

**Base de Dados**



**Sistema de Gestão de um Clube Desportivo**

**Docentes**

Joaquim Sousa Pinto ([jsp@ua.pt](mailto:jsp@ua.pt))

Carlos Costa ([carlos.costa@ua.pt](mailto:carlos.costa@ua.pt))

**Maria Rafaela Abrunhosa** 107658

**Marta Inácio** 107826

P8g9 - LEI

06 de junho de 2023

**Índice**

1. **Introdução** ………………………………………………………………………………………………………………..3
2. **Análise de Requisitos** …………………………………………………………………………………………….4
3. **Conceptualização da base de dados** ……………………………………………………………………6
   1. Diagrama Entidade Relacionamento…..…………………………………………………….6
   2. Esquema Modelo Relacional ……………………………………………………………………..7
   3. Diagrama da Base de Dados - Modelo Relacional SQL ………………………….8
4. **Construção da Base de Dados** ………………………………………………………………………………9
   1. Data Definition Language (DDL) ………………………………………………………………..9
   2. Data Manipulation Language (DML) …………………………………………………………9
5. **SQL Programming** …………………………………………………………………………………………………11
   1. Indexes ………………………………………………………………………………………………………..11
   2. Views ……………………………………………………………………………………………………………11
   3. UDF’s ……………………………………………………………………………………………………………12
   4. Stored Procedures ……………………………………………………………………………………..13
   5. Triggers ……………………………………………………………………………………………………….14
6. **Interface Gráfica** ……………………………………………………..……………………………………...…….16
7. **Segurança** ……………………………………………………………………………………………………………….19
8. **Tecnologias Utilizadas** ………………………………………………………………………………………….20
9. **Observações** …………………………………………………………………………………………………………..21
10. **Vídeo** …………………..…………………………………………………………………………………………………..22
11. **Ficheiros entregues em anexo** …………………………………………………………………………….23
12. **Conclusão** ……………………………………………………………………………………………………………….24
13. **Bibliografia** …………………………………………………………..…………………………………………………25

**Introdução**

Este projeto foi proposto no âmbito da Unidade Curricular de Base de Dados. Este projeto recai na gestão de um clube desportivo na perspetiva de um administrador do clube permitindo que o sistema seja capaz de gerir entidades desde instalações, a modalidades e a pessoas como atletas e funcionários associados ao clube.

Os objetivos do desenvolvimento deste trabalho são aplicar todos, ou quase todos, os conceitos lecionados nas aulas teóricas e práticas da melhor forma. De uma forma geral, pretendemos seguir o processo da criação de uma base de dados até à sua manipulação, começando no desenho do modelo da mesma e passando, posteriormente, à sua gestão e manipulação através de SQL.

**Análise de Requisitos**

O Sistema de Gestão de um Clube Desportivo procura modelar algumas funcionalidades tendo em conta os seguintes requisitos:

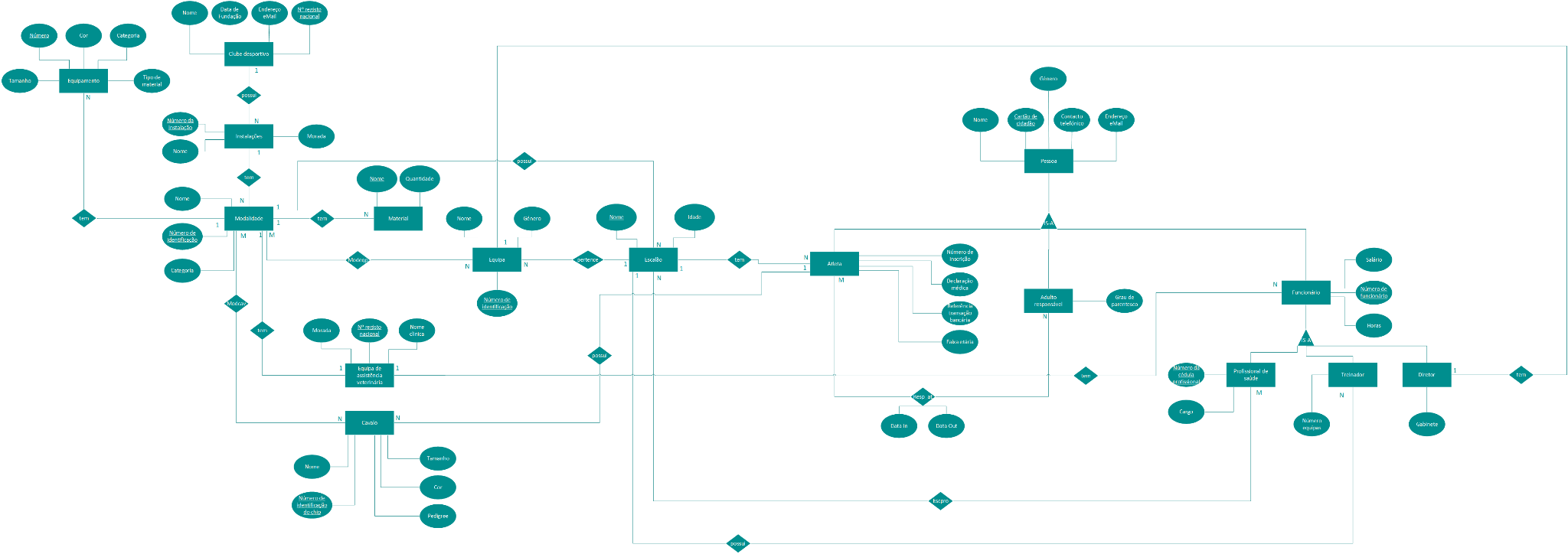
* Um clube desportivo treina várias modalidades incluindo ténis, basquetebol, hipismo, futebol e rugby.
* O clube desportivo possui diversas instalações caracterizadas por nome, número da instalação e morada.
* Cada instalação tem modalidades associadas.
* Uma modalidade pode ser considerada como sendo coletiva ou individual. Sendo coletiva e individual a modalidade é categorizada como mista. Esta admite como atributos um nome, um número de identificação e uma categoria.
* Cada modalidade coletiva tem várias equipas.
* Uma equipa é caracterizada por nome, número de identificação e género.
* Uma equipa pertence a um escalão e uma modalidade tem vários escalões.
* Um escalão é caracterizado por um nome e um intervalo de idade que o represente.
* Uma pessoa associada ao clube desportivo pode ser identificada com um nome, cartão de cidadão, contacto telefónico, endereço email e género.
* Um atleta, um funcionário e um adulto responsável são pessoas.
* Cada escalão tem vários atletas caracterizados por número de inscrição, declaração médica e idade.
* Cada atleta necessita de efetuar um pagamento mensal, registrando a referência da transação bancária.
* Os atletas com idade inferior a 18 anos possuem um adulto responsável caracterizado por grau de parentesco.
* Um clube possui diversos funcionários caracterizados por número de funcionário e salário.
* Cada equipa possui 1 diretor de equipa que é um funcionário e tem como atributo o número do seu gabinete.
* Cada equipa tem treinadores que são funcionários .
* Cada escalão possui profissionais de saúde a este associados, funcionando como equipa médica.
* Os profissionais de saúde são um funcionário e são identificados por um número da cédula profissional e cargo.
* A modalidade de hipismo possui uma equipa de assistência veterinária constituída por funcionários e caracterizada por morada, número de registo nacional e nome da clínica a que está associada, caracterizados por nome, número de identificação do chip, tamanho, cor e descendência/pedigree.
* Cada cavaleiro tem-lhe atribuído um cavalo.
* Cada equipa tem um conjunto de equipamentos de diferentes tamanhos e cores utilizados pelos atletas. Para além disso, os equipamentos são caracterizados por um categoria, tipo de material e quantidade.
* Cada modalidade tem um conjunto de material disponível para o treino das várias equipas, caracterizado pelo nome do material e quantidade.

**Conceptualização da base de dados**

Os esquemas a seguir apresentados sofreram algumas alterações após terem sido reconhecidos aspectos que podiam ser melhorados:

Sucintamente foram eliminadas algumas entidades e substituídas por um atributo na super-entidade associada, no caso da modalidade. Noutras entidades foi necessário acrescentar alguns atributos de forma a justificar a criação dessa entidade.

