



Universidade Federal de Sergipe
CCET – DCOMP
Disciplina: *Sistemas de Apoio à Decisão*
Professor: *Gilton José Ferreira da Silva*
Aluno: *Matheus Mendonça Menezes*

Atividade 4

a) Criar e configurar uma conta em um repositório GIT:

As figuras abaixo ilustram os processos de criação e configuração de uma conta GIT.

Site: <https://github.com>

The screenshot shows the GitHub 'Join GitHub' page. At the top, it says 'Join GitHub' and 'The best way to design, build, and ship software.' Below this is a progress bar with three steps: 'Step 1: Set up a personal account' (active), 'Step 2: Choose your plan', and 'Step 3: Tailor your experience'. The main section is 'Create your personal account'. It has three input fields: 'Username' (with a note: 'This will be your username — you can enter your organization's username next.'), 'Email Address' (with a note: 'You will occasionally receive account related emails. We promise not to share your email with anyone.'), and 'Password' (with a note: 'Use at least one lowercase letter, one numeral, and seven characters.'). Below the fields is a checkbox area with the text: 'By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).' At the bottom is a green button labeled 'Create an account'.

Join GitHub
The best way to design, build, and ship software.

Step 1: Set up a personal account
Step 2: Choose your plan
Step 3: Tailor your experience

Create your personal account

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

You'll love GitHub

- Unlimited collaborators
- Unlimited public repositories
- ✓ Great communication
- ✓ Frictionless development
- ✓ Open source community

Figura 1: Criação de conta no GitHub

Para se criar uma conta no GitHub é necessário escolher:

- Username: Nome de usuário;
- Email Address: Endereço de e-mail;
- Password: Senha.

Após terem sido inseridos os dados acima, o usuário receberá um e-mail de confirmação, através do qual validará sua conta. A partir daí será possível criar novos repositórios, como será descrito a seguir.

Este tutorial cobre a criação de um repositório fazendo uso do **GitHubDesktop**, o mesmo pode ser obtido no site: <https://desktop.github.com> e possui versões para Mac e Windows, como ilustrado na figura abaixo.

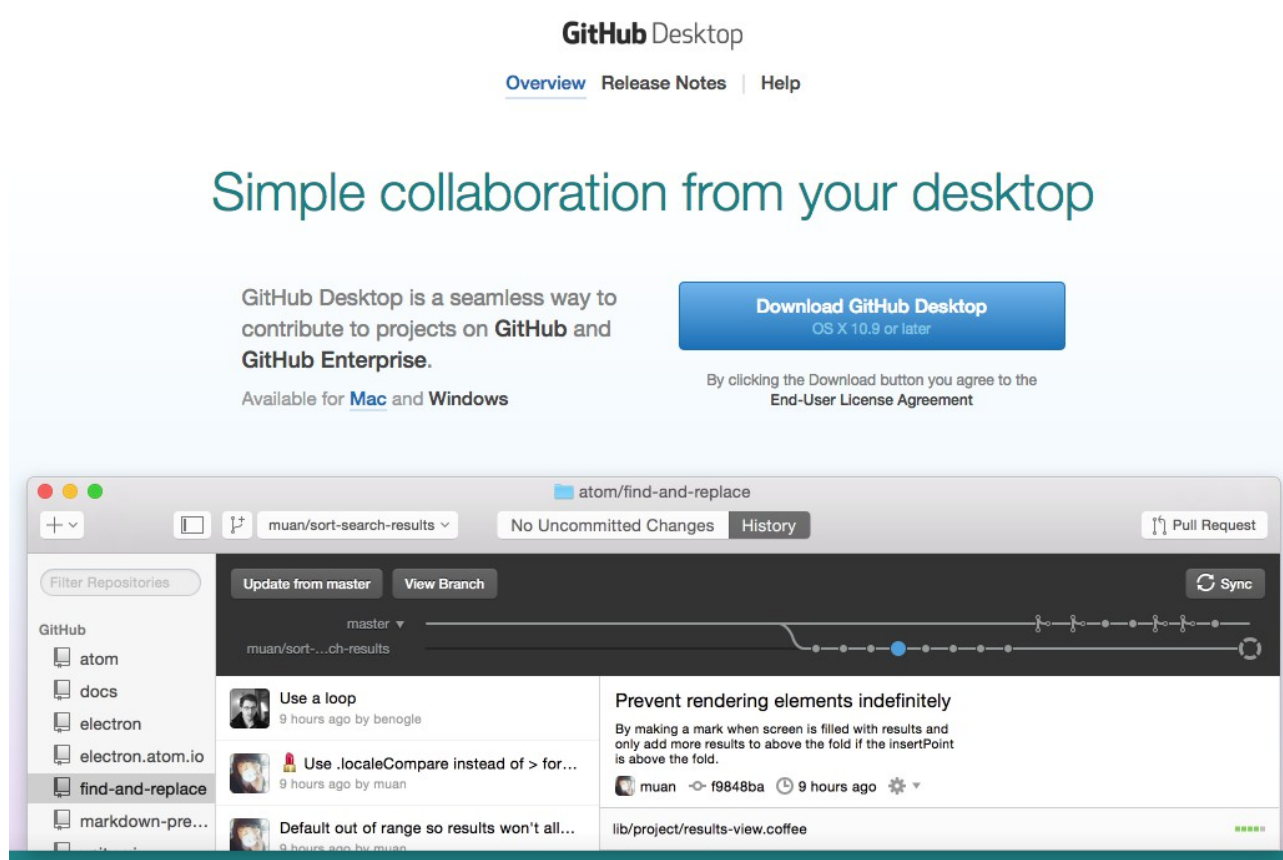


Figura 2: Página de download do GitHub Desktop

Após ter criado a conta e instalado o GitHub Desktop, chegou a hora de criar um repositório. Para este trabalho foi usada a versão para Mac do aplicativo, a figura abaixo ilustra o processo de criação de um repositório no citado aplicativo. Como pode ser observado, foi criado o repositório **sad_ufs_[matheus]_[menezes]**.

