**Centro de Alto Rendimento**

Base de dados de gestão de um centro de alto rendimento

Universidade de Aveiro

Licenciatura em Engenharia de Computadores e Informática

Base de Dados P2G1

Regente: Prof. Carlos Costa

Professor Orientador: Prof. Tiago Marques Godinho

André Clérigo, 98485

Pedro da Rocha, 98256

26 de junho 2021

Índice

[Introdução 3](#_Toc107150099)

[Análise de Requisitos 4](#_Toc107150100)

[Entidades e multiplicidade 4](#_Toc107150101)

[Diagrama Entidade Relação (DER) 5](#_Toc107150102)

[Esquema Relacional (ER) 6](#_Toc107150103)

[Stored Procedure (SP) 6](#_Toc107150104)

[View 7](#_Toc107150105)

[User Defined Function (UDF) 8](#_Toc107150106)

[Trigger 8](#_Toc107150107)

[Index 10](#_Toc107150108)

[Conclusão 10](#_Toc107150109)

# Introdução

No âmbito da unidade curricular de Base de Dados, da Licenciatura em Engenharia de Computadores e Informática foi-nos proposto a criação de um projeto que fizesse a gestão de um sistema funcional e com complexidade razoável aplicável ao mundo real. O seguinte relatório irá descrever sucintamente o nosso projeto dando enfase nas partes que achamos mais fulcrais para o funcionamento do mesmo.

O código enviado em anexo contém o código usado para criar a base de dados e utilizar a mesma no diretório **SQL-Code**, desenvolvemos um WFA em C# e o código relativo à mesma encontra-se no diretório **Aplicacao**.

Para que seja possível a testagem da nossa aplicação noutra base de dados para além daquela que foi usada no desenvolvimento nas aulas práticas é necessário aceder ao ficheiro **AppData.cs** e mudar os atributos **BD\_STRING**, **username** e **password** para as credenciais desejadas.

A screenshot of a computer

Description automatically generated with medium confidence

Figura - Ficheiro AppData.cs onde podemos alterar a conexão à base de dados utilizada

Segue também um conjunto de vídeos demonstrativos usados para a apresentação dentro do diretório de **vídeos**.

# Análise de Requisitos

Podem registar-se atletas, identificados distintamente pelo seu número de inscrição nas seguintes modalidades: natação, judo, ténis, basquetebol, atletismo, voleibol, badminton, rugby.

* Existem 3 tipos de empregados: treinador, secretariado e empregados de saúde, conhecidos pelos seus números de funcionários.
* Para um treinador registar a avaliação de uma sessão de treino é necessário o atleta ter reservado anteriormente o estabelecimento onde pretendia treinar.
* Os estabelecimentos dividem-se entre piscinas, pistas e ginásios.
* De maneira a reservar espaços, os atletas têm que ter um diagnóstico ‘apto’ para treinar, realizado numa consulta pelos empregados de saúde, qualquer que seja a sua especialidade fisioterapia, medicina geral, cardiologia, oftalmologia, nutrição.

# Entidades e multiplicidade

* **Atleta** é identificado por um n\_inscricao e tem atributos não únicos como por exemplo, nome, modalidade, estado\_saude, data\_nasc e idade (atributo calculado a partir da data\_nasc).
* **Empregado** é identificado por um n\_funcionario e nif (sendo n\_funcionario a chave primária) e tem atributos não únicos como o salário e nome.
* **Treinador**, **Secretariado** e **Empregado de saúde** que são entidades que se relacionam com o Empregado (is-a *disjoint*) e tem atributos específicos de modalidade, id\_estabelecimento e especialidade respetivamente.
* **Reserva** é identificada por um num\_reserva e tem atributos não únicos como por exemplo, data\_reserva, hora\_inicio, duração e hora\_fim (calculado a partir da hora\_inicio e duracao). Os atributos n\_insc\_atleta, n\_func\_treinador e id\_estabelecimento são as chaves estrangeiras utilizadas.
* **Consulta** é identificada por um id\_consulta e tem atributos não únicos como por exemplo, data\_consutla e estado\_saude (enum de “apto” ou “inapto”). Usada n\_func\_emp e n\_insc\_atleta como chaves estrangeiras.
* **Sessão de Treino** é identificada por um num\_reserva usado como chave estrangeira e tem uma avaliação de 0 a 20.
* **Estabelecimento** identificado por um id e tem atributos não únicos como por exemplo, data\_manutencao, data\_aberto (após às 05h00), data\_fecado (antes das 23h30) e uma lotação para espectadores que deve ser maior que 0.
* **Pista, Piscina** e **Ginásio** que são entidades que se relacionam com o Estabelecimento (is-a *disjoint*) e tem atributos específicos de tipo\_solo, temp\_agua e qtd\_aparelhos respetivamente.

