**PRORIO**

PROGRAMA DE ALOCAÇÃO DE HORÁRIO

**Autores**: Emerson Cantalice, Doglas Lima, Bruno Wesley.

**<LOGO>**

O sistema PRORIO tem como finalidade eliminar o tempo de processamento e organização dos dados que informam os horários disponíveis dos professores de uma instituição de ensino, com suas preferências de disciplina. Afim que o coordenador do curso seja capaz de visualizar essas informações e montar os horários individuais dos alunos.

Terá acesso ao sistema: coordenadores e professores. Coordenadores visualizaram os dados preenchidos por professores e professores preencheram seus horários disponíveis e preferências para o horário.

O sistema será web e contará com as seguintes tecnologias:

**- BANCO DE DADOS**

- Postgres 9.5

**- BACK-END**

- Java 9

- Spring Security

- Spring MVC (API RESTful)

- Persistência com JDBC

**- FRONT-END**

- AngularJs 1.5

**DICIONÁRIO (BANCO DE DADOS)**

**Tabela:**

Campo:

Descrição:

Campo:

Descrição:

Campo:

Descrição:

**Tabela:**

Campo:

Descrição:

Campo:

Descrição:

Campo:

Descrição:

**Tabela:**

Campo:

Descrição:

Campo:

Descrição:

Campo:

Descrição:

**Tabela:**

Campo:

Descrição:

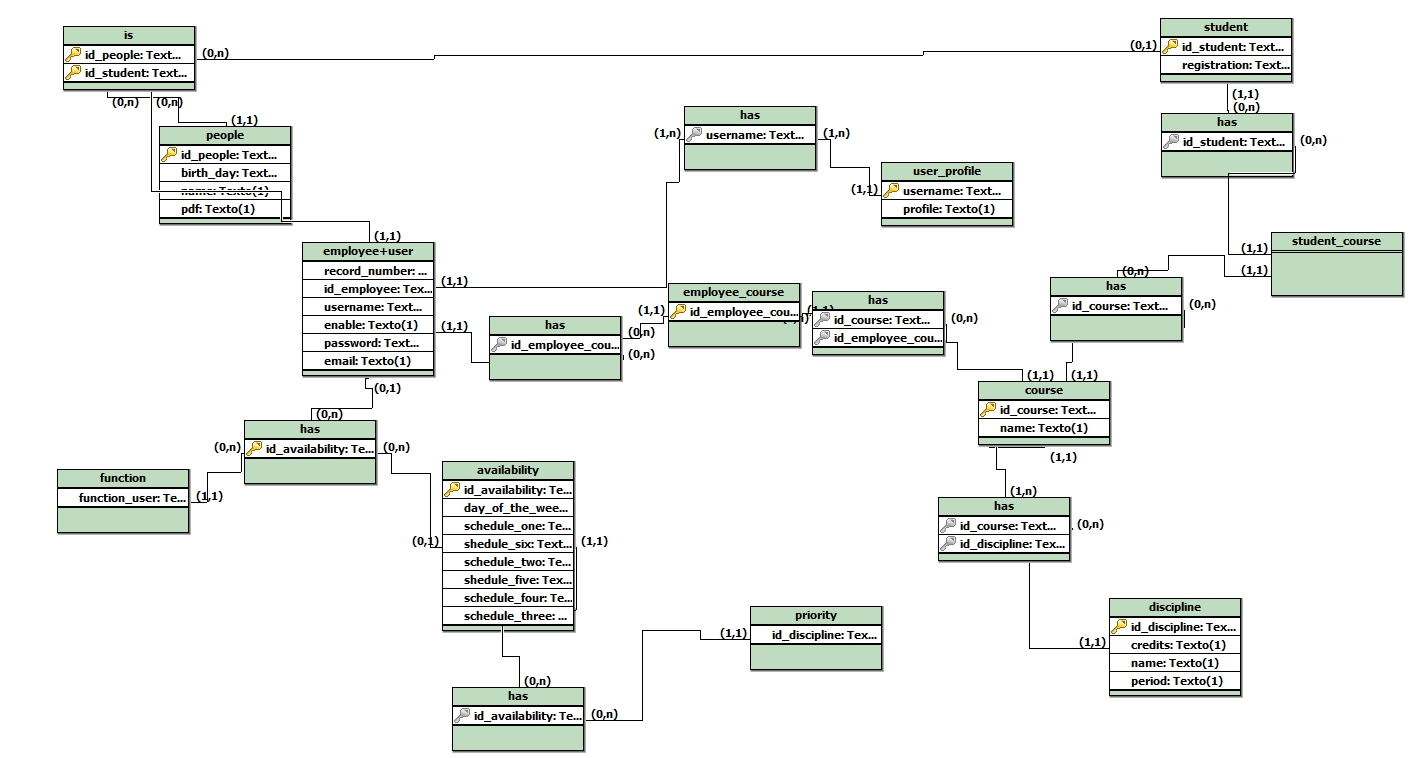
Campo:

