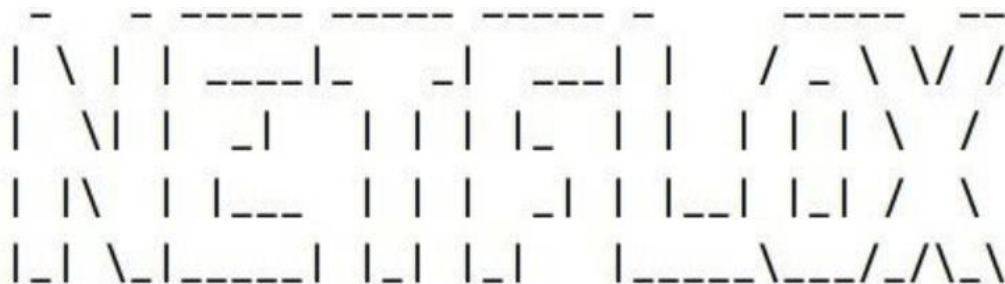


UNIVERSIDADE DE COIMBRA  
MESTRADO INTEGRADO EM ENGENHARIA FÍSICA  
BASE DE DADOS - DISCIPLINA ISOLADA

---



## PROJECT FINAL REPORT

---



UNIVERSIDADE DE COIMBRA

ANTÓNIO MONTEIRO | 2018279239  
GONÇALO GOUVEIA | 2018277419

<i><b>INDEX</b></i>	
<i>INDEX</i> -----	1
<i>IMAGE INDEX</i> -----	2
<i>ABSTRACT</i> -----	2
<i>FIRST REPORT TASKS</i> -----	3
<i>PROJECT MANAGEMENT</i> -----	3
<i>SECOND REPORT GROUP TASKS</i> -----	3
<i>ER DIAGRAM</i> -----	4
<i>PHYSICAL ER DIAGRAM</i> -----	5
<i>LEVEL INTERACTION</i> -----	5
<i>GRAPHIC INTERFACE</i> -----	6
<i>CLIENT FUNCTIONS</i> -----	7
<i>ADMIN FUNCTIONS</i> -----	8
<i>TRANSACTION</i> -----	10
<i>TRIGGER PL/SQL</i> -----	11
<i>NORMALIZAÇÃO</i> -----	11
<i>COMMENTS</i> -----	12

---

## IMAGE INDEX

---

Figura 1 - Diagrama ER. ....	4
Figura 2 - Diagrama físico.....	5
Figura 3- representação gráfica das interações entre níveis.....	6
Figura 4- registo e login do cliente, semelhante ao do Administrador .....	6
Figura 5- login do Admin .....	6
Figura 6- modelo de interface gráfica.....	6
Figura 7-alugar artigo.....	7
Figura 8- sistema de visualização de mensagens, ver mensagens lidas, ver não lidas e voltar a ver as não lidas .....	7
Figura 9- procurar todos os novos filmes .....	8
Figura 10- histórico de aluguer .....	8
Figura 11- consultar saldo.....	8
Figura 12- alterar preço.....	9
Figura 13- enviar mensagem a apenas um cliente (exemplo) .....	9
Figura 14- adicionar saldo a cliente .....	9
Figura 15- remover artigo e participantes correspondentes.....	9
Figura 16- adicionar artigo .....	9

---

## ABSTRACT

---

Cada vez mais se lida com grandes quantidades de dados, o que leva à necessidade da construção de estruturas que consigam não só armazenar todos esses dados, como também organizá-los e torná-los de fácil acesso.

A empresa NETFLOX pretende um sistema que lhe permita gerir o seu negócio online, que consiste essencialmente em alugar artigos de vários tipos (séries, filmes e documentários) para visualização online e gerir o seu inventário de artigos disponíveis para alugar. Assim, é necessário construir uma aplicação de bases de dados que permita que clientes possam, por exemplo, pesquisar e alugar artigos disponíveis, e que permita que os administradores, que são manualmente adicionados à base de dados, possam desempenhar tarefas típicas de gestão do sistema.

---

**FIRST REPORT TASKS**

---

- Tempo gasto com a disciplina.
- Distribuição final de tarefas por cada elemento do grupo.
- Todo o código desenvolvido para este projeto, ficheiros JSON.
- Diagrama Entidade Relacionamento (indicando que alterações sofreu desde a entrega I, se existirem).
- Diagrama físico, versão simplified no ONDA .
- Screenshots ilustrativos do aspeto geral da aplicação e das suas partes mais importantes.
- Distribuição final de tarefas por cada elemento do grupo.

