**INSTITUIÇÃO ANHEMBI MORUMBI DE ENSINO SUPERIOR DO GRUPO ÂNIMA EDUCAÇÃO**

**LUCAS SIMÕES CARVALHO DA SILVA, GIOVANNI LOPES CAMPOS, MARCELO HENRIQUE DA SILVA VENTURA, ANDY HYONG TAE CHOI YOUN, FELIPE DAURA LANZONE FERREIRA**

**TRABALHO DE USABILIDADE, DESENVOLVIMENTO WEB, MOBILE E JOGOS: SITE CSGOBRASIL**

SÃO PAULO

2023

|  |  |  |  |
| --- | --- | --- | --- |
| CURSO(S): | ADS, Ciência da Computação | TURMA(S): |  |
| CAMPUS: | Mooca | SEMESTRE(S): | 3º |

|  |  |  |  |
| --- | --- | --- | --- |
| NOME: | Marcelo Henrique da Silva Ventura | RA: | 12522128126 |
| NOME: | Felipe Daura Lanzone Ferreira | RA: | 1252214120 |
| NOME: | Giovanni Lopes Campos | RA: | 12522149685 |
| NOME: | Andy Hyong Tae Choi Youn | RA: | 12522142446 |
| NOME: | Lucas Simões Carvalho da Silva | RA: | 12522161179 |
| NOME: |  | RA: |  |
| NOME: |  | RA: |  |

**RESUMO**

Este trabalho tem como objetivo apresentar um sistema web criado com a linguagem Javascript e com auxílio da linguagem de estilo CSS e a linguagem de marcação HTML. Desta maneira, apresentando suas características e tecnologias relacionadas. Além de apresentar a motivação para o tema do sistema. Este projeto utilizará a linguagem Java para o back-end, MySql para banco de dados e CSS, HTML e JavaScript para o front-end.

Palavras-chave: Usabilidade. Java. JavaScript.

Sumário

[1. INTRODUÇÃO 6](#_Toc137310925)

[2. OBJETIVOS 7](#_Toc137310926)

[2.1 OBJETIVOS GERAIS 7](#_Toc137310927)

[2.2 OBJETIVOS ESPECÍFICOS 7](#_Toc137310928)

[3. JUSTIFICATIVA 7](#_Toc137310929)

[4. IMPLEMENTAÇÃO DO CÓDIGO 8](#_Toc137310930)

[4.1 ESTILO DA PÁGINA COM CSS 8](#_Toc137310931)

[4.1.1 Carrossel da página principal 8](#_Toc137310932)

[4.1.2 Footer e Header estáticos 9](#_Toc137310933)

[4.1.3 Página de cadastro 9](#_Toc137310934)

[4.2 JAVASCRIPT 11](#_Toc137310935)

[5. USABILIDADE 12](#_Toc137310936)

[6. BACK-END DA APLICAÇÃO 13](#_Toc137310937)

[7.CONCLUSÃO 14](#_Toc137310938)

[8. REFERÊNCIAS 15](#_Toc137310939)

# 1. INTRODUÇÃO

A usabilidade não é exatamente a experiência do usuário sobre um produto, mas está fortemente relacionada. O fato de algo ter usabilidade o torna, na visão do usuário, algo fácil e agradável de usar, assim possibilitando a realização de atividades sem grande esforço. No entanto, um artefato que proporciona uma boa experiência ao usuário oferece muitos outros bons aspectos além da usabilidade em si e torna tal artefato prazeroso na realização das tarefas a qual ele foi direcionado, fazendo com que os usuários sintam vontade de usar a interface novamente.

Este trabalho tem como ideal apresentar um *site* com as funcionalidades de compra e venda de *skins* (diferentes tipos de padrões de cores para um item em um jogo) para o jogo “Counter-Strike: Global Offensive”. O *site* em questão foi criado utilizando a linguagem Javascript e com as demais linguagens de marcação e estilo HTML (HyperText Markup Language, ou Linguagem de Marcação de Hipertexto) e CSS (*Cascading Style Sheet*, ou Folha de Estilo em Cascatas), respectivamente. O *back-end* deste projeto utiliza da linguagem Java e a tecnologia de *Web service REST*, a qual será explicada no decorrer deste documento, para a comunicação do sistema distribuído.

