

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Unidade Curricular: Base de Dados 1

Relatório Relativo ao Trabalho Prático

Realizado por:

Miguel Batista-22976

Pedro Carreiras-23000

Gabriel Morujão - 23019

Gonçalo Alves - 23029

Viseu, 2023

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório relativo ao Trabalho Prático

Licenciatura em Engenharia Informática

Unidade Curricular de Base de Dados 1

Trabalho Prático

Ano Letivo 2022/23

Viseu, 2023

**ÍNDICE**

[1. Introdução 1](#_Toc136819792)

[2. DDL 2](#_Toc136819793)

[3. DML 4](#_Toc136819794)

[4. Consultas em SQL 5](#_Toc136819795)

[5. Objetos lógicos 7](#_Toc136819796)

[6. Aplicação Cliente/Servidor 11](#_Toc136819797)

[7. Conclusão 13](#_Toc136819798)

[8. Referências Bibliográficas 14](#_Toc136819799)

**Índice de Figuras**

[Figura 1 - *DROP TABLE* 2](#_Toc136819800)

[Figura 2 - *CREATE TABLE* 3](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819801)

[Figura 3 - Inserção de dados em algumas tabelas da base de dados 4](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819802)

[Figura 4 - Exemplo de Subconsulta 5](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819803)

[Figura 5 - OUTER JOIN 6](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819804)

[Figura 6 - *TRIGGER* 8](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819805)

[Figura 7 - Procedimento Armazenado 9](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819806)

[Figura 8 - *CURSOR* 10](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819807)

[Figura 9 - *WEB Form* 11](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819808)

[Figura 10 - Inserção de um utilizador (*Webform.aspx.cs*) 11](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819809)

[Figura 11 - Inserção de um utilizador (*BaseDeDados.cs*) 12](file:///C:\Users\pedro\Desktop\Relatorio_BD.docx#_Toc136819810)

# Introdução

O presente relatório tem como objetivo descrever os processos envolvidos no desenvolvimento dos scripts de criação e inserção de dados (primeira entrega), consultas em SQL, envolvendo junções, subconsultas, procedimentos armazenados, funções, triggers cursores (segunda entrega, bem como a criação de uma aplicação Cliente/Servidor (*ASP.net*) desenvolvida em *Visual Studio* (terceira entrega).

Este documento está organizado em 8 capítulos que foram divididos para uma melhor compreensão do leitor acerca do trabalho desenvolvido.

# DDL

*DDL* (*Data Definition Language*) é uma linguagem utilizada para definir e gerir a estrutura de objetos numa base de dados, neste caso, criação de tabelas. Os comandos DDL permitem criar, alterar ou excluir esses objetos da base de dados. Os comandos aos quais recorremos com mais frequência foram o comando *CREATE* e o comando *DROP que* têm como objetivo o seguinte:

* *CREATE*: O comando *CREATE* é utilizado para criar novos objetos na base de dados.
* *DROP*: O comando *DROP* é usado para excluir objetos da base de dados.

Na fase de criação deste ficheiro começamos por criar os comandos *DROP TABLE, de forma ordenada*, que graças à condição *IF EXISTS*, apenas irá apagar a tabela, caso esta exista. Estes comandos foram posteriormente comentados e procedemos à criação dos comandos de criação de tabelas (*CREATE TABLE*). Com o comando *CREATE TABLE* procedemos à criação das diversas tabelas, também de forma ordenada, que irão fazer parte da nossa base de dados bem como os diversos atributos bem como a criação das chaves primárias e estrageiras que cada tabela possuí.

O objetivo do *script DDL* no nosso projeto, é estabelecer a estrutura lógica e física da nossa base de dados, especificando os tipos de dados utilizados, as relações entre as identidades e as propriedades que irão garantir a qualidade e a consistência dos dados.

