# Instruções rápidas para correr a aplicação:

1. Instalar o Node.js no Sistema: <https://nodejs.org/en/download/> (aplicação)
2. Instalar o MongoDB no Sistema: <https://www.mongodb.com/download-center> (aplicação)
3. Correr o ficheiro install.bat OU executar manualmente o comando npm install na pasta do projecto
4. Correr o ficheiro database.bat OU executar manualmente o comando mongod em qualquer local do sistema
5. Correr o ficheiro start.bat OU executar manualmente o comando node bin/www na pasta do projecto
6. A aplicação deve estar acessível na porta 3000. Pode ser acedida localmente com no endereço <http://localhost:3000>

# Detalhes de implementação do programa:

O programa é uma aplicação web feita em [Node.js](https://nodejs.org/en/), uma framework assincrona baseada em javascript. Dado isso, o código da aplicação é primáriamente escrito em ficheiros com a extensão .js. Nesta pasta poderão ser encontrados também diversos ficheiros relevantes à implementação de uma aplicão em Node.

Uma instalação do Node encapsula também a instalação de um outro programa: npm, *Node Package Manager,* com a função de facilitar a instalação de bibliotecas javascript necessárias para o projecto em Node correr. A definição de quais bibliotecas o projecto precisa encontra-se no ficheiro package.json. Este é um ficheiro comum à grande maioria dos projectos de Node, e é normalmente considerado um ”sumário” do estado actual do projecto, com outra informação. Uma destas informações adicionais é o script de arranque, ou seja, a indicação de como começar a aplicação. No caso deste projecto em particular, o ficheiro que inicializa o Node encontra-se na pasta bin, com o nome www (sem extensão). O código rlevante à aplicação própriamente dita, no entanto, é chamada a partir do ficheiro app.js no topo do directório, e a partir desta, outros ficheiros vão-se encontrar dentro de pastas cujas funções vou descrever de seguida após um aparte.

Antes de descrever o resto da aplicação, gostaria de chamar a atenção a alguns detalhes. Em primeiro lugar, as bibliotecas obtidas por npm. Para instalar as bibliotecas necessárias ao projecto, o comando npm install deve ser corrido num terminal a partir do directório do projecto. Isto vai gerar uma pasta node\_modules na qual se encontram as bibliotecas. Uma das bibliotecas, mocha, serve para correr scripts de teste ao código da aplicação (<https://mochajs.org/>). O script de teste encontra-se definido em package.json, e portanto pode ser arrancado com o commando npm test mas os ficheiros com os testes própriamente ditos encontram-se definidos dentro da pasta test. De momento, apenas é testada a consistência de dados de algumas funções da base de dados. Ainda relativamente aos testes, existe um ficheiro no directório principal chamado .travis.yml, este é um ficheiro muito básico mas que permite correr os script de teste num serviço de integração contínua (<https://travis-ci.com/>) sempre que ele foi submetido a um repositório externo, para verificar se ocorrem erros relativos a funcionalidades anteriores numa nova iteração. Finalmente, os ficheiros .bat são ficheiros de script batch para facilitar correr os comandos de instalar e lançar a aplicação em Windows.

# Pastas:

- Config

Pasta com ficheiros com definição de variáveis importantes. Por exemplo, em dbURL.js encontra-se o url para a base de dados mongo. O default deve ser para uma base de dados local, mas pode ser definda uma externa.

- Controllers

Controllers neste contexto são ficheiros que defiem funções que fazem a transição entre a base de dados e a aplicação. Há 4 módulos de acesso: Comunidades, Utilizadores, Ofertas e Mensagens.

- Routes

Os ficheiros nestas pastas, funcionam ambos para fazer o routing dos pedidos dependendo da página pedida a cada um dos módulos correctos, como de controladores dos dados a ser enviados em cada um dos módulos. Os mesmos módulos podem ser encontrados, assim como uns ficheiros na raiz correspondentes a outras pa´ginas adicionais.

- Views

Os ficheiros aqui apresentados tem uma extensão diferente, .ejs. Os fichiros .ejs, são funcionalmente idênticos a ficheiros html or php, mas tem uma funcionalidade acrescida: Pode utilizar código javascript e dados enviados pelo router/controlador para ajudar a gerar a página final que vai ser enviada ao cliente. A pasta templates tem ficheiros parciais que são incluidas noutras, como por exemplo os dados da navbar.

- Public

Pasta com ficheros que podem er acedidos directamente pelo cliente. São imagens, stylesheets e bibliotecas/scripts.