

Documentação para uso do banco de dados e API:

Por Arthur Moreira

Ao abrir o drive, faça o download do arquivo .zip “partedoarthur” e extraia os arquivos. Haverá dois arquivos na pasta, sendo um o código do banco de dados “dbcode”, e o outro uma pasta “time-projeto”, uma pasta projeto pycharm. Basta copiar a pasta e colar na pasta PycharmProjects, caminho: **C:\Users\Aluno\PycharmProjects**.

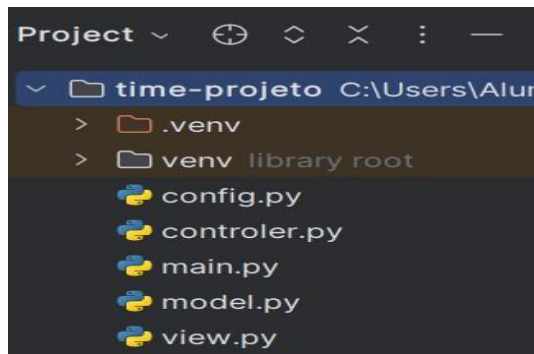
execute o Pycharm e abra o projeto, posteriormente instale as bibliotecas necessárias:

Flask

Flask-Sqlalchemy

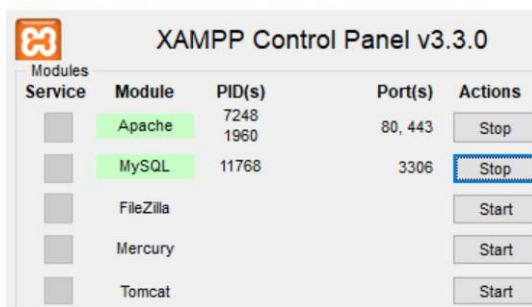
Mysqclient

Visualização:



Inicie o Xampp:

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]



No MySql Workbench:

Inicie a conexão com xampp e crie o banco de dados **com o nome “time”**:

O código para a criação das tabelas e para a inserção dos dados iniciais está a seguir:

```
CREATE TABLE Usuarios (  
user_id INT AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(100),  
senha VARCHAR(254),  
email VARCHAR(100),  
cpf INT  
);
```

```
insert into Usuarios (nome, senha, email, cpf)  
values('arthur', 'admin', 'arthur@aluno.com', 97643539506);
```

```
CREATE TABLE Cursos (  
curso_id INT AUTO_INCREMENT PRIMARY KEY,  
nomeCurso VARCHAR(100),  
descricao VARCHAR(254),  
diasSemanaCurso VARCHAR(100),  
dataInicial DATE,  
dataFinal DATE,  
cargaHoraria INT,  
user_id int,  
FOREIGN KEY(`user_id`) REFERENCES Usuarios(user_id)  
);
```

```
insert into Cursos (nomeCurso, descricao, diasSemanaCurso, dataInicial,dataFinal,  
cargaHoraria, user_id)  
values('Python', 'Um curso muito legal', 'Quinta-feira', '14/05/2024','14/08/2024', 80 , 1);
```

```
CREATE TABLE Salas (  
sala_id INT AUTO_INCREMENT PRIMARY KEY,  
nomeDaSala VARCHAR(100),  
numeroDaSala INT  
);
```

```
CREATE TABLE Turmas (  
turma_id INT AUTO_INCREMENT PRIMARY KEY,  
nomeDaTurma VARCHAR(100),  
alunos TEXT,  
numeroMaximoDeAlunos INT,  
sala_id int,  
FOREIGN KEY(`sala_id`) REFERENCES Salas(sala_id));  
CREATE TABLE Feriados (  
feriado_id INT AUTO_INCREMENT PRIMARY KEY,  
datas DATE,  
nomes TEXT  
);
```

```
CREATE TABLE NLetivos (  
  nLetivo_id INT AUTO_INCREMENT PRIMARY KEY,  
  datas DATE  
);
```

```
CREATE TABLE PossivelPonte (  
  ponte_id INT AUTO_INCREMENT PRIMARY KEY,  
  datas DATE  
);
```

```
CREATE TABLE CursoProfessor (  
  prof_id INT AUTO_INCREMENT PRIMARY KEY,  
  curso_id int,  
  FOREIGN KEY(`curso_id`) REFERENCES Cursos(curso_id)  
);
```

```
CREATE TABLE CursoAluno (  
  curso_id int,  
  FOREIGN KEY(`curso_id`) REFERENCES Cursos(curso_id),  
  user_id int,  
  FOREIGN KEY(`user_id`) REFERENCES Usuarios(user_id),  
  prof_id int,  
  FOREIGN KEY(`prof_id`) REFERENCES CursoProfessor(prof_id)  
);
```



Uso das rotas e requisições:

GET <http://127.0.0.1:5000/user>

retorna:

```
{  
  "mensagem": "Lista de Usuarios",  
  "usuarios": [  
    {  
      "cpf": 2147483647,  
      "email": "arthur@aluno.com",  
      "nome": "Arthur",  
      "senha": "Saudade",  
      "user_id": 1  
    }  
  ]  
}
```

GET <http://127.0.0.1:5000/curso>

retorna:

```
{
  "cursos": [
    {
      "cargaHoraria": 80,
      "curso_id": 1,
      "dataFinal": null,
      "dataInicial": null,
      "descricao": "Um curso muito legal",
      "diasSemanaCurso": "Quinta-feira",
      "nomeCurso": "Python",
      "user_id": 1
    }
  ],
  "mensagem": "Lista de cursos"
}
```

POST <http://127.0.0.1:5000/user>

exige:

```
{
  "nome": "teste",
  "senha": "teste",
  "email": "teste@teste.com",
  "cpf": "123456789"
}
```

retorna:

```
{
  "mensagem": "Usuário cadastrado com sucesso",
  "user": {
    "cpf": 123456789,
    "email": "teste@teste.com",
    "nome": "teste",
    "senha": "teste",
    "user_id": 2
  }
}
```

POST <http://127.0.0.1:5000/curso>

exige:

```
{  
  "cargaHoraria": 99,  
  "dataFinal": "",  
  "dataInicial": "",  
  "descricao": "Um curso muito muito muito difícil",  
  "diasSemanaCurso": "Terça-feira",  
  "nomeCurso": "Java",  
  "user_id": 1  
}
```

retorna:

```
{  
  "mensagem": "Curso cadastrado com sucesso",  
  "user": {  
    "cargaHoraria": 99,  
    "curso_id": 2,  
    "dataFinal": null,  
    "dataInicial": null,  
    "descricao": "Um curso muito muito muito difícil",  
    "diasSemanaCurso": "Terça-feira",  
    "nomeCurso": "Java",  
    "user_id": 1  
  }  
}
```

POST <http://127.0.0.1:5000/login>

exige:

```
{  
  "senha": "teste",  
  "email": "teste@teste.com"  
}
```

retorna:

```
{  
  "mensagem": "Login com sucesso"  
}
```

POST <http://127.0.0.1:5000/logout>

exige:

```
{  
  "user_id": 1  
}
```

retorna:

```
{  
  "mensagem": "Logout bem Sucedido"  
}
```

Por enquanto é isso...