

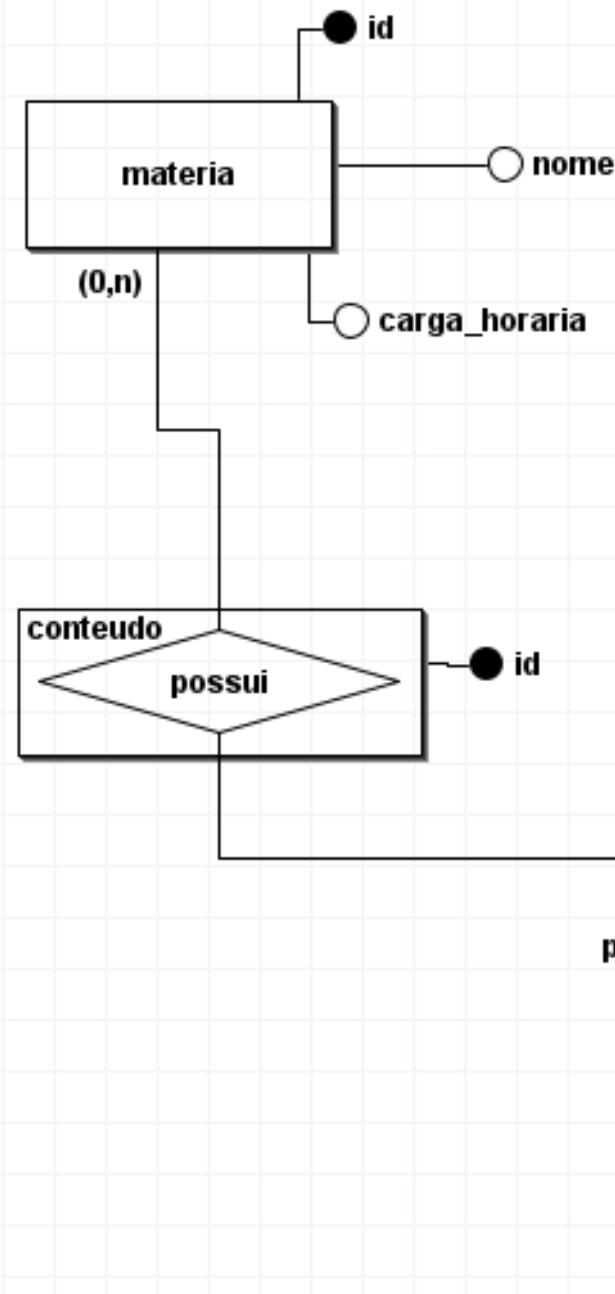
# ETAPA MODEL INICIALIZADA

Professor Rômulo Cesar Silvestre

Se você encontrar na história N-N  
Cria-se uma entidade associativa  
Quebrando duas 1-N

Se você encontrar na história 1-1  
Não se preocupe, o brModelo irá  
unificar a tabela.

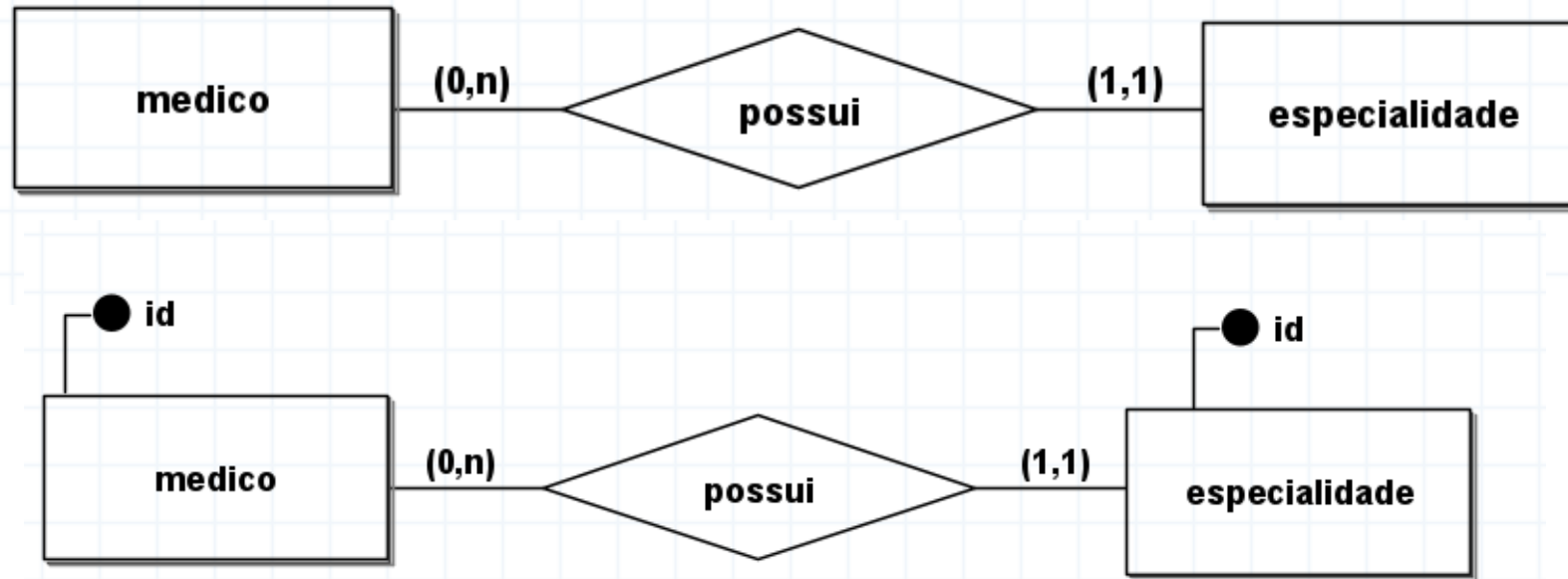
Se você encontrar na história 1-N  
É o mais popular, entidade fraca  
recebe chave primária da forte.