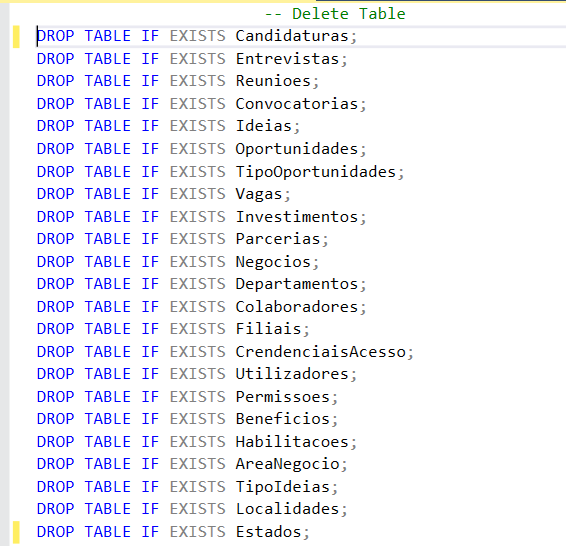


Figura 1 - *DROP TABLE*

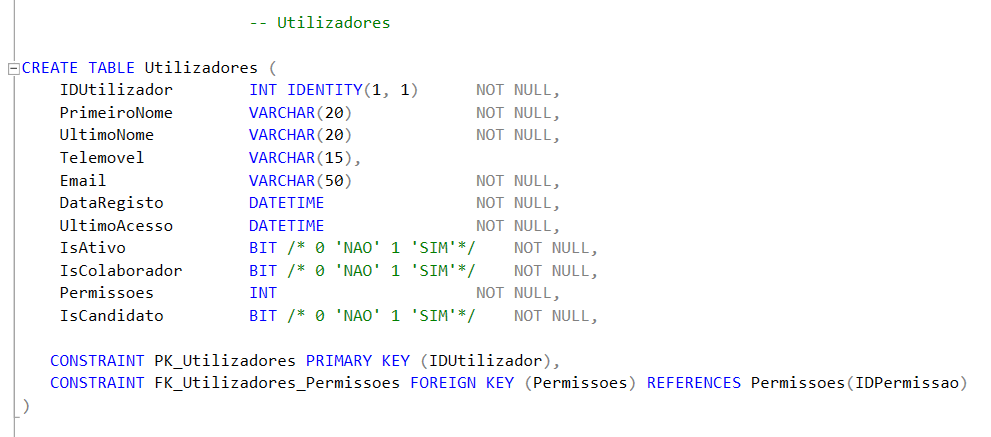


Figura 2 - *CREATE TABLE*

# DML

Na fase de criação do ficheiro do *script DML*, é nos permitido inserir dados na base de dados, com o objetivo de, mais tarde, podermos proceder à sua manipulação.

O objetivo principal deste *script* é permitir a realização de operações de inserção, atualização e exclusão de dados nas tabelas da nossa base de dados e, para tal recorremos, maioritariamente aos comandos INSERT INTO e VALUES que desempenham o seguinte papel:

* *INSERT INTO*: É utilizado para adicionar novos registos numa tabela da base de dados, sendo ainda possível especificar as colunas em que queremos inserir dados.
* *VALUES*: Esta clausula é utilizada em conjunto com o comando *INSERT INTO* para especificar os valores que serão inseridos nas colunas correspondentes.

A Inserção de dados, efetuada com os comandos INSERT INTO e VALUES, foram utilizados para adicionar novos registos nas diversas tabelas da nossa base de dados, com o objetivo de especificar os valores que devem ser inseridos em cada coluna.

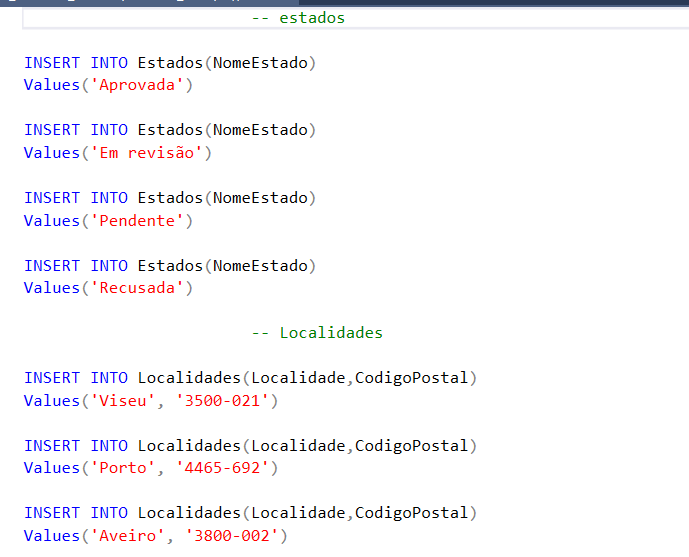


Figura 3 - Inserção de dados em algumas tabelas da base de dados

# Consultas em SQL

As consultas em SQL são instruções utilizadas para apresentar dados específicos de uma determinada base de dados relacional, permitindo obter resultados relevantes tendo em conta os dados armazenados e com base em determinados critérios e/ou condições. Para efetuar estas consultas utilizamos comandos como *SELECT, WHERE*, *GROUP BY*, bem como junções (internas e externas (*INNER JOIN* e *OUTER JOIN*)). Estes comandos desempenham os seguintes papeis:

* *SELECT*: É utilizado para obter dados de uma ou mais tabelas pertencentes a uma base de dados, permitindo especificar as colunas pretendidas.
* *WHERE*: É utilizada para filtrar os resultados da consulta com base numa condição específica, permitindo especificar critérios, aos quais, os atributos das tabelas devem atender para serem incluídos no resultado.
* *GROUP BY*: É utilizada para agrupar os resultados da consulta com base numa ou mais colunas, permitindo realizar agregações e cálculos nos dados agrupados.
* *INNER JOIN*: Combina registos de duas tabelas, sendo que apenas são retornados os que têm correspondência em ambas as tabelas.
* *OUTER JOIN*:  *R*etorna todos os registos de uma tabela e os registos correspondentes da tabela com a qual a primeira se relaciona, mesmo não havendo correspondência.

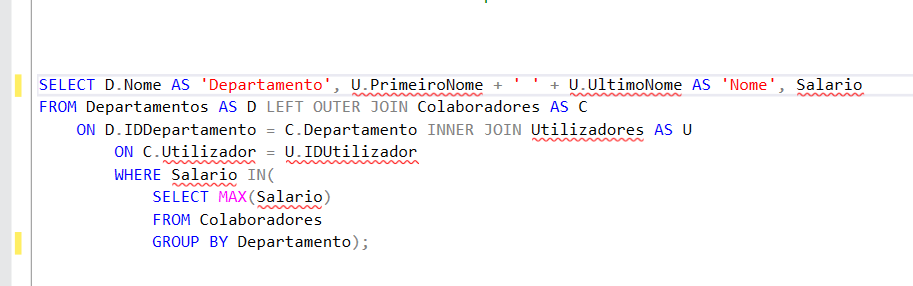
As consultas SQL são poderosas e flexíveis, permitindo obter dados de maneira eficiente, manipulando os dados de forma a obter informações relevantes a partir deles. Elas são essenciais para interagir com a base de dados e são amplamente utilizadas no desenvolvimento de aplicações, análise de dados, etc...

Figura 4 - Exemplo de Subconsulta

O objetivo desta subconsulta é apresentar os nomes dos departamentos, dos colaboradores e os seus respetivos salários, filtrando e apresentando apenas os colaboradores que possuem salário mais elevado de cada departamento.

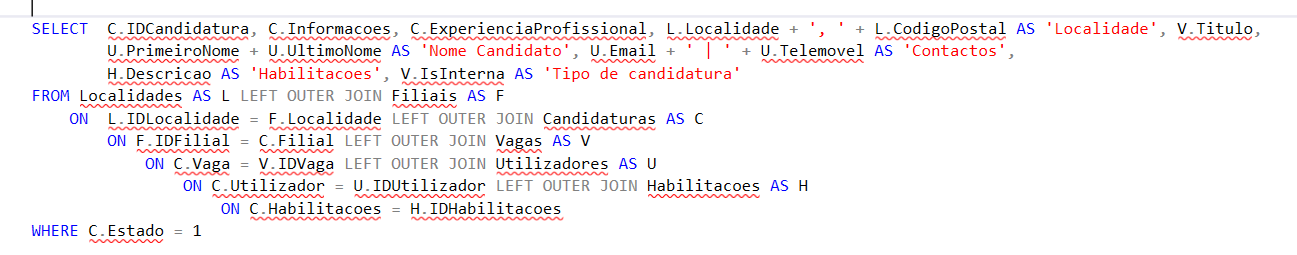


Figura 5 - OUTER JOIN

O objetivo da junção apresentada, é oferece o conjunto de todas as informações sobre as candidaturas ativas.