Diagrama Entidade Relacionamento



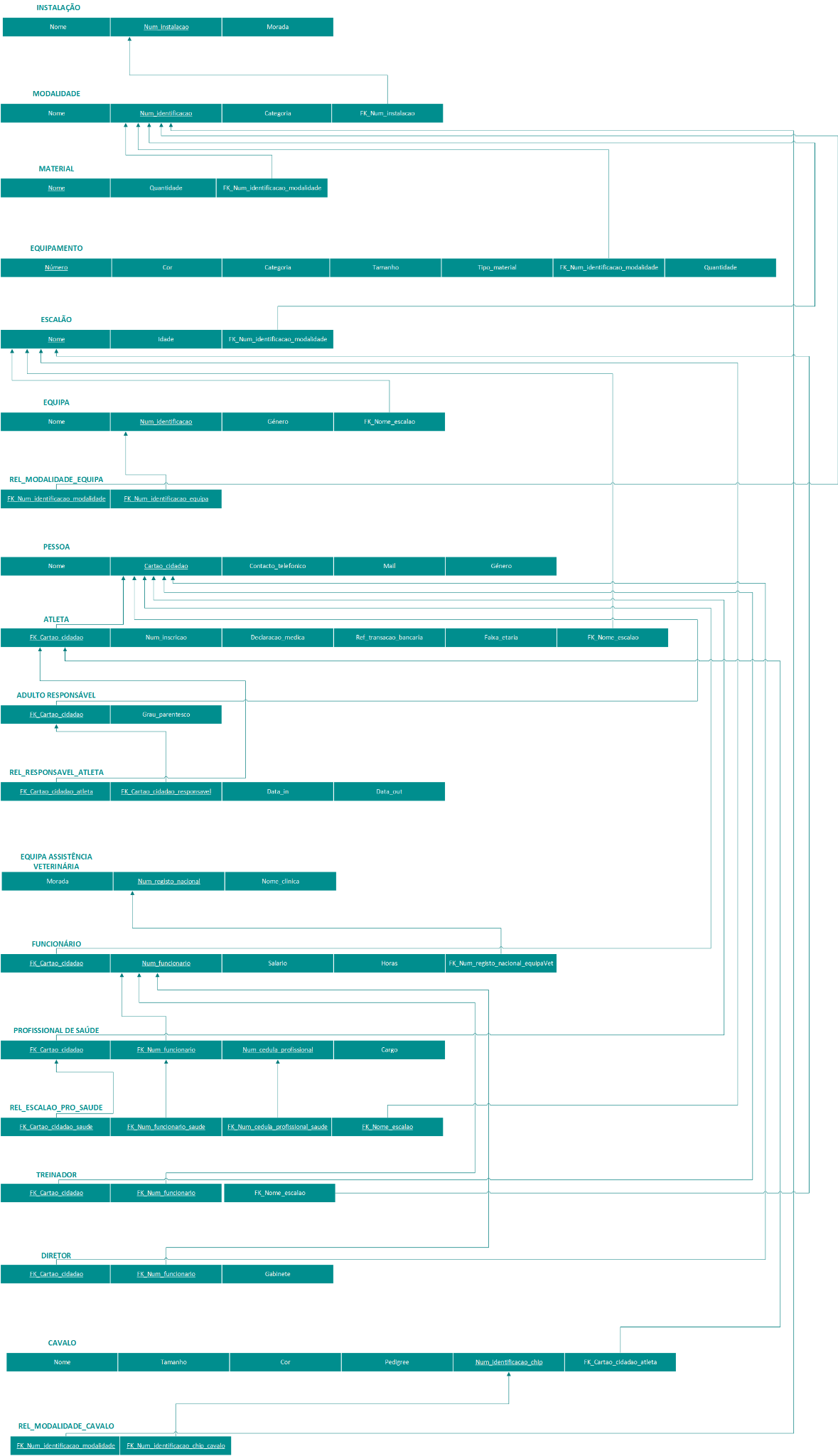
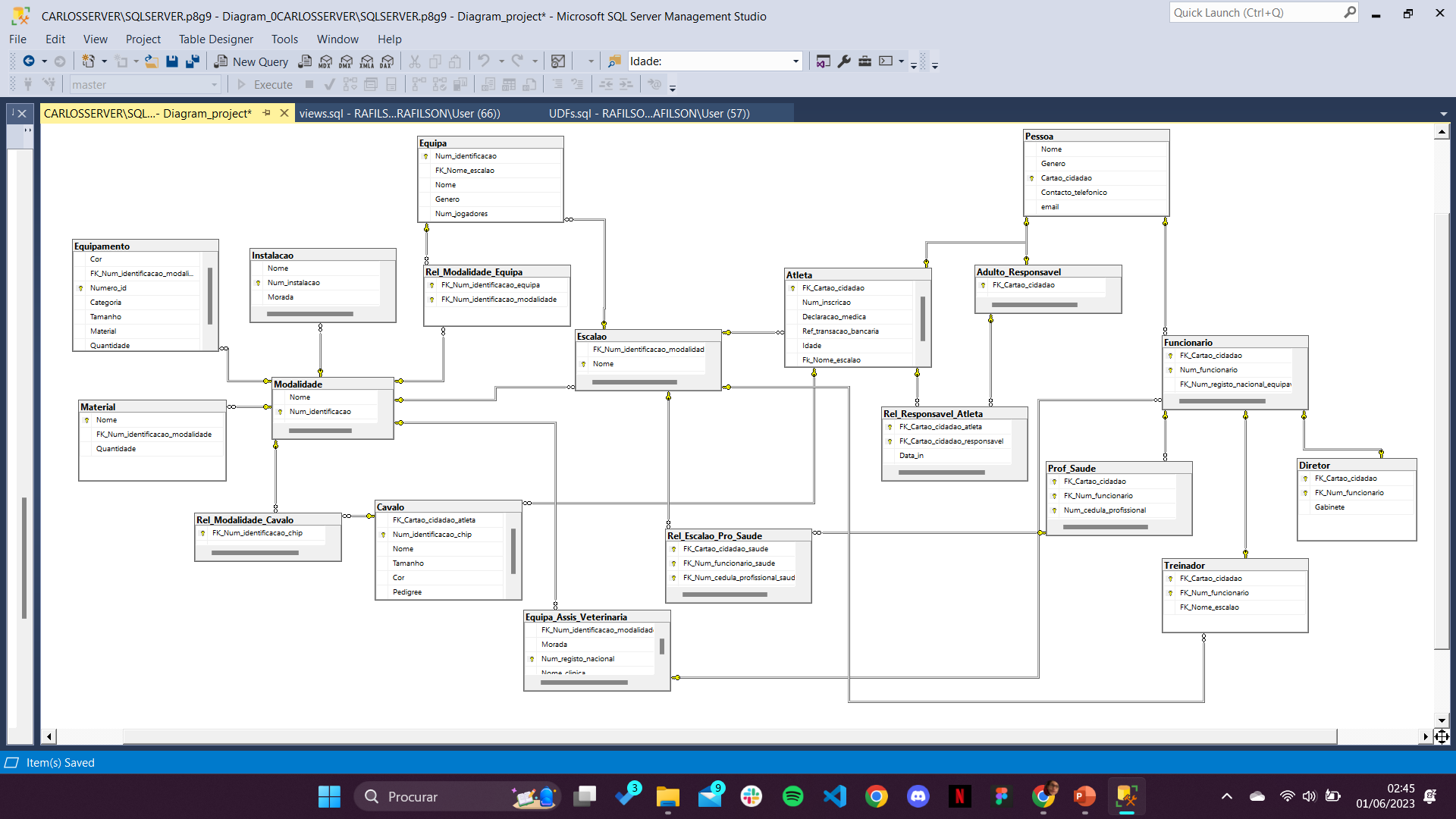
Esquema Modelo Relacional

Diagrama da Base de Dados - Modelo Relacional SQL

Este diagrama foi gerado após a criação das tabelas na base de dados.