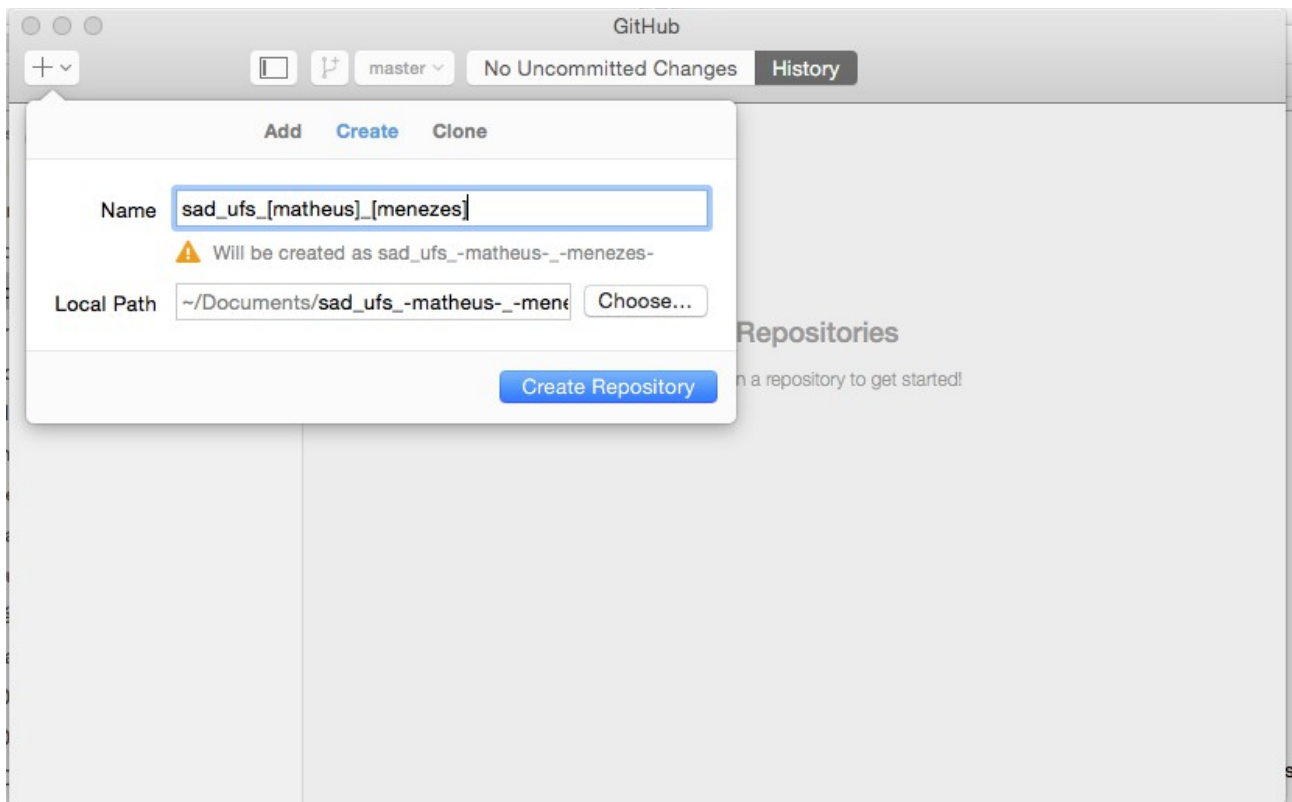


Figura 3: Criação de repositório - passo 1

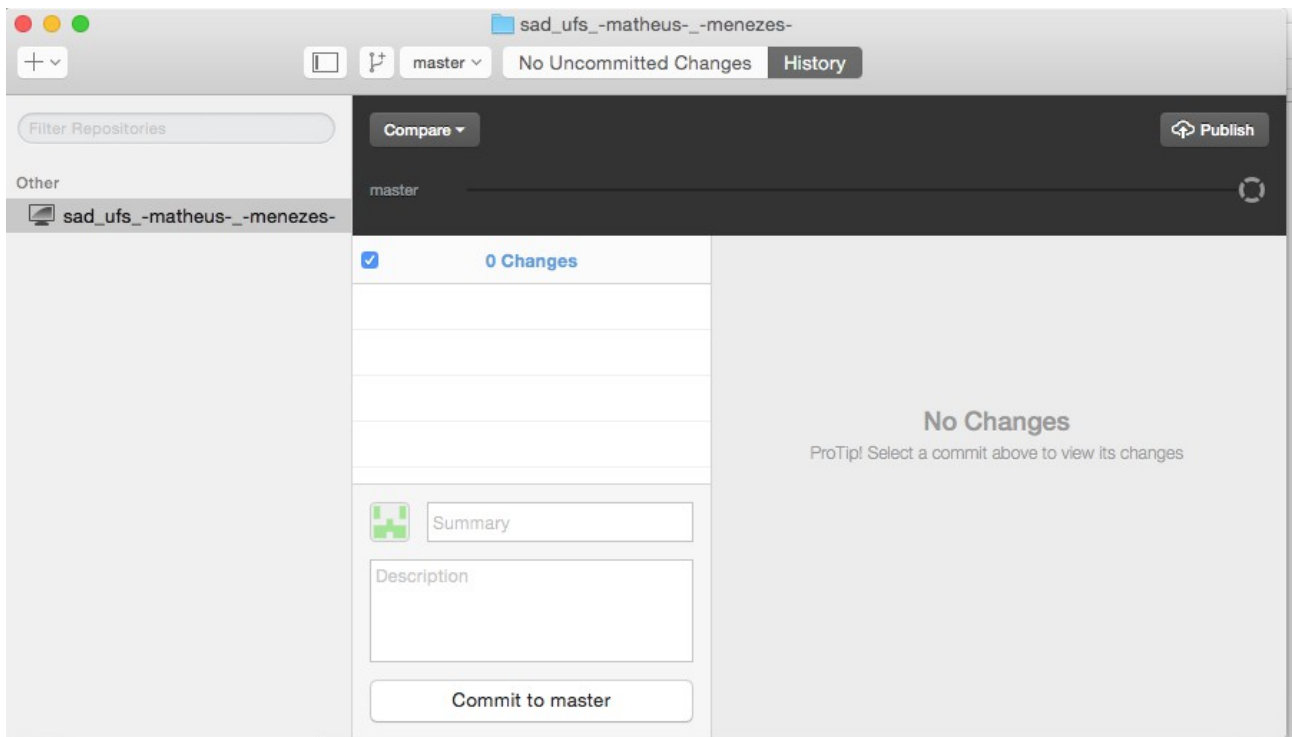


Figura 4: Criação de repositório - passo 2

b) Modele e crie um banco de dados relacional com no mínimo 8 (oito) tabelas:

Para a criação e modelagem do banco de dados foi usado o PostgreSQL. Foi feito o download do mesmo em: <https://www.postgresql.org/download/macosx/>. Este trabalho fez uso do Postgres.app para Mac, que roda diretamente da barra de menu do sistema, sem necessidade de instalação. As Figuras 5 e 6 ilustram o servidor PostgreSQL rodando no Mac OSX.



Figura 5: Servidor PostgreSQL no Mac

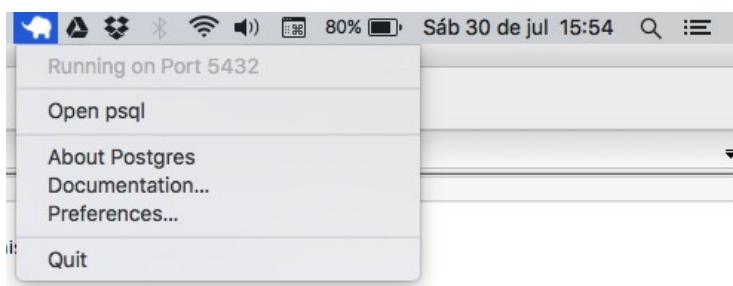


Figura 6: Postgres.app rodando na barra de menu do Mac

Para administração do banco de dados PostgreSQL foi utilizado a aplicação pgAdmin, também para Mac, que pode ser baixado em <https://www.pgadmin.org/download/macosx.php>. Vale ressaltar que tanto o PostgreSQL quanto o pgAdmin podem ser usados gratuitamente em Linux e Windows também. As Figuras 7 e 8 mostram o pgAdmin se conectando ao servidor PostgreSQL e criando um banco de dados chamado **matheus**. O servidor roda por padrão na porta 5432. O *username* e *password* são os mesmos utilizados pelo usuário do sistema Mac OSX.

