#### Instituição de Ensino FEI

# Tópicos Avançados de Redes de Comput.

## CE5320

### Projeto de HTTP

#### Integrantes:

• Guilherme Gomes Chagas - 22.119.013-5

• Luca Milla - 22.119.002-8

• Matheus Luiz Gonçalves Pelicer - 22.119.024-2

• Thiago Soares Cardoso da Silva - 22.119.044-0

Ciência da Computação FEI

5º Ciclo - 2021

Período Noturno

### Arquivos

O servidor de HTTP é composto por 1 arquivo ***.py*** e há 2 arquivos ***html*** e 1 arquivo ***.py*** de cliente disponíveis para testar as funcionalidades do servidor.

O arquivo do servidor utiliza as bibliotecas de **Socket**, **OS**, **SYS**, **CODECS** e **Glob**. O arquivo de cliente utiliza apenas a biblioteca de **Socket**. Todas estas bibliotecas estão nativamente disponíveis caso o **Anaconda** (Jupyter Notebook) esteja instalado.

O servidor é capaz de receber mais de uma conexão por graças ao método ***.fork()***.

O arquivo *post\_form.html* possui dois campos de **input** e um botão para para encaminhar os dados para outra página html.

### Métodos HTTP

O servidor tem suporte para os métodos **GET**, **POST**, **PUT** e **DELETE**:

#### GET

Após inicializar o servidor basta conectar-se há um dos arquivos disponíveis de teste dentro da pasta***/server\_files*** através do navegador. Para fazer isso basta adicionar “localhost:9000” e o *path* do arquivo que deseja abrir.

As mensagens de respostas HTTP que podem surgir deste método são: **200**, **301**, **400** e **404**, que são respectivamente, Ok, File Moved, Bad Request e Not Found.

#### POST

Para testar o funcionamento do método **POST**, será necessário inicializar o arquivo ***/post\_form.html*** e preencher o formulário presente dentro da página html.

As mensagens de respostas HTTP que podem surgir deste método são: **200**, **201** e **400**, que são respectivamente, Ok, Created e Bad Request.

#### PUT

Para realizar o método **PUT** é necessário inicializar o cliente e mandar o comando *“PUT path\_do\_arquivo”*.

As mensagens de respostas HTTP que podem surgir deste método são: **200**, **201** e **400**, que são respectivamente, Ok, Created e Bad Request.

#### DELETE

Para realizar o método **DELETE** é necessário inicializar o cliente e mandar o comando *“DELETE path\_do\_arquivo”*.

As mensagens de respostas HTTP que podem surgir deste método são: **200** e **400**, que são respectivamente, Ok e Bad Request.