O documento em questão irá abordar o tema do sistema e sua importância no atual contexto do mundo e como a tecnologia escolhida para a comunicação entre o *front-end* e *back-end* da aplicação é a mais adequada para a arquitetura do sistema apresentado.

Ademais, serão apresentados a estrutura do código do sistema e as motivações que levaram a escolha do tema do sistema e as tecnologias utilizadas no mesmo. Também será apresentado as principais funcionalidades do sistema e a integração do código com o *back-end* que foi criado como tema principal para a UC (unidade curricular) de Sistemas Distribuídos e Mobile.

# 2. OBJETIVOS

## 2.1 OBJETIVOS GERAIS

Este projeto tem como objetivo geral demonstrar de maneira sistemática o funcionamento de um *site* em Javascript com um *back-end* em Java. Desta maneira, serão mostradas imagens, justificativas, explicações e códigos para o sistema desenvolvido durante o andamento da UC de Usabilidade, desenvolvimento web, mobile e jogos.

## 2.2 OBJETIVOS ESPECÍFICOS

Dentre os objetivos específicos do trabalho estão:

* Elencar as funcionalidades do sistema;
* Demonstrar a importância do sistema na contemporaneidade;
* Apresentar as tecnologias utilizadas para o desenvolvimento do *software*;
* Explicar a estrutura em código que sustenta a aplicação;
* Ilustrar brevemente o *back-end* do projeto.

# 3. JUSTIFICATIVA

O jogo para computador “Counter-Strike: Global Offensive” é um sucesso global que apresentou este ano um recorde histórico tendo no dia 25 de março um total de 1.507.212 jogadores simultâneos nos seus servidores, segundo a base de dados da empresa Steam. Além disso, o jogo também traz grande rendimento para a empresa desenvolvedora, Valve, que lucra em base 650 milhões de dólares por ano com as caixas que a empresa vende para o público dentro do jogo, segundo a própria empresa.

Com base em tais dados, foi decidido dentro do grupo a criação de um sistema de compra e venda de *skins* do jogo, que movimenta milhões de dólares por ano, segundo a empresa desenvolvedora do jogo.

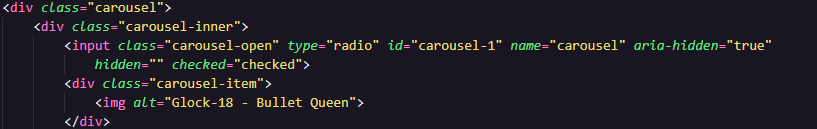
Dentre a tecnologia utilizada, tem-se a integração REST com *web servisse*, pois este é o método mais simples e prático de realizar a comunicação entre um *front-end* e um *back-end*. Através de requisições utilizando URLs (Uniform Resource Locator) se torna extremamente simples e rápido de se enviar dados entre as duas partes da aplicação.