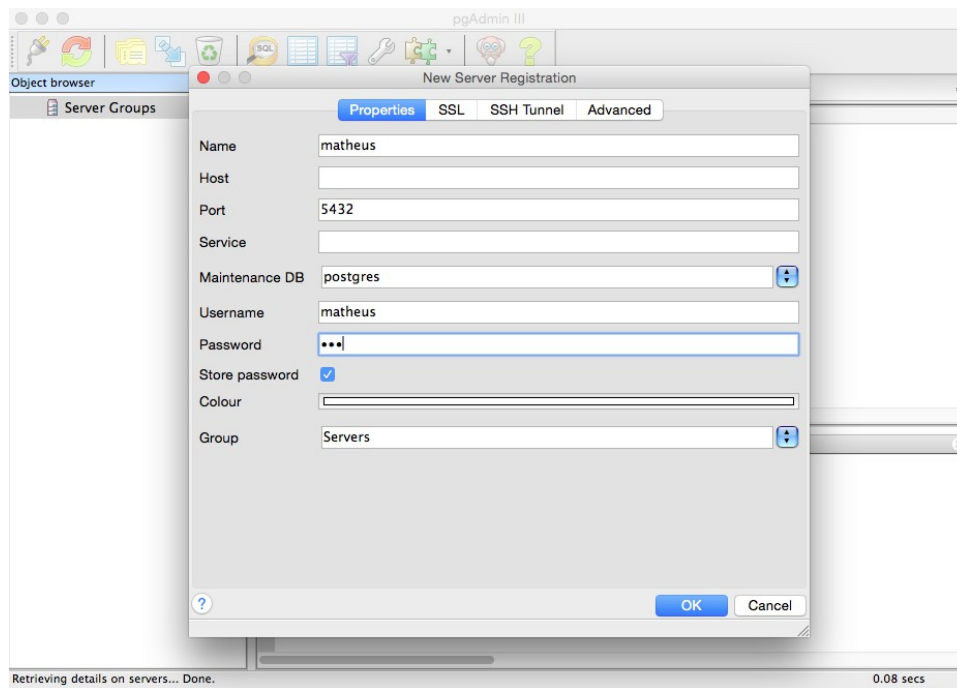


Figura 7: Adicionando servidor ao pgAdmin

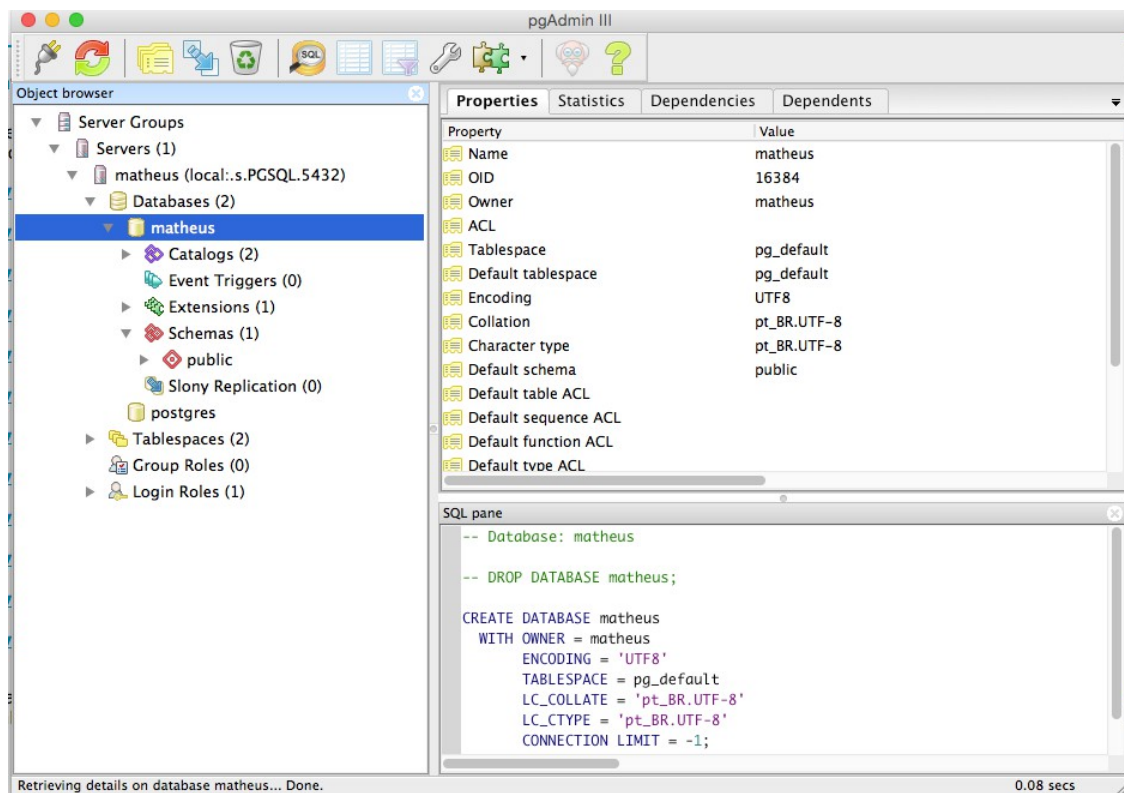


Figura 8: pgAdmin conectado ao servidor PostgreSQL local

As Figuras 9 e 10 ilustram o processo de criação de um *schema* no banco de dados “matheus”, onde serão dispostas as tabelas referentes à “**Atividade_4**”.

