Projeto BD 2024/2025
University Management System

Base de Dados — LEI 2024/2025

Nome	Nº Estudante	Contacto
Vasco Guilherme da Silva Alves	2022228207	960399272
Lucas Pinto Oliveira	2023219472	969244925
João Tomás Gomes Forte Neto	2023234004	967476669

Conteúdo

1 Abstract

O objetivo deste projeto é desenvolver competências que nos permitam realizar outros projetos envolvendo bases de dados. Para alcançar esse objetivo, pretendemos focar nas seguintes competências:

- Melhorar nossas habilidades de organização.
- Desenvolver modelos de dados eficientes e bem estruturados.
- Aprimorar nossa capacidade de criar modelos que atendam às necessidades dos projetos.
- Adquirir conhecimentos para instalar e configurar sistemas de gestão de bases de dados.

2 Diagrama

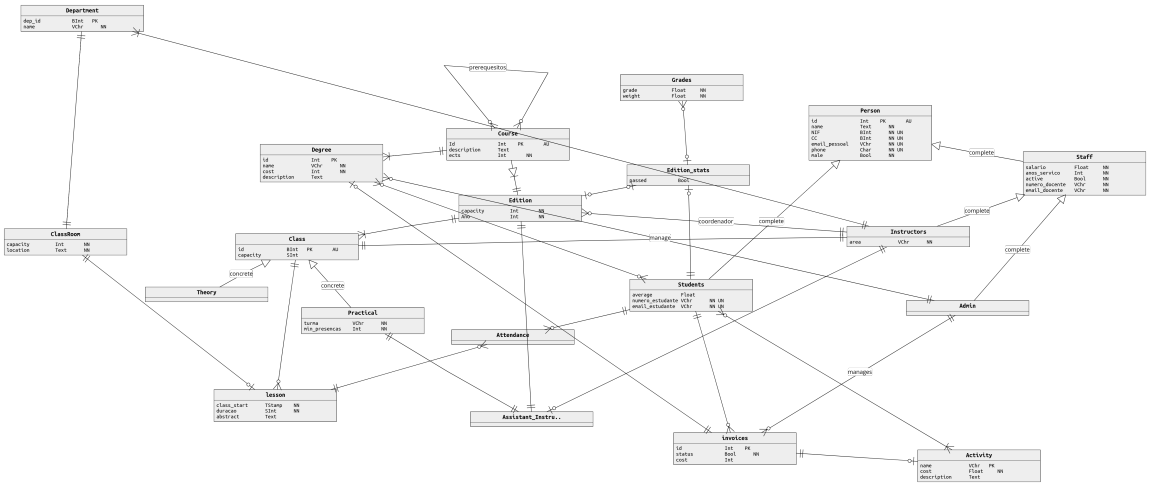


Figura 1: Diagrama de Relacionamento de Entidades

3 Transactions

Após uma discussão com o grupo todo achamos pertinente implementar o uso de transações nas seguintes operações:

- Inscrição de um aluno num curso (degree) por causa do débito que deve ser associado ao estudante;
- Inscrição de um aluno numa cadeira (edition) por causa da capacidade máxima da cadeira;
- Inscrição de um aluno numa atividade (activity) por causa do débito que deve ser associado ao estudante;

4 Potenciais Conflitos

Consideramos que podem ocorrer conflitos (principalmente de concorrência) nos seguintes casos:

- Na criação de uma nova turma (class) por causa de sobreposição de aulas com outras turmas na mesma sala (classroom), ou mesmo capacidade da turma superior à sala;
- Ao inscrever um aluno numa turma, pode ocorrer sobreposição nas aulas das turmas que ele está inscrito;
- A inscrição de um aluno numa cadeira (edition) por causa da capacidade máxima da cadeira;

4.1 Soluções

Para resolver o primeiro conflito devemos, ao criar uma turma, verificar antes se a sala de aulas já está ocupada; caso esteja, não poderemos criar uma turma com aquelas aulas. A mesma lógica se aplica à inscrição de um aluno numa turma; porém, esta última deve ser executada com o uso de transações para garantir que uma vaga não seja ocupada por um aluno que nem irá participar nessa turma. O último possível conflito já é resolvido com as transações. Caso o aluno se tente inscrever numa cadeira e a cadeira já esteja cheia, não será possível se inscrever. Como esta operação é executada com transações, não será ocupada uma vaga numa cadeira em vão.

