

NETFLOX

Projeto Final Bases de Dados

Relatório - Entrega II

João Vasco - UC2019236378

Gonçalo Cavaleiro - UC2018279569

Conteúdo

[Organização final geral da aplicação 3](#_Toc58702268)

[Ficheiros: 3](#_Toc58702269)

[Aplicação Principal 3](#_Toc58702270)

[Outros Ficheiros 3](#_Toc58702271)

[Menus: 4](#_Toc58702272)

[Menus secundários – Utilizador 4](#_Toc58702273)

[Menus secundários – Administrador 5](#_Toc58702274)

[Fotos interface: 6](#_Toc58702275)

[Diagrama Entidade Relacionamento 9](#_Toc58702276)

[Alterações sofridas 10](#_Toc58702277)

[Diagrama físico simplificado 11](#_Toc58702278)

[Alterações sofridas 11](#_Toc58702279)

[Tempo gasto 12](#_Toc58702280)

[Horas por Aula e Extra-Aula total 12](#_Toc58702281)

[Distribuição de tarefas para entrega II 12](#_Toc58702282)

[Outros aspetos 13](#_Toc58702283)

# Organização final geral da aplicação

## Ficheiros:

### Aplicação Principal

* main.py Aplicação principal
* adminMenus.py Menus Administrador
* userMenus.py Menus Utilizador
* commonMenus.py Menus Comuns
* config.ini Configuração da Base de Dados

### Outros Ficheiros

* cavaleiro.py Trabalho Gonçalo Cavaleiro
* TablesProcNoData.sql Ficheiro com tabelas e procedure
* TablesProcWithData.sql Ficheiro com BD de teste completa

Para correr a aplicação deve ser executado o ficheiro main.py num terminal. Este ficheiro depende dos ficheiros:

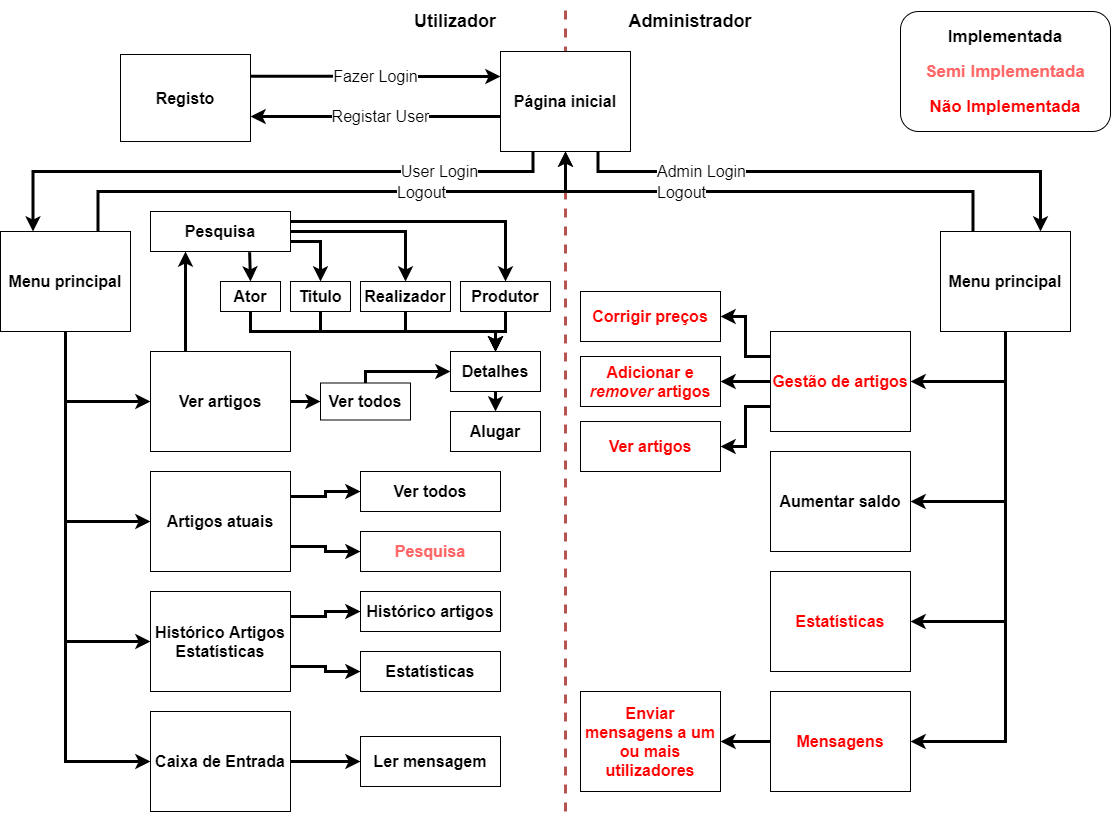
* config.ini
* adminMenus.py
* userMenus.py
* commonMenus.py

