**PRORIO**

PROGRAMA DE ALOCAÇÃO DE HORÁRIO

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



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

**DICIONÁRIO (BANCO DE DADOS)**

**Tabela: People**

Campo: id\_people

Descrição: Identificador(Gerado automaticamente).

Campo: name

Descrição: Nome da Pessoa.

Campo: birth\_day

Descrição: Ano de Nascimento da Pessoa.

Campo: cpf

Descrição: CPF referente à Pessoa.

**Tabela: Officials**

Campo: id\_employee

Descrição: Identificador (Gerado automaticamente)

Campo: id\_people

Descrição: Chave estrangeira identificador referente a tabela People.

Campo: record\_number

Descrição: Número da matricula do funcionário.

**Tabela: Functions**

Campo: id\_employee

Descrição: Chave estrangeira identificador referente a tabela Officials.

Campo: function\_user

Descrição: Define o cargo do funcionário (teacher ou coordinator)

**Tabela: Availability**

Campo: id\_availability

Descrição: Identificador (Gerado automaticamente)

Campo: id\_employee

Descrição: Chave estrangeira identificador referente a tabela Officials.

Campo: Day\_of\_the\_week

Descrição: Dia da semana em que se encontra disponível.

Campo: schedule\_one

Descrição: Disponibilidade no primeiro horário da manhã (0 = não disponível, 1 = disponível).

Campo: schedule\_two

Descrição: Disponibilidade no segundo horário da manhã (0 = não disponível, 1 = disponível).

Campo: schedule\_three

Descrição: Disponibilidade no primeiro horário da tarde (0 = não disponível, 1 = disponível).

Campo: schedule\_four

Descrição: Disponibilidade no segundo horário da tarde (0 = não disponível, 1 = disponível).

Campo: schedule\_five

Descrição: Disponibilidade no primeiro horário da noite (0 = não disponível, 1 = disponível).

Campo: schedule\_six

Descrição: Disponibilidade no segundo horário da noite (0 = não disponível, 1 = disponível).

**Tabela:** **Priorities**

Campo: id\_availability

Descrição:Chave estrangeira identificador referente a tabela Availability

Campo: schedule\_one

Descrição: Preferencia da disciplina no primeiro horário manhã.

Campo: schedule\_two

Descrição: Preferencia da disciplina no segundo horário manhã.

Campo: schedule\_three

Descrição: Preferencia da disciplina no primeiro horário tarde.

Campo: schedule\_four

Descrição: Preferencia da disciplina no segundo horário tarde.

Campo: schedule\_five

Descrição: Preferencia da disciplina no primeiro horário noite.

Campo: schedule\_six

Descrição: Preferencia da disciplina no segundo horário noite.

**Tabela:** **Users**

Campo: id\_employee

Descrição: Chave estrangeira identificador referente a tabela Officials

Campo: username

Descrição: Nome de usuário para login.

Campo: password

Descrição: Chave de acesso para login.

Campo: enabled

Descrição: Status de disponibilidade do Usuário.

Campo: email

Descrição: Email do usuário.

**Tabela: user\_profile**

Campo: id\_profile

Descrição:

Campo: username

Descrição: Entrada de acesso do usuário.

Campo: role

Descrição: Cargo, função do usuário.

**Tabela: courses**

Campo: id\_course

Descrição: Identificador (Gerado automaticamente)

Campo: name

Descrição: Nome do curso.

**Tabela: disciplines**

Campo: id\_discipline

Descrição: Identificador (Gerado automaticamente)

Campo: name

Descrição: Nome da disciplina

Campo: id\_course

Descrição: Chave estrangeira identificador referente a tabela Courses.

Campo: period

Descrição: Período a qual a disciplina é alocada em sua matriz.

Campo: credits

Descrição: Valor de créditos da disciplina.

**Tabela: employee\_courses**

Campo: id\_employee\_course

Descrição: Identificador (Gerado automaticamente)

Campo: id\_employee

Descrição: Chave estrangeira identificador referente a tabela Officials.

Campo: id\_course

Descrição: Chave estrangeira identificador referente a tabela Courses.

**Tabela: students**

Campo: id\_student

Descrição: Identificador (Gerado automaticamente)

Campo: id\_people

Descrição: Chave estrangeira identificador referente a tabela People.

Campo: registration

Descrição: Matricula do aluno.

**Tabela: students\_courses**

Campo: id\_student\_course

Descrição: Identificador (Gerado automaticamente)

Campo: id\_student

Descrição: Chave estrangeira identificador referente a tabela Student.

Campo: id\_course

Descrição: Chave estrangeira identificador referente a tabela Courses.