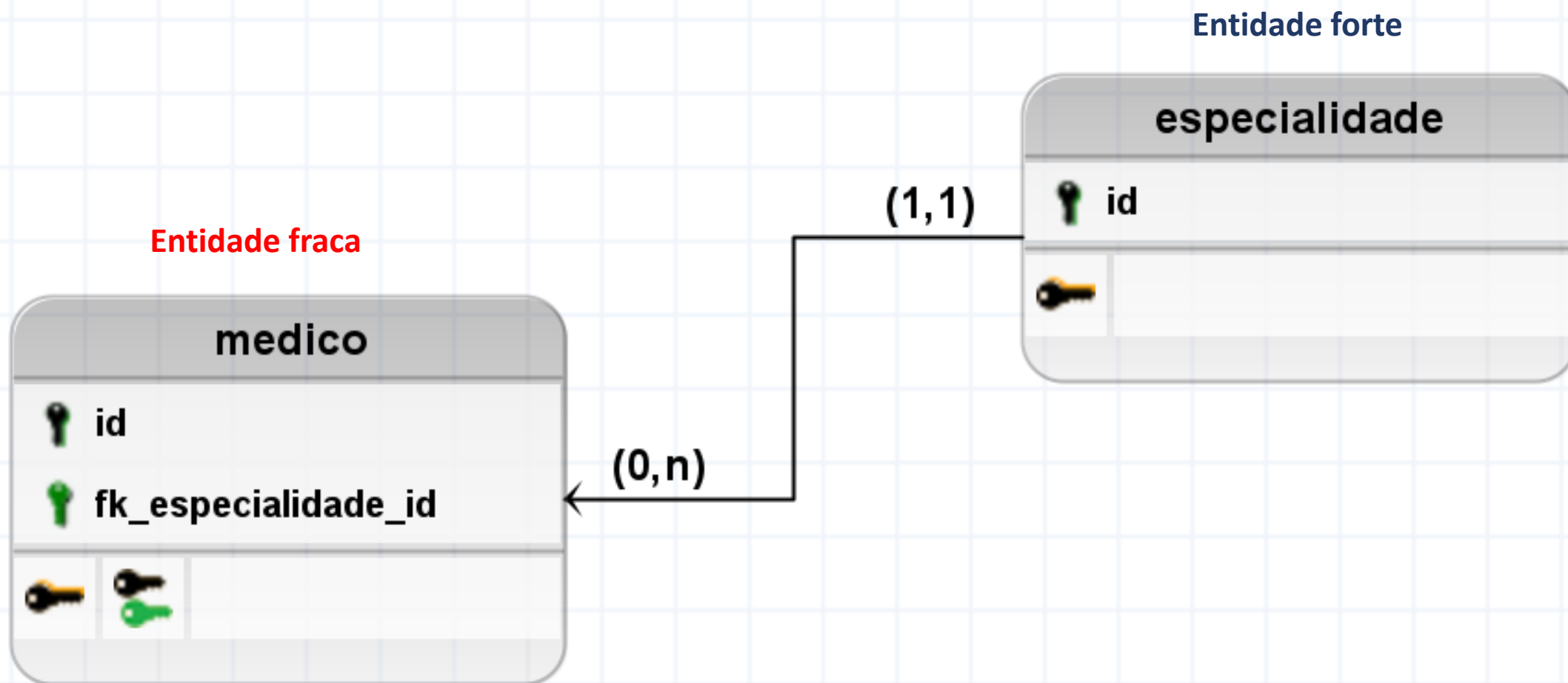
## User History = Requisito Funcional

Uma matéria pode estar associada vários módulos  
E um módulo deve ter um ou vários módulos.

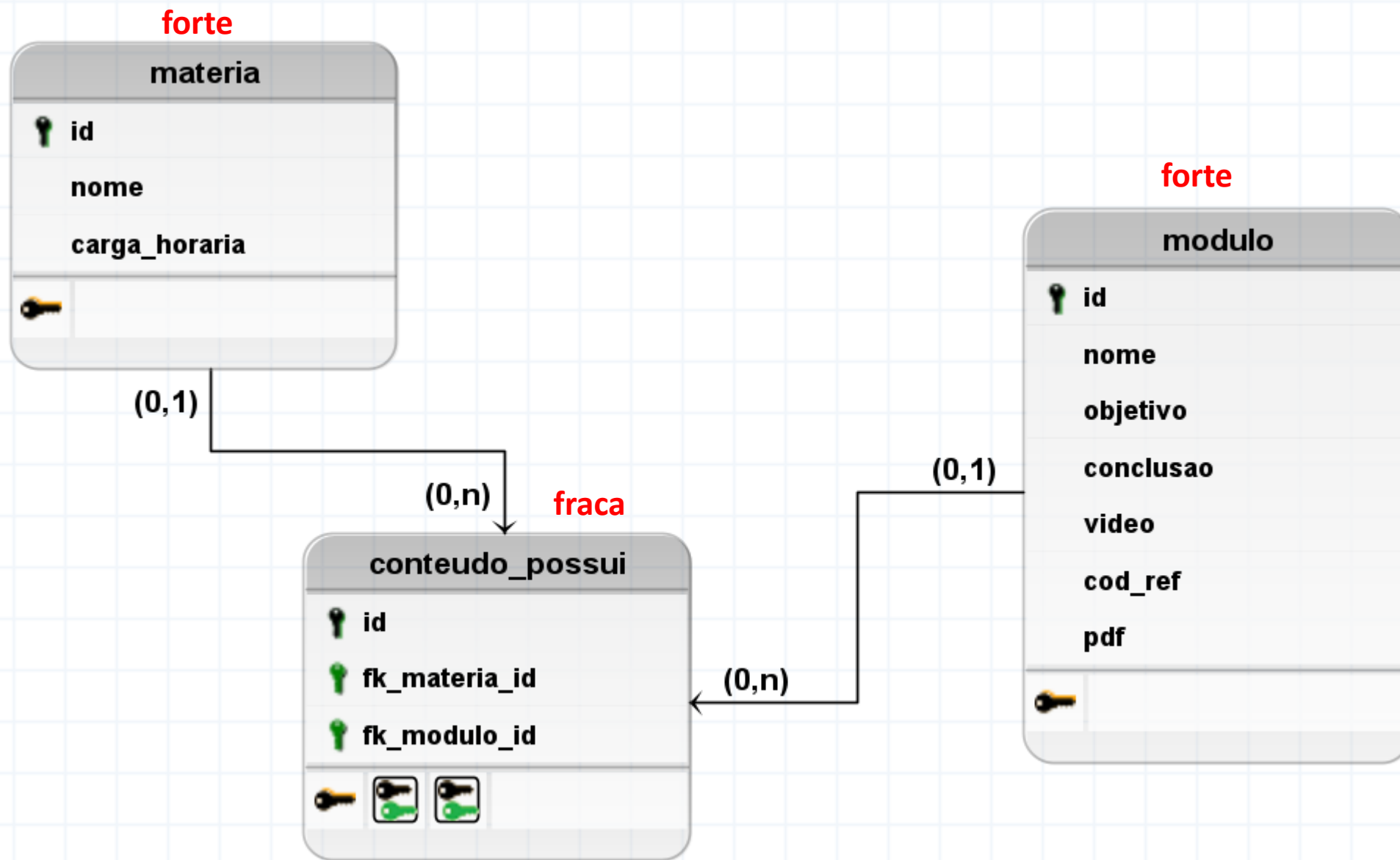
Para cada relacionamento crie uma história.

