

# Base de dados de uma faculdade

## Parte III

Miguel Norberto Costa Freitas - up201906159  
Joana Teixeira Mesquita - up201907878  
Diogo Miguel Chaves dos Santos Antunes Pereira – up201906422

Mestrado Integrado em Engenharia Informática e de Computação  
Bases de dados  
Turma 5 - Grupo 1

20 de maio de 2021

## Contents

<b>Contexto:</b> .....	3
<b>UML:</b> .....	5
<b>Esquema Relacional:</b> .....	6
<b>Dependências funcionais:</b> .....	8
<b>Restrições:</b> .....	10
<b>Interrogações:</b> .....	11
<b>Gatilhos:</b> .....	11

## Contexto:

Pretende-se armazenar informação relativa ao funcionamento de uma faculdade.

Esta base de dados deverá armazenar os seus Estudantes, sobre os quais interessa saber o seu nome, NIF, id, morada e telefone. Cada Estudante frequenta pelo menos uma Unidade Curricular que é caracterizada por um nome, a nota mínima para obter aprovação e a sua taxa de reprovação. Cada Unidade Curricular tem ainda associado um Docente Regente da UC e é lecionada ainda em um ou mais Cursos, caracterizados por um nome, uma nota mínima de ingresso e o número de vagas.

Ao frequentar uma Unidade Curricular cada estudante realiza pelo menos uma atividade de Avaliação, caracterizada sempre pela data em que ocorre, pela sua nota final e pelo peso de dita componente na avaliação final da UC.

Existem três tipos de Avaliação: Testes e Exames Finais, que têm como parâmetros adicionais a duração que os Estudantes têm para os realizar e as horas a que se realizam, bem como Trabalhos que possuem um prazo de entrega.

Cada Estudante está inscrito num ou mais Cursos e pertence sempre a uma ou mais Turmas, da qual interessa saber o ano e o número. Cada Turma tem a si associada variados docentes, vários Blocos Horários e uma Unidade Curricular.

Um Bloco Horário é caracterizado por uma hora de início, uma hora de fim, um dia da semana e um tipo de aula, bem como uma sala onde a aula se realiza.

Existe o Staff, que é constituído por Docentes e Não Docentes, sendo caracterizado pelo nome, NIF e salário. Do Docente interessa também conhecer a sua especialização, se é diretor de algum Curso e que UCs é que leciona. Acerca de um Não Docente interessa ainda saber a sua duração de contrato.

