

Diagrama UML:

Da mesma forma, todos os funcionários do ginásio também precisarão de ter os seus dados pessoais, conforme descrito anteriormente, armazenados. Na verdade, é essencial para qualquer estabelecimento comercial guardar registo de todos os seus trabalhadores e, também, de todos os seus respetivos ofícios. Além disso, será necessário guardar informações relacionadas à sua contratação, como salários, data de contratação, tempo de contrato, horário e nível de certificação académica. De facto, o seu salário depende da atividade que exerce.

Dentro da categoria de funcionários, haverá subdivisões de acordo com as funções específicas que desempenham. Estas servirão, essencialmente, para diferenciar todas as profissões dos funcionários. Através destas, podemos consultar os seus salários e efetuar os pagamentos. Exemplos disso incluem treinadores, instrutores e nutricionistas.

Para a administração e manutenção da infraestrutura de cada ginásio precisamos de armazenar certos detalhes sobre cada um. Principalmente, será importante conhecer a sua morada. Dentro de cada ginásio, haverá diversos equipamentos, desde máquinas até halteres, e outros instrumentos dos quais necessitamos de saber o nome, as datas de compra, manutenção e expiração da garantia. Além disso, haverá salas específicas para a realização de determinadas aulas de grupo. Uma gestão eficaz das aulas de grupo permitirá que os membros do ginásio acessem facilmente informações sobre o tipo de aula, horário de início, duração, nome da aula e o instrutor responsável.

No ginásio, diversos tipos de adesão estarão disponíveis para os membros, com variações de preço e duração, bem como atividades a que essas adesões dão direito. Não só estas informações são vitais para o controlo e gestão de membros do ginásio, mas, também, são o que permite efetuar os devidos pagamentos das mensalidades. Essas atividades ou planos incluirão os planos de treino elaborados pelos treinadores, as aulas de grupo ministradas pelos instrutores e os planos de alimentação criados pelos nutricionistas. Finalmente, o ginásio também de um determinado número de cacifos, dos quais, todos os membros deste podem usufruir livremente.

B – Relational Schema – Implementação Inicial sem IA

Pessoa (nif, primeiroNome, ultimoNome, dataNascimento, idade, genero, email, numeroTelemovel, morada, contactoEmergencia)

Membros (membroID, dataInicioAdesao, dataFimAdesao, precoMensalidade, planoMensalidade, planoAlimentar, planoTreino, nif-> Pessoa)

Nutricionistas (nutricionistaID, nivelInstrucao)

Instrutores (instrutorID, nivelInstrucao)

TreinadoresPessoais (treinadorPessoalID, nivelInstrucao)

Funcionarios (funcionarioID, posicao, salario, horario, nif->Pessoa)

Equipamentos (equipamentoID, nomeEquipamento, dataCompra, dataGarantia, dataManutencao)

Ginásio (ginasioID, morada, cidade, estado, pais, telefone, email, salaID)

AulasGrupo (aulaGrupoID, instrutorID->Instrutores, nomeAula, descricao, horaInicio, horaFim, duracao, salaID->Ginasio)

Inscricoes (membroID-> Membros, ginasioID-> Ginasio)

FuncionáriosMembros (membroID -> Membros, funcionarioID ->Funcionários)

EquipmantosMembros (membroID -> Membros, equipamentoID-> Equipamentos)

GinasioFuncionários (ginasioID -> Ginasio, funcionarioID-> Funcionarios)

Relational Schema - Implementação final com IA

Pessoa (nif, primeiroNome, ultimoNome, dataNascimento, idade, genero, email, numeroTelemovel, morada, contactoEmergencia)

Membros (membroID, dataInicioAdesao, dataFimAdesao, precoMensalidade, planoMensalidade, planoAlimentar, planoTreino, nif-> Pessoa)

Funcionarios (funcionarioID, posicao, salario, horario, nif->Pessoa)

Equipamentos (equipamentoID, nomeEquipamento, dataCompra, dataGarantia, dataManutencao)

Ginasio (ginasioID, morada, cidade, estado, pais, telefone, email, salaID, numeroCacifos)

AulasGrupo (aulaGrupoID, instrutorID->Instrutores, nomeAula, descricao, horaInicio, horaFim, duracao, salaID->Ginasio)

Inscricoes (membroID-> Membros, ginasioID-> Ginasio)

FuncionáriosMembros (membroID -> Membros, funcionarioID ->Funcionários)

EquipmantosMembros (membroID -> Membros, equipamentoID-> Equipamentos)

GinasioFuncionários (ginasioID -> Ginasio, funcionarioID-> Funcionarios)

AulasGrupoGinasioFuncionarios (aulaGrupoID ->AulasGrupo, ginasioID-> Ginasio, funcionarioID -> Funcionarios)

Análise de Dependências Funcionais e Formas Normais

Pessoa (nif, primeiroNome, ultimoNome, dataNascimento, idade, genero, email, numeroTelemovel, morada, contactoEmergencia)

- **FDs:** nif -> primeiroNome, ultimoNome, dataNascimento, idade, genero, email, numeroTelemovel, morada, contactoEmergencia
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** Todos os atributos não primos são totalmente dependentes funcionalmente da superchave nif. Nenhum atributo não primo depende de um subconjunto adequado da chave.

