

# Manual de administração da aplicação CienciaScraper

## Introdução

Este manual aborda os fundamentos de administração e manutenção aplicação desenvolvida, garantindo assim uma correta intervenção do sistema por um administrador.

Para correr a aplicação sem erros e com a melhor experiência possível, deve garantir os seguintes requisitos mínimos:

- **Sistema de Gerenciamento de Bases de Dados (DBMS):**
  - Para ambientes Windows, Microsoft SQL Server Management Studio (SSMS) 18.0 ou superior;
  - Para ambientes macOS, Microsoft Azure Data Studio 1.37.0 ou superior.
- **Navegador da Internet:** Browser baseado em Chromium, atualizado até a última versão (Microsoft Edge, Google Chrome)
- **Sistema operativo:** Windows 7 ou superior; Mac OS X 10 ou superior.
- **Memória RAM:** 4 GB
- **Rede:** Conexão de internet banda larga

A gestão e administração da aplicação é realizada exclusivamente através da manipulação da base de dados. Não existe uma página de *back-office* que permita a manipulação das tabelas de metadados.

Para que sempre sejam feitas manutenções consistentes, e de modo a tirar o maior proveito da aplicação, recomendamos que o administrador da aplicação possua as seguintes competências:

- Conhecimentos de JavaScript;
- Conhecimentos de HTML, XML e XPath;
- Conhecimentos de SQL.

Caso o administrador falhe em alguma das competências acima, não será possível garantir o pleno funcionamento da aplicação, bem como garantir que as intervenções realizadas tenham sucesso.

Este guia irá mostrar como realizar as operações básicas de manutenção e gestão:

- Gestão de utilizadores
- Gestão de relatórios
- Gestão do motor de extração dos currículos

## Gestão de utilizadores

Para realizar administrações nesta área, é necessário possuir:

- Conhecimentos de SQL.

Os utilizadores são criados na tabela [User].

Poderá, nesta tabela:

- Alterar o username de um utilizador;
- Alterar o e-mail de um utilizador;
- Alterar o nome e apelido de um utilizador;
- Alterar a password de um utilizador;
- Alterar o perfil na aplicação;
- Definir se o utilizador está ativo ou inativo;
- Visualizar a data de criação.

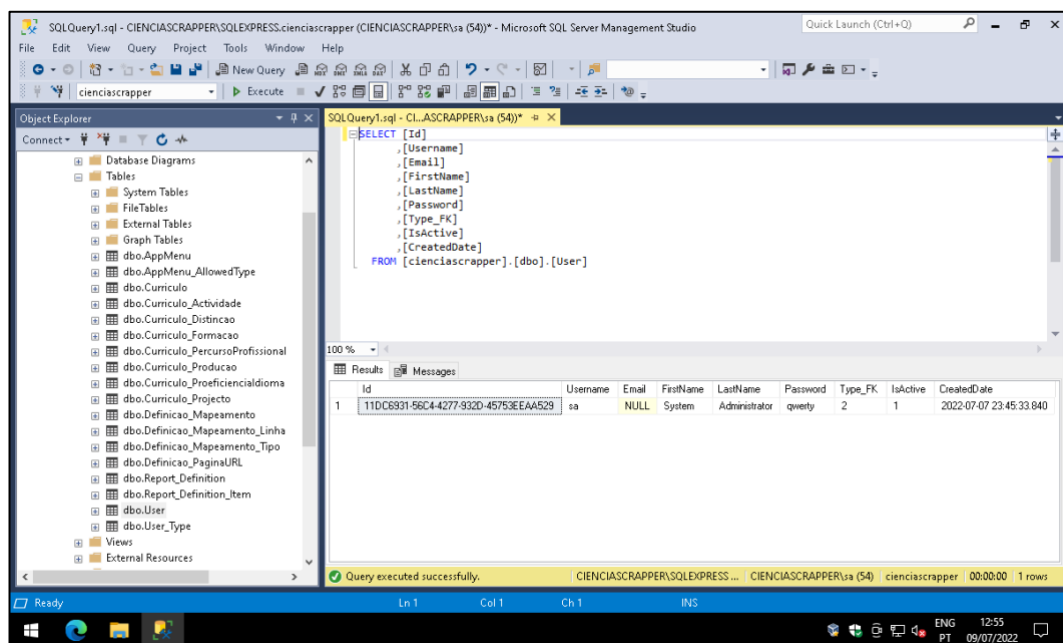


Figura I: Consulta à tabela de utilizadores

O dicionário de dados é como segue:

Nome do campo	Tipo de dado	Obrigatório	Descrição
Id	uniqueidentifier	Sim	Chave primária da tabela de utilizadores
Username	nvarchar	Sim	Nome de autenticação do utilizador, e deve ser único
Email	nvarchar	Não	E-mail do utilizador
FirstName	nvarchar	Sim	Nome do utilizador
LastName	nvarchar	Sim	Apelido do utilizador
Password	nvarchar	Sim	Palavra-passe do utilizador
Type_FK	int	Sim	Chave estrangeira do perfil de utilizador
IsActive	bit	Sim	Indica se o utilizador está ativo
CreateDate	datetime	Sim	Indica a data de criação do utilizador

Tabela I: Dicionário de dados para [User]

Para criar uma conta, recomendamos que sempre faça através da interface gráfica da aplicação. Em caso de ser necessário criação massiva de utilizadores, também poderá fazer através de comandos SQL à tabela [User], preenchendo sempre os campos obrigatórios e respeitando as restrições da base de dados.

## Gestão de relatórios

Para realizar administrações nesta área, é necessário possuir:

- Conhecimentos de SQL.

Os utilizadores são criados na tabela [Report\_Definition].

Poderá, nesta tabela:

- Criar relatórios;
- Alterar relatórios existentes;
- Desativar relatórios.

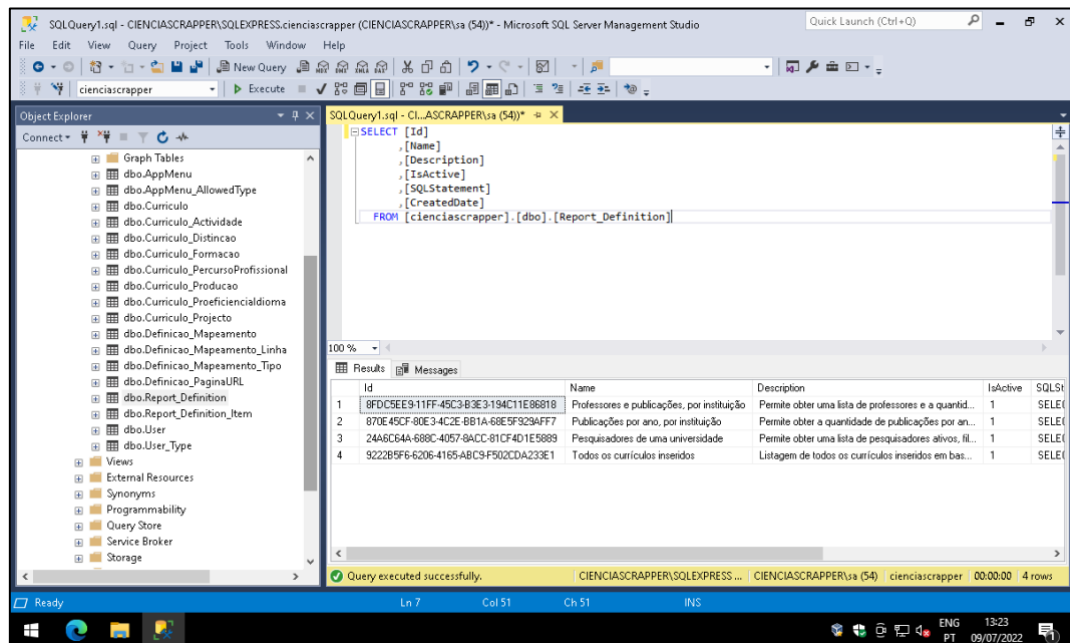


Figura II: Consulta à tabela de relatórios

Apesar de não existir um campo que indique esta tipologia, um relatório pode ser de dois tipos:

- **Relatório estático/simples**
  - Os relatórios estáticos/simples são aqueles que é apenas obtida uma listagem de informação com base em parâmetros de filtragem pré-definidos pelo administrador do sistema.
  - É necessário existir registos apenas na tabela [Report\_Definition]
- **Relatório dinâmico**
  - Os relatórios dinâmicos são aqueles em que, para mostrar alguma informação, é necessário que o utilizador defina parâmetros de filtragem. Estes são inseridos na query de consulta, e então apresentados os resultados aos utilizadores.
  - É necessário existir registos na tabela [Report\_Definition] e [Report\_Definition\_Item].

