

# Base de Dados L.EIC012

2021/2022

Trabalho Orientado por:

Prof. Carla Alexandra Teixeira Lopes

Grupo 106

Bernardo Campos

Huigui Guo

Tiago Marques

# Índice

Capa.....	1
Índice .....	2
Contexto .....	3
Diagramas UML.....	4
Esquema relacional .....	5
Análise de Dependências Funcionais e Formas Normais .....	6
Restrições .....	9
Interrogações .....	13
Gatilhos .....	14

# Contexto

Com este projeto pretende-se criar uma base de dados para uma aplicação que permita aos seus utilizadores a comunicação por chamadas de voz e vídeo, ou pela troca de mensagens, de maneira semelhante ao *Discord*.

A base de dados armazena várias informações relativas ao **Utilizador**: o seu nome, id, password, email, número de telefone e foto de perfil.

Utilizadores podem trocar pedidos de amizade entre si, que podem ou não ser aceites, sendo também possível bloquear o contacto com outros utilizadores. Existem diferentes **Status** que os utilizadores podem definir para si próprios, como *online*, *offline*, ocupado (não incomodar) e invisível.

Outra forma de interação é a partir de servidores, dos quais utilizadores podem fazer parte. Sobre um **Servidor** é guardado o seu nome e uma foto. Utilizadores têm diferentes papéis em cada servidor, que autorizam a várias ações dentro deste, como acesso a áreas restritas, permissão a convidar novos utilizadores e outras atividades administrativas.

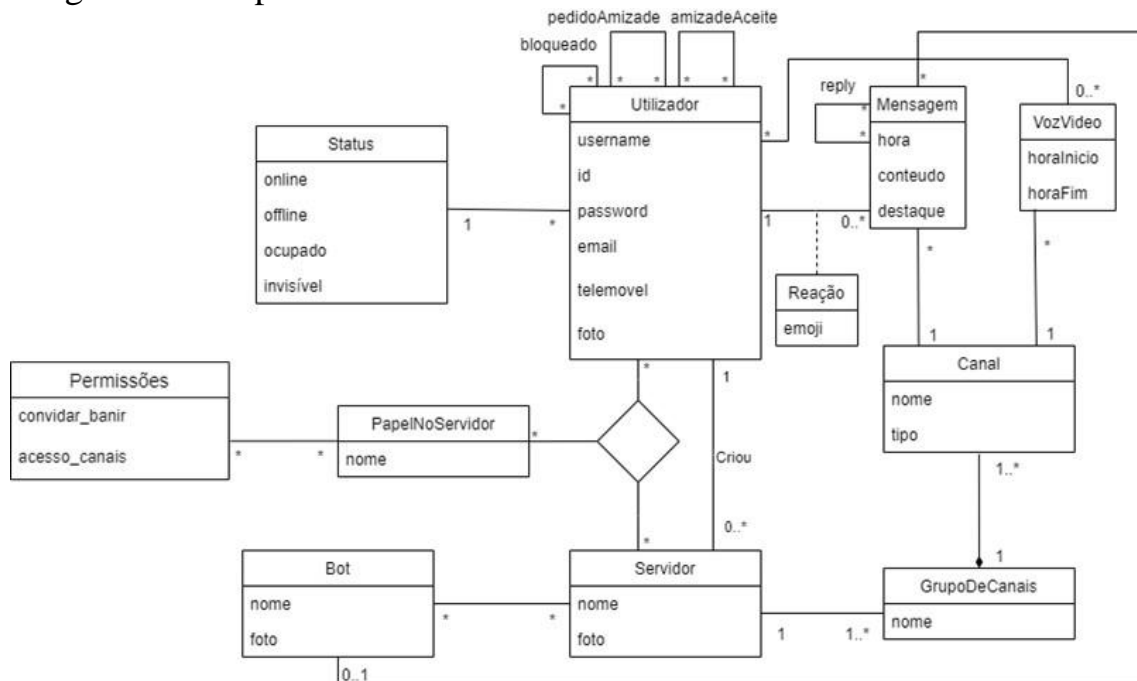
**Bots**, também com nome e foto de perfil, podem fazer parte de servidores de maneira idêntica à dos utilizadores. Têm como objetivo trazer novas funcionalidades ao servidor, como reprodução de música ou jogos simples, e facilitar a manutenção do servidor. *Bots* interpretam mensagens em canais de texto como comandos.

