****

***Bases de Dados***

*Grupo 608*

Nuno Miguel Fernandes Marques – up201708997@fe.up.pt

Pedro Miguel Novais do Vale – up201806083@fe.up.pt

Tiago Miguel Barbosa Marques – up201704733@fe.up.pt

***Descrição da Base dados da aplicação de “chat”***

Neste projeto decidimos fazer uma base de uma aplicação de chat, usando por base aplicações como *Discord* e *WhatsApp.*

Nesta aplicação cada utilizador tem um **perfil** único, existem **chats** privados e chats em **salas** publicas com **mensagens** dos utilizadores.

Perfis podem ser do tipo **grátis** ou **premium**. Perfis grátis tem um limite de emojis e **reações**, perfis premium não têm limite, mas pagam uma mensalidade.

Utilizadores podem fazer pedidos de **amizade** a outros utilizadores, estes pedidos podem ser aceites ou rejeitados. Ao aceitar um pedido de amizade os dois utilizadores tornam-se amigos. Também é possível bloquear pedidos de outros utilizadores.

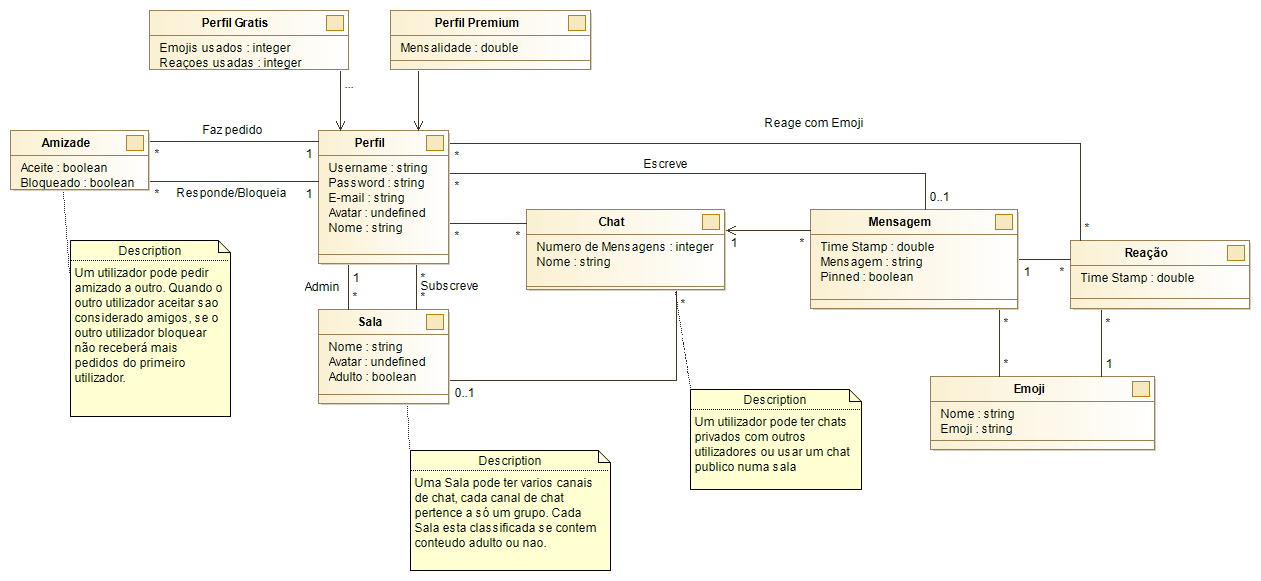
Dois utilizadores com amizade podem trocar mensagens num **chat** privado. As mensagens permitem o uso de emojis. É também possível reagir a mensagens usando um emoji na mensagem escolhida e faz “pin” a mensagens, mantendo mensagens importantes no topo.

Existem também **salas** que os utilizadores podem criar, tornando-se administradores da sala, ou subscrever. Cada sala pode conter vários chats públicos em que qualquer utilizador pode participar e tem as mesmas funcionalidades do chat privado em termos de emojis e reações.

