

Faculdade de Informática e Administração Paulista

**Data Application & Data Science**

**ENTREGÁVEL DA DISCIPLINA**

**INTEGRANTES**

|  |  |
| --- | --- |
| **RM**  **(SOMENTE NÚMEROS)** | **NOME COMPLEMENTO**  **(SEM ABREVIAR)** |
| **89171** | **Bianca Man Tchuin Chang Lee** |
| **89045** | **Danilo Zequim de Moura** |
| **87087** | **Eric Brianez Giannetti** |
| **86931** | **Matheus Pismel de Jeronymo** |
| **87680** | **Otavio de Magalhães Gomes** |
| **87149** | **Zack Lorenzzo Corrêa** |

Sumário

[1 – Descrição do Projeto e Regras de Negócio 4](#_Toc101105912)

[2 – Dicionário de Dados 5](#_Toc101105913)

[3 – Projeto Físico do Banco de Dados 8](#_Toc101105914)

[4 – DDL da base de dados 9](#_Toc101105915)

# 1 – Descrição do Projeto e Regras de Negócio

O projeto consiste na criação de uma plataforma de jogos presenciais, como pega-pega, esconde-esconde, queimada e até mesmo jogos para adultos como sueca e eu nunca.

O usuario poderia entrar dentro do nosso App e informar em quantas pessoas está e se possui algum item que possa ser usado no jogo, como uma bola ou uma baralho de cartas. Em seguida, o app criaria uma lista de sugestões para o usuario escolher que condizem com as informações fornecidas. Logo em seguida o app iria explicar o jogo aos participantes e tirar possíveis dúvidas.

Dessa forma temos as seguintes regras de negocio:

* As principais tabelas do projeto deverão ter um campo com data de alteração, para termos um controle histórico das análises que se sucederão;
* O usuario ao se cadastrar deverá forncer obrigatoriamente cpf, e-mail e senha;
  + CPF e E-mail deverão ser informações únicas na tabela;
* O usuario poderá ter vários endereços cadastrados, contudo apenas o item mais recente cadastrado será marcado como um endereço validoç
* Todo jogo deverá obrigatoriamente ter o nome e o número mínimo de participantes necessários informado.
* Os jogos poderão ou não ser jogos validos. Um jogo valido é uma das brincadeiras que recebeu apoio suficiente da comunidade para entrar dentro do app;
  + Para esse controle os usuários poderão fazer comentários nos jogos informando se jogaram ou não o jogo e o que acharam, bem como deixar um comentário em texto;
* Todo jogo deverá ter atrelado a ele sua lista de regras que poderão ser regras opcionais ou não ao jogo, permitindo maior personalização dos jogos pelo usuário;
* Por fim os jogos poderão ter itens para serem utilizados, todo item será composto por seu nome e uma macrocategoria, por exemplo uma bola de futebol terá como macrocategoria bola;
  + Um jogo poderá ter mais de um item, bem como esse item pode ser usado em mais de um jogo.
* pARA ESSE PROJETO A BASE SERÁ CONSTRUINDA EM POSTGRESQL

## 2 – Dicionário de Dados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | TB\_Usuario | | | |
| **Descrição** | Tabela para o cadastro dos usuários | | | |
| **Coluna** | **Tipo de Dados** | **Tamanho** | **Constraint** | **Descrição** |
| Id\_usuario | INTEGER |  | PK | Identificador único do usuário |
| Nr\_cpf\_cnpj | VARCHAR | 11 | NN | Número do CPF ou CNPJ |
| Nm\_usuario | VARCHAR | 50 |  | Nome do usuário |
| Ds\_email | VARCHAR | 50 | NN | E-Mail do usuário |
| Ds\_senha | VARCHAR | 50 | NN | Senha do usuário |
| Dt\_nascimento | DATE |  |  | Data de nascimento do usuário, caso seja um PJ data de fundação |
| Dt\_alteracao | DATE |  |  | Data do input ou alteração do dado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | TB\_Endereco | | | |
| **Descrição** | Tabela para o cadastro do endereço usuários | | | |
| **Coluna** | **Tipo de Dados** | **Tamanho** | **Constraint** | **Descrição** |
| Id\_endereco | INTEGER |  | PK | Identificador único do endereço |
| Id\_usuario | INTEGER |  | FK | Identificador único do usuário |
| Nm\_pais | VARCHAR | 15 |  | Nome do país que o endereço se encontra |
| Cd\_uf | VARCHAR | 2 |  | Código da unidade da federação que o endereço se encontra |
| Nm\_cidade | VARCHAR | 50 |  | Nome da cidade que o endereço se encontra |
| Nr\_cep | INTEGER |  |  | Número do CEP |
| Nm\_rua | VARCHAR | 50 |  | Nome da rua que o endereço se encontra |
| Nr\_endereco | INTEGER |  |  | Número do estabelecimento |
| Fl\_valido | ENUMERATOR |  |  | Flag se é um endereço valido ou não (aceita apenas valores de sim ou não) |
| Dt\_alteracao | DATE |  |  | Data do input ou alteração do dado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | TB\_Avaliacao | | | |
| **Descrição** | Tabela para o cadastro das avaliações dos usuários sobre um jogo | | | |
| **Coluna** | **Tipo de Dados** | **Tamanho** | **Constraint** | **Descrição** |
| Id\_avaliacao | INTEGER |  | PK | Identificador único do endereço |
| Id\_usuario | INTEGER |  | FK | Identificador único do usuário |
| Id\_jogo | INTEGER |  | FK | Identificador único do jogo |
| Ds\_avaliacao | TEXT |  |  | Texto com o comentário do usuário sobre o jogo |
| tp\_avlaiacao | ENUMERATOR |  |  | Campo para descrever se foi uma experiência positiva ou negativa (aceitando apenas esses dois valores) |
| Fl\_jogado | ENUMERATOR |  |  | Flag para informar se o usuário jogou ou não (aceita apenas valores de sim ou não) |
| Dt\_alteracao | DATE |  |  | Data do input ou alteração do dado |

