

Lab 05 - Utilização de base de dados em Node.js

Programação e Integração de Serviços

Docente: António Carlos Ferreira Pinto

2021/2022

1 - Código base

```
let express = require('express');
let app = express();
let bodyParser = require('express');

const PORT = process.env.PORT || 8081;
app.listen(PORT, () => console.log("Listening on http://localhost:" + PORT + "
..."));
app.use(express.static('www'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: false}));
```

Crie ainda uma pasta "www" e insira o código fornecido em Materiais/LAB5-Materiais disponível no Teams.

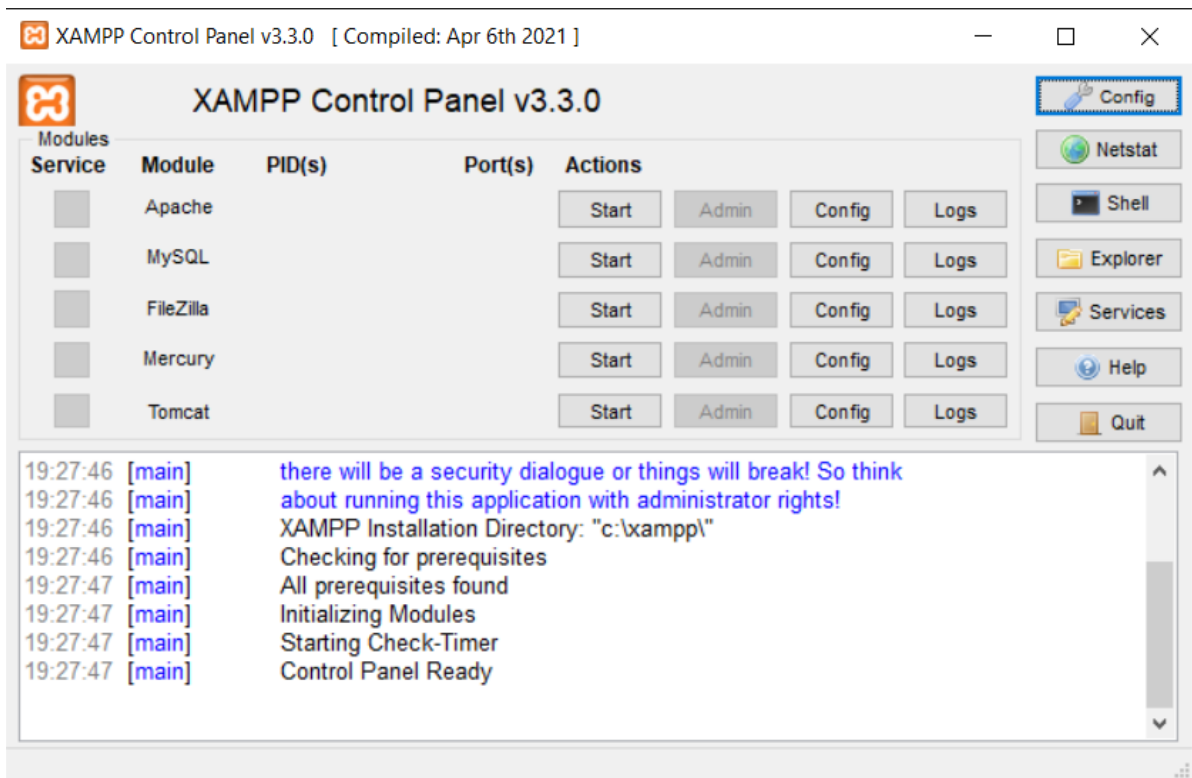
2 - Instalação do XAMPP

Vá ao website: <https://www.apachefriends.org/download.html> e faça o download mais recente do XAMPP

