



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

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

Orientado por

Professor Ricardo Ferreira

Entregue em

02/05/2022

1. Índice

1. Índice	2
2. Objetivos	3
3. Desenvolvimento	4
Modelo JSON	4
Modelo entidade e associação – Gráfico	6
Modelo Relacional (SQL)	8
4. Conclusões	11

2. Objetivos

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

Para demonstrar 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.

3. 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:

```
{
  "tipo": "conta",
  "_id": 100,
  "n_conta": 101100111,
  "nib": "101100111-101",
  "tipo_de_conta": "Conta a Prazo",
  "cartoes": [7777, 7778],
  "saldo": 5000
}
```

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:

```
{
  "tipo": "cliente",
  "nif": 251437888,
  "nome": "Jorge Ramos",
  "telemovel": 969918230,
  "tipo_titular": "Principal",
  "situacao_laboral": "Empregado",
  "rendimento_anual": 250000,
  "conta": 101100111
}
```

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:

```
{
  "tipo": "colaborador",
  "nome": "João André",
  "telemovel": 967733661,
  "nif": 251437876
},
```

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

Documento Movimento:

```
{
  "tipo": "movimento",
  "_id": 1,
  "conta_dest": 101100112,
  "conta_origem": 101100114,
  "valor": 50
},
```

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.

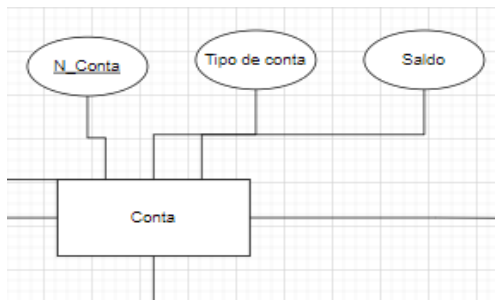
Documento Cartão:

```
{
  "tipo": "cartao",
  "_id": 7777,
  "n_cartao": 101100111101100111,
  "pin": 1234,
  "tipo_cartao": "debito"
}, {
```

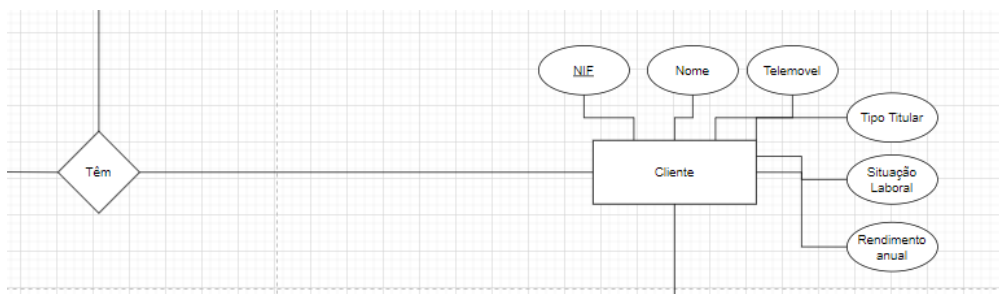
No cartão tem também o “_id” que está preenchida manualmente, para ser associado com o Documento “conta”.

Modelo entidade e associação – Gráfico

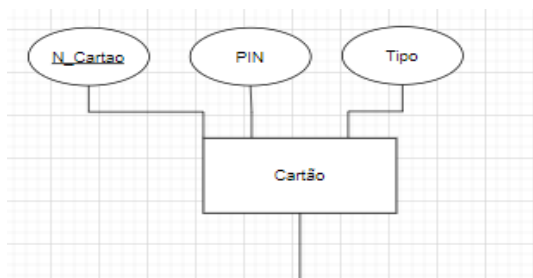
Neste bloco vai ter a indicação em gráfico das entidades e suas associações:



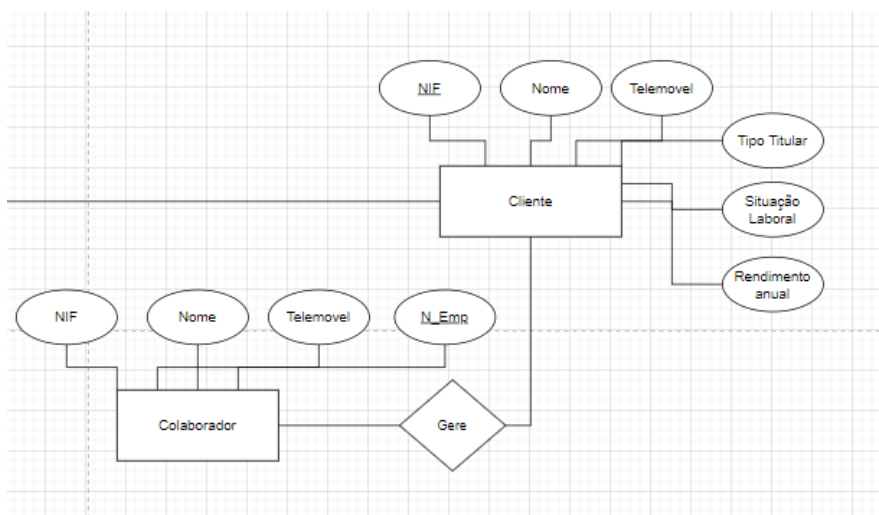
Com associação "têm" para a entidade "cliente":