O dicionário de dados para a definição de relatório é como segue:

Nome do campo	Tipo de dado	Obrigatório	Descrição
Id	uniqueidentifier	Sim	Chave primária da tabela de definições de relatório
Name	nvarchar	Sim	Nome do relatório
Description	nvarchar	Não	Descrição do relatório, para ajudar o utilizador
IsActive	bit	Sim	Define se o relatório está ativo e disponível para execução
SQLStatement	nvarchar	Sim	Consulta SQL preparada para execução do relatório
CreatedDate	datetime	Sim	Data de criação do relatório

Tabela II: Dicionário de dados para [Report\_Definition]

No caso de relatórios dinâmicos, é necessário declarar os parâmetros na tabela [Report\_Definition\_Item]. O dicionário de dados é como segue:

Nome do campo	Tipo de dado	Obrigatório	Descrição
Id	uniqueidentifier	Sim	Chave primária da tabela de definições de filtro
Report_Definition_FK	uniqueidentifier	Sim	Chave estrangeira, que relaciona os filtros com a definição de relatório
ParameterName	nvarchar	Sim	Nome do parâmetro. Deve existir na query construída "SQLStatement"
ParameterLabel	nvarchar	Sim	Rótulo do campo, para apresentar ao utilizador no front-end.
Datatype	nvarchar	Sim	Obrigatório estar preenchido com: "String", "Date", ou "Integer"

Tabela III: Dicionário de dados para [Report\_Definition\_Item]

Existem palavras reservadas, que podem ser utilizadas para personalizar o resultado obtido na aplicação.

Funcionalidade	Nome das colunas	Descrição da coluna	Exemplo de utilização
Gerar hyperlink com URL	URL	Hyperlink navegável	<pre>SELECT [Id] ,CONCAT('/curriculum.html?id=', [Id]) AS 'URL' ,'Id' AS 'URL_Target' FROM [Curriculo]</pre>
	URL_Target	Coluna para atribuir o hyperlink	

Tabela IV: Nomes de coluna reservados

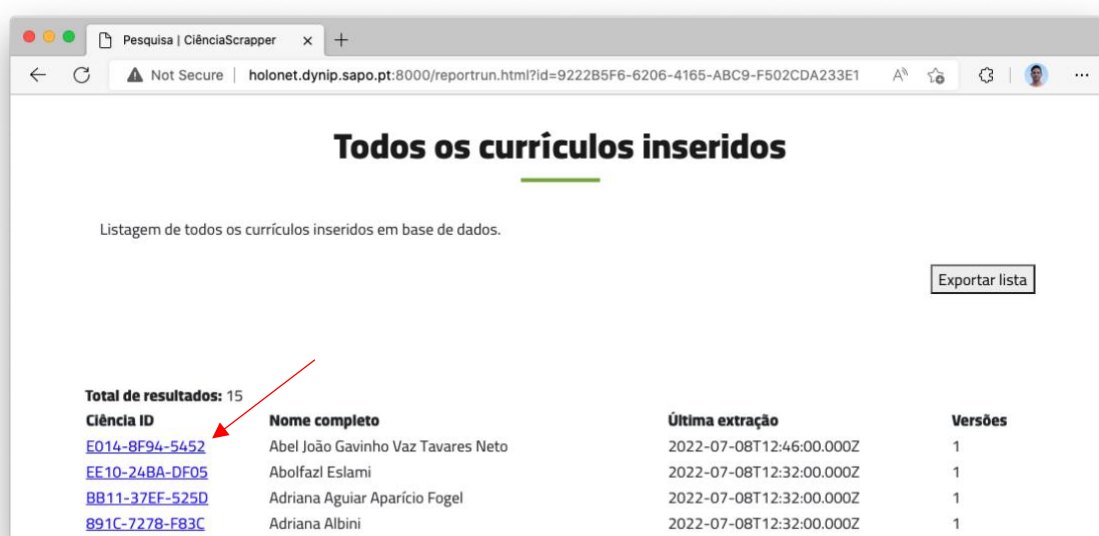
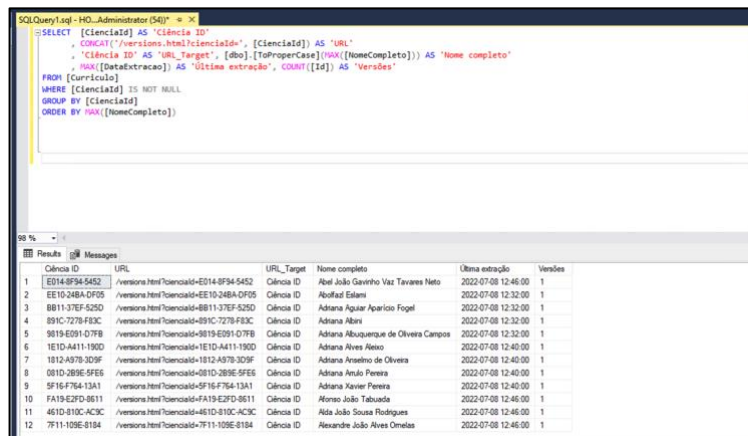