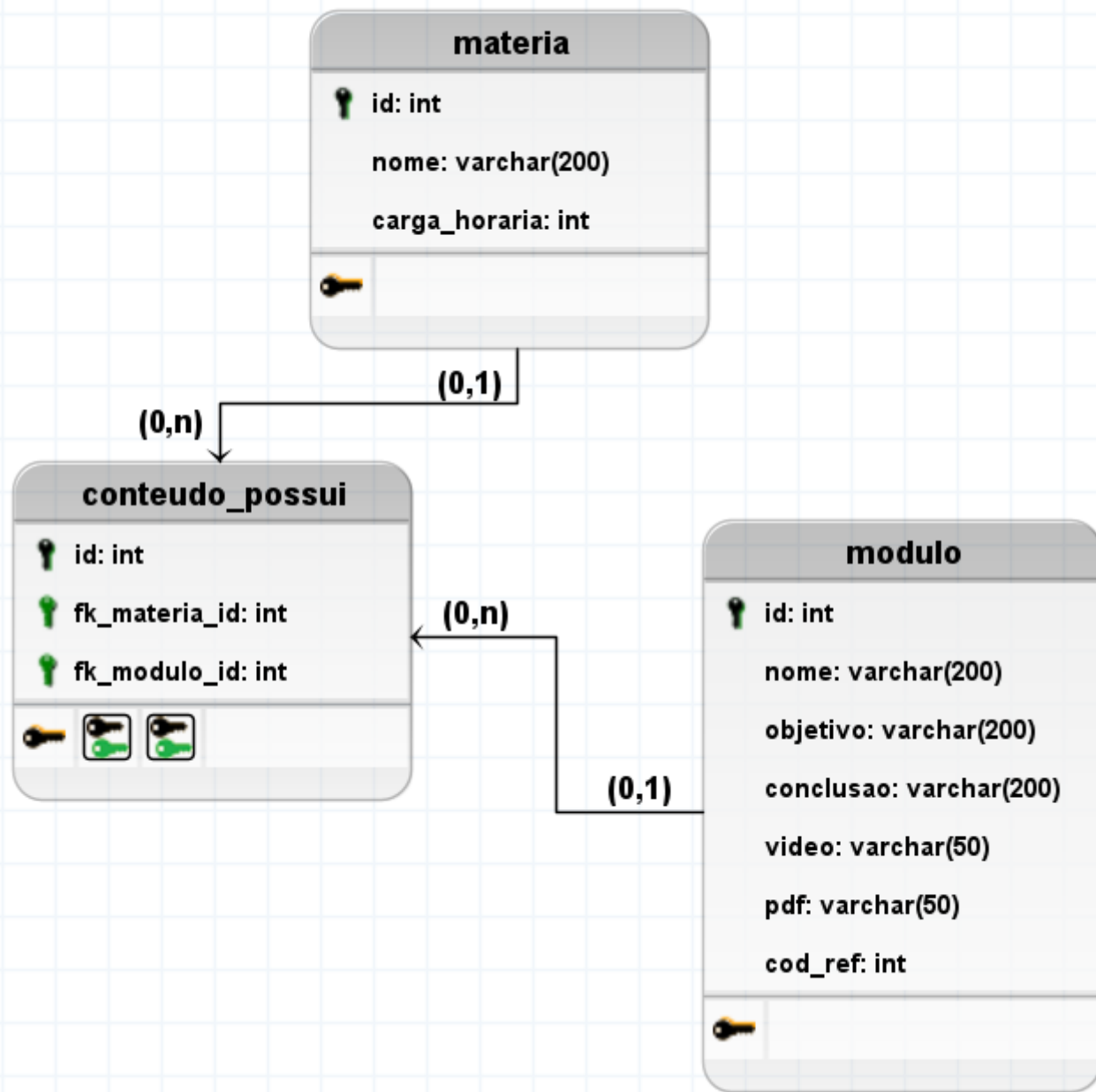


Se você encontrar na história 1-N  
É o mais popular, entidade fraca  
recebe chave primária da forte.



A entidade fraca é que recebe a chave primária da outra entidade.








# Programador BACKEND

```
1  from app import db
2  class Materia(db.Model):
3      __tablename__ = "materia"
4      id = db.Column(db.Integer, primary_key=True, autoincrement=True)
5      numero = db.Column(db.Integer)
6      nome = db.Column(db.String(200))
7      carga_horaria = db.Column(db.Integer)
8
```

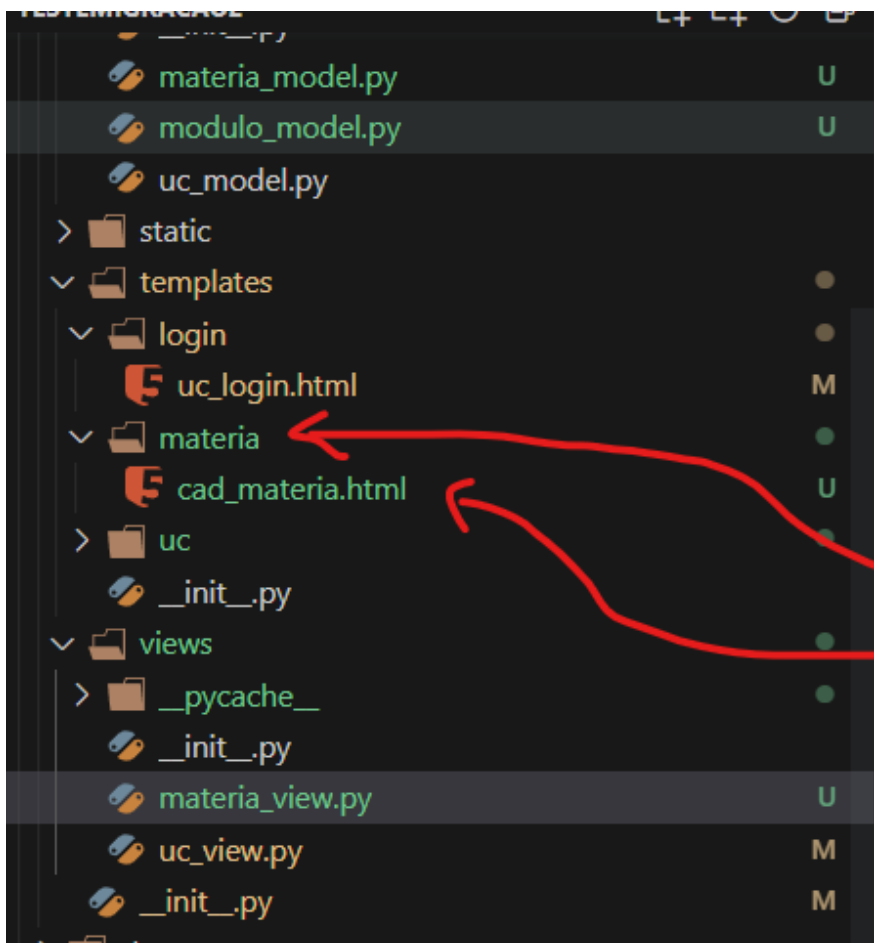
- init
- migrate
- upgrade

 <b>alembic_version</b> ▼
💡 version_num VARCHAR(32)
Indexes ▶

 <b>materia</b> ▼
💡 id INT
◊ numero INT
◊ nome VARCHAR(200)
◊ carga_horaria INT
Indexes ▶

 <b>modulo</b> ▼
💡 id INT
◊ nome VARCHAR(200)
◊ objetivo VARCHAR(200)
◊ conclusao VARCHAR(200)
◊ video VARCHAR(50)
◊ pdf VARCHAR(50)
◊ cod_ref INT
Indexes ▶





```
1  
2 from app import app  
3 from flask import render_template  
4  
5 @app.route('/cadmateria')  
6 def cadastrar_materia():  
7     return render_template("materia/cad_materia.html")  
8
```

```
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Login - ProjectScore</title>
7 <!--comentário html-->
8 <!--insira o bootstrap aqui!-->
9 <link rel="stylesheet" href="{{url_for('static',filename='css/bootstrap.css')}}">
10 </head>
11 <body>
12     <div class="mb-3">
13         <label for="nome" class="form-label">nome</label>
14         <input type="text" class="form-control" id="nome" placeholder="">
15     </div>
16     <div class="mb-3">
17         <label for="carga_horaria" class="form-label">carga horária</label>
18         <input type="text" class="form-control" id="carga_horaria" rows="3">
19         <button>cadastrar</button>
20     </div>
21
22
23 </body>
24 </html>
```