# Objetos lógicos

Os objetos lógicos são fundamentais para estrutura dos dados de uma base de dados. Cada objeto tem seu papel específico na organização, manipulação e recuperação dos dados armazenado. Os objetos lógicos que utilizamos no nosso projeto foram os seguintes:

*STORED PROCEDURES* (Procedimentos armazenados): Têm como objetivo realizar uma ou mais ações numa base de dados de forma organizada e reutilizável. Os principais objetivos dos procedimentos armazenados incluem:

* **Reutilização de código**: Permitem encapsular lógica de negócios complexa e reutilizá-la.
* **Melhoria do desempenho**: A lógica é pré-compilada e armazenada, o que pode resultar num melhor desempenho em comparação com a execução de consultas individuais.
* **Controlo de acesso**: Permite que os utilizadores executem apenas determinadas ações ou operações específicas.

*TRIGGERS*: São mecanismos que permitem a execução automática de ações como inserções, atualizações ou eliminações de registo de uma determinada tabela. Os principais objetivos dos *TRIGGERS* são:

* **Manutenção de integridade de dados**: Podem ser utilizados para garantir que as alterações dos dados atendam a certas regras de validação ou restrições definidas.
* **Atualização de informações**: Podem ser utilizados para atualizar informações noutras tabelas que estejam relacionadas com a tabela que sofreu a alteração.
* **Aplicação de regras de segurança**.
* **Registo de auditoria**: Podem ser utilizados para registar informações sobre as alterações realizadas numa tabela.

*CURSOR*: São estruturas que permitem percorrer e manipular os resultados de consultas de forma iterativa e sequencial, fornecendo um mecanismo para percorrer registos de uma tabela ou resultado de uma consulta linha a linha, permitindo assim a recuperação e manipulação de dados de forma mais eficaz. Os principais objetivos dos cursores incluem:

* **Processamento linha a linha**: Permitem executar lógica de negócios complexa e realizar operações detalhadas em cada registo, atualizando o valor em tempo real.
* **Acesso personalizado aos dados**: Fornecem controlo sobre os registos, permitindo a manipulação de dados de forma individual.
* **Manipulação de conjuntos de resultados**: Quando é necessário trabalhar com um conjunto de resultados menor ou específico em vez de um conjunto de resultados completo.