Figura III: Exemplo de atribuição de hyperlink por meio das colunas “URL” e “URL\_Target”

Para a criação de um relatório simples, o administrador deve criar um comando SQL estruturado para que o motor de relatórios consiga executar a consulta. Para exemplificar, tentaremos criar um relatório que faça a listagem de todos os currículos inseridos na base de dados, e a data de última inserção.

Os currículos são armazenados na tabela [Currículo].



The screenshot shows a SQL query in a text editor and its results in a table. The query is as follows:

```
SELECT [CienciaId] AS 'Ciência ID',  
       CONCAT('/versions.html?cienciaId=', [CienciaId]) AS 'URL',  
       'Ciência ID' AS 'URL_Target', [dbo].[ToProperCase](MAX([NomeCompleto])) AS 'Nome completo',  
       MAX([DataExtracao]) AS 'Última extração', COUNT([Id]) AS 'Versões'  
FROM [Currículo]  
WHERE [CienciaId] IS NOT NULL  
GROUP BY [CienciaId]  
ORDER BY MAX([NomeCompleto])
```

The results table has the following columns: Ciência ID, URL, URL\_Target, Nome completo, Última extração, and Versões. It contains 12 rows of data.

Ciência ID	URL	URL_Target	Nome completo	Última extração	Versões
EE14-8F94-5452	/versions.html?cienciaId=EE14-8F94-5452	Ciência ID	Atel João Gavinho Vaz Tavares Neto	2022-07-08 12:46:00	1
EE10-24BA-DF05	/versions.html?cienciaId=EE10-24BA-DF05	Ciência ID	Rafael Estani	2022-07-08 12:32:00	1
BB11-37EF-525D	/versions.html?cienciaId=BB11-37EF-525D	Ciência ID	Adriana Aguiar Aparicio Fogel	2022-07-08 12:32:00	1
891C-7278-F83C	/versions.html?cienciaId=891C-7278-F83C	Ciência ID	Adriana Albari	2022-07-08 12:32:00	1
9819-E091-07F8	/versions.html?cienciaId=9819-E091-07F8	Ciência ID	Adriana Albuquerque de Oliveira Campos	2022-07-08 12:32:00	1
1E1D-A411-196D	/versions.html?cienciaId=1E1D-A411-196D	Ciência ID	Adriana Alves Rêgo	2022-07-08 12:40:00	1
1B12-67B3-329F	/versions.html?cienciaId=1B12-67B3-329F	Ciência ID	Adriana Assunção de Oliveira	2022-07-08 12:40:00	1
081D-2B95-5FE5	/versions.html?cienciaId=081D-2B95-5FE5	Ciência ID	Adriana Araújo Pereira	2022-07-08 12:40:00	1
5F16-F764-13A1	/versions.html?cienciaId=5F16-F764-13A1	Ciência ID	Adriana Xavier Pereira	2022-07-08 12:40:00	1
FA19-E2FD-8611	/versions.html?cienciaId=FA19-E2FD-8611	Ciência ID	Alonso João Tabuada	2022-07-08 12:46:00	1
461D-810C-AC3C	/versions.html?cienciaId=461D-810C-AC3C	Ciência ID	Aida João Sousa Rodrigues	2022-07-08 12:46:00	1
7F11-109E-8184	/versions.html?cienciaId=7F11-109E-8184	Ciência ID	Alexandre João Alves Ometas	2022-07-08 12:46:00	1

Portanto, para criar uma consulta estruturada à tabela [Currículo] através de um relatório simples, podemos inserir o seguinte registro:

### Nome do relatório

Todos os currículos inseridos

### Descrição

Listagem de todos os currículos inseridos em base de dados.