---

**PROJECT MANAGEMENT**

---

Aluno/nmr	Presenças nas aulas T e PL. - Extra-aula (h)	Esforço gasto para esta entrega (h)
Antonio Monteiro 2018279239	aulasPL	~20
Gonçalo Gouveia 2018277419	aulasPL	~20

Ambos somos alunos de Engenharia Física a fazer a cadeira isolada de Base de Dados, não temos horário compatível com nenhuma aula da cadeira de BD, todo o demais esforço de acompanhar a matéria leccionada parte de estudo coletivo/individual e autónomo.

---

**SECOND REPORT GROUP TASKS**

---

Para completar as tarefas delineadas para esta entrega, repartimos o trabalho como mostrado na seguinte tabela.

Gonçalo Gouveia   2018277419	António Monteiro   2018279239
Organização de código	Alterações no EDR1
Scripts em Python/SQL	Scripts em Python/SQL
Elaboração do <i>final report</i>	PL/SQL
Verificação de erros	Verificação de erros

*Figure 1- Tarefas delineadas para o segundo relatório.*

## ER DIAGRAM

Para esta etapa final da entrega consideramos e reconhecemos que o diagrama anterior tinha erros e implementações defeituosas ou menos óbvias de relacionar. Pelo que refizemos o Diagrama.

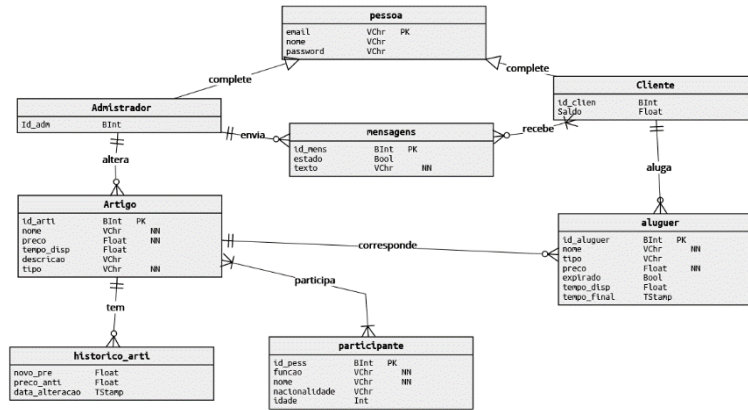


Figura 1 - Diagrama ER.

- Relação estabelecida entre as entidades **Admin | Mensagem** é **[0,...,n]:[1:1]**, um Admin pode enviar zero ou nenhuma mensagem e cada mensagem tem que ser enviada por um e apenas um Admin
- Relação estabelecida entre as entidades **Cliente | Mensagem** é **[0,...,n]:[1,...,n]**, já que um cliente pode ou não receber várias mensagens, que a mesma mensagem pode ser recebida por um ou vários clientes.
- Relação estabelecida entre as entidades **Cliente | Aluguer** é **[0,...,n]:[1:1]**, pois um cliente pode ter nenhum ou muitos alugueres e cada aluguer corresponde unicamente a um cliente.
- Relação estabelecida entre as entidades **Aluguer | Artigo** é **[0,...,n]:[1:1]**, já que a cada processo de aluguer corresponde a aquisição de um Item, e a cada item pode estar associado a n alugueres.
- Relação estabelecida entre as entidades **Artigo | Participante** é **[1,...,n]:[1,...,n]**, cada filme, série, documentário pode ter um ou vários atores associados à obra, e cada actor pode pertencer a mais que um item de lista de aluguer.
- Relação estabelecida entre as entidades **Admin | Artigo** é **[0,...,n]:[1:1]**, um administrador pode querer interagir com nenhum ou vários artigos e cada alteração está associada a apenas a um administrador.

## PHYSICAL ER DIAGRAM

O esquema físico da base de dados compreende todas as tabelas que compõem a versão final da base de dados e os seus respectivos atributos.

