

Relatório do Projeto – Etapa 2

Licenciatura em Engenharia Informática 3º ano
Laboratório de aplicações web e base de dados

Autores

Gonçalo Lourenço Costa - 76131

Moisés Gomes dos Santos – 73876

João Miguel Moreira Marques – 77209

João Guedes Barbosa - 76062

Índice

1.	Introdução	1
2.	Enquadramento Teórico	2
3.	Desenvolvimento.....	3
	Requisitos funcionais:	3
	Requisitos não funcionais:	4
	Diagrama de Casos-de-uso:.....	5
	Diagrama Entidade-Relacionamento (E-R):	6
	Mapeamento entre o modelo E-R para o modelo relacional	7
	Implementação do modelo físico da base de dados recorrendo à linguagem SQL.....	8
	Diagrama da base de dados (apresentação gráfica das tabelas e suas relações)	10
	Especificação das interfaces recorrendo a mockups (Backoffice)	11
	Especificação das interfaces recorrendo a mockups (Frontoffice)	12

1. Introdução

Este trabalho, desenvolvido no âmbito da Unidade Curricular de Laboratório de Aplicações Web e Bases de Dados, representa um desafio emocionante no campo da Engenharia de Software e Engenharia Web. O objetivo central deste projeto é criar uma aplicação web que sirva como plataforma para uma rede social, onde os utilizadores podem se conectar, comunicar e compartilhar informações.

O projeto está dividido em três fases distintas de desenvolvimento, cada uma com sua própria avaliação. A primeira fase, aborda a análise dos requisitos de dados do sistema, a especificação do modelo conceptual de dados usando diagramas Entidade-Relacionamento (E-R) e a análise dos requisitos funcionais do sistema, com a especificação do modelo funcional através de diagramas de Casos-de-Uso.

A segunda fase, abordará o mapeamento entre o modelo conceptual de dados e o modelo relacional, bem como a implementação do modelo físico da base de dados através da linguagem SQL. Além disso, os alunos serão desafiados a criar especificações de interfaces de utilizador para o backoffice e o frontoffice utilizando mockups.

Por fim, a terceira fase, aborda a base na implementação da integridade da base de dados do sistema e na implementação da lógica funcional que conecta as interfaces de utilizador à base de dados.

Neste projeto, temos a oportunidade de aplicar os conhecimentos adquiridos nas áreas de Engenharia de Software, Bases de Dados e Engenharia Web para criar uma aplicação web robusta e funcional.

2. Enquadramento Teórico

De modo geral, o processo de levantamento de requisitos de um software é responsável por compreender as necessidades dos usuários e do ambiente de aplicação.

Neste trabalho, dividimos os requisitos em funcionais e não funcionais. Os funcionais tendem a ter uma linguagem mais natural, muitas vezes servem para que se entenda o que o cliente quer, e para isso usa-se uma linguagem mais corrente. Mostram o que o site deverá fazer e não como o vai fazer. Já os não funcionais tendem a ter uma linguagem um pouco mais técnica, falam de assuntos geralmente mais direcionados aos desenvolvedores. Diversos estudos demonstram que o insucesso na engenharia de requisitos é a principal causa do fracasso de projetos de software. [Arlow e Neustadt 2002]. E por isto, é importante uma grande reflexão sobre os requisitos, uma boa discussão entre desenvolvedores e clientes para que o projeto seja um sucesso.

Os diagramas de casos de uso são: a representação das funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema e, que com ele interagem (Bezerra, 2007). Os diagramas de casos de uso são constituídos por atores, casos de uso e as respetivas relações entre eles. São muito bons para os desenvolvedores saberem que caminho devem seguir, mas também para mostrarem e discutirem com os clientes sobre o que o sistema deve fazer. A sua especificação também é relevante, clarifica o que no que consiste realmente o caso de uso, e quais os diversos caminhos que podem ser tomados.

O modelo Entidade-Relacionamento é um modelo conceitual utilizado na modelagem de dados, que permite representar as entidades de um sistema, seus atributos e os relacionamentos entre elas. Ele é baseado na ideia de que tudo no sistema é uma entidade, como uma pessoa, objeto ou evento e essas entidades têm relacionamentos uns com os outros. Permite representar graficamente as entidades, atributos e relacionamentos em um diagrama, facilitando a compreensão da estrutura do banco de dados. A estrutura deste modelo representa as chaves primárias a sublinhado e a riscado as chaves estrangeiras, adicionando a esta última a respetiva referência.

3. Desenvolvimento

Requisitos funcionais:

RF1. **Fazer registo** - O sistema deve permitir que o utilizador seja capaz de registar os seus dados para mais tarde iniciar sessão.

RF2. **Verificar Conta** - O sistema deve enviar ao utilizador um email de confirmação de conta.

RF3. **Alterar registos** - O sistema deve permitir que o utilizador altere os seus dados de registo, caso necessário

RF4. **Iniciar Sessão** - O sistema deve permitir o utilizador utilizar os seus dados de registo para dar início de sessão.

RF5. **Criação de grupos**- O sistema deve permitir o utilizador criar grupos.

RF6. **Publicação de posts**- O sistema deve permitir a publicação de posts tais com: imagens, texto, vídeo.