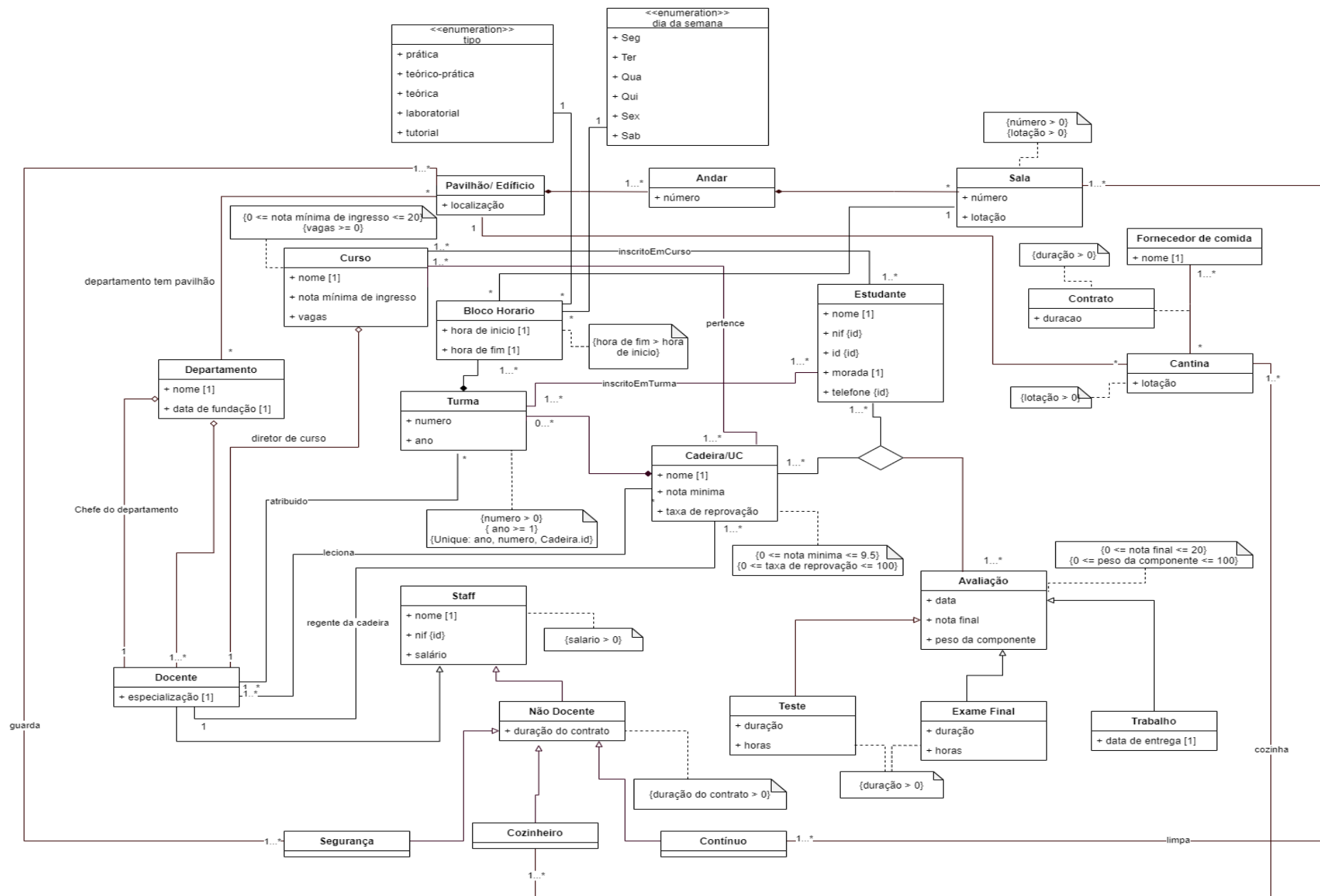
Existem ainda três tipos de Não Docentes, Seguranças, que têm associado um ou mais Edifícios para guardar, Cozinheiros, associados a uma ou mais Cantinas onde cozinham e ainda Contínuos, que se associam a uma ou mais Salas que limpam.

Existem ainda Departamentos dentro da faculdade, acerca dos quais interessa saber o seu nome e data de fundação. Cada Docente pertencerá a um único Departamento e interessa ainda saber qual Docente é o chefe de cada Departamento. Um Departamento pode localizar-se num ou mais

Edifícios que têm uma certa localização, constituído por vários Andares numerados, que por sua vez podem possuir várias Salas também estas numeradas e com lotação máxima.

Por fim, a faculdade pode ter várias Cantinas, acerca das quais interessa saber a sua lotação máxima e o Edifício no qual se localizam. Interessa ainda armazenar informação acerca dos Fornecedores de Comida de cada Cantina, que possuem um nome e um Contrato com uma certa duração, com esta.

## UML:



## Esquema Relacional:

Staff(idStaff , nome, nif, salario)

Docente(idDocente->Staff, idDepartamento->Departamento, especialização, nif, nome, salario)

NaoDocente(idNaoDocente->Staff, duracaoDoContrato, nif, nome, salario)

Segurança(idSgurança->NaoDocente, duracaoDoContrato, nif, nome, salario)

Cozinheiro(idCozinheiro->NaoDocente, duracaoDoContrato, nif, nome, salario)

Continuo(idContinuo->NaoDocente, duracaoDoContrato, nif, nome, salario)

Cantina(idCantina, idPavilhao->Pavilhao, lotacao)

Estudante(idEstudante, nome, morada, nif, telefone)

FornecedorDeComida(idFornecedor, nome)

Cadeira(idCadeira, regente->Docente, nome, notaMinima, taxaReprovacao)

Avaliacao(idAvaliacao, data, notaFinal, pesoComponente)

