IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

CTeSP em Programação Web, Dispositivos e Aplicações Móveis.

Programação de Serviços Web.

To Do Heroes

Aluno : Júlio Sousa, Nº 190208014

Docente : André Antunes

Data : 20/01/2021

Ano Letivo 2020/2021

ÍNDICE

INTRODUÇÃO

Desde o início das aulas neste segundo ano, tivemos de criar vários projectos para quase cada uma das disciplinas correspondentes, com o objectivo de solidificar e, claro, demonstrar o nosso conhecimento e/ou potencial através deles.

Sinto a necessidade de expressar emoção sobre este projecto, porque acredito que pode ter potencial nas mãos certas.

Quando estávamos à procura de projectos para implementar, vi-me obrigado a procurar algum tipo de necessidade, que pudesse ser resolvida total ou parcialmente com a ajuda de software.

Esta solução tinha de ser simples, não para o programador mas sim para o cliente. Tinha de ser uma ferramenta fácil de usar e facilmente acessível.

Após algum tempo de reflexão, encontrei uma necessidade que poderia ser resolvida total ou parcialmente com a ajuda de software, facilmente acessível e utilizável pelo cliente.

A necessidade que escolhi resolver descrita numa palavra é "Desorganização".

Não é novidade que as empresas e/ou organizações aumentam o seu número de elementos à medida que o tempo passa, isto porque procuram a necessidade de atender mais objectivos e de se posicionarem melhor no mercado.

Devido a isto, cada vez mais empresas e/ou organizações procuram implementar transformações tecnológicas nas suas áreas de desenvolvimento, o que pode obviamente contribuir para a realização de objetivos ou metas.

To Do Heroes, é um projecto ambicioso que procura organizar, gerir, segmentar, distribuir, armazenar e comunicar, objectivos, projectos ou metas. Dentro de uma empresa ou organização.

1. ESPECIFICAÇÃO DAS FUNCIONALIDADES

A aplicação deve permitir as seguintes funcionalidades:

Registo de Utilizadores.

Criar tarefas que estejam relacionadas com datas, podendo expirar, ser associadas a projetos ou não, ser associadas a prioridade ou não, ser associadas a vários utilizadores ou não e ter ou não comentários.

Os utilizadores podem ter contatos (*Friends*), procurar por utilizadores na plataforma, receber convites e devem ter utilizadores administradores de projecto.

Os administradores devem ser capazes de alterar as senhas, ver o gráfico do número de contas premium, agrupar diferentes utilizadores num único projecto, bloquear a conta do utilizador cliente.

2. ESPECIFICAÇÃO DO SISTEMA A IMPLEMENTAR

Este projecto, correspondente à disciplina de programação de serviços web, apresentará a metodologia do tipo *REST* e a estrutura do código é baseada no modelo *MVC*. *Node.js* será utilizado para abrir os canais de comunicação entre o servidor e o nosso PC.

Várias bibliotecas como a *Express*, *Passport* e outras serão utilizadas (de que falaremos mais detalhadamente na seção de especificações técnicas) para a criação do servidor e a validação do utilizador.

A seguir serão descritas todas as ferramentas e tecnologias que se encontram neste projecto.

3. Base de dados

3.1. Modelo de Dados

O modelo de dados baseia-se num modelo de tipo relacional para o qual será utilizado o sistema de gestão de bases de dados *MySql*, especificamente o *MySql Workbench*, que é uma das ferramentas de concepção visual de bases de dados para os sistemas de bases de dados *MySql*.

De acordo com a figura *Fig. Modelos de dados*, podemos fazer as seguintes afirmações:

- Pode haver mais do que um tipo de utilizador, mas um utilizador está associado a um e apenas a um tipo.
- Um utilizador pode ter zero ou vários contactos, mas um contacto está associado a um ou vários utilizadores.
- Um utilizador pode ter zero ou vários convites, mas um convite está associado a um ou vários utilizadores.
- Um utilizador pode ter zero ou múltiplas tarefas, mas uma tarefa está associada a um ou vários utilizadores.
- Um utilizador pode ser associado a zero ou vários projectos, mas um projecto está associado a um ou vários utilizadores.

