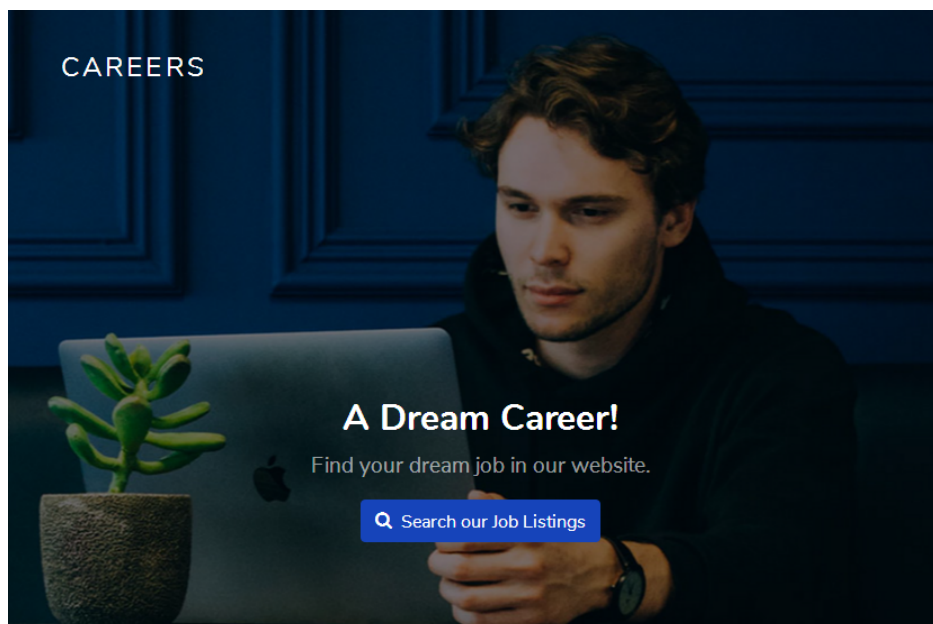


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

---

O objectivo do exame é construir um site web em programação "front-end" usando o *Framework VueJS 3*. O site é suportado por uma base de dados relacional. Informação sobre a base de dados encontra-se em ANEXO.



O site web consiste no portal de uma empresa de empregos que permite a utilizadores registados **publicar ofertas de emprego**.

Pode ver uma implementação do site no URL

<http://daw.deei.fct.ualg.pt/~a444440/dist/>

O site tem as seguintes páginas:

1. “**Home**” é a página de rosto do site;
2. “**Job Listings**” é a página principal do site;
3. “**Register**” para registo de utilizadores;
4. “**Login**” para o login de utilizadores;
5. “**Publish**” permite ao utilizador registado introduzir uma oferta de emprego

6. “**Message**” permite enviar mensagens; (**dado**)

**As APIs de acesso à base dados são dadas, bem como é dada uma VueJS "store" que contém todas as "ações" AJAX necessárias para acesso às APIs**

**É dado acesso a um conjunto de templates HTML construídos em *Bootstrap* fornecidos pela empresa de web design Colorlib. A adaptação dos templates ao framework VueJS **não pode alterar** o layout e o estilo!**

Solicita-se ao aluno a realização do site web em *Framework VueJS 3*, apenas das vistas acima descritas.

**Deve ser usado o seguinte mapeamento entre URLs e vistas:**

```
const routes = [  
  {  
    path: '/',  
    component: Home  
  },  
  {  
    path: '/jobs',  
    component: Jobs  
  },  
  {  
    path: '/register',  
    component: Register  
  },  
  {  
    path: '/login',  
    component: Login  
  },  
  {  
    path: '/publish',  
    component: Publish  
  },  
  {  
    path: '/message',  
    component: Message  
  },  
]  
]
```

## PRELIMINARES

- **VUEJS**

Faça o download de uma instalação base do VueJS 3 adaptada para este exame em<sup>1</sup>

<https://github.com/jmatbastos/EXAME/archive/refs/heads/master.zip>

e expanda o ficheiro `EXAME-master.zip` para dentro pasta

`C:\XAMPP\htdocs`

- Se necessário mude o nome da pasta de `EXAME-master` para `EXAME`

- **VUE STORE**

Na pasta `C:\XAMPP\htdocs\EXAME\src\store` encontra o código de todas as acções necessárias para acesso AJAX à base de dados.

