

***Relatório – Bases de Dados***

***“Rede Social”***

Trabalho Realizado por:

Marta Lopes, ei12106;

João Nogueira, up201303882;

Luís Oliveira, up201304515.

# Página de Rosto

**Título do Projeto:** Base de dados – Rede Social;

**Curso:** Mestrado Integrado em Engenharia Informática e Computação;

**Unidade Curricular:** Bases de Dados

**Ano letivo:** 2014/15;

**Grupo:**

* Marta Lopes, ei12106;
* João Nogueira, up201303882;
* Luís Oliveira, up201304515;

Turma 5;

**Data de Entrega:** 20 de Abril de 2015.

Índice

[Página de Rosto 2](#_Toc417237513)

[Descrição da Base de Dados da “Rede Social” 4](#_Toc417237514)

[Descrição dos Atributos 5](#_Toc417237515)

[Perfil 5](#_Toc417237516)

[Evento 5](#_Toc417237517)

[Grupo 5](#_Toc417237518)

[Post 5](#_Toc417237519)

[Chat 5](#_Toc417237520)

[Mensagem 5](#_Toc417237521)

[Amizade 5](#_Toc417237522)

[Naturalidade 5](#_Toc417237523)

[Reputação 5](#_Toc417237524)

[Diagrama de Classes UML 6](#_Toc417237525)

[Modelo Relacional 7](#_Toc417237526)

[Instruções de LDD-SQL 8](#_Toc417237527)

[Criação da Base de Dados 8](#_Toc417237528)

[Preenchimento da Base de Dados 11](#_Toc417237529)

[Observações finais 14](#_Toc417237530)

# Descrição da Base de Dados da “Rede Social”

Para este projeto optámos por fazer uma base de dados de uma rede social. Usamos como ponto de partida a famosa rede social *Facebook*¸ no entanto fizemos várias alterações e adaptações para que o UML não ficasse demasiado complexo.

Na nossa rede Social cada utilizador possui um **Perfil**. Esse perfil tem como atributo o ID de **Naturalidade**, uma outra classe pertencente à nossa Base de dados. Como forma de comunicar entre eles, os perfis podem criar **Chats**. Cada **Chat** pode ser constituído por vários utilizadores, podendo cada um deles introduzir no **Chat** as **Mensagens** (outra classe da nossa Base de Dados) que assim desejar.

Como forma de manter o conhecimento das relações entre os vários utilizadores, existe a classe **Amizade**. Um utilizador pode pedir qualquer outro em **Amizade** cabendo a este aceitar ou não.

Os utilizadores podem também criar e pertencer a **Grupos**. Estes são também uma forma de comunicação entre os vários **Perfis**. Pertencendo a um grupo, o utilizador comunica através de **Posts** que pode fazer nesse mesmo **Grupo**. Esses **Posts** podem não só ser feitos em grupos, como noutros perfis ou em **Eventos**. Os **Eventos** são também uma classe da nossa Base de dados que tem um Perfil criador ao qual podem aderir vários outros perfis.

Os **Perfis** podem atribuir **Reputação**, seja ela positiva ou negativa, a outros **Perfis** ou a **Posts**. Estes dois destinos da **Reputação** têm um atributo Inteiro que é alterado sempre que lhe é atribuída **Reputação**.

# Descrição dos Atributos

## Perfil

* Avatar - URL;
* Contacto - INTEGER;
* Data de nascimento - DATE;
* Morada - TEXT;
* idNaturalidade - INTEGER;
* Nome - INTEGER;
* Reputação - INTEGER.

## Evento

* Data - DATE;
* Descrição – TEXT;
* Local - TEXT;
* Nome - TEXT.

## Grupo

* Descrição - TEXT;
* idAdministrador - INTEGER;
* Nome - TEXT;
* Privacidade - TEXT;

## Post

* Data - DATE;
* Hora - TEXT;
* idCriador - INTEGER;
* Reputação - INTEGER;
* Texto - TEXT.

## Chat

* Nome - TEXT;
* Número de mensagens – INTEGER;

## Mensagem

* Data - DATE;
* Hora - TEXT;
* Texto - TEXT.

## Amizade

* Aceite - BOOLEAN.