### Consulta SQL

```
SELECT [CienciaId] AS 'Ciência ID', CONCAT('/versions.html?cienciaId=',  
[CienciaId]) AS 'URL', 'Ciência ID' AS 'URL_Target',  
[dbo].[ToProperCase](MAX([NomeCompleto])) AS 'Nome completo', MAX([DataExtracao])  
AS 'Última extração', COUNT([Id]) AS 'Versões' FROM [Currículo] WHERE [CienciaId]  
IS NOT NULL GROUP BY [CienciaId] ORDER BY MAX([NomeCompleto])
```

### Comando SQL para criar o relatório

```
INSERT INTO [Report_Definition] ([Name], [Description], [IsActive], [SQLStatement])  
VALUES ('Todos os currículos inseridos', 'Listagem de todos os currículos inseridos  
em base de dados.', 1, 'SELECT [CienciaId] AS 'Ciência ID',  
CONCAT('/versions.html?cienciaId=', [CienciaId]) AS 'URL', 'Ciência ID' AS  
'URL_Target', [dbo].[ToProperCase](MAX([NomeCompleto])) AS 'Nome completo',  
MAX([DataExtracao]) AS 'Última extração', COUNT([Id]) AS 'Versões' FROM  
[Currículo] WHERE [CienciaId] IS NOT NULL GROUP BY [CienciaId] ORDER BY  
MAX([NomeCompleto])')
```

Para a criação de um relatório dinâmico, o administrador deve criar um comando SQL estruturado para que o motor de relatórios consiga executar a consulta. Deverá conter filtros com palavras-chave, sendo estas inseridas como registros na tabela [Report\_Definition\_Item]. Para exemplificar, tentaremos criar um relatório que faça a listagem de todos os currículos inseridos na base de dados, que permita obter os pesquisadores ativos de uma determinada instituição.

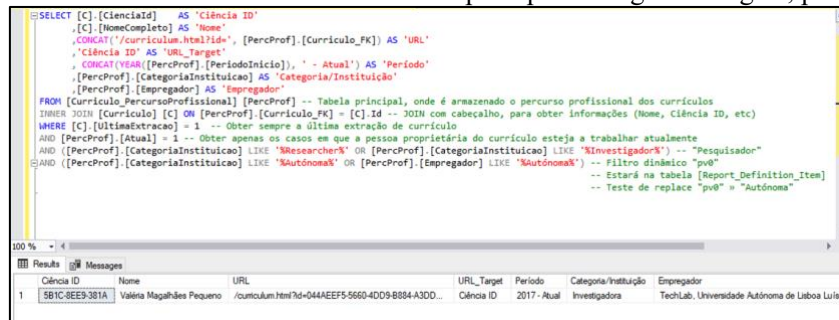
Os currículos são armazenados na tabela [Currículo].

Os empregos/percurso profissional estão armazenados na tabela [Currículo\_PercursoProfissional].

Exemplo de consulta SQL, com parâmetro dinâmico “pv0”:

```
SELECT [C].[CienciaId] AS 'Ciência ID',
       [C].[NomeCompleto] AS 'Nome',
       CONCAT('/curriculum.html?id=', [PercProf].[Currículo_FK]) AS 'URL',
       'Ciência ID' AS 'URL_Target',
       CONCAT(YEAR([PercProf].[PeriodoInicio]), ' - Atual') AS 'Período',
       [PercProf].[CategoriaInstituicao] AS 'Categoria/Instituição',
       [PercProf].[Empregador] AS 'Empregador'
FROM [Currículo_PercursoProfissional] [PercProf] -- Tabela principal, onde é armazenado o percurso profissional dos currículos
INNER JOIN [Currículo] [C] ON [PercProf].[Currículo_FK] = [C].Id -- JOIN com cabeçalho, para obter informações (Nome, Ciência ID, etc)
WHERE [C].[UltimaExtracao] = 1 -- Obter sempre a última extração de currículo
AND [PercProf].[Atual] = 1 -- Obter apenas os casos em que a pessoa proprietária do currículo esteja a trabalhar atualmente
AND ([PercProf].[CategoriaInstituicao] LIKE '%Researcher%' OR [PercProf].[CategoriaInstituicao] LIKE '%Investigador%') -- "Pesquisador"
AND ([PercProf].[CategoriaInstituicao] LIKE '%pv0%' OR [PercProf].[Empregador] LIKE '%pv0%') -- Filtro dinâmico "pv0"
-- Estará na tabela [Report_Definition_Item]
```

