2023

Modelação de base de dados

Exemplo de Sistema de LOGIN

Tiago Silva Marinho - 8998



Índice

[Índice das Figuras 2](#_Toc138753100)

[Resumo 3](#_Toc138753101)

[Introdução 4](#_Toc138753102)

[Desenvolvimento 5](#_Toc138753103)

[Modelo Conceptual da Base de Dados. 5](#_Toc138753104)

[Identificação das Entidades e respetivos atributos 5](#_Toc138753105)

[Relacionamentos entre Entidades 6](#_Toc138753106)

[Diagrama E-R da base de dados 7](#_Toc138753107)

[Modelo Lógico da Base de Dados 8](#_Toc138753108)

[Identificação das Entidades e respetivos atributos 8](#_Toc138753109)

[Relacionamentos entre Entidades 9](#_Toc138753110)

[Modelo Relacional da base de dados 9](#_Toc138753111)

[Modelo Físico da Base de Dados 10](#_Toc138753112)

[Apreciação crítica 12](#_Toc138753113)

[WEBGRAFIA 13](#_Toc138753114)

[Anexos 14](#_Toc138753115)

# Índice das Figuras

[Figura 1- entidade USER 5](#_Toc138753116)

[Figura 2- entidade FUNCAO 5](#_Toc138753117)

[Figura 3- entidade LIVRO 6](#_Toc138753118)

[Figura 4- relação USER\_FUNCAO 6](#_Toc138753119)

[Figura 5- Diagrama-ER 7](#_Toc138753120)

[Figura 6- MR entidade LIVRO 8](#_Toc138753121)

[Figura 7- MR entidade USER 8](#_Toc138753122)

[Figura 8- MR entidade FUNCAO 8](#_Toc138753123)

[Figura 9- MR relação USER\_FUNCAO 9](#_Toc138753124)

[Figura 10- Modelo Relacional 9](#_Toc138753125)

[Figura 11- anexo 1 14](#_Toc138753126)

[Figura 12- anexo 2 15](#_Toc138753127)

# Resumo

O presente trabalho consiste na modelação da base de dados de um sistema com controle de acesso baseado em funções (Sistema Login).

Este Sistema permite proteger os dados confidenciais do sistema e garantir que apenas os utilizadores autorizados (admin) têm acesso aos dados, informações e processos requeridos ao desempenho das suas respetivas funções.

Para este modo eu criei 1 diagrama E-R, 1 modelo relacional e 1 modelo físico da base de dados (em sql).

No total para o sistema de login eu criei 3 entidades: user, session e games

Sendo as tabelas user e funcao usadas para o login

**Palavras-chave:**

Controle de Acesso, RBAC, utilizador, função, permissão, SGDB, MySQL, SQL. Livrario, query, login, registo, admin e user

# Introdução

Este sistema de login permite-nos fornecer aos utilizadores diferentes níveis de acesso, tendo por base as funções desempenhadas por eles.

Para a suade e bom funcionamento deste sistema, deve-se seguir 3 regras primárias:

* Regra 1 - Atribuição de função

Um utilizador pode exercer a permissão, se tiver atribuída uma função.

* Regra 2 - Autorização de função

A função ativa de um utilizador deve ser autorizada para o utilizador. Com a Regra 1, esta regra garante que os utilizadores possam aceder à sua sessão e junto com a sessão, as suas funções para as quais estão autorizados.

* Regra 3 - Autorização de permissão

Um utilizador pode exercer uma permissão somente se a permissão for autorizada para a função ativa do utilizador. Com as Regras 1 e 2, essa regra garante que os utilizadores possam exercer apenas as permissões para as quais estão autorizados.

Para o melhor entendimento faremos um exemplo, imaginemos um sistema simples, onde cada utilizador pode ter uma das seguintes funções: administrador, técnico ou programador, conforme a tabela.

|  |  |  |  |
| --- | --- | --- | --- |
| **PERMISSÃO** |  | **Sessão/Session** |  |
| **administrador** | **programador** | **técnico** |
| **editar** | *sim* | *sim* | *não* |
| **apagar** | *sim* | *não* | *não* |
| **ler** | *sim* | *sim* | *sim* |

Como se pode ver, todos os utilizadores têm permissão de leitura, mas apenas os utilizadores com perfil administrador têm permissão para apagar.