# Diagrama Entidade Relação (DER)

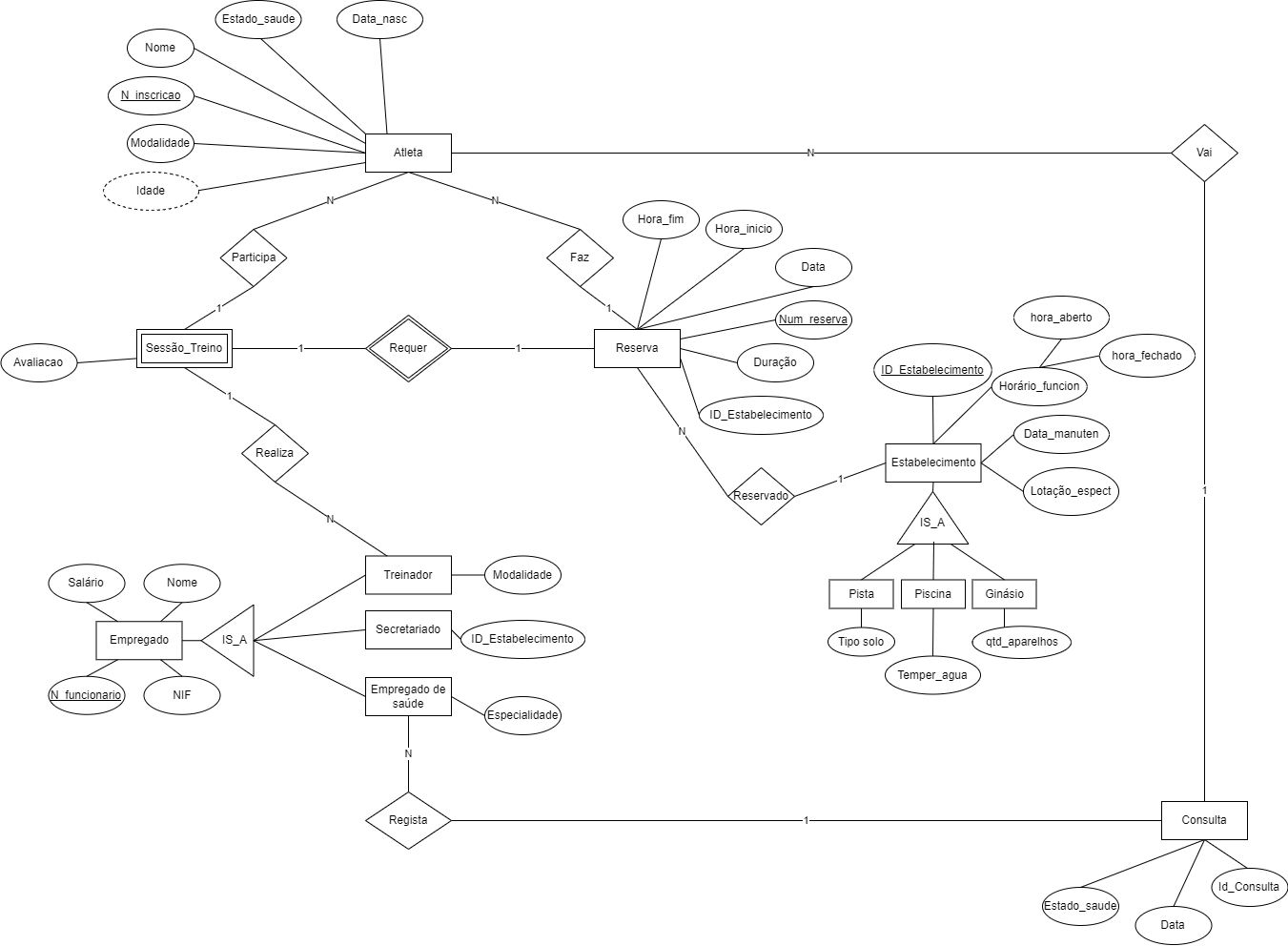


Figura - Diagrama Entidade Relação

# Esquema Relacional (ER)

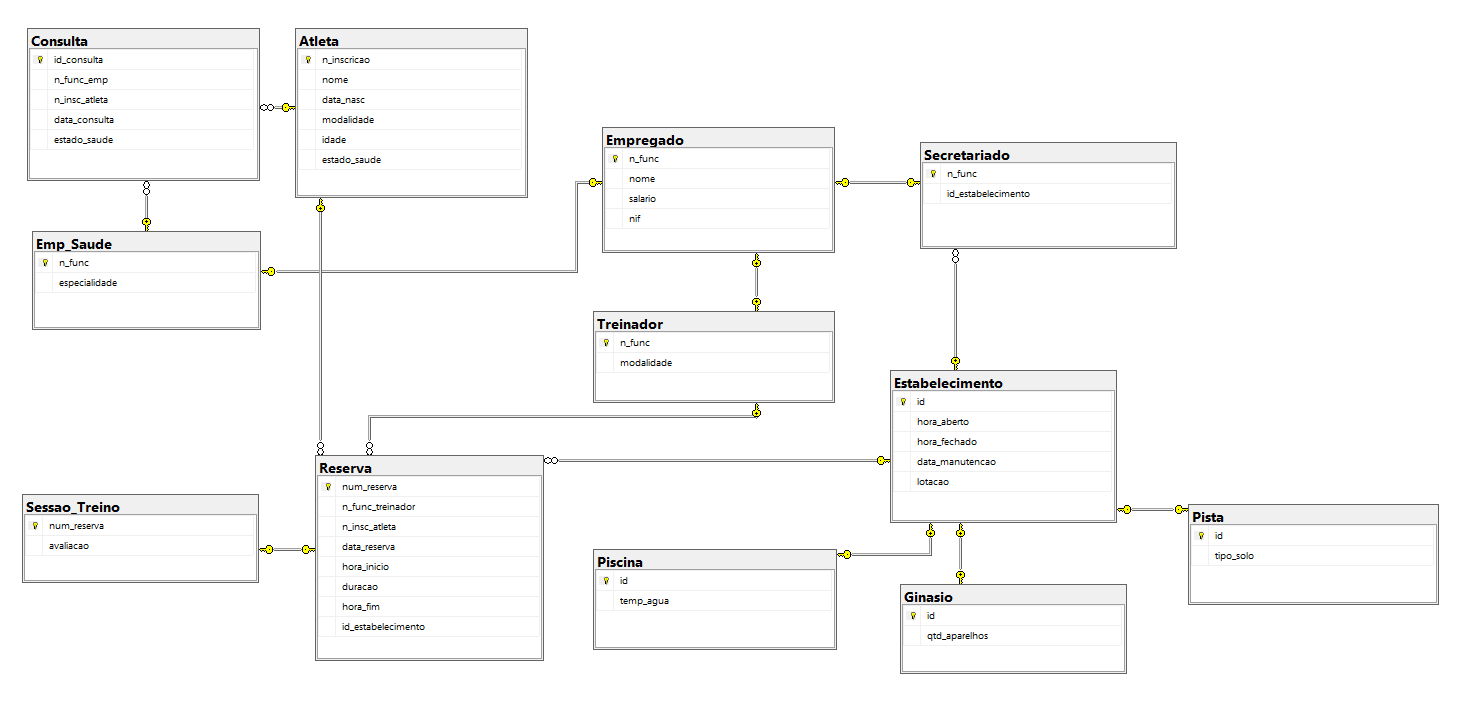


Figura - Esquema Relacional

# Stored Procedure (SP)

Para garantir uma relação is-a *disjoint* nas entidades Pista, Piscina, Ginásio, Secretariado, Empregado