



RELATÓRIO FINAL

BASES DE DADOS

MAIO DE 2021

HENRIQUE NUNES, UP201906852

MATEUS SILVA, UP201906232

MELISSA SILVA, UP201905076

ÍNDICE

CAPA.....	1
ÍNDICE.....	2
INTRODUÇÃO.....	3
CONTEXTO.....	4
Estudantes	4
Docentes	4
Não-Docentes.....	4
Classificações (Disciplinas, Avaliações e Estudantes)	5
Quartos e Blocos	5
Clubes	5
Turmas	5
DIAGRAMA	6
MODELO RELACIONAL.....	7
Dependências Funcionais	7
Análise de Dependências Funcionais e Formas Normais	9
Restrições.....	11
Restrições Não Implementadas.....	13
INTERROGAÇÕES.....	14
GATILHOS.....	15
Primeiro Gatilho	15
Segundo Gatilho.....	15
Terceiro Gatilho	15

INTRODUÇÃO

No âmbito da disciplina *Base de Dados*, também conhecida por *BDAD*, do segundo semestre do segundo ano do nosso curso, foi-nos pedida a conceção de um contexto original para a criação de uma base de dados. Esta foi considerada a primeira etapa da componente *Projeto* da unidade curricular, sendo esperado de nós o diagrama UML correspondente ao contexto por nós elaborado e, também, um relatório inicial de acompanhamento que explicasse o conceito criado.

O diagrama mencionado foi elaborado na aplicação *LucidChart*, cuja implementação no contexto das aplicações da *Google* nos permitiu trabalhar conjuntamente em prol do nosso objetivo: modelar uma base de dados de forma compreensível e fazendo jus às nossas ideias.

O presente relatório contextualizará então a nossa ideia final para o projeto, escolhida por nós a partir de duas ideias distintas que verificámos fornecer suficiente espaço de desenvolvimento para a concretização do projeto na sua vertente pedagógica e também pessoal — visto termos percebido que não nos faltavam ideias para qualquer uma delas.

Tomamos a liberdade de nomear o nosso projeto *O Colégio Yamazaki*, seguindo uma ideia de um dos nossos membros, com origem num projeto pessoal¹ seu. Cumprimos o requerimento de o contexto conter pelo menos dez classes diferentes.

Para esta segunda entrega, além da revisão do contexto e respetivo diagrama UML, eram esperados ainda a tradução do nosso contexto para um modelo relacional, além da sua implementação em SQL usando *SQLite*.

¹ Ver REFERÊNCIAS.

CONTEXTO

Um colégio privado deseja automatizar a maioria dos seus processos administrativos, tendo definido algumas regras para a modelação de uma base de dados que lhes sirva para esse propósito.

Há três tipos de **pessoas** que frequentam o colégio: **estudantes**, **docentes** e **não-docentes**. Todos estes são caracterizados por um NIF (número de identificação fiscal), género (masculino, feminino ou não-binário), nome completo e data de nascimento.

ESTUDANTES

O que distingue os estudantes são o seu ano escolar (que vai desde o 7º ao 12º ano) e um número *único* de estudante. Cada estudante possui ainda um conjunto de no mínimo 6 **disciplinas** a frequentar de momento, cada uma com zero ou mais **avaliações** associadas. Além disto, um estudante pode ser *interno* (vivendo no colégio) ou não, pelo que se sim, ter-lhe-á associado um **quarto** individual que poderá trocar de período a período em ocasiões especiais.

Cada estudante pertence a uma **turma**.

Finalmente, um estudante pode inscrever-se em **Clubes**, sem nenhuma obrigação de o fazer. Da mesma forma, poderá ser **responsável** de tantos clubes quanto deseje.

DOCENTES

Os docentes diferenciam-se, além dos seus atributos gerais partilhados com outras entidades do colégio, pelo *número de turmas* — um professor tem de ter pelo menos uma **turma** e, no máximo, cinco —, *número de disciplinas* que lecionam, que é sempre maior ou igual a um.

Existem ainda dois valores de *avaliação* para ajudar os docentes a melhorar os seus métodos: *pedagógica* e *estudantil*.

A avaliação pedagógica é uma avaliação atribuída a professores por professores, sendo feita de quatro em quatro anos numa escala de 1 a 10. O valor guardado é sempre o mais recente. A avaliação estudantil é atribuída pelos estudantes que o professor tem como alunos, também existente numa escala de 1 a 10. O valor guardado é, mais uma vez, sempre o mais recente.

Finalmente, um professor pode responsabilizar-se por um **Clube** estudantil do colégio, ajudando na sua manutenção e supervisão. No máximo, pode responsabilizar-se por dois, por recomendação/regra do Colégio.