Descrição:

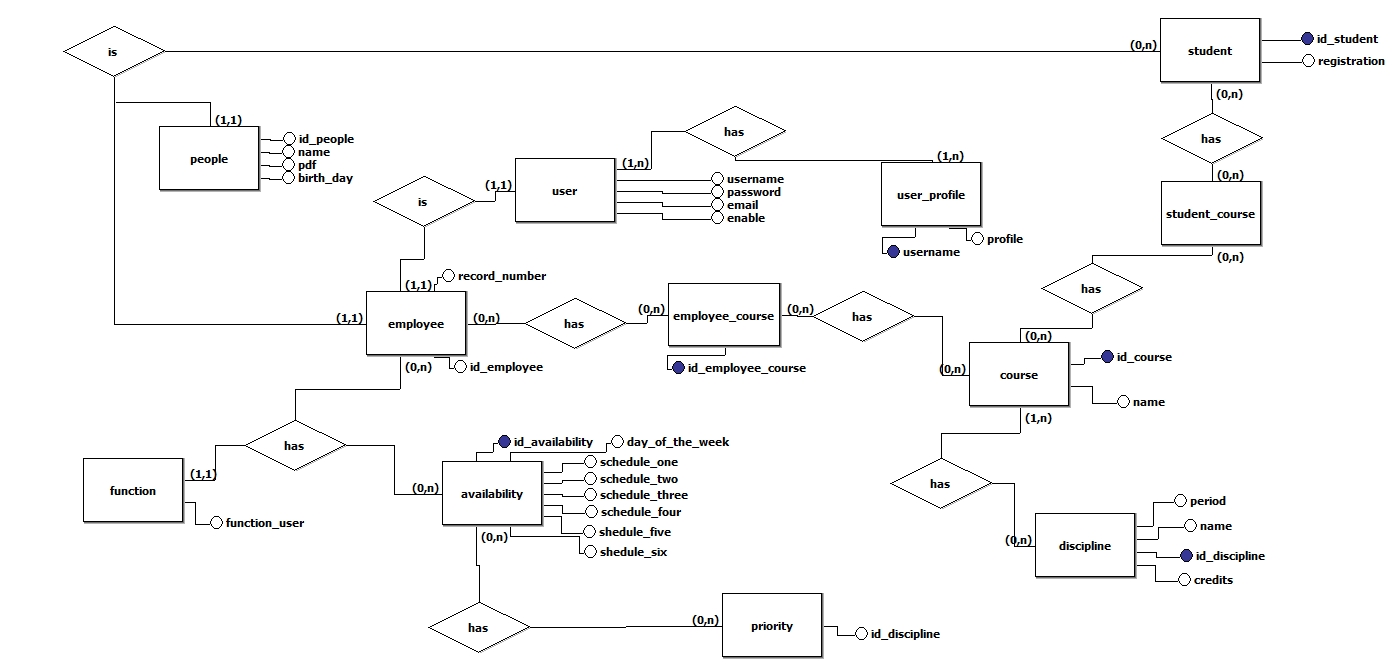
Campo:

Descrição:

**MODELO ENTIDADE RELACIONAMENTO**

****

**MODELO LOGICO E RELACIONAL**

****

**SCRIPT (DDL E DML)**

CREATE DATABASE prorio;