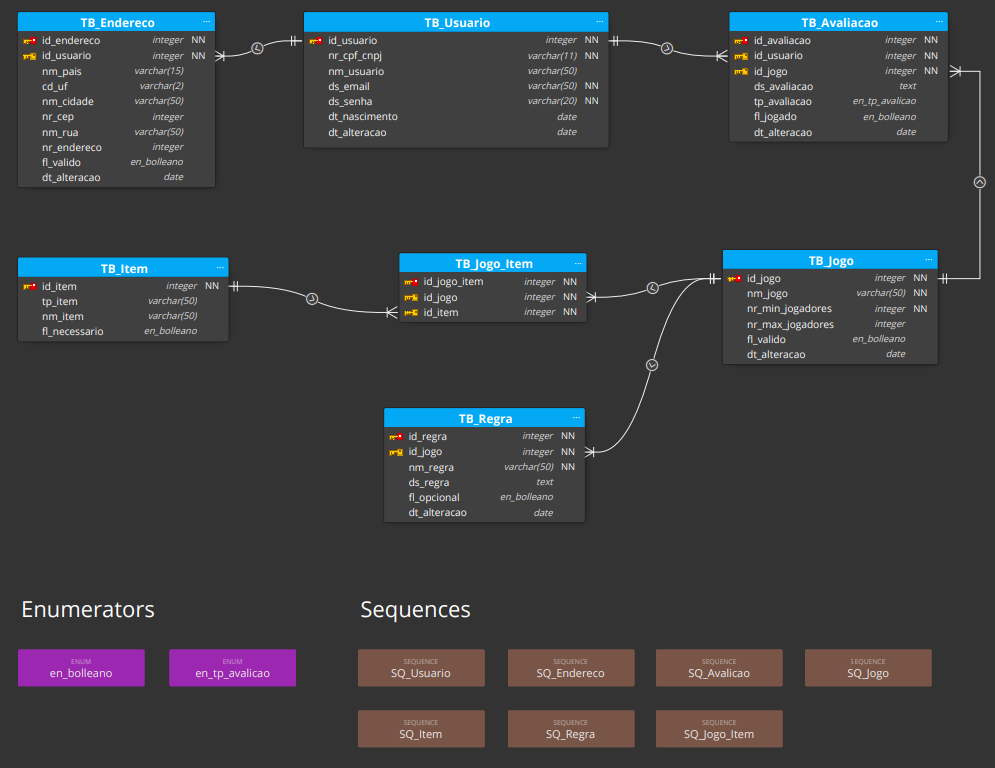
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | TB\_Jogo | | | |
| **Descrição** | Tabela para o cadastro dos jogos da plataforma | | | |
| **Coluna** | **Tipo de Dados** | **Tamanho** | **Constraint** | **Descrição** |
| Id\_jogo | INTEGER |  | FK | Identificador único do jogo |
| Nm\_jogo | VARCHAR | 50 | NN | Nome do jogo |
| Nr\_min\_jogadores | INTEGER |  | NN | Número mínimo de jogadores para poder jogar |
| Nr\_max\_jogadores | INTEGER |  |  | Número máximo de jogadores para poder jogar, se nulo será entendido que não há limite |
| Fl\_valido | ENUMERATOR |  |  | Flag para indicar se o jogo é valido ou não. Jogos validos são jogos que já passaram pelo crivo da comunidade e estarão disponibilizados no app |
| Dt\_alteracao | DATE |  |  | Data do input ou alteração do dado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | TB\_Regra | | | |
| **Descrição** | Tabela para o cadastro das regras de um jogo | | | |
| **Coluna** | **Tipo de Dados** | **Tamanho** | **Constraint** | **Descrição** |
| Id\_regra | INTEGER |  | PK | Identificador único do endereço |
| Id\_jogo | INTEGER |  | FK | Identificador único do jogo |
| Nm\_regra | VARCHAR | 50 |  | Nome da regra |
| Ds\_regra | TEXT |  |  | Descrição da regra do jogo |
| Fl\_jogado | ENUMERATOR |  |  | Flag para informar se é uma regra opcional do jogo ou se é uma regra obrigatória (aceita apenas valores de sim ou não) |
| Dt\_alteracao | DATE |  |  | Data do input ou alteração do dado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | TB\_Item | | | |
| **Descrição** | Tabela para o cadastro dos itens de um jogo | | | |
| **Coluna** | **Tipo de Dados** | **Tamanho** | **Constraint** | **Descrição** |
| Id\_item | INTEGER |  | PK | Identificador único do item a ser utilizado em um jogo |
| tp\_item | VARCHAR | 50 |  | Macro categoria do item, ou seja, uma grande categoria de classificação dos itens |
| Nm\_item | VARCHAR | 50 |  | Nome do item |
| Fl\_necessario | ENUMERATOR |  |  | Flag para informar se o item é obrigatório no jogo (aceita apenas valores de sim ou não) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | TB\_Jogo\_Item | | | |
| **Descrição** | Tabela para o relacionamento N:N da TB\_Jogo e TB\_Item | | | |
| **Coluna** | **Tipo de Dados** | **Tamanho** | **Constraint** | **Descrição** |
| Id\_jogo\_item | INTEGER |  | PK | Identificador único do relacionamento jogo e item |
| Id\_jogo | INTEGER |  | FK | Identificador único do jogo |
| Id\_item | INTEGER |  | FK | Identificador único do item a ser utilizado em um jogo |