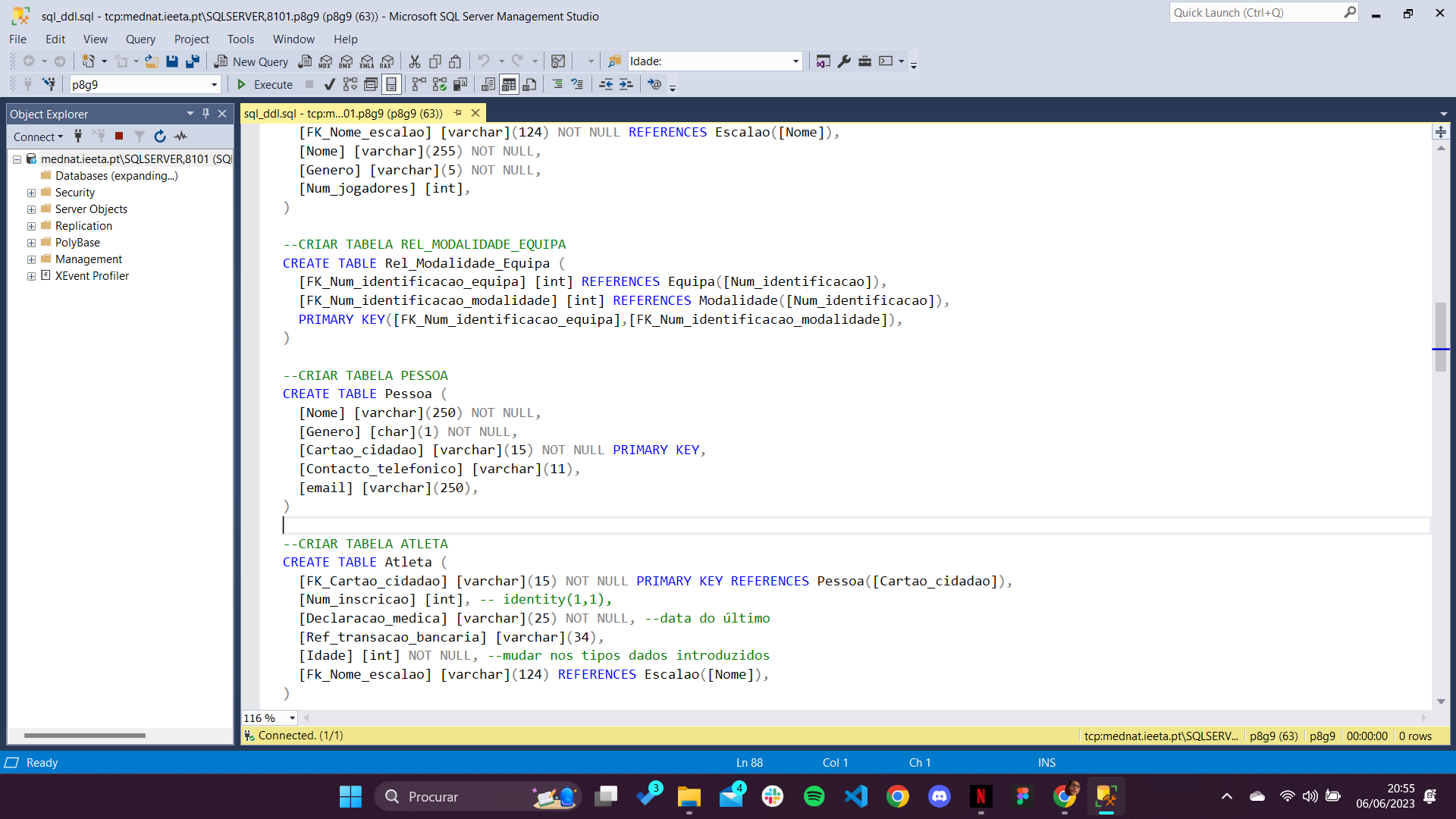
**Construção da Base de Dados**

Durante o processo de construção da base de dados, foi usada a linguagem SQL. Para criar as tabelas foi utilizada a Data Definition Language (DDL) e para inserir dados na tabela, editar e eliminar dados foi utilizada a Data Manipulation Language (DML).

Data Definition Language (DDL)

A DDL permite definir várias entidades e é utilizada para especificar a informação acerca de cada relação. Desta forma, a DDL é utilizada para criar, alterar e eliminar tabelas (CREATE, ALTER e DROP).

A DDL foi utilizada principalmente para criar as tabelas e definir os atributos das entidades bem como o seu tipo de dados e algumas restrições.





Data Manipulation Language (DML)

Para a inserção de dados na tabela foi utilizada a DML que permite a manipulação de dados numa base de dados. Esta manipulação consiste na inserção, eliminação e atualização de dados de uma tabela sendo, ainda, possível consultar os mesmos.

A DML é também, por isso, incorporada em SQL Programming como Stored Procedures.

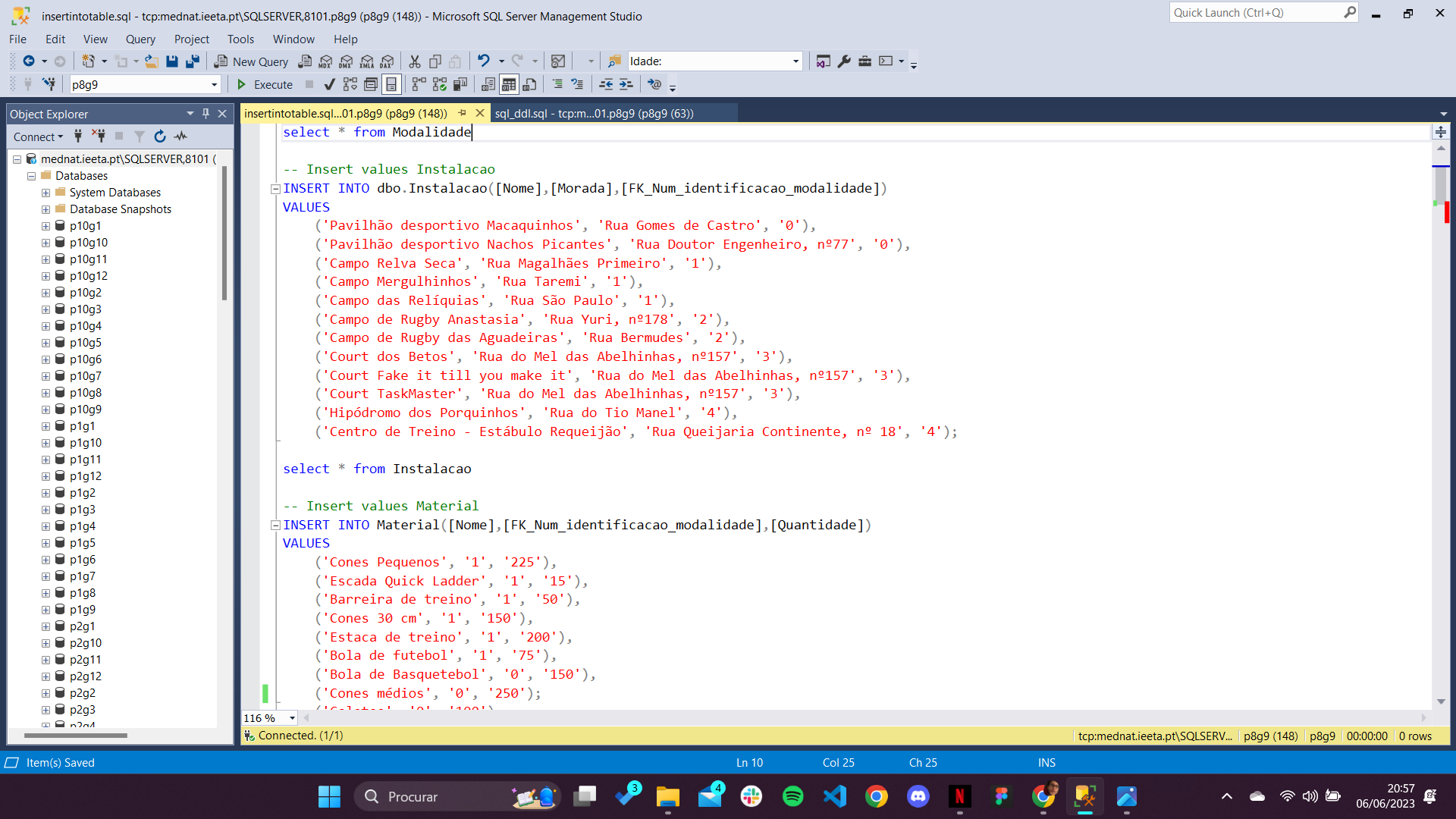


Figura 5 - Inserção de dados na tabela

“Instalação” e na tabela “Material”

**SQL Programming**

O SQL Programming é um termo que resume o que é uma linguagem de domínio específico desenvolvida não só para a gestão e manipulação de dados relacionais numa base de dados bem como, para o processamento de fluxo destes dados.

Serão apresentados desde Views a UDF’s, Stored Procedures e Triggers seguidos de alguns exemplos, apesar de seguirem os ficheiros completos em anexo.

Indexes

Os índices são estruturas de dados que permitem aceder aos dados de uma outra forma rápida.

Segundo esta definição, criamos dois índices, um para a Modalidade e um para a Pessoa ao se pesquisar pelo nome da mesma.