Servidores têm canais, com nome, que podem ser de texto ou de voz e vídeo. Em canais de texto utilizadores podem trocar mensagens entre si. É também possível “destacar” certas mensagens para que sejam visíveis durante mais tempo, dar *reply* a essas mensagens ou reagir com *emojis*. Nos canais de voz é registada a hora de início e de fim das chamadas, tal como a hora em que cada utilizador entrou e saiu da chamada.

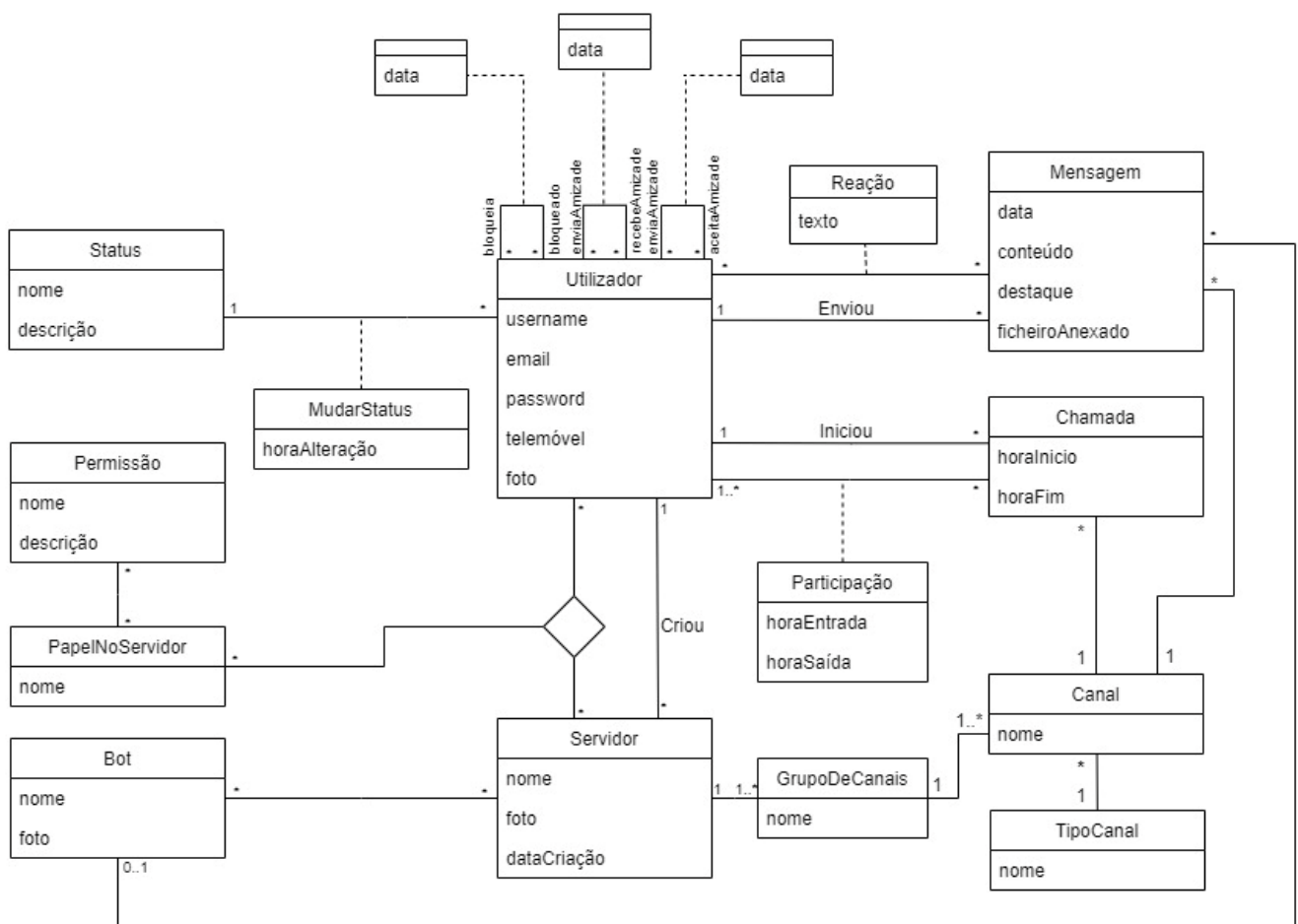
Estes canais, por motivos de melhor organização, fazem parte de grupos, normalmente associados à função que lhes é dada pelos membros do servidor.

