

**ResearchBase**

**Projeto Final de Base de Dados 2021/22**

Prof. Carlos Costa

**Artur Afonso Ferra Correia (102477)**

**Daniel Luís Fernandes Carvalho (77036)**

Licenciatura em Engenharia Informática

Turma P8, Grupo 1

Índice

[1. Introdução 3](#_Toc106990061)

[2. Análise de Requisitos 4](#_Toc106990062)

[3. Desenho Conceptual da Base de Dados 6](#_Toc106990063)

[4. SQL-DDL e SQL-DML 9](#_Toc106990064)

[5. Camada de Interação com a Base de Dados 10](#_Toc106990065)

[6. Interfaces e Funcionalidades Implementadas 13](#_Toc106990066)

[6.1. Views 14](#_Toc106990067)

[6.2. Secção “Estudos” 15](#_Toc106990068)

[6.3. Secções Participantes, Investigadores e Entidades Patronais 18](#_Toc106990069)

[7. Conclusão 20](#_Toc106990070)

# Introdução

A construção da base de dados proposta para este projeto está assente na idealização de uma plataforma, a ResearchBase, que reúne estudos científicos abertos à participação do público em geral. Desta maneira, o objetivo é aproximar os grupos de investigação que necessitem de participantes para os seus estudos, de pessoas interessadas em realizar os mesmos, em troca de ume remuneração.

O desenvolvimento da interface que contacta com a base de dados foi feita no ponto de vista de um administrador do sistema, permitindo a gestão dos participantes e investigadores inscritos na plataforma, bem como a gestão dos vários estudos científicos presentes nesta, garantindo que as operações principais a si associadas são feitas em segurança e salvaguardadas na base de dados.

O desenvolvimento efetuado teve como objetivo a aplicação prática dos conhecimentos adquiridos ao longo do semestre na unidade curricular em todo o processo, desde a concetualização da base de dados até aos passos finais de implementação de interfaces com uma camada de interação com a base de dados.

A organização do conteúdo enviado na Entrega Final em diretórios foi a seguinte:

* **APF-E v.Final** – Pasta com o conteúdo apresentado na APF-E.
* **Diagramas** – Contém os vários diagramas para esquematização da base de dados desenvolvidos ao longo do semestre.
* **Docs** – Contém a Apresentação e o Relatório Final.
* **Forms** – Contém todo o código utilizado para o desenvolvimento da interface, com ficheiro principal do projeto a ser o ResearchBase.sln.
* **Scripts** – Contém todos os scripts SQL utilizado no dsenvolvimento do projeto.
* **Demo** – Contém o vídeo utilizado para demonstração da plataforma, que foi apresentado durante a Apresentação Final.

# Análise de Requisitos

Para inicializar o levantamento de requisitos, decidimos restringir a nossa base de dados a estudos de dois tipos principais:

* Ensaios Clínicos de farmacocinética de fase I;
* Ensaios de Investigação de áreas científicas mais generalizadas, realizados quer por institutos de investigação académicos, quer por departamentos de investigação empresariais (que podem ser desde questionários, a simulações ou *user testing* de produtos)

A restrição dos ensaios clínicos a ensaios de fase I deve-se ao facto de estes serem os únicos completamente abertos a pessoas da população geral, com participantes saudáveis, ao contrário dos ensaios de fases mais avançadas, que dessa forma saem fora do âmbito da plataforma.

Após esta decisão, o levantamento de requisitos foi inicializado com uma pesquisa sobre as necessidades e restrições que este tipo de atividades implica, tendo-se definido os seguintes requisitos relacionados com os estudos desenvolvidos:

**Tabela 1 –** Requisitos relacionados com os tipos de Estudos suportados pela plataforma.

|  |  |
| --- | --- |
| **Tipo de Estudo** | **Requisitos** |
| Ambos os tipos | * A cada estudo deve estar associado um Investigador Principal, que faz parte de um instituto de investigação ou empresa * A seleção de participantes para os estudos é feita através de critérios de seleção, relacionados com os dados pessoais destes (género, faixa etária, entre outros) * Estudos têm 3 fases possíveis: Em Recrutamento, Ativo e Concluído |
| Ensaios Clínicos de Fase I | * Os ensaios de farmacocinética suportados são de 2 tipos: bioequivalência ou interação medicamentosa * Os fármacos utilizados devem ser aprovados para teste e referenciados através de um código único de acordo com o standard GTIN-14 aplicado pela GS-1 Portugal. As farmacêuticas produtoras dos fármacos devem também ser identificados por um código de referência único * Participantes que queiram inscrever-se neste tipo de ensaios devem ter análises clinicas gerais em dia para provar que são saudáveis, sendo estas referenciadas na base de dados por um código * Ensaios clínicos podem ter várias tomas, consultas de follow-up e um protocolo associado, referenciado na base de dados por um código * Ensaios devem ser previamente aprovados por parte da INFARMED e da CEIC, sendo o comprovativo de aprovação guardado na base de dados através de um código * Um participante está limitado a realizar o máximo de 2 ensaios clínicos por ano, com uma remuneração máxima total de 1200€ |
| Estudos de Investigação Gerais | * Estudos podem ser de sessão única, ou longitudinais * Vários modelos de procedimento suportados pela plataforma: realização de questionários, de experiências físicas, *testing* de produtos, simulações, estudos de grupo, entre outros |