***Atributos***

|  |  |
| --- | --- |
| **Perfil:**   * Username * Password * Endereço e-mail * Avatar * Nome | **Perfil Premium:**   * Mensalidade   **Perfil Grátis:**   * Emojis usados * Reações usadas |
| **Amizade:**   * Aceite * Bloqueado | **Chat:**   * Número de mensagens * Nome |
| **Sala:**   * Nome * Avatar * Adulto | **Mensagem:**   * Timestamp * Mensagem * Pinned * Número de reações |
|  |  |
| **Reação:**   * Timestamp | **Emoji:**   * Nome * Emoji |

***Diagrama UML***



***Esquema Relacional***

**chat (**idchat, nome, idsala->sala)

**subscricaosala (**idperfil->perfil, idsala->sala)

**sala (**idsala, idperfil->perfil, nome, avatar, adulto)

**emoji (**idemoji, nome, emoji)

**mensagem (**idmensagem, idchat->chat, idperfil->perfil, mensagem, tempo, pinned)

**emojimensagem (**idmensagem->mensagem, posicao, idemoji->emoji)

**reacao (**idmensagem->mensagem, idperfil->perfil, idemoji->emoji, tempo)

**amizade (**idperfil1->perfil, idperfil2->perfil, amizade)

**perfil (**idperfil, nome, username, password, email, avatar)

**perfilgratis (**idperfil->perfil, emojis, reacoes)

**perfilpremium (**idperfil->perfil, mensalidade)

***Análise de dependências funcionais***

**perfil:**

idperfil->username, e-mail, nome, password, avatar

username->idperfil, e-mail, nome, password, avatar

e-mail->idperfil, nome, password, avatar

**perfilgratis:**

idperfil->emojis, reacoes

**perfilpremium:**

idperfil->mensalidade

**amizade:**

idperfil1, idperfil2 -> amizade

**chat:**

idchat -> idsala, nome

**emoji:**

idemoji -> nome, emoji

**emojimensagem:**

idmensagem, posicao -> idemoji

**mensagem:**

idmensagem->idchat, idperfil, mensagem, tempo, pinned

**reacao:**

idmensagem, idperfil ->idemoji, tempo

**sala:**

idsala->idperfil, nome, avatar, adulto

***Formas normais***

Para estar na terceira forma normal é necessário não existir dependência transitiva. Verificando a análise de dependências funcionais verificamos que em nenhum caso existe qualquer dependência transitiva, ou seja, nenhum atributo da tabela depende de um atributo que depende de outro atributo.

Todos os casos também estão na forma normal de Boyce-Codd pois o lado esquerdo das dependências são sempre superkeys.

***Restrições***

**Not Null**

Esta restrição foi aplicada a atributos que tem que obrigatoriamente existir para a formação da classe, nomeadamente:

* Nome do ***Chat***
* Nome e Avatar do ***Emoji***
* A string e o timestamp da ***Mensagem***
* A posição na string de ***Emojis nas mensagens***
* O nome, username, password e e-mail do ***Perfil***

**Unique**

Esta restrição foi aplicada a atributos que necessitam não ter valores duplicados, apesar de não serem chaves. Neste caso apenas utilizado no username e e-mail do **perfil**, de maneira a manter um username único e apenas permitir a criação de uma conta por e-mail.

**Primary Key**

Esta restrição foi aplicada a atributos em que não possam existir dois ou mais iguais e que são chaves da sua classe, geralmente aplicado aos ids das classes.

**Check**

Esta restrição foi aplicada de maneira a restringir valores de atributos nos seguintes casos:

* Verificar se **adulto** em **sala**, **pinned** em **mensagem** e **amizade** em **amizade** estão entre 0 e 1, ou seja, **verdadeiro ou falso**.
* Verificar se o atributo **password** em **perfil** tem, pelo menos, **6 caracteres**.

**Foreign Key**

Esta restrição foi aplicada em classes que estão relacionadas com outras classes. Como por exemplo a classe reacao, uma reação está associada a um perfil único, numa mensagem única e usando um emoji único. Logo esta classe tem 3 foreign keys, neste caso idmensagem, idperfil e idemoji. Outros são:

* Idsala em **Chat**
* Idperfil e id sala em **subscricaosala**
* Idperfil em **sala**
* Idchat e idperfil em **mensagem**
* Idmensagem e idemoji em **emojimensagem**
* Idperfil1 e idperfil2 em **amizade**
* Idperfil em **perfilgratis** e **perfilpremium**

***Queries***

Foram criados dez queries com as seguintes características:

**Query1:** apresenta o número total de contas, quer premium (**PerfilPremium**) quer grátis (**PerfilGrátis**), separadamente.