# 4. IMPLEMENTAÇÃO DO CÓDIGO

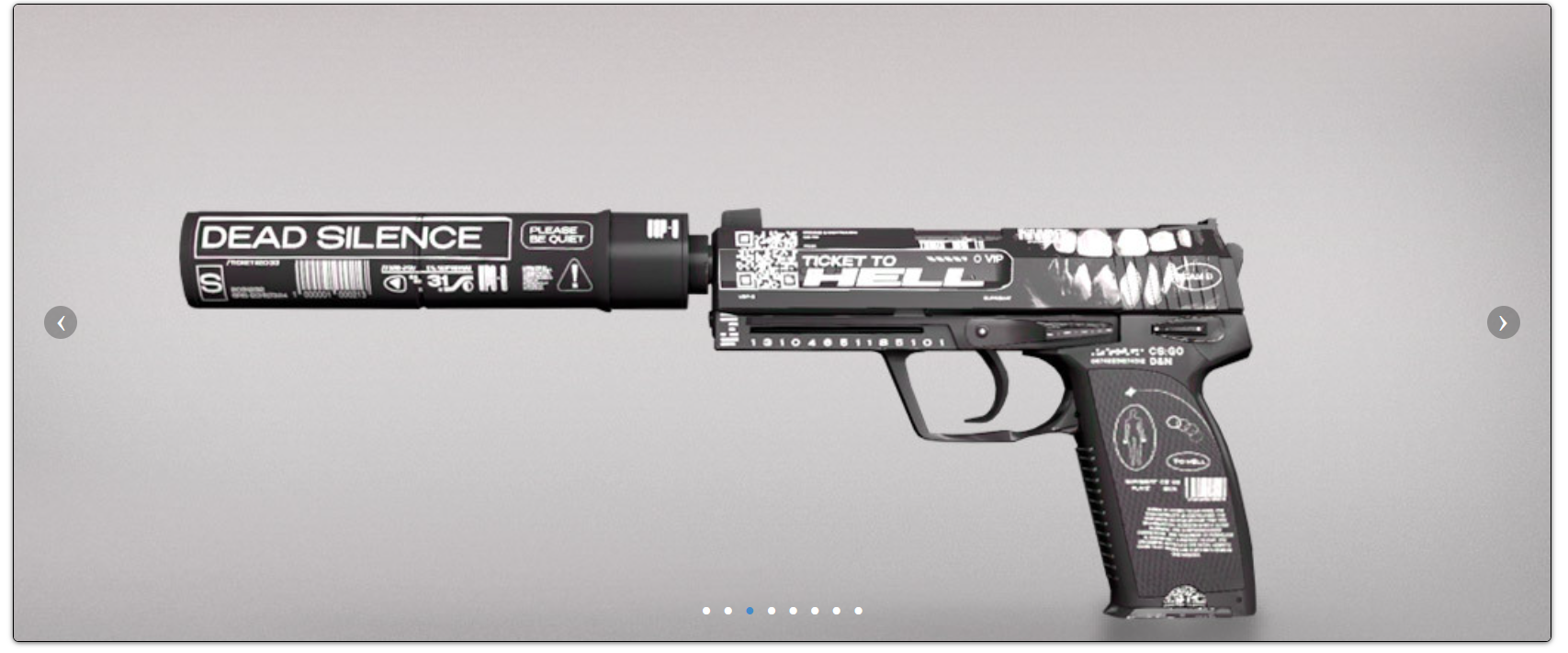
## 4.1 ESTILO DA PÁGINA COM CSS

### 4.1.1 Carrossel da página principal

A aplicação possui diversas páginas em HTML com seus respectivos estilos. No entanto, há algumas páginas com um desenvolvimento maior de seu CSS, e um destes exemplos é na *home page* do *site*.



Através das classes na imagem anterior, foi criado um carrossel de imagens, que podem ser alternadas através do clique em uma das setas das laterais da imagem. Este carrossel foi feito inteiramente com HTML e CSS, utilizando-se dos *inputs* do tipo *radio* para controlar qual imagem está atualmente sendo mostrada. Além disso, com Javascript, o carrossel foi incrementado mudando de imagem de tempos em tempos, a seguir será mostrado uma imagem do carrossel.



### 4.1.2 Footer e Header estáticos

Ademais, a utilização de um *header* e *footer* estáticos facilitaram a criação das páginas. Como, em todas as páginas do *site* seriam necessários a presença de um cabeçalho e rodapé idênticos para toda sessão, então através da *tag* HTML “iframe” foi colocado em todas as páginas uma referência para uma outra página HTML que contém apenas o rodapé ou o cabeçario.