- Uma tarefa pode ser associada a zero ou múltiplos projectos e um projecto pode ser associado a zero ou múltiplas tarefas.
- Uma tarefa está associada a zero ou múltiplos comentários, mas um comentário está associado a uma e apenas uma tarefa.
- Uma tarefa pode ser associada a zero ou múltiplas prioridades e uma prioridade pode ser associada a múltiplas tarefas.

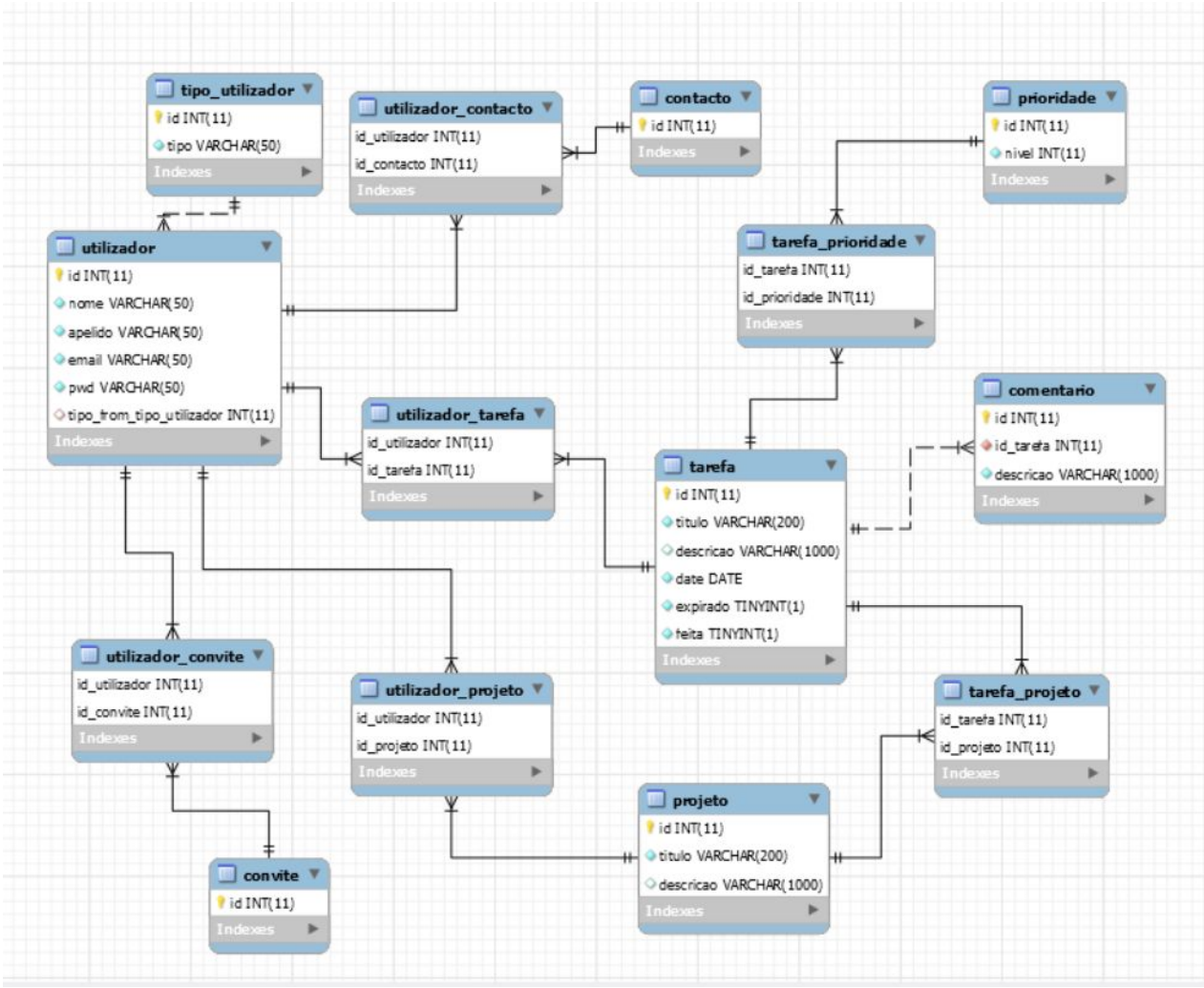


Fig. Modelos de dados

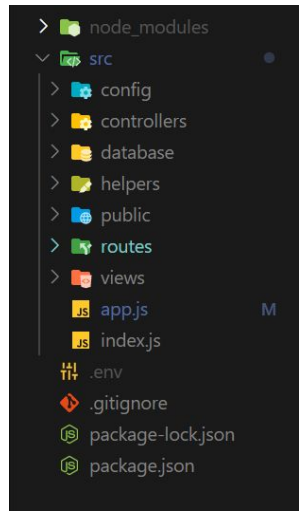
4. Ferramentas e Tecnologias Back-End

4.1. App.js e Index.js

Primeiro, foram instaladas certas dependências (que são explicadas mais detalhadamente na seção de especificações técnicas) no ficheiro *app.js*, foram efectuadas importações para a dependência de *Express*, *Passport*, *Session*, *Express-handlebars*, entre outras. Neste ficheiro, foram feitas as configurações relevantes para o correcto funcionamento da aplicação

