

# Aplicação para Visualização de Ações

Paulo Júnior D. S. Ramiro<sup>1</sup>, Renan P. Barreto<sup>1</sup>

<sup>1</sup>Centro de Ciências Computacionais - Universidade Federal do Rio Grande (FURG)  
Caixa Postal 474 – 96.201-900 – Rio Grande – RS – Brasil

paulojr2016canaa@outlook.com, renan.barreto@furg.br

## 1. Introdução

Com o intuito de exercitar os conhecimentos adquiridos na disciplina de Sistemas Distribuídos e para a avaliação prática do segundo semestre foi criada uma página web para múltiplos usuários utilizando Python 3 [Python Software Foundation 2022] para visualização de informações de ações. Hospedada utilizando a ferramenta *Streamlit* [Snowflake Inc 2022] e com a capacidade de ser acessada por *browsers*, a aplicação foi desenvolvido para a disciplina, tendo como requerimentos:

- Suporte a conexões simultâneas ;
- Uso de dados adquiridos através de uma *API*;
- Multi-usuários;
- Devidamente comentado e documentado;
- Aplicação/Página online via *browsers*;

A linguagem utilizada foi Python [Python Software Foundation 2022], escolhida pela sua simplicidade, disponibilidade de bibliotecas e documentações públicas. Utilizamos Yahoo Finance como a *API* de origem dos dados de ações, a qual há uma biblioteca na linguagem escolhida para seu uso.

Yahoo Finance possui dados históricos de múltiplas ações e desde 2017 também possui dados sobre criptomoedas. Logo com as ferramentas escolhidas é possível trabalhar com diferentes dados e construir a visualização destes para os ativos escolhidos.

## 2. Arquitetura

O trabalho apresentado possui a arquitetura Cliente-Servidor, a qual um cliente envia requisições a um servidor e recebe respostas do mesmo. A aplicação descrita neste documento representa o cliente do sistema, a qual recebe as informações da *API* do Yahoo Finance.

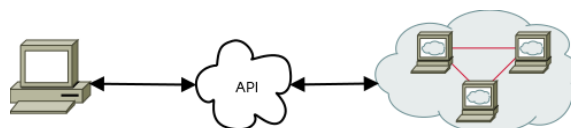


Figura 1. Ilustração de um Cliente-Servidor através de uma API.

*API*, que significa *Application Programming Interface*, é uma interface que possui protocolos e definições para acessos a serviços e dados. Este tipo de implementação permite que aplicações se comuniquem com outras, mesmo que possuam implementações diferentes.

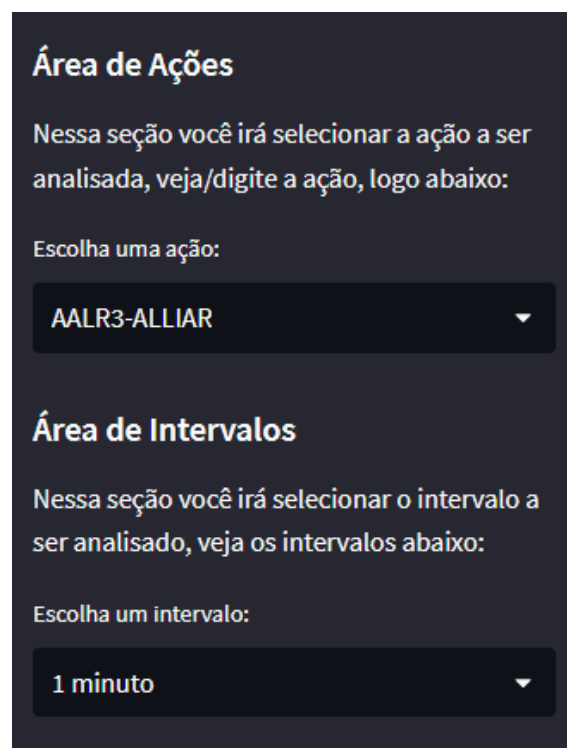
Com isto é possível a criação de aplicações as quais utilizam serviços, ou dados, de outras aplicações ou servidores. Além de permitir a comunicação, o uso de *APIs* pode gerar mais segurança para as aplicações provedoras de serviços, já que a mesma possui uma interface com regras e padrões ao invés de permitir acesso direto a seus dados e código.

### 3. Aplicação

O website possui a funcionalidade de cliente, fazendo requisições a *API* do Yahoo Finance, e utilizando a biblioteca YFinance[Aroussi 2020] para acessar os dados. Os dados podem ser acessados de acordo com o período e ativo selecionado.