Durante a realização deste trabalho realizei 3 modelos, para o melhor entendimento e a melhor documentação do trabalho. Estes modelos serão falados a maior detalhe no “DESENVOLVIMENTO”.

# Desenvolvimento

## Modelo Conceptual da Base de Dados.

O modelo conceptual de uma base de dados é construído fundamentalmente por três tipos de elementos: entidades, atributos e os relacionamentos entre as entidades. O diagrama E-R (Entidade-Relacionamento) é utilizado como representação visual do modelo conceptual da base de dados.

Pretendemos implementar um sistema de controle de acesso que restringe o acesso ao sistema, apenas a utilizadores autorizados, que podem ter diferentes tipos de permissões no sistema. Considerando que, sessão indica o cargo que define um nível de autoridade dentro do sistema e permissões no modo de acesso aos recursos do sistema. Os utilizadores do sistema não recebem diretamente as permissões, apenas as adquirem por meio da função (ou funções) desempenhada(s).

### Identificação das Entidades e respetivos atributos

Foram identificadas as entidades UTILIZADOR, FUNÇÃO e LIVRO.

* Entidade USER

Atributos: username, email, id, password e create\_datetime

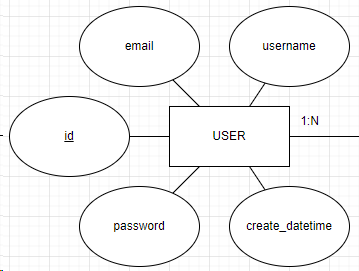


Figura 1- entidade USER

* Entidade Session (Sessão)
* Atributos: nome, id, descricao

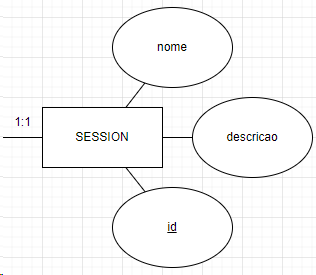


Figura 2- entidade SESSION

* Entidade GAMES

Atributos: id, tipo, empresa, nome e anocreacao

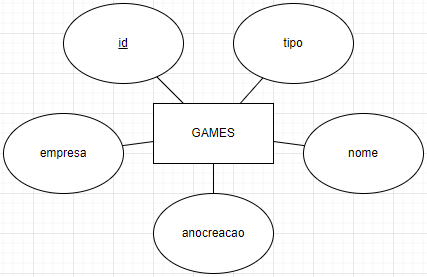


Figura 3- entidade GAMES

### Relacionamentos entre Entidades

Foram identificadas as seguintes relações:

* Entre as Entidades USER e SESSION

A relação é do tipo um para muitos (1:N) com participação obrigatória de ambas as entidades:

o Um **utilizador** tem uma ou e somente uma sessão.

o Uma **sessão** é desempenhada por um ou vários utilizadores.

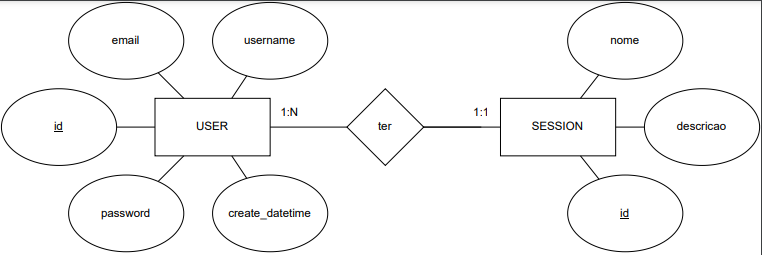


Figura 4- relação USER\_SESSION

* Não existe relação entre GAMES e as outras tabelas

### Diagrama E-R da base de dados

O modelo conceptual da base de dados RBAC é representado no seguinte diagrama E-R

