.

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 5](#_Toc103951167)

[2 – Dicionário de Dados 6](#_Toc103951168)

[4 – Projeto Físico do Banco de Dado 11](#_Toc103951169)

[5 – Data Base Credentials 12](#_Toc103951170)

[6 – Data Definition Language – DDL 12](#_Toc103951171)

[7 – Data Manupulation Language – DML (INSERT) 17](#_Toc103951172)

[7 – Data Manupulation Language – DML (UPDATE / DELETE) 18](#_Toc103951173)

[8 – Data Query Language – DQL (SELECT) 19](#_Toc103951174)

[8.1 – Relatório simples contendo apenas uma tabela com dados ordenados. 19](#_Toc103951175)

# 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 |

## Tela de computador com jogo Descrição gerada automaticamente4 – Projeto Físico do Banco de Dado

## 5 – Data Base Credentials

**Host**

ec2-52-86-115-245.compute-1.amazonaws.com

**Database**

d92em23mmlh07d

**User**

yjixuohawbxqpj

**Port**

5432

**Password**

61a11b1ebe328f19fc17be47b54382fc78a89a613d3eaea7633ff4cf4a3d73ad

**URI**

postgres://yjixuohawbxqpj:61a11b1ebe328f19fc17be47b54382fc78a89a613d3eaea7633ff4cf4a3d73ad@ec2-52-86-115-245.compute-1.amazonaws.com:5432/d92em23mmlh07d

**Heroku CLI**

heroku pg:psql postgresql-amorphous-27570 --app almanak-fiap

## 6 – Data Definition Language – DDL

**DROP:**

drop sequence SQ\_Usuario ;

drop sequence SQ\_Avalicao;

drop sequence SQ\_Jogo;

drop sequence SQ\_Endereco;

drop sequence SQ\_Item;

drop sequence SQ\_Regra;

drop sequence SQ\_OneToOne;

drop table TB\_Endereco;

drop table TB\_Avaliacao;

drop table TB\_Jogo\_Item;

drop table TB\_Regra;

drop table TB\_Usuario;

drop table TB\_Jogo;

drop table TB\_Item;

drop table TV\_OneToOne;

**CREATE:**

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;

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

create sequence SQ\_Avaliacao;

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

create sequence SQ\_Jogo;

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

create sequence SQ\_Endereco;

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

create sequence SQ\_Item;

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

create sequence SQ\_Regra;

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

create sequence SQ\_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),

  CONSTRAINT un\_jogo\_nm\_jogo UNIQUE(nm\_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),

  CONSTRAINT un\_item\_nm\_item UNIQUE(nm\_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 varchar(500),

  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);

## 7 – Data Manupulation Language – DML (INSERT)

-- TB\_Usuario

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '41762758393', 'Nathan Miguel Julio da Rosa', 'nathan\_darosa@knowconsulting.com.br', 'cmU7akalUF', '1972-05-02', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '24360679700', 'Isadora Lorena Porto', 'isadoralorenaporto@elevaconsultancy.com.br', 'HtoZwZnNtQ', '1979-02-12', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '18454737952', 'Rita Joana Assis', 'rita\_joana\_assis@unitau.com.br', '3kXYAFizjI', '1987-02-08', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '92677700204', 'Nelson Renan Juan da Costa', 'nelson-dacosta81@aulicinobastos.com.br', 'iZAfgDdtoP', '1958-03-12', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '43591226670', 'Anthony Alexandre Ribeiro', 'anthony.alexandre.ribeiro@powerblade.com.br', 'thesZ9MxhK', '1976-05-16', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '7083270600', 'Amanda Mirella Luzia de Paula', 'amanda\_depaula@sectron.com.br', 'Ao2qauxK1p', '1987-03-04', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '11130881253', 'Theo Hugo Fernando Moreira', 'theo\_moreira@inductothermgroup.com.br', 'xJkhh6sqFE', '1973-04-25', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '51228335249', 'Noah Cauã André Costa', 'noah\_costa@fredericodiniz.com', 'feRfGlCQ2S', '1976-01-07', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '50163095230', 'Davi Victor Castro', 'davi.victor.castro@raya3.com.br', 'qtWxs3XzCB', '1974-01-02', current\_date);

INSERT INTO "TB\_Usuario" values (nextval('sq\_usuario'), '58042952090', 'Geraldo Enzo Oliveira', 'geraldo\_oliveira@systemsadvisers.com', 'o5d3A0e358', '1979-01-23', current\_date);

-- TB\_Endereco

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 1, 'Brasil', 'CE', 'Juazeiro do Norte', 63050080, 'Rua Afonso Melo', 615, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 2, 'Brasil', 'SE', 'Aracaju', 49092030, 'Rua Francisco Xavier da Paixão', 129, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 3, 'Brasil', 'DF', 'Brasília', 73350151, 'Quadra EQ 1/2 Bloco A', 611, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 4, 'Brasil', 'MG', 'Poços de Caldas', 37704388, 'Rua Luiz José Dias Neto', 160, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 5, 'Brasil', 'ES', 'Serra', 29166835, 'Rua das Mangas', 220, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 6, 'Brasil', 'AC', 'Rio Branco', 69906402, 'Rua Nossa Senhora de Fátima', 324, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 7, 'Brasil', 'GO', 'Aparecida de Goiânia', 74926786, 'Rua Hugo de Carvalho Ramos', 554, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 8, 'Brasil', 'SC', 'Joinville', 89224380, 'Rua Arraias', 670, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 9, 'Brasil', 'PI', 'Picos', 64600074, 'Travessa Joaquim Paraíba de Oliveira', 918, 'sim', current\_date);