5 Plano

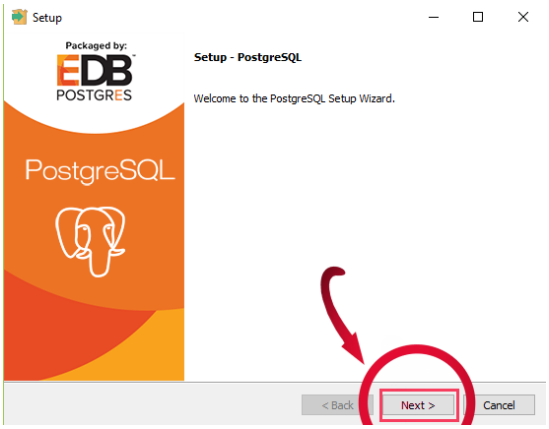
1. **Quick Checkpoint 1**
    - Instalação do DBMS: Cada membro da equipa fará a instalação nos seus PCs individuais.
    - Manual de Instalação: Será criado pelo Lucas.
    - Implementação da REST API: Será realizada pelo Vasco.
2. **Quick Checkpoint 2**
    - Implementação do processo de autenticação: Será feito pelo João.
    - Criação do Manual do Utilizador: Será realizado pelo Vasco.
3. **Final Delivery**
    - Modelo de relação: Será criado pelo Lucas.
    - Revisão Final e Demonstração: Será realizada em conjunto por todo o grupo.

6 Manual de Instalação

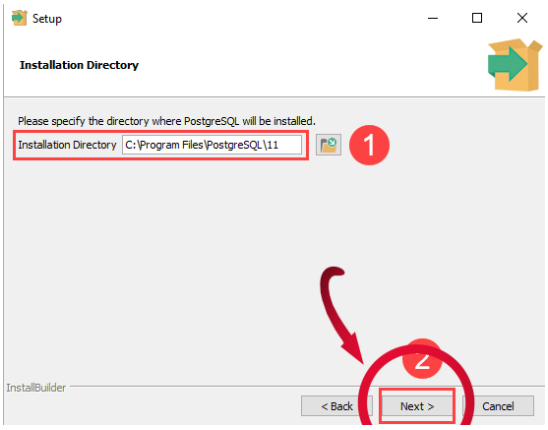
6.1 Instalar PostreSQL

6.1.1 Windows

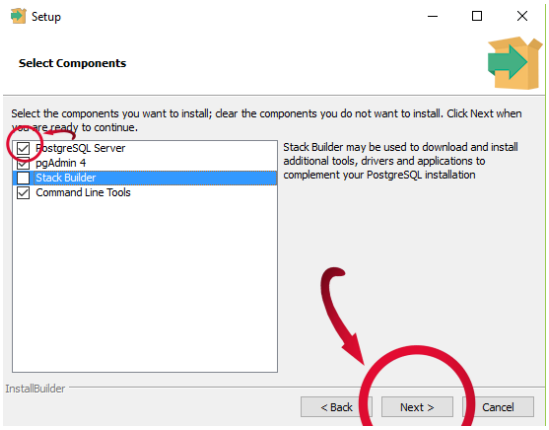
Faz download do instalador: www.enterprisedb.com/downloads/postgres-postgresql-downloads