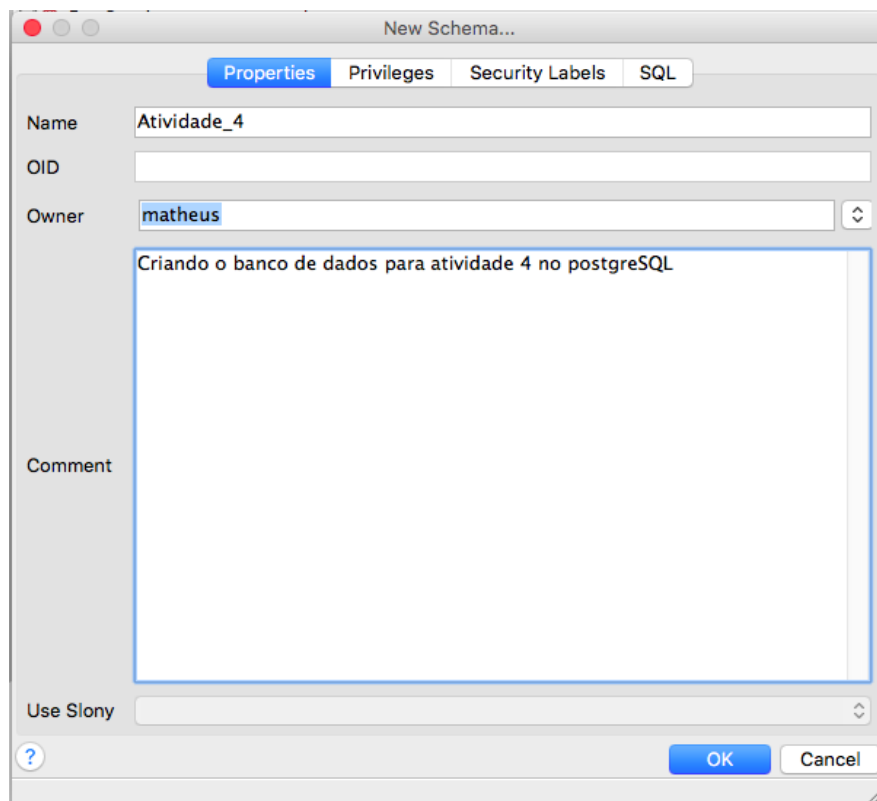


Figura 9: Criação de schema

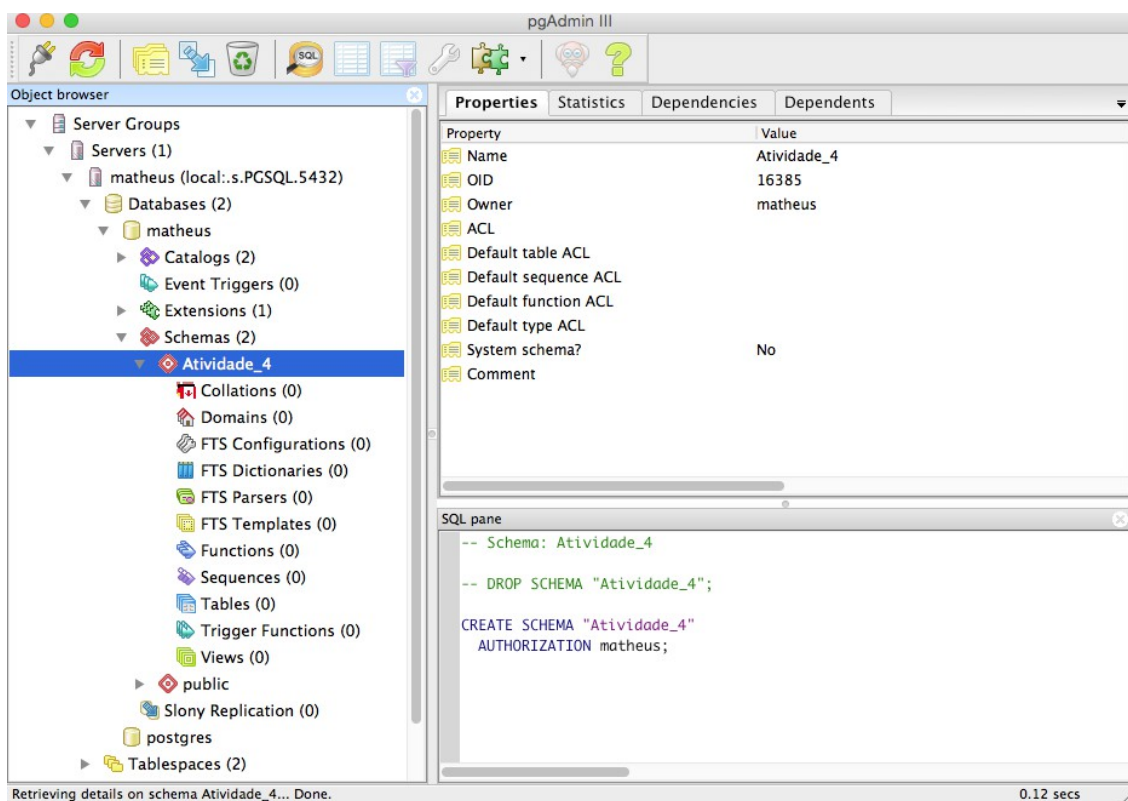


Figura 10: Schema criado

O script usado para a criação das tabelas foi:

Ficheiro: /Users/matheus/Desktop/cria tabelas.sql

```
CREATE SCHEMA atividade_4

CREATE TABLE Atividade_4.funcionario(
    Pnome VARCHAR(30) NOT NULL,
    Snome VARCHAR(150) NOT NULL,
    CPF VARCHAR(11),
    datanasc DATE,
    endereco VARCHAR(8),
    sex CHAR(1),
    salario REAL CHECK(salario >0),
    CPF_Supervisor VARCHAR(11),
    Dnr INT,
    PRIMARY KEY(CPF),
    FOREIGN KEY (CPF Supervisor) REFERENCES Atividade_4.funcionario(CPF)
    ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE Atividade_4.departamento (
    Dnome VARCHAR(30) NOT NULL,
    Dnumero INT, CPF_Gerente VARCHAR(11),
    data_inicio_gerente DATE,
    PRIMARY KEY(Dnumero),
    FOREIGN KEY (CPF Gerente) REFERENCES Atividade_4.funcionario(CPF)
    ON DELETE SET NULL ON UPDATE CASCADE
);

ALTER TABLE Atividade_4.funcionario ADD CONSTRAINT depto FOREIGN KEY (Dnr) REFERENCES
Atividade_4.departamento(Dnumero)
    ON DELETE SET NULL ON UPDATE CASCADE;

CREATE TABLE Atividade_4.localizacao_dep(
    Dnumero INT,
    DLocal VARCHAR(30),
    PRIMARY KEY(Dnumero, DLocal),
    FOREIGN KEY (Dnumero)
    REFERENCES Atividade_4.departamento(Dnumero)
    ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE Atividade_4.projeto(
    projnome VARCHAR(30) NOT NULL,
    projnumero INT, projlocal VARCHAR(30),