NÃO-DOCENTES

Estas entidades são discerníveis por três características: *data de contratação*, *tipo de serviço* prestado e o *turno* em que trabalham.

O tipo de um Não-Docente é sempre um dos seguintes:

- *Administrativo*: secretária/o nos Serviços Académicos do Colégio;
- *Auxiliar de Limpeza*;
- *Auxiliar de Gestão*: trabalha em algum dos serviços do Colégio, nomeadamente aqueles fornecidos aos alunos (ex.: Bar dos Estudantes);

Em especial, os auxiliares de limpeza estão associados a um **quarto** que são responsáveis de limpar e vigiar concomitantemente à duração do seu turno. O valor

de turno de qualquer não-docente restringe-se a um valor de 1 ou 2, referentes, respetivamente, ao primeiro ou segundo turno do dia.²

Turnos podem ser mudados entre não docentes do mesmo tipo sem grandes burocracias, desde que todos os turnos diários continuem preenchidos e assegurados.

CLASSIFICAÇÕES (DISCIPLINAS, AVALIAÇÕES E ESTUDANTES)

Cada **disciplina** tem um nome, que geralmente poderá distinguir o ano de escolaridade a que está associado. Cada uma destas possui zero ou mais **avaliações**³, caracterizada pela nota (de 0 a 20), *ano letivo* de atribuição e se a disciplina foi *completa* ou não, o que é conseguido com uma nota igual ou maior que 10.

QUARTOS E BLOCOS

Cada andar do edifício do colégio no intervalo do décimo ao vigésimo andar possui três blocos de 10 quartos cada um. Um **quarto** é caracterizado pelos seus *auxiliares de limpeza* responsáveis, *estudante detentor* atual e *número único* de três dígitos, estando inserido num dado **bloco**.

O *número* do **bloco** é identificado no *código único* de cada **quarto**, cujos não-docentes associados não podem partilhar o turno. Por outras palavras, cada **quarto** tem dois auxiliares de limpeza não-docentes de turnos diferentes.

O código de um quarto divide-se em três algarismos:

- o primeiro algarismo vai de 0 a 9, onde 0 corresponde ao décimo andar do colégio e 9 ao décimo nono andar;
- o segundo algarismo vai de 0 a 2, onde 0 corresponde ao primeiro bloco de um dado andar do colégio e 2 ao terceiro;
- o terceiro e último algarismo vai de 0 a 9, onde 0 corresponde ao quarto 1 de um dado bloco.

Portanto, 000 indica o primeiro quarto do primeiro bloco do décimo andar do colégio, por exemplo.

Sem **blocos**, não há **quartos**.

CLUBES

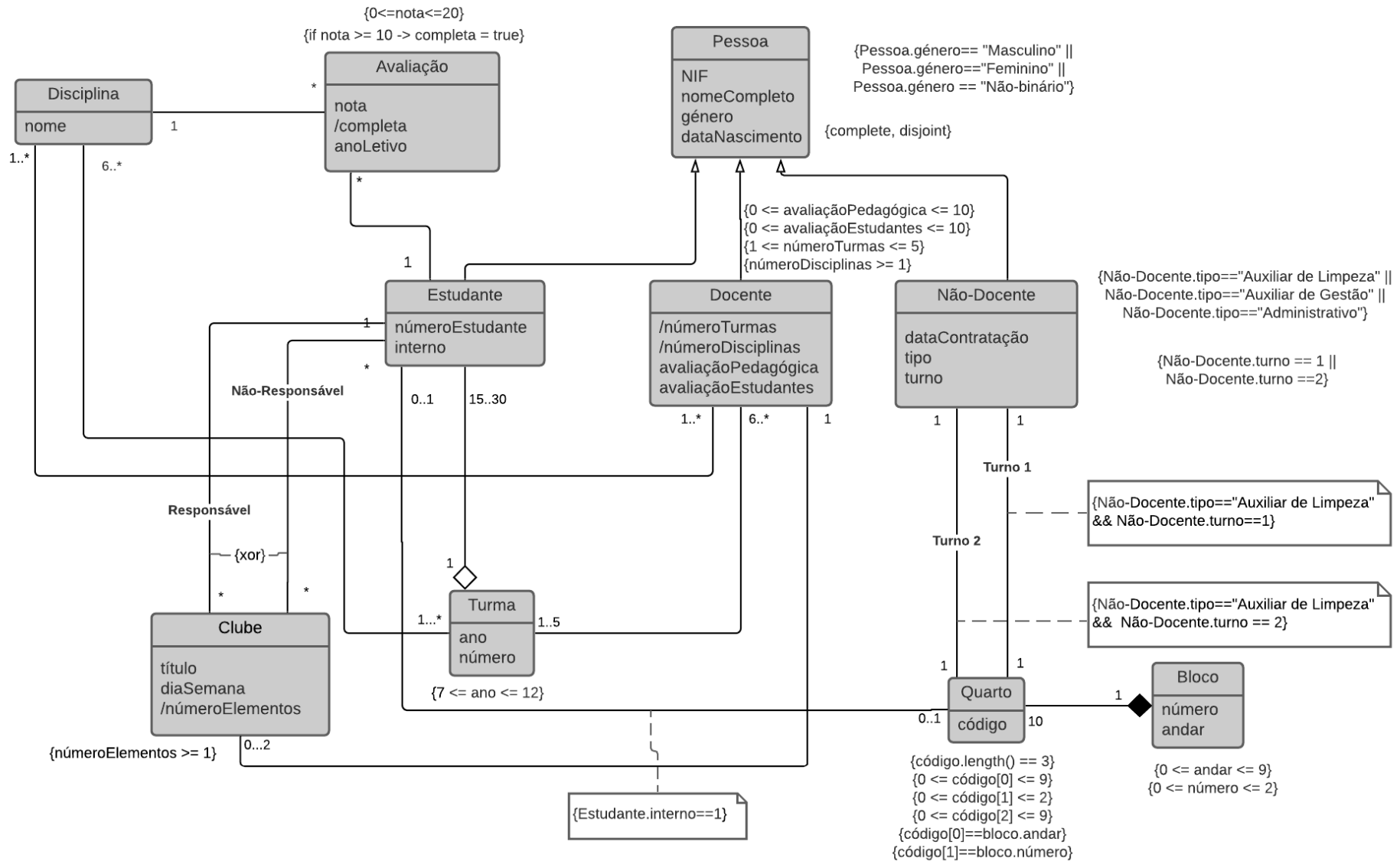
Cada clube possui pelo menos um elemento. Com tal condição garantida, um clube é caracterizado por um *título, dia da semana* em que faz reuniões e um número de elementos. Lembrar que cada clube possui um **estudante responsável** e um **docente** responsável.