Nos ficheiros `"jobs.js"`, `"jobcategories.js"`, `"user.js"`  
**actualize os URLs**

`http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/...`

e substitua `"12345"` pelo seu nº de aluno

- **API E CREDENCIAIS DE ACESSO À BASE DE DADOS**

**Com o seu editor preferido abra o ficheiro**

`C:\XAMPP\htdocs\EXAME\api\db.php`

**e actualize com as suas credenciais de acesso à base de dados**

```
$hostname = "10.10.23.184";  
$db_name = "db_a12345";  
$db_user = "a12345";  
$db_passwd = "PASS";
```

---

<sup>1</sup> em alternativa, se tem a aplicação git instalada no seu PC/portátil:

`C:\XAMPP\htdocs> git clone https://github.com/jmatbastos/EXAME.git`

(substitua '12345' pelo seu nº de aluno e 'PASS' pela sua password de acesso à base de dados no servidor de produção...)

**Com o seu programa scp preferido (scp, FileZilla, WinSCP, etc..) faça o upload da pasta**

- "api "

**para dentro** da pasta "EXAME"<sup>2</sup> no servidor 10.10.23.184<sup>3</sup>

- **ATUALIZAÇÃO DA BASE DE DADOS**

**Faça login com o PuTTY** no servidor 10.10.23.184 **e atualize a sua base de dados** no servidor de produção:

```
a12345@daw2:~$ cd public_html/api  
  
a12345@daw2:~/public_html/EXAME/api$  
mysql -ua12345 -pPASS db_a12345 < ~/public_html/EXAME/api/db.SQL
```

**OPCIONAL: Método alternativo de atualização da base de dados:**

Use o software **phpMyAdmin** disponível no URL

<http://daw.deei.fct.ualg.pt/phpMyAdmin>

**e importe** o ficheiro

C:/XAMPP/htdocs/EXAME/api/db.SQL

Encontra no **ANEXO 1** a descrição da estrutura da base de dados.

Encontra no **ANEXO 2** a descrição das APIs que permitem o acesso à base de dados

---

<sup>2</sup> A pasta "EXAME" já foi previamente criada. Se a pasta não existe, faça login com o PuTTY no servidor 10.10.23.184 e crie uma pasta com o nome "EXAME" dentro da pasta "public\_html":

```
a12345@daw2:~$ cd public_html  
a12345@daw2:~/public_html$ mkdir EXAME
```

<sup>3</sup> Também pode fazer o upload para o servidor ssh.deei.fct.ualg.pt

## TESTE A INSTALAÇÃO DAS API

A partir do seu browser preferido vá ao seguinte URL

`http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/jobs.php`

(substitua "12345" pelo seu número de aluno)