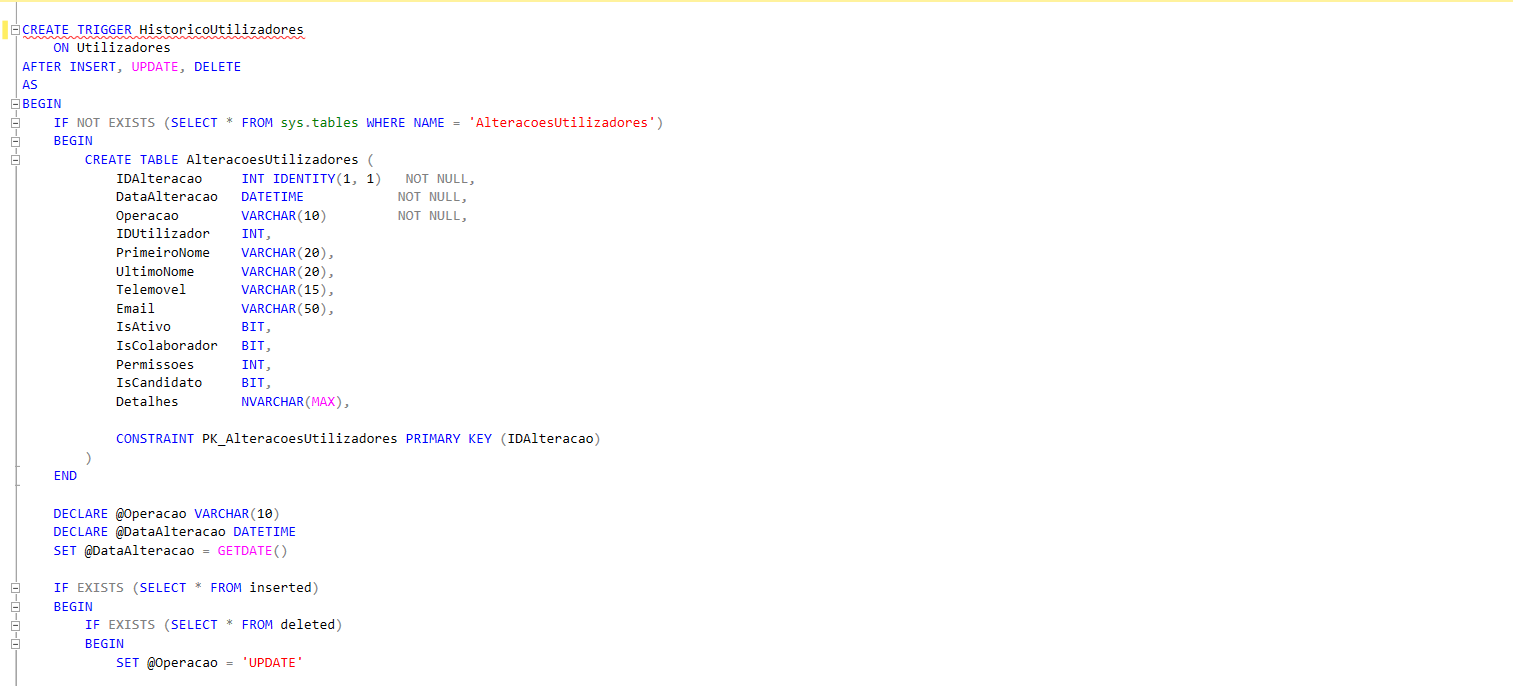
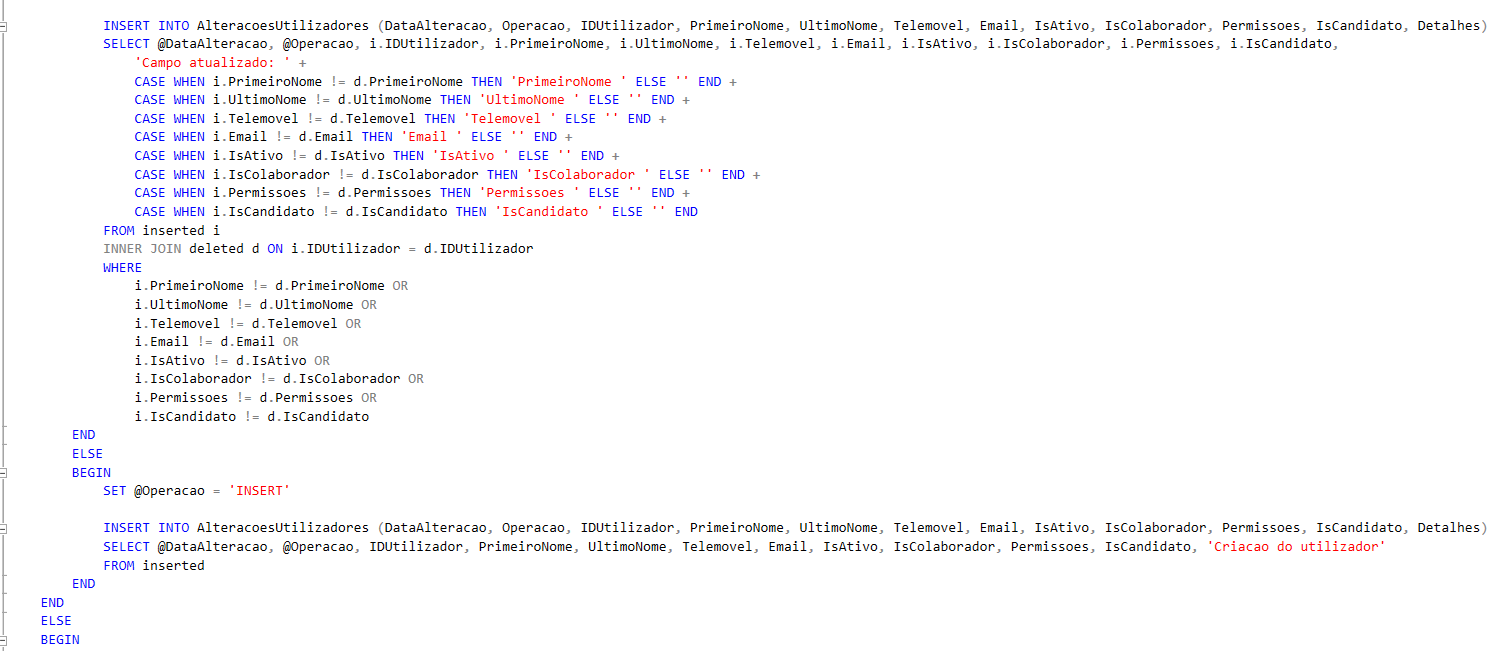
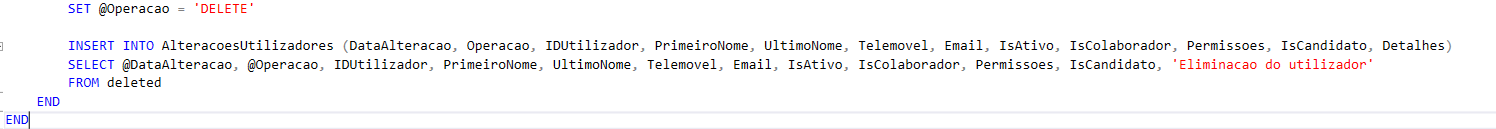