TURMAS

Cada turma tem pelo menos 15 **estudantes** e no máximo 30, além de possuir pelo menos 6 **docentes** associados.

² Os turnos não são criados de forma igual, contudo: Auxiliares de Limpeza possuem turnos de Dia e Noite, enquanto Administrativos possuem turnos de Manhã e Tarde, devido aos Serviços Académicos fecharem durante a noite.

³ Em contexto de reprovação, a avaliação repete-se no ano letivo seguinte.



MODELO RELACIONAL

O nosso modelo relacional, feito a partir do diagrama UML e contexto acima, é o seguinte.

Estudante(NIF, nomeCompleto, género, dataNascimento, númeroEstudante, interno, turma->Turma)
{númeroEstudante} UK
Docente(NIF, nomeCompleto, género, dataNascimento, númeroTurmas, númeroDisciplinas, avaliaçãoPedagógica, avaliaçãoEstudantes)
NãoDocente(NIF, nomeCompleto, género, dataNascimento, dataContratação, tipo, turno)
Bloco(número, andar)
Quarto(código, bloco_n->Bloco.número, bloco_a->Bloco.andar, turnoDia->NãoDocente, turnoNoite->NãoDocente, estudante->Estudante)
{turnoDia} UK
{turnoNoite} UK
{estudante} UK
Turma(ID, ano, número)
{ano, número} UK
Avaliação(ID, nota, completa, anoLetivo, aluno->Estudante, disciplina->Disciplina)
{anoLetivo, aluno, disciplina} UK
Disciplina(ID, nome)
{nome} UK
Clube(título, diaSemana, númeroElementos, docente->Docente, responsável->Estudante)

NãoResponsávelClube(clube->Clube, estudante->Estudante)
DocenteTurma(docente->Docente, turma->Turma)
DisciplinaDocente(disciplina->Disciplina, docente->Docente)
DisciplinaTurma(disciplina->Disciplina, turma->Turma)

DEPENDÊNCIAS FUNCIONAIS

As dependências funcionais encontradas no nosso modelo relacional são as seguintes. De notar que todas as relações que estão na forma trivial estão na Forma Normal de Boyce-Codd.

Estudante

- NIF → nomeCompleto, género, dataNascimento, númeroEstudante, interno, turma
O número de identificação fiscal de qualquer pessoa é único, pelo que permite distinguir qualquer entidade do colégio.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- númeroEstudante → NIF, nomeCompleto, género, dataNascimento, idade, interno, turma
Devido ao número de estudante ter sido concebido como único, também é possível identificar um estudante apenas por ele.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.

Docente

- NIF → nomeCompleto, género, dataNascimento, númeroTurmas, númeroDisciplinas, avaliaçãoPedagógica, avaliaçãoEstudantes
O número de identificação fiscal de qualquer pessoa é único, pelo que permite distinguir qualquer entidade do colégio.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.

