecido pelos docentes

   Leia atentamente todo o código fornecido, familiarizando-se com as funcionalidades implementadas e as sugestões que se encontram em comentário.

3. Configure a base de dados na plataforma Firebase. https://firebase.google.com/docs/database/rest/start

4. Implemente o código necessário em Python para permitir ao conector fornecido enviar os dados extraidos da simulação para a uma API alojada localmente, responsável por posteriormente inserir os mesmos na base de dados.

Como referência, pode utilizar a documentação Quickstart:
https://flask-restful.readthedocs.io/en/latest/quickstart.html
Um exemplo:
https://blog.miguelgrinberg.com/post/designing-a-restful-api-using-flask-restful

Aulas 2 e 3 – Desenvolvimento do Front-End para Visualização

1. Implemente o *Front-End* com o dashboard de monitorização. Comece pelo tutorial "Getting Started" disponibilizado na documentação do Next.js:

   https://nextjs.org/docs/getting-started

   O *front-end* deve possibilitar ao utilizador, **no mínimo**:
   - Monitorizar continuamente através de um gráfico os valores do acelerómetro extraídos do CoppeliaSim;
   - Actualizar o intervalo de actualização dos dados (por exemplo de 1 em 1 segundo para 2 em 2 segundos) através de *input* do utilizador;

Aulas 4 – Finalização do Trabalho e Funcionalidades Adicionais

1. Faça o deployment da aplicação web no Vercel

   ### Funcionalidades adicionais

   Algumas sugestões:

   - Implementar autenticação na aplicação web;
   - Estender a monitorização para gerar alarmes caso os dados ultrapassem determinado valor limite;
   - Implementar uma funcionalidade de previsão dos dados;
   - Etc.

4. Avaliação

A avaliação do trabalho tem a seguinte ponderação:

- Correta implementação e demonstração de funcionamento do trabalho previsto para a aula 1:
  - 8 valores
- Correta implementação e integração da aplicação web (dashboard):
  - 8 valores
- Deployment na plataforma Vercel:
  - 2 valores
- Correta implementação e demonstração de funcionalidades adicionais relevantes:
  - 2 valores.

Docentes:

Ricardo Peres ra.peres@fct.unl.pt

José Barata jab@fct.unl.pt