

# **Aplicação mobile para previsão de resultados no CS:GO**

**Gabriel Henrique Alberto de Jesus<sup>1</sup>, Carlos Gusmão Teixeira<sup>2</sup>, Jean Carlos Gomes Aguiar<sup>3</sup>, Joao V**

<sup>1</sup>Departamento de Ciência da Computação - Instituto de Ensino Superior de Brasília  
Brasília – DF – Brazil

`gabriel.henrique@iesb.edu.br, carlos.teixeira@iesb.edu.br, jean.aguiar@iesb.edu.br, joao.v`

## ***Resumo.***

### **1. Introdução**

Como o grande crescimento das competições de E-Sports nos últimos anos, muitos espectadores ficam sem as informações mais recentes e corretas. Observamos também que mesmo que alguns espectadores consigam essas informações, essas informações só se encontram em websites e não para aplicativos mobile.

### **2. Objetivo Geral**

### **3. Referencial Teórico**

#### **3.1. Linguagem Python**

A linguagem de programação Python foi criada no início dos anos 90 pelo matemático holandês Guido van Rossum para ser a sucessora da linguagem ABC. Python é uma linguagem de programação de alto nível que pode ser usada em diferentes tipos de aplicações, indo de programas mais básicos até programas mais sofisticados como Data Science, com uma comunidade de desenvolvedores bastante ativa e disponibilidade de várias bibliotecas poderosas.

#### **3.2. Selenium**

É uma biblioteca do python que foi desenvolvida para automatizar os navegadores, foi desenvolvido principalmente para testar aplicações web pelo browser de forma automatizada. Neste trabalho usaremos o Selenium para simular um navegador web para conseguir pegar os dados HTML de sites dinâmicos. O Selenium faz a função de um emulador de navegadores(Google Chrome, Mozilla Firefox), simulando um usuário.

#### **3.3. Implementação do Web Scraping**

Web Scraping são algoritmos autômatos para extração de informações de sites web, o utilizamos neste trabalho com a biblioteca BeautifulSoup 4, para extrair as informações do HTML.

##### **3.3.1. BeautifulSoup 4**

É uma biblioteca do python que foi desenvolvida para extração de dados de arquivos HTML, sendo o seu principal uso para web scraping, para receber arquivos HTML e os transformar em objeto BeautifulSoup, que organiza o HTML em uma árvore sintática, a partir desse objeto podemos extrair as informações contidas no arquivo HTML.

### 3.4. Scikit Learn

Scikit-learn é uma biblioteca gratuita, de código aberto, para Machine Learning (Aprendizado de Máquina) em Python. Ela também fornece uma seleção de recursos eficientes para modelagem estatística, análise e mineração de dados, além de suporte ao aprendizado supervisionado e não supervisionado.

### 3.5. MongoDB

MongoDB é um banco de dados NoSQL, open source, de alta performance e flexível. O MongoDB é orientado a documentos, os dados são armazenados como documentos utilizando do formato JSON e cada documento pode conter dados com diferentes campos. O MongoDB foi criado pela empresa DoubleClick em 2009 e foi implementado na linguagem C++.

### 3.6. Flutter

Flutter é um framework opensource criado pelo Google e tem foco em desenvolvimento para dispositivos móveis(Android ou iOS), mas também começando a ser usada para desktop ou web, e trabalhando em conjunto com a linguagem de programação Dart.

## 4. Desenvolvimento

Após escolhermos o tema do nosso projeto, Web Scrapping, começamos a pesquisar as melhores ferramentas para desenvolvermos nossa aplicação, a linguagem Python se mostrou ser a melhor opção pois tem uma grande variedade de bibliotecas e uma dessas bibliotecas foi a BeautifulSoup, com essa biblioteca conseguimos pegar toda a informação HTML dos sites, a utilizamos como base para rodar nossa aplicação, porém quando começamos a implementar a biblioteca na base de sites, o BeautifulSoup mostrou ter uma limitação com sites dinâmicos, ou seja, consegue fazer mudanças na página web sem fazer qualquer alteração o código fonte, e o BeautifulSoup não consegue pegar essas partes dinâmicas dos sites e para solucionar esse problema achamos outra biblioteca python chamada Selenium que emula um navegador web, e com isso conseguimos simular um usuário assim conseguindo extrair o HTML completo do site, mas como o Selenium é uma biblioteca em que pode ser feita o scrapper mas de uma maneira mais limitada e o foco dela é em emular um usuário no navegador, logo escolhemos utilizar as duas bibliotecas, o Selenium para emular um usuário no navegador e assim obtendo o HTML final e BeautifulSoup extrair as informações do HTML.

Após ter acesso a todas a informações HTML necessárias, era preciso um banco de dados para armazenar as informações dos produtos obtidos, escolhemos o MongoDB para usar nesse projeto já que é um banco NoSQL, pois os produtos não tinham sempre as mesmas informações ligadas a eles, e o MongoDB é mais flexível nessa parte, porque ele armazena as informações em objetos JSON(JavaScript Object Notation) que são guardados em coleções.

## 5. Resultado parciais

..

## 6. Conclusão

O objetivo deste trabalho foi a proposição que seja capaz de realizar previsões de resultados de partidas de CS:GO baseando-se com o índice de acertos dos jogadores oferecendo um índice maior do que dos mercados de apostas, feito a partir de análise de resultados reais com as previsões dadas.

## 7. Pontos fortes e fracos dos artigos

Artigo 1: Predicting Winning Team and Probabilistic Ratings in “Dota 2” and “Counter-Strike: Global Offensive” Video Games [https://link.springer.com/chapter/10.1007/978-3-319-73013-4\\_17](https://link.springer.com/chapter/10.1007/978-3-319-73013-4_17) Ponto forte : *Utiliza arquivos brutos dos jogos para fazer uma análise mais profunda de cada partida* Ponto fraco : *Usa sistemas de medição de habilidade não focados na ocs : go*

Artigo 2: Summarizing E-sports matches and tournaments: the example of counter-strike: global offensive - <https://dl.acm.org/doi/10.1145/3304113.3326116> Ponto forte: classifica e busca rankings de jogadores para fazer uma predição mais precisa Ponto fraco: A classificação é baseada em opinião sobre os jogadores colhida em fóruns

Artigo 3: CScore – A Real Time Application for E-sports - <https://www.diva-portal.org/smash/get/diva2:938821/FULLTEXT01.pdf> Ponto forte: Aplicação em tempo real. Ponto fraco: Apesar de um artigo muito grande e relacionado ao nosso tema, eles foram prolixos demais e deixaram muito dos objetivos do artigo para trabalhos futuros

Artigo 4: Interpretable Real-Time Win Prediction for Honor of Kings – a Popular Mobile MOBA Esport - <https://arxiv.org/pdf/2008.06313.pdf> Ponto forte: Desenvolvimento mobile para um aplicativo em tempo real Ponto fraco: Ponto focal de dados não relacionado ao nosso projeto

Artigo 5: Dataset csgo: <https://sol.sbc.org.br/index.php/dsw/article/download/17412/17248/> Ponto forte: Outra maneira de se conseguir dados das partidas, no caso deles fazendo análise de “demos” e pegando cada log de ação feito na partida. Ponto fraco: Apesar de terem feito um dataset robusto, tem muitas informações que não fazem muita diferença.

Artigo 6: Exploiting Filtering approach with Web Scrapping for Smart Online Shopping Ponto forte: Scraper usando as mesmas bibliotecas do Python. Ponto fraco: Tema não muito relacionado com o nosso.

[?], [?], [?], [?], [?], [?], [?] e [?], [?], [?], [?], [?]

## References