#### 3.1. Requisições e Dados

As requisições ao servidor são feitas utilizando como parâmetro um intervalo e o nome do ativo, com o período máximo sendo definido de acordo com o intervalo escolhido, como mostra a figura 2



The image shows a dark-themed web interface with two main sections. The top section is titled 'Área de Ações' in a light blue font. Below the title, there is a light blue instruction: 'Nessa seção você irá selecionar a ação a ser analisada, veja/digite a ação, logo abaixo:'. Underneath this, the text 'Escolha uma ação:' is followed by a dark blue dropdown menu that currently displays 'AALR3-ALLIAR'. The bottom section is titled 'Área de Intervalos' in the same light blue font. It has a light blue instruction: 'Nessa seção você irá selecionar o intervalo a ser analisado, veja os intervalos abaixo:'. Below this, the text 'Escolha um intervalo:' is followed by a dark blue dropdown menu that currently displays '1 minuto'.

**Figura 2. Escolha de parâmetros para requisição.**

A biblioteca Pandas [NumFOCUS 2022] permite a formatação dos dados para análise e manipulação. Os quais são transformados em um objeto de lista em Python. Os dados então podem ser utilizados para criação de gráficos e tabelas para a exibição destes de forma simplificada, ordenada e de fácil entendimento.

Entre os dados estão o *timestamp*, valores de alta e baixa, volume e valores de abertura e fechamento. Os dados são demonstrados para o usuário da sessão em formato de tabela, gráfico *candlestick* e gráfico de linha. Também é possível o download da tabela de dados em formato CSV.

### 3.2. Frontend

A aplicação a qual o usuário interage possui sua interface, ou *frontend*, criado com a utilização da ferramenta *Streamlit* [Snowflake Inc 2022]. Esta ferramenta permite a criação de aplicativos Python com interface amigável com esforço reduzido.

O ambiente permite a fácil prototipação de *WebApps* com capacidade para multi-usuários, permitindo a hospedagem e compartilhamento destes aplicativos. A junção do *Streamlit* com o *Pandas* permite a implementação de sistemas simples mas capazes de manipular dados e criar visualizações para os mesmos como mostra a figura 3.



Figura 3. Uso do *Pandas*, *Plotly* e *Streamlit* para geração de gráficos interativos.

A interface permite a escolha dos parâmetros para as requisições, assim como a visualização do resultado em uma tabela. Os dados podem ser ordenados na tabela de acordo com a vontade do usuário, clicando sobre os títulos das colunas. Os gráficos gerados com *Plotly* [Plotly 2022], *candlesticks* e de linha, podem ser ajustados com ampliação de um ponto ao selecionar os limites, movidos em seu eixo horizontal e baixados como imagem e visualizados em tela cheia.

### 4. Dificuldades Encontradas

O uso de *APIs* permite a comunicação entre diferentes aplicativos e pode gerar mais segurança, entretanto as limitações impostas pelas mesmas, assim como a falta de controle sobre o formato dos dados, podem criar problemas para manipulação dos dados ou a utilidade dos mesmos. O segundo ponto descrito pode ser enfrentado utilizando a documentação das ferramentas utilizadas. Entretanto o Yahoo Finance não apresenta, de forma fácil de ser encontrada dada sua descontinuação, uma documentação para os seus dados, sendo necessário utilizar bibliotecas e documentações de terceiros para o uso dos mesmos.

### 5. Conclusão

Na nossa visão, a aplicação desenvolvida para a avaliação da disciplina atingiu os objetivos fixados. Entre os objetivos atingidos estão a capacidade de suportar múltiplos usuários e servir a estes usuários dados de uma *API* de terceiros de forma online, sem a necessidade de instalação. A aplicação também manipulou os dados e os exibiu de forma agradável e interativa, permitindo ao usuário o download destes dados em forma de tabela ou gráficos.

O trabalho também demonstrou a capacidade de criação de aplicações distribuídas sem o desenvolvimento de um *backend* e servidores ao utilizar as informações de *APIs*

assim como a possibilidade desenvolvimento destas interfaces para oferta como serviço para aplicações que necessitam destes serviços ou dados.

A versão online da aplicação pode ser encontrada em: <http://abre.ai/prsd2>

Código fonte e instruções para instalação e execução local podem ser encontradas em: [https://github.com/PauloRamiro/Sistemas\\_Distribuidos\\_Trabalho\\_2](https://github.com/PauloRamiro/Sistemas_Distribuidos_Trabalho_2)

## **Referências**

Aroussi, R. (2020). yfinance. [Online; accessed 14-December-2022].

NumFOCUS (2022). pandas. [Online; accessed 20-December-2022].

Plotly (2022). Plotly open source graphing library for python. [Online; accessed 20-December-2022].

Python Software Foundation (2001-2022). Python 3.11.1 documentation. [Online; accessed 17-December-2022].

Snowflake Inc (2022). Streamlit. [Online; accessed 20-December-2022].