Text

Description automatically generated with medium confidence

**Licenciatura Engenharia Informática**

**Ciência de Dados**

**Grupo G9**

**Segundo Semestre**

**Executado por**

2341 Jorge Ramos Jorge.ramos@my.istec.pt

2343 João Nunes [joao.nunes@my.istec.pt](mailto:joao.nunes@my.istec.pt)

2177 Rui Boto rui.boto@my.istec.pt

**Orientado por**

Professor Ricardo Ferreira

**Entregue em**

02/05/2022

# Índice

[1. Índice 2](#_Toc102417384)

[2. Objetivos 3](#_Toc102417385)

[3. Desenvolvimento 4](#_Toc102417386)

[Modelo JSON 4](#_Toc102417387)

[Modelo entidade e associação – Gráfico 6](#_Toc102417388)

[Modelo Relacional (SQL) 8](#_Toc102417389)

[4. Conclusões 11](#_Toc102417390)

# Objetivos

O objetivo deste projeto é a aplicação dos conhecimentos na conceção de uma base de dados em MongoDB.

Para demostrar foi usado um exemplo da base de dados relacional anteriormente feita em SQL sobre uma entidade Bancária.

Para essa base de dados vai ser apresentado o modelo JSON da base de dados, com as suas associações.

# Desenvolvimento

## Modelo JSON

Foi utilizado a mesma collection para armazenar, em documentos diferentes, todos as “tabelas” da base de dados, na maioria dos documentos foi colocado manualmente o “\_id” para posteriormente ser associado a outros documentos que fizessem sentido.

No paragrafo seguinte vai ser apresentado os documentos que foram utilizados:

Documento Conta:

Text

Description automatically generated

A conta tem os “\_id” inseridos manualmente, a começar em 100, têm um número de conta, nib, tipo de conta que identifica se a conta e a “prazo”, “ações”, os seus cartões associados e por fim o saldo.

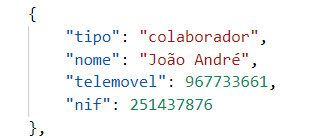
Documento Cliente:

Text

Description automatically generated

O documento Cliente tem não tem “\_id” manual, este Id foi gerado automaticamente, outras informações e por fim a “conta” onde é identificado o número de conta associado a este cliente.

Documento Colaborador:



O documento colaborador, inicialmente tinha o objetivo de ter permissões, no entanto não conseguimos alcançar esse objetivo.

Documento Movimento:

Graphical user interface, text

Description automatically generated

No movimento está a informação da conta de origem e conta de destino de dado movimento, em seguida têm o valor que foi movimentado, numa camada aplicacional, o valor seria descontado na conta de destino e incrementado na conta de origem.

Documento Cartão:

Graphical user interface, text

Description automatically generated

No cartão tem também o “\_id” que está preenchida manualmente, para ser associado com o Documento “conta”.

## Modelo entidade e associação – Gráfico

Neste bloco vai ter a indicação em gráfico das entidades e suas associações:

Diagram

Description automatically generated

Com associação “têm” para a entidade “cliente”:

A picture containing text, whiteboard

Description automatically generated

A Associação “têm” também se associa com a entidade “Cartão”:

Diagram

Description automatically generated

A entidade “Colaborador” têm associação “gere” com a entidade “Colaborador”:

Diagram

Description automatically generated

A Entidade conta têm associação “faz” com a Entidade “Acao”:

Diagram

Description automatically generated

A entidade “Acao” têm por sua vez 3 Disjoints que especifica cada tipo de ação:

A picture containing text, indoor, dirty

Description automatically generated

A entidade “conta” também têm 3 Disjoints para especificar o tipo de conta:

A picture containing text, whiteboard

Description automatically generated

A entidade Conta Ações “C\_Acoes” têm uma associação “faz” com a entidade “I\_Acoes”, que por sua vez tem associação “investe” para a entidade “Acoes”, assim será possível associar as ações compradas e as suas quantidades entre diferentes ações que estejam disponíveis.

A picture containing text, whiteboard

Description automatically generated

## Modelo Relacional (SQL)

O modelo relacional conta com 16 tabelas e estas são as chaves primárias, estrangeiras e suas associações:

Começando pela tabela “Conta” esta têm a PK(N\_Conta) e esta tabela associa-se com 6 tabelas:

1. Conta\_Cliente (1:1)
2. Movimento (1:N)
3. Cartão (1:N)
4. C\_Prazo (N:1)
5. C\_Acoes (N:1)
6. C\_Obrigacoes (N:1)

Diagram

Description automatically generated

A tabela “C\_Acoes” tem como PK(ID\_CAcoes) por sua vez uma associação com a tabela “IAcoes\_CAcoes” que liga a outra tabela “I\_Acoes” esta liga também a tabela “Acoes”:

Diagram

Description automatically generated

A tabela “Cliente” tem como PK(NIF) tem associações com 3 outra tabelas:

1. Cartão (1:N)
2. Colaborador(N:1)
3. Conta\_Cliente(1:1)

Diagram

Description automatically generated

A “Conta\_Cliente”, pode ter uma “Ação” e esta ação pode ser “A\_PagamentosEstado”,”A\_PagamentosServico” ou A\_DebDireto, as associações estão feitas da seguinte forma:

Conta\_Cliente 🡪 Acao (1:N)

Conta\_Cliente 🡪 A\_PagamentoEstado (1:N)

Conta\_Cliente 🡪 🡪 A\_PagamentoServicos (1:N)

Conta\_Cliente 🡪 A\_DebDireto (1:N)

Accao 🡪 A\_PagamentoEstado(1:1)

Accao 🡪 A\_PagamentoServicos(1:1)

Accao 🡪 A\_DebDireto(1:1)

Diagram

Description automatically generated

# Conclusões

Este trabalho, com a mudança drástica para um modelo de MongoDB foi muito complicado entender como funcionava, a sua implementação é em quase tudo deferente da de modelos relacional (SQL).

Outra parte que também foi bastante difícil foi a forma como podíamos juntar o Python para a adição e manipulação de dados.

Mesmo a manipulação de dados por querys é bastante diferente de SQL dificultando o recolher de dados filtrados.