Deverá receber uma página web com dados no formato JSON:

```
[{"title":"Junior programmer","id":"11","image":"5.jpg","location":"FARO","salary":"10000","description":"we expect aVUEJS expert but only have money to pay a junior programmer","company":"FAGAR","created_at":"2023-11-11 18:04:42","publisher":"JOSE","category":"Temporary"}, {"title":"Front End Developer in Vue.js","id":"5","image":"5.jpg","location":"London","salary":"50000","description":"Front-end developer who masters JavaScript, Angular, React, Vue","company":"Ebay","created_at":"2019-12-18 23:33:15","publisher":"Luc Bresson","category":"Internship"}, {"title":"FullStack Developer in Shopify","id":"3","image":"3.jpg","location":"Brussels","salary":"80000","description":"Do you look beyond the code and back-end solution?","company":"Facebook","created_at":"2019-12-18 23:30:51","publisher":"Neil Diamond","category":"Full-time"}, {"title":"Dropbox Product Designer","id":"4","image":"4.jpg","location":"Dublin","salary":"60000","description":"Back-end developer with expertise in Java","company":"Yahoo","created_at":"2019-12-18 23:30:51","publisher":"Marc Andressen","category":"Internship"}, {"title":"Senior Programmer in Laravel","id":"2","image":"2.jpg","location":"Faro","salary":"70000","description":"You will work alongside some of the best developers","company":"Spotify","created_at":"2019-12-18 23:25:53","publisher":"Peter Smith","category":"Part-time"}, {"title":"Dropbox Product Designer","id":"1","image":"1.jpg","location":"Barcelona","salary":"50000","description":"Developing cutting-edge digital solutions","company":"Software startup","created_at":"2019-12-18 23:25:53","publisher":"John Stewart","category":"Freelancer"}]
```

## TESTE A INSTALAÇÃO DO FRAMEWORK

- Execute o comando

```
C:\XAMPP\htdocs\EXAME> npm install
```

- Execute o comando

```
C:\XAMPP\htdocs\EXAME> npm run dev
```

A partir do seu browser preferido vá ao seguinte URL

**`http://localhost:8080/~a12345/EXAME/dist/`**

Deverá receber uma página web de boas-vindas



Welcome to Your EXAME Vue App

# 1. [3 valores] FUNCIONALIDADE “HOME”

A funcionalidade “Home” é página de rosto do site

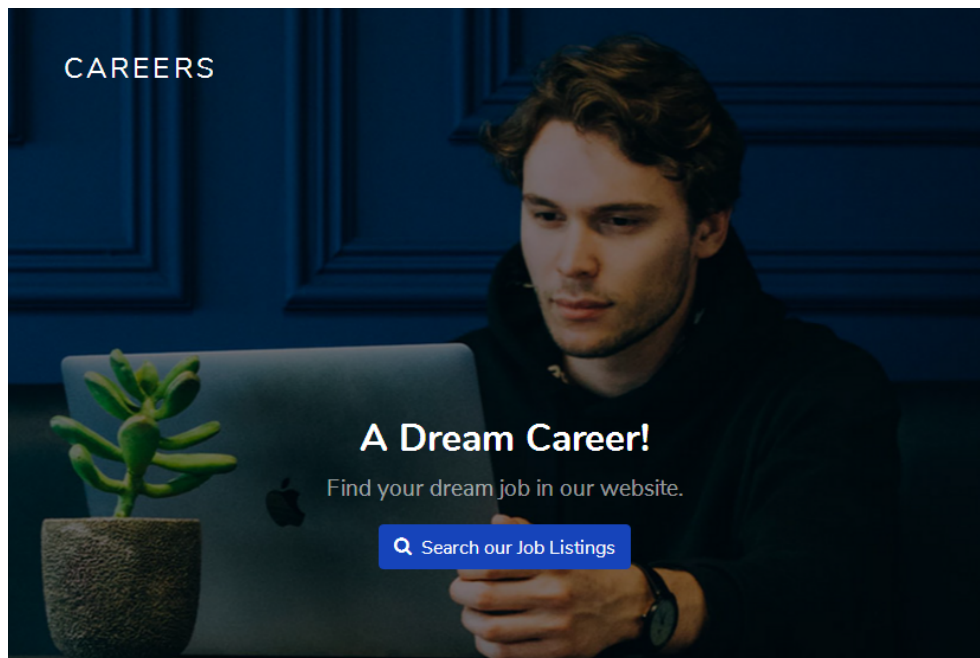
Construa a vista para esta página no ficheiro

```
C:\XAMPP\htdocs\EXAME\src\views\Home.vue
```

SUGESTÃO: Adapte o template “index.html” fornecido pela empresa de web design. Encontra este template na pasta “demo” (C:\XAMPP\htdocs\EXAME\demo).