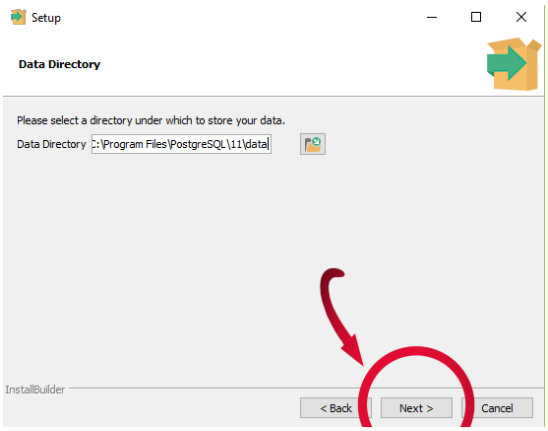
1. Carregar no butão next



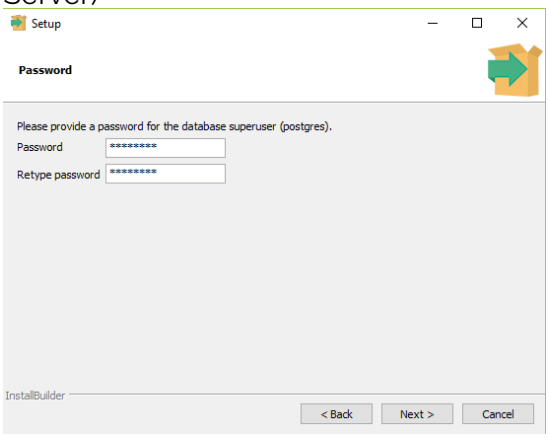
2. Escolher uma pasta (default path é Program Files), e carregar no butão next



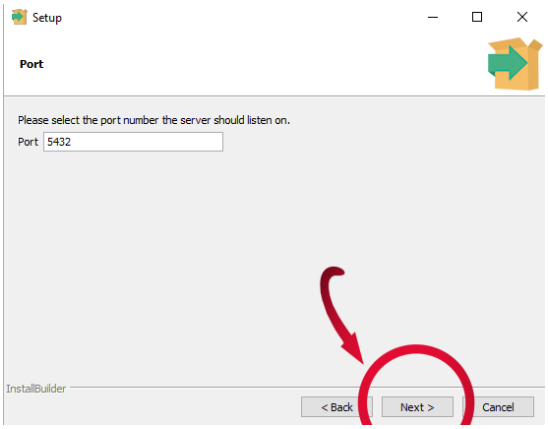
3. Escolher os componentes para instalar (o unico componente necessario para utilizar a base de dados é PostgreSQL Server)



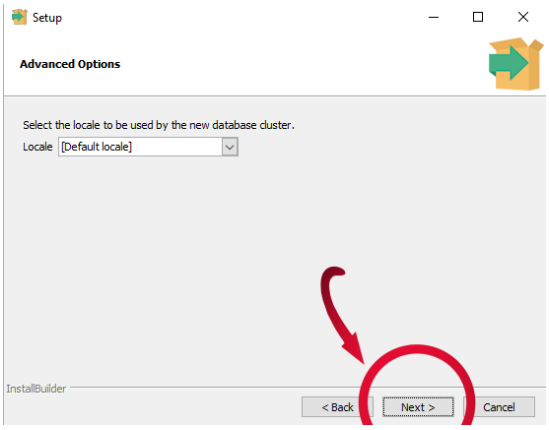
4. Escolher a pasta para o lugar onde a informação vai ser instalada e carregar next



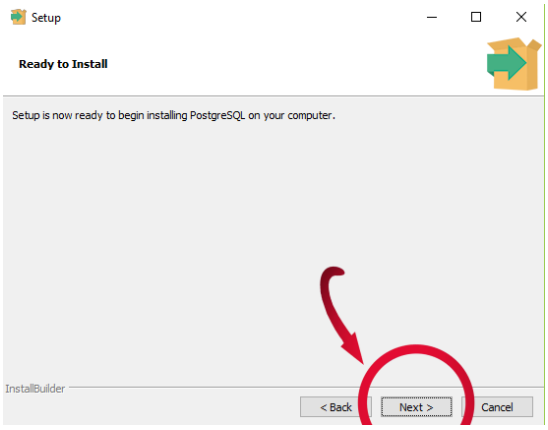
5. Escolher a password da database superuser(Importante não esquecer vai ser utilizado mais tarde) e carregar em next