CREATE TYPE mode AS ENUM ('0','1');

CREATE TYPE function\_user AS ENUM ('teacher','coordinator');

CREATE TABLE people(

id\_people SERIAL,

name varchar(100) NOT NULL,

birth\_date Date NOT NULL,

cpf varchar(13) NOT NULL,

CONSTRAINT pk\_people PRIMARY KEY (id\_people)

);

CREATE TABLE officials(

id\_employee SERIAL,

id\_people int,

record\_number varchar(13) NOT NULL,

CONSTRAINT pk\_employee PRIMARY KEY (id\_employee),

CONSTRAINT fk\_employee\_people FOREIGN KEY (id\_people) REFERENCES people (id\_people) ON DELETE CASCADE

);

CREATE TABLE functions(

id\_employee int,

function function\_user,

CONSTRAINT fk\_employee FOREIGN KEY (id\_employee) REFERENCES officials (id\_employee) ON DELETE CASCADE

);

CREATE TABLE availability(

id\_availability serial,

id\_employee int UNIQUE,

Day\_of\_the\_week varchar(15) not null,

schedule\_one mode NOT NULL,

schedule\_two mode NOT NULL,

schedule\_three mode NOT NULL,

schedule\_four mode NOT NULL,

schedule\_five mode NOT NULL,

schedule\_six mode NOT NULL,

CONSTRAINT pk\_availability PRIMARY KEY (id\_availability),

CONSTRAINT fk\_employee FOREIGN KEY (id\_employee) REFERENCES officials (id\_employee) ON DELETE CASCADE

);

CREATE TABLE priorities(

id\_availability int UNIQUE,

schedule\_one VARCHAR(50),

schedule\_two VARCHAR(50),

schedule\_three VARCHAR(50),

schedule\_four VARCHAR(50),

schedule\_five VARCHAR(50),

schedule\_six VARCHAR(50),

CONSTRAINT fk\_priorities\_availability FOREIGN KEY (id\_availability) REFERENCES availability (id\_availability) ON DELETE CASCADE

);

CREATE TABLE users (

id\_employee int,

username VARCHAR(45),

password VARCHAR(45) NOT NULL ,

enabled int NOT NULL DEFAULT 1,

email varchar(60) NOT NULL UNIQUE,

CONSTRAINT pk\_username PRIMARY KEY (username),

CONSTRAINT fk\_user\_employee FOREIGN KEY (id\_employee) REFERENCES officials (id\_employee) ON DELETE CASCADE

);

CREATE TABLE user\_profile (

id\_profile SERIAL,

username varchar(45) NOT NULL,

role varchar(45) NOT NULL,

PRIMARY KEY (id\_profile),

CONSTRAINT fk\_user FOREIGN KEY (username) REFERENCES users(username) ON DELETE CASCADE

);

CREATE TABLE courses (

id\_course SERIAL,

name varchar(45) NOT NULL UNIQUE,

PRIMARY KEY (id\_course)

);

CREATE TABLE disciplines (

id\_discipline SERIAL,

name varchar(45) NOT NULL,

id\_course int NOT NULL,

period int NOT NULL,

credits int NOT NULL,

PRIMARY KEY (id\_discipline),

CONSTRAINT fk\_user FOREIGN KEY (id\_course) REFERENCES courses(id\_course) ON DELETE CASCADE

);

CREATE TABLE employee\_courses(

id\_employee\_course SERIAL,

id\_employee int,

id\_course int,

CONSTRAINT pk\_employee\_course PRIMARY KEY (id\_employee\_course),

CONSTRAINT fk\_employee\_ref FOREIGN KEY (id\_employee) REFERENCES officials (id\_employee) ON DELETE CASCADE,

CONSTRAINT fk\_course FOREIGN KEY (id\_course) REFERENCES courses (id\_course) ON DELETE CASCADE

);

CREATE TABLE students(

id\_student SERIAL,

id\_people int,

registration varchar(13)not null,

CONSTRAINT pk\_student PRIMARY KEY (id\_student),

CONSTRAINT fk\_student FOREIGN KEY (id\_student) REFERENCES people (id\_people) ON DELETE CASCADE

