Filtragem de Ocorrências por Associado

Documentação de Projeto

Versão: 3.0

6 de novembro de 2022

**Sumário**

1. [Introdução 3](#_bookmark0)
   1. [Finalidade 3](#_bookmark1)
   2. [Escopo Geral 3](#_bookmark2)
      1. [Escopo Sprint 1 3](#_bookmark3)
      2. Escopo Sprint 2 ................................................................................................................3
      3. Escopo Sprint 3 ................................................................................................................3
   3. [Visão Geral 3](#_bookmark4)
2. [Diagrama de Classes 4](#_bookmark5)
3. [Mer/Der 4](#_bookmark6)
   1. [Modelo Entidade Relacionamento 4](#_bookmark7)
   2. [Diagrama Entidade Relacionamento 5](#_bookmark8)
4. [Manual de Inicialização 5](#_bookmark9)
5. [Manual de Utilização 7](#_bookmark10)

# Introdução

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema. O objetivo deste documento é comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema

## Escopo Geral

Desenvolver Crawler que consiga buscar, ler o arquivo e identificar o conteúdo relacionado aos Professores Associados, realizar os cruzamentos necessários com as bases de dados de associados, que suportarão está solução, gravar todos os processos publicados relacionados aos associados, e emitir relatório diário para que a equipe consiga disparar os e-mails para os professores citados no Diário Oficial. Caso consigam também realizar a automação do disparo destes e-mails seria uma sofisticação para a solução.

## Escopo Sprint 1

Realizar o cadastro dos associados e armazenar em Banco de dados, fazer o download das páginas do dia atual e em que os associados cadastrados sejam citados. Exibir no front-end, uma tabela que mostre o Nome do associado, E-mail e o botão que abrirá a página dele. Automação dos downloads e exclusão dos PDF.

* + 1. **Escopo Sprint 2**

Realizar a raspagem de trechos da página e exibir no front-end uma página que mostre o nome do associado, o e-mail e o trecho.

* + 1. **Escopo Sprint 3**

Criação dos relatórios para controle das ocorrência dos associados.

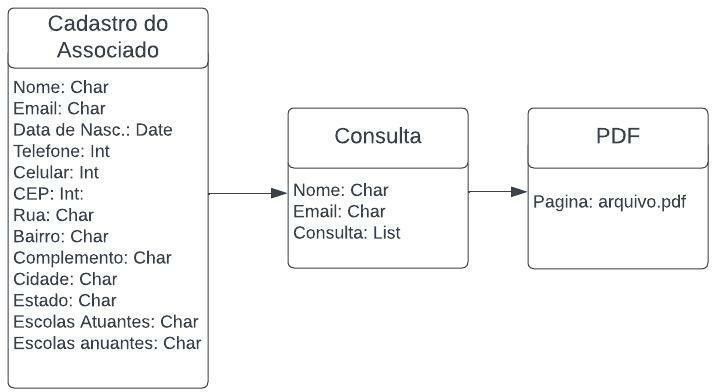
## Visão Geral

Nesse documento conterá informações sobre o software necessário para maior entendimento e aplicação e está organizado em:

* Diagrama de Classes.
* Arquitetura do Software
* DER/MER
* Manual de Inicialização
* Manual de Utilização

# Diagrama de Classes

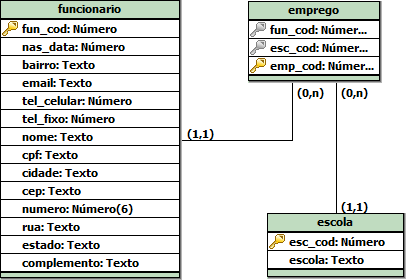
Nesse conteúdo é apresentado o Diagrama de Classes, descrevendo a estrutura do nosso sistema, mostrando as classes, atributos, operações e as relações entre os objetos.



# Mer/Der

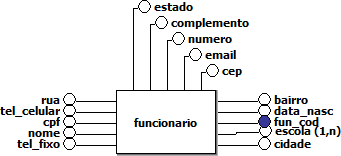
## Modelo Entidade Relacionamento

Utilizamos MER para descrever os objetos do sistema através de entidades com suas propriedades que são os seus atributos e seus relacionamentos.



## Diagrama Entidade Relacionamento

Utilizamos o DER para representar em forma gráfica o que foi descrito no MER.



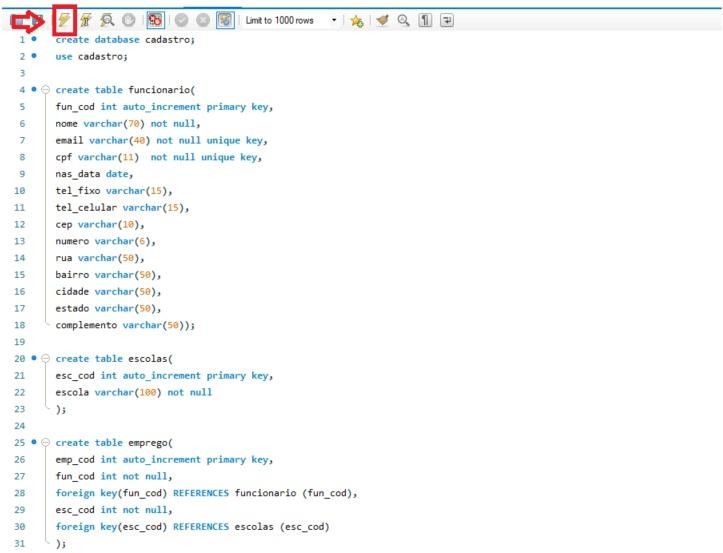
# Manual de Inicialização

Siga passo a passo para fazer a inicialização de forma correta.