Simular o motor de relatório: substituir “pv0” pela string de filtragem, por exemplo, “Autónoma”:



Portanto, para criar um relatório dinâmico conforme exemplificado acima, necessitaremos:

### Nome do relatório

Pesquisadores de uma universidade

### Descrição

Permite obter uma lista de pesquisadores ativos, filtrados por instituição.

### Consulta SQL

```
SELECT [C].[CienciaId] AS 'Ciência ID', [C].[NomeCompleto] AS 'Nome',
CONCAT('/curriculum.html?id=', [PercProf].[Currículo_FK]) AS 'URL', 'Ciência ID' AS
'URL_Target', CONCAT(YEAR([PercProf].[PeriodoInicio]), ' - Atual') AS 'Período',
[PercProf].[CategoriaInstituicao] AS 'Categoria/Instituição', [PercProf].[Empregador] AS
'Empregador' FROM [Currículo_PercursoProfissional] [PercProf] INNER JOIN [Currículo] [C] ON
[PercProf].[Currículo_FK] = [C].Id WHERE [C].[UltimaExtracao] = 1 AND [PercProf].[Atual] = 1
AND ([PercProf].[CategoriaInstituicao] LIKE '%Researcher%' OR
[PercProf].[CategoriaInstituicao] LIKE '%Investigador%') AND
([PercProf].[CategoriaInstituicao] LIKE '%pv0%' OR [PercProf].[Empregador] LIKE '%pv0%')
```

### Parâmetros da consulta

- Nome do parâmetro: pv0
- Tipo: String

### Comando SQL para criar o relatório

```
INSERT INTO [Report_Definition] ([Name], [Description], [IsActive], [SQLStatement]) VALUES
('Pesquisadores de uma universidade', 'Permite obter uma lista de pesquisadores ativos,
filtrados por instituição.', 1, 'SELECT [C].[CienciaId] AS 'Ciência ID', [C].[NomeCompleto]
AS 'Nome', CONCAT('/curriculum.html?id=', [PercProf].[Currículo_FK]) AS 'URL',
'Ciência ID' AS 'URL_Target', CONCAT(YEAR([PercProf].[PeriodoInicio]), ' - Atual') AS
'Período', [PercProf].[CategoriaInstituicao] AS 'Categoria/Instituição',
[PercProf].[Empregador] AS 'Empregador' FROM [Currículo_PercursoProfissional] [PercProf]
INNER JOIN [Currículo] [C] ON [PercProf].[Currículo_FK] = [C].Id WHERE [C].[UltimaExtracao]
= 1 AND [PercProf].[Atual] = 1 AND ([PercProf].[CategoriaInstituicao] LIKE '%Researcher%'
OR [PercProf].[CategoriaInstituicao] LIKE '%Investigador%') AND
([PercProf].[CategoriaInstituicao] LIKE '%pv0%' OR [PercProf].[Empregador] LIKE
'%pv0%')')
```

Após inserir, capturar o [Id] gerado pela base de dados para o registo criado. Será utilizado no statement SQL abaixo:

Vamos supor que o [Id] gerado foi = "24A6C64A-688C-4057-8ACC-81CF4D1E5889". Portanto:

```
INSERT INTO [Report_Definition_Item] ([Report_Definition_FK], [ParameterName], [ParameterLabel], [Datatype]) VALUES ('24A6C64A-688C-4057-8ACC-81CF4D1E5889', 'pv0', 'Instituição/Empregador', 'String')
```

## Gestão do motor de extração dos currículos

Para realizar administrações nesta área, é necessário possuir:

- Conhecimentos de JavaScript;
- Conhecimentos de HTML, XML e XPath;
- Conhecimentos de SQL.

