

# Node.js, Express Framework, MySQL

## Instalação do node.js

1. Fazer download no Node.js: <https://nodejs.org>
2. Correr o instalador e seguir as instruções
3. Reiniciar o computador
4. Aceder à linha de comandos e testar a criação de um projeto

## Criação do primeiro projeto node.js

1. Abrir a linha de comandos
2. Criar uma pasta para o projeto: `mkdir olaMundo`
3. Aceder à pasta: `cd olaMundo`
4. Inicializar o package.json: `npm init`
5. Responder às questões colocadas (carregar em ENTER significa “concordo”). Exemplo:

```
C:\olaMundo>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (olamundo)
version: (1.0.0)
description: primeira aplicacao
entry point: (index.js)
test command:
git repository:
keywords:
author: Eu
license: (ISC)
About to write to C:\olaMundo\package.json:

{
  "name": "olamundo",
  "version": "1.0.0",
  "description": "primeira aplicacao",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Eu",
  "license": "ISC"
}
Is this OK? (yes)
```

6. Criar o ficheiro index.js na pasta do projeto e copiar para lá o código seguinte:

```
const http = require('http'); // Carrega o módulo HTTP
const server = http.createServer( // Cria uma instância do servidor
  // Enviar uma resposta HTTP com conteúdo "Ola mundo":
  function(request, response) {
    response.writeHead(200, {'Content-Type': 'text/html'}); // Inicia a
    resposta como página HTML
    response.write('<html><body><h1>Ola Mundo</h1></body></html>');
    response.end(); // Termina a resposta
  }
);
server.listen(3000); // O servidor escuta na porta 3000
console.log('Servidor a correr na porta 3000');
```

7. Iniciar o servidor: na linha de comandos executar o comando `node index.js`
8. Abrir o browser em `localhost:3000` e verificar que a mensagem é apresentada

## Usar a Framework Express

1. (Caso esteja a utilizar a rede da UPT – PCs fixos ou uptwlan) Definir o servidor de proxy. Na consola:

```
set http_proxy=http://proxy.uportu.pt:3128
set https_proxy=http://proxy.uportu.pt:3128
```

2. Instalar o módulo express usando a opção `--save` para guardar a dependência no projeto

```
npm install express --save
```

3. Criar um ficheiro app.js para configurar o servidor da aplicação com o seguinte código:

```
const express = require('express');
const app = express();
const server = app.listen(3000, function(){
  console.log("servidor a funcionar na porta %s", server.address().port);
});
```

4. Criar 4 rotas com diferentes comportamentos:

- a. `"/`: Mostra “Raiz”
- b. `"/ajuda`: Mostra “Ajuda”
- c. `"/hora`: Mostra a data/hora atual no formato dd/mm/aaaa hh:MM

**Nota:** documentação da função `Date()`: [https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/Date](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Date)

- d. `"/antigos`: Permite aceder a ficheiros estáticos HTML/CSS/JS (copiar a pasta “solução\_html” com a solução dos exercícios de HTML/CSS/JS das primeiras aulas)

## Usar MySQL com node.js

1. Instalar (com `--save`) os módulos `sequelize` e `mysql`
2. Utilizando `mysql` (não esquecer de criar o ficheiro de ligação), construir uma página que permita visualizar toda a informação sobre os carros presentes na base de dados (não esquecer a criação de rotas). Exemplo:

Marca	Modelo	Cilindrada	Fotografia
Audi	A3	2000	
Ford	Fiesta	1300	
Ford	Focus	1600	
Renault	Clio	1200	

3. Utilizando `sequelize`, construir uma página que permita visualizar toda a informação sobre as marcas presentes na base de dados. Exemplo:

Cod	Marca	Nome	Marca
1		Audi	
2		Ford	
3		Renault	
6		Fiat	
7		mitsubishi	