## 3 – Projeto Físico do Banco de Dados



## 4 – DDL da base de dados

SET check\_function\_bodies = false;

/\* Enum 'en\_bolleano' \*/

CREATE TYPE en\_bolleano AS ENUM('sim', 'nao');

/\* Enum 'en\_tp\_avalicao' \*/

CREATE TYPE en\_tp\_avalicao AS ENUM('positivo', 'negativo');

/\* Sequence 'SQ\_Usuario' \*/

create sequence SQ\_Usuario owned by TB\_Usuario.id\_usuario;

/\* Sequence 'SQ\_Avalicao' \*/

create sequence SQ\_Avalicao owned by TB\_Avaliacao.id\_avaliacao;

/\* Sequence 'SQ\_Jogo' \*/

create sequence SQ\_Jogo owned by TB\_Jogo.id\_jogo;

/\* Sequence 'SQ\_Endereco' \*/

create sequence SQ\_Endereco owned by TB\_Endereco.id\_endereco;

/\* Sequence 'SQ\_Item' \*/

create sequence SQ\_Item owned by TB\_Item.id\_item;

/\* Sequence 'SQ\_Regra' \*/

create sequence SQ\_Regra owned by TB\_Regra.id\_regra;

/\* Sequence 'SQ\_Jogo\_Item' \*/

create sequence SQ\_Jogo\_Item owned by TB\_Jogo\_Item.id\_jogo\_item;

/\* Table 'TB\_Usuario' \*/

CREATE TABLE "TB\_Usuario"(

  id\_usuario integer NOT NULL,

  nr\_cpf\_cnpj varchar(11) NOT NULL,

  nm\_usuario varchar(50),

  ds\_email varchar(50) NOT NULL,

  ds\_senha varchar(20) NOT NULL,

  dt\_nascimento date,

  dt\_alteracao date,

  PRIMARY KEY(id\_usuario),

  CONSTRAINT un\_usuario\_cpf\_cnpj UNIQUE(nr\_cpf\_cnpj),

  CONSTRAINT un\_usuario\_email UNIQUE(ds\_email)

);

COMMENT ON TABLE "TB\_Usuario" IS

  'Tabela para armazenamento do dados do usuario.';

/\* Table 'TB\_Endereco' \*/

CREATE TABLE "TB\_Endereco"(

  id\_endereco integer NOT NULL,

  id\_usuario integer NOT NULL,

  nm\_pais varchar(15),

  cd\_uf varchar(2),

  nm\_cidade varchar(50),

  nr\_cep integer,

  nm\_rua varchar(50),

  nr\_endereco integer,

  fl\_valido en\_bolleano,

  dt\_alteracao date,

  PRIMARY KEY(id\_endereco)