RF7. **Comentários a posts**- O sistema deve permitir ao utilizador autenticado comentar posts de grupo públicos e aos quais pertence.

RF8. **Partilhar Posts**- O sistema deve permitir ao utilizador autenticado a partilha de posts entre utilizadores autenticados.

RF8. **Partilha de ficheiros**- O sistema deve permitir aos administradores decidir se quer guardar os ficheiros partilhados numa pasta do projeto ou se guarda os ficheiros na base de dados em base64.

RF9. **Pesquisar grupos**- O sistema deve permitir ao utilizador a pesquisa de grupos e caso o mesmo seja de livre acesso então terá logo acesso a todo o conteúdo, caso o grupo seja restrito o utilizador deverá esperar pela aprovação de acesso e só depois ficara com acesso ao conteúdo caso seja aceite.

RF10. **Bloquear comunicações**- O sistema deve permitir ao utilizador autenticado que bloqueie comunicações com outro utilizador autenticado.

RF11. **Pesquisar dentro do site**- O sistema deve permitir ao utilizador autenticado pesquisar palavras ou outros utilizadores dentro do site através de uma barra de pesquisa.

Requisitos não funcionais:

RNF1. O sistema deverá ser programado em SQL/HTML.

RNF2. O sistema deverá utilizar a tecnologia MySQL para fazer a criação e gestão da DataBase.

RNF3. O site deve ter uma cláusula de proteção de dados.

RNF4. O sistema deverá consumir poucos recursos do navegador.

RNF5. O sistema deverá dispor de 3 tipos de utilizador: público, autenticado, administrador.

RNF6. O sistema deve guardar os ficheiros na base de dados em base64 se assim for solicitado.

RNF7. No registo do usuário o email do respetivo deve ser válido.

RNF8. Ao registar-se, o usuário deve escolher uma senha que contenha obrigatoriamente uma letra maiúscula, um número e um símbolo, com mínimo de 8 máximo de 30 caracteres.

RNF9. A interface deve ser de fácil utilização.

RNF10. O sistema deverá reconhecer comentários impróprios e eliminá-los

Diagrama de Casos-de-uso:

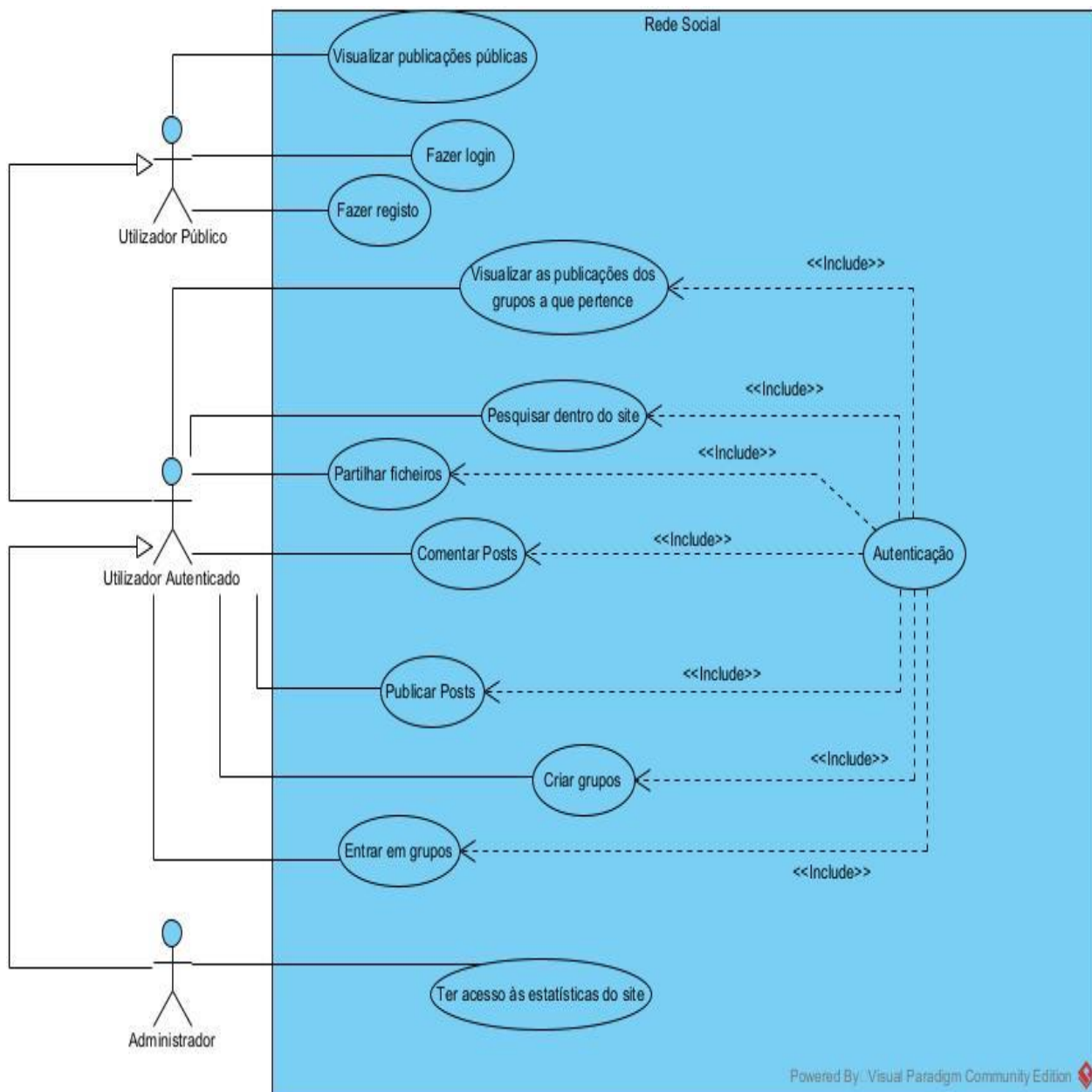


Figura 1- Diagrama Casos de Uso

Diagrama Entidade-Relacionamento (E-R):

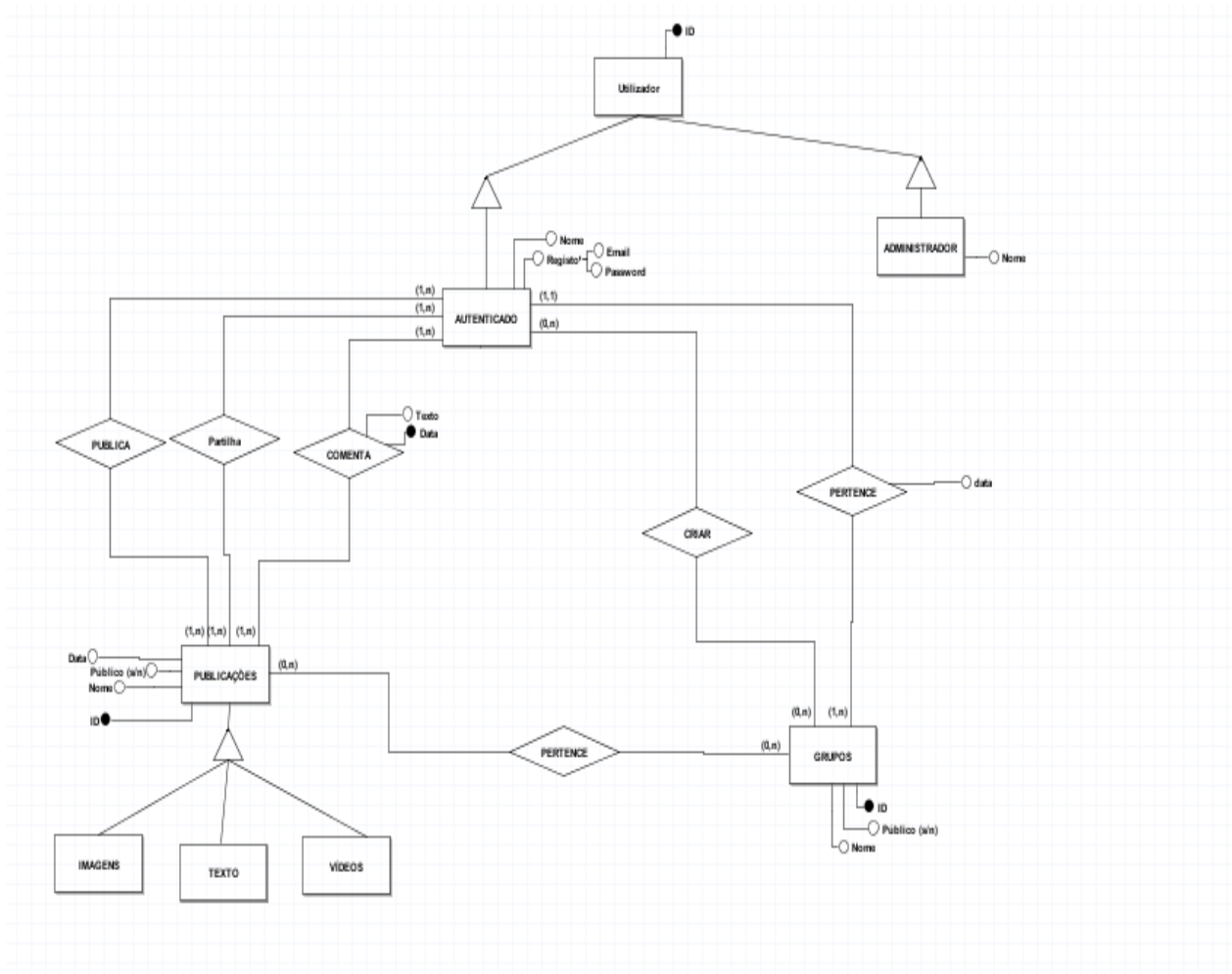


Figura 2-Diagrama E-R

Mapeamento entre o modelo E-R para o modelo relacional

Utilizador (**ID**)

Administrador (ID_Utilizador, nome)

ID_Utilizador referencia Utilizador

Autenticado (ID_Utilizador, Nome, Registo_email, Registo_password)

ID_Utilizador referencia Utilizador

Publica (ID_Autenticado, id_publicações)