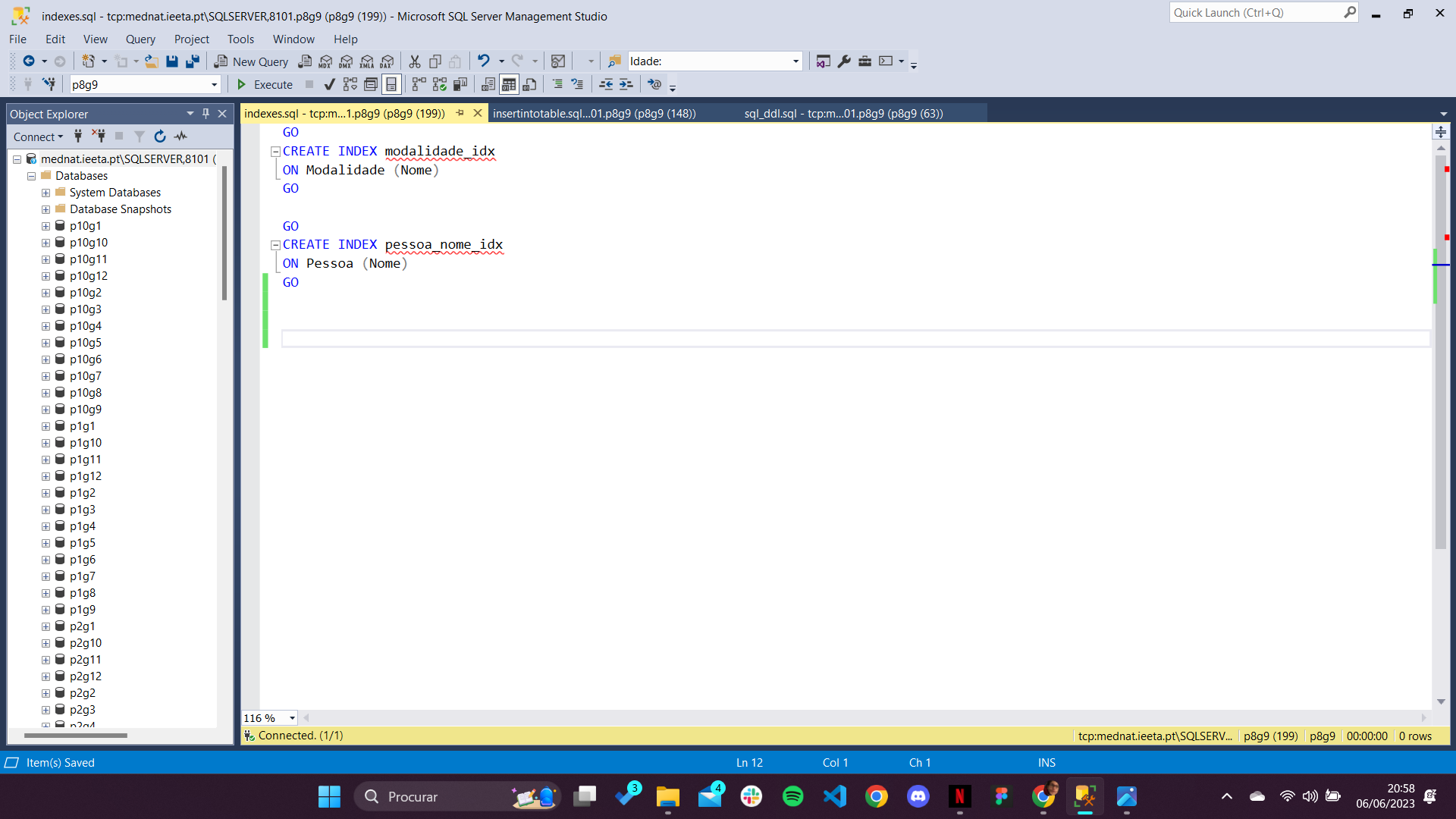


Figura 6 - Indexes implementados

nas tabelas “Modalidade” e “Pessoa”

Views

As Views podem ser utilizadas como fonte de dados como se fossem uma tabela num conjunto de operações SQL. De uma forma geral, permitem apresentar dados.

Criamos 4 Views, eliminando alguns atributos da apresentação da tabela e juntando diferentes tabelas. Estas Views foram aplicadas entre a Modalidade e Instalação, Material e Equipamento.

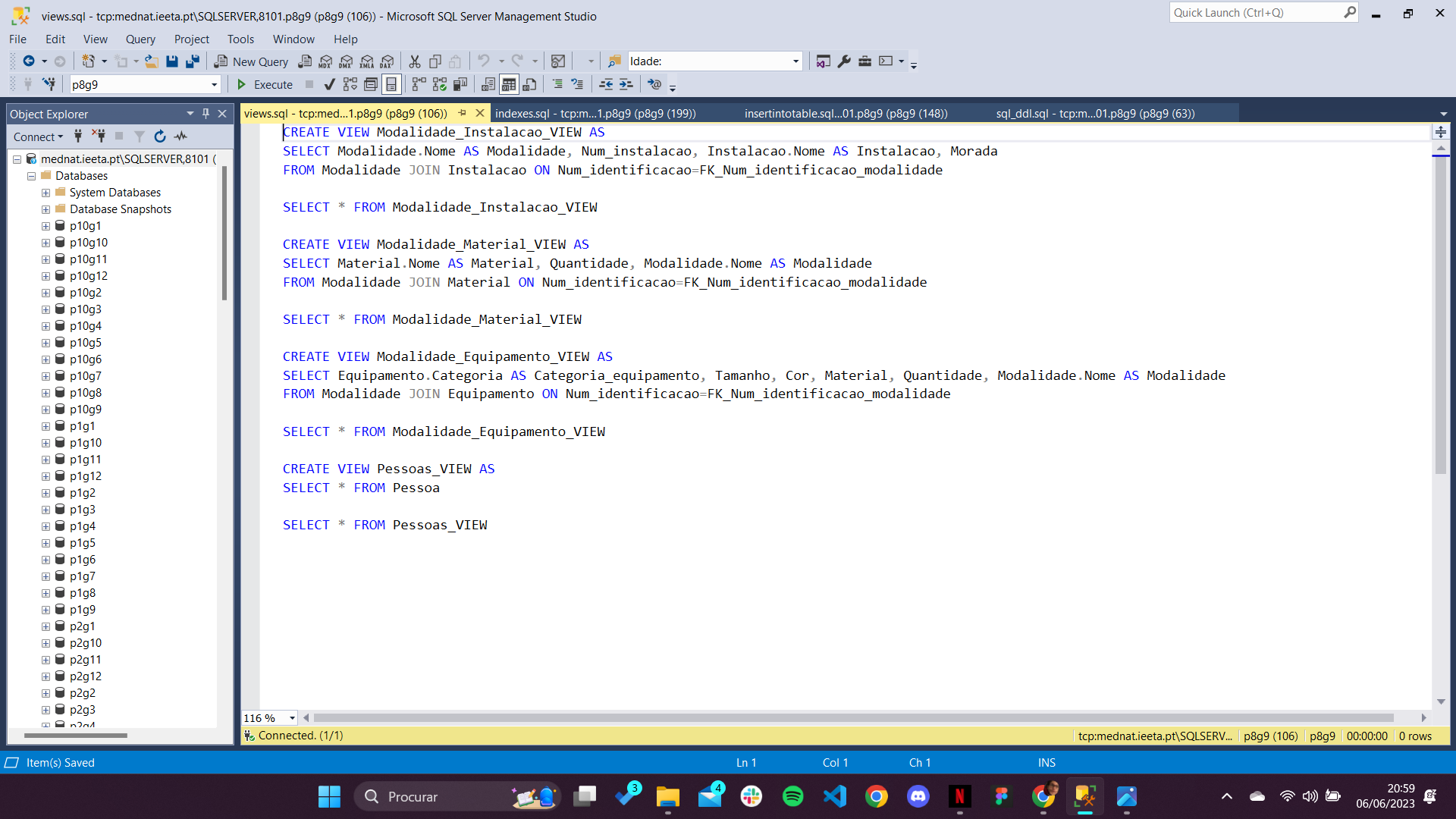


Figura 7 - Views implementadas nas

tabelas “Instalação”, “Material” e “Equipamento”

UDF’s

Os UDF’s são aplicados como funções contendo expressões complexas que pode ser utilizados como fonte de dados nas consultas e nas cláusulas WHERE e HAVING.

Foram criados vários UDF’s destacando-se os de gerar o número de inscrição de um atleta e o número de identificação do chip que serão incrementados por 1 sempre que um novo atelta ou cavalo, respetivamente, sejam adicionados.

Existe um UDF’s para ir buscar o número de jogadores de uma determinada equipa admitindo como argumento de entrada o escalão dessa equipa de forma a que aumento quando um atleta é adicionado ou que diminua quando um atelta é eliminado.

Por fim, existem dois UDF’s que retornam o género, um UDF retorna o género de uma equipa e admite como argumento de entrada o escalão desta e o segundo UDF retorna o género de um atleta admitindo como argumento de entrada o cartão de cidadão deste. Estes últimos UDF’s serão úteis para gerir a inscrição de um atleta numa determinada equipa, o uso dos mesmos será apresentado mais à frente.