A aplicação não necessita de uma base de dados local para funcionar e testar uma vez que foi criada uma na nuvem, porém é possível criar as tabelas com o ficheiro *TablesProcNoData.sql* e criar as tabelas com dados de teste usando o ficheiro *TablesProcWithData.sql.* Estes ficheiros foram exportados diretamente da Base de Dados e são do diferentes do script do ONDA.

Dentro da pasta *ONDA* estão ficheiro JSON e o script SQL para criar as tabelas criado pelo ONDA. Dentro da pasta também estão fotos das tabelas.

Dentro da pasta *Fotos App* estão fotos dos menus da aplicação.

## Menus:



### Menus secundários – Utilizador

#### Ver artigos

Foi reduzido o número de menus, havendo agora apenas dois

* Ver artigos disponíveis
* Pesquisar um artigo
  + Título
  + Ator
  + Produtor
  + Realizador

Após a seleção de um artigo é possível consultar os detalhes do mesmo e fazer o aluguer.

#### Artigos atuais

Neste menu é possível o utilizador ver que artigos tem alugados no momento através de dois menus

* Ver artigos disponíveis
* Pesquisar um artigo
  + Título
  + Ator
  + Produtor
  + Realizador

Dentro deste menu também é possível consultar os detalhes de um artigo.

#### Histórico e estatísticas

Passou a haver dois menus

* Histórico de artigos
* Estatísticas

Dentro do histórico de artigos o utilizador pode ver quais os artigos que alugou, consultar os detalhes e volta a alugar um artigo.

Nas estatísticas o utilizador pode ver quanto já gastou no total bem como já gastou em cada uma das seguintes categorias

* Séries
* Filmes
* Documentários

#### Caixa de Entrada

Não houve alterações a este menu,

* + Ver a caixa de entrada (Existe distinção entre lida e não lida)
  + Ler a mensagem

### Menus secundários – Administrador

#### Aumentar saldo

Neste menu o administrador consegue aumentar o saldo a um utilizador, foi definido um limite por carregamento de saldo.

Os restantes menus não sofreram alterações ou então não foram implementados.