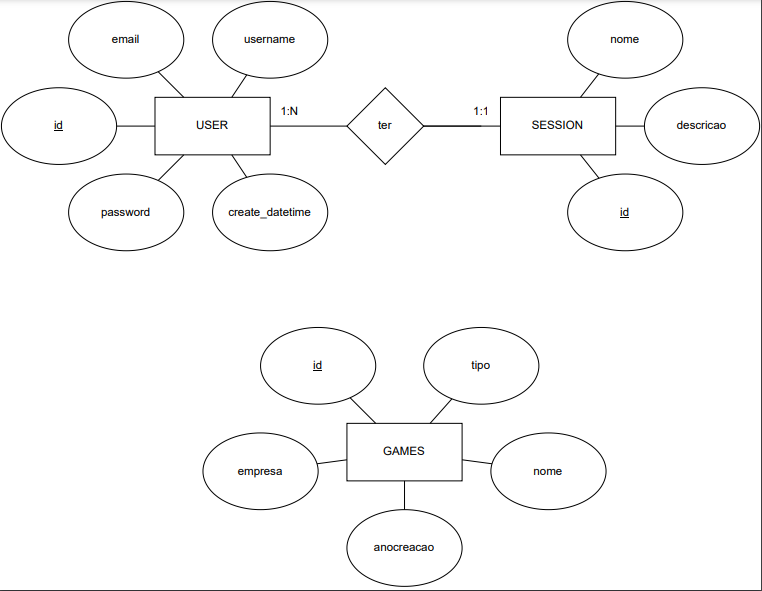


Figura 5- Diagrama-ER

## Modelo Lógico da Base de Dados

### Identificação das Entidades e respetivos atributos

O modelo lógico, descreve como os dados e os relacionamentos serão armazenados na base de dados e define como será implementado no SGBD.

O SGDBR escolhido é MySQL. O modelo relacional permite representar as entidades respetivas relações como tabelas. Para isso, basta converter as relações, identificadas anteriormente, no modelo relacional.

A relação entre as entidades UTILIZADOR e SESSION é uma relação do tipo um para muitos (1:N) com participação obrigatória de ambas as entidades:



Figura 6- MR entidade GAMES

* A tabela GAMES não está relacionada a outra base de dados

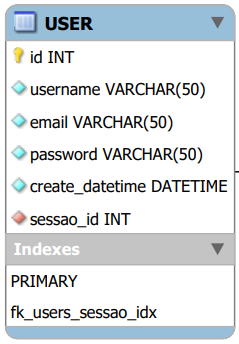


Figura 7- MR entidade USER

* Um **utilizador** só pode ter 1 sessão: USER ou ADMIN.

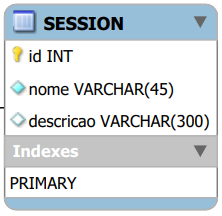


Figura 8- MR entidade SESSION

* Uma **sessão (user ou admin)** pode ser desempenhada por um ou vários utilizadores (se houver mais do que 1 admin ou mais do que 1 utilizador).

### Relacionamentos entre Entidades

Relação USER\_SESSION:

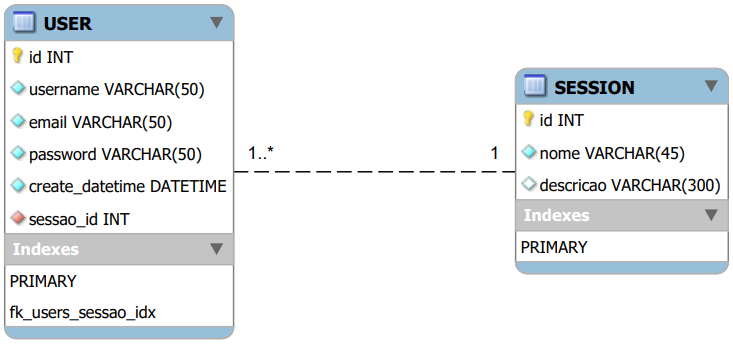


Figura 9- MR relação USER\_SESSION

* Relação muitos para 1 (1 para muitos) entre USER e SESSION
* Não existe uma relação entre GAMES com outras TABELAS