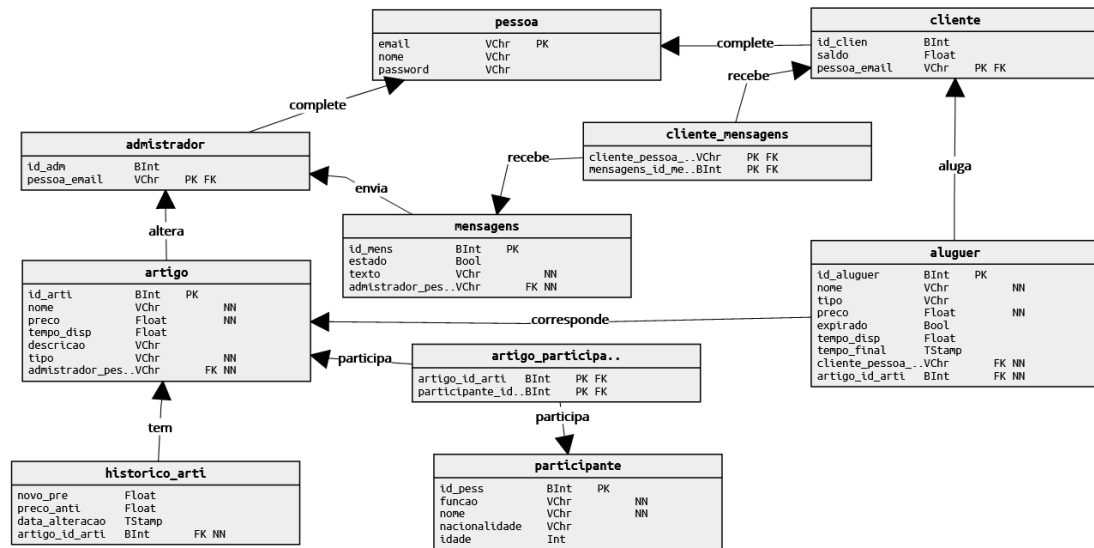
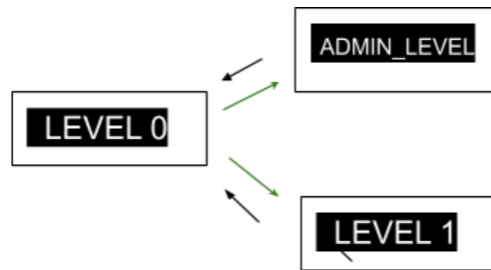


Figura 2 - Diagrama físico

## LEVEL INTERACTION



- Assim que o usuário entra no site, encontra-se **nível\_0** e pode fazer as seguintes opções: pode fazer Login ou pode se Registrar. Fazendo qualquer escolha entra no **level\_1**. Caso seja selecionada a opção registrar, o usuário insere os seus dados (email, password, nome), caso o email inserido já exista na base de dados levanta uma mensagem de erro, em que pede para tentar voltar a registrar-se.
- Após se registrar e fazer login, ou após login, o cliente pode fazer as seguintes escolhas: listar artigos, procurar artigos, comprar artigos, ver histórico de cliente, ver mensagens, ver saldo e logout.

- Caso, no login se verifique que o mail é de um administrador, entra no **admin\_level**, que tem mais opções disponíveis tais como mandar mensagem a clientes, adicionar itens, ver estatísticas do site e eliminar itens segundo alguns critérios e adicionar saldo a clientes caso o cliente tenha depositado dinheiro.
- Um administrador é introduzido da base de dados manualmente.
- Em qualquer altura o cliente/Admin pode fazer logout, que o faz voltar ao **nível\_0** do site.

*Figura 3- representação gráfica das interações entre níveis*

```
Bem vindo! Deseja 1-Registar-se 2-Fazer Login como cliente 3-Fazer Login como administrador
Qual o seu nome? roberto
Qual o seu email? roberto@mail
Qual a Password? roberto
Conta criada com sucesso! Recebeu um saldo grátis de 5€ para gastar! Só precisa de fazer login :-))
Bem vindo! Deseja 1-Registar-se 2-Fazer Login como cliente 3-Fazer Login como administrador
Email: roberto@mail
Password: roberto
Bem vindo!
Conta de Roberto
```