**Tipo : Mode**

Valor: 0

Descrição: Não disponível

Valor: 1

Descrição: Disponível

**Tipo : function\_user**

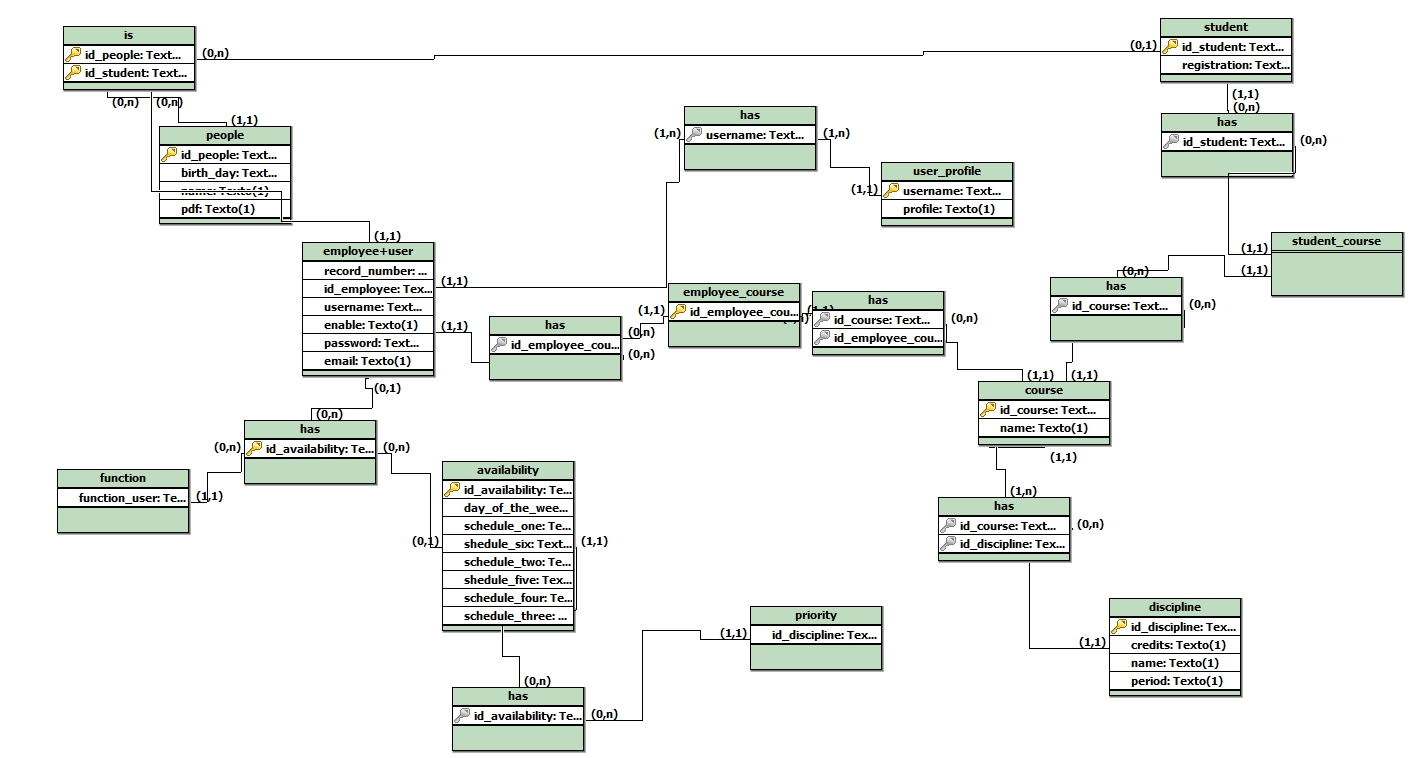
Valor: teacher

Descrição: Usuário função, cargo professor;

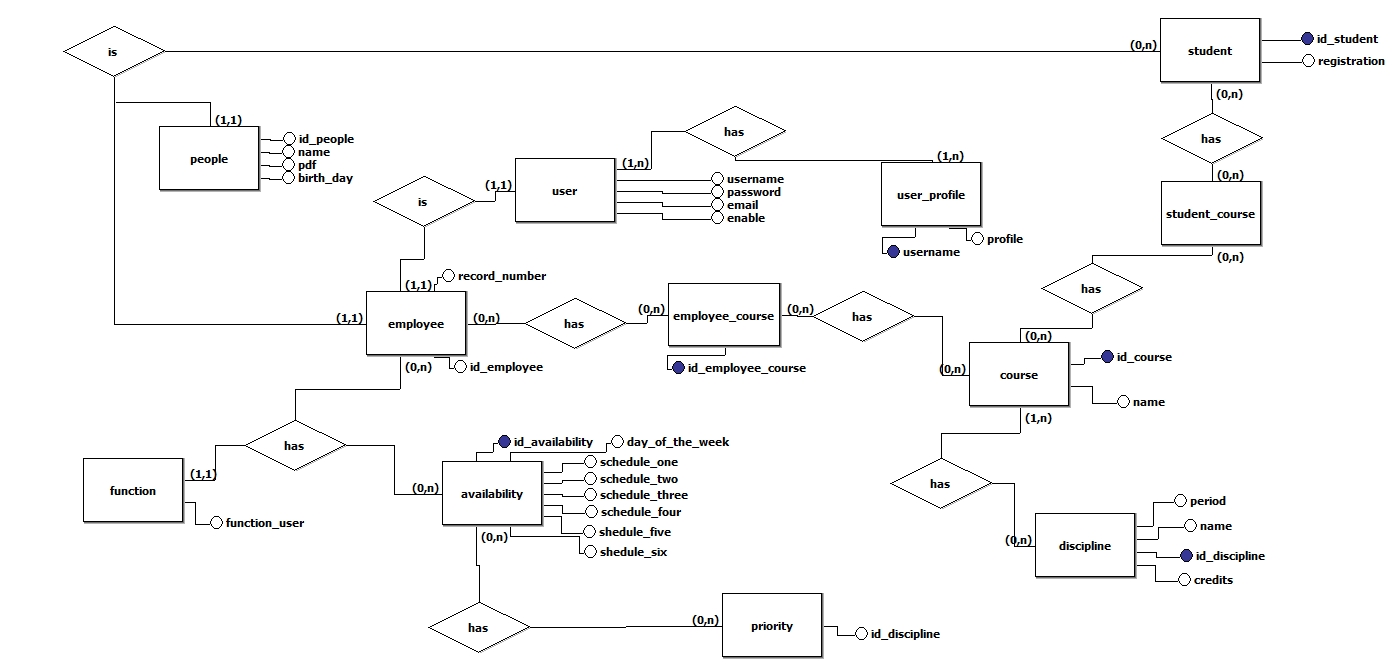
Valor: coordinator

Descrição: Usuário função, cargo coordenador.

**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;