1. É necessário ter o Visual Studio Code, Node.js, MySQL, Python 3.10 e o Git instalado na máquina;
2. Caso não os tenha, instale pelo <https://code.visualstudio.com/Download>

[,https://nodejs.org/en/,](https://nodejs.org/en/) [https://dev.mysql.com,](https://dev.mysql.com/) <https://www.python.org/downloads/>e [https://git-scm.com/downloads;](https://git-scm.com/downloads)

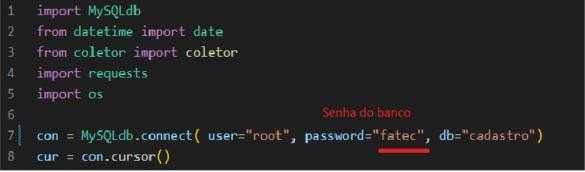
1. Crie uma pasta de sua preferência em sua máquina;
2. Abra o cmd e digite: git clone <https://github.com/All-Tech-DSM/API-2.git>;
3. Feche o cmd;
4. Abra o MySQL e execute o arquivo script.sql;



1. Vá à pasta 📂 API-2;
2. Abra a pasta no seu Editor de Códigos;
3. Em seguida acesse a pasta 📂 server e abra o arquivo index.js;
4. Vá na linha 06 e configure a rota de seu MySQL;



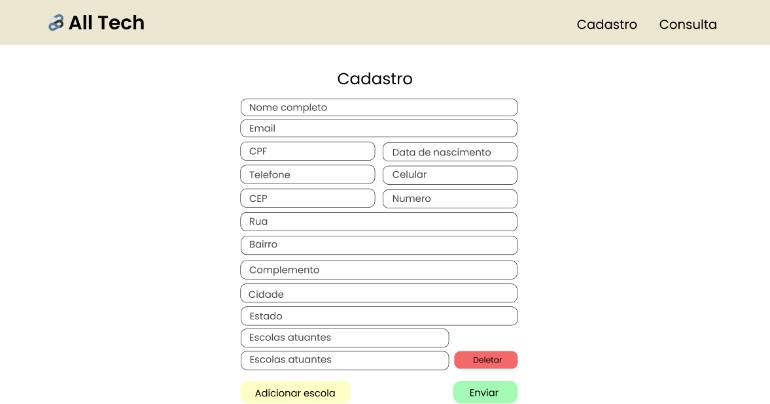
1. Em seguida salve o arquivo e feche-o;
2. Abre o arquivo *“*buscador.py*”* e configure os dados do MySQL na linha 7;



1. Ainda na pasta 📂 server abra o cmd e digite: *“*npm install*”*;
2. Em seguida, digite o comando *“*pip install -r req.txt*”*;
3. Após a instalação, digite: node index.js;
4. Crie outro terminal e entre na pasta 📂 `server`:
5. Digite: python app.py;
6. Abra um novo terminal; **17.**Em seguida, acesse à pasta 📂 script e abra o cmd; **18.**Digite: npm install;
7. Em seguida, digite: npm start;
8. A página já estará estanciada em sua localhost;

# 5. Manual de Utilização

Após executar todos os comandos anterior, o sistema devera abrir essa tela.



Nesta tela é onde será realizada o cadastro dos associados. Após preencher todos os campos e clicando no botão enviar, todos esses dados serão armazenados no banco de dados ‘Cadastro’.

Concluindo o cadastro você já pode fazer sua primeira consulta de associado, clicando no canto superior direito em ‘Consulta’ deverá aparecer está tela.



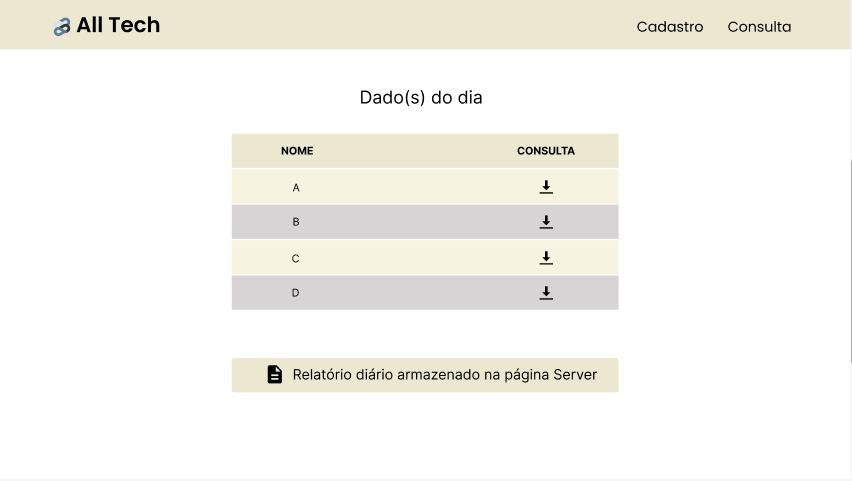
Nessa tela será mostrado todos os associados que foram citados naquele dia juntamente com um botão que ao clicar irá abrir em uma nova aba o PDF da página em que eles são citados.



Nessa tela é mostrado o trecho adquirido através da raspagem do associado escolhido.

**Relatorio Diario**

Todos os dias que houver menção dos associados cadastrados em nosso sistema, será gerado um relatorio com as suas identificações e seus respectivos trechos. Esses Documentos estão sendo armazenados no diretorio 📂Server com a identificação “Relatorio<Data do dia em questão>”.



**Local de Armazenamento**

* Api-2/server/relatorio<data>