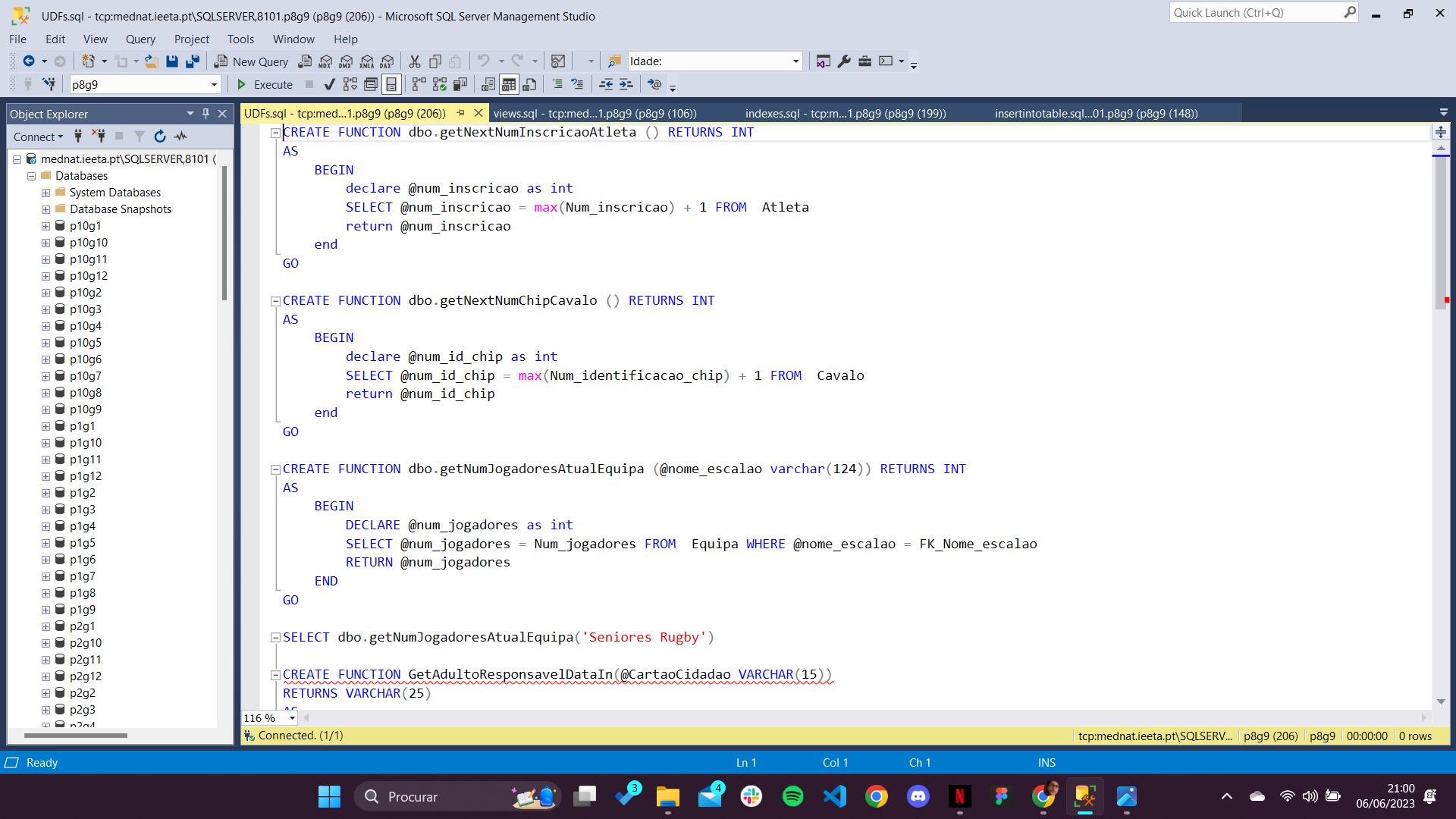


Figura 8 - UDF’s para calcular o número de inscrição e do chip do cavalo e UDF que retorna o nº de jogadores

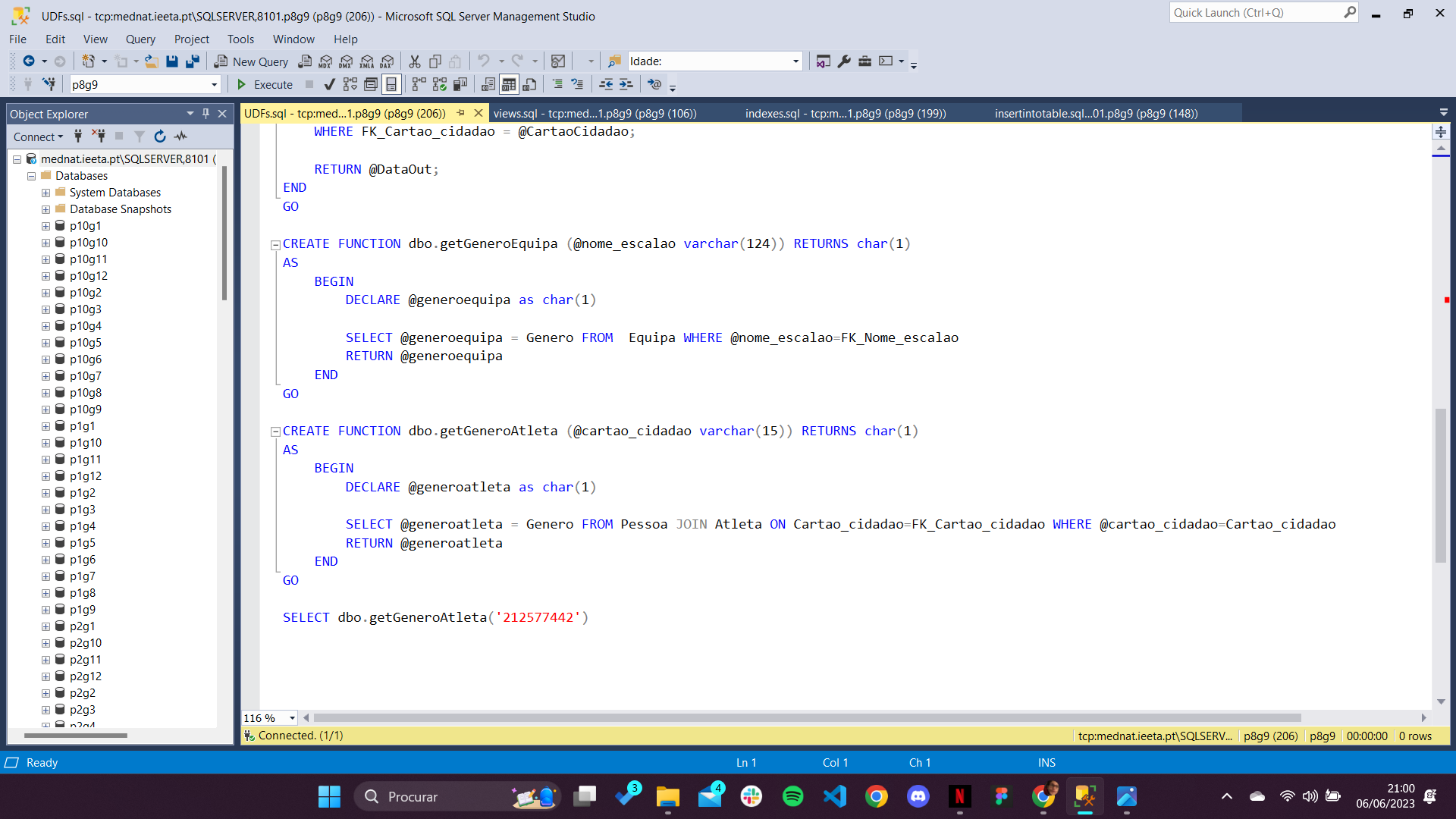


Figura 9 - UDF’s para retornar o género de uma Equipa e de um Atleta

Stored Procedures

As Stored Procedures destacam-se por serem um conjunto de instruções batch, compiladas pelo SQL-Server, identificadas por um nome.

Neste projeto foram usadas várias Stored Procedures podendo ser divididas em três grupos principais: de consulta, de criação e de eliminação.

É importante referir que todas as Stored Procedures que fazem alterações a várias tabelas, seja inserção , atualização ou eliminação de dados, incorporam transações para que não haja inconsistências entre tabelas dependentes umas das outras (são, também, aplicadas nos Triggers).