de Saúde e Treinador fizemos SPs que criam uma entidade mãe e logo de seguida criam a respetiva entidade pretendida.

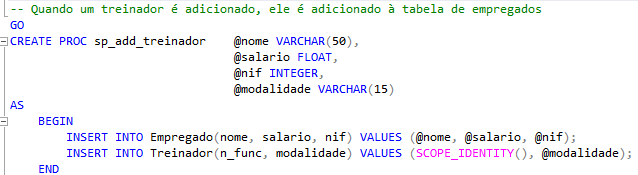


Figura - Stored Procedure para adicionar Treinador

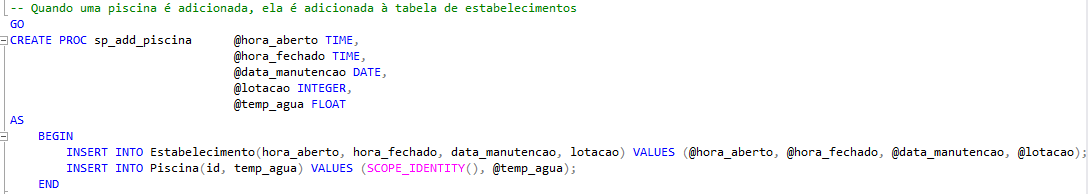


Figura - Stored Procedure para adicionar Piscina