NãoDocente

- NIF → nomeCompleto, género, dataNascimento, dataContratação, tipo, turno
O número de identificação fiscal de qualquer pessoa é único, pelo que permite distinguir qualquer entidade do colégio.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.

Bloco

Sem dependências funcionais não triviais.

Quarto

- código → bloco_n, bloco_a, turnoDia, turnoNoite, estudante
O código de um quarto permite determinar os números de identificação do seu quarto, os não-docentes responsáveis pelos turnos diurno e noturno e o estudante detetor.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- turnoDia → código, bloco_n, bloco_a, turnoNoite, estudante
O auxiliar de limpeza associado ao turno diurno de um quarto é suficiente para o identificar, já que cada auxiliar de limpeza apenas faz turnos diurnos num único quarto.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- turnoNoite → código, bloco_n, bloco_a, turnoDia, estudante
O auxiliar de limpeza associado ao turno noturno de um quarto é suficiente para o identificar, já que cada auxiliar de limpeza apenas faz turnos noturnos num único quarto.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- estudante → código, bloco_n, bloco_a, turnoDia, turnoNoite
Como cada estudante apenas possui um quarto, o estudante associado a um quarto basta para o identificar totalmente.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.

Turma

- ID → ano, número
O número de identificação de uma turma permite identificar o seu ano e número.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- ano, número → ID
O ano e número de uma turma permite identificar o seu ID, já que cada combinação destes é única para cada turma.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.

Avaliação

- ID → nota, anoLetivo, aluno, disciplina
O identificador de uma avaliação permite determinar todos os seus dados associados.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- anoLetivo, aluno, disciplina → ID, nota
Avaliações podem ser identificadas pela combinação do seu ano letivo, aluno e disciplina por serem únicas para cada ano letivo.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- nota → completa
Uma nota positiva permite determinar se uma disciplina foi completa.
Terceira Forma Normal: Viola.
Forma Normal de Boyce-Codd: Viola.

Disciplina

- ID → nome
O identificador de uma disciplina permite determinar o seu nome.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.
- nome → ID
O nome de uma disciplina é suficiente para obter o seu identificador.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.

Clube

- título → diaSemana, númeroElementos, docente, responsável
O título de um clube permite determinar todos os seus dados.
Terceira Forma Normal: Não viola.
Forma Normal de Boyce-Codd: Não viola.

NãoResponsávelClube

Sem dependências funcionais não triviais.

DocenteTurma

Sem dependências funcionais não triviais.

DisciplinaDocente

Sem dependências funcionais não triviais.

DisciplinaTurma

Sem dependências funcionais não triviais.

ANÁLISE DE DEPENDÊNCIAS FUNCIONAIS E FORMAS NORMAIS

Como aprendemos, o critério para que uma relação não viole a **Forma Normal de Boyce-Codd (BCNF)** é que, para todas as dependências funcionais não triviais $A \rightarrow B$, A seja uma *key* (isto é: A^+ contém todos os atributos da relação).

Para uma relação estar conforme a **Terceira Forma Normal (3NF)**, é necessário que, em cada dependência funcional não trivial $A \rightarrow B$, A seja uma *key* ou B seja composto apenas por atributos primos (ou seja, que pertençam a uma *key*). No entanto, se uma relação não violar a BCNF, então não violará 3NF, já que BCNF é um *subset* de 3NF: A será sempre uma *key*.

Todas as dependências identificadas, excetuando a dependência *nota* \rightarrow *completa*, em *Avaliação*, são capazes de determinar todos os atributos das respectivas tabelas partindo do seu lado esquerdo e, portanto, estão conforme as BCNF e 3NF. Desta forma, todas as tabelas estão conforme a BCNF e 3NF, com a exceção de *Avaliação*, onde existem violações.

Na primeira exceção mencionada, o elemento que se encontra no seu lado esquerdo (*nota*) não é uma *key*, constituindo assim uma violação à BCNF. Por outro lado, pelo mencionado anteriormente e por o lado direito da dependência não estar contido em nenhuma *key*, viola também a 3NF.

Para resolver este problema seria necessário decompor *Avaliação* da seguinte maneira:

Avaliação1(*nota*, *completa*)

Avaliação2(*nota*, *ID*, *anoLetivo*, *aluno*, *disciplina*)

Optámos por não o fazer devido a ser uma dependência funcional especialmente simples: a decomposição duplicaria as entradas de dados das notas em *Avaliação*, já que *completa* depende apenas de *nota* ser maior ou igual a 10 ou não.

RESTRIÇÕES

A tabela seguinte fornece detalhes acerca das restrições elaboradas para o nosso modelo relacional.