Esta secção está diretamente relacionada com o motor de extração implementado na aplicação. Recomendamos que a análise, manutenções ou intervenções que sejam necessários nesta parte da aplicação estejam sempre acompanhadas de análises em conjunto entre:

- **Tabelas [Definicao\_PaginaURL], [Definicao\_Mapeamento\_Tipo], [Definicao\_Mapeamento] e [Definicao\_Mapeamento\_Linha]**
  - Iremos abordar a seguir o dicionário de dados para as tabelas.
- **Código JavaScript localizado no ficheiro**  
**"..\cienciascrapper\controllers\scrape\_cienciavitae.js"**
  - Este se encontra devidamente comentado, para possibilitar adquirir conhecimento através de uma leitura crítica ao código.

Conforme indicado acima, existem 4 tabelas de grande importância para o motor de extração dos currículos, implementado com base em conceitos de *web scraping*. São elas:

Nome da tabela	Descrição	Relacionamento
Definicao_PaginaURL	Contém a URL ativa que o motor considera para localizar um CV através do Ciência Id	---
Definicao_Mapeamento_Tipo	Tabela de metadados, que contém a tipologia dos mapeamentos. Pode ser considerado como "morfologia da secção" na página do CV	Um "tipo" pode conter vários mapeamentos.
Definicao_Mapeamento	Tabela de metadados, que mapeia o XPath da secção do currículo com uma tabela do SQL Server. Não é auto-suficiente, é obrigatório que existam Linhas.	Um "mapeamento" contém um "tipo"; Um "mapeamento" irá conter várias "linhas".
Definicao_Mapeamento_Linha	Tabela de metadados, que mapeia o XPath da secção do currículo, ou o índice/coluna da tabela extraída no HTML com uma coluna da tabela do SQL Server.	Uma "linha" contém um "mapeamento"

Tabela V: Tabelas relevantes para o motor de web scraping, e os seus significados

O dicionário de dados para a tabela [Definicao\_PaginaURL] do SQL é como segue:

Nome da coluna	Tipo de dado	Descrição
Id	uniqueidentifier	Chave primária da tabela de URLs
URL	nvarchar	URL absoluta para a pesquisa, onde será concatenado o Ciência ID
IsActive	bit	Indica se está ativo e deve ser considerado pelo motor. Apenas 1 registo pode ter IsActive = 1
CreatedDate	datetime	Indica a data de criação do registo.

Tabela VI: Dicionário de dados para a tabela [Definicao\_PaginaURL]

O dicionário de dados para a tabela [Definicao\_Mapeamento\_Tipo] do SQL é como segue:

Nome da coluna	Tipo de dado	Descrição
Id	uniqueidentifier	Chave primária da tabela de tipologia de secções
Nome	nvarchar	Designação administrativa da tipologia de secção

Tabela VII: Dicionário de dados para a tabela [Definicao\_Mapeamento\_Tipo]

O dicionário de dados para a tabela [Definicao\_Mapeamento] do SQL é como segue:

Nome da coluna	Tipo de dado	Descrição
Id	uniqueidentifier	Chave primária da tabela de mapeamento secção vs. Tabela
NomeTabela	nvarchar	Nome da tabela SQL
TipoDefinicao_FK	uniqueidentifier	Chave estrangeira da tipologia da secção
Ordem	int	Ordem para extração
Descricao	ntext	Descrição administrativa do mapeamento
XPath	nvarchar	Caminho XPath a considerar para a extração
IsActive	bit	Indica se está ativo e, portanto, se o motor deve considerar.
DataCriacao	smalldatetime	Indica a data de criação
DataModificacao	smalldatetime	Indica a data de modificação (manual, usar GETDATE() em UPDATE)

Tabela VIII: Dicionário de dados para a tabela [Definicao\_Mapeamento]

O dicionário de dados para a tabela [Definicao\_Mapeamento\_Linha] do SQL é como segue:

Nome da coluna	Tipo de dado	Descrição
Id	uniqueidentifier	Chave primária da tabela de mapeamento com colunas SQL
Definicao_FK	uniqueidentifier	Chave estrangeira do cabeçalho do mapeamento
NomeCampo	nvarchar	Nome do campo SQL, da tabela SQL declarada no cabeçalho [Definicao_Mapeamento]
Ordem	int	Ordem para extração
Descricao	ntext	Descrição administrativa do mapeamento
TipoDado	nvarchar	Obrigatório preenchimento com um dos valores: "Boolean", ou "Data", ou "Integer", ou "Texto"
XPath_Pesquisa	nvarchar	Xpath de pesquisa, caso se trate de uma extração onde Tipologia = "Simples". Neste caso. [Definicao_Mapeamento].[XPath] estará vazio
ElementoEsperado	nvarchar	Se Tipologia = "Simples", estamos à procura de informação num determinado elemento HTML. Indicar aqui o elemento.
IndiceEsperado	int	Se Tipologia != "Simples", então estamos a extrair secções compostas por tabelas. Indicar o número da coluna a considerar, com índice a começar em 0.
Boolean_PalavrasChave	nvarchar	Se TipoDado = "Boolean", verificará se o índice da coluna contém uma determinada palavra.
DataCriacao	smalldatetime	Data de criação da linha de mapeamento
DataModificacao	smalldatetime	Data de modificação da linha de mapeamento (manual, usar GETDATE() em UPDATE)