    Dnum INT, PRIMARY KEY(projnumero),
    FOREIGN KEY (Dnum) REFERENCES Atividade_4.departamento(Dnumero)
    ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE Atividade_4.trabalha_em(
    Fcpf VARCHAR(11),
    Pnr INT, horas INT,
    PRIMARY KEY(Fcpf, Pnr),
    FOREIGN KEY (Fcpf) REFERENCES Atividade_4.funcionario(CPF)
    ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (Pnr) REFERENCES Atividade_4.projeto(projnumero)
    ON DELETE SET NULL ON UPDATE CASCADE);

CREATE TABLE Atividade_4.dependente(
    Fcpf VARCHAR(11),
    nome_dependente VARCHAR(200),
    sexo CHAR(1),
    datanasc DATE,
    parentesco VARCHAR(30),
    PRIMARY KEY(Fcpf, nome_dependente),
    FOREIGN KEY (Fcpf) REFERENCES Atividade_4.funcionario(CPF)
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Atividade_4.formacao(
    Fcpf VARCHAR(11),
    formacao_academica VARCHAR(60),
    universidade VARCHAR(60),
    PRIMARY KEY(Fcpf),
    FOREIGN KEY (Fcpf) REFERENCES Atividade_4.funcionario(CPF)
    ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE Atividade_4.encomendado_por(
    Fcpf VARCHAR(11),
    encomendou VARCHAR(60),
    pais VARCHAR(60),
    PRIMARY KEY(Fcpf),
    FOREIGN KEY (Fcpf) REFERENCES Atividade_4.funcionario(CPF)
    ON DELETE SET NULL ON UPDATE CASCADE
);
```

Figura 11: Script SQL - parte 2

A Figura 12 ilustra as tabelas já criadas e disponíveis no pgAdmin, onde podem ser analisadas individualmente. Na figura 13 estão detalhados os atributos da tabela

atividade_4.departamento.