Pode ver o template no browser no URL

<file:///C:/XAMPP/htdocs/EXAME/demo/index.html>



“Search our Job Listings!” é um link (“/jobs”) que permite mostrar a página com as ofertas de emprego já introduzidas na base de dados

## NOTA IMPORTANTE:

- **os links** para os recursos estáticos css e javascript utilizados no template (`<link rel="stylesheet" href="...">` `<script src="...">`) **estão já colocados** no ficheiro "index.html" (**dado**)
- **os recursos estáticos** utilizados no template (img, css, fonts, js) **estão já colocados** em pastas com o mesmo nome (img, css, etc) dentro da pasta "public" (**dado**)
- Prefere construir o site com as imagens na pasta `src/assets`? Terá de copiar a pasta `demo/img` para dentro da pasta `assets`.



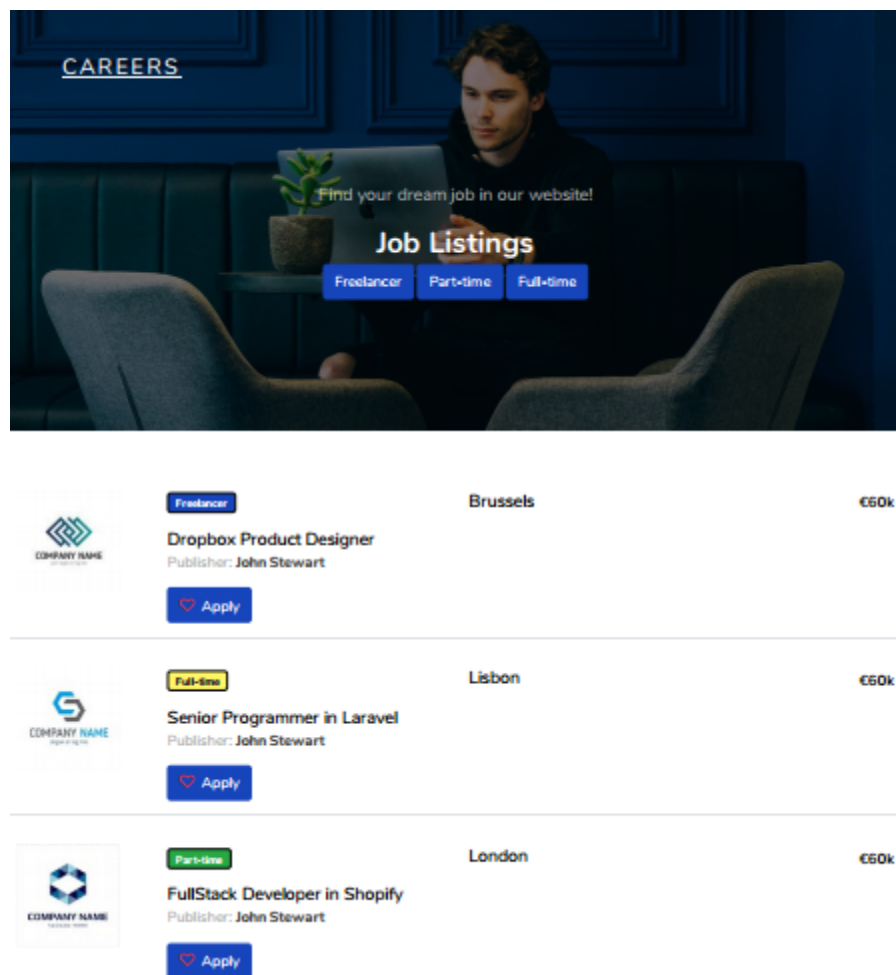
## 2. [7 valores] FUNCIONALIDADE “JOB LISTINGS”

A funcionalidade “Job Listings” é página principal do site e contém uma lista dos empregos publicados.