Foram ainda consideradas as seguintes funcionalidades requeridas pelos diferentes utilizadores da plataforma, nomeadamente os investigadores responsáveis pelos estudos e os participantes destes:

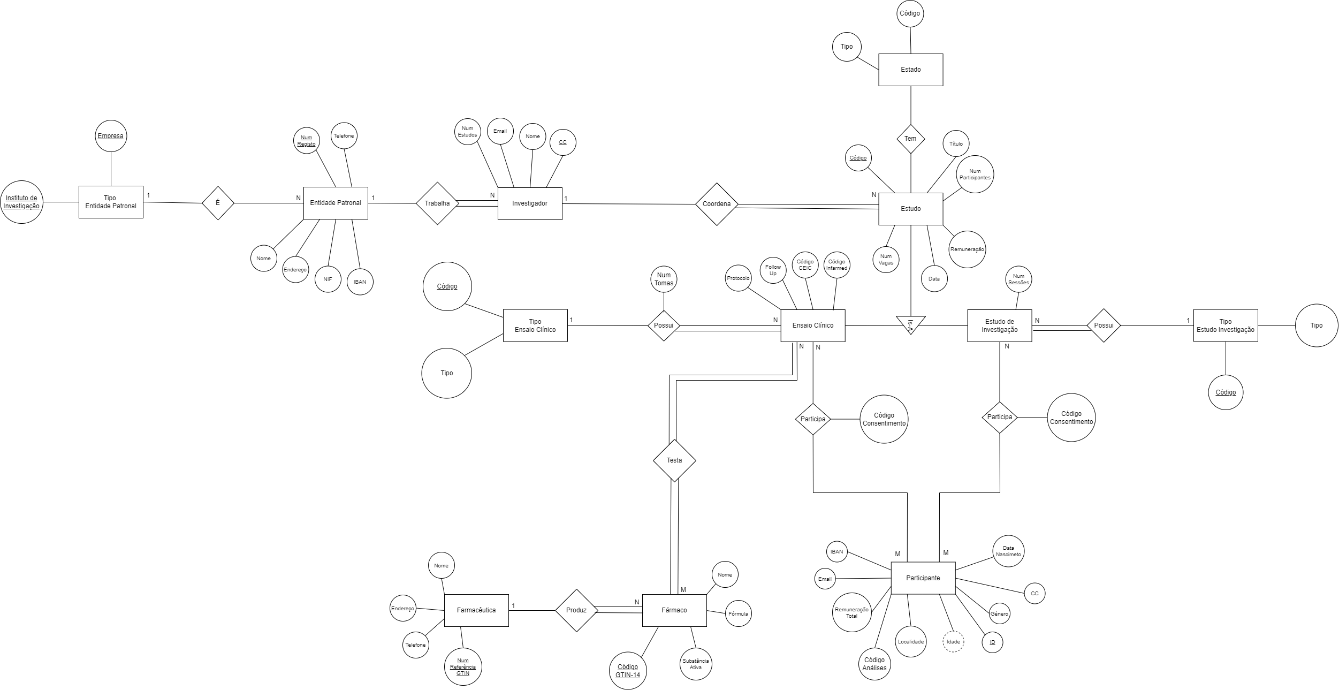
**Tabela 2 –** Requisitos e funcionalidades associadas aos tipos de utilizadores da plataforma.

|  |  |
| --- | --- |
| **Utilizador** | **Requisitos** |
| Participantes | * A identidade dos Participantes está protegida, sendo estes identificados na plataforma com recurso a um ID gerado por esta * São recompensados pela participação nos estudos, podendo associar um IBAN à sua conta onde serão transferidos os pagamentos * Devem assinar um documento de consentimento antes da participação num Estudo, ao qual está associado um código armazenada na base de dados |
| Investigadores | * Podem alterar a informação e os critérios de seleção dos seus estudos, quando estes ainda estão no estado inicial |

# Desenho Conceptual da Base de Dados

Ao longo do desenvolvimento do projeto foram feitas diversas adaptações face ao plano inicial, para que este se tornasse o mais próximo possível da realidade, e modelasse de forma mais fiel a base de dados desenvolvida. Como resultado, foram realizadas algumas adaptações ao desenho conceptual apresentado na APF-E.

Seguidamente será apresentado o estado atual do DER e DR, e listadas as mudanças efetuadas, bem como o porquê de elas terem sido feitas.