Figura 6 - *TRIGGER*

O objetivo deste *TRIGGER* é inserir todas as alterações, inserções e eliminações efetuadas na tabela, sendo que para tal é efetuado uma pesquisa por "alterações utilizadores" nas tabelas do sistema. No caso de a tabela não existir, o *TRIGGER* cria e insere lá as informações.

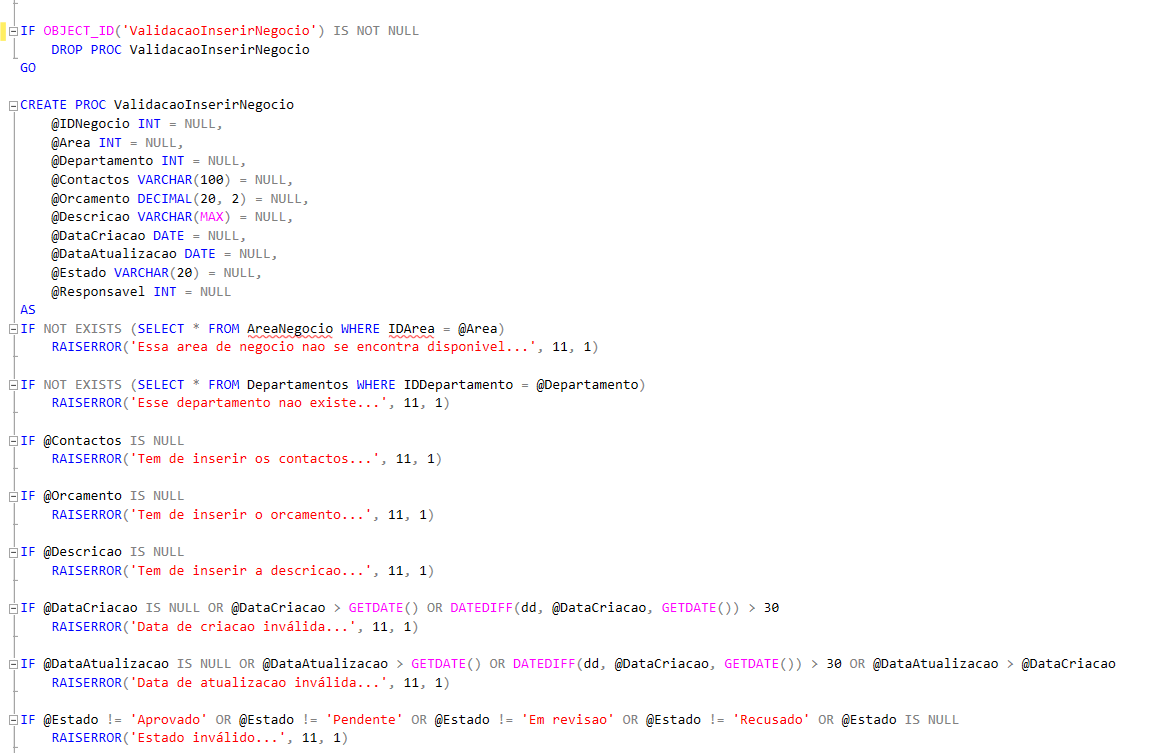
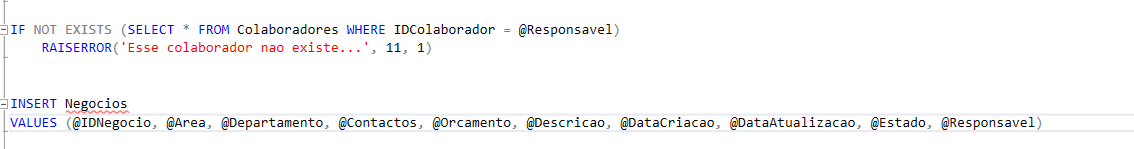


Figura 7 - Procedimento Armazenado

O objetivo deste Procedimento Armazenado é validar as inserções dos dados na tabela Negócios.