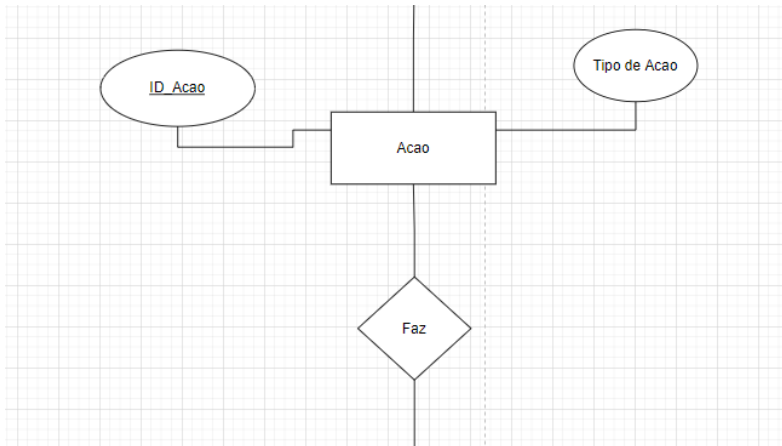
A Associação "têm" também se associa com a entidade "Cartão":



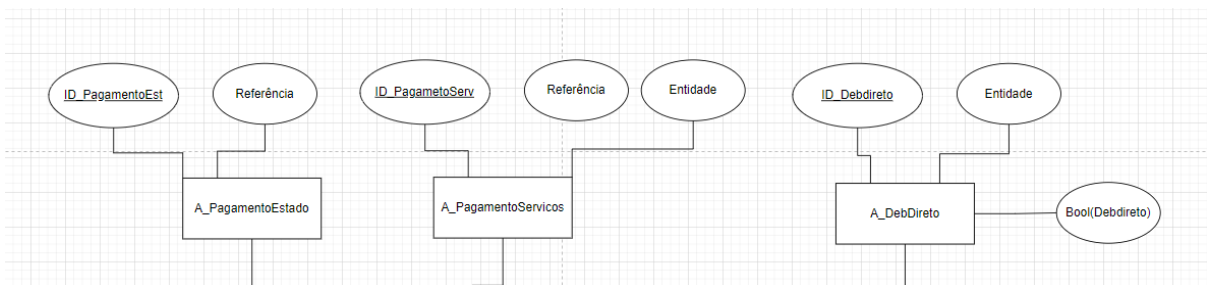
A entidade "Colaborador" têm associação "gere" com a entidade "Colaborador":



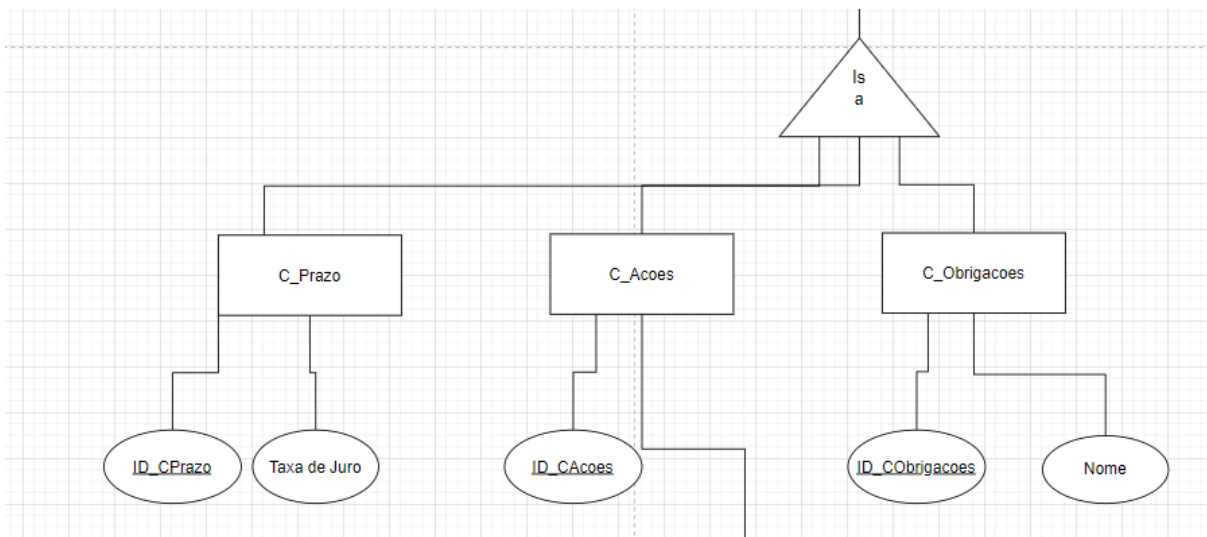
A Entidade conta têm associação “faz” com a Entidade “Acao”:



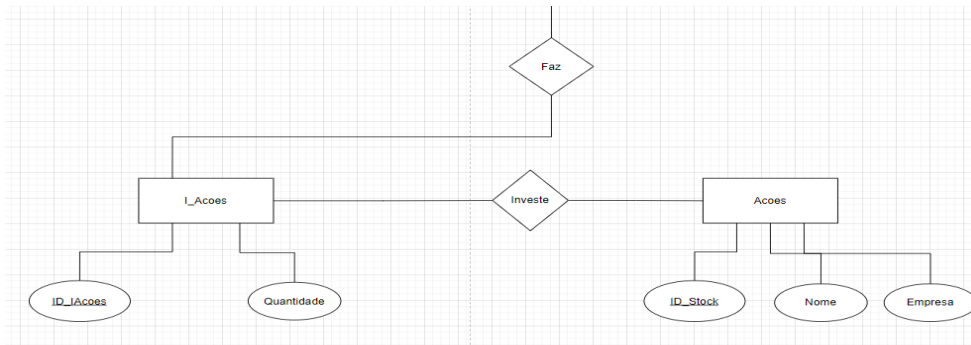
A entidade “Acao” têm por sua vez 3 Disjoints que especifica cada tipo de ação:



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



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.

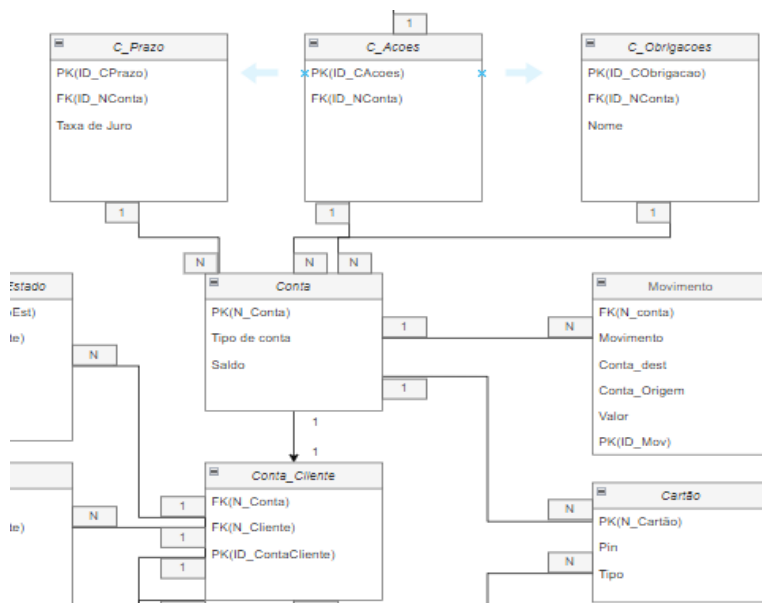


Modelo Relacional (SQL)

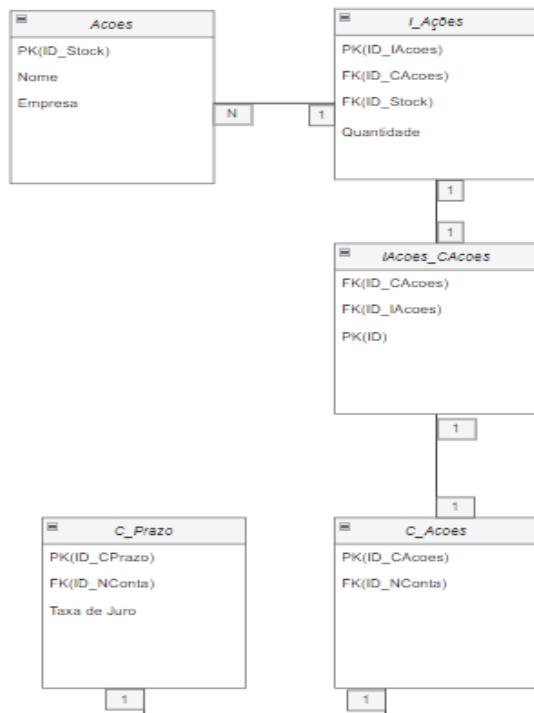
O modelo relacional conta com 16 tabelas e estas são as chaves primárias, estrangeiras e suas associações:

Começando pela tabela “Conta” esta tem 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)

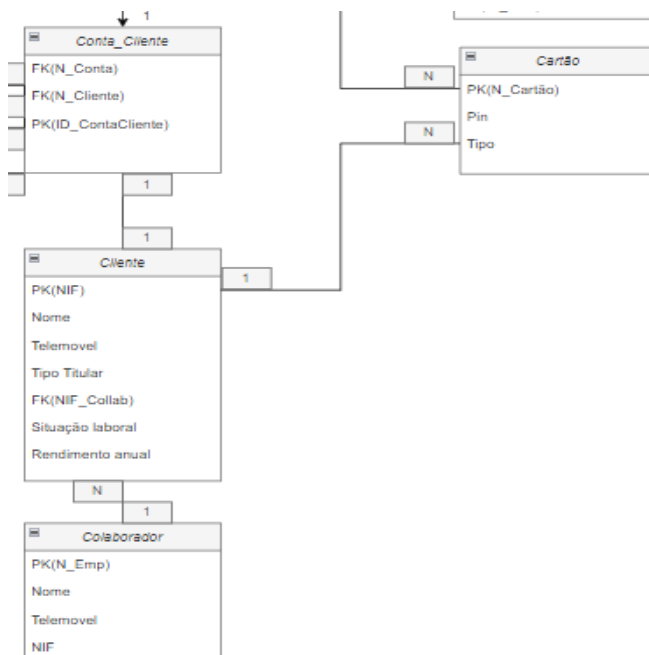


A tabela “C_Acoes” tem como PK(ID_CAcoes) por sua vez uma associação com a tabela “IAcoes_CAcoes” que liga a outra tabela “I_Acoes” esta liga também a tabela “Acoes”:



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)



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)

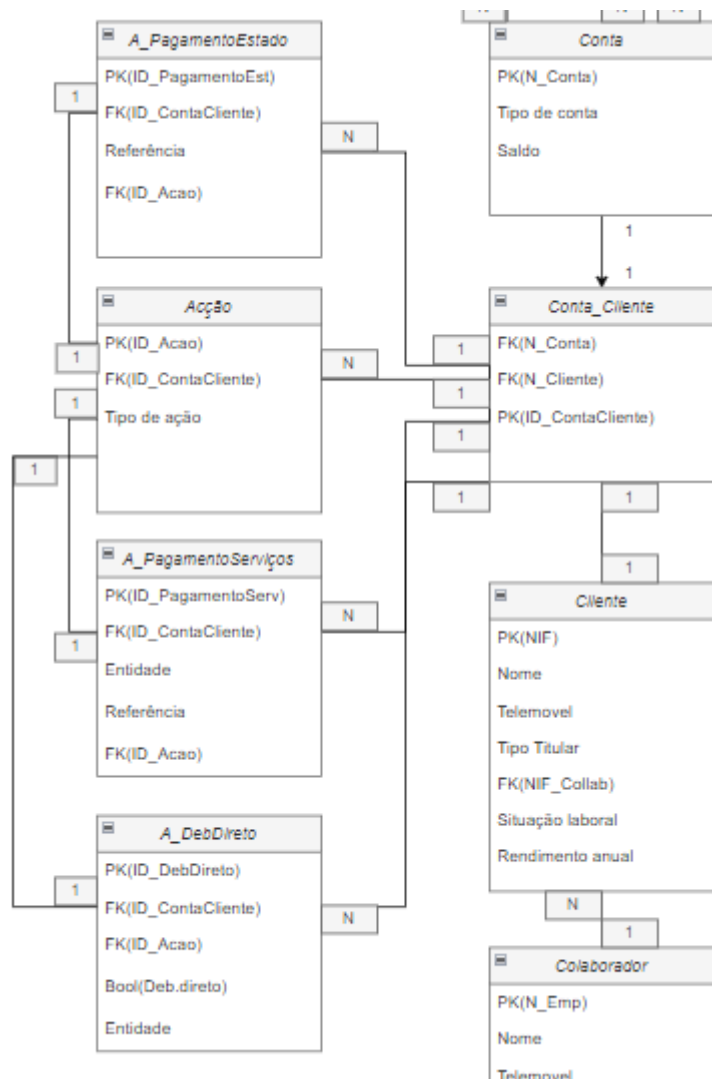
Conta_Cliente → → A_PagamentoServicos (1:N)

Conta_Cliente → A_DebDireto (1:N)

Accao → A_PagamentoEstado(1:1)

Accao → A_PagamentoServicos(1:1)

Accao → A_DebDireto(1:1)



4. 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 queries é bastante diferente de SQL dificultando o recolher de dados filtrados.