# Diagramas UML

## Diagrama conceptual inicial



## Diagrama conceptual final



# Esquema relacional

**Status** (idStatus, nome, descrição)

**Utilizador** (idUser, username, email, password, telemóvel, foto)

**MudarStatus** (idStatus->Status, idUser->Utilizador, horaAlteração)

**Permissão** (idPermissão, nome, descrição)

**PapelNoServidor** (idPapelNoServidor, nome)

**Bot** (idBot, nome, foto)

**Servidor** (idServidor, nome, foto, dataCriação, idCriador->Utilizador)

**TipoCanal** (idTipoCanal, nome)

**GrupoDeCanais** (idGrupoDeCanais, nome, idServidor->Servidor)

**Canal** (idCanal, nome, idGrupoDeCanais->GrupoDeCanais, idTipoCanal->TipoCanal)

**Mensagem** (idMensagem, hora, conteúdo, destaque, ficheiroAnexado, idBot->Bot, idUser->Utilizador, idCanal->Canal)

**Chamada** (idChamada, horaInicio, horaFim, idUserQueIniciou->Utilizador, idCanal->Canal)

**Participação** (idChamada->Chamada, idUser->Utilizador, horaEntrada, horaSaída)

**Reação** (idReação, idMensagem->Mensagem, texto)

**Bloqueado** (idUserBloqueia->Utilizador, idUserBloqueado->Utilizador, data)

**PedidoAmizade** (idUserEnvia->Utilizador, idUserRecebe->Utilizador, data)

**AmizadeAceite** (idUserEnvia->Utilizador, idUserAceita->Utilizador, data)

**BotServidor** (idBot->Bot, idServidor->Servidor)

**PermissãoPapel** (idPermissão->Permissão, idPapelNoServidor->PapelNoServidor)

**UtilizadorServidorPapel** (idUser->Utilizador, idServidor->Servidor, idPapelNoServidor->PapelNoServidor)

# Análise de Dependências Funcionais e Formas Normais

**Status** (idStatus, nome, descrição)

*FDs*: idStatus->nome, descrição

nomes, descrição -> idStatus

Formas: BCNF e 3NF

**Utilizador** (idUser, username, email, password, telemóvel, foto)

*FDs*: idUser->username, email, password, telemóvel, foto

Formas: BCNF e 3NF

**MudarStatus** (idStatus->Status, idUser->Utilizador, horaAlteração)

*FDs*: idStatus, idUser-> horaAlteração

Formas: BCNF e 3NF

**Permissão** (idPermissão, nome, descrição)

*FDs*: idPermissão->nome, descrição

nome, descrição -> idPermissão

Formas: BCNF e 3NF

**PapelNoServidor** (idPapelNoServidor, nome)

*FDs*: idPapelNoServidor->nome

nome->idPapelNoServidor

Formas: BCNF e 3NF

**Bot** (idBot, nome, foto)

*FDs*: idBot->nome, foto

Formas: BCNF e 3NF

**Servidor** (idServidor, nome, foto, dataCriação, idCriador->Utilizador)

*FDs:* idServidor-> nome, foto, dataCriação, idCriador

nome, foto, dataCriação, idCriador->idServidor

Formas: BCNF e 3NF

**TipoCanal** (idTipoCanal, nome)

*FDs:* idTipoCanal-> nome

nome->idTipoCanal

Formas: BCNF e 3NF

**GrupoDeCanais** (idGrupoDeCanais, nome, idServidor->Servidor)

*FDs:* idGrupoDeCanais-> nome, idServidor

Formas: BCNF e 3NF

**Canal** (idCanal, nome, idGrupoDeCanais->GrupoDeCanais,  
idTipoCanal->TipoCanal)

*FDs:* idCanal-> nome, idGrupoDeCanais, idTipoCanal

Formas: BCNF e 3NF

**Mensagem** (idMensagem, hora, conteúdo, destaque, ficheiroAnexado,  
idBot->Bot, idUser->Utilizador, idCanal->Canal)

*FDs:* idMensagem-> hora, conteúdo, destaque, ficheiroAnexado,  
idBot, idUser, idCanal

hora, conteúdo, destaque, ficheiroAnexado, idBot, idUser,  
idCanal -> idMensagem