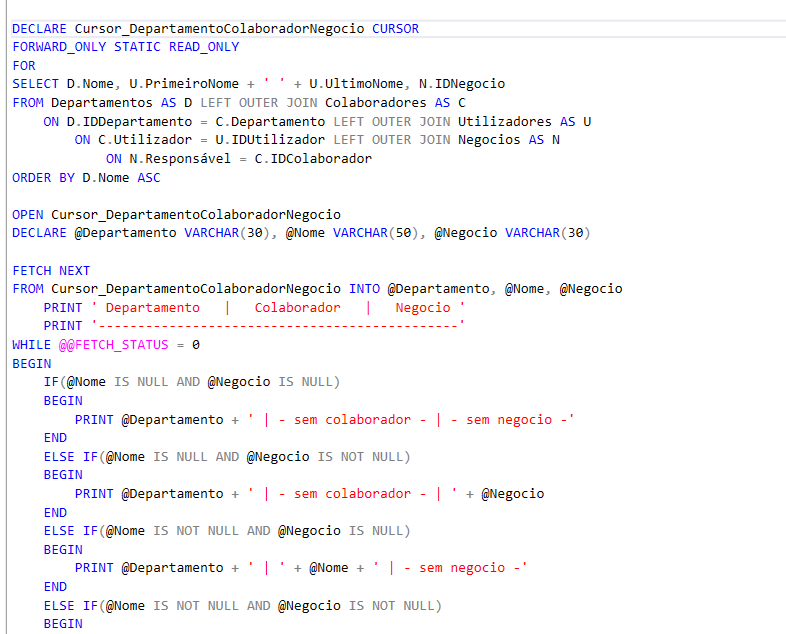
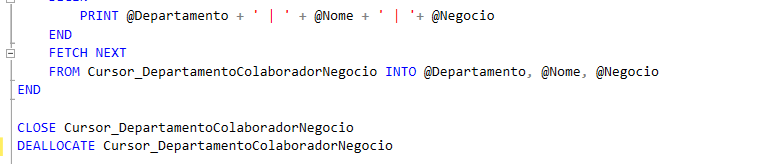


Figura 8 - *CURSOR*

O objetivo deste cursor é apresentar o colaborador, o departamento a que pertence e os negócios que este está envolvido

# Aplicação Cliente/Servidor

O objetivo da aplicação Cliente/Servidor é fazer com que seja possível a visualização do conteúdo da tabela “Utilizadores”, apresentando todos os utilizadores existentes, sendo possível efetuar a pesquisa de um utilizador em específico pelo seu ID. É também possível realizar a criação, a eliminação de um utilizador bem como editar as informações de um utilizador já existente.

Todos os dados são apresentados numa *gridview* com a devida ordem, sendo esta organizada pelo o ID de cada utilizador.