**Query2:** apresentainformação sobre salas (exceto o id) relacionadas com conteúdo adulto, ou seja, qualquer **Sala** onde a variável ‘adulto’ seja 1 (verdadeiro).

**Query3:** revela uma tabela com os nomes (‘perfil.nome’) dos utilizadores que enviaram um pedido de amizade (**Amizade**) ao utilizador Mário Teixeira, com ‘idperfil’ = 3.

**Query4:** obtéminformação sobre salas (exceto o id) cujo o nome (‘sala.nome’) inicie pela letra ‘S’. Apresenta o nome dos administradores (obtidos por perfil.nome de **Perfil**) e a ordenação de cada **Sala** é feita pelo seu nome de forma alfabética. Esta *query* poderia ser usada quando um utilizador estivesse a pesquisar as salas que existem e começa a escrever um ‘s’ e começam-lhe logo a aparecer resultados.

**Query5:** obtém os chats privados, ou seja, qualquer **Chat** onde ‘chat.idsala’ seja NULL. Estes são os chats supostamente gerados quando se forma uma amizade, sendo restrito a dois utilizadores (não pertencentes a uma sala).

**Query6:** apresentatodas as mensagens (**Mensagem**) enviadas entre as 20:00 e as 23:59. Como não é feita a distinção dos dias esta interrogação preocupa-se apenas com esse intervalo de tempo (‘mensagem.tempo’). As mensagens são depois ordenadas pelo nome do chat a que pertencem (chat.nome).

**Query7:** *soma dos emojis usados na sala 3*, em que verificamos a relação entre **EmojiMensagem** e **Mensagem** e, por sua vez, **Chat**, onde se utiliza o método COUNT para manter o registo de emojis usados na sala 3, ordenado por ordem decrescente de utilização.

**Query8:** *percentagem de mensagens que o utilizador id 2 manda e que recebem reações*, fazendo a verificação da relação entre **Mensagem** e **Perfil**, utilizando um COUNT para saber o número de mensagens totais mandadas. Também fazemos a relação entre a **Reação** e o **Perfil**, numa subquerie no Select, utilizando um COUNT para obter o total de reações às suas mensagens. Dividindo este último pelo primeiro, obtém-se a percentagem.

**Query9:** *utilizadores que tenham um número de mensagem maior que a média que o número de mensagem por chat*. Para tal, utilizamos o método AVG para determinarmos a media de mensagens por chat e ao relacionarmos **Mensagem** com o **Perfil,** o COUNT serviu para vermos o número de mensagens que cada utilizador mandou, utilizando HAVING para fazer a restrição.

**Query10:** *sala com mais subscrições de perfis premium.* A partir de uma subquerie no From, em que relacionamos a **Subscricaosala** com uma subquerie, que nos retorna os id de perfis premium, obtendo, através do COUNT, o numero de subscrições premium de todas as salas. Utilizando o método MAX, obtém-se a sala pretendida.

***Gatilhos***

Foram criados três gatilhos com as seguintes características:

**Adicionar Perfil**

**Após** ser criado um **Perfil**, o gatilho automaticamente gera um perfil grátis associado a este perfil. Isto faz com que todos os perfis ao serem criados serão do tipo grátis e terão a hipótese de fazer ‘upgrade’ mais tarde.

**Verificação em pedidos de amizade**

O segundo gatilho verifica **antes** de inserir um pedido de **amizade** que os dois perfis em questão não são iguais. Isto é o pedido de amizade não é um pedido para o próprio perfil. Se esta for a situação um erro é gerado.

**Verificação de remoção de perfil grátis**

O último gatilho que **antes** de remover um **perfil grátis**, este **perfil** tem já um **perfil premium** criado. Isto é para remover as restrições do **perfil grátis**, o utilizador terá que ter feito ‘upgrade’. Caso o utilizador não tenha um **perfil premium** o pedido é então ignorado.

Nota: A verificação do gatilho 3 requer usar o “povoar.sql” primeiro.