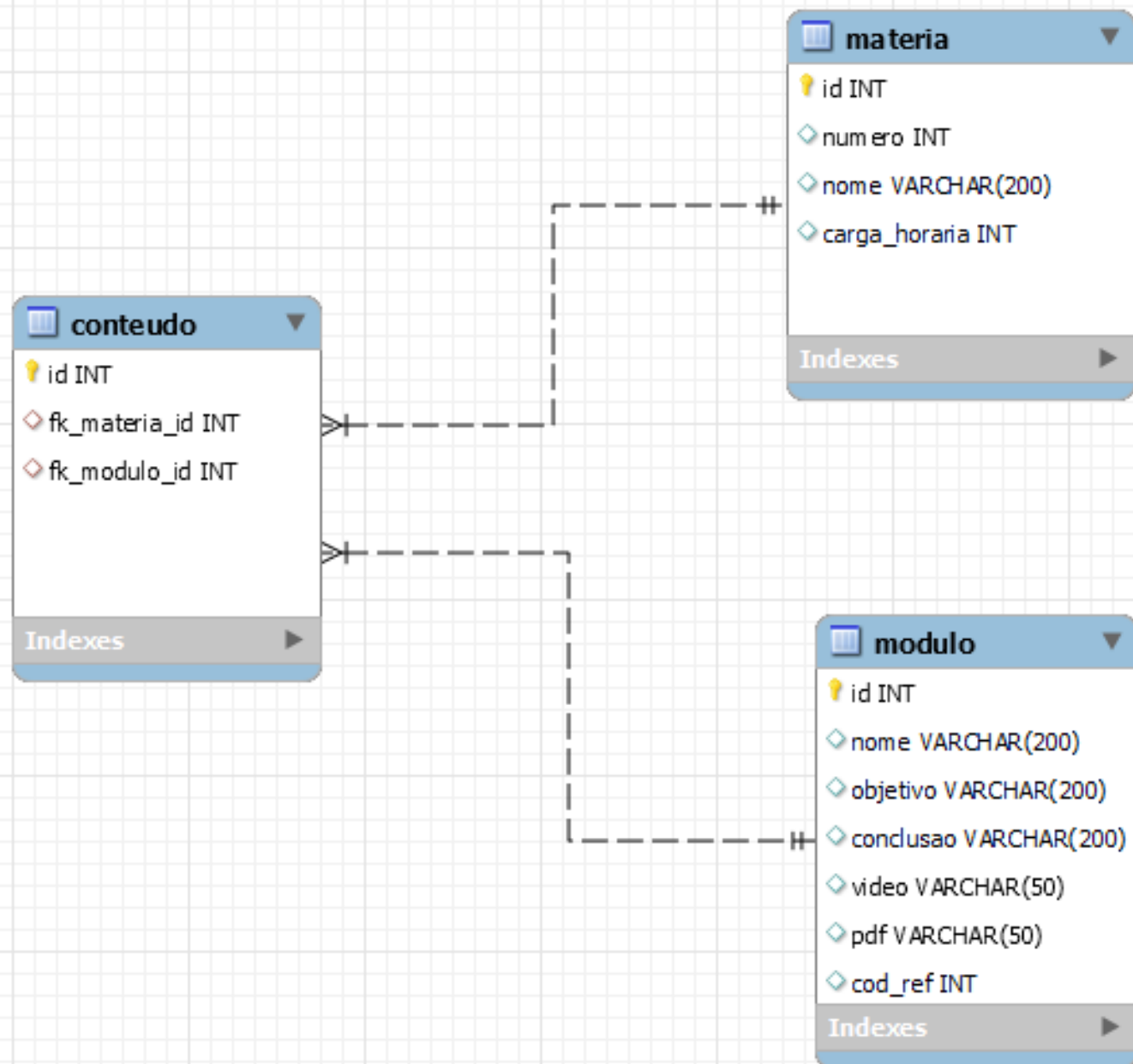


127.0.0.1:5000/cadmateria

nome

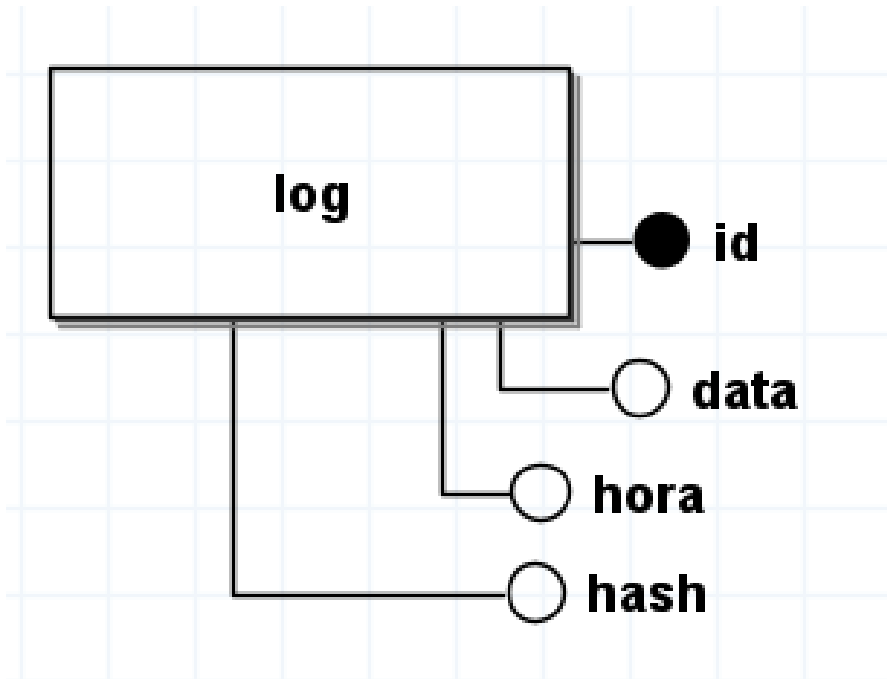
carga horária

cadastrar



```
1  from app import db
2
3  # Define ConteudoPossui class
4  class ConteudoPossui(db.Model):
5      __tablename__ = 'conteudo'
6
7      id = db.Column(db.Integer, primary_key=True)
8      fk_materia_id = db.Column(db.Integer, db.ForeignKey('materia.id'))
9      fk_modulo_id = db.Column(db.Integer, db.ForeignKey('modulo.id'))
```