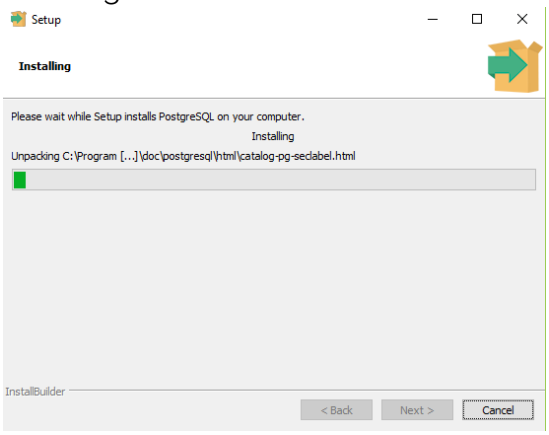
6. Carregar em Next



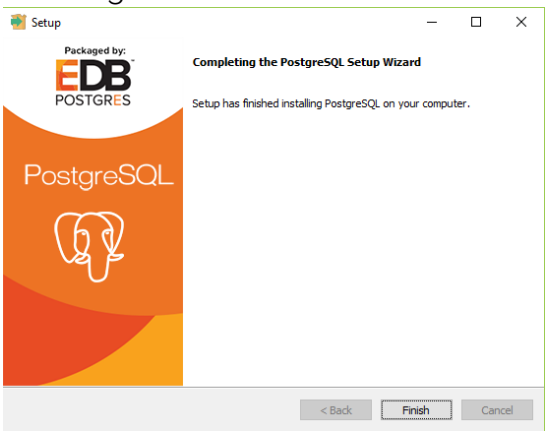
7. Carregar em Next



8. Carregar em Next



9. Esperar para instalar



10. Carreger em Finish

6.1.2 Linux

Debian-based:

```
sudo apt install postgresq
```

Arch-based:

```
sudo pacman -S postgresql
sudo -u postgres initdb -D /var/lib/postgres/data
sudo systemctl start postgresql
```

6.2 Python API e Importar o esqueleto da base de dados

```
git clone https://github.com/lvcxss/UniversityManagementSystem
cd UniversityManagementSystem/API
pip install -r requirements.txt

cd ..
createdb -U postgres -h localhost -p 5432 -template=template0 projeto
psql -h localhost -p 5432 -U "postgres-d "projeto-f backup.sql
```

Isto irá importar as tabelas, os triggers, as functions, as sequences e as views necessárias.

7 User Manual

7.1 Overview dos Endpoints

PUT /dbproj/user
Faz o login do utilizador, seja aluno, administrador ou docente, devolve um token correspondente.

```
{ "email": "behelita23@uc.pt", "password": "123"}
```

POST /dbproj/register/student
Regista um estudante novo, requer token de administrador.

```
{ "name":"slk", "email": "sk@skbu.com", "phone": "+999 133 156 563",
"cc": 122111, "nif": 218128, "password": "123", "gender": "M",
"email_institucional": "cmi@uc.pt", "numero_estudante": "uc232442",
"average": 17 }
```

POST /dbproj/register/staff
Regista um administrador novo, requer token de administrador.

```
{ "name":"behelitta", "email": "behelita@gmail.com", "phone": "+351 123
666 123", "cc": 1234566, "nif": 98724198421, "password": "123", "gender":
"M", "email_institucional": "behelita23@uc.pt", "numero_docente":
"uc200032", "salario": 7128, "anos_servico": 4, "active": true }
```

POST /dbproj/register/instructor
Regista um docente novo, requer token de administrador.

```
{ "name":"cleverson", "email": "2@skbi.com", "phone": "+351 111 223
123", "cc": 9251, "nif": 522, "password": "123", "gender": "M",
"email_institucional": "cuh@uc.pt", "numero_docente": "uc1111", "salario":
9994, "anos_servico": 4, "active": true, "area": "comp sci"}
```

POST /dbproj/enroll_degree/<degree_id>
Permite registar um utilizador num curso.

```
{ "student_id":9 }
```

POST /dbproj/enroll_activity/<activity_id>
Permite ao utilizador que fez login registar numa atividade.

POST /dbproj/enroll_course_edition/<course_edition_id>
Permite ao utilizador que fez login registar em turmas.

```
{ "classes": [1, 2] }
```

POST /dbproj/submit_grades/<course_edition_id>
Permite ao regente da edição da cadeira submeter notas para os alunos.

```
{ "period_id": 1, "grades": [[23, 1]] }
```

GET /dbproj/student_details/<student_id>
Devolve informação sobre as cadeiras em que o aluno está inscrito.

GET /dbproj/degree_details/<degree_id>
Devolve informação sobre o curso e as respectivas edições.

GET /dbproj/top3
Devolve os 3 melhores alunos do ano lectivo atual organizados por curso e atividades.

GET /dbproj/top_by_district/
Devolve os melhores alunos de cada districto.

GET /dbproj/report
Devolve as edições com o maior número de aprovações por mês.

DELETE /dbproj/delete_details/<student_id>
Apaga o utilizador.