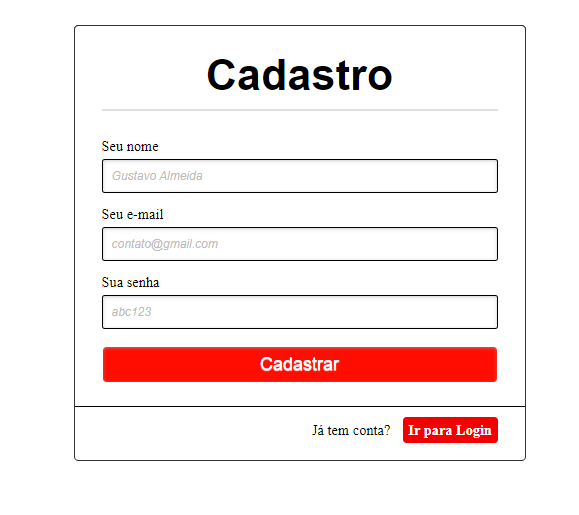


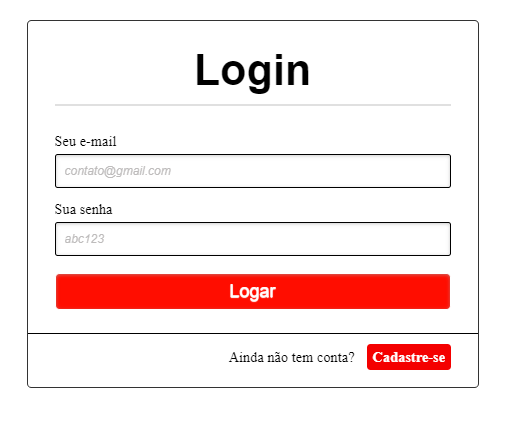
Além de tornar o código mais limpo e fácil de se analisar, esta técnica permite com que qualquer código em Javascript só precise ser aplicado uma única vez para um destes elementos.

### 4.1.3 Página de cadastro

A página de cadastro do *site* também possui uma construção de CSS diferente das demais, já que esta se utiliza de um modelo padrão da *tag* “forms” para a criação um formulário de cadastro e login na mesma página.

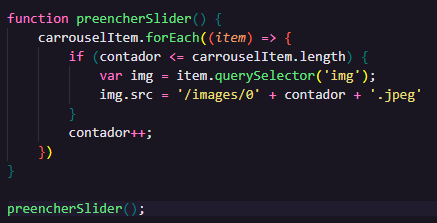
Através de uma animação feita em CSS, o formulário de cadastro se transforma no formulário de login, tornando fácil para o usuário acessar qualquer um dos formulários sem a necessidade de trocar de página.



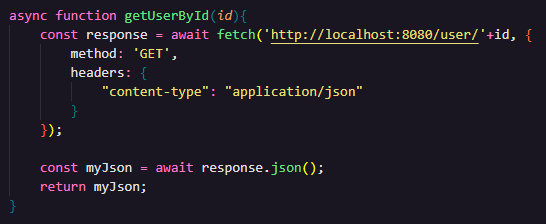


## 4.2 JAVASCRIPT

Todas as páginas HTML deste projeto são interativas e por tal motivo, todas utilizam-se de Javascript para sua interação. As principais ações que o Javascript executa são controle dos elementos da página, alteração do estilo dos elementos caso necessário e para realização de requisições HTTP.



Esta primeira imagem mostra um exemplo do controle dos elementos da página através de Javascript. Neste caso, ao invés de colocar todas as imagens, uma por uma no corpo do arquivo HTML, foi utilizado um *for* em todas as *tags* que terão uma imagem e através do próprio Javascript o arquivo correto foi colocado.



A imagem anterior retrata uma função que foi muito utilizada neste projeto, a qual seria o uso da função *fetch*. Esta função é derivada da API FETCH, a qual foi integrada a biblioteca do node a partir da versão 18.

Esta API foi amplamente usada para pegar, deletar, atualizar e adicionar dados ao banco de dados da aplicação. Através de requisições HTTP, o *front-end* envia e/ou solicita informações para o *back-end­* (feito em Java) e após uma resposta executa suas funcionalidades congruentes ao código Javascript do *front-end*.



Outra funcionalidade importante e útil para o projeto foi o “*local storage*”, que é uma propriedade do DOM dos navegadores, que permite que certos dados sejam salvos como uma variável dentro do armazenamento do navegador. Ele se assemelha ao *cache* ou *cookies* comuns na *web*. O que torna sua implementação mais simples do que os demais é que com apenas uma linha em Javascript é possível salvar qualquer dado no armazenamento local do navegador.

Nesta aplicação, o *local storage* foi usado exclusivamente para o armazenamento do id do usuário que está naquele navegador, e pode mudar dependendo do usuário que se cadastrar ou fazer *login* no *site*.