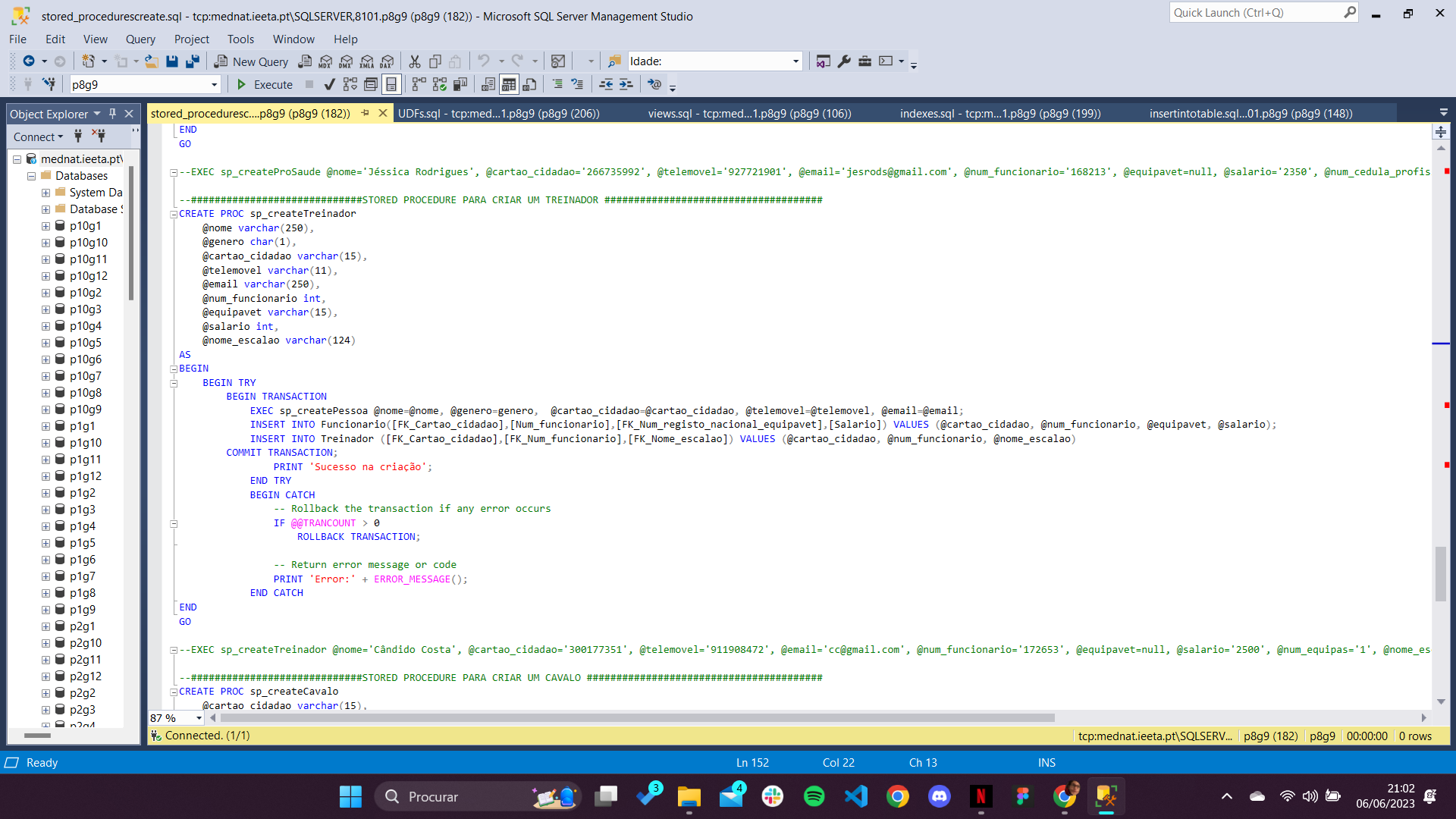


Figura 10 - Stored Procedure para adicionar um treinador

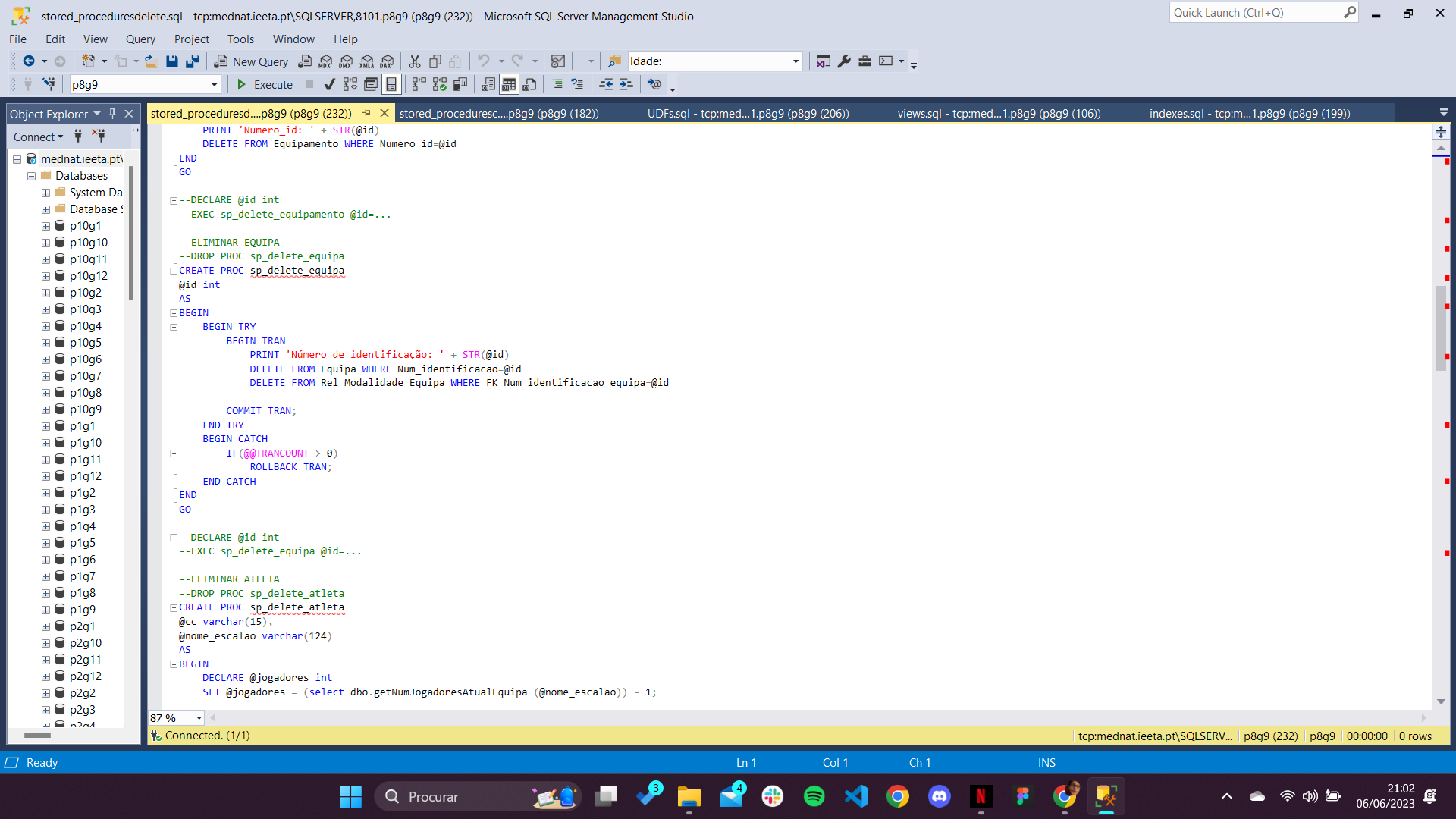


Figura 11 - Store Procedure para eliminar uma equipa

Triggers

Os Triggers são um tipo especial de Stored Procedures que são executados em determinados eventos associados à manipulação de dados. Estes são disparados um vez por cada operação de modificação de dados.

Os Triggers estão associados a uma tabela e atuam sobre esta.

Neste projeto, forma usados Triggers do tipo AFTER apesar de os Triggers do tipo INSTEAD OF serem preferíveis a nível de gastos de recursos. Assim sendo, estes Triggers eram ativados caso ocorresse algum INSERT na tabela associada ao Trigger.

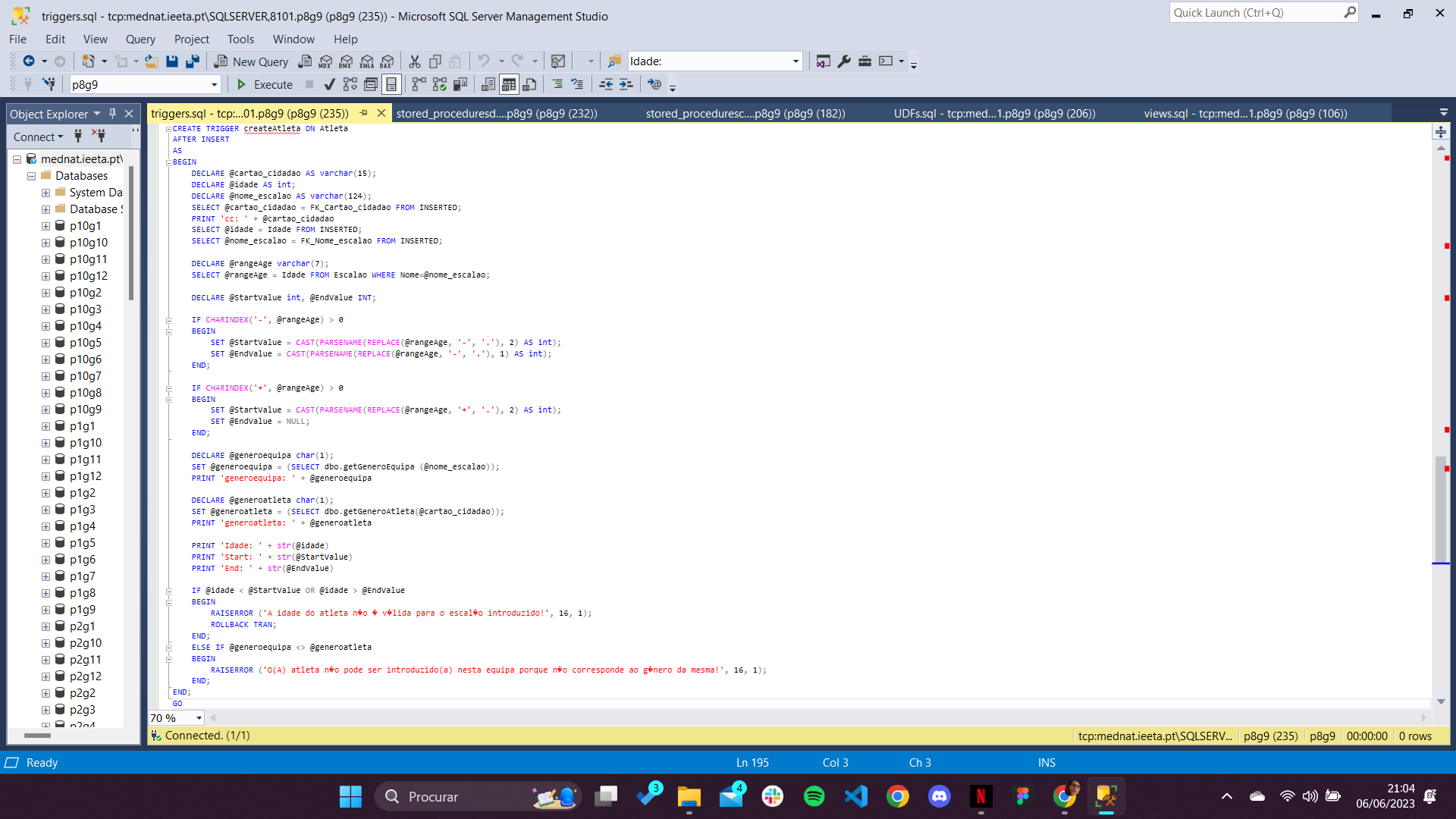


Figura 12 - Trigger que avalia a adição de um Atleta à base de dados

**Interface Gráfica**

A interface gráfica foi desenvolvida no Visual Studio em C#, recorrendo ao Windows Forms.