Relação	Atributo	Restrição	Implementação
Turma	ID	Identificador.	PRIMARY KEY
	ano	Todas as turmas têm de ter um ano que só pode ter valores entre 7 e 12. Não existem turmas com igual combinação ano + número.	NOT NULL CHECK (ano>=7 AND ano<=12) UNIQUE (ano, número)
	número	Todas as turmas têm de ter um número. Não existem turmas com igual combinação ano + número.	NOT NULL UNIQUE (ano, número)
Estudante	NIF	Serve como identificador.	PRIMARY KEY
	nomeCompleto	Todos os estudantes têm um nome.	NOT NULL
	género	Só pode ter um de quatro valores possíveis: "Masculino", "Feminino" ou "Não-Binário" ou NULL.	CHECK(género IN ('Masculino', 'Feminino', 'Não-Binário'))
	dataNascimento	Todos os estudantes têm uma data de nascimento.	NOT NULL
	númeroEstudante	Não existem estudantes com o mesmo númeroEstudante.	UNIQUE
	interno	Não tendo informação em contrário dada pelo Estudante, assumimos que este não é interno. Trata-se de um valor booleano (0 ou 1).	DEFAULT 0 CHECK(interno IN (0, 1))
	turma	Não se pode apagar uma turma que esteja associada a um Estudante. Atualizar os dados de uma turma implica a sua atualização em Estudante.	ON DELETE RESTRICT ON UPDATE CASCADE REFERENCES Turma
Docente	NIF	Serve como identificador.	PRIMARY KEY
	nomeCompleto	Todos os docentes têm um nome.	NOT NULL
	género	Este atributo só pode ter um de quatro valores possíveis: "Masculino", "Feminino" ou "Não-Binário" ou NULL.	CHECK(género IN ('Masculino', 'Feminino', 'Não-Binário'))
	dataNascimento	Todos os docentes têm uma data de nascimento.	NOT NULL
	númeroTurmas	Esse número encontra-se entre 1 e 5.	CHECK (númeroTurmas>=1 AND númeroTurmas<=5)
	númeroDisciplinas	Este número terá de ser maior que 1.	CHECK (númeroDisciplinas>=1)
	avaliaçãoPedagógica	O seu valor tem de estar entre 1 e 10. Caso não seja inserido nenhum valor, o valor será, por defeito, 5.	CHECK (avaliaçãoPedagógica>=1 AND avaliaçãoPedagógica<=10) DEFAULT 5
NãoDocente	NIF	Serve como identificador.	PRIMARY KEY
	nomeCompleto	Todos os não docentes têm um nome.	NOT NULL
	género	Este atributo só pode ter um de quatro valores possíveis: "Masculino", "Feminino" ou "Não-Binário" ou NULL.	CHECK(género IN ('Masculino', 'Feminino', 'Não-Binário'))
	dataNascimento	Todos os não docentes têm uma data de nascimento.	NOT NULL
	dataContratação	Todos os não docentes têm uma data de contratação. Não é possível contratar pessoas com menos de 18 anos.	NOT NULL CHECK(strftime('%Y', dataNascimento)+18< strftime('%Y', dataContratação))
	tipo	Este atributo só pode ter um de quatro valores possíveis: "Auxiliar de Limpeza" ou "Auxiliar de Gestão" ou "Administrativo" ou NULL.	CHECK(tipo IN ('Auxiliar de Limpeza', 'Auxiliar de Gestão', 'Administrativo'))
	turno	Este atributo só pode ter um de três valores possíveis: 1 ou 2 ou NULL. Nenhum "Auxiliar de Limpeza" pode ter turno com valor NULL.	CHECK (turno == 1 OR turno == 2) CHECK(turno IS NOT NULL OR tipo NOT LIKE 'Auxiliar de Limpeza')
Bloco	número	Todos os blocos têm um número. O valor deste encontra-se entre 0 e 2. Não existem blocos com igual combinação número + andar.	NOT NULL CHECK (número>=0 AND número<=2) PRIMARY KEY(número, andar)
	andar	Todos os blocos têm um andar. O valor deste encontra-se entre 0 e 9. Não existem blocos com igual combinação número + andar.	NOT NULL CHECK (andar>=0 AND andar<=9) PRIMARY KEY(número, andar)