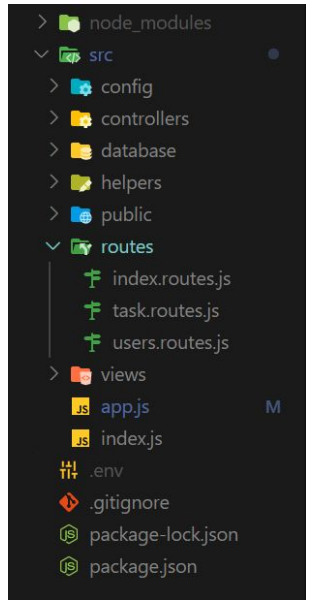
Como a configuração do *view engine*, que estará ligada à importação de *Express-handlebars*, a chamada dos caminhos (que estarão em pastas e ficheiros diferentes), a configuração da sessão e a chamada ao ficheiro *passport.js* em *./config* que contém o código necessário para validar a entrada do utilizador na aplicação, bem como as permissões para entrar em cada uma das páginas da aplicação web.

No entanto, o ficheiro *app.js* não é o encarregado de iniciar o servidor, essa tarefa seria deixada ao ficheiro *index.js* para chamar o componente *app* em *app.js* e o porto, a fim de iniciar a aplicação através de um método chamado *listen*.



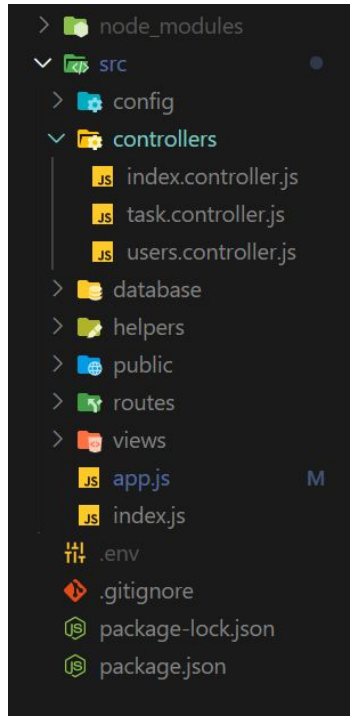
4.2. Rotas

Todas as rotas estão em *./routes*, nestes ficheiros residem todos os pontos finais e encaminham os dados para o seu respectivo controlador, que actuará de acordo com os dados que forem passados.