ID_Autenticado referencia Autenticado

id_publicacoes referencia Publicações

Comenta (ID_Autenticado, id_publicações, Data_Comentario, texto)

ID_Autenticado referencia Autenticado

id_publicações referencia Publicações

Pertence (id_autenticado, id_grupos, data)

id_autenticado referencia Autenticado

id_grupos referencia Grupos

Pertence2(ID_publicações, id_grupos)

ID_Publicações referencia Publicações

ID_Grupos referencia Grupos

Criar (id_grupo, id_Autenticado)

id_grupos referencia Grupos

id_Autenticado referencia Autenticado

Partilha (id_publicações)

id_publicacoes referencia Publicações

Publicações (**ID**, Publico, Nome, Data)

Imagens (id_publicações)

Texto (id_publicações)

Vídeos (id_publicações)

Grupos (**ID**, Publico, Nome)

Implementação do modelo físico da base de dados recorrendo à linguagem SQL

use master

create database Lab

use Lab

```
Create TABLE Utilizador(  
ID      Integer      NOT NULL PRIMARY KEY,  
);
```

```
CREATE TABLE Administrador(  
Nome VARCHAR(50) NOT NULL,  
ID_Utilizador INTEGER,  
FOREIGN KEY (ID_Utilizador) REFERENCES Utilizador(ID),  
);
```

```
CREATE TABLE Autenticado(  
ID      Integer NOT NULL PRIMARY KEY,  
Nome VARCHAR(50) NOT NULL,  
ID_Utilizador INTEGER,  
Registo_Email VARCHAR(50),  
Registo_Password VARCHAR(30),  
FOREIGN KEY (ID_Utilizador) REFERENCES Utilizador(ID),  
);
```

```
CREATE TABLE Publicacoes(  
Data_Publicacao Integer,  
ID Integer NOT NULL PRIMARY KEY,  
Publico CHAR NOT NULL,  
check(Publico IN ('S', 'N')),  
Tipo VARCHAR NOT NULL,  
check(Tipo IN ('Imagens', 'Texto', 'Videos')),  
);
```

```
CREATE TABLE Publica(  
ID_Autenticado INTEGER,  
ID_Publicacoes INTEGER,  
FOREIGN KEY (ID_Autenticado) REFERENCES Autenticado(ID),  
);
```

```
CREATE TABLE Partilha(  
ID_Autenticado INTEGER,  
ID_Publicacoes INTEGER,  
FOREIGN KEY (ID_Autenticado) REFERENCES Autenticado(ID),  
);
```

```
CREATE TABLE Comenta(  
Data_Comentario Integer,  
ID_Autenticado INTEGER,  
ID_Publicacoes INTEGER,  
Texto VARCHAR(100),  
FOREIGN KEY(ID_Autenticado) REFERENCES Autenticado(ID),  
);
```

```
CREATE TABLE Grupos(  
ID Integer NOT NULL PRIMARY KEY,  
Nome VARCHAR(50),  
Publico CHAR NOT NULL,  
check(Publico IN ('S', 'N')),  
);
```

```
CREATE TABLE Criar(  
ID_Grupos INTEGER,  
ID_Autenticado INTEGER,  
FOREIGN KEY(ID_Grupos) REFERENCES Grupos(ID),  
FOREIGN KEY(ID_Autenticado) REFERENCES Autenticado(ID),  
);
```

```
CREATE TABLE Pertence1(  
ID_Grupos INTEGER,  
ID_Publicacoes INTEGER,  
FOREIGN KEY(ID_Grupos) REFERENCES Grupos(ID),  
FOREIGN KEY(ID_Publicacoes) REFERENCES Publicacoes(ID),  
);
```

```
CREATE TABLE Pertence2(  
ID_Grupos INTEGER,  
ID_Autenticado INTEGER,  
FOREIGN KEY(ID_Grupos) REFERENCES Grupos(ID),  
FOREIGN KEY(ID_Autenticado) REFERENCES Autenticado(ID),  
);
```

Diagrama da base de dados (apresentação gráfica das tabelas e suas relações)