*Figura 4- registo e login do cliente, semelhante ao do Administrador*

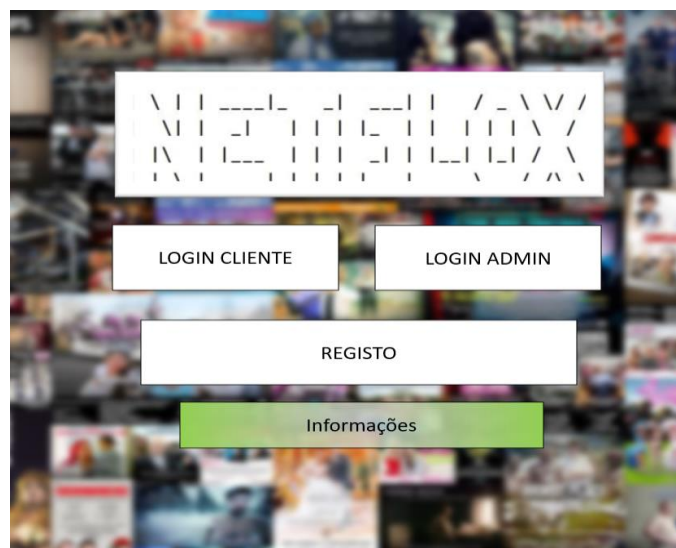
```
Bem vindo! Deseja 1-Registar-se 2-Fazer Login como cliente 3-Fazer Login como administrador
Email: roberto@mail
Password: roberto@mail
Bem vindo!
1-Enviar mensagem 2-Alterar preço de um artigo 3-Remover um artigo 4-Adicionar um artigo 5-Alterar o saldo de um cliente 6-Estatísticas 7-Logout
```

*Figura 5- login do Admin*

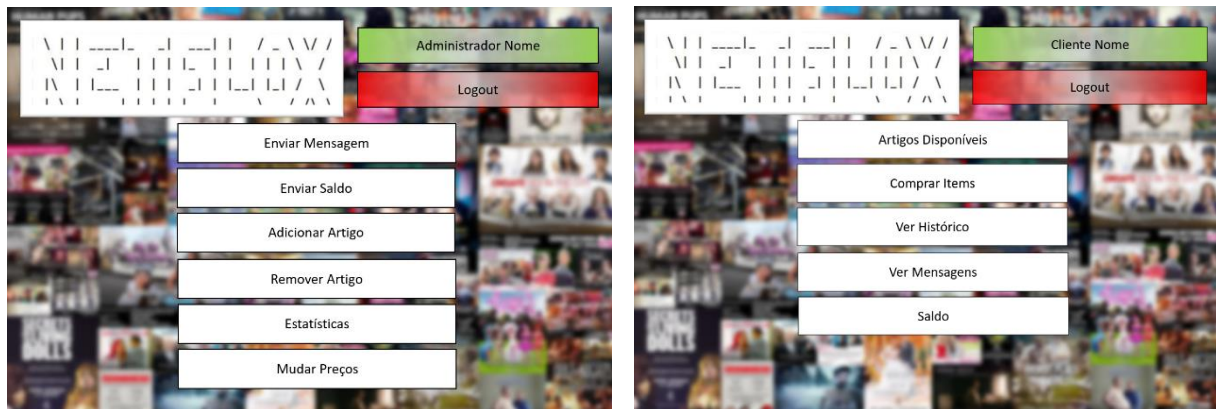
---

## GRAPHIC INTERFACE

---



*Figura 6- modelo de interface gráfica*



## CLIENT FUNCTIONS

- A opção **alugar()**, permite ao cliente alugar um item da lista dos artigos, apenas se o cliente tiver dinheiro para o poder alugar. Após procura de um (ou mais) artigo(s) o cliente verá disponível a opção de alugar (ou não) o(s) artigo(s) procurado(s).

```
selecione o nome do(a) filme/série que pretende alugar:Roberto Jones
Pretende alugar o filme/série? 1-Sim 2-Não.

Transação efetuada!)
Conta de Roberto
```