**Figura 1 –** Versão atual do DER. Este diagrama está também incluído na Entrega Final, para uma melhor visualização.

Levando em consideração as sugestões do professor, e ao iniciar a criação de tabelas na base de dados, foram feitas as seguintes alterações:

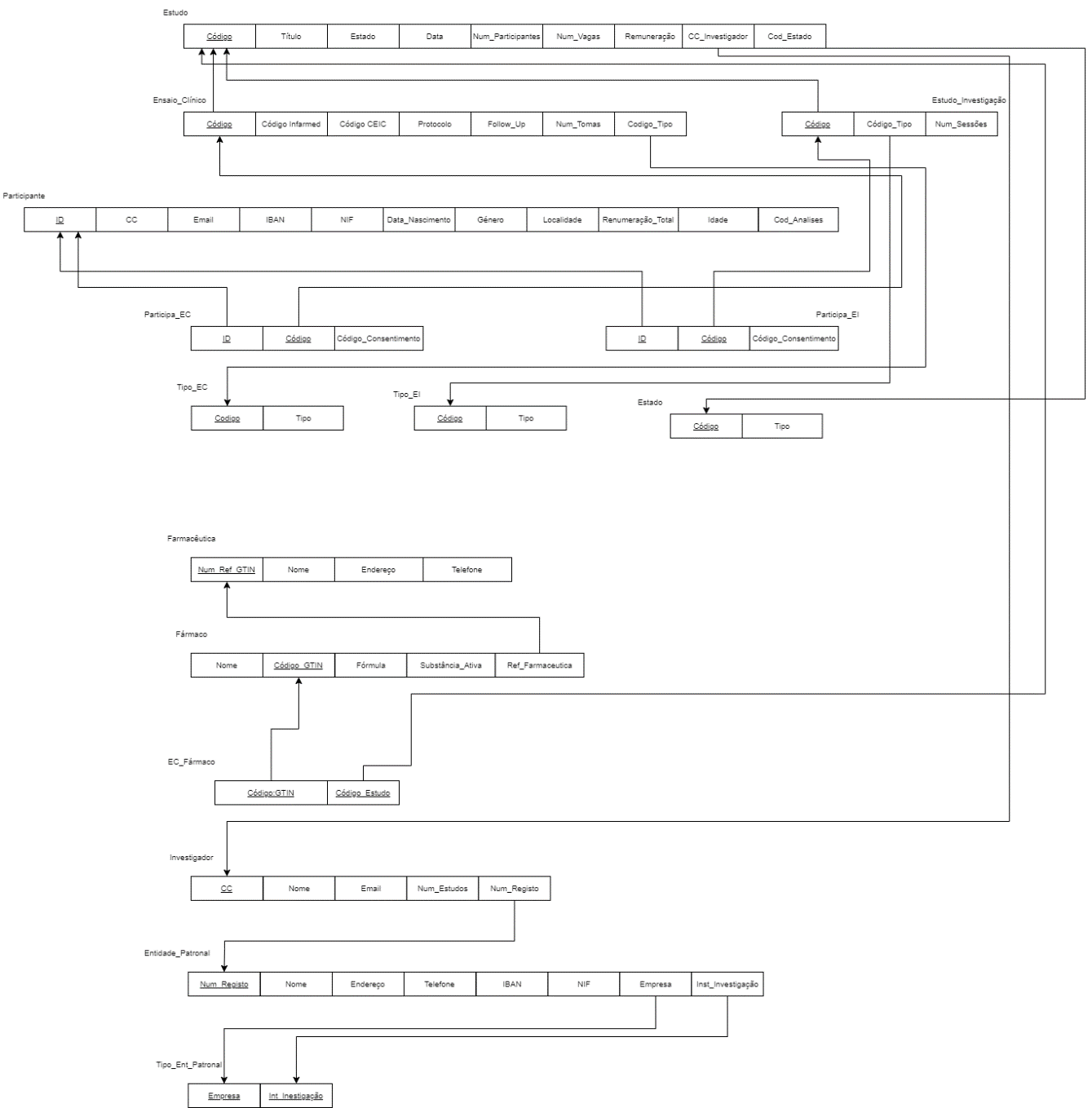
1. Retirar especialização da Entidade Patronal, sendo criada uma nova entidade Tipo Entidade Patronal que lista os dois tipos possíveis, incluída como chave estrangeira na tabela da Entidade Patronal
2. Condições de Seleção, Tipo Estudo Investigação, Tipo Ensaio Clínico e Estado passaram a ser entidades, com moldes similares à alteração anterior
3. Análises clínicas passou a ser um atributo do Participante
4. Adicionados atributos novos ao Fármaco

No caso do ponto um, não era necessário especializar a Entidade Patronal, sendo que as especializações Instituto e Empresa não tinham atributos para além dos da Entidade Patronal.

No ponto dois, todos eram atributos anteriormente, contudo necessitavam de ser atributos compostos, o que não era correto.

No ponto três, as Análises Clínicas eram um atributo de Participa\_EC (entre Participante e Ensaio Clínico), o que iria complicar a implementação do rastreio de participantes com estudos clínicos em dia, visto que estes iriam se encontrar separados do Participante geral, o que levou à mudança para ser um atributo desta última entidade.

Finalmente, o ponto quatro, foram acrescentados códigos aos fármacos de acordo com a pesquisa efetuada sobre validação dos mesmos em Portugal e referenciada na Análise de Requisitos.