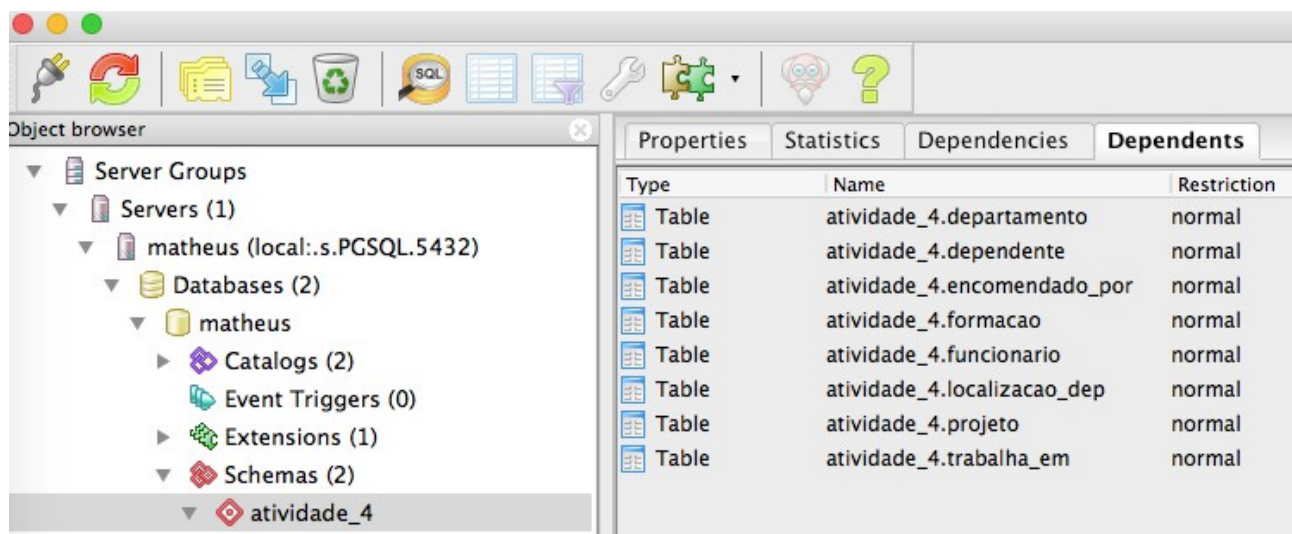


Figura 12: Tabelas do banco

```
matheus=# matheus=# \d atividade_4.departamento
Table "atividade_4.departamento"

```

Column	Type	Modifiers
dnome	character varying(30)	not null
dnumero	integer	not null
cpf_gerente	character varying(11)	
data_inicio_gerente	date	

Indexes:

Figura 13: Detalhe da tabela departamento

Para a modelagem do banco de dados foi usada a aplicação **SQL Power Architect**, em sua versão para Mac. A aplicação possui também versões para outros sistemas como Linux e Windows. A Figura 14 registra o SQL Power Architect em execução para as atividades deste trabalho.

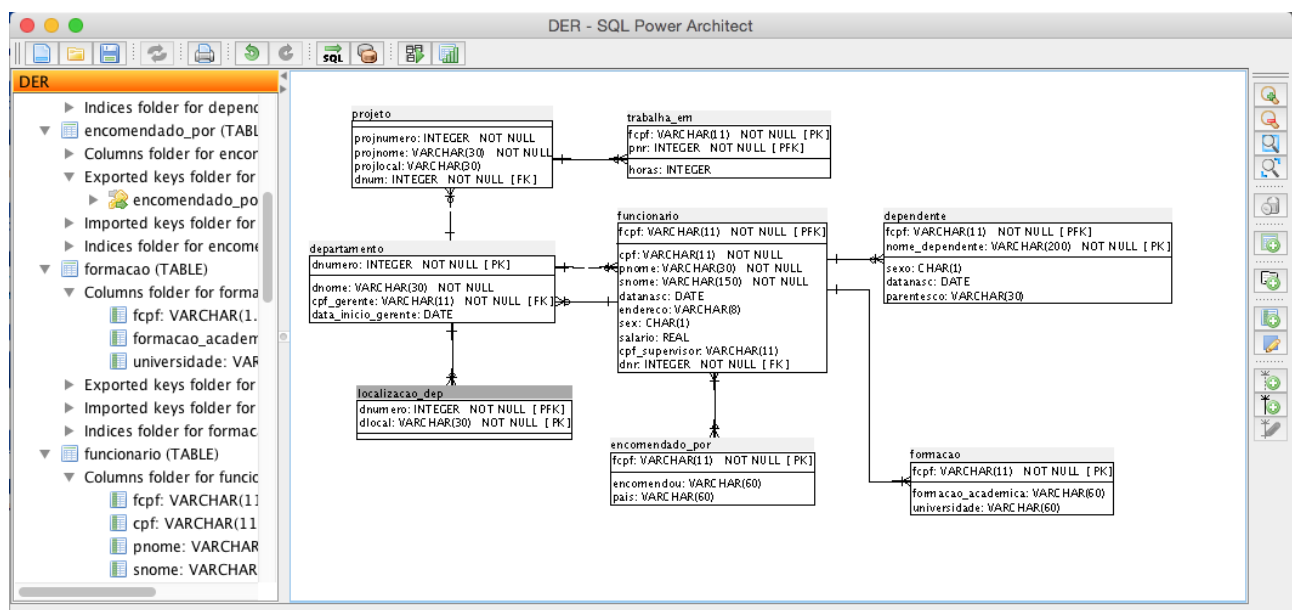


Figura 14: Modelagem no SQL Power Architect

A Figura 15 mostra a tela inicial do Pentaho em execução no Mac OSX.