Formas: BCNF e 3NF

**Chamada** (idChamada, horaInicio, horaFim,  
idUserQueIniciou->Utilizador, idCanal->Canal)

*FDs:* idChamada-> horaInicio, horaFim, idUserQueIniciou, idCanal

horaInicio, horaFim, idUserQueIniciou, idCanal -> idChamada

Formas: BCNF e 3NF

**Participação** (idChamada->Chamada, idUser->Utilizador, horaEntrada, horaSaída)

*FDs:* idChamada, idUser -> horaEntrada, horaSaída

Formas: BCNF e 3NF

**Reação** (idReação, idMensagem->Mensagem, texto)

*FDs:* idReação-> idMensagem, texto

Formas: BCNF e 3NF

**Bloqueado** (idUserBloqueia->Utilizador, idUserBloqueado->Utilizador, data)

*FDs:* idUserBloqueia, idUserBloqueado -> data

Formas: BCNF e 3NF

**PedidoAmizade** (idUserEnvia->Utilizador, idUserRecebe->Utilizador, data)

*FDs:* idUserEnvia, idUserRecebe-> data

Formas: BCNF e 3NF

**AmizadeAceite** (idUserEnvia->Utilizador, idUserAceita->Utilizador, data)

*FDs:* idUserEnvia, idUserAceita -> data

Formas: BCNF e 3NF

**BotServidor** (idBot->Bot, idServidor->Servidor)

Formas: BCNF e 3NF

**PermissãoPapel** (idPermissão->Permissão, idPapelNoServidor->PapelNoServidor)

Formas: BCNF e 3NF

**UtilizadorServidorPapel** (idUser->Utilizador, idServidor->Servidor, idPapelNoServidor->PapelNoServidor)

Formas: BCNF e 3NF



# Restrições

## ***Status***

Cada utilizador só pode ter um status – idStatus PRIMARY KEY

Todos os *status* devem ter nome e descrição – nome NOT NULL, descrição NOT NULL

## **Utilizador**

Não pode haver dois utilizadores com o mesmo ID – idUSER PRIMARY KEY

Todos os utilizadores devem ter nome, email, password, telemóvel e foto

- nome NOT NULL, email NOT NULL, password NOT NULL, telemóvel NOT NULL, foto NOT NULL

## **MudarStatus**

O ID de *Status* e de Utilizador devem corresponder a um ID das tabelas *Status* e Utilizador – idStatus REFERENCES Status(idStatus), idUser REFERENCES Utilizador(idUser)

Todos os mudar status devem ter hora de alteração – horaAlteração NOT NULL

## **Permissão**

Não deve haver duas permissões com o mesmo ID – idPermissão PRIMARY KEY

Todas as permissões devem ter nome e descrição. O nome de uma permissão deve ser único.

- UNIQUE nome NOT NULL, descrição NOT NULL

### **PapelNoServidor**

Não deve haver dois papéis no servidor com o mesmo ID – idPapelNoServidor NOT NULL

Todos os papéis no servidor devem ter um nome único. – UNIQUE nome NOT NULL

### **Bot**

Não deve haver dois bots com o mesmo ID -idBot PRIMARY KEY

Todos os bots devem ter nome e foto -nome NOT NULL, foto NOT NULL

### **Servidor**

Não deve haver dois servidores com o mesmo ID -idServidor PRIMARY KEY

Todos os servidores devem ter nome, foto e data de criação – nome NOT NULL, foto NOT NULL, dataCriação NOT NULL

O idCriador deve corresponder ao um ID da tabela Utilizador – idCriador REFERENCES Utilizador(idUser)

### **TipoCanal**

Não deve haver dois tipos de canais com o mesmo ID -idTipoCanal PRIMARY KEY

Os tipos de canais devem ter um nome único – UNIQUE nome NOT NULL

### **GrupoDeCanais**

Não deve haver dois grupos de canais com o mesmo ID – idGrupoDeCanais PRIMARY KEY

Todos os grupos de canais devem ter nome – nome NOT NULL

O idServidor deve corresponder a um ID da tabela Servidor – idServidor REFERENCES Servidor(idServidor)

## **Canal**

Não deve haver dois canais com o mesmo ID – idCanal PRIMARY KEY

Todos os canais devem ter um nome – nome NOT NULL

O idGrupoDeCanais deve corresponder a um ID da tabela GrupoDeCanais, e idTipoCanal deve corresponder a um ID da tabela TipoCanal -  
idGrupoDeCanais REFERENCES GrupoDeCanais(idGrupoDeCanais),  
idTipoCanal REFERENCES TipoCanal(idTipoCanal)

## **Mensagem**

Não deve haver duas mensagens com o mesmo ID – idMensagem PRIMARY KEY

Todas as mensagens devem ter hora e conteúdo – hora NOT NULL, conteúdo NOT NULL

O idBot, idUser, idCanal devem corresponder a um ID das tabelas Bot, Utilizador e Canal respetivamente – idBot REFERENCES Bot(idBot), idUser REFERENCES Utilizador(idUser), idCanal REFERENCES Canal(idCanal)