Quarto	código	Cada Quarto tem um seu identificador. Este atributo deverá ter 3 dígitos, pelo que o central deverá estar entre 0 e 2.	PRIMARY KEY CHECK((LENGTH(código)=3) AND ((código % 100)/10 >= 0) AND ((código % 100)/10 < 3))
	bloco_n	Todos os quartos têm um bloco_n. O valor de bloco_n tem de ser igual ao valor do segundo dígito de código. Caso um Bloco associado a um Quarto seja apagado, o Quarto também é apagado. Atualizar um Bloco implica atualizar os Quartos associados.	NOT NULL ON DELETE CASCADE ON UPDATE CASCADE CHECK(bloco_n=((código % 100)/10) FOREIGN KEY(bloco_n,bloco_a) REFERENCES Bloco(número,andar)
	bloco_a	Todos os quartos têm um bloco_a. O valor de bloco_a tem de ser igual ao valor do primeiro dígito de código. Caso um Bloco associado a um Quarto seja apagado, o Quarto também é apagado. Atualizar um Bloco implica atualizar os Quartos associados.	NOT NULL ON DELETE CASCADE ON UPDATE CASCADE CHECK(bloco_a=((código - (código%100))) FOREIGN KEY(bloco_n,bloco_a) REFERENCES Bloco(número,andar)
	turnoDia	Apenas um NãoDocente pode ser associado a Quarto no atributo turnoDia ou turnoNoite. Não se podem apagar NãoDocentes associados a um Quarto nestes atributos. Atualizar um NãoDocente implica atualizar o Quarto associado. O NãoDocente associado a Quarto em turnoDia não pode ser igual ao NãoDocente associado a Quarto em turnoNoite. Cada quarto precisa de ter um funcionário associado para cada turno.	UNIQUE ON DELETE RESTRICT ON UPDATE CASCADE CHECK(turnoDia<>turnoNoite) NOT NULL REFERENCES NãoDocente
	turnoNoite		
	estudante	Apenas um Estudante pode ser associado a Quarto. Caso o Estudante associado seja eliminado, o atributo estudante ficará com o valor NULL. Caso a informação de um Estudante associado seja atualizada, a informação deste em Quarto também.	UNIQUE ON DELETE SET NULL ON UPDATE CASCADE REFERENCES Estudante
Disciplina	ID	Cada Disciplina tem um identificador.	PRIMARY KEY
	nome	Todas as disciplinas têm um nome e nenhum par de disciplinas têm o mesmo nome.	UNIQUE NOT NULL
Avaliação	ID	Cada Avaliação tem um identificador.	PRIMARY KEY
	nota	Todas as avaliações têm uma nota entre 0 e 20.	NOT NULL CHECK ((nota >= 0) AND (nota <= 20))
	completa	Elemento derivado de nota: caso nota seja superior ou igual a 10, completa = 1, caso contrário, completa = 0.	AS (nota >= 10)
	anoLetivo	Não existem avaliações com igual combinação anoLetivo + aluno + disciplina.	NOT NULL UNIQUE (anoLetivo, aluno, disciplina)
	aluno	Todas as avaliações têm um aluno e uma disciplina. Caso o Estudante ou a Disciplina associado/a a Avaliação seja eliminado/a, Avaliação também o será. Atualizações em qualquer um destes propagam-se a Avaliação. Não existem duas avaliações com igual combinação anoLetivo + aluno + disciplina.	NOT NULL ON DELETE CASCADE ON UPDATE CASCADE UNIQUE (anoLetivo, aluno, disciplina) (aluno) REFERENCES Estudante (disciplina) REFERENCES Disciplina
	disciplina		
Clube	título	Cada Clube tem o seu identificador.	PRIMARY KEY
	diaSemana	Pode ter um de oito valores possíveis: "Segunda" ou "Terça" ou "Quarta" ou "Quinta" ou "Sexta" ou "Sábado" ou "Domingo" ou NULL.	CHECK(diaSemana IN ('Segunda', 'Terça', 'Quarta', 'Quinta', 'Sexta', 'Sábado', 'Domingo'))
	númeroElementos	Este atributo terá de ter um valor igual ou superior a 1.	CHECK(númeroElementos >= 1)
	docente	Todos os clubes precisam de um docente e de um aluno responsável. Caso um Docente ou Estudante associado a Clube seja eliminado, Clube também o será. Caso a informação de um destes seja atualizada, tal propaga-se até Clube.	NOT NULL ON DELETE CASCADE ON UPDATE CASCADE (docente) REFERENCES Docente (responsável) REFERENCES Estudante
	responsável		
NãoResponsávelClube	clube	Todos os NãoResponsávelClube têm um clube e um estudante. Caso um Clube ou Estudante associado a NãoResponsávelClube num destes atributos seja eliminado, NãoResponsávelClube também o será. Caso a informação destes seja atualizada, a informação presente em NãoResponsávelClube também será.	PRIMARY KEY(clube, estudante) NOT NULL ON DELETE CASCADE ON UPDATE CASCADE (clube) REFERENCES Clube (estudante) REFERENCES Estudante
	estudante		