## Fotos interface:

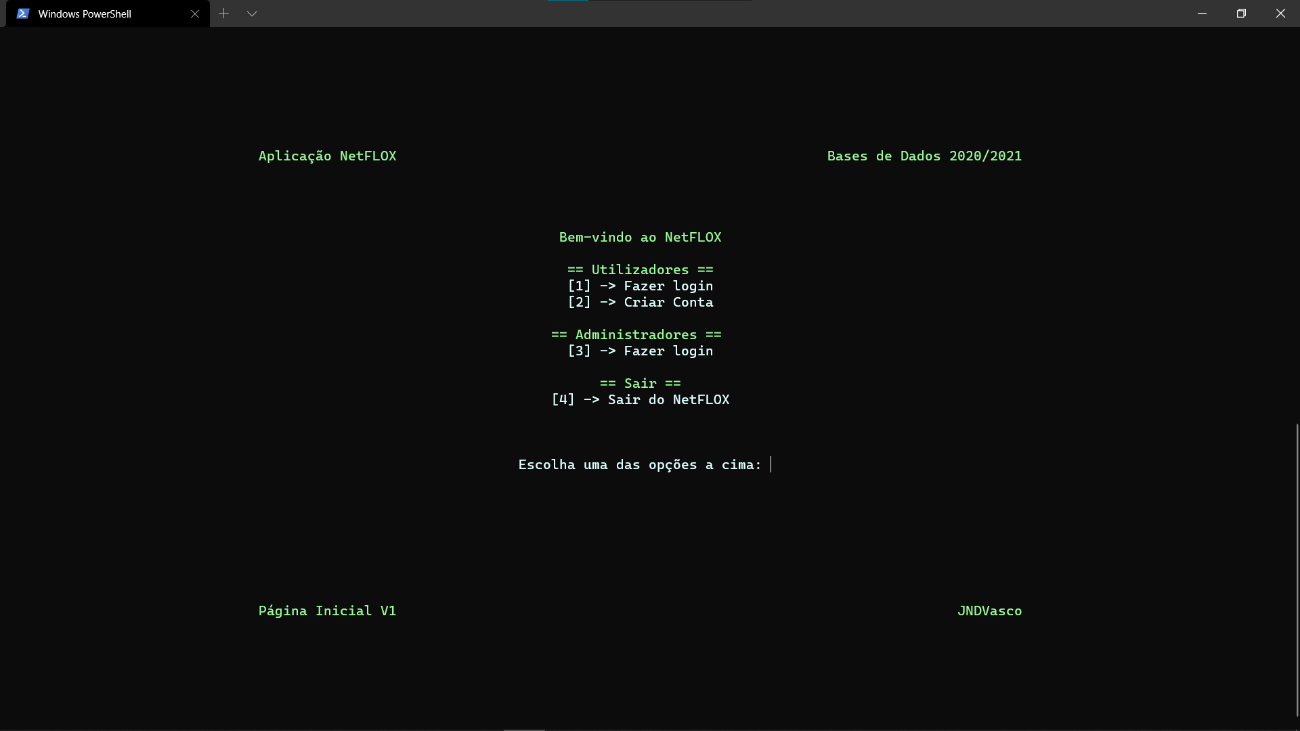


Figura 1 - Página inicial

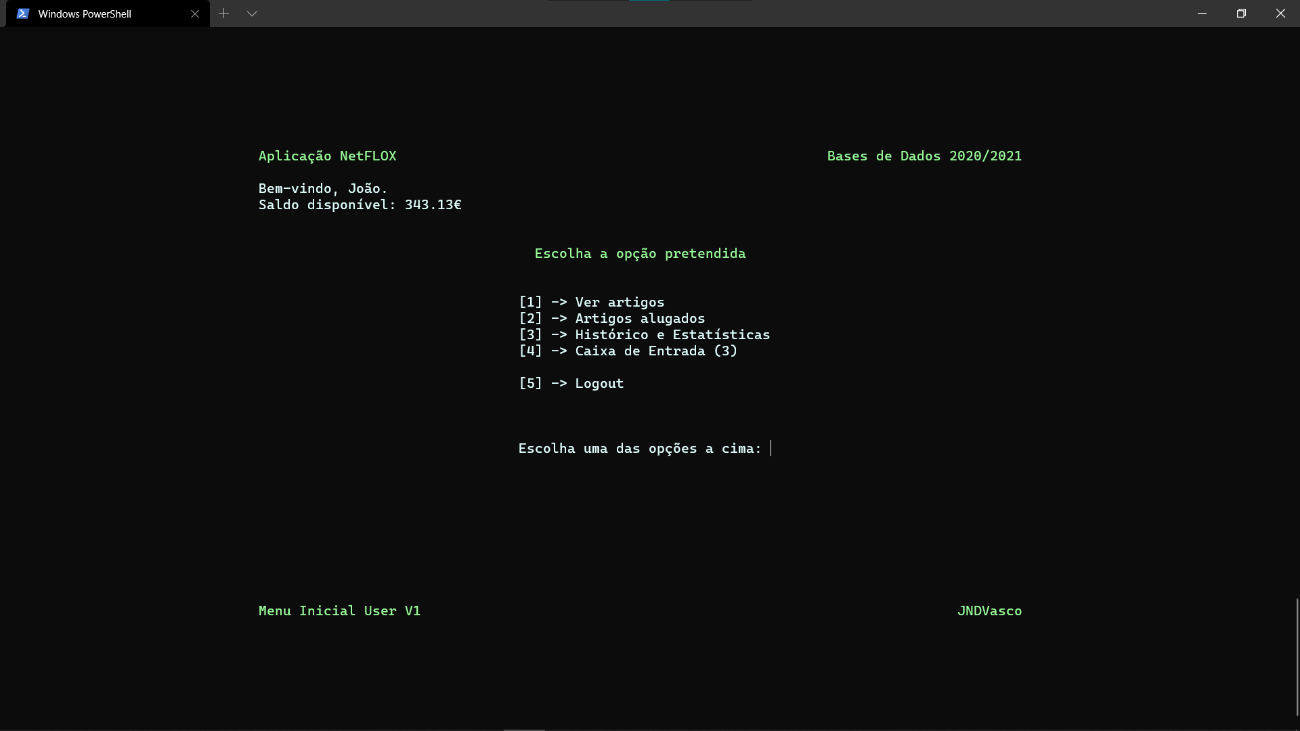


Figura 2 - Menu utilizador



Figura 3 - Menu de pesquisa de artigos

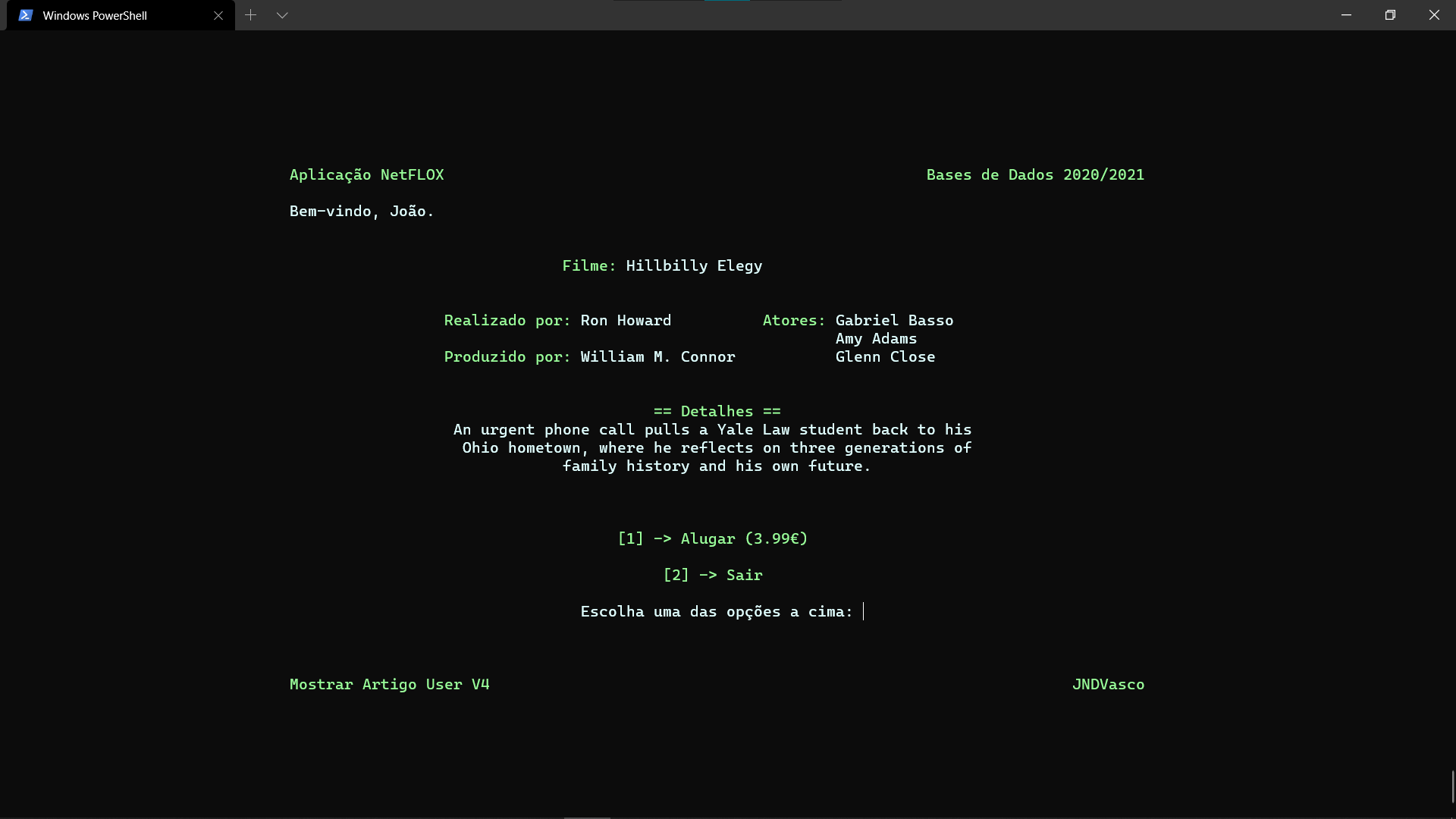


Figura 4 - Detalhes de um artigo



Figura 5 - Aluguer de um artigo

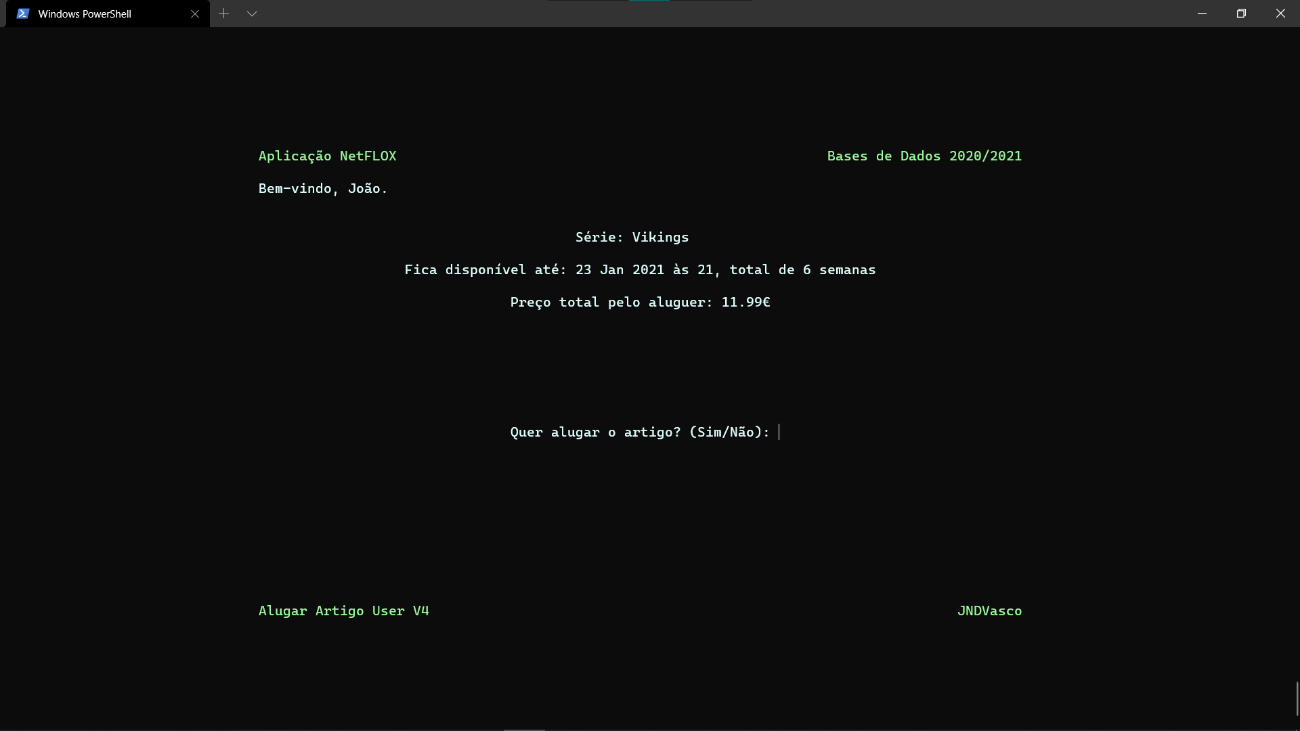


Figura 6 - Caixa de Entrada

# Diagrama Entidade Relacionamento

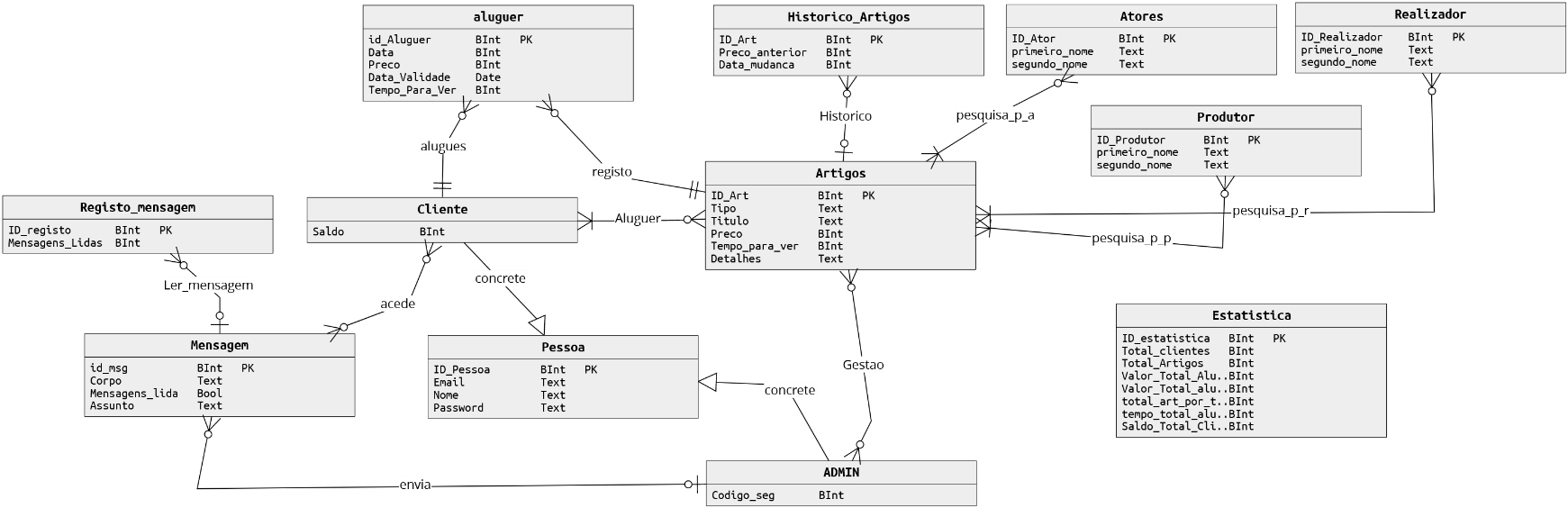


Figura 6 - Diagrama Entidade - Relacionamento

## Alterações sofridas

### Novas entidades:

* Super-Entidade Pessoa
* Produtor, Realizador e Atores
* Aluguer
* Registo mensagem

Foi criada um Super-Entidade Pessoa de forma a dividir o cliente e o admin em tabelas separadas, mas uma vez que estes partilhavam campos optamos por criar uma Super-Entidade em modo Concrete.