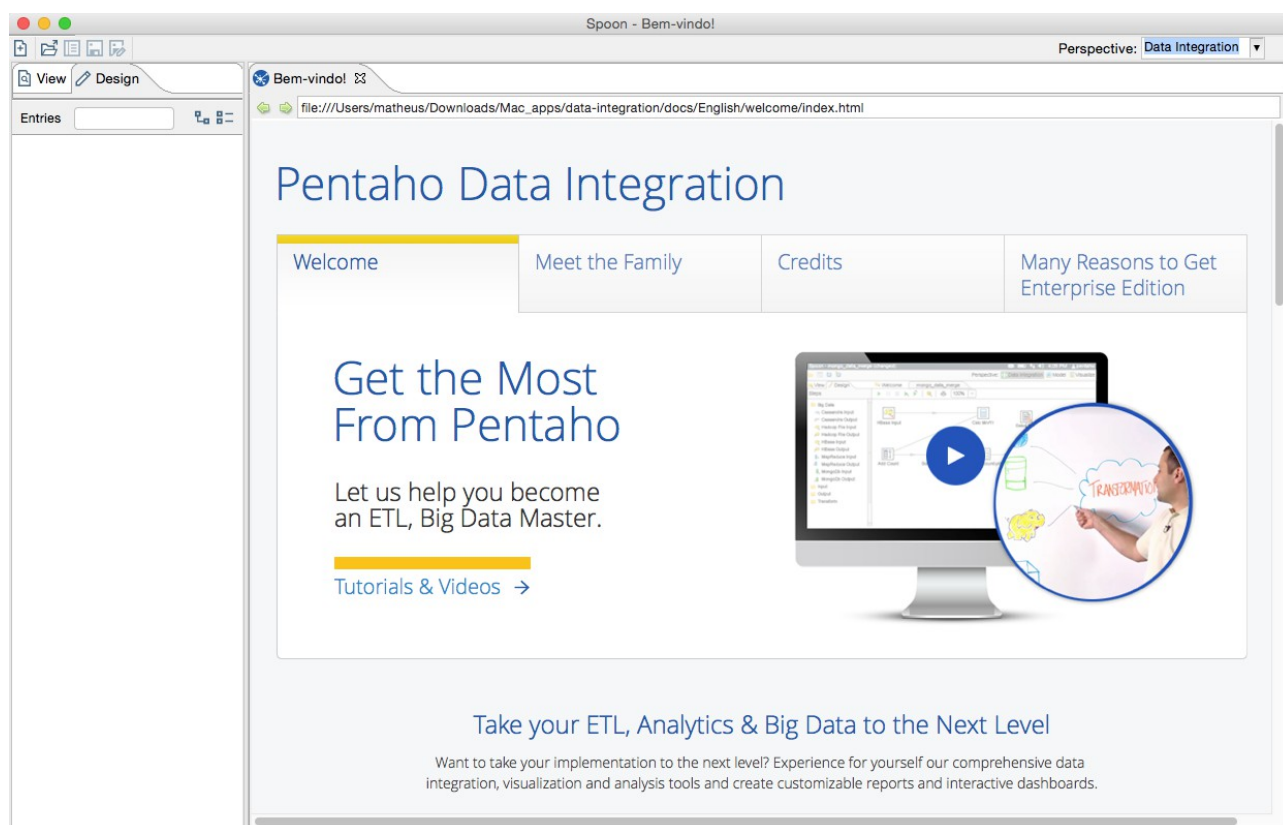


Figura 15: Tela inicial do Pentaho

As demais figuras mostram o Pentaho em execução e acessando o banco de dados PostgreSQL utilizado neste trabalho.