Para mostrar a informação presente na base de dados, utilizámos DataGridView na qual conseguimos apresentar a informação por colunas. Estas permitem a ordenação da informação em função da informação presente em cada uma delas, isto revela-se importante por exemplo na procura de determinados dados.

O layout é visualmente simples, é constituído por tabBar que permite aceder às diversas tabs presentes, correspondentes às diversas tabelas presentes na base de dados. Existe um botão para consultar a informação correspondente à tabela em questão. Depois existem botões para criar, eliminar ou editar dependendo da tabela e das ações possíveis na mesma.

Durante a inserção, eliminação e/ou edição de dados na interface, são verificados certos argumentos de entrada e caso não verifiquem certas condições é apresentado uma mensagem de erro devolvida por exemplo por triggers.

Para além disso, existe uma barra de pesquisa que devido à pouca experiência com a linguagem C#, não foi totalmente implementada.

Em seguida, está uma imagem exemplo da interface gráfica.

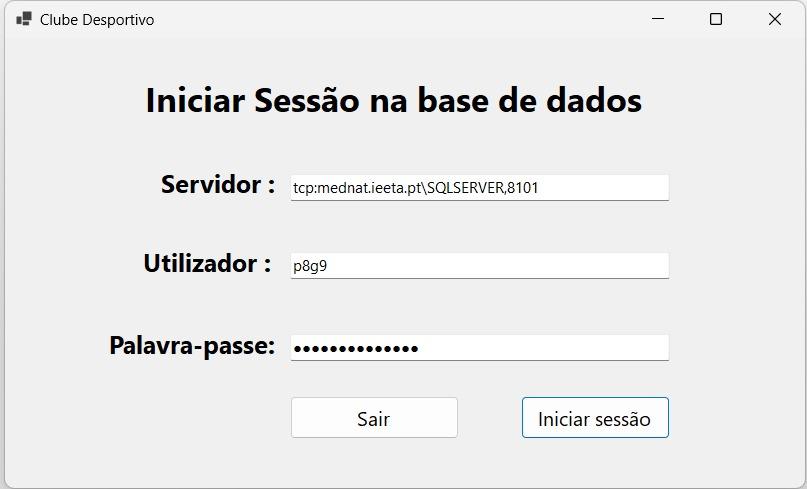


Figura 13 - Página Inicial

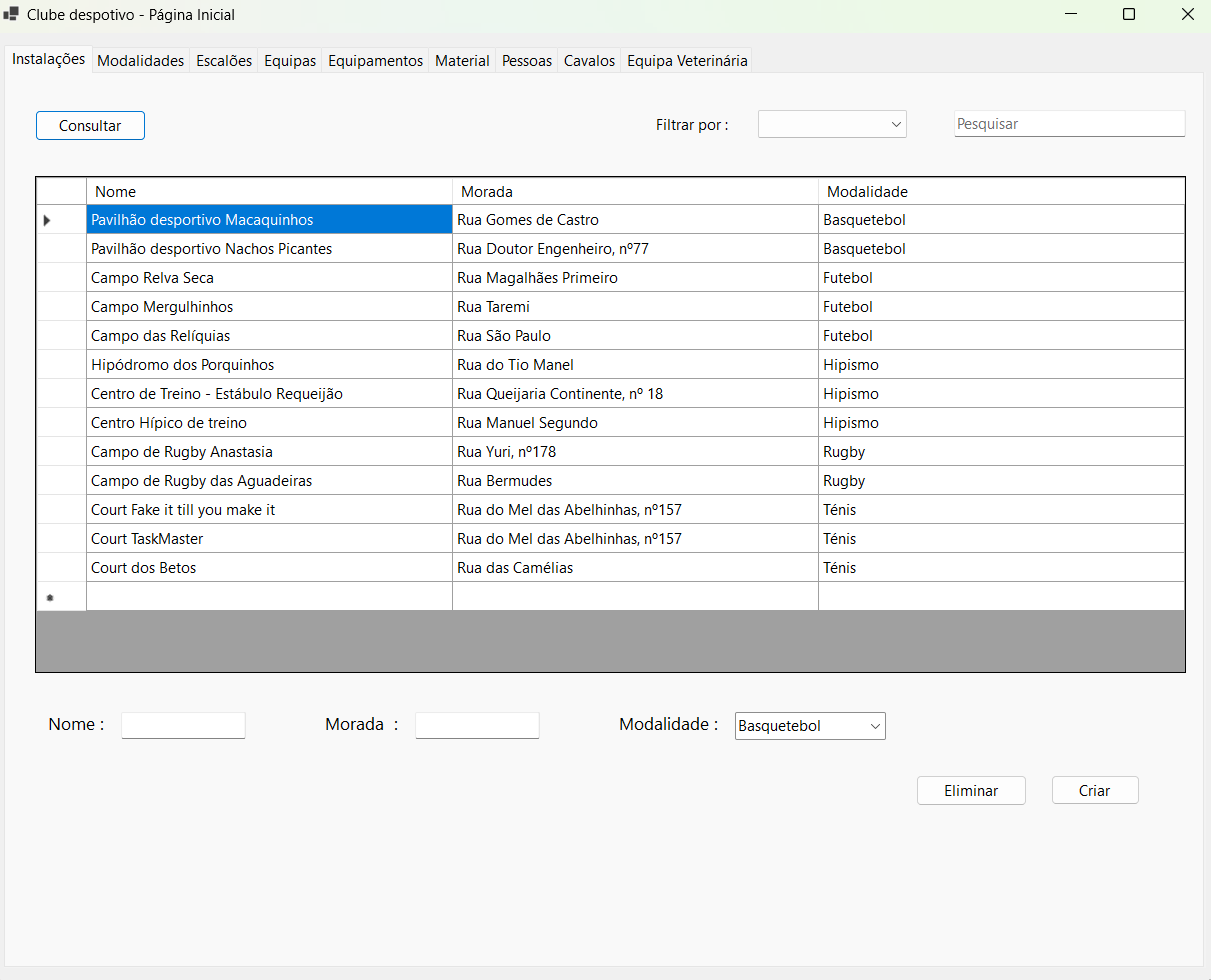


Figura 14 - Página das Instalações (form)

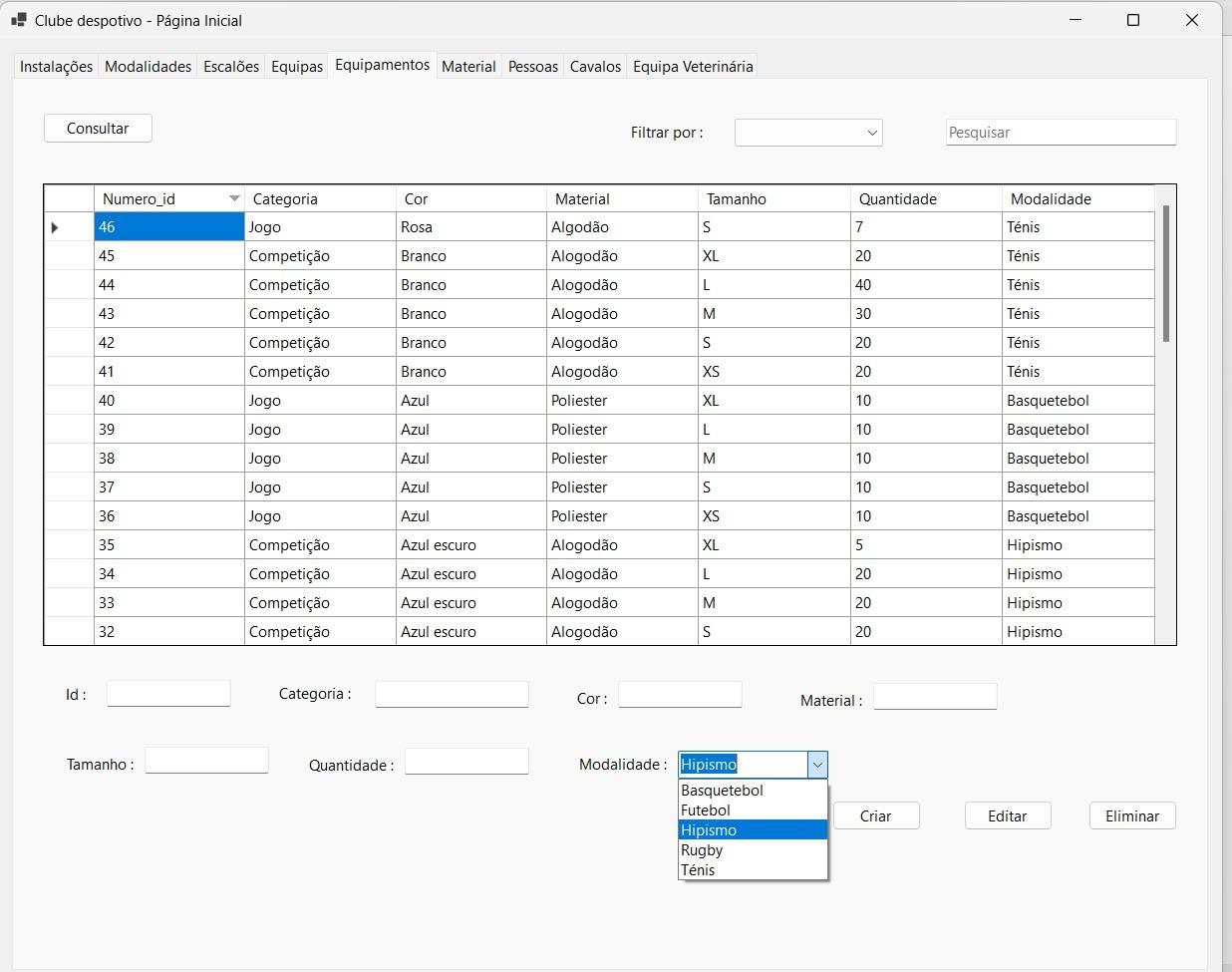


Figura 15 - Página dos Equipamentos (form)