# Aula PI



**Tabela de log** é utilizada pelos programadores para Registrar ações dos usuários.  
É como se fosse um histórico.

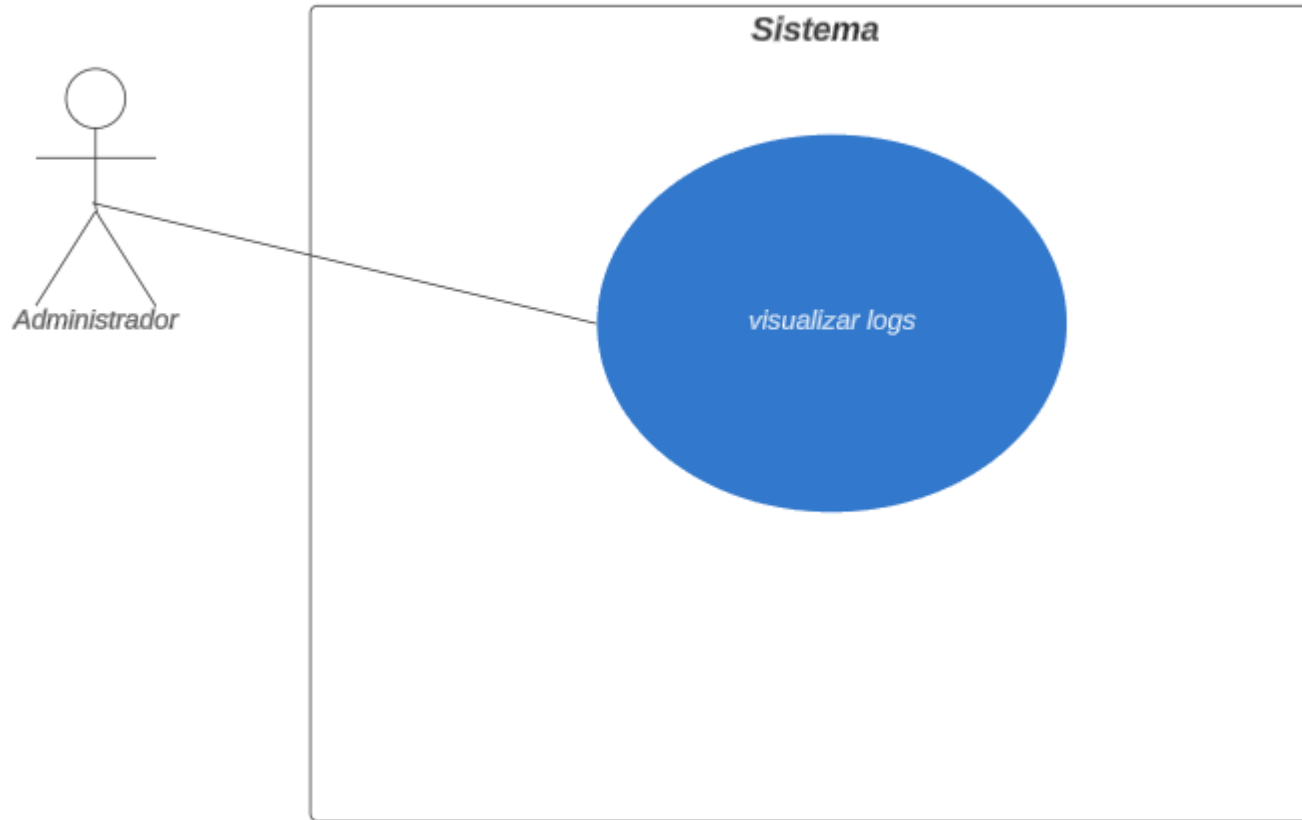
A administrador do sistema, ao logar pode clicar  
Em visualizar histórico e usuários.

Ele vai visualizar todos os logs do sistema em uma tabela html c  
com os dados dos logs.

Ele pode filtrar dos dados por data.

## UC 01 - Visualizar Logs

strarclic | May 10, 2024



**O Caso de Uso Visualizar Logs** é utilizado pelos programadores para Registrar ações dos usuários. É como se fosse um histórico.

O administrador do sistema, ao logar pode clicar em visualizar logs.

Ele vai visualizar todos os logs do sistema em uma tabela html com os dados dos logs.

Ele pode filtrar os dados por data.

# Primeira ação: baixar o projeto base

MINGW64:/c/Users/romulo.leite/Desktop

```
romulo.leite@L02005021877 MINGW64 ~
```

```
$ cd Desktop
```

```
romulo.leite@L02005021877 MINGW64 ~/Desktop
```

```
$ |
```



MINGW64:/c/Users/romulo.leite/Desktop

romulo.leite@L02005021877 MINGW64 ~

\$ cd Desktop

romulo.leite@L02005021877 MINGW64 ~/Desktop

\$ git clone <https://github.com/romulosilvestre/projetobase.git>

```
romulo.leite@L02005021877 MINGW64 ~/Desktop
$ git clone https://github.com/romulosilvestre/projetobase.git
Cloning into 'projetobase'...
remote: Enumerating objects: 3649, done.
remote: Counting objects: 100% (3649/3649), done.
remote: Compressing objects: 100% (2897/2897), done.
remote: Total 3649 (delta 700), reused 3649 (delta 700), pack-reused 0
Receiving objects: 100% (3649/3649), 30.50 MiB | 5.44 MiB/s, done.
Resolving deltas: 100% (700/700), done.
```

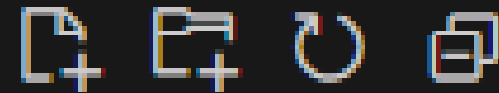
```
romulo.leite@L02005021877 MINGW64 ~/Desktop/starclinic (master)
```


```
$
```

```
|
```

```
romulo.leite@L02005021877 MINGW64 ~/Desktop/starclinic (master)  
$ code .|
```