);

COMMENT ON TABLE "TB\_Endereco" IS

  'Tabela para armazenamento dos enderecos do usuario.';

/\* Table 'TB\_Jogo' \*/

CREATE TABLE "TB\_Jogo"(

  id\_jogo integer NOT NULL,

  nm\_jogo varchar(50) NOT NULL,

  nr\_min\_jogadores integer NOT NULL,

  nr\_max\_jogadores integer,

  fl\_valido en\_bolleano,

  dt\_alteracao date,

  PRIMARY KEY(id\_jogo)

);

COMMENT ON TABLE "TB\_Jogo" IS 'Tabela para o cadastro dos jogos da plataforma.';

/\* Table 'TB\_Item' \*/

CREATE TABLE "TB\_Item"(

  id\_item integer NOT NULL,

  tp\_item varchar(50),

  nm\_item varchar(50),

  fl\_necessario en\_bolleano,

  PRIMARY KEY(id\_item)

);

COMMENT ON TABLE "TB\_Item" IS

  'Tabela para o cadastro de itens que podem ser usados para os jogos.';

/\* Table 'TB\_Jogo\_Item' \*/

CREATE TABLE "TB\_Jogo\_Item"(

  id\_jogo\_item integer NOT NULL,

  id\_jogo integer NOT NULL,

  id\_item integer NOT NULL,

  PRIMARY KEY(id\_jogo\_item)

);

COMMENT ON TABLE "TB\_Jogo\_Item" IS

  'Tebela relacionamento entre o item e o jogo que utilizara esse item.';

/\* Table 'TB\_Regra' \*/

CREATE TABLE "TB\_Regra"(

  id\_regra integer NOT NULL,

  id\_jogo integer NOT NULL,

  nm\_regra varchar(50) NOT NULL,

  ds\_regra text,

  fl\_opcional en\_bolleano,

  dt\_alteracao date,

  PRIMARY KEY(id\_regra)

);

COMMENT ON TABLE "TB\_Regra" IS 'Tabela para o cadastro das regras do jogo.';

/\* Table 'TB\_Avaliacao' \*/

CREATE TABLE "TB\_Avaliacao"(

  id\_avaliacao integer NOT NULL,

  id\_usuario integer NOT NULL,

  id\_jogo integer NOT NULL,

  ds\_avaliacao text,

  tp\_avaliacao en\_tp\_avalicao,

  fl\_jogado en\_bolleano,

  dt\_alteracao date,

  PRIMARY KEY(id\_avaliacao)

);

COMMENT ON TABLE "TB\_Avaliacao" IS

  'Tabela para o cadastro das avalizações do usuario.';

/\* Relation 'FK\_Endereco\_Usuario' \*/

ALTER TABLE "TB\_Endereco"

  ADD CONSTRAINT "FK\_Endereco\_Usuario"

    FOREIGN KEY (id\_usuario) REFERENCES "TB\_Usuario" (id\_usuario);

/\* Relation 'FK\_Item\_Jogo' \*/

ALTER TABLE "TB\_Jogo\_Item"

  ADD CONSTRAINT "FK\_Item\_Jogo"

    FOREIGN KEY (id\_jogo) REFERENCES "TB\_Jogo" (id\_jogo);

/\* Relation 'FK\_Jogo\_Item' \*/

ALTER TABLE "TB\_Jogo\_Item"

  ADD CONSTRAINT "FK\_Jogo\_Item"

    FOREIGN KEY (id\_item) REFERENCES "TB\_Item" (id\_item);

/\* Relation 'FK\_Regra\_Jogo' \*/

ALTER TABLE "TB\_Regra"

  ADD CONSTRAINT "FK\_Regra\_Jogo"

    FOREIGN KEY (id\_jogo) REFERENCES "TB\_Jogo" (id\_jogo);

/\* Relation 'FK\_Avaliacao\_Usuario' \*/

ALTER TABLE "TB\_Avaliacao"

  ADD CONSTRAINT "FK\_Avaliacao\_Usuario"

    FOREIGN KEY (id\_usuario) REFERENCES "TB\_Usuario" (id\_usuario);

/\* Relation 'FK\_Avaliacao\_Jogo' \*/

ALTER TABLE "TB\_Avaliacao"

  ADD CONSTRAINT "FK\_Avaliacao\_Jogo"

    FOREIGN KEY (id\_jogo) REFERENCES "TB\_Jogo" (id\_jogo);