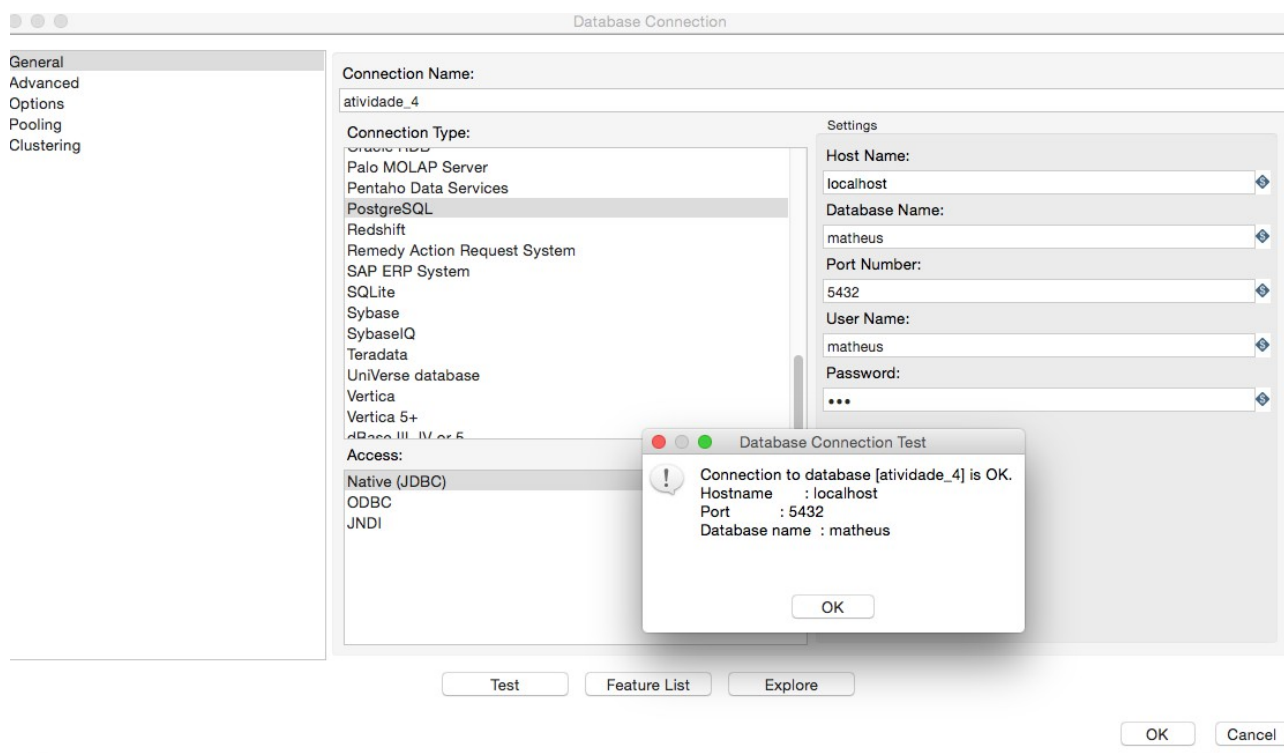


Figura 16: Conexão ao banco de dados

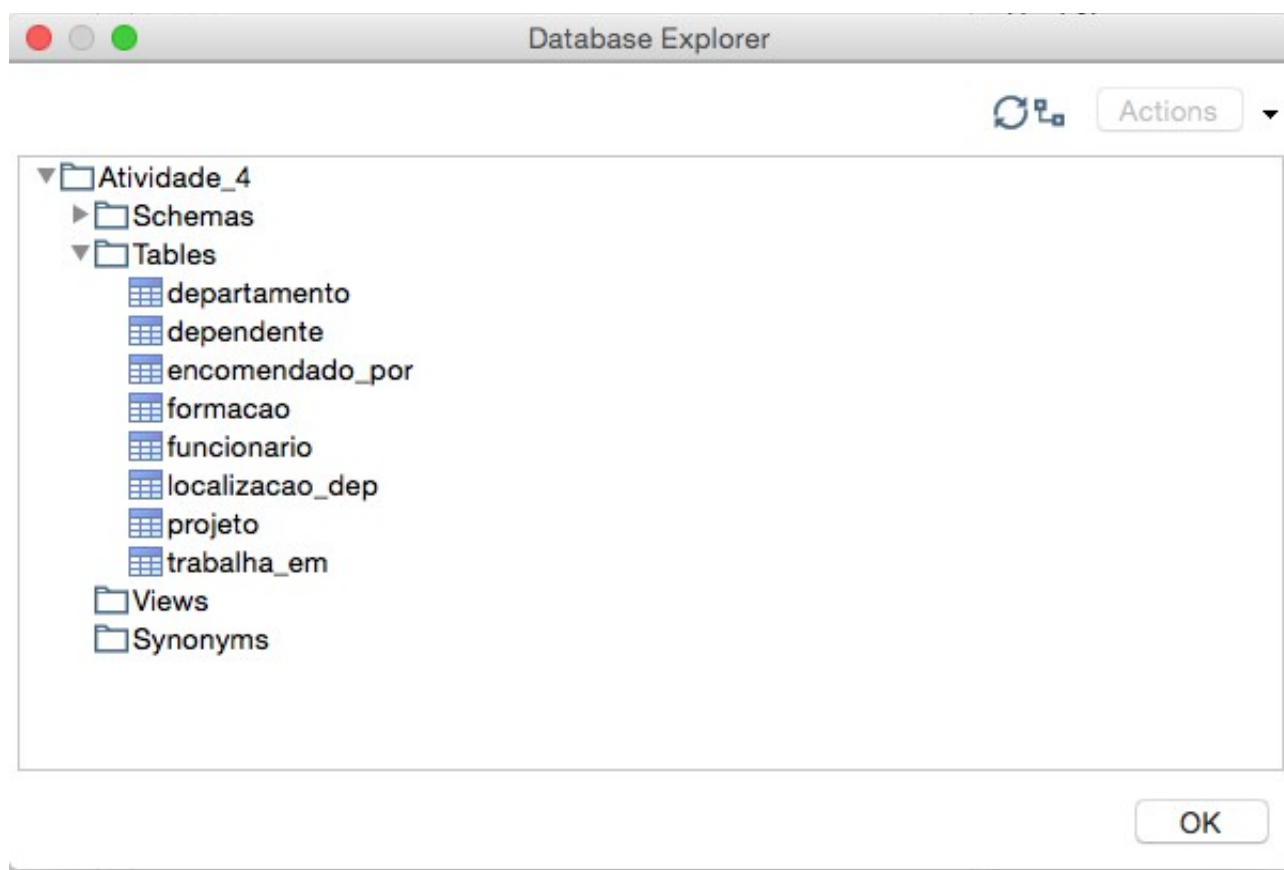


Figura 17: Acessando conteúdo do banco