# 5. USABILIDADE

O *site* desenvolvido neste projeto adota uma abordagem cuidadosa em relação às normas de usabilidade e implementa práticas táticas sugeridas por Nielsen para melhorar a experiência do usuário. Essas estratégias foram aplicadas para garantir que o *site* seja intuitivo, acessível e eficiente para os visitantes.

O *layout* do *site* é organizado de forma clara e coerente, os elementos de navegação, como menus e links, são colocados em locais estratégicos e de fácil acesso, permitindo que os usuários encontrem facilmente o que estão procurando.

Além disso, o *site* adota práticas de design responsivo, garantindo que ele seja adaptável a diferentes tamanhos de tela.

As diretrizes de Nielsen também são aplicadas na forma como as informações são apresentadas. O *site* evita a sobrecarga de informações e presa pelo mínimo de informação necessária, assim dividindo o conteúdo em seções e utilizando um cabeçario que está sempre visível para que o usuário saiba todas as opções disponíveis.

Para melhorar ainda mais a usabilidade do *site*, uma abordagem de teste com usuários foi implementada. Diversas pessoas foram convidadas a interagir com a interface do *site* e a testar suas funcionalidades por conta própria. Essas pessoas tiveram a oportunidade de fornecer *feedback* sobre a experiência de uso, identificando possíveis problemas e sugerindo melhorias.

Em suma, o *site* em questão atende às normas de usabilidade e emprega táticas sugeridas por Nielsen para oferecer uma experiência de usuário otimizada. Através de um design intuitivo, layout organizado, design responsivo e testes com usuários, o *site* busca constantemente melhorias para garantir que os visitantes possam acessar as informações desejadas de forma fácil, rápida e eficiente.

# 6. BACK-END DA APLICAÇÃO

Utilizando a arquitetura REST, foi possível projetar uma API REST para o *site* de *skins* do jogo “Counter-Strike: Global Offensive”, que segue princípios como a utilização adequada dos métodos HTTP, a definição de URIs consistentes e a transferência de dados no formato JSON.

Utilizando-se destas tecnologias, o servidor do *back-end* recebe as requisições do *front-end* e então retorna os dados necessários, tornando a aplicação muito mais segura para os usuários, já que seus dados estarão atrás de uma camada de segurança fornecida pelo *back-end*.

# 7.CONCLUSÃO

Ao longo deste documento, foi explorado a implementação de um *front-end* utilizando as tecnologias Javascript, HTML e CSS, que se comunica com um *back-end* em Java por meio da FETCH API, seguindo a arquitetura REST. Desta maneira criando um *web service* de fácil e rápida utilização para o usuário.

Durante o processo de desenvolvimento, foi priorizado a usabilidade, considerando as diretrizes e melhores práticas recomendadas por diversos pesquisadores da área, tendo como exemplo, Jakob Nielsen.

Através do uso do Javascript, interatividade e dinamismo foi adicionado nas páginas, permitindo que os usuários interajam com elementos e façam requisições HTTP de forma assíncrona. A FETCH API foi utilizada para enviar essas requisições para o back-end em Java, seguindo os princípios da arquitetura REST.

Através da arquitetura REST, o *front-end* estabeleceu uma comunicação eficiente e padronizada com o *back-end*, permitindo a troca de dados de forma consistente e confiável. Essa abordagem contribuiu para a separação de responsabilidades entre as camadas do sistema e facilitou a manutenção e escalabilidade da aplicação.

Em resumo, a implementação do *front-end* com a integração da FETCH API e a adoção da arquitetura REST, resultou em uma aplicação que prioriza a usabilidade e proporciona uma experiência de usuário positiva. A atenção às práticas de design responsivo, organização do conteúdo e interatividade contribuiu para um *site* coeso e eficiente, além de completamente funcional.

# 8. REFERÊNCIAS

<https://www.yworks.com/yed-live/>;

Barbosa, Simone e Bruno Silva: Interação humano-computador. Elsevier Brasil, 2010;

Rogers, Yvonne, Helen Sharp e Jennifer Preece: Design de interação. Bookman Editora, 2013;

<https://blog.counter-strike.net>;

<https://steamdb.info>;

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP>;

https://www.csgodatabase.com/skins;