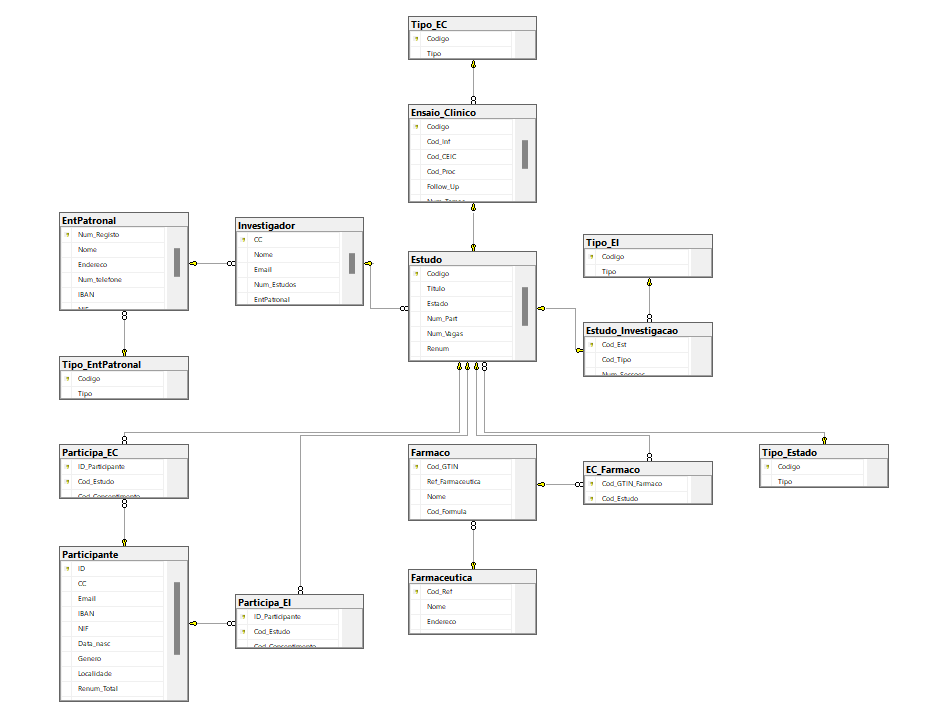


**Figura 2 -** Versão atual do ER. Este diagrama está também incluído na Entrega Final, para uma melhor visualização.

O ER foi alterado em uníssono com o DER:

* Foram adicionadas tabelas novas: Estado, Tipo\_Ent\_Patronal, Tipo\_EC, Tipo\_EI
* Retiradas as tabelas de especialização de Entidade Patronal
* Adicionado o atributo Cod\_Analises ao Participante
* Adicionados diversos atributos aos Fármacos

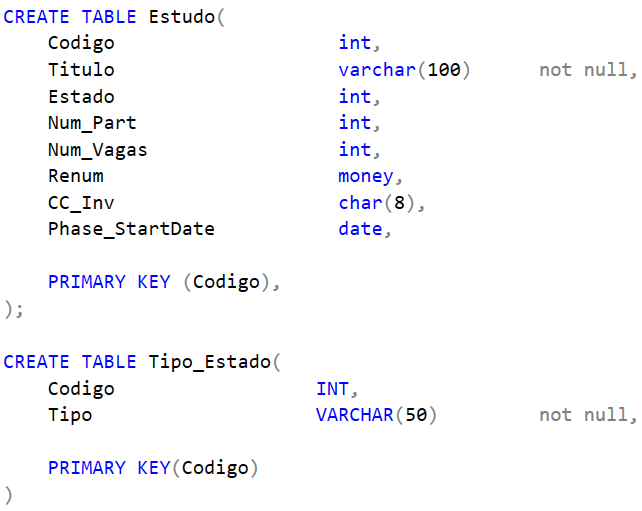
Por fim o diagrama gerado através do SQL Server da nossa Base de Dados:



**Figura 3** – Diagrama Gerado pelo SQL Server.

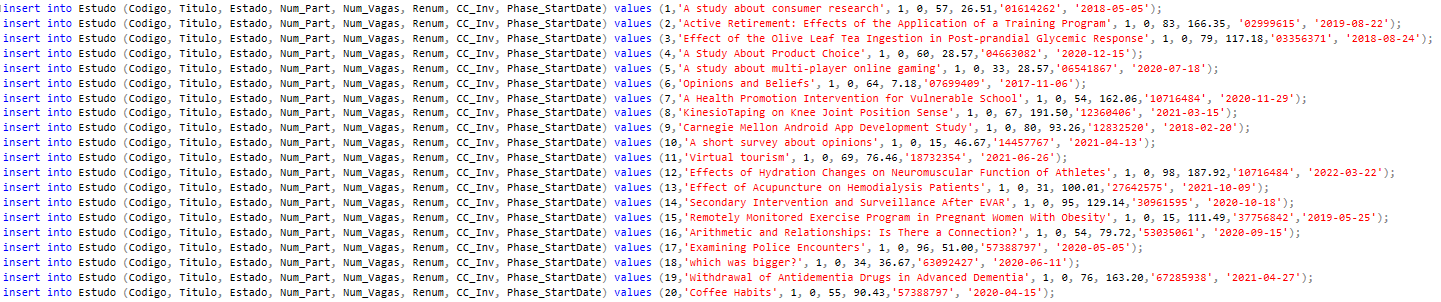
# SQL-DDL e SQL-DML

Para a criação da base de dados foi utilizado o script SQL\_DDL\_RB, feito de acordo com o desenho da BD anteriormente apresentado.



**Figura 4 -** Exemplo do SQL\_DDL\_RB

Para popular a BD foram utilizados os scripts SQL\_DML\_\*, onde se recorreu a um gerador online de dados, para facilitar o processo.