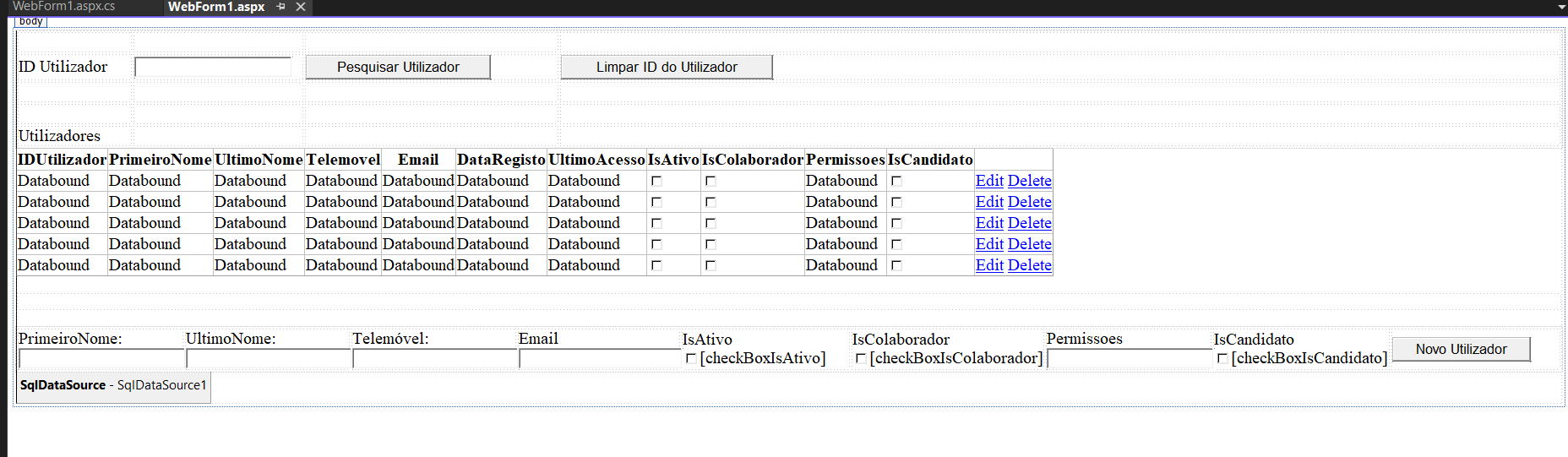


Figura 9 - *WEB Form*

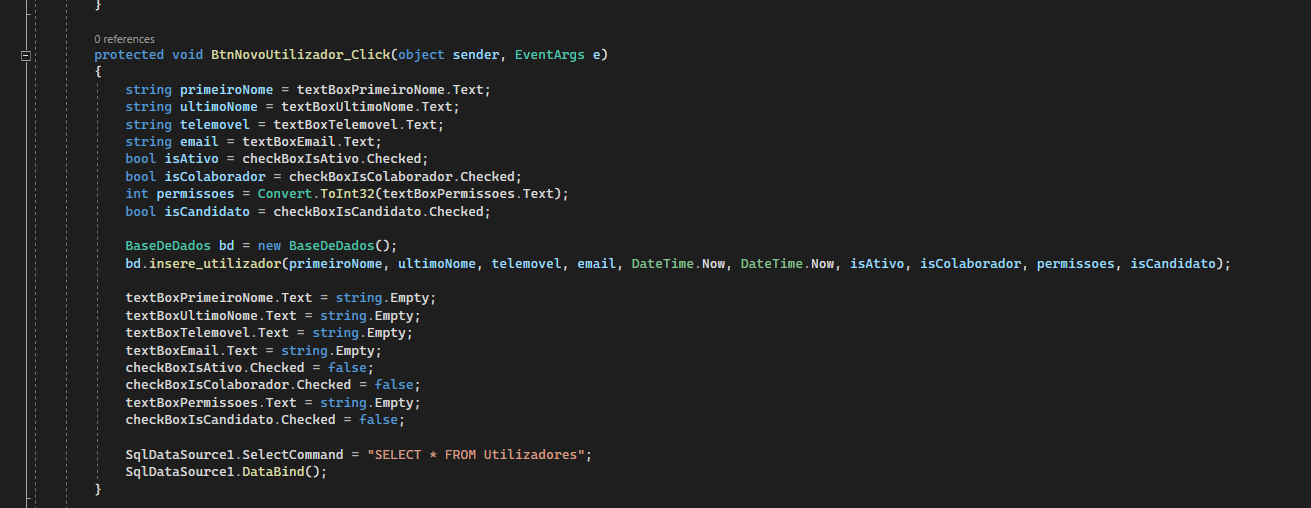


Figura 10 - Inserção de um utilizador (*Webform.aspx.cs*)



Figura 11 - Inserção de um utilizador (*BaseDeDados.cs*)

# Conclusão

Após a realização deste relatório foi possível aplicar os conhecimentos empregados em sala de aula relativos ao funcionamento do *SQL Server Management Studio* e do *ASP.NET Web aplication* no *Visual Studio*. Sendo a disciplina de Base de dados 1 focada na sensibilização aos alunos para a importância da informação e da integração das Tecnologias de Informação na estratégia das organizações, como um mecanismo de resolução de problemas e de criação de oportunidades. Sendo que tem como objetivo desenvolver nos alunos noções sobre os princípios, características e vantagens da gestão de dados recorrendo a sistemas de BD, proporcionando uma visão mais ampla dos aspetos mais importantes.

# Referências Bibliográficas

<https://stackoverflow.com/questions/tagged/sql>

<https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>

T-SQL Fundamentals, 4th edition (March 9, 2023)

<https://moodle.estgv.ipv.pt/course/view.php?id=5668>

<https://github.com/topics/sql>

Professional Microsoft SQL Server 2019 by Alastair Aitchison