Construa a vista `Jobs.vue` para esta página.

SUGESTÃO: Adapte o template “`job-listings.html`” fornecido pela empresa de web design.

Encontra este template na pasta “demo” (C:\XAMPP\htdocs\EXAME\demo).



Utilize os métodos

```
jobsStore.getJobsDB(),  
jobCategoriesStore.getCategoriesDB()
```

para fazer o download de uma lista da oferta de empregos, bem como das categorias de ofertas de emprego, existentes na base de dados

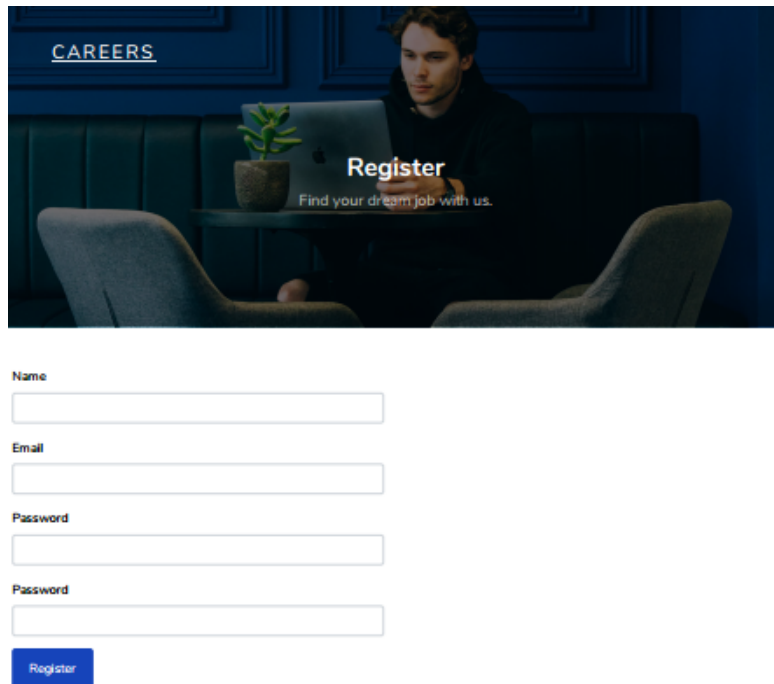
- A página principal do site ("/jobs"), mostra todas as ofertas de emprego
- A página principal do site tem um menu de "job categories" (Full-time, Part-time etc) **criado dinamicamente** a partir da tabela "jobcategories" existente na base de dados. Para cada "job category" deverá haver um evento que permite filtrar as ofertas de emprego de uma determinada categoria
- Para cada oferta de emprego mostre a imagem, a categoria, o título, o "publisher", a localização, a data de publicação, e o salário existentes na tabela "jobs" da base de dados

### 3. [2 valores] FUNCIONALIDADE “REGISTER”

Construa a vista `Register.vue` para esta página.

SUGESTÃO: Adapte o template “`register.html`” fornecido pela empresa de web design.

Encontra este template na pasta “demo” (C:\XAMPP\htdocs\EXAME\demo).



The image shows a person sitting on a couch in a dimly lit room, using a laptop. Overlaid on the laptop screen is a 'Register' form with the text 'Find your dream job with us.' Below this, there is a separate registration form with the following fields:

- Name:
- Email:
- Password:
- Password:
- Register:

Utilize os métodos

```
userStore.userExistsDB(user) e  
userStore.addUserDB()
```