✓ **STARCLINIC**



>  \_\_pycache\_\_

>  app

>  docs

>  migrations

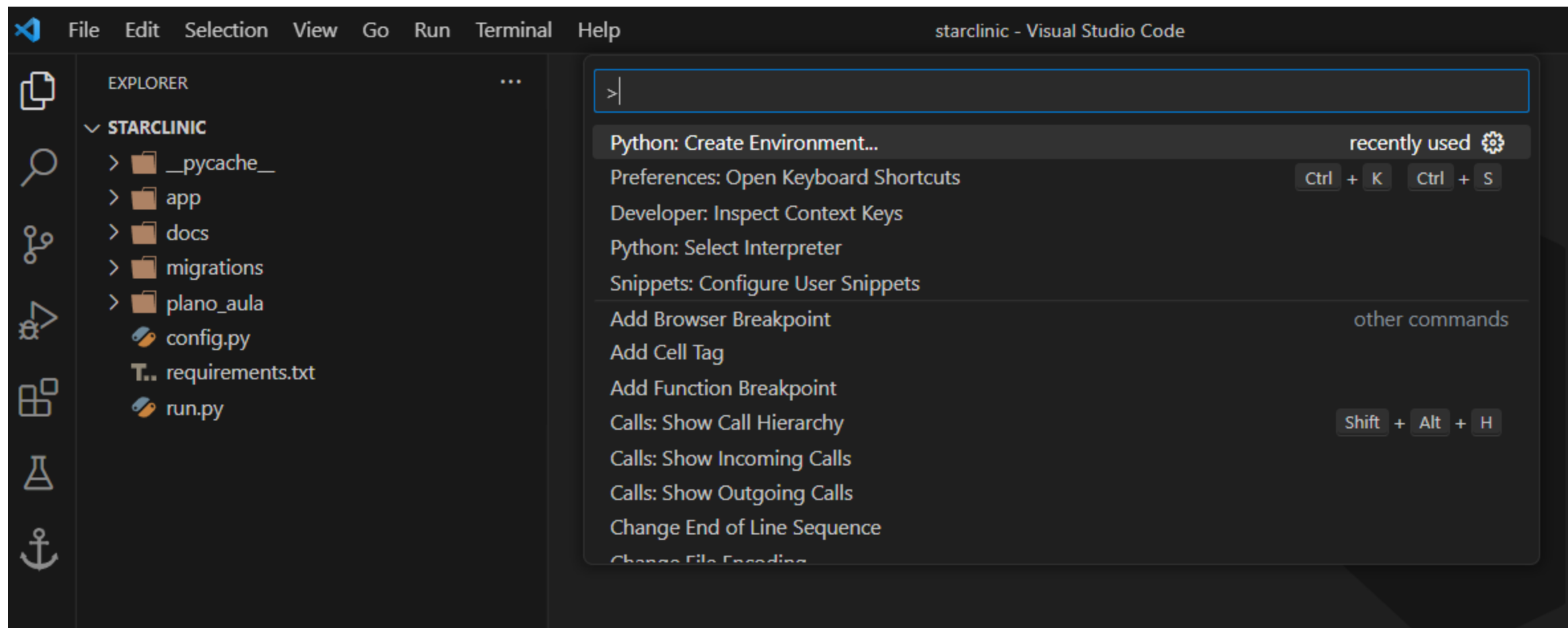
>  plano\_aula

 config.py

T.. requirements.txt

 run.py

## CTRL+SHIFT+P



>|

Python: Create Environment...

Preferences: Open Keyboard Shortcuts

Select an environment type

**Venv** Creates a `.venv` virtual environment in the current workspace

**Conda** Creates a `.conda` Conda environment in the current workspace





Select a Python installation to create the virtual environment

+ Enter interpreter path...

Python 3.12.3 64-bit (microsoft store) ~\AppData\Local\Microsoft\WindowsApps\python3.12.... Global

Python 3.9.13 64-bit C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39\_64\python.exe



Select dependencies to install

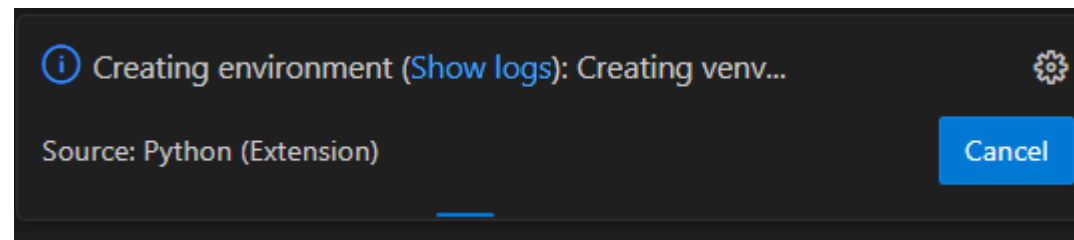
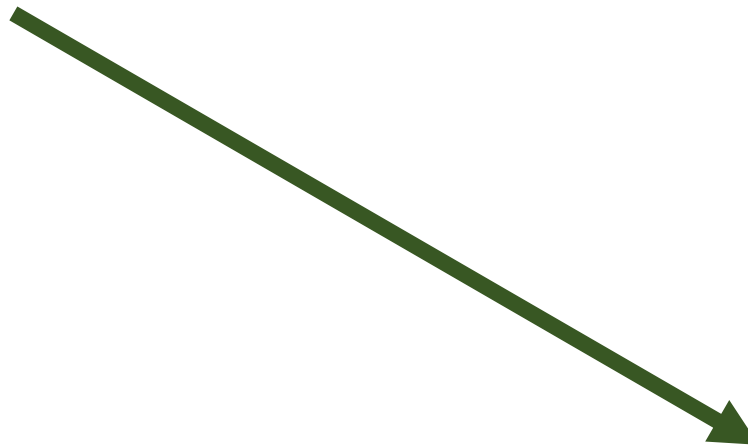
0 Selected

OK




requirements.txt

Fique atento na mensagem abaixo



✓ **STARCLINIC**


>  \_\_pycache\_\_

>  .venv

>  app

>  docs

>  migrations

>  plano\_aula

 config.py

T.. requirements.txt

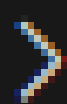
 run.py

---

Microsoft Windows [versão 10.0.22621.2283]  
(c) Microsoft Corporation. Todos os direitos reservados.  
(.venv) C:\Users\romulo.leite\Desktop\starclinic>



models



\_\_pycache\_\_



\_\_init\_\_.py



area\_model.py



competencia\_model.py




curso\_model.py



uc\_model.py

## log

 **id: INT**

**hora: TIME**

**data: DATE**

**hash: VARCHAR(50)**



# Usando inteligência artificial – Generativa (*algo novo*)

- **GPT é uma categoria de inteligência artificial:**
  - Generative Pre-trained Transformer (Transformador Pré-treinado Generativo, em tradução livre), um tipo de modelo de linguagem grande (Large Language Model, LLM, na sigla em inglês).
- **ChatGPT** – é um produto (OpenAI)
- **Copilot Notebook** – é um produto (Microsoft)
- **Gemini** – é um produto (Google)



# Saber fazer pergunta!

Gere para mim um modelo no python - SQLAlchemy (ORM) com os seguintes campos: ID (primary key) , hora do tipo tempo, data do tipo data, hash para armazenar o tipo mysql uuid!



ChatGPT

Entendi, você quer que a classe seja chamada de `LogUsuario`. Aqui está o código atualizado:

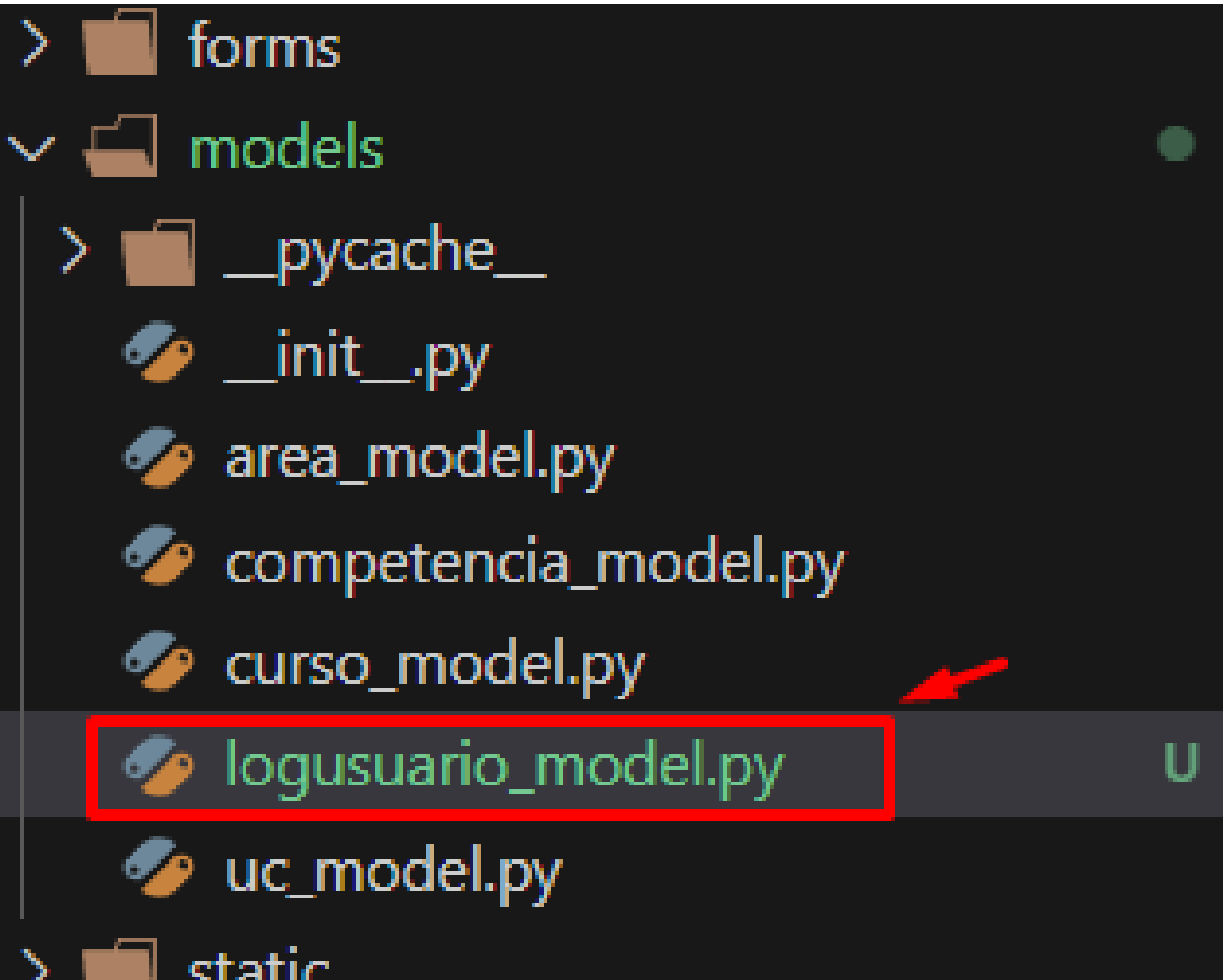
python

 Copy code

```
from app import db

class LogUsuario(db.Model):
    __tablename__ = "log_usuario"
    id = db.Column(db.Integer, primary_key=True, autoincrement=True)
    hora = db.Column(db.Time)
    data = db.Column(db.Date)
    hash = db.Column(db.String(36), unique=True)
```

Agora a classe se chama `LogUsuario` e possui os campos `hora`, `data` e `hash`, conforme solicitado.

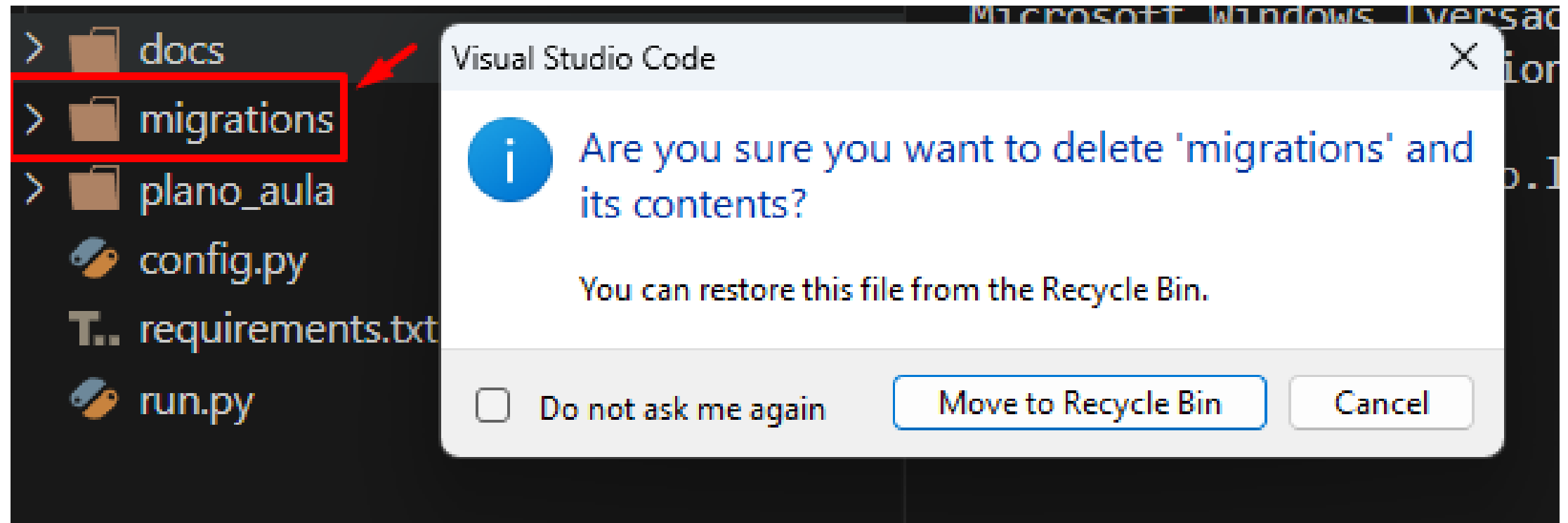


```
1 from app import db
2
3 class LogUsuario(db.Model):
4     __tablename__ = "log_usuario"
5     id = db.Column(db.Integer, primary_key=True, autoincrement=True)
6     hora = db.Column(db.Time)
7     data = db.Column(db.Date)
8     hash = db.Column(db.String(36), unique=True)
9
```

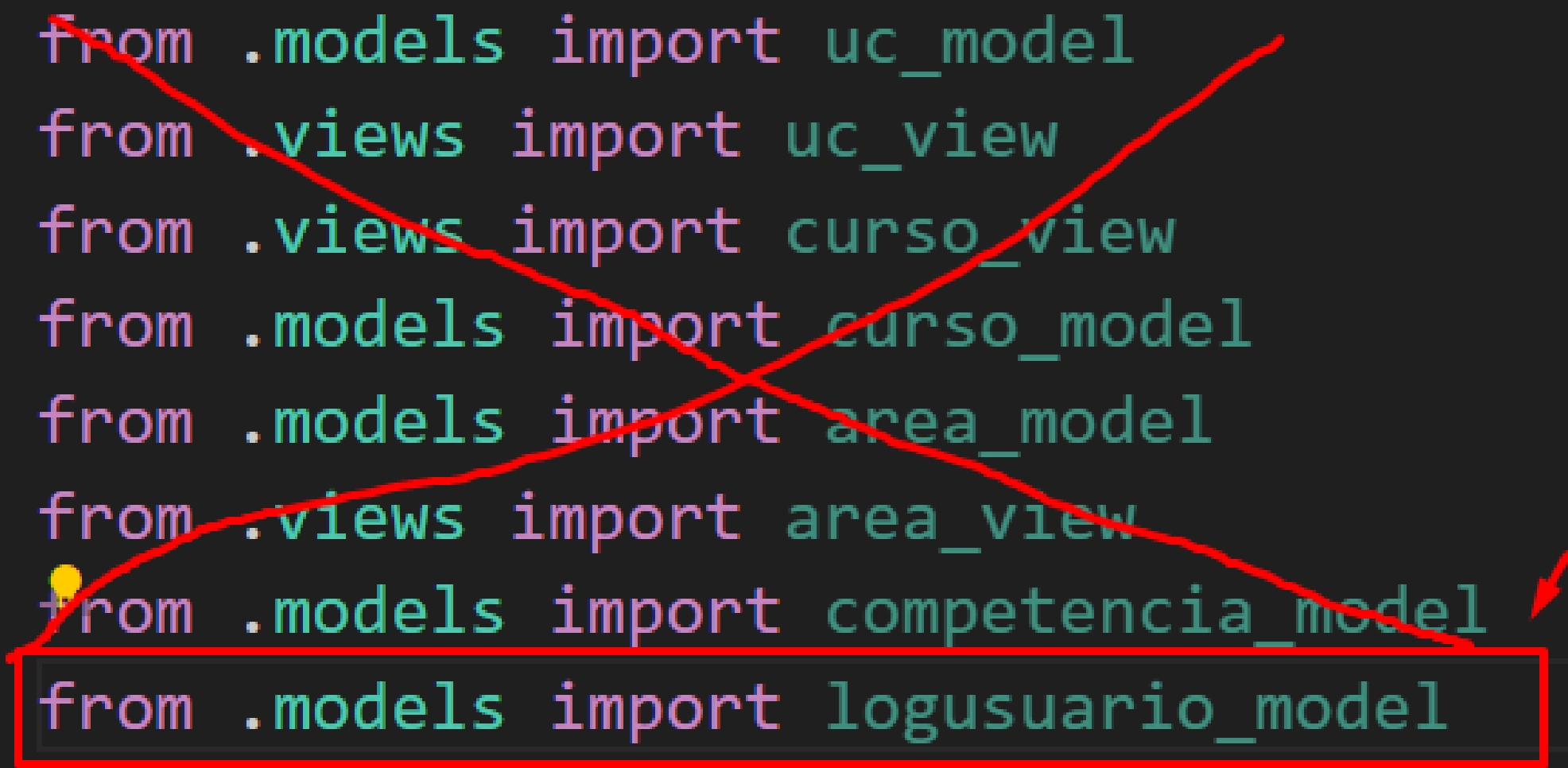
Vitóriaaaaaa!