4.3. Controladores

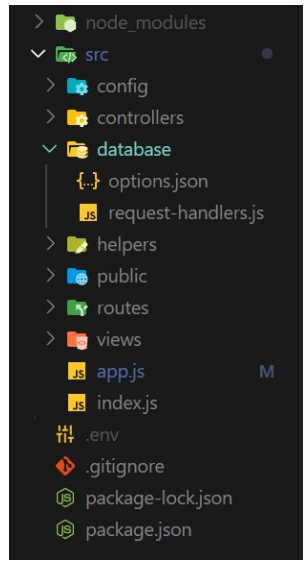
Os controladores estão localizados no caminho *./controllers*, aqui pode encontrar todos os controladores separados em ficheiros descritivos, de acordo com a sua tarefa, em geral, estes condutores, conforme o caso, fazem referências a funções encontradas num ficheiro chamado *request-handlers.js*, que são responsáveis por fazer a consulta com a base de dados.



4.4. DATABASE (MODELOS)

No caminho *./database* podemos encontrar o arquivo *request-handlers.js*.

Todas as funções nele encontradas referem-se a uma única função, que se chama *sendRequest*. Esta função devolve um objeto de promessa que, por sua vez, realiza a consulta da base de dados.



5. Ferramentas e Tecnologias Front-end

Ao nível do frontend temos dois pontos a descrever, interfaces e ficheiros estáticos. Todas as interfaces funcionarão através das bibliotecas da Express-handlebars e terão a extensão .hbs.

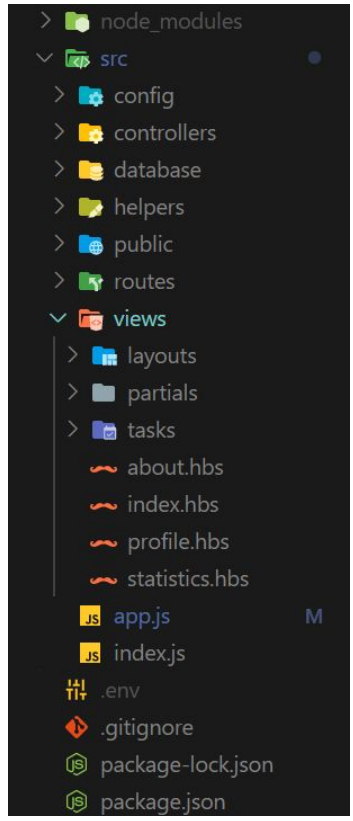
Por ficheiros estáticos, entendemos aqueles que foram encontrados pelo navegador, que será capaz de aceder livremente.

5.1. Interfaces

As interfaces podem ser encontradas no caminho ./views, dentro deste caminho podemos encontrar um conjunto de pastas e ficheiros .hbs, todos com nomes descritivos de acordo com a sua função.

Foi utilizado código de handlebars para minimizar a quantidade de código a escrever e também para adicionar uma funcionalidade dinâmica às interfaces.

default.js é o ficheiro com a interface principal a qual faz referência a todas as outras interfaces.

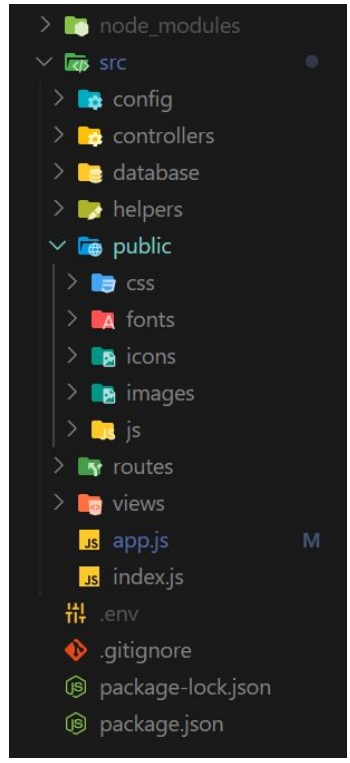


5.2. FICHEIROS ESTÁTICOS

Todos os ficheiros estáticos serão encontrados no caminho ./public, este caminho armazena um conjunto de ficheiros e pastas que contribuem principalmente para a aparência da aplicação web, mas também fazem pedidos para o backend da aplicação.