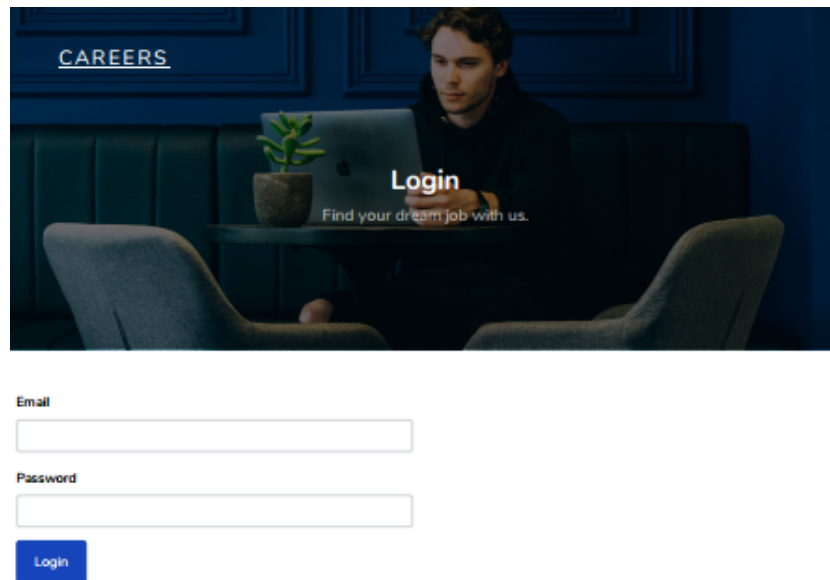
para registrar um novo cliente na base de dados

- Se o email ainda não existe na base de dados, regista o utilizador na tabela “users” e envia uma mensagem de sucesso ao utilizador.
- Se o email já existe na base de dados aparece uma mensagem de erro no formulário

## 4. [3 valores] FUNCIONALIDADE “LOGIN” & “LOGOUT”

Construa a vista `Login.vue` para esta página.

SUGESTÃO: Adapte o template “`login.html`” fornecido pela empresa de web design. Encontra este template na pasta “demo” (C:\XAMPP\htdocs\EXAME\demo).



Utilize o método

```
userStore.loginUserDB(user)
```

para fazer o login do utilizador e registá-lo na "store"

- em caso de sucesso no login:
  - re-direcciona para a página “message” com a mensagem “Welcome back!”
  - regista na store os dados do utilizador retirados da base de dados
  - Na página “Job Listings” o hyperlink “Login” transforma-se no hyperlink “Logout” e o hyperlink “Register” transforma-se no texto “Welcome user” (onde “user” é o nome do utilizador registado)
  - Na página “Job Listings” aparecem dois novos links “Publish” e “My Offers”
- em caso de insucesso no login
  - não sai da página `login`.
  - Envia a mensagem de erro “Bad credentials”

Utilize o método `userStore.logoutUserDB()` para encerrar a sessão

## 5. [3 valores] FUNCIONALIDADE “PUBLISH”

A funcionalidade “Publish” **permite ao utilizador registado** introduzir uma nova oferta de emprego

Construa a vista `Publish.vue` para esta página.

SUGESTÃO: Adapte o template “`publish.html`” fornecido pela empresa de web design. Encontra este template na pasta “demo” (C:\XAMPP\htdocs\EXAME\demo).

Company Name	Job Logo
<input type="text" value="Enter here the company name"/>	<input type="text" value=""/>
Job Title	Job Category
<input type="text" value="Enter here a job title"/>	<input type="text" value=""/>
Job Description	
<div>Write here a few lines to describe the job...</div>	
Salary	Job Location
<input type="text" value="Annual salary in euros..."/>	<input type="text" value="Enter here the city..."/>
<input type="button" value="Publish"/>	

Utilize o método `jobsStore.newJobDB()` para introduzir uma nova oferta de emprego na base de dados

Envie a mensagem "Success, your job offer is registered".