# Migrations

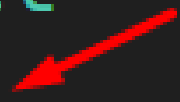
- **Migrate**



```
20
21 #determinar o que vai ter no projeto
22 from .models import uc_model
23 from .views import uc_view
24 from .views import curso_view
25 from .models import curso_model
26 from .models import area_model
27 from .views import area_view
28 from .models import competencia_model
29 | from .models import logusuario_model
30
```



```
3  #importando o SQLAlchemy
4  from flask_sqlalchemy import SQLAlchemy
5  from flask_migrate import Migrate, upgrade
6  from flask_wtf import CSRFProtect
7  from flask_babel import Babel
8  #importando o Celery
```



```
pip install flask_babel
```



```
DROP DATABASE projectscore;  
CREATE DATABASE projectscore;
```

```
(.venv) C:\Users\romulo.leite\Desktop\starclinic>pip list
```

Package	Version
---------	---------

-----	-----
-------	-------

alembic	1.13.1
---------	--------

Babel	2.15.0
-------	--------

blinker	1.7.0
---------	-------

click	8.1.7
-------	-------

colorama	0.4.6
----------	-------

Flask	3.0.3
-------	-------

flask-babel	4.0.0
-------------	-------

Flask-Migrate	4.0.7
---------------	-------

Flask-Script	2.0.6
--------------	-------

Flask-SQLAlchemy	3.1.1
------------------	-------

Flask-WTF	1.2.1
-----------	-------

greenlet	3.0.3
----------	-------

itsdangerous	2.2.0
--------------	-------

Jinja2	3.1.3
--------	-------

Mako	1.3.3
------	-------

MarkupSafe	2.1.5
------------	-------

mysql-connector-python	8.3.0
------------------------	-------

mysqlclient	2.2.4
-------------	-------

pip	24.0
-----	------

pytz	2024.1
------	--------

SQLAlchemy	2.0.29
------------	--------

typing_extensions	4.11.0
-------------------	--------

Werkzeug	3.0.2
----------	-------

WTForms	3.1.2
---------	-------

```
PS C:\Users\romulo.leite\Desktop\starclinic> flask db init
```

```
Creating directory 'C:\\Users\\romulo.leite\\Desktop\\starclinic\\migrations' ... done
```

```
Creating directory 'C:\\Users\\romulo.leite\\Desktop\\starclinic\\migrations\\versions' ... done
```

```
Generating C:\\Users\\romulo.leite\\Desktop\\starclinic\\migrations\\alembic.ini ... done
```


```
Generating C:\\Users\\romulo.leite\\Desktop\\starclinic\\migrations\\env.py ... done
```


```
Generating C:\\Users\\romulo.leite\\Desktop\\starclinic\\migrations\\README ... done
```


```
Generating C:\\Users\\romulo.leite\\Desktop\\starclinic\\migrations\\script.py.mako ... done
```

```
Please edit configuration/connection/logging settings in 'C:\\Users\\romulo.leite\\Desktop\\starclinic\\migrations\\alembic.ini' before proceeding.
```


```
PS C:\\Users\\romulo.leite\\Desktop\\starclinic> █
```


 powershell


 cmd

✓  views


>  \_\_pycache\_\_

 \_\_init\_\_.py

 area\_view.py


 curso\_view.py

 uc\_view.py

 \_\_init\_\_.py

>  docs

>  migrations 

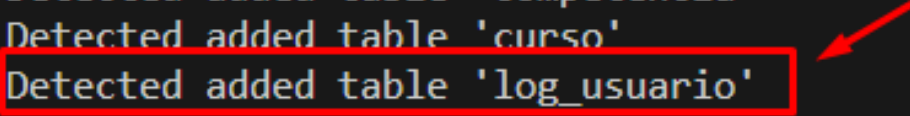
>  plano\_aula

 config.py

```
flask db migrate -m "Initial migration."
```



```
PS C:\Users\romulo.leite\Desktop\starclinic> flask db migrate -m "Initial migration."
INFO [alembic.runtime.migration] Context impl MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
INFO [alembic.autogenerate.compare] Detected added table 'area'
INFO [alembic.autogenerate.compare] Detected added table 'competencia'
INFO [alembic.autogenerate.compare] Detected added table 'curso'
INFO [alembic.autogenerate.compare] Detected added table 'log_usuario'
INFO [alembic.autogenerate.compare] Detected added table 'unidadecompetencia'
Generating C:\Users\romulo.leite\Desktop\starclinic\migrations\versions\6e9088c8f5c2_initial_migration.py ...
done
```



```
> flask db upgrade
```


```
impl MySQLImpl.
```

```
me non-transactional DDL.
```


```
pgrade -> 6e9088c8f5c2, Int
```


▼  projectscore



▼  Tables

▶  alembic\_version

▶  area

▶  competencia

▶  curso

▶  log\_usuario 

▶  unidadcompetencia



unidadecompetencia
id INT
numero INT
nome VARCHAR(200)
carga_horaria INT
competencia_geral VARCHAR(255)
Indexes

area
id INT
nome VARCHAR(200)
Indexes

competencia
id INT
nome VARCHAR(200)
Indexes

curso
id INT
nome VARCHAR(200)
Indexes

log_usuario
id INT
hora TIME
data DATE
hash VARCHAR(36)
Indexes

# ETAPA MODEL FINALIZADA



Happy Birthday!

myloview