**Materiais e Métodos**

Será apresentado nesta seção um pouco sobre o tema e o que temos no mercado atualmente. Serão apresentadas as tecnologias escolhidas para o desenvolvimento da aplicação e suas aplicações.

***Javascript***

De acordo com Resig e Bibeault (2013), *Javascript* é uma linguagem de programação versátil e de amplo uso atualmente, que nos fornece ferramentas para operar tanto no *front-end* quanto no *back-end*.

Ainda segundo Resig e Bibeault (2013), *Javascript* é de extrema importância, visto que a mesma performa entre as 3 linguagens mais utilizadas no mundo. De acordo com uma pesquisa realizada pela empresa de desenvolvimento de *software* *RedMonk: “The RedMonk Programming Language Rankings: June 2019”*, na qual foi consultada a base de dados de dois dos maiores repositórios de código aberto na Internet (GitHub e StackOverflow), Javascript é hoje a linguagem de programação mais utilizada no globo. Isto nem sempre foi assim, porém hoje é uma realidade. Esperamos que aplicações web deem ao usuário uma rica interação com a interface, proporcionando uma experiência envolvente. Mais do que nunca, desenvolvedores precisam ter um profundo entendimento sobre a linguagem que traz vida aos sistemas *web*. Além disso, o *javascript* não é limitado mais somente aos *browsers*, como antigamente. Hoje, códigos *javascript* podem ser executados no *server-side* graças a ambientes e frameworks, como o *Node.Js*.

*Javascript* possui ainda muitas peculiaridades, como a possibilidade de executar funções de forma assíncrona. Tal funcionalidade é de grande utilidade, aumentando consideravelmente o poder de processamento da linguagem.

Com a evolução da linguagem *Javascript* durante as décadas e a crescente demanda por novas tecnologias que melhor pudessem resolver os problemas da atualidade, surgiu então, em 2009, o *Node.Js*, criado por Ryan Dahl.

**Node.Js**

*Node.js* é um interpretador de códigos *javascript,* que permite que códigos escritos nesta linguagem possam ser executados fora do navegador. Isto abre um mundo inteiro de novas possibilidades, arquiteturas e formas de construir aplicações.

É possível, por exemplo, criar aplicações *web*, *desktop*, *client-side* ou *server-side*.

O ambiente *Node.js* utiliza a *engine* da *Google*: V8, um interpretador de códigos *javascript*, utilizado dentro do navegador *Google Chrome*.

Segundo Ryan Dahl, criador do node.js, a própria criação do software teve muito a ver com o *Google*, sua release do *Google Chrome* e com o interpretador V8. Sem este acontecimento, o nascimento do *node.js* não teria sido possível.

**Selenium**

*Selenium* é um *framework* para testes de aplicações *web*. Foi criado inicialmente por Jason Huggins, em 2004. Com *Selenium*, conseguimos automatizar diversas tarefas nos navegadores, desde pequenas rotinas até grandes testes *(Selenium Website, 2020).*

O *framework selenium* suporta diversas linguagens, como *Python*, *Ruby* e *Javascript*. Para este trabalho, será utilizada a linguagem *Javascript*.

Este *framework* nos fornece diversas ferramentas para manipular e simular interações com o navegador, interagindo diretamente com alguns componentes do mesmo, como DOM - *Document Object Model*. Podemos acessar métodos, atributos *HTML*, inserir e alterar informações e até mesmo simular completamente a interação de um ser humano utilizando um navegador.

Podemos pensar em problemas reais, corporativos, que seriam beneficiados com o uso de tecnologias como esta. Para o propósito deste trabalho, tomaremos como exemplo um *e-commerce*, que precisa constantemente se manter dentro dos preços praticados pelo mercado, para continuar concorrendo pelas melhores posições nos sites de busca.

Com uma quantidade imensa de concorrentes, uma quantidade maior ainda de produtos para observar e preços que mudam constantemente, todos os dias, faz-se necessário o uso de tecnologias, como automações, para aumento da competitividade.