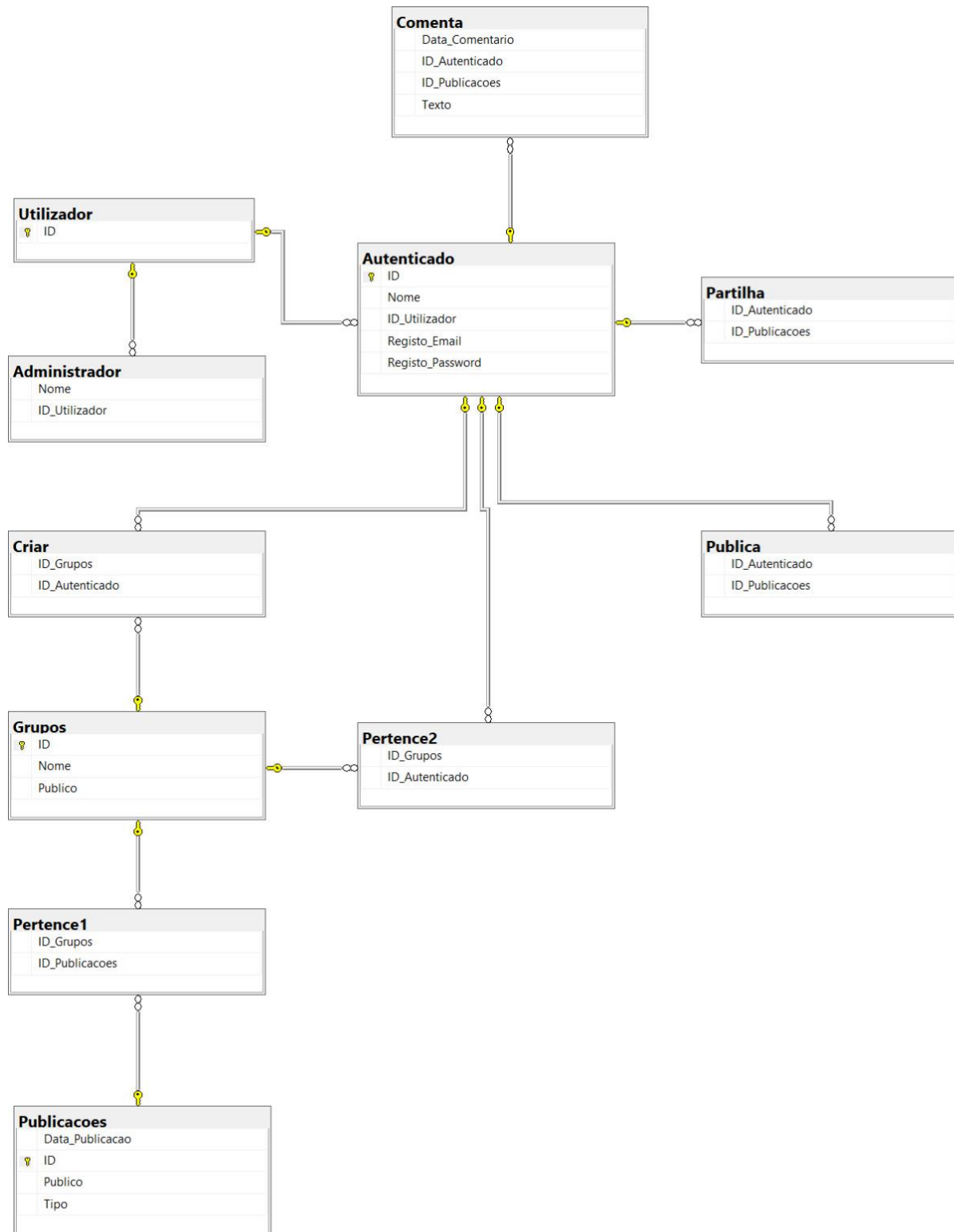


Figura 3-Diagrama da Base de Dados

Especificação das interfaces recorrendo a mockups (Backoffice)