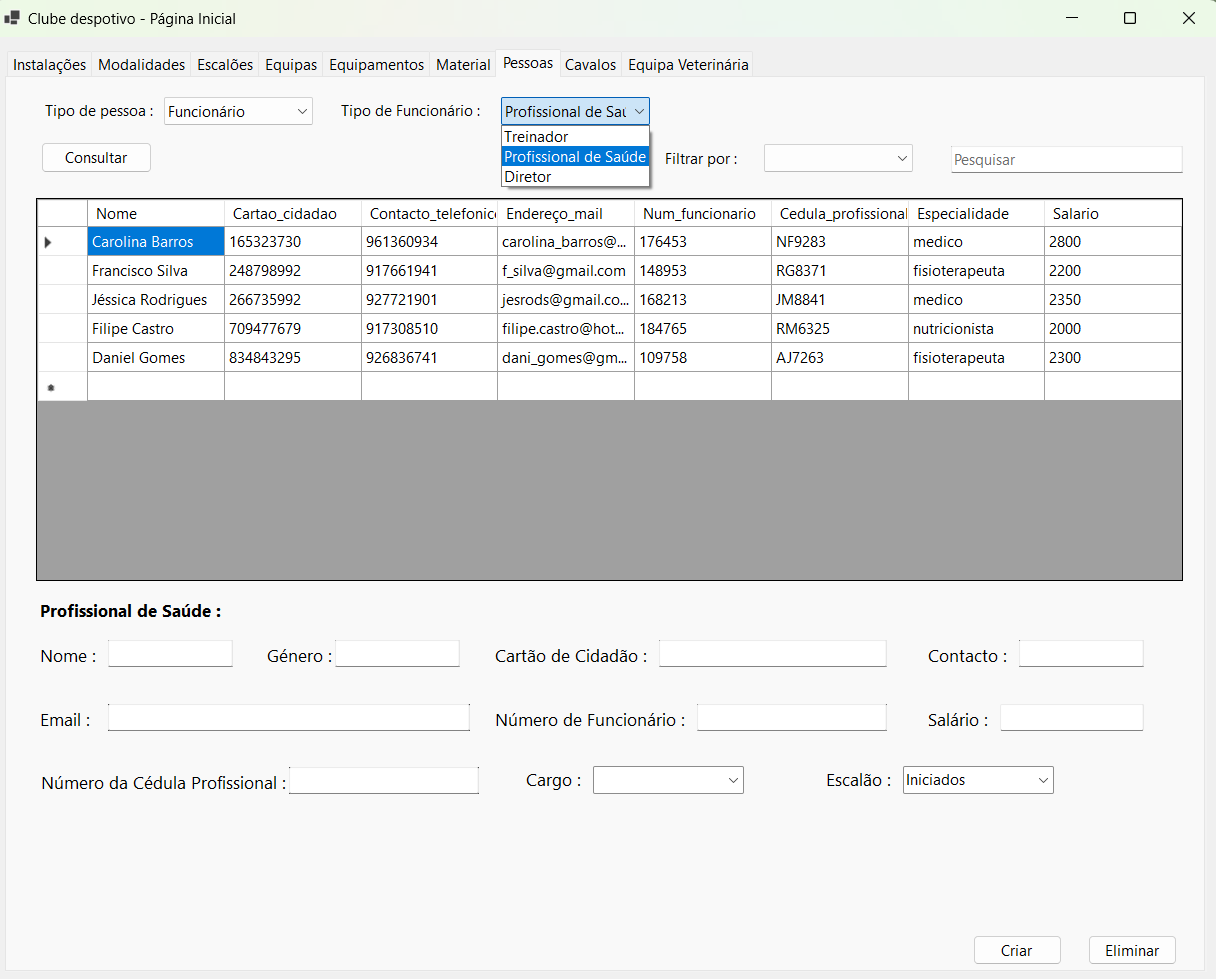
****

Figura 16 - Página com informação dos profissionais de saúde (form)

**Segurança**

Para minimizar os riscos a ataques de SQL Injection, a implementação da base de dados foi feita para ter em conta os dados inseridos, sendo validados através de Triggers apresentando, assim, quando algum argumento é inválido, mensagens de erro que expliquem qual é a informação errada e o porquê. Todo o processo baseou-se no uso de Stored Procedures em detrimento de SQL Dinâmico.

**Tecnologias Utilizadas**

As tecnologias utilizadas no decorrer da realização deste trabalho foram o Visio Professional para o desenho dos diagramas, DER e Modelo Relacional, o SQL Server Management para a criação das tabelas e a manipulação dos dados e o Visual Studio 2019 para o desenvolvimento da interface em C#, visualização da mesma e interligação entre a interface com a base de dados.

**Observações**

Após a primeira entrega do projeto e a reunião com o professor da aula prática, os diagramas DER e Modelo Relacional foram sujeitos a algumas mudanças para tornar o sistema e a base de dados menos confusos e mais eficientes. Assim, os diagramas apresentados na secção 3, Conceptualização da base de dados, já contém as alterações necessárias e encontram-se coerentes com o resultado final do projeto.

Uma vez que os diagramas são relativamente grandes e extensos estes serão enviados em anexo para uma melhor visualização dos mesmos.

A análise de requisitos também foi melhorada em termos de conter as entidades e os seus atributos um pouco mais descritos.

Quanto à manipulação de dados da tabela, nem todas as entidades admitem todas as ações de criar, atualizar e eliminar. Foi feita uma lista em qual das entidades é que fazia sentido ter cada uma destas ações.

A nível de SQL Programming, foram apresentados Indexes, Views, UDF’s, Stored Procedures e Triggers, no entanto não foram implementados Cursores.

Esta implementação não foi feita uma vez que não achamos relevante nem necessário tendo em conta o nosso sistema.

**Vídeo**

Foi gravado um vídeo demonstrativo das principais funcionalidades da interface gráfica que foram descritas e ainda mostrar algumas mensagens de erro ao se tentar adicionar algo que não é considerado possível graças aos Triggers.

O link que se segue contém o vídeo em questão: <https://youtu.be/qlw7YrTjcog>

**Ficheiros entregues em anexo**

1. der\_clube\_desportivo (diagrama entidade-relacionamento)
2. modelorelacional\_clube\_desportivo (modelo relacional)
3. diagrama\_basedados (diagrama da base de dados - modelo relacional SQL)
4. sql\_ddl.sql (criação das tabelas da base de dados)
5. insertintotable.sql (inserção de dados na tabela)
6. indexes.sql (criação dos índices)
7. views.sql (criação das views)
8. UFDs.sql (criação dos UDF’s)
9. stored\_proceduresconsult.sql (criação das stored procedures de consulta)
10. stored\_procedurescreate.sql (criação das stored procedures de criação)
11. stored\_proceduresdelete.sql (criação das stored procedures de eliminação)
12. triggers.sql (criação dos triggers)
13. test.sql (inserção de mais alguns dados para testar os triggers)
14. interface.cs (código c# da interface)
15. sistemagestao\_clubedesportivo.pptx (powerpoint utilizado na apresentação)
16. vídeo da interface gráfica

**Conclusão**

Chegando aqui, podemos concluir que os principais objetivos do desenvolvimento deste projeto foram concluídos da melhor forma tendo como resultado um Sistema de Gestão de um Clube Desportivo funcional.

A implementação da interface gráfica permitiu demonstrar o funcionamento das queries SQL e mostrar que a mesma faz o que foi proposto.

Consideramos que o projeto foi concluído com sucesso e que foi capaz de aplicar os conhecimentos adquiridos tanto nas aulas práticas como nas aulas teóricas.

Foi um projeto interessante que nos ensinou um pouco de gestão de tempo e planificação desde o planeamento do sistema da base de dados até ao seu desenvolvimento total. Permitiu-nos conhecer um pouco mais sobre SQL Programming e até mesmo programação em C# uma vez que foi utilizado para a implementação da interface, promovendo, ainda, autonomia na aprendizagem.

Concluindo, foi um projeto que nos conseguiu incentivar ao longo do seu progresso.

**Bibliografia**

Slides da Unidade Curricular disponibilizados na página da disciplina do eLearning@UA.