Foram criadas tabelas separadas para os atores, produtores e realizadores. Desta forma é mais simples adicionar mais que um de cada tipo a um artigo, tento também o benefício de evitar entradas repetidas.

Foi criada uma tabela aluguer de forma a registar os artigos alugados por cada cliente. Esta tabela contém a data em que o artigo foi alugado e até quando está alugado, bem como o preço gasto quando foi feito o aluguer.

A tabela Registo Mensagens foi criada para manter um registo das mensagens já lidas.

### Entidades e relações removidas

* Histórico Clientes
* Relação entre ADMIN e Cliente para aumento de saldo
* Relação entre ADMIN e Tabela de Estatística

# Diagrama físico simplificado

Figura 7 - Diagrama físico simplificado

## Alterações sofridas

### Novas entidades

* Aluguer
* Registo Mensagem
* Artigos\_atores, Atores
* Artigos realizador, realizador
* Artigos produtor, produtor

### Entidades e relações removidas:

* Entidade Historico\_clientes e respetivas relações
* Relação Aumentar\_Saldo
* Relação ver\_est
* Entidade e respetivas relações Artigos\_Historico\_Artigos
* Admin\_mensagem

Estas alterações devem se às mudanças anteriormente explicadas.

# Distribuição de tarefas para entrega II

|  |  |
| --- | --- |
| Tarefa | Membro do grupo |
| Desenvolvimento do relatório | João Vasco, Gonçalo Cavaleiro |
| Tabelas | G.Cavaleiro |
| Conexão base de dados | João Vasco |
| Procedimento e Transação | João Vasco |
| User CLI[[1]](#footnote-1) | João Vasco |
| Aumenta saldo admin | João Vasco |
| Resto interface Admin | Não realizado |
| Código principal | João Vasco |

# Tempo gasto

## Horas por Aula e Extra-Aula total

## Horas Entrega II

# Outros aspetos

## Nota de Gonçalo Cavaleiro

No âmbito da realização deste projeto, eu, Gonçalo Gouveia Cavaleiro, reconheço que fui negligente na realização do mesmo. Tendo feito muito pouco ou quase nada do que era suposto. Desta forma peço que o meu colega de trabalho não seja prejudicado por minha causa, tendo em conta que ele demonstrou ter conhecimentos da linguagem de programação Python, e com os comandos aprendidos em SQL. Ao dizer isto, tenho perfeita consciência que a minha nota será influenciada pela negativa, mas perante as circunstâncias é o que se tem de fazer.

1. Do inglês “command-line interface” [↑](#footnote-ref-1)