Entre os ficheiros e pastas temos ficheiros .css dedicados a estilos, .svg .ttf .woff dedicados a fontes e ícones, e .png .jpg dedicados a imagens.

Dentro do caminho .public/js podemos encontrar um conjunto de ficheiros .js, cada um deles cheio de código JavaScript, que ajudam com todas as funcionalidades encontradas na aplicação web, tais como o funcionamento dos modos, gráficos e outros eventos.



6. Serviço REST e Endpoints

A metodologia REST será implementada ao longo de todo o projecto, sendo esta a ponte entre o Frontend e o Backend da aplicação.

Através desta metodologia serão feitos pedidos de acordo com os verbos GET,POST,PUT e DELETE, a fim de realizar ações relativas ao Backend da aplicação, bem como a criação, obtenção, atualização e eliminação de dados (CRUD).

Os pedidos serão recebidos através da url do navegador, que de acordo com a sua estrutura realizará uma determinada consulta ou ação na base de dados.

```
let sendHttpRequest = (url,method,data,callback) => {
    let xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function () {
            if ((xhr.readyState == XMLHttpRequest.DONE) && (this.status === 200)) {
            let response = JSON.parse(xhr.response);
            callback(response);
        }
    }
    xhr.open(method,url);
    xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhr.send(JSON.stringify(data));
}
```

```
// Edit Task
router.put("/task/edit", isAuthenticated, taskController.updateTask);
```

7. Especificaciones Técnicas

Express - É uma estrutura web, escrita em JavaScript e alojada dentro do ambiente de execução do NodeJS.

Será utilizado para a criação do servidor e para a gestão das dependências instaladas na aplicação.

Passaporte - É uma estrutura de código aberto para a gestão da autenticação que funciona em combinação com Connect e Express.

A sua configuração pode ser implementada nos ficheiros encontrados no caminho ./config e ./helpers. Será utilizado no projeto não só para autenticação do utilizador antes de entrar na aplicação, mas também para serialização e desserialização da aplicação, bem como para saber se um utilizador tem ou não permissão de acesso a algumas das páginas da aplicação web.

Express-handlebars - é uma biblioteca JavaScript que nos permite escrever etiquetas HTML, podendo definir o que imprime, o contexto e a forma como o faz.

Descrita acima, esta biblioteca será utilizada para a criação das interfaces, os dados a serem mostrados nelas serão enviados através da metodologia REST.

8. CONCLUSÃO

Após a realização deste projecto, posso concluir que o processo de construção de uma aplicação web foi emocionante para mim.

Todos os processos de integração das funcionalidades, tanto no frontend, backend, visualização do cliente, realização de boas práticas, têm sido um desafio.

Posso dizer que existiam certas teorias a rever e novos aspectos a investigar, no entanto, sinto que tenho sido capaz de ultrapassar todos os obstáculos.

Para além deste projecto não cumprir todas as especificações estabelecidas na proposta e do seu início, a conclusão deste projecto seria algo interessante de ver e implementar como um negócio ou simples hobby-practice.

Tenho muito a agradecer a esta disciplina e ao Professor André Antunes, porque foi realmente o que eu esperava ao nível da exigência e da metodologia.

9. WEBGRAFIA

https://www.w3schools.com/html/html_form_input_types.asp

https://sweetalert.js.org/

https://getbootstrap.com/

https://around.createx.studio/components/typography.html

https://www.chartjs.org/

https://www.npmjs.com/

https://www.npmjs.com/package/express

http://www.passportjs.org/

https://www.npmjs.com/package/express-session

https://handlebarsjs.com/

https://www.mysql.com/

https://nodejs.org/es/