*Figura 7-alugar artigo*

- A opção **mensagem()**, permite ao cliente ver as mensagens recebidas pelo administrador, sendo possível distinguir as mensagens lidas das não lidas. Quando o cliente visualiza as mensagens não lidas estas passam a um estado = True.

```
1-Enviar mensagens 2-Alterar preço de um artigo 3-Remover um artigo 4-Adicionar um artigo 5-Alterar o saldo de um cliente 6-Estatísticas 7-logout
Obrigado por usar os nossos serviços
Bem vindo! Deseja 1-Registar-se 2-Fazer login como cliente 3-Fazer login como administrador
Email: roberto@uoi.pt
Password: roberto
Bem vindo!
Conta de Roberto
1-Todos os artigos disponíveis para alugar 2-Procure artigos novos 3-Procure artigos alugados 4- Ver historico 5-Ver mensagens 6- Consultar saldo 7-logout
1-mensagens não lidas 2-mensagens lidas

Mensagem: Olá Roberto!
Conta de Roberto
1-Todos os artigos disponíveis para alugar 2-Procure artigos novos 3-Procure artigos alugados 4- Ver historico 5-Ver mensagens 6- Consultar saldo 7-logout
1-mensagens não lidas 2-mensagens lidas

Mensagem: Olá Roberto!
Conta de Roberto
1-Todos os artigos disponíveis para alugar 2-Procure artigos novos 3-Procure artigos alugados 4- Ver historico 5-Ver mensagens 6- Consultar saldo 7-logout
1-mensagens não lidas 2-mensagens lidas
Conta de Roberto
```

*Figura 8- sistema de visualização de mensagens, ver mensagens lidas, ver não lidas e voltar a ver as não lidas*



- A opção **procurar\_todos()**, permite ao cliente procurar itens por tipo, título, atores e realizador.

```
1-Todos os artigos disponiveis para alugar 2-Procure artigos novos 3-Procure artigos alugados 4- Ver historico 5-Ver mensagens 6- Consultar saldo 7-Logout
Como deseja pesquisar? 1- Nome 2- Género 3- Realizador 4-Ator:
Qual o interviniente que pretende pesquisar?Indiana Jones

Título do filme-Indiana Jones
Tipo-Aventura
Preço-4.8€
Tempo disponível-15 dias
Descrição-Arqueólogo em apuros
```

Figura 9- procurar todos os novos filmes

- A opção **procura\_alugados()**, permite ao cliente procurar os itens alugados por tipo, título, realizador, ator. Quando a data final de aluguer já passou apresenta que o artigo procurado expirou e não estará mais disponível para visualização.

```
Conta de Roberto
1-Todos os artigos disponiveis para alugar 2-Procure artigos novos 3-Procure artigos alugados 4- Ver historico 5-Ver mensagens 6- Consultar saldo 7-Logout
Como deseja pesquisar? 1- Nome 2- Género 3- Realizador 4-Ator:
Qual o nome do artigo?Indiana Jones

Título do filme-Indiana Jones
Tipo-Aventura
Preço-4.8€
Pode ver o filme!
```

Figura 7- procurar todos os filmes alugados

- A opção **historico()**, permite ao cliente aceder a informações de quanto dinheiro já gastou no site, tal como, aceder a informações dos alugueres, como o nome, preço e estado dos seus alugueres.

```
1-Todos os artigos disponiveis para alugar 2-Procure artigos novos 3-Procure artigos alugados 4- Ver historico 5-Ver mensagens 6- Consultar saldo 7-Logout
Nome do artigo: Indiana Jones Preço do artigo: 4.8 -> artigo ativo
O total gasto é de 4.8€
```

Figura 10- histórico de aluguer

- A opção **saldo()**, permite ao cliente conferir qual o seu saldo. O saldo nunca será negativo pois o cliente nunca conseguirá alugar um artigo com saldo insuficiente.

```
Conta de Roberto
1-Todos os artigos disponiveis para alugar 2-Procure artigos novos 3-Procure artigos alugados 4- Ver historico 5-Ver mensagens 6- Consultar saldo 7-Logout
Tem 1.8€ disponiveis para gastar
Conta de Roberto
```

Figura 11- consultar saldo

---

## ADMIN FUNCTIONS

---

- A opção **mudar\_preco()**, permite a um administrador alterar o preço de um artigo (é mantido um histórico com as correções que se pode consultar quando se visualiza os detalhes de um artigo).

```

1-Enviar mensagem 2-Alterar preço de um artigo 3-Remover um artigo 4-Adicionar um artigo 5-Alterar o saldo de um cliente 6-Estatísticas 7-Logout:
qual o artigo a alterar? Digite o nome do item: Roberto Jones
qual o novo preço? 1
alterações efetuadas!