**Figura 5** - Exemplo SQL\_DML\_Estudos

# Camada de Interação com a Base de Dados

A interface da nossa plataforma foi desenvolvida utilizando a Framework Windows Forms para a plataforma .NET no Visual Studio, através da linguagem de programação Visual Basic.

Para a interação com a base de dados foram criados cinco formulários principais, acrescentando que, para além destes, também existem outros que servem como complemento para a implementação de todas as funcionalidades.

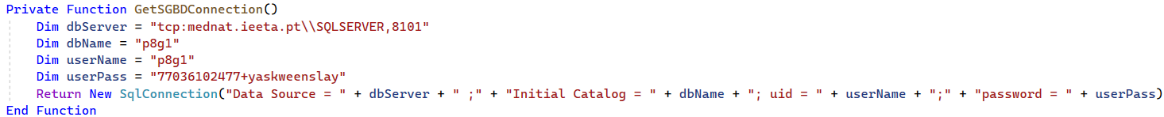
* **Main.vb -** Onde foi feita a conexão com a base de dados através da classe SqlConnection.
* **Participantes.vb**
  + ParticipantesStats.vb
* **Investigador.vb**
  + InvestigadorStats.vb
* **EntidadePatronal.vb**
  + EntPatronalStats.vb
* **Estudos.vb -** Onde foi implementada toda a lógica e funcionalidades envolventes nos processos dos Estudos, como a criação, recrutamento e finalização.
* Recrutar.vb
* ParticipantesEstudo.vb
* EditEC.vb
* EditEI.vb

Para a criação dos formulários foram utilizados vários recursos do VB, como por exemplo Label, TextBox, ComboBox, Button. Porém a peça fundamental foi o DataGridView, um objeto que permite a apresentação de dados em formato de tabela, sendo que muitas das funções da Main.vb servem para popular os diferentes DataGridView encontrados nos demais formulários.

**Funções implementadas na Main.vb:**

* GetSGBDConnection()

Esta function foi feita para estabelecer a conexão à base de dados através da classe SqlConnection.



**Figura 6 –** Função GetSGBDConnection()

* PopulateList(grid As DataGridView, text As String)

Este Sub foi criado para popular as DataGridView através de um comando SQL, sendo que se recorre à classe SqlCommand e SqlDataAdapter para o efeito.

Uma imagem com texto

Descrição gerada automaticamente

**Figura 7 –** Sub PopulateList()

* PopulateListSP(grid As DataGridView, procedureName As String, parameters As Dictionary(Of String, Object))

Este Sub foi criado para popular uma DatGridView através da chamada de um SP, para aquelas tabelas que necessitam de mais verificações

Uma imagem com texto

Descrição gerada automaticamente

**Figura 8 –** Sub PopulateListSP()

* CallSP(procedureName As String, parameters As Dictionary(Of String, Object), type As Int16)

Esta Function foi criada para chamar um SP, cujo objetivo é fazer alguma mudança à BD, para além disso também foi dividida no caso da necessidade de ter um valor de retorno. Este valor de retorno é um status, que serve posteriormente para a apresentação das mensagens de erro.

Uma imagem com texto

Descrição gerada automaticamente

**Figura 9 –** Função CallSP()

* FecthSimpleData(query As String)

Esta function foi criada para obter informação das tabelas da BD através de um comando SQL.

Uma imagem com texto

Descrição gerada automaticamente

**Figura 10 –** Função FetchSimpleData()

Em todos os Sub e Function temos mensagens de erros apresentados caso o comportamento do SqlCommand não seja o esperado, que são apresentadas em MessageBox.

Para garantir a segurança da BD e prevenir SQL Injection por parte do utilizador, recorremos a uma parametrização das instâncias criadas da classe SqlCommand, como apresentado anteriormente nos Sub e Function utilizados.

# Interfaces e Funcionalidades Implementadas

Neste capítulo são apresentadas as funcionalidades presentes na plataforma desenvolvida, bem como a descrição da interação da interface do lado do cliente com a base de dados e o código SQL desenvolvido que permite a implementação destas.

Para o desenvolvimento da plataforma foram implementadas 4 secções principais, cada uma dedicada às 4 entidades mais importantes da base de dados desenvolvida. Estas secções e as respetivas funcionalidades estão descritas na tabela seguinte:

**Tabela 3 –** Funcionalidades implementadas na plataforma ResearchBase

|  |  |
| --- | --- |
| **Secção** | **Funcionalidades** |
| Estudos | * Inserção de novos Estudos * Edição de Estudos existentes na base de dados, desde que estes ainda estejam no estado 1 * Pesquisa de Estudos pelo seu Título * Filtração de Estudos de acordo com o seu tipo e com o seu estado atual * Processo de recrutamento de Participantes elegíveis para os Estudos que estejam no estado 1 * Finalização de Estudos ativos (no estado 2), com pagamento da respetiva remuneração aos seus Participantes * Acesso à lista de Participantes recrutados para os Estudos no estado 2, ou estado 3 (Terminados) |
| Participantes | * Visualização de todos os Participantes inscritos na plataforma e da informação referente a estes * Adição de novos Participantes * Edição de alguns dos dados dos Participantes já existentes * Para cada Participante, é possível visualizar informação relativa aos Estudos em que este participou, bem como estatísticas como a remuneração total recebida por este ou o número de estudos por si completados |
| Investigadores | * Visualização de todos os Investigadores inscritos na plataforma e da informação referente a estes * Adição de novos Investigadores * Edição de alguns dos dados dos Investigadores já existentes * Lista de Estudos coordenados por cada Investigador e informação estatística referente a estes dados |
| Entidade Patronal | * Visualização de todos os institutos de investigação registados na plataforma e da informação referente a estes * Adição de novos Institutos * Visualização da lista de Estudos coordenados por investigadores associados a cada Entidade, bem como o total de dinheiro que a Entidade já investiu nos seus Estudos |