DocenteTurma	docente	Todos os DocenteTurma têm um docente e uma turma. Caso um Docente ou Turma associado/a a DocenteTurma seja eliminado/a, DocenteTurma também o será. Caso a informação de um destes seja atualizada, a informação deste(s) em DocenteTurma também será.	PRIMARY KEY(docente, turma) NOT NULL ON DELETE CASCADE ON UPDATE CASCADE (docente) REFERENCES Docente (turma) REFERENCES Turma
	turma		
DisciplinaDocente	disciplina	Todos os DisciplinaDocente têm uma disciplina e um docente. Caso uma Disciplina ou Docente associada/o a DisciplinaDocente seja eliminada/o, DisciplinaDocente também o será. Caso a informação de um destes seja atualizado, a informação em DisciplinaDocente também o será.	PRIMARY KEY(disciplina, docente) NOT NULL ON DELETE CASCADE ON UPDATE CASCADE (disciplina) REFERENCES Disciplina (docente) REFERENCES Docente
	docente		
DisciplinaTurma	disciplina	Todos os DisciplinaTurma têm uma disciplina e uma turma. Caso uma Disciplina ou Turma associada a DisciplinaTurma seja eliminada, DisciplinaTurma também o será. Caso a informação de uma destas seja atualizada, a informação em DisciplinaTurma também o será.	PRIMARY KEY(disciplina, turma) NOT NULL ON DELETE CASCADE ON UPDATE CASCADE (disciplina) REFERENCES Disciplina (turma) REFERENCES Turma
	turma		

RESTRIÇÕES NÃO IMPLEMENTADAS⁴

Relembramos que este texto é relativo à 2ª Entrega do trabalho, como especificamos no texto na altura não tínhamos conhecimentos de Vistas ou Gatilhos, algo que agora temos e assim conseguimos implementar algumas destas restrições.

- Em determinadas relações, estabelecemos um número específico para as multiplicidades, o qual não conseguimos confirmar nesta implementação, por exemplo: na relação entre Estudante e Turma, conseguimos implementar a multiplicidade de que Estudante necessita de ter uma turma, no entanto não conseguimos implementar que Turma tivesse entre 15 e 30 Estudantes.
- númeroDisciplinas — elemento derivado de multiplicidades entre Disciplina e Docente;
- númeroTurmas — elemento derivado de multiplicidades entre Turma e Docente;
- númeroElementos — elemento derivado de multiplicidades entre Turma e Estudante;
- Relação *xor* entre características Responsável e Não Responsável de Clube — criámos, contudo, a tabela NãoResponsávelClube;
- NãoDocentes associados à tabela Quarto serem de tipo == "Auxiliar de Limpeza", o que implicaria envolvimento de dados de tabelas distintas;
- Lógica por detrás do atributo interno de Estudante.⁵

⁴ Um antigo atributo idade de Estudante foi descartado; uma restrição referente a ele presente nesta secção aquando da 2ª Entrega foi, portanto, retirada.

⁵ Não listámos esta restrição na 2ª Entrega por gralha nossa. Listamo-la aqui para lhe dar o lugar que deveria ter tido então.

INTERROGAÇÕES

Como pedido, listamos agora o conjunto de dez interrogações feitas para a nossa base de dados.

1. Quais os números únicos e nomes de Estudantes que possuam Avaliações positivas a qualquer Disciplina? Inclua as notas e disciplinas em questão.
 - **Ficheiro:** int1.sql
2. Quais os Estudantes menores de idade?
 - **Ficheiro:** int2.sql
3. Quais os pares de Turmas que estão a ter aulas lecionadas por um mesmo Professor?
 - **Ficheiro:** int3.sql
4. Nomes dos Estudantes que frequentam o *Clube de Xadrez* e o *Clube de Luta*?
 - **Ficheiro:** int4.sql
5. A que Estudante, Não-Docentes e Bloco corresponde cada Quarto? Ordene os resultados pelos IDs dos Quartos.
 - **Ficheiro:** int5.sql
6. Liste, para todos os Estudantes: média, idade, número de Clubes que frequenta e número de Clubes que administra?
 - **Ficheiro:** int6.sql
7. Quais os nomes dos Professores com idade acima dos 40 anos que são responsáveis por pelo menos um Clube?
 - **Ficheiro:** int7.sql
8. Qual o aluno interno com a maior média ao conjunto de disciplinas de Português, Inglês, Francês (de 12º ano)?
 - **Ficheiro:** int8.sql
9. Quais os Estudantes que completaram todas as Disciplinas da sua turma?
 - **Ficheiro:** int9.sql
10. Quais os Estudantes inscritos em mais que 2 Clubes que tenham mais que uma reprovação registada?
 - **Ficheiro:** int10.sql

GATILHOS

Listamos agora os gatilhos implementados para a terceira entrega do trabalho. Devido à limitação dada de três gatilhos, optámos por escolher duas das restrições não implementadas que considerámos mais pertinentes e interessantes e introduzir uma outra ideia que tivemos separadamente (primeiro gatilho).