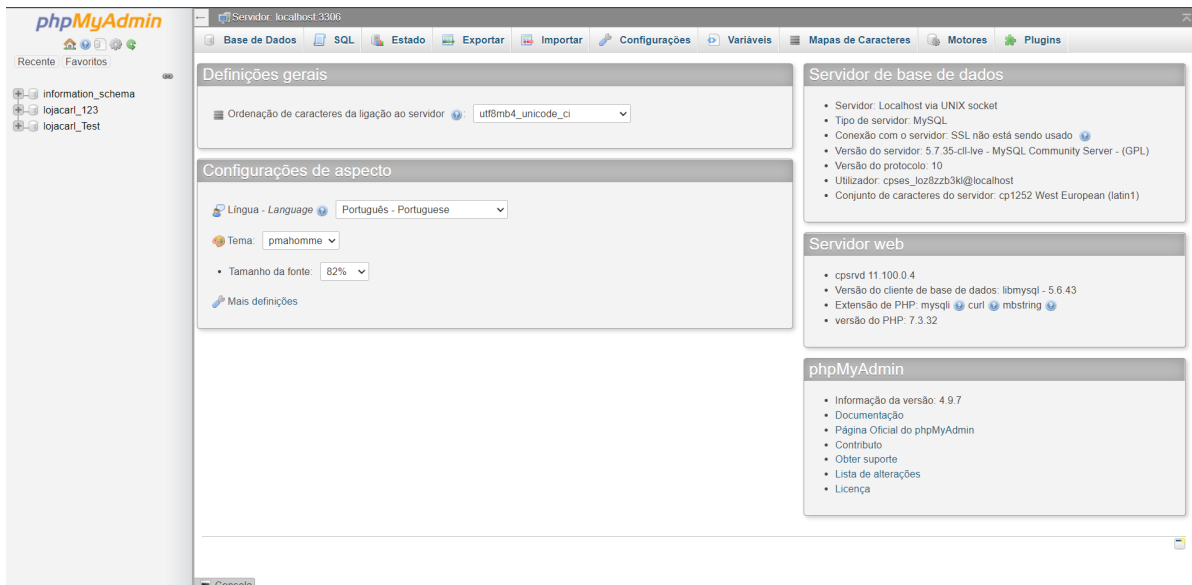
! Ao instalar a aplicação tenha o cuidado para não instalar na pasta C:/Programas porque pode criar conflito de permissões.

3 - Inicialização do servidor Apache e da base de dados

Depois de instalar o XAMPP, aparecem 5 módulos, Apache e MySQL serão os necessários correr para criar a conexão com a base de dados. Ao carregar em Config -> my.ini é possível ver as configurações da base de dados.



Depois de Iniciar tanto o MySQL e o Apache, clique em Admin. Irá abrir uma página do "phpMyAdmin" no seu browser que permite dar acesso a todas as funcionalidades da base de dados.



4 - Conectar á base de dados

Agora no código do servidor, é necessário fazer a conexão com a base de dados. Para isso precisa de aceder ao código do servidor

```
const mysql = require('mysql');

//database
const db = mysql.createConnection({
  host: '127.0.0.1',
  user: 'root',
  database: 'lab5'
});

//Connect to db
```

```

db.connect((err) => {
  if(err) throw err;
  console.log('Connected to Database')
});
//exemplo de utilização
/*
  app.get('/data', (req, res) =>{
    let sqlquery = `SELECT * FROM UTILIZADORES `;
    db.query(sqlquery, (err, results) =>{
      if(err) throw err;
      res.send(results);
    });
  });
*/

```

```

--create tables
CREATE TABLE SIZE (ID INT NOT NULL AUTO_INCREMENT, SIZE VARCHAR(255), PRIMARY
KEY (ID));
create table PET (ID INT NOT NULL AUTO_INCREMENT, NOME VARCHAR(255) NULL,
BIRTH_DATE DATE NULL, FK_SIZE_ID INT NULL , PRIMARY KEY (ID), FOREIGN KEY
(FK_SIZE_ID) REFERENCES SIZE(ID) );
create table USER (ID INT NOT NULL AUTO_INCREMENT, NOME VARCHAR(255) NULL,
BIRTH_DATE DATE NULL, PRIMARY KEY (ID) );
CREATE TABLE USER_PET (ID INT NOT NULL AUTO_INCREMENT , FK_USER_ID int NOT NULL,
FK_PET_ID INT NOT NULL ,PRIMARY KEY (ID), FOREIGN KEY (FK_USER_ID) REFERENCES
USER(ID), FOREIGN KEY (FK_PET_ID) REFERENCES PET(ID));
--inserts
INSERT INTO size(SIZE) VALUES ('Grande');
INSERT INTO size(SIZE) VALUES ('Médio');
INSERT INTO size(SIZE) VALUES ('Pequeno');

INSERT INTO pet( NOME, BIRTH_DATE, FK_SIZE_ID) VALUES ('Nero','2012-12-02','1');
INSERT INTO pet( NOME, BIRTH_DATE, FK_SIZE_ID) VALUES ('Boneca','2014-08-
12','3');

INSERT INTO user(ID, NOME, BIRTH_DATE) VALUES (1,'Antônio','2000-04-24');
INSERT INTO user(ID, NOME, BIRTH_DATE) VALUES (2,'Maria','2002-10-14');

INSERT INTO user_pet(FK_USER_ID, FK_PET_ID) VALUES (1,1);
INSERT INTO user_pet(FK_USER_ID, FK_PET_ID) VALUES (1,2);
INSERT INTO user_pet(FK_USER_ID, FK_PET_ID) VALUES (2,2);

```

5 - Criar métodos GET e POST

Agora depois de criada a base de dados crie no servidor as rotas GET e POST de utilizador e de animal. É de notar que as tabelas têm uma relação de n para n;

Está implementado o código exemplo de um formulário para um usuário. Crie agora um formulário semelhante que receba um id e retorne a informação desse animal

Finalmente crie um formulário para animal e para um usuário para que seja possível enviar criar um registo de um novo usuário e de um novo animal na página post.html