Teste(idTeste -> Avaliacao, duração, hora, data, notaFinal, pesoComponente)

ExameFinal(idExame -> Avaliacao, duração, hora, data, notaFinal, pesoComponente)

Trabalho(idTrabalho -> Avaliacao, dataEntrega, data, notaFinal, pesoComponente)

Turma(idTurma, idCadeira->Cadeira, numero, ano)

BlocoHorario(idBlocoHorario, idTurma -> Turma, idSala -> Sala, horaDeInicio, horaDeFim, diaDaSemana, tipo)

Curso(idCurso, diretor->Docente, nome, notaMinimaDeIngresso, vagas)

Pavilhao(idPavilhao, localizacao)

Andar(idAndar, idPavilhao->Pavilhao, numero)

Sala(idSala, idAndar->Andar, numero, lotacao)

Departamento(idDepartamento, chefeDeDepartamento->Docente, nome, dataFundacao)

ContratoFornecerComida(idCantina->Cantina, idFornecedor->FornecedorDeComida, duracaoContrato)

CozinhaEm(idCozinheiro->Cozinheiro, idCantina->Cantina)

InstanciaDeAvaliacao(idEstudante->Estudante, idCadeira->Cadeira, idAvaliacao->Avaliacao)

InscritoEmTurma(idEstudante->Estudante, idTurma ->Turma)

InscritoEmCurso(idEstudante->Estudante, idCurso -> Curso)

Leciona(idDocente ->Docente, idCadeira->Cadeira)

PertenceA(idCadeira ->Cadeira, idCurso->Curso)

AtribuidoA(idDocente ->Docente, idTurma ->Turma)

Limpa(idContinuo->Continuo, idSala->Sala)

Guarda(idSeguranca->Segurança, idPavilhao->Pavilhao)

DepartamentoTemPavilhao(idPavilhao -> Pavilhao, idDepartamento -> Departamento)

## Dependências funcionais:

idStaff -> nome, nif, salario

idDocente -> idDepartamento, especialização, nome, nif, salario

idNaoDocente -> duracaoDoContrato, nome, nif, salario

idSeguranca -> duracaoDoContrato, nome, nif, salario

idCozinheiro -> duracaoDoContrato, nome, nif, salario

idContinuo -> duracaoDoContrato, nome, nif, salario

idCantina -> idPavilhao, lotacao

idEstudante -> nome, nif, morada, telefone

idFornecedor -> nome

idCadeira -> regente, nome, notaMinima, taxaReprovacao

idAvaliacao -> data, notaFinal, pesoComponente

idTeste -> duracao, data, hora, notaFinal, pesoComponente

idExame -> duracao, data, hora, notaFinal, pesoComponente

idTrabalho -> dataEntrega, data, notaFinal, pesoComponente

idTurma -> idCadeira, numero, ano

idBlocoHorario -> idTurma, idSala, horaDeInicio, horaDeFim, diaDaSemana, tipo

idCurso -> diretor, nome, notaMinimaDeIngresso, vagas

idPavilhao -> localizacao

idAndar -> idPavilhao, numero

idSala -> idAndar, numero, lotacao

idDepartamento -> chefeDeDepartamento, nome, dataFundacao

idCantina, idFornecedor -> duracaoContrato

As nossas dependências funcionais não violam nem a 3ª forma normal nem a forma normal de Boyce-Cod. Para explicar a razão, como cada forma normal tem como regra que a forma normal anterior não está a ser violada, começemos por falar da 1ª forma normal. A 1ª forma normal, dita que cada atributo só pode receber valores atômicos, ou seja, no contexto da nossa base de dados, se um estudante pode estar em várias turmas, a turma não deve ser um atributo do estudante com vários valores, mas uma relação entre turma e estudante (no nosso caso temos a relação InscritoEm) que guarda o id do estudante que está inscrito bem como o da turma, não quebrando assim esta forma normal. A 2ª forma normal exige que a 1ª