);

CREATE TABLE students\_courses(

id\_student\_course SERIAL,

id\_student int not null,

id\_course int not null,

CONSTRAINT pk\_student\_course PRIMARY KEY (id\_student\_course),

CONSTRAINT fk\_student FOREIGN KEY (id\_student) REFERENCES students (id\_student) ON DELETE CASCADE,

CONSTRAINT fk\_course FOREIGN KEY (id\_course) REFERENCES courses (id\_course) ON DELETE CASCADE

);

**CARGA**

INSERT INTO people(name,birth\_date,cpf) VALUES

('Emerson','12-12-1991','123456'),

('Junior','12-12-1991', '132456'),

('Matheus','12-12-1991', '13245'),

('Amanda','12-12-1991', '132572'),

('Fernanda','12-12-1991', '132456'),

('Leydson','12-12-1991', '13245'),

('Germano','12-12-1991', '13245'),

('Alberto','12-12-1991', '132572');

INSERT INTO officials(id\_people,record\_number) VALUES (1,'123456'),(2,'123457'), (3,'123458'), (4,'123958');

INSERT into functions(id\_employee,function) values

(1,'coordinator'), (2,'teacher'), (3,'teacher'), (4,'teacher');

INSERT INTO availability(id\_employee,Day\_of\_the\_week ,schedule\_one, schedule\_two, schedule\_three, schedule\_four,schedule\_five,schedule\_six) VALUES

(1,'segunda','0','1','0','1','0','1'),

(2,'terça','1','1','1','1','0','1'),

(3,'segunda','1','1','0','0','0','1'),

(4,'quinta','0','0','1','1','0','1');

INSERT INTO priorities(id\_availability,schedule\_one, schedule\_two, schedule\_three, schedule\_four,schedule\_five,schedule\_six) VALUES

(1,'Programação 1',null ,null ,'Programação 2',null ,null ),

(2,null ,'Programação 1',null ,null ,null ,null ),

(3,null ,null ,null ,null ,null ,null ) ,

(4,null ,null ,null ,'Programação 12',null ,null );

INSERT INTO users(id\_employee,username, password, enabled, email) VALUES

(1,'escantalice', '123', 1, 'emersoncantalicee@gmail.com'),

(2,'maoliveira', '123', 1, 'matheusoliveira@gmail.com'),

(3,'ebrito', '123', 1, 'ebrito@gmail.com');

INSERT INTO user\_profile(username,role) VALUES

('escantalice', 'ROLE\_ADMIN'),

('maoliveira', 'ROLE\_DBA'),

('ebrito', 'ROLE\_USER'),

('escantalice', 'ROLE\_USER');

INSERT INTO courses(id\_course, name) VALUES

(1,'Sistemas de Informação'),

(2,'Administração'),

(3,'Medicina');

INSERT INTO disciplines(id\_course, name, period, credits) VALUES

(1, 'Programação 1', 1, 2),

(1, 'Praticas de Programação', 3, 2), (1, 'Programação 2', 2, 2) (1, 'Praticas de Programa (2, 'Adminstração Básica', 1, 1),

(2, 'Teoria da Adminstração', 1, 2),

(2, 'Analise de recursos', 2, 1),

(3, 'Fisiologia', 1, 2),

(3, 'Doenças Criticas', 2, 1),

(3, 'Farmacologia', 3, 1);

INSERT INTO employee\_courses(id\_employee,id\_course) VALUES (1,1),(2,2),(3,3),(4,1);

INSERT INTO students(id\_people, registration) VALUES

(5,'1423080002'),

(6,'1423080003'),

(7,'1423080004'),

(8,'1423080005');

INSERT into students\_courses(id\_student,id\_course) values (1,1),(2,2),(3,3),(4,3);

**CONSULTAS**

**select**

people.name,courses.name, students.id\_student

**from**

people, students, students\_courses, courses

**where**

students.id\_student = students\_courses.id\_student

**and**

courses.id\_course = students\_courses.id\_course

**and**

people.id\_people = students.id\_student

**and**

students\_courses.id\_student = 2;