```

Figura 12- alterar preço

- A opção **enviar\_mensagem()**, permite ao administrador enviar mensagens a um cliente, sendo que pode seleccionar enviar a um único cliente ou enviar uma mensagem a todos os clientes registados no site.

```

Sem vindo! Deseja 1-Registar-se 2-Fazer Login como cliente 3-Fazer Login como administrador!
Email: admin@igmail
Password: 123456789
Sem vindo!
1-Enviar mensagem 2-Alterar preço de um artigo 3-Remover um artigo 4-Adicionar um artigo 5-Alterar o saldo de um cliente 6-Estatísticas 7-Logout:
selecione se quer mandar mensagem a 1-clientes especificos ou 2-todos os clientes:
quer mandar mensagem a quantas pessoas?:
mail da 1ª pessoa: robert@igmail
escreva a mensagem a enviar: ola roberto!
Mensagem enviada!

```

Figura 13- enviar mensagem a apenas um cliente (exemplo)

- A opção **saldo()**, permite o Admin aumentar o saldo de um cliente assumindo que o cliente envia o dinheiro para o administrador.

```

1-Enviar mensagem 2-Alterar preço de um artigo 3-Remover um artigo 4-Adicionar um artigo 5-Alterar o saldo de um cliente 6-Estatísticas 7-Logout:
Introduza o email do cliente de quem quer alterar o saldo: robert@igmail
Qual o novo saldo? 50
Operação concluída com sucesso!

```

Figura 14- adicionar saldo a cliente

- A opção **remove\_artigo()**, permite ao Admin remover um artigo da lista de artigos caso um item ainda não tenha sido alugado por nenhum cliente.

```

1-Enviar mensagem 2-Alterar preço de um artigo 3-Remover um artigo 4-Adicionar um artigo 5-Alterar o saldo de um cliente 6-Estatísticas 7-Logout:
Elimine apenas um item de cada vez. Introduza o nome do item que pretende eliminar: Jones
Alterações efetuadas!

```

Figura 15- remover artigo e participantes correspondentes

- A opção **adicionar\_artigo()**, permite ao Admin adicionar um artigo, e os seus detalhes, tais como tipo de artigo, título do artigo, tempo disponível de aluguer e preço do mesmo. Ao adicionar o artigo, é também pedido ao admin que introduza alguns participantes do filme.

```

1-Enviar mensagem 2-Alterar preço de um artigo 3-Remover um artigo 4-Adicionar um artigo 5-Alterar o saldo de um cliente 6-Estatísticas 7-Logout:
Título do artigo: Jones
Preço do artigo: 3
Tipo do artigo: Thriller
Tempo disponível após aluguer do artigo: 15
Sinopse do filme/série: Roberto como passageiro
Introduzir os participantes do filme/série. Quantos participantes quer adicionar?:
Introduza o nome do 1º participante:  Spielberg
Qual o papel do participante? (se é ator, realizador,...) Realizador
Qual a idade do participante? 73
Qual a nacionalidade do participante? Americana
Alterações efetuadas!

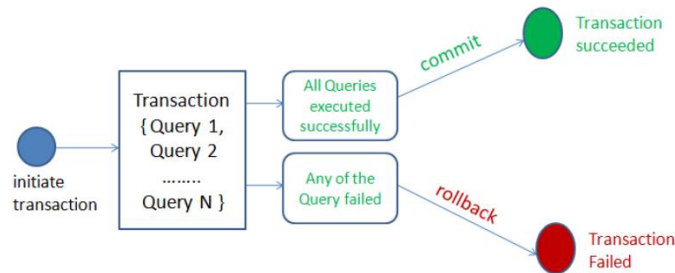
```

Figura 16- adicionar artigo

---

## TRANSACTION

---



As transações são um conjunto de instruções SQL executadas em bloco. Ou todas as instruções têm êxito ou nenhuma delas.

Se a transação termina corretamente, podemos usar o comando **COMMIT** para armazenar permanentemente suas modificações. Caso contrário, podemos usar o comando **ROLLBACK** para cancelar todas as modificações e restaurar o banco de dados ao seu estado inicial.