## **Chamada**

Não deve haver duas chamadas com o mesmo ID – idChamada PRIMARY KEY

O idUserQueIniciou e idCanal devem corresponder a um ID das tabelas Utilizador e Canal respetivamente – idUserQueIniciou REFERENCES Utilizador(idUser), idCanal REFERENCES Canal(idCanal)

Hora de fim tem que ser posterior à hora de início – CHECK (horaFim>horaInicio)

## **Participação**

O idChamada e idUser devem corresponder a um ID das tabelas Chamada e Utilizador respetivamente – idChamada REFERENCES

Chamada(idChamada), idUser REFERENCES Utilizador(idUser)

Hora de saída tem que ser posterior à hora de entrada –  
CHECK(horaSaída>horaEntrada)

## **Reação**

Não deve haver duas reações com o mesmo id – idReação PRIMARY KEY

Todas as reações devem ter texto – texto NOT NULL

O idMensagem deve corresponder a um ID da tabela Mensagem –  
idMensagem REFERENCES Mensagem(idMensagem)

## **Bloqueado**

O idUserBloqueia e idUserBloqueado devem corresponder a um ID da tabela Utilizador – idUserBloqueia REFERENCES Utilizador(idUser),  
idUserBloqueado REFERENCES Utilizador(idUser)

Deve sempre existir uma data – data NOT NULL

## **PedidoAmizade**

O idUserEnvia e idUserRecebe devem corresponder a um ID da tabela Utilizador – idUserEnvia REFERENCES Utilizador(idUser), idUserRecebe  
REFERENCES Utilizador(idUser)

Deve sempre existir uma data – data NOT NULL

## **AmizadeAceite**

O idUserEnvia e idUserAceita devem corresponder a um ID da tabela Utilizador – idUserEnvia REFERENCES Utilizador(idUser), idUserAceita  
REFERENCES Utilizador(idUser)

Deve sempre existir uma data – data NOT NULL

## BotServidor

O idBot e idServidor devem corresponder a um ID das tabelas Bot e Servidor respetivamente – idBot REFERENCES Bot(idBot), idServidor REFERENCES Servidor(idServidor)

## PermissãoPapel

O idPermissão e idPapelNoServidor devem corresponder a um ID das tabelas Permissão e PapelNoServidor respetivamente – idPermissão REFERENCES Permissão(idPermissão), idPapelNoServidor REFERENCES PapelNoServidor(idPapelNoServidor)

## UtilizadorServidorPapel

O idUser, idServidor, idPapelNoServidor devem corresponder a um ID das tabelas Utilizador, Servidor e PapelNoServidor respetivamente – idUser REFERENCES Utilizador(idUser), idServidor REFERENCES Servidor(idServidor), idPapelNoServidor REFERENCES PapelNoServidor(idPapelNoServidor)

## Interrogações

De seguida, apresentam-se as *queries* que achámos pertinentes para o contexto da nossa base de dados e do trabalho.

1. Número total de utilizadores na plataforma
2. Servidores que contém a palavra “*gaming*” no nome
3. Pedidos de amizade que foram bloqueados
4. Amigos em segundo grau (amigos de amigos) de cada um dos utilizadores
5. Utilizadores que têm acesso a canais privados
6. Mensagem com mais reações
7. O par de utilizadores que reagiram a mais mensagens entre si
8. Servidor com mais membros
9. Utilizador com mais amigos\*
10. Número total de interações (mensagens, participações em chamadas e reações a mensagens) de cada utilizador\*

\*estas interrogações, embora mostrem o resultado correto para os dados atualmente inseridos, não estão totalmente corretas.

## Gatilhos

Foram adicionados alguns *triggers* que são uteis para a monitorização e manutenção da base de dados.

1. Verifica se ao se enviar uma mensagem a um certo canal, o utilizador que a enviou tem acesso a esse canal
2. Após uma amizade ser aceite, o pedido de amizade (ainda pendente) é eliminado da base de dados
3. Quando um utilizador que participa de uma chamada se junta a outra, vai ser desconectado da primeira.

## Participação dos elementos do grupo

Na entrega final o trabalho foi igualmente distribuído entre Bernardo Campos e Tiago Marques, sem a colaboração de Huigui Guo.

No entanto, nas primeiras duas entregas o trabalho foi igualmente distribuído entre os três membros do grupo.