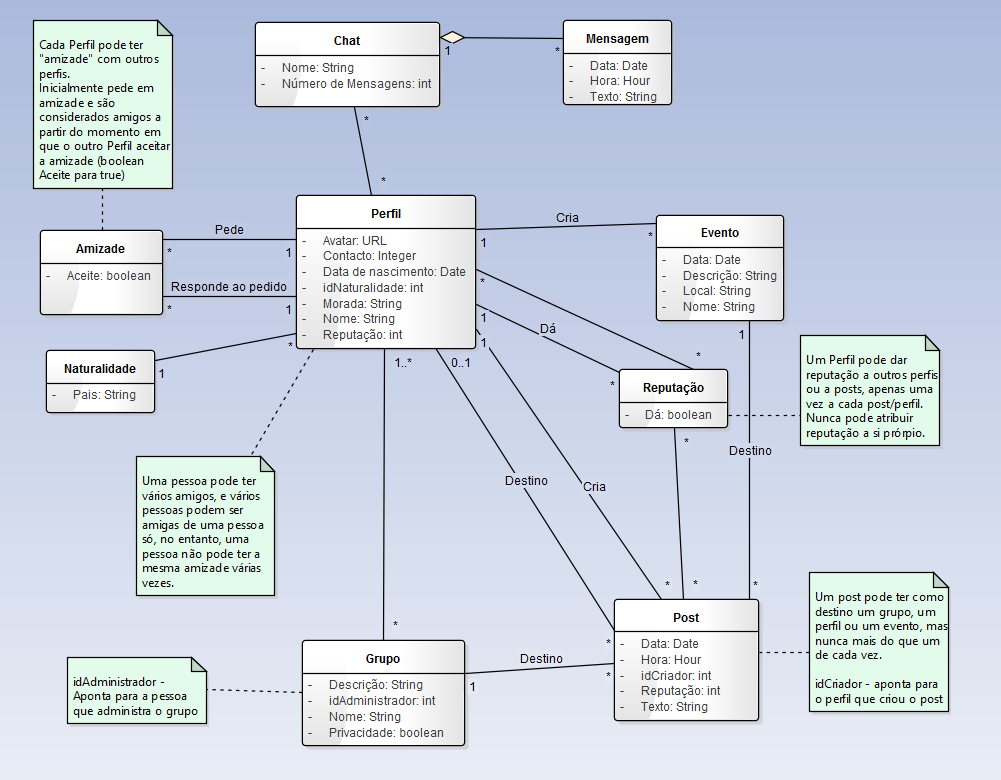
## Naturalidade

* Cidade - TEXT.

## Reputação

* Dá - BOOLEAN.

# Diagrama de Classes UML



# Modelo Relacional

Naturalidade(idNaturalidade, País)

Perfil(idPerfil, idNaturalidade -> Naturalidade, Avatar, Contacto, Data de nascimento, Morada, Nome, Reputação)

Amizade(idPerfilPede -> Perfil, idPerfilAceita -> Perfil, aceita)

Chat(idChat, Nome, Número de mensagens)

Mensagem(idMensagem, idChat -> Chat, Data, Hora, Texto)

Evento(idEvento, idCriador -> Perfil, Data, Descrição, Local, Nome)

Grupo(idGrupo, idAdministrador -> Perfil, Descrição, Nome, Privacidade)

Post(idPost, idCriador -> Perfil, Data, Hora, Reputação, Texto)

PostPerfil(idPost -> Post, idPerfil -> Perfil)

PostGrupo(idPost -> Post, idGrupo -> Grupo)

PostEvento(idPost -> Post, idEvento -> Evento)

PerfilChat(idPerfil - > Perfil, idChat -> Chat)

PerfilGrupo(idPerfil -> Perfil, idGrupo -> Grupo)

Reputação(idReputação, dá)

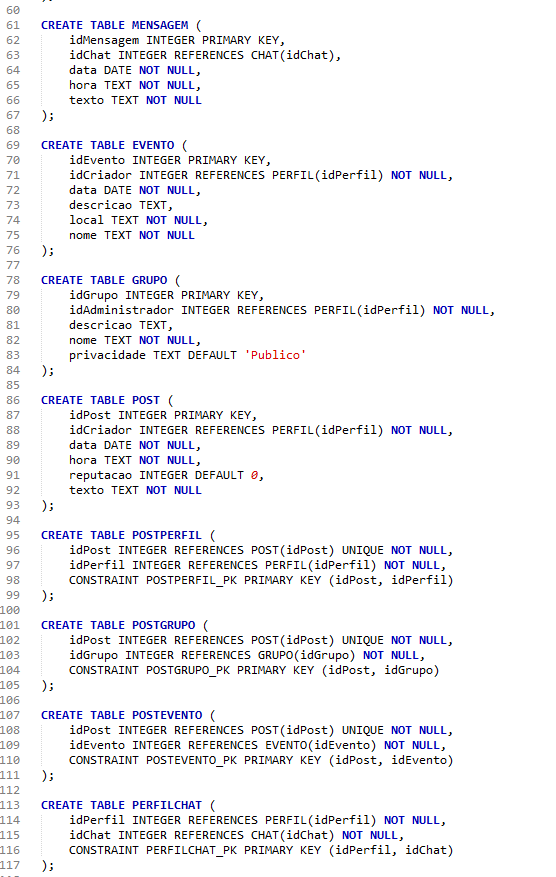
ReputaçãoPerfil(idAtribuiu -> Perfil, idPerfil -> Perfil, idReputação -> Reputação)