Nota:

- o "logotipo" da empresa é um "select/dropdown"; as imagens já estão pré-definidas (para evitar o upload de imagens...)
- a "categoria" da oferta de emprego é um "select/dropdown"; as categorias já estão pré-definidas

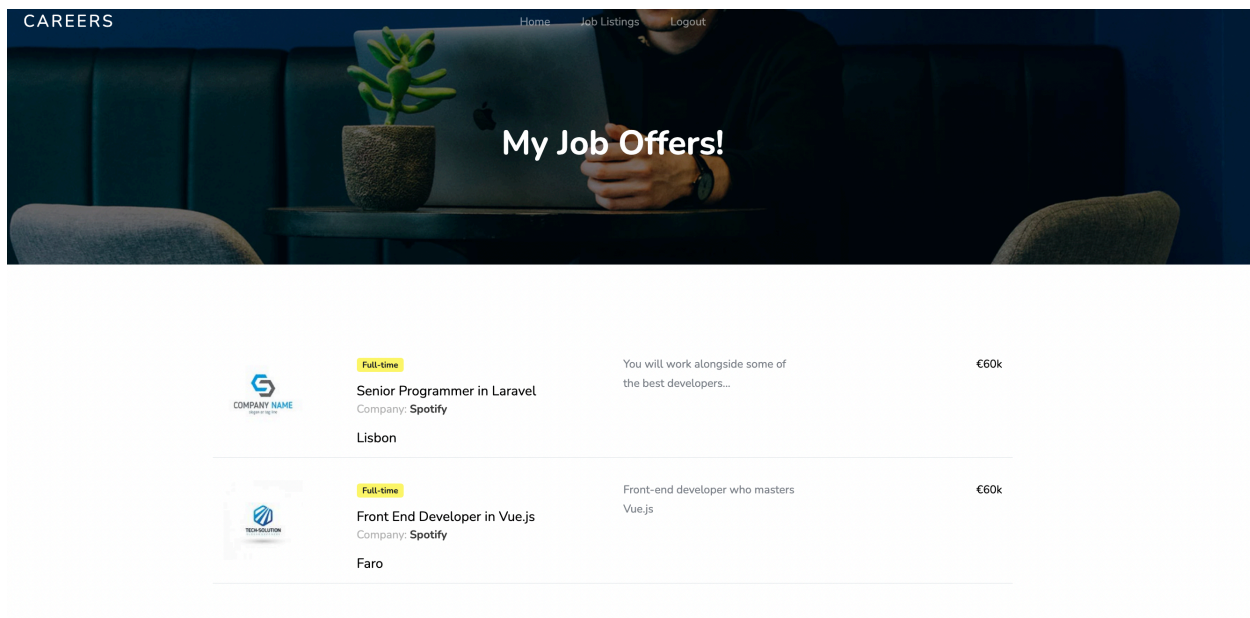
## 6. [2 valores] FUNCIONALIDADE “My Job Offers”

A funcionalidade “My Job Offers” **permite ao utilizador registado** fazer um filtro das ofertas de emprego registadas, e mostrar exclusivamente as ofertas de emprego da sua autoria.

Construa a vista `MyOffers.vue` para esta página.

SUGESTÃO: Adapte o template “`myoffers.html`” fornecido pela empresa de web design.

Encontra este template na pasta “demo” (C:\XAMPP\htdocs\EXAME\demo)



Construa na vista `MyOffers.vue` o método responsável pela filtragem referida.

# BUILD/UPLOAD

**Atualize** o “publicPath” no ficheiro

```
C:\XAMPP\htdocs\EXAME\vite.config.js
```

com o conteúdo

```
base: '/~a12345/EXAME/dist/'
```

(substitua '12345' pelo seu nº de aluno!)

**Execute a compilação/"build" com o comando**

```
C:\XAMPP\htdocs\EXAME> npm run build
```

**Faça o upload** com scp/WinSCP/FileZilla das pastas

- "src"
- "dist"

para a pasta “EXAME” no seu site web pessoal no servidor de produção

**Teste o funcionamento do site** no URL

<http://daw.deei.fct.ualg.pt/~a12345/EXAME/dist/>

(substitua '12345' pelo seu nº de aluno...)