# View

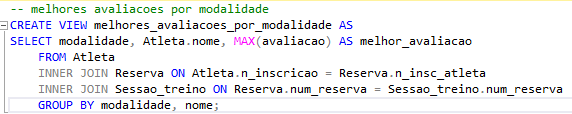
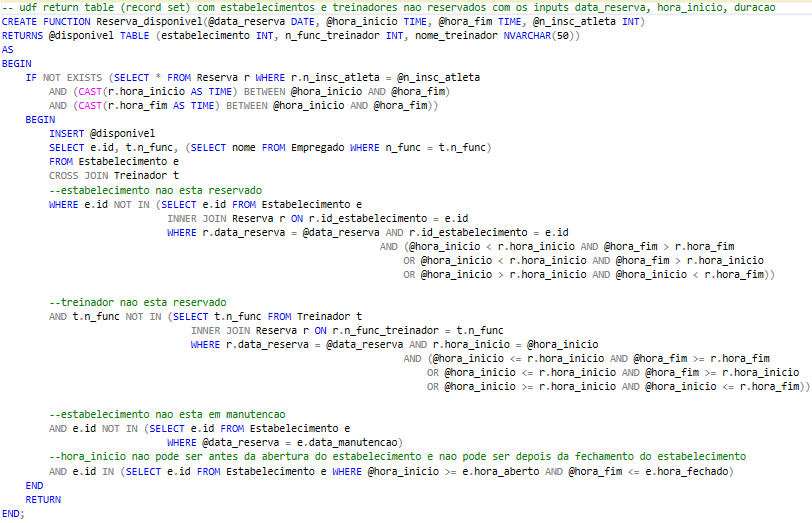
Como os atletas têm a opção de verificar as estatísticas da sua avaliação e de outros por modalidade praticada, é criada uma tabela auxiliar (*view*) para ser mostrada ao utilizador da interface de maneira a comparar as suas avaliações com o panorama geral.