Membros (membroID, dataInicioAdesao, dataFimAdesao, precoMensalidade, planoMensalidade, planoAlimentar, planoTreino, nif-> Pessoa)

- **FDs:** membroID -> dataInicioAdesao, dataFimAdesao, precoMensalidade, planoMensalidade, planoAlimentar, planoTreino, nif
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** A chave membroID determina de forma única todos os outros atributos. Nenhum atributo não primo depende de um subconjunto adequado da chave.

Funcionarios (funcionarioID, posicao, salario, horario, nif->Pessoa)

- **FDs:** funcionarioID -> posicao, salario, horario, nif
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** A chave funcionarioID determina de forma única todos os outros atributos. Nenhum atributo não primo depende de um subconjunto adequado da chave.

Equipamentos (equipamentoID, nomeEquipamento, dataCompra, dataGarantia, dataManutencao)

- **FDs:** equipamentoID -> nomeEquipamento, dataCompra, dataGarantia, dataManutencao
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** A chave equipamentoID determina de forma única todos os outros atributos. Nenhum atributo não primo depende de um subconjunto adequado da chave.

Ginasio (ginasioID, morada, cidade, estado, pais, telefone, email, salaID, numeroCacifos)

- **FDs:** ginasioID -> morada, cidade, estado, pais, telefone, email, salaID, numeroCacifos
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** A chave ginasioID determina de forma única todos os outros atributos. Nenhum atributo não primo depende de um subconjunto adequado da chave.

AulasGrupo (aulaGrupoID, instrutorID->Instrutores, nomeAula, descricao, horaInicio, horaFim, duracao, salaID->Ginasio)

- **FDs:** aulaGrupoID -> instrutorID, nomeAula, descricao, horaInicio, horaFim, duracao, salaID
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** A chave aulaGrupoID determina de forma única todos os outros atributos. Nenhum atributo não primo depende de um subconjunto adequado da chave.

Inscricoes (membroID-> Membros, ginasioID-> Ginasio)

- **FDs:** Não existem (Nenhuma FD foi fornecida)
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** Como não há FDs não triviais fornecidas, automaticamente satisfaz BCNF e 3NF.

FuncionáriosMembros (membroID -> Membros, funcionarioID ->Funcionários)

- **FDs:** Não existem
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** Como não há FDs não triviais fornecidas, automaticamente satisfaz BCNF e 3NF.

EquipmentosMembros (membroID -> Membros, equipamentoID-> Equipamentos)

- **FDs:** Não existem
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** Como não há FDs não triviais fornecidas, automaticamente satisfaz BCNF e 3NF.

GinasioFuncionários (ginasioID -> Ginasio, funcionarioID-> Funcionarios)

- **FDs:** Não existem
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** Como não há FDs não triviais fornecidas, automaticamente satisfaz BCNF e 3NF.

AulasGrupoGinasioFuncionarios (aulaGrupoID ->AulasGrupo, ginasioID->Ginasio, funcionarioID -> Funcionarios)

- **FDs:** Não existem (Nenhuma FD foi fornecida)
- **BCNF:** Sim
- **3NF:** Sim
- **Justificação:** Como não há FDs não triviais fornecidas, automaticamente satisfaz BCNF e 3NF.

Em cada caso, as relações estão em BCNF e 3NF porque satisfazem as condições dessas formas normais. Cada relação tem uma chave primária (ou superchave) clara, e os atributos não primos são totalmente dependentes funcionalmente da chave. Além disso, não há dependências transitivas.

Utilização de Inteligência Artificial

Na segunda fase do projeto, a inteligência artificial, representada pelo Chat GPT, demonstrou-se significativamente mais útil e pertinente em comparação com a primeira fase. Sua contribuição foi crucial para as alterações e adições realizadas no projeto.

Inicialmente, o Chat GPT ofereceu insights valiosos ao sugerir a eliminação de classes consideradas "inúteis". Em particular, a proposta de eliminar a classe "Cacifos" e incorporar seus dados na classe "Ginásio" simplificou o processo de população do banco de dados SQLite. Além disso, recomendou a eliminação das classes "TreinadoresPessoais", "Nutricionistas" e "Instrutores", propondo a inclusão desses dados em uma classe que abrangesse todos os funcionários. Essas mudanças não apenas reduziram o ruído no diagrama, mas também eliminaram algumas classes redundantes.

A inteligência artificial também desempenhou um papel crucial na revisão do esquema relacional inicial proposto pelo grupo. A sugestão de ajustes na maneira como lidamos com as relações entre as classes, especialmente a relação múltipla entre "Ginásio", "Funcionários" e "AulasGrupo", foi particularmente valiosa.

No processo de justificação das dependências funcionais, a parte mais desafiadora do projeto para o grupo, o Chat GPT foi consultado para esclarecimentos. Sua assistência foi fundamental para compreender que as relações entre as classes não tinham dependências funcionais e que, posteriormente, estavam na Terceira Forma Normal e na Forma Normal de Boyce-Codd.

Embora tenha havido uma diminuição na relevância da inteligência artificial na fase final do projeto, ela ainda ofereceu sugestões úteis, incluindo a recomendação para alterar alguns tipos de dados, como a mudança dos números de telefone de "TEXT" para "INT". No encerramento do projeto, o Chat GPT foi novamente utilizado para gerar mais valores, contribuindo para a população adicional do banco de dados SQLite.