forma normal seja cumprida, bem como, que a chave primária consista em apenas um atributo e, caso contrário, que todos os atributos dependam de todas as chaves primárias da relação e não de nenhum subconjunto destas. Na nossa base de dados temos a relação ContratoForneceComida que possui duas chaves primárias que são necessárias para aceder ao único outro atributo da relação (duração), nas restantes relações com mais do que uma chave primária as relações não têm outros atributos aos quais aceder, por isso também respeitam a 2ª forma normal. A 3ª forma normal, para além de necessitar que a 2ª forma normal seja cumprida, dita que nenhum atributo que não seja chave não pode depender de nenhum outro atributo que também não seja chave, algo que não acontece no nosso modelo relacional, visto que, foi criado de modo a que os atributos de cada tabela dependessem apenas da chave primária desta. Por fim, a forma normal de Boyce-Cod é uma extensão da 3ª forma normal que diz que numa relação  $A \rightarrow B$  ou  $A$  é uma chave, ou  $A \rightarrow B$  é trivial, ou seja,  $B$  é um subconjunto de  $A$ , no entanto no nosso modelo relacional todas as relações são dependentes de uma chave.

## Restrições:

Um Pavilhão tem de ter uma localização.

A lotação da Cantina tem de ser maior que zero.

Um Fornecedor de comida tem sempre um nome.

Um Estudante tem sempre um nome e uma morada. O NIF e o número de telefone são característicos de cada Estudante.

Um membro de Staff tem sempre um nome, um NIF único e tem um salário não nulo.

Um Docente tem de ter uma especialização.

Um Não Docente tem de ter uma duração de contrato positiva.

Uma Unidade Curricular tem de ter sempre um nome, uma nota mínima para aprovação entre 0 e 9.5 e uma taxa de reprovação entre 0% e 100%.

Um Curso tem sempre um nome, uma nota mínima de ingresso entre 0 e 20 e um número de vagas positivo.

Uma Avaliação tem de ter sempre uma nota final entre 0 e 20 e um peso da componente entre 0% e 100%, assim como uma data válida.

Um Teste e um Exame Final têm de ter uma duração e uma hora válida (não negativa ou nula).

Um Trabalho tem de ter obrigatoriamente uma data de entrega.

Um Bloco Horário tem de ter obrigatoriamente uma hora de início e uma hora de fim (sendo que a hora de fim tem de ser posterior à hora de início), um tipo, que pode ser uma de cinco opções: prática, teórico-prática, teórica, laboratorial e tutorial e o dia da semana pode ser um dia de segunda a sábado representado pelas primeiras três letras do dia da semana (exemplo: segunda -> Seg).

Uma Turma tem de ter um número e ano positivos, e não pode haver duas Turmas com o mesmo ano, número e Unidade Curricular simultaneamente.

Uma Sala tem de ter um número e uma lotação positiva.

Um Departamento tem de ter sempre um nome e uma data de fundação.

A duração de um contrato entre um Fornecedor de Comida e uma Cantina tem de ser positiva e não nula.

## Interrogações:

1. Os horários completos do estudante com o nome “Diogo Pereira”.
2. Todas as aulas práticas a ter lugar na segunda-feira.
3. O nome dos estudantes que tiveram as notas mais altas a uma cadeira e a respetiva cadeira.
4. O número de salas a cargo de cada contínuo.
5. O número de turmas em que cada estudante está inscrito.
6. O nome e salário de todos os funcionários cujo nome começa por P.
7. A média do número de testes por aluno.
8. A média das notas de todos os estudantes contando apenas avaliações em que tiveram aprovação.
9. O nome de todos os professores que dão aulas a cadeiras do curso de Mestrado Integrado em Engenharia Informática e de Computação.
10. A informação de todos os departamentos fundados no mês de Setembro à 21 anos ou mais.

## Gatilhos:

1. Diminui o número de vagas no curso quando um estudante é inscrito, aumenta-as quando um estudante deixa de ser parte de um curso e impede a inscrição do estudante se não houver vagas.
2. Impede blocos horários coincidentes (que decorrem ao mesmo tempo) de terem a si associados a mesma sala.
3. Impede um estudante de se inscrever em turmas de cadeiras que não pertençam a um curso em que está inscrito.