Tabela IX: Dicionário de dados para a tabela [Definicao\_Mapeamento\_Linha]



A identificação de XPaths ideais para alimentação do motor de web scraping é um trabalho que requer muito conhecimento de HTML, da própria framework "XPath", e de conhecimentos em JavaScript caso necessário realizar pequenas alterações ao script de extração.

É impossível, num simples manual, ensinar todas as técnicas necessárias para a identificação com sucesso destes parâmetros. Recomendamos consulta de documentação sobre XPaths na Internet, bem como as APIs disponibilizadas pelo Puppeteer

#### Ligações relevantes para XPath:

- [https://www.w3schools.com/xml/xpath\\_intro.asp](https://www.w3schools.com/xml/xpath_intro.asp)
- [https://www.w3schools.com/xml/xpath\\_syntax.asp](https://www.w3schools.com/xml/xpath_syntax.asp)
- <https://developer.mozilla.org/en-US/docs/Web/XPath>
- <https://devhints.io/xpath>

#### Ligações relevantes para Puppeteer:

- <https://github.com/puppeteer/puppeteer>
- <https://devdocs.io/puppeteer/>

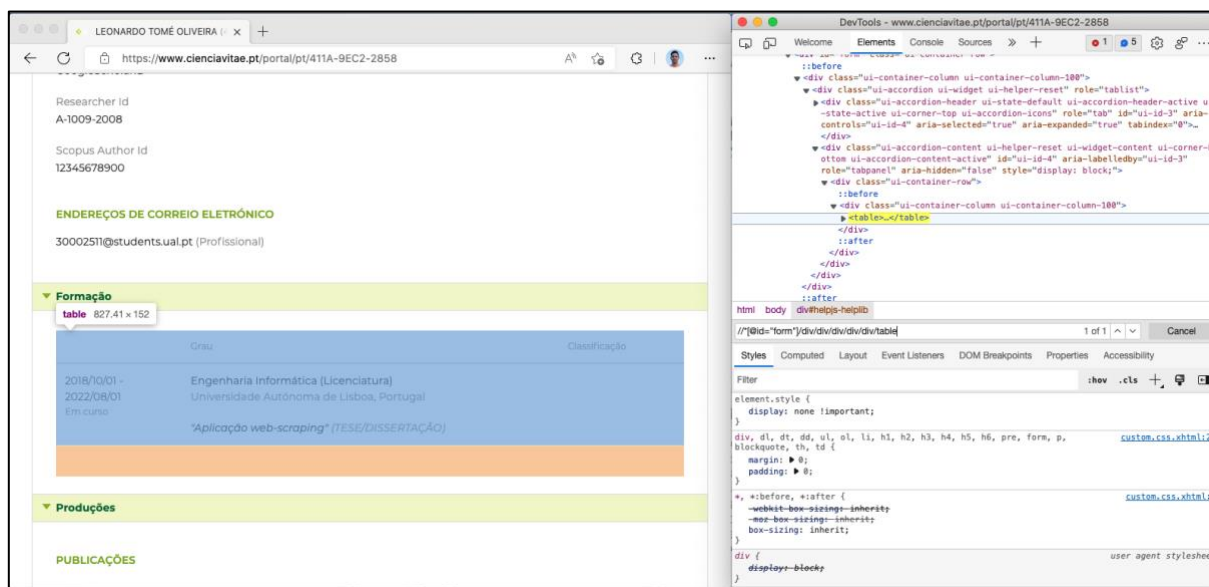


Figura IV: Exemplo de identificação XPath para a secção "Formação" do CV, por meio da query:

`"//*[@id="form"]/div/div/div/div/div/table"`