DESAFIO TÉCNICO – AUTOMAÇÃO SITE DE DIGITAÇÃO

## Escopo do Projeto

* Trata-se de uma automação de um site de digitação, onde ela deve capturar os dados do campo de palavras e inserir no campo de digitação uma por vez;
* Após o preenchimento dos dados e a obtenção dos resultados, eles devem ser salvos em um banco de dados a escolha do programador;

## Informações Principais

1. O site a ser automatizado: <https://10fastfingers.com/typing-test/portuguese>;
2. Linguagem de Programação: C#;
3. Biblioteca/Pacotes: .net Core, Selenium e SQL Lite;
4. Github: https: [//github.com/RaposaNevada/autodigitacao.git](https://github.com/RaposaNevada/autodigitacao.git)
5. Utilizando a inspeção de elementos do site foi possível adquirir os ids e classes para utilização no mapeamento com Selenium:
   1. Id=”words”;
   2. Id=”inputfield”;
   3. Id=”result-table”;
   4. Id=”wpm”;
   5. Id=”keystrokes”;
   6. Id=”accuracy”;
   7. Id=”correct”;
   8. Id=”wrong”;
6. Navegador Padrão para a automação: Microsoft Edge;
7. Foi formado uma tabela com os resultados para ser adicionada ao banco de dados temporário SQL Lite;
8. A escolha do banco de dados se deu a disponibilidade e autorização das ferramentas utilizadas no notebook corporativo;

## Desenvolvimento

1. Foi iniciado o desenvolvimento com base na procura dos elementos de palavra e também do campo de digitação para que a estrutura do primeiro tópico do desafio fosse cumprida. Após a localização do id “words” foi idenficado que a palavra ativa era destacada com um class=”highlight”, o que foi utilizado para a captura da palavra a ser inserida no campo de id “inputfield”. O Console.WriteLine foi de extrema importância para que possamos identificar se o processo de localização da palavra ativa e o input dela estivesse funcionando, sendo possível acompanhar esse processo até pelo console.
2. Como esse processo deve acontecer durante 1 minuto até que o resultado apareça, foi utilizada uma estrutura de repetição “while” e criado um temporizador para que o processo de localizar a palavra e adicionar no campo texto fosse repetido até que o temporizador finalizasse.
3. Ao fim do tempo, a tabela de resultados aparecia e dela precisávamos retirar as informações wpm, keystrokes, accuracy, correct e wrong para transportamos ao banco de dados. Para isso, foi localizado as informações pelos seus respectivos Ids e montada uma tabela temporária de nome “resultsTable”.
4. O SQL Lite é um banco de dados em biblioteca que é capaz de armazenar os dados de pequenas aplicações em um arquivo. A sua interação pode ocorrer tanto com o aplicativo como também direto pelo código, o que foi o caso utilizado. Foi criado um banco de dados de nome ”results.db” e uma tabela “Results”, adicionado a tabela temporária “resultsTable” antes criada nessa nova tabela.

## Conclusão

* Após o desenvolvimento da aplicação e os testes, concluísse que a os requisitos do desafio foram preenchidos. Porém, há sempre espaço para melhorias e utilização de pacotes na automação. É importante destacar que o código foi desenvolvido com a limitação de ferramentas disponíveis para o programador.