INSERT INTO "TB\_Endereco" values (nextval('sq\_endereco'), 10, 'Brasil', 'GO', 'Catalão', 75704400, 'Rua Antônio Lopes Correia', 145, 'sim', current\_date);

-- TB\_Jogo

INSERT INTO "TB\_Jogo" values (nextval('sq\_jogo'), 'Taco', 4, null, 'sim', current\_date);

INSERT INTO "TB\_Jogo" values (nextval('sq\_jogo'), 'Pega Pega', 2, null, 'sim', current\_date);

INSERT INTO "TB\_Jogo" values (nextval('sq\_jogo'), 'Truco', 4, 6, 'sim', current\_date);

INSERT INTO "TB\_Jogo" values (nextval('sq\_jogo'), 'Sueca', 2, null, 'sim', current\_date);

-- TB\_Regra

INSERT INTO "TB\_Regra" values (nextval('sq\_regra'), 1, 'principal', 'Uma pessoa de cada dupla joga um taco para frente, com a missão de aproximá-lo ao máximo que puder de um determinado ponto – por exemplo: o meio fio de uma rua, ou um muro. Ganha essa a disputa o par do taco que aproximar mais perto do ponto escolhido. Se o bastão encostar ou passar do ponto, a dupla perde a disputa.', 'sim', current\_date);

INSERT INTO "TB\_Regra" values (nextval('sq\_regra'), 2, 'principal', 'Escolhe-se um pegador e os demais se espalham pelo espaço de jogo. Quando alguém é pego, dá a mão para o pegador e passa a atuar em dupla com ele. Em seguida, em trio, quarteto e assim sucessivamente, formando uma “corrente”, até que reste apenas um fugitivo, que será declarado vencedor.', 'sim', current\_date);

INSERT INTO "TB\_Regra" values (nextval('sq\_regra'), 3, 'principal', 'Em cada rodada, um jogador deve colocar uma de suas cartas na mesa, e o jogador com a carta mais forte vence a rodada. Quem ganhar 2 dessas rodadas ganha a mão e marca 1 ponto, e uma nova mão se inicia. Durante as rodadas, os jogadores têm a opção de pedir Truco, Seis, Nove e Doze, aumentando o valor da rodada.', 'sim', current\_date);

INSERT INTO "TB\_Regra" values (nextval('sq\_regra'), 4, 'principal', 'Rodada - uma sequência de 4 jogadas, onde cada jogador joga uma carta, e o vencedor leva as 4 cartas. Carta mais alta - a carta de maior valor do naipe. A ordem das cartas é a seguinte, da menor para a maior: 2, 3, 4, 5, 6, Q, J, K, 7, A. Seguir o naipe - jogar um carta do mesmo naipe da primeira carta da rodada.', 'sim', current\_date);

-- TB\_Item

INSERT INTO "TB\_Item" values (nextval('sq\_item'), 'Cartas', 'Baralho', 'sim');

INSERT INTO "TB\_Item" values (nextval('sq\_item'), 'Bebida', 'Bebida alcoolica', 'sim');

INSERT INTO "TB\_Item" values (nextval('sq\_item'), 'Bastao', 'Taco', 'sim');

INSERT INTO "TB\_Item" values (nextval('sq\_item'), 'Bola', 'Bola de futebol', 'sim');

-- TB\_Jogo\_Item

INSERT INTO "TB\_Jogo\_Item" values (nextval('sq\_jogo\_item'), 1, 1);

INSERT INTO "TB\_Jogo\_Item" values (nextval('sq\_jogo\_item'), 2, 2);

INSERT INTO "TB\_Jogo\_Item" values (nextval('sq\_jogo\_item'), 3, 3);

INSERT INTO "TB\_Jogo\_Item" values (nextval('sq\_jogo\_item'), 4, 4);

-- TB\_Avaliacao

INSERT INTO "TB\_Avaliacao" values (nextval('sq\_avaliacao'), 1, 1, null, 'positivo', 'sim', current\_date);

INSERT INTO "TB\_Avaliacao" values (nextval('sq\_avaliacao'), 2, 1, null, 'positivo', 'sim', current\_date);

INSERT INTO "TB\_Avaliacao" values (nextval('sq\_avaliacao'), 3, 1, null, 'positivo', 'sim', current\_date);

INSERT INTO "TB\_Avaliacao" values (nextval('sq\_avaliacao'), 1, 2, 'Jogo muito cansativo.', 'negativo', 'sim', current\_date);

# 7 – Data Manupulation Language – DML (UPDATE / DELETE)

-- UPDATE

update "TB\_Usuario" set ds\_senha = '123456' where id\_usuario = 1;

-- DELETE

delete from "TB\_Endereco" where id\_usuario = 10;

# 8 – Data Query Language – DQL (SELECT)

## 8.1 – Relatório simples contendo apenas uma tabela com dados ordenados.

-- Relatorio para acompanhamento das avaliacoes dos jogos

select

nm\_jogo,

tp\_avaliacao,

count(\*) as nr\_avaliacoes

from "TB\_Usuario"

inner join "TB\_Avaliacao"

    using (id\_usuario)

left join "TB\_Jogo"

    using (id\_jogo)

group by 1,2

order by 1,2