Implementámos transações nalgumas das funções como por exemplo na `procura_alugados()`. Em baixo, a título de exemplo, está a transação implementada na função `alterar_saldo()`, em que caso exista algum erro no update do saldo na tabela do cliente, não vai existir um commit dos dados na BD. Acontece sim, um rollback à última alteração no sistema.

```
def alterar_saldo():
    try:
        aux = input("Introduza o email do cliente de quem quer alterar o saldo: ")
        newbalance = int(input("Qual o novo saldo? "))
        maillist = []
        cur.execute("SELECT * FROM cliente WHERE pessoa_email = %s", [aux])
        for linha in cur.fetchall():
            maillist.append(linha['pessoa_email'])

        if aux not in maillist:
            print("A pessoa que selecionou não está registada. Tente novamente!")
            alterar_saldo()
        else:
            cur.execute(f"UPDATE cliente SET saldo = {newbalance} WHERE pessoa_email = '{aux}'")
            conn.commit()
            print("Operação concluída com sucesso!")
    except psycopg2.DatabaseError as erro:
        print('Ocorreu um erro! Operação anulada.')
        print(erro)
        conn.rollback()
    except:
        print('Ocorreu um erro! Operação anulada.')
        conn.rollback()
```

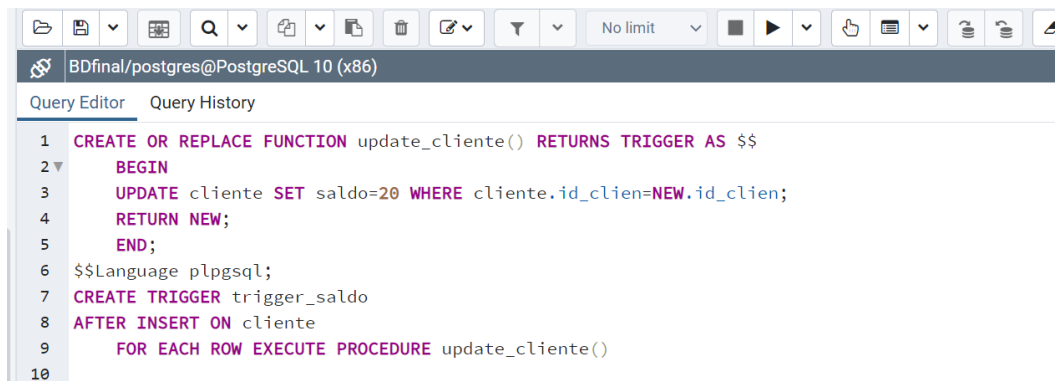
---

## TRIGGER PL/SQL

---

**PL/SQL** (*Procedural Language/ Structured Query Language*) é uma extensão da linguagem padrão SQL.

É a linguagem básica para criar programas complexos e poderosos, não só em base de dados, mas também em diversas ferramentas Oracle. Para este trabalho implementamos um procedimento em PL/SQL em que sempre que se insere uma pessoa na tabela, esta recebe um saldo de registo de 20 unidades monetárias



```
1 CREATE OR REPLACE FUNCTION update_cliente() RETURNS TRIGGER AS $$
2 BEGIN
3     UPDATE cliente SET saldo=20 WHERE cliente.id_clien=NEW.id_clien;
4     RETURN NEW;
5 END;
6 $$Language plpgsql;
7 CREATE TRIGGER trigger_saldo
8 AFTER INSERT ON cliente
9     FOR EACH ROW EXECUTE PROCEDURE update_cliente()
10
```

Figure 2- *plpgsql trigger*

---

## NORMALIZAÇÃO

---

De uma forma geral, um dos objectivos ao projetar uma aplicação de bases de dados é ter tabelas sem dados redundantes (ou com o mínimo de redundância)

Um correta normalização impede que existam

- Anomalias de inserção.
- Anomalias de atualização.
- Anomalias de apagamento.

É fácil perceber que todas as relações estão na forma normal de Boyce-Codd, uma vez que todos os atributos são dependentes apenas do atributo-chave. Criamos um ID específico para cada entidade que facilitou a normalização para a forma normal de Boyce-Codd uma vez que todos os atributos dependem do atributo-chave que criamos.

---

### *COMMENTS*

---

Acreditamos que as tarefas delineadas para esta etapa do projeto foram concluídas com êxito.

Se tivéssemos mais tempo poderíamos ter implementado ou melhores funções como um melhor search engine, poderíamos ter corrigido erros como questões de **segurança e privacidade de dados** e faríamos uma possível **interface gráfica**.

Concluindo, reconhecemos a utilidade e a praticidade de Base de Dados, tal como as suas relações e algumas ferramentas de manipulação das mesmas.

Gostámos de concretizar e desenvolver este projeto, atendendo a que ambos frequentamos esta cadeira como cadeira isolada. Por último reconhecemos que teríamos aproveitado mais a cadeira caso pudessemos ter comparecido às aulas teóricas.