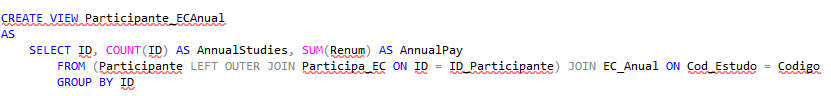
As Views, Stored Procedures e Triggers desenvolvidos que permitiram a aplicação destas funcionalidades serão apresentados de seguida.

## Views

Foram criadas algumas views no desenvolvimento deste projeto, com o objetivo de proporcionar uma melhor visualização de alguma da informação guardada na base de dados, bem como de diminuir a complexidade de alguns dos *queries* feitos ao longo do trabalho.

As Views desenvolvidas foram, desta forma:

* **EC\_Anual –** Vista sobre todos os Ensaios Clínicos realizados no ano presente, guardando a informação relativa à remuneração e ao estado de progresso destes. Para tal, considera a fase em que o estado atual do Estudo foi iniciado, registado no atributo “Phase\_StartDate” da tabela Estudo, sendo efetuado um JOIN sobre as tabelas Estudo e Ensaio\_Clínico.
* **Participante\_ECAnual –** Lista todos os participantes em Ensaios Clínicos realizados no ano atual, agregando o número de estudos feitos por estes e a remuneração total que eles ganharam neste ano, nestes estudos. É feita sobre um LEFT OUTER JOIN entre a vista EC\_Anual e a tabela Participa\_EC.
* **Lista\_Participantes –** Agrega todos os participantes dos Ensaios Clínicos e Estudos de Investigação realizados na plataforma, de maneira a permitir operações que necessitem de informação de ambos os tipos de estudos. É gerada com base numa UNION entre as tabelass Participa\_EC e Participa\_EI.
* **EC\_Total e EI\_Total –** Listam a informação completa dos Ensaios Clínicos e dos Estudos de Investigação respetivamente, juntando os atributos destas duas relações aos atributos da sua relação-pai, a tabela Estudo, com recurso ao JOIN.

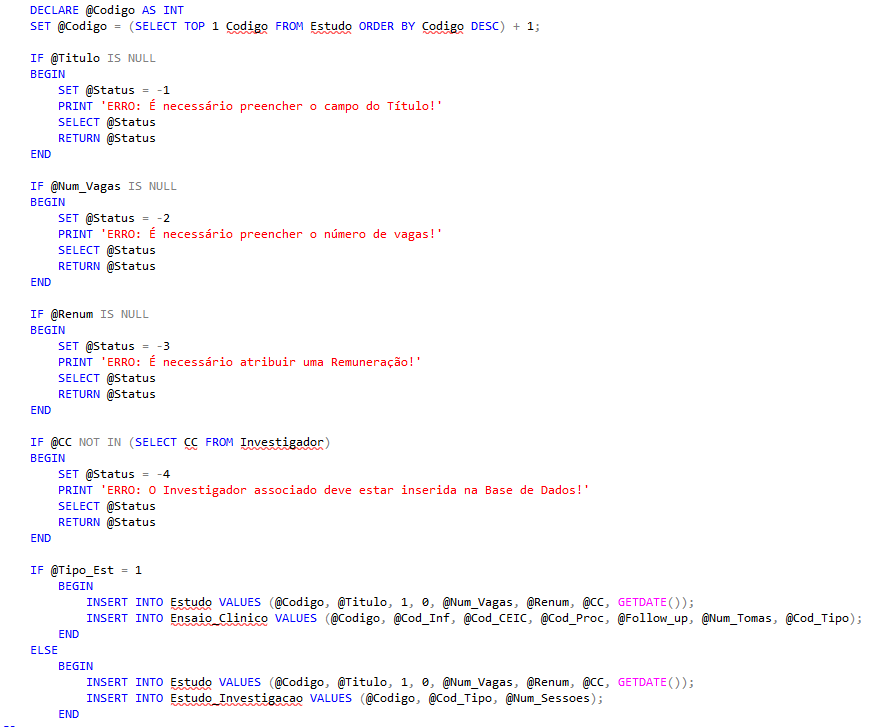


**Figura 11 –** Exemplo da criação da View Participante\_ECAnual.

## Secção “Estudos”

**Adição de Novos Estudos**

A adição de novos Estudos à plataforma é feita com base na Stored Procedure insertStudy e no trigger insertStudyTrigger.