Figura - View melhor avaliação por modalidade

# User Defined Function (UDF)

De forma a devolver uma tabela com todas as combinações disponíveis (isto é, nem o atleta nem ninguém tem alguma reserva a essa hora) para determinada data e hora de treino, podendo os estabelecimentos e o treinador serem indicados pelo utilizador ou considerar todos, utilizámos uma UDF que regista esses valores na tabela devolvida.



# 

# Trigger

Para manter a integridade dos dados da reserva ao serem adicionadas e não existirem colisões de horários, criou-se um *trigger* que é ativado ao adicioná-la, que verifica se as horas de treino da reserva acontecem durante o horário do estabelecimento e que a data de reserva não colide com a data de manutenção. Certifica ainda que o atleta está apto de saúde para treinar e que a data da última consulta não foi há mais de 6 meses. Por último, confere que o atleta não tem já uma reserva nessa data e hora.

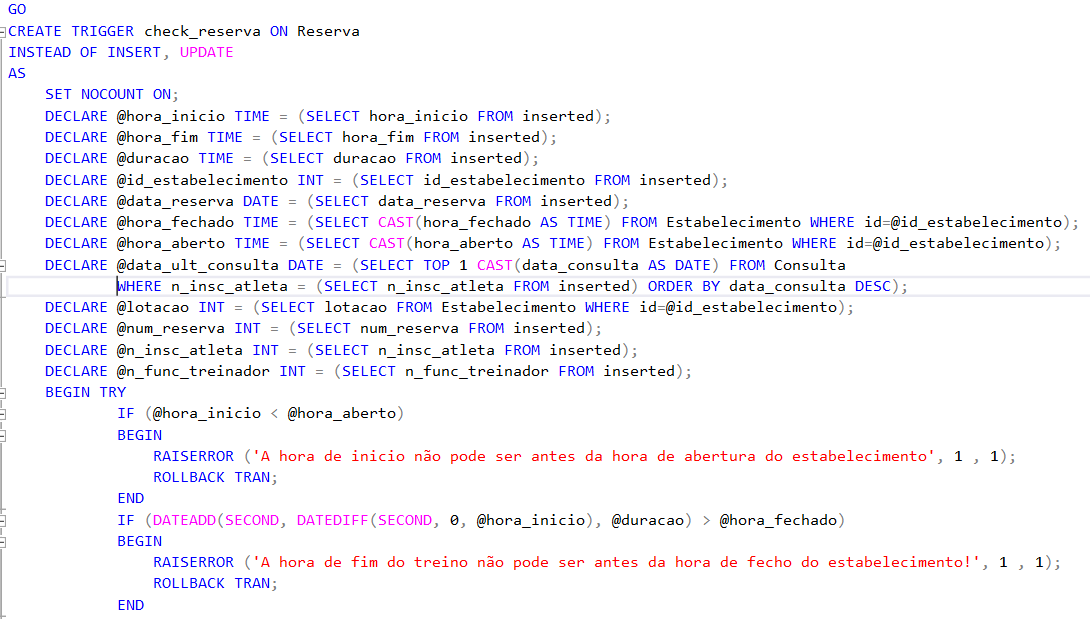
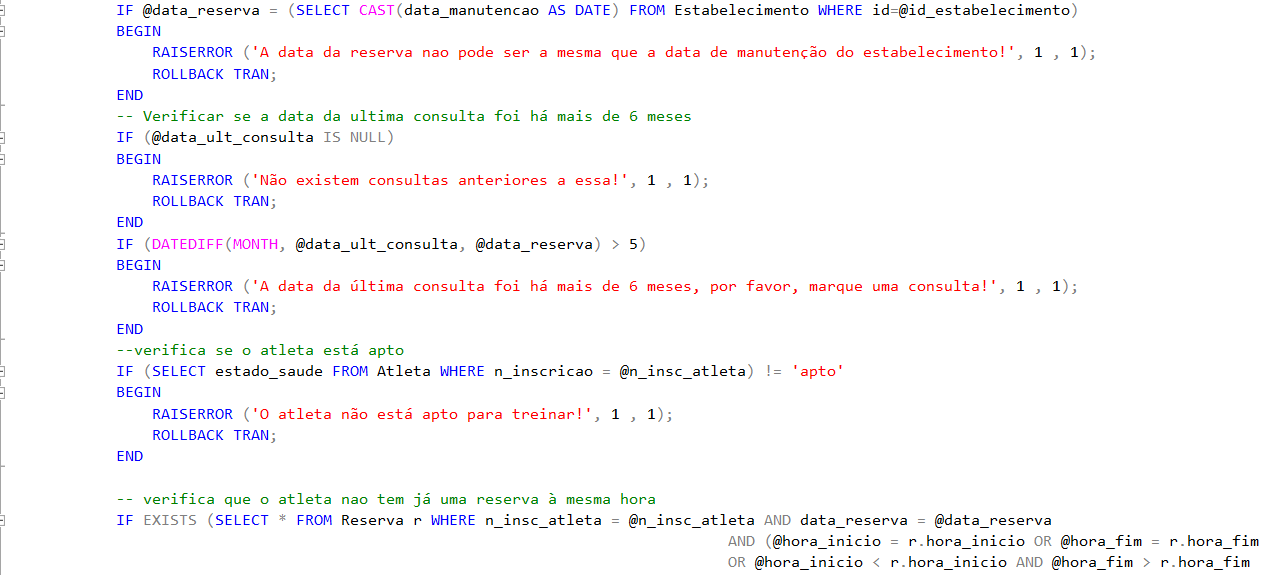