### Modelo Relacional da base de dados

O modelo relacional da base de dados RBAC é representado no seguinte diagrama

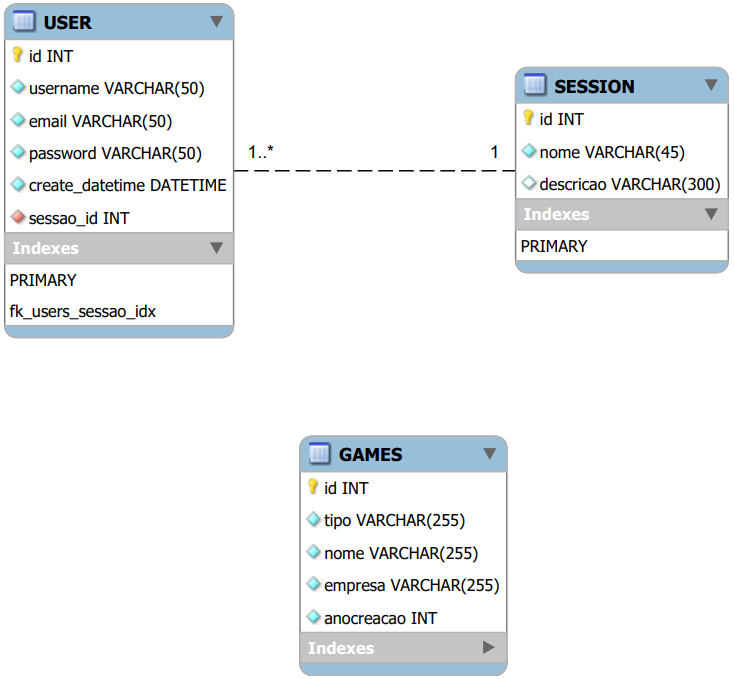


Figura 10- Modelo Relacional

## Modelo Físico da Base de Dados

No Modelo físico são detalhados os componentes da estrutura física da base de dados, como tabelas, campos e chaves. O script do modelo físico é composto por uma sequência de comandos SQL, que quando executados criam a base de dados, tabelas e registos.

De seguida, é apresentado o script de criação da base de dados.

CRIAR E USAR BASE DE DADOS:

CREATE DATABASE loginsystem;

Use loginsystem;

CRIAR TABLE GAMES:

CREATE TABLE IF NOT EXISTS `loginsystem`.`games` (

`id` INT NOT NULL AUTO\_INCREMENT,

`tipo` VARCHAR(255) NOT NULL,

`nome` VARCHAR(255) NOT NULL,

`empresa` VARCHAR(255) NOT NULL,

`anocreacao` VARCHAR(4) NULL DEFAULT NULL,

PRIMARY KEY (`id`));

CRIAR TABLE FUNCAO:

CREATE TABLE IF NOT EXISTS `loginsystem`.`session` (

`id` INT NOT NULL AUTO\_INCREMENT,

`nome` VARCHAR(45) NOT NULL,

`descricao` VARCHAR(300) NULL,

PRIMARY KEY (`id`));

CRIAR TABLE USERS:

CREATE TABLE IF NOT EXISTS `loginsystem`.`users` (

`id` INT NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(50) NOT NULL,

`email` VARCHAR(50) NOT NULL,

`password` VARCHAR(50) NOT NULL,

`create\_datetime` DATETIME NOT NULL,

`session\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_users\_session\_idx` (`session\_id` ASC),

CONSTRAINT `fk\_users\_session`

FOREIGN KEY (`session\_id`)

REFERENCES `loginsystem`.`session` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

AUTO\_INCREMENT = 3;

INSERIR EM FUNCAO, AS FUNÇOES ADMIN E USER

INSERT INTO `session` (`id`, `nome`, `descricao`) VALUES

(1, 'Admin', 'System admin'),

(2, 'User', 'System user');

INSERIR USER ROOT (ADMIN) NA BASE DE DADOS

INSERT INTO `users` (`id`, `username`, `email`, `password`, `create\_datetime`,`session\_id`) VALUES (1, 'root', 'root', '63a9f0ea7bb98050796b649e85481845', '2023-06-23 16:53:00',1);

(1, 'root', 'root', '63a9f0ea7bb98050796b649e85481845', '2023-06-23 16:53:00',1);

# Apreciação crítica

Esta proposta de trabalho é uma boa oportunidade para desenvolver habilidades em programação e gestão de projetos. O objetivo principal é implementar um sistema de logins em PHP ou Java, com a realização de planeamento, modelação da base de dados e implementação de pelo menos duas funcionalidades. Além disso, é importante garantir que cada usuário seja encaminhado para a página correta e que o projeto seja devidamente documentado.

A modelação da base de dados é um passo crítico, pois influencia a estruturação do sistema como um todo. A exigência de apresentar o modelo conceptual, lógico e físico da base de dados é uma boa prática, pois permite a compreensão de como os dados são organizados e como as informações são armazenadas.

A implementação das funcionalidades de registro de novo usuário e login é fundamental para a realização do sistema de logins. É importante garantir a segurança das informações dos usuários e a privacidade de seus dados pessoais, como senhas e informações de contato.