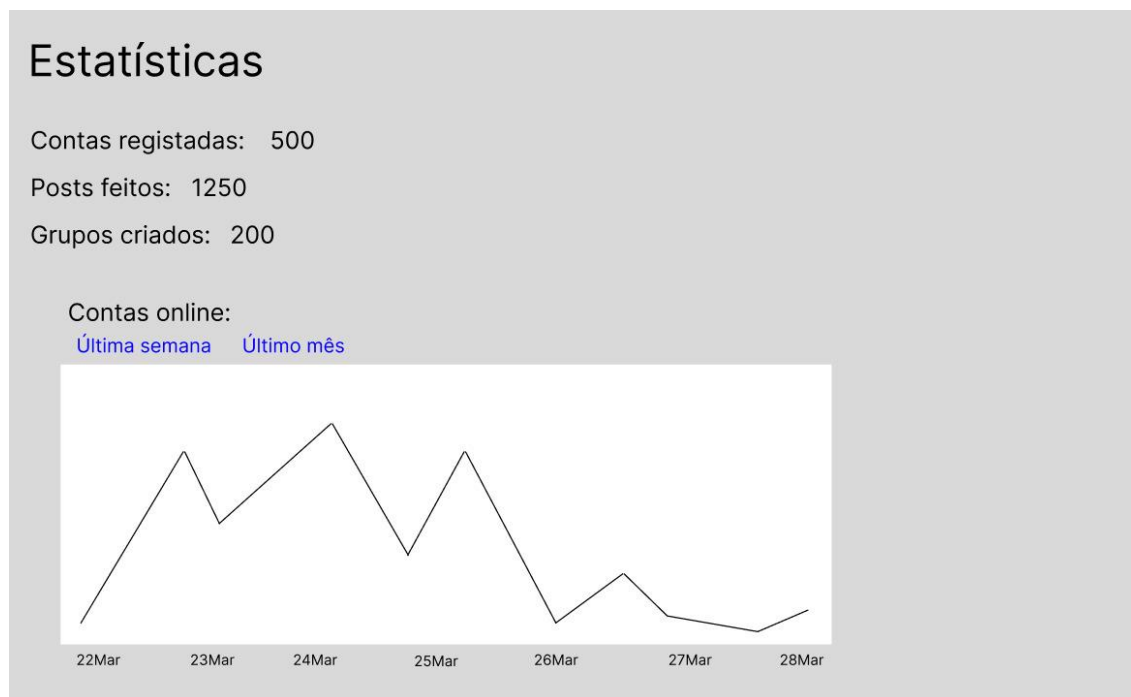


Figura 4-Estatísticas

Lista de administradores

Nome: <input type="text"/>	ID: <input type="text"/>	Email: <input type="text"/>
Administrador 1	854865589	administrador1@gmail.com
Administrador 2	907646572	administrador2@gmail.com
Administrador 3	275587235	administrador3@gmail.com
Administrador 4	765362340	administrador4@gmail.com
Administrador 5	576223478	administrador5@gmail.com
Administrador 6	356652352	administrador6@gmail.com

Figura 5-Lista de Administradores

Especificação das interfaces recorrendo a mockups (Frontoffice)

Frame 1



[Voltar à página inicial](#)



Email/Utilizador:

Password:

[Iniciar Sessão](#)

[Registe-se aqui](#)

Figura 6- Log In



[Voltar à página inicial](#)



Registo

Primeiro Nome:

Último Nome:

Data Nascimento:

Palavra Passe:

Confirmar Palavra Passe:

[Regista-te](#)

[Faça aqui o Log In!](#)