**Figura 12 –** Corpo da Stored Procedure insertStudy.

A **SP insertStudy** vai receber como parâmetro todos os campos referentes aos atributos das tabelas Estudo, Ensaio\_Clinico e Estudo\_Investigacao, passados pela sua chamada no lado do cliente. Além disto, é passado também um parâmetro Tipo\_Est, que assume o valor 1 se o estudo a ser criado for um Ensaio Clínico, ou 2 se for um Estudo de Investigação, e é declarado um output, Status, com valor *default* 1.

Todos os parâmetros referentes aos atributos vão ter como valor *default* NULL, caso não sejam passados. Desta forma, é verificado se os campos obrigatórios das tabelas estão ou não presentes na chamada da SP: Se não estiverem, Status vai assumir um valor de retorno associado ao campo em falta, sendo utilizado no lado do cliente para produzir uma mensagem de erro que avisa o utilizador sobre a necessidade de preencher o campo em questão aquando da inserção de um estudo.

Caso todos os campos obrigatórios estejam presentes, o estudo é então inserido como tuplo na relação Estudo e na relação Ensaio\_Clinico/Estudo\_Investigação, com estado *default* 1, número de participantes igual a 0 e a Phase\_StartDate a ter o valor da data atual. Após inserção, é ativado o **trigger insertStudyTrigger** que incrementa o número de estudos no tuplo do investigador responsável, na tabela Investigador.

**Edição de Estudos Existentes**

A edição de Estudos é feita com recurso à **SP editStudy**. O funcionamento desta é muito similar à SP insertStudy, sendo também realizada a verificação da presença de campos obrigatórios na chamada da *procedure*, e a passagem de uma mensagem de erro ao lado do cliente caso estes não estejam presentes.

**Pesquisa de Estudos por Título**

A pesquisa de Estudos pelo seu título é feita com recurso à **SP searchEstudo**, apresentada de seguida, associada ao **índice estudoTitulo**, implementado sobre o atributo Título da relação Estudo, de maneira a otimizar o tempo de pesquisa (sobretudo quando a base de dados estiver preenchida com mais entradas).

Uma imagem com texto

Descrição gerada automaticamente

**Figura 13 –** Stored Procedure searchEstudo e índice estudoTitulo, usados na pesquisa de Estudos por título.

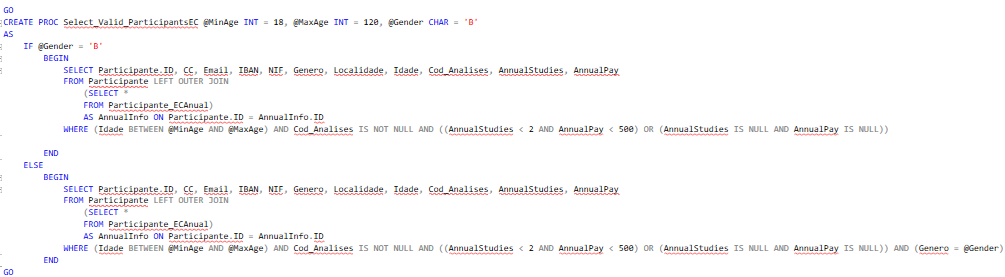
É de notar que, no lado do cliente, esta SP é chamada sempre que este insira um novo caracter na barra de pesquisa.

**Recrutamento de Participantes**

De maneira a iniciar o processo de recrutamento para um novo Estudo na plataforma, é possível ao utilizador definir os critérios de seleção dos Participantes elegíveis para um determinado Estudo, considerando características (opcionais) como o intervalo de idades e o género da pessoa. Para tal, é realizada uma chamada á base de dados com recurso às **SPs Select\_Valid\_ParticipantsEI** e **Select\_Valid\_ParticipantsEC**, conforme o estudo se trate de um Estudo de Investigação ou de um Ensaio Clínico, respetivamente.

Nestes SPs, são passados como parâmetros as idades mínimas, idades máximas e o género definido pelo utilizador no lado do cliente, que assumem valores *default* 18, 120 e B (Todos), respetivamente, caso não sejam passados na chamada. A filtração das pessoas elegíveis é feita com recurso a *queries* à relação Participante considerando estes parâmetros, sendo apresentada a *data set* resultante no lado do cliente.

No caso dos Ensaios Clínicos é ainda feita uma filtração adicional, considerando as leis de participação em Ensaios Clínicos referidas na Análise de Requisitos: é verificado se os Participantes têm as análises clínicas em dia e se ainda não atingiram o limite de Ensaios Clínicos e de remuneração ganha nestes para o ano em questão, com recurso à **view Participante\_ECAnual**.



**Figura 14 –** Stored Procedure Select\_Valid\_PartipantEC.

Quando os Participantes são adicionados ao Estudo pelo utilizador, é chamada uma SP, **Add\_ParticipantEI** ou **Add\_ParticipantEC**, que vai verificar se o número de vagas no estudo já foi preenchido. Caso tenha sido, é passada uma mensagem de erro apresentada no lado do cliente, sendo impedida a efetuação da inserção. Caso não tenha, os participantes e o estudo são registados nas tabelas Participa\_EI ou Participa\_EC, conforme se trate de um Estudo de Investigação geral ou de um Ensaio Clínico, respetivamente. O número de participantes no estudo é então incrementado, com os **triggers acceptEC e acceptEI**.