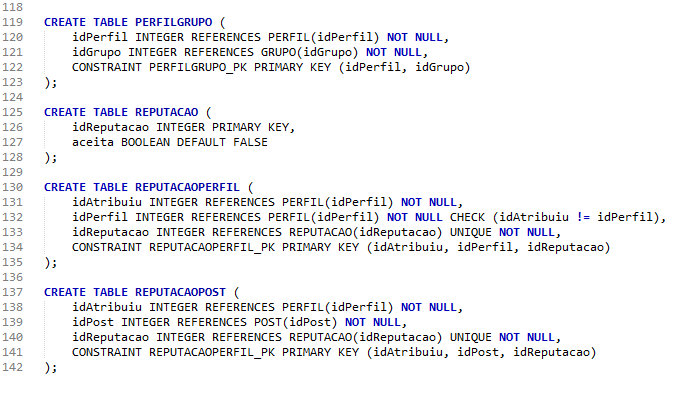
ReputaçãoPost(idAtribuiu -> Perfil, idPost -> Post, idReputação -> Reputação)

# Instruções de LDD-SQL

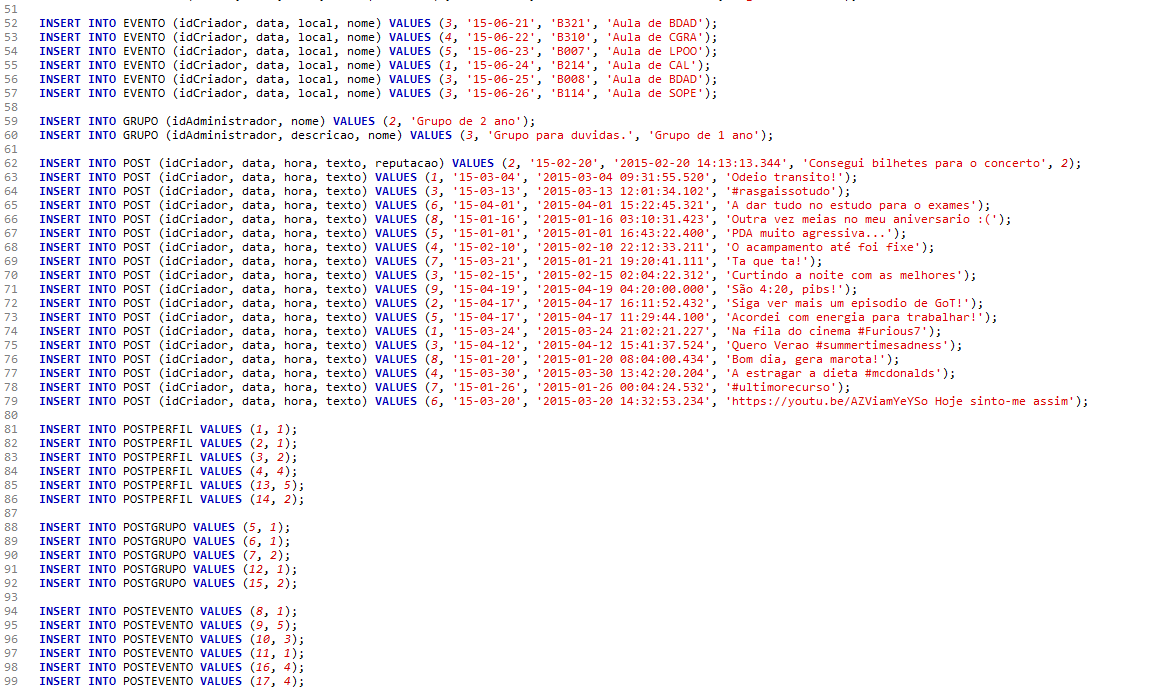
## Criação da Base de Dados

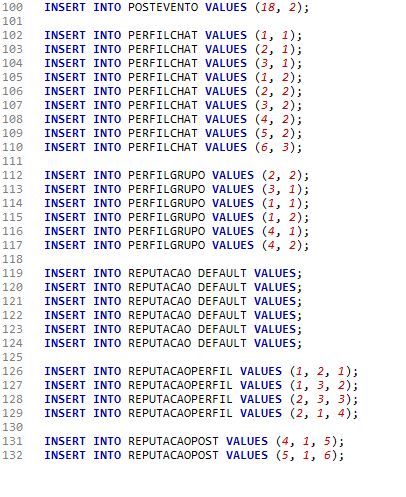






## Preenchimento da Base de Dados





# Observações finais

Foi necessário fazer algumas alterações em relação à primeira entrega, tendo em conta o *Feedback* do dado pelo professor.

Encontramos alguns problemas ao longo da criação da Base de Dados devido ao facto de ainda não aplicarmos *Triggers*. Pode ver-se no código em comentários alguns desses problemas.

Contamos corrigi-los aquando da Terceira entrega do Projeto de Bases de Dados.

Por fim, consideramos que este trabalho serviu para nos incutir mais interesse na área de Bases de Dados e nos permitiu perceber mais aprofundadamente a forma como funcionam.