Figura 7-Registo

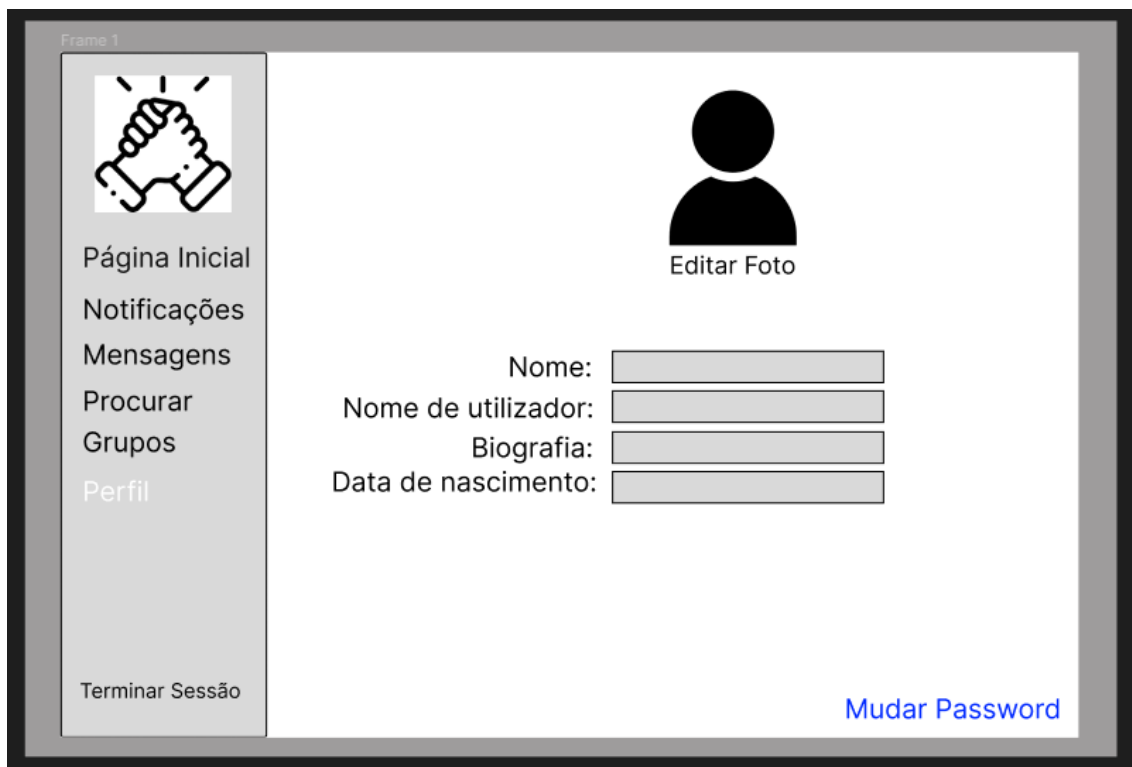


Figura 8-Editar Perfil

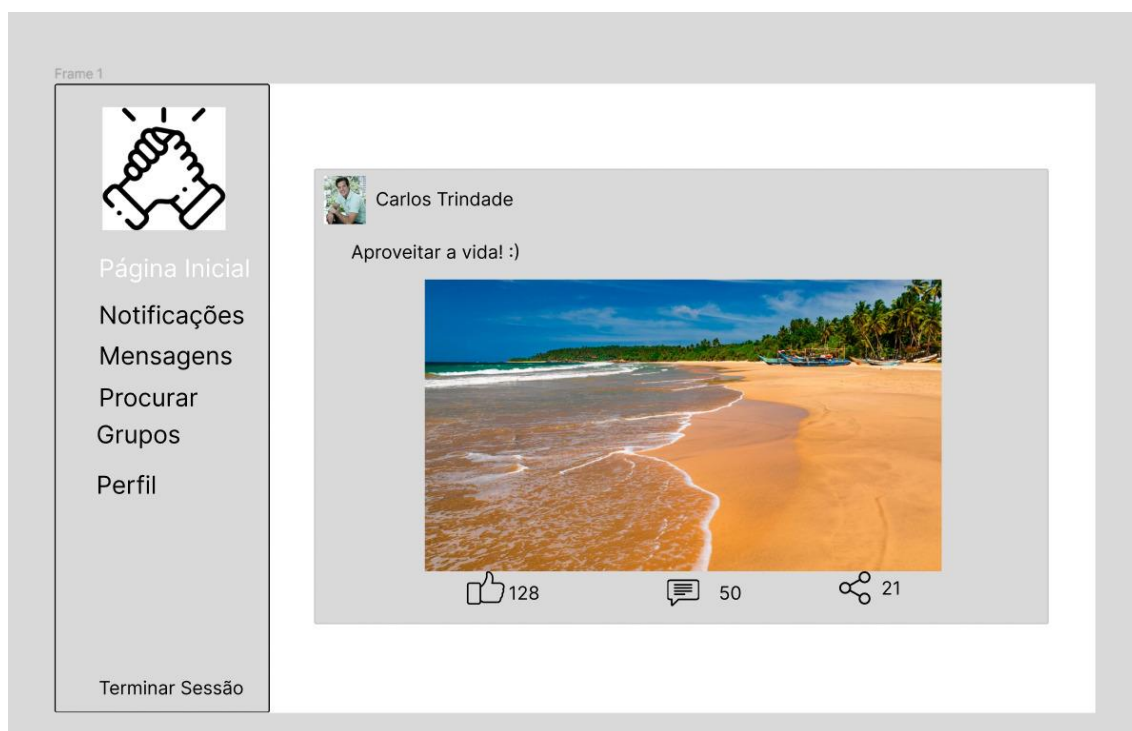


Figura 9-Página Inicial



Figura 10-Notificações