**Uma imagem com texto

Descrição gerada automaticamente**

**Figura 15 –** Stored Procedure Add\_ParticipantEI.

O processo de Recrutamento de um Estudo no estado 1 pode ser finalizado a qualquer momento, sendo para tal chamada a **SP finishRecruiting**. Esta SP vai verificar se o estudo já tem Participantes recrutados (sendo produzido uma mensagem de erro no lado do cliente caso não tenha, utilizando o mecanismo anteriormente explicado para a adição de novos Estudos), e atualizar o estado do Estudo para 2 (Ativo). É de notar que foi decidido permitir que um Estudo possa ser finalizar o recrutamento mesmo que o número de vagas não tenha sido preenchido.

**Finalização de um Estudo Ativo**

Os Estudos no estado 2 podem ser finalizados na interface. Para tal, é chamada **a SP completeStudy**, que tem como função pagar aos Participantes a remuneração relativa ao Estudo completado, e atualizar o estado do Estudo pra 3 (Completado).

Foi utilizado um **cursor** que vai percorrer a lista de Participantes do Estudo em questão (identificado pelo seu código através de um parâmetro da SP), através da **view Lista\_Participantes** anteriormente referida.

Uma imagem com texto

Descrição gerada automaticamente

**Figura 16 –** Stored Procedure completeStudy.

## Secções Participantes, Investigadores e Entidades Patronais

Os formulários dos Participantes, Investigadores e Institutos são bastante semelhantes, pois as suas funcionalidades baseiam-se em adicionar, editar e procurar tuplos, ou visualizar as suas estatísticas. Deste modo, a apresentação das funcionalidades vai ser feita de forma agrupada.

**Adição de Novos Tuplos**

É possível adicionar novos tuplos a qualquer uma das entidades, recorrendo aos SP associados a cada uma**, addParticipante, addInvestigador, addEntidadePatronal**.

O funcionamento destas procedures é muito semelhante ao descrito para a SP addEstudo na secção anterior, sendo feita a validação dos argumentos passados na chamada para verificar a presença, ou não, dos campos obrigatórios, e retornada uma mensagem de erro no lado do cliente caso se verifique que um destes campos não está presente na chamada.

**Edição de Tuplos**

A edição de tuplos está presente nas entidades Participante e Investigador, através das SP **editParticipante** e **editInvestigador**, semelhantes a **editEstudo** anteriormente explicado. Não foi adicionada a capacidade de edição à relação Entidade Patronal, uma vez que se considerou que a informação que descreve estes institutos não deve ser alterada.

**Pesquisa de Tuplos**

É possível a realização da pesquisa de tuplos nas três entidades, com recurso aos SP’s e triggers seguintes:

* **Participante –** Pesquisa pelo CC, com a SP **searchParticipante** e o índice **ccIndex** aplicado neste atributo.
* **Investigador –** Pesquisa pelo nome, com a SP **searchInvestigador**  e o **índice nameIndex** aplicado neste atributo.
* **Entitade Patornal -** Pesquisa pelo nome, com a SP **searchEntPatronal** e o **índice** **nameIndex** aplicado neste atributo.

**Estatísticas**

Foi implementada a capacidade de visualizar estatísticas referentes a cada uma das entidades mencionadas, com recurso a *queries* diretas passadas por SQL Commands no lado do cliente, e que envolvem várias funções de agregação. As estatísticas presentes em cada entidade são as seguintes:

* **Participante –** É possível ver a lista de Estudos em que cada Participante individual entrou, verificar o número total de Estudos completados por este e a remuneração por si ganha.
* **Investigador –** É possível ver a lista de Estudos coordenados por cada Investigador individual.
* **Entitade Patornal –** É possível aceder à lista de Estudos coordenados por Investigadores associados a cada uma das entidades, bem como ver a remuneração total paga por cada Entidade aos Participantes.



**Figura 17 –** Exemplo da query utilizada para determinar a remuneração total paga por uma Entidade Patronal pelos estudos por si coordenados.

# Conclusão

Consideramos que a plataforma desenvolvida foi de encontro aos requisitos e objetivos definidos no início do projeto, tendo sido implementadas as funcionalidades estabelecidas como mais importantes no início do desenvolvimento do projeto.

No entanto, a falta de tempo impediu a implementação de características adicionais. Entre estas, temos como exemplo a possibilidade de pesquisar pelos fármacos presentes na base de dados, e de apresentar estatísticas relacionadas com esta entidade, como por exemplo a lista de Ensaios Clínicos em que foram usados. Outro exemplo seria a implementação de transações que permitam o controlo de concorrência e a recuperação de falhas. Esta implementação não foi considerada essencial, visto a dimensão do projeto e a escala relativamente baixa dos dados inseridos na base de dados, sendo que seria, no entanto, produtivo a sua implementação para permitir uma escalabilidade da plataforma no futuro.

Ainda assim, consideramos que cumprimos o objetivo de implementar na prática os conceitos teóricos adquiridos ao longo do semestre na unidade curricular, tendo o desenvolvimento deste projeto sido uma mais-valia para o nosso conhecimento na área de desenvolvimento de bases de dados.