O encaminhamento de cada usuário para a página correta também é uma etapa importante, pois garante a experiência do usuário e a eficiência do sistema. É importante que o sistema seja capaz de identificar automaticamente o tipo de usuário e encaminhá-lo para a página correta.

Por fim, a exigência de documentação do projeto, incluindo o código-fonte, a descrição do projeto, o relatório do projeto e o planeamento, é uma boa prática de desenvolvimento de software. Isso permite que outras pessoas possam entender e contribuir para o projeto, além de ajudar a equipe a manter-se organizada e focada nos objetivos.

# WEBGRAFIA

GITHUB:

Repositório de TiagoMarinho-8998 (Tiago Silva Marinho);

<https://github.com/TiagoMarinho-8998/Project-MiniPAP.git>

# Anexos

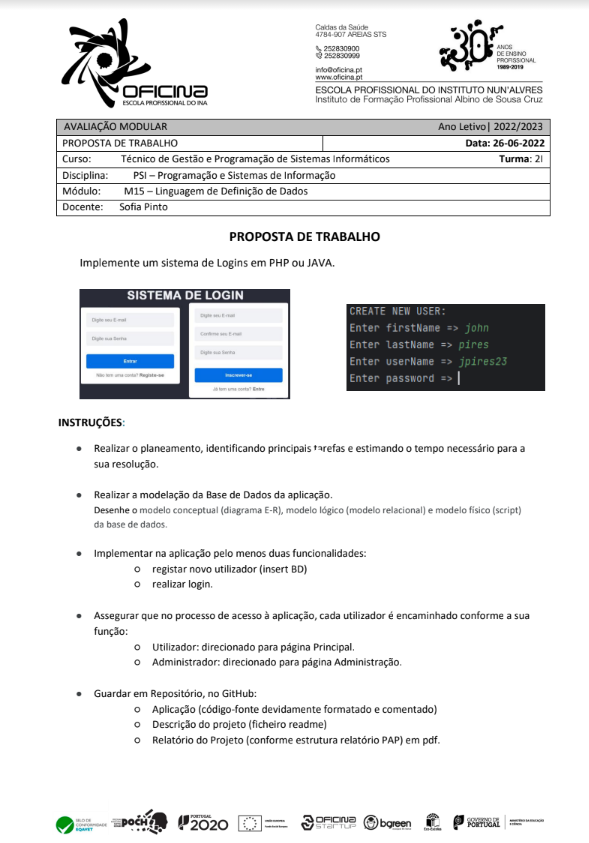


Figura 11- anexo 1

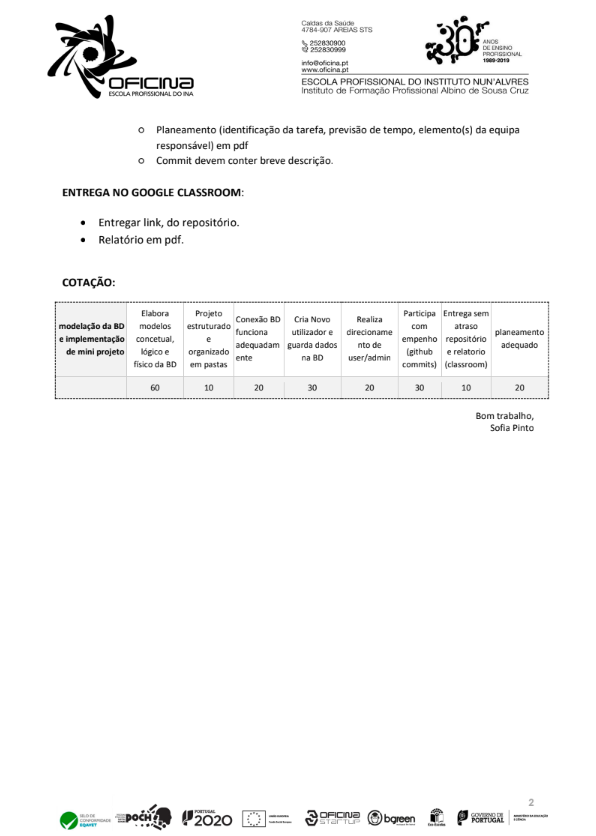


Figura 12- anexo 2