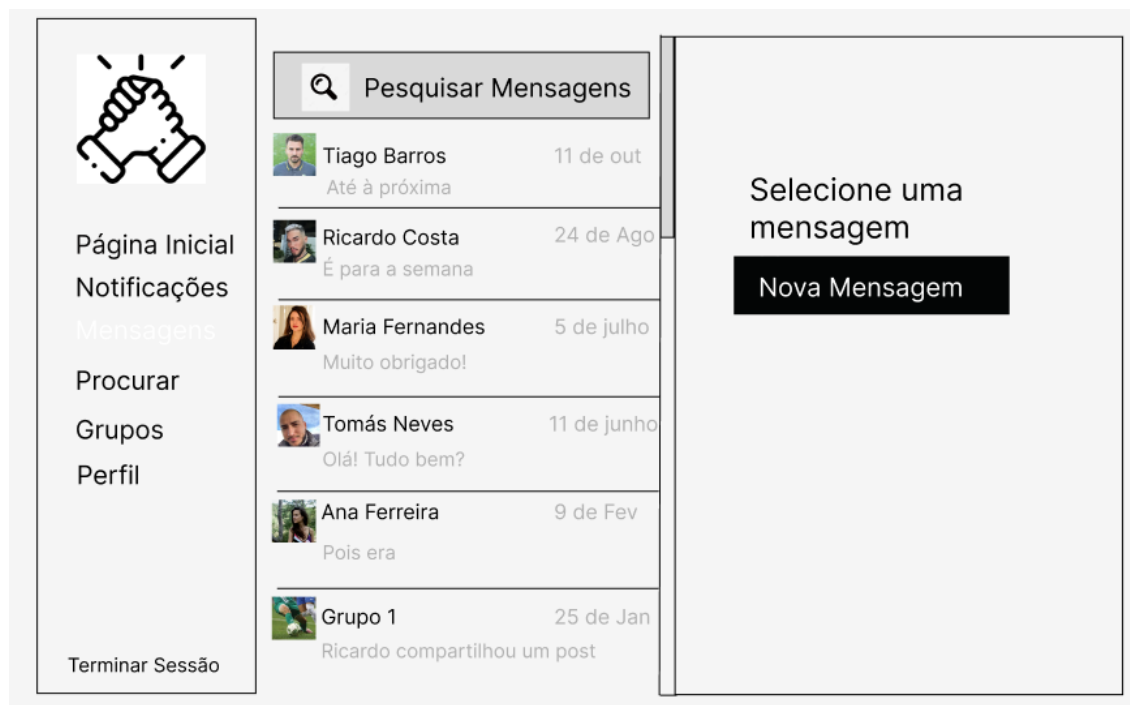


Figura 11-Mensagens

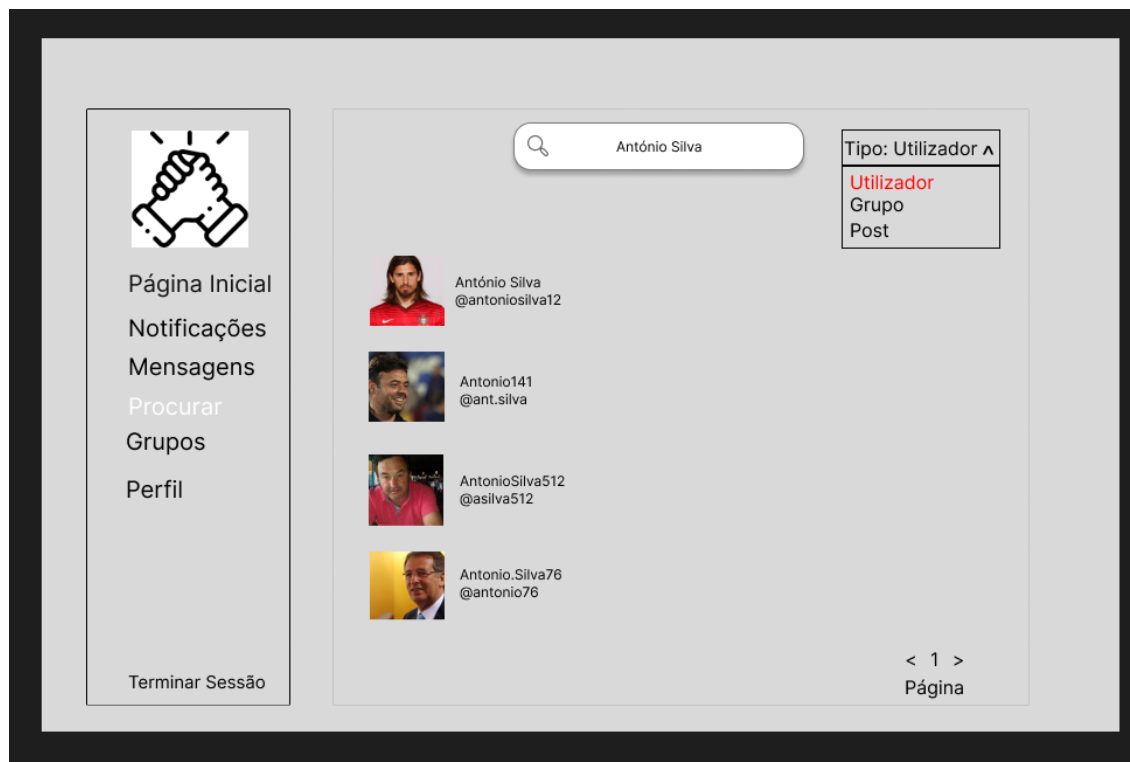


Figura 12-Procurar

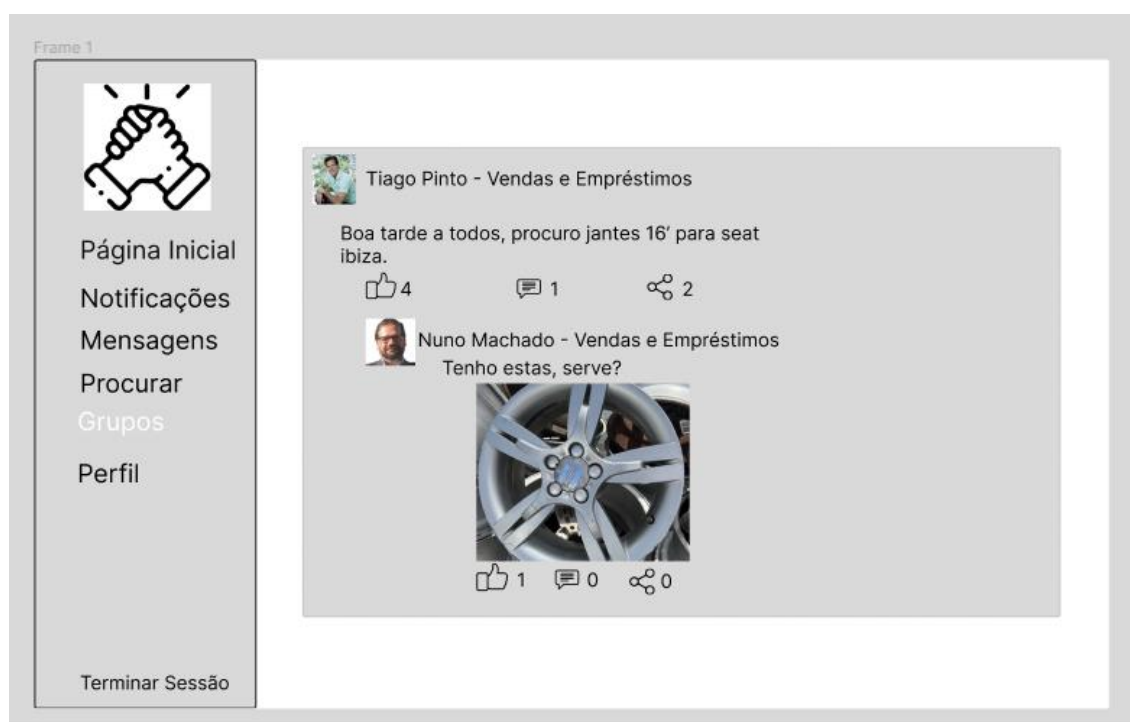


Figura 13-Grupos



Figura 14-Perfil do Utilizador