Para cada gatilho, incluímos, como é pedido, alguns pareceres acerca de gatilhos complementares que seriam necessários para garantir a implementação completa de cada uma das restrições.

Lembramos que os ficheiros devem ser invocados pela seguinte ordem:
criar.sql -> gatilhon_adiciona.sql -> povoar.sql ->
gatilhon_verifica.sql -> gatilhon_remove.sql.

PRIMEIRO GATILHO — INSCRIÇÃO EM CLUBES

Este primeiro gatilho garante que nenhum Estudante é capaz de se inscrever num Clube (como Não Responsável) que possua o mesmo horário de atividade, expresso pelo atributo diaSemana, que outro Clube em que já esteja inscrito ou de que seja responsável por.

Esta restrição requereria um segundo gatilho com ação BEFORE INSERT ON Clube de corpo semelhante ao que submetemos para garantir que não poderia ser violada. De igual forma, seriam necessários gatilhos com ação BEFORE UPDATE para as tabelas NãoResponsávelClube e Clube.

- **Ficheiros:** gatilho1_adiciona.sql, gatilho1_remove.sql, gatilho1_verifica.sql;

SEGUNDO GATILHO — CONTAGEM DE TURMAS

Este segundo gatilho implementa o atributo derivado númeroTurmas em cada instância de Docente. Sempre que se adicionar uma nova entrada à tabela DocenteTurma, o valor será incrementado.

Esta restrição requereria um gatilho complementar de ação AFTER UPDATE ON DocenteTurma e ainda outro de ação AFTER DELETE ON DocenteTurma de forma a contemplar tanto o incremento como o decremento do número de turmas de um dado Docente. Por outro lado, gatilhos BEFORE INSERT e BEFORE UPDATE of númeroTurmas em Docente garantiriam que a única forma de alterar este atributo seria por alterações a DocenteTurma.

As restrições aos limites inferior e superior para o atributo mencionado já se encontram implementadas no ficheiro criar.sql, por recomendação do professor.

- **Ficheiros:** gatilho2_adiciona.sql, gatilho2_remove.sql, gatilho2_verifica.sql;

TERCEIRO GATILHO — IMPLEMENTAÇÃO DE INTERNO

O terceiro e último gatilho implementa o atributo interno de Estudante, garantindo que um Estudante que possua um Quarto terá o atributo com valor igual a 1 e, caso contrário, igual a 0.

A funcionalidade deste gatilho gerou algumas dúvidas, assim pensamos que fosse melhor apresentar uma pequena explicação do seu teste de verificação.

Começamos por criar dois Quartos, 018 e 019, com os alunos 123456789 e 231234589, respetivamente. O objetivo do teste é trocar os Estudantes de quarto e verificar se o atributo interno é alterado devidamente.

Coloca-se então 231234589 no quarto 018, associando antes o valor NULL ao atributo estudante do quarto 019 — pois cada aluno só pode estar associado a um quarto num determinado momento (restrição UNIQUE). Ao imprimir o valor do atributo interno de 123456789, este valor será 0, pois este *momentaneamente* não se encontra em nenhum quarto e para 231234589, interno será 1.

De seguida, introduzimos 123456789 no quarto 019 e, ao fazermos de novo a verificação de interno, este é 1 para ambos, pelo que o gatilho exerce a sua função corretamente.

Esta restrição beneficiaria da adição de gatilhos complementares de ação AFTER INSERT ON para Quarto (garante que não se insere um estudante interno a que ainda não se atribuiu quarto), um AFTER DELETE ON Quarto (caso um quarto seja eliminado, o Estudante que nele habita deve tornar-se não interno) e, um último AFTER UPDATE ON Estudante (ao alterar-se interno para 0, deve ir-se ao quarto que antes lhe era atribuído, colocando estudante a NULL; ao alterar o atributo interno deste Estudante para 1, deveria ocorrer erro por este não ter nenhum quarto atribuído).

Ficheiros: gatilho3_adiciona.sql, gatilho3_remove.sql,
gatilho3_verifica.sql;

REFERÊNCIAS

Como mencionado na introdução, a ideia de que fruiu este trabalho provém de algo pessoal a um dos membros do nosso grupo, Mateus Silva: nomeadamente um livro, de autoria do mesmo, intitulado *Noites Sem Sono*.

A obra completa está disponível para leitura *online*⁶, totalmente em Português, numa história com fortes ligações à educação e às relações entre alunos e professores. E, para os mais difíceis de agradar, há ainda imenso mistério e ação à mistura para garantir uma leitura compulsiva!

Aconselhamos a leitura do livro, caso esta descrição deixe o leitor com curiosidade em saber o que o *Colégio Yamazaki* de facto esconde!

⁶ No serviço de leitura *online*, Tapas: [CLICAR AQUI](#)