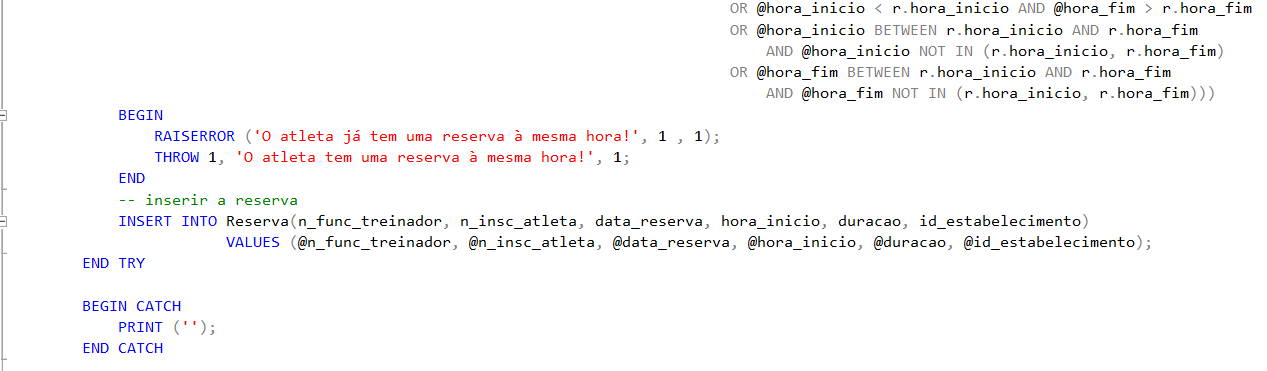
 .

Figura - Trigger check reserva

# Index

Com vista a iterar mais eficazmente pelas reservas existentes, organizou-se a tabela de Reserva pelas datas, visto que o acesso mais complexo à mesma verifica os dados por esse mesmo atributo.



Figura - Index por data

# Conclusão

Podemos concluir que os principais objetivos e as principais funcionalidades da aplicação para a gestão de um centro de alto rendimento foram cumpridas. O empregado do secretariado pode adicionar, remover Atletas, Empregados de Saúde, Treinadores, Secretariado, Piscinas, Pistas e Ginásios.

Os empregados de saúde podem registar consultas e verificar o histórico de consultas de um atleta.

Os treinadores podem registar uma sessão de treino dando uma avaliação (0-20) ao mesmo, podendo consultar as reservas feitas para si.

Os atletas podem consultar as suas reservas, fazer reservas novas, consultar as suas avaliações e as melhores avaliações de cada modalidade (leaderboard por modalidade).