**NÃO espere pelo último minuto do exame para fazer o upload!!**

**NÃO faça upload da pasta "node\_modules"!!**

# ANEXO 1. Estrutura da base de dados

- O acesso à base de dados MySQL pode ser feita em linha de comando (substitua “12345” pelo seu número de aluno)

```
a12345@daw2:~$mysql -u a12345 -p -h 10.10.23.184 db_a12345
```

ou usando software **phpMyAdmin** disponível no URL

<http://daw.deei.fct.ualg.pt/phpMyAdmin>

```
--
-- Table structure for table `publishers`
--

CREATE TABLE IF NOT EXISTS `publishers` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(255) default NULL,
  `email` varchar(255) default NULL,
  `created_at` datetime NOT NULL,
  `updated_at` datetime NOT NULL,
  `password_digest` varchar(255) default NULL,
  `remember_digest` varchar(255) default NULL,
  `admin` tinyint(1) default NULL,
  `activation_digest` varchar(255) default NULL,
  `activated` tinyint(1) default NULL,
  `activated_at` datetime default NULL,
  `reset_digest` varchar(255) default NULL,
  `reset_sent_at` datetime default NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `index_publishers_on_email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `publishers`
--

INSERT INTO `publishers` VALUES (1,'John
Stewart','john.stewart@gmail.com','2019-12-18 23:14:58','2019-12-18
23:14:58','42CFD85A880905FA8B680D0B53FDF0C1',NULL,NULL,NULL,NULL,
NULL), (2,'Peter Smith','peter.smith@spotify.com','2019-12-18
23:14:58','2019-12-18
23:14:58','2646F109C7263629DC7EC9FFF3662FFD',NULL,NULL,NULL,NULL,
NULL), (3,'Neil Diamond','neil.diamond@facebook.com','2019-12-18
23:16:57','2019-12-18
23:16:57','D2B58B63DDD255DC8C1041D47E01F57D',NULL,NULL,NULL,NULL,
NULL), (4,'Marc Andressen','marc.andressen@yahoo.com','2019-12-18
23:16:57','2019-12-18
23:16:57','FA0B17CEBB9EADA7AD58BF824474B99F',NULL,NULL,NULL,NULL,
NULL), (5,'Luc Bresson','luc.bresson@ebay.com','2019-12-18
```



```

23:19:37','2019-12-18
23:19:37','0BE083C2E2C9D04239483381EFA2471C',NULL,NULL,NULL,NULL,NULL,NULL
,NULL);

--
-- Table structure for table `jobcategories`
--

CREATE TABLE IF NOT EXISTS `jobcategories` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(255) default NULL,
  `description` varchar(255) default NULL,
  `image` varchar(255) default NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `jobcategories`
--

INSERT INTO `jobcategories` VALUES (1,'Full-time',NULL,NULL),(2,'Part-
time',NULL,NULL),(3,'Freelancer',NULL,NULL),(4,'Internship',NULL,NULL),
(5,'Temporary',NULL,NULL);

--
-- Table structure for table `jobs`
--

CREATE TABLE IF NOT EXISTS `jobs` (
  `id` int(11) NOT NULL auto_increment,
  `cat_id` int(11) NOT NULL,
  `publisher_id` int(11) NOT NULL,
  `title` varchar(255) default NULL,
  `company` varchar(255) default NULL,
  `created_at` datetime NOT NULL,
  `description` varchar(255) default NULL,
  `location` varchar(255) default NULL,
  `salary` int(11) default NULL,
  `image` varchar(255) default NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`cat_id`) REFERENCES `jobcategories` (`id`),
  CONSTRAINT FOREIGN KEY (`publisher_id`) REFERENCES `publishers` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `jobs`
--

INSERT INTO `jobs` VALUES (1,3,1,'Dropbox Product Designer','Software
startup','2019-12-18 23:25:53','Developing cutting-edge digital
solutions','Barcelona',50,'1.jpg'),(2,2,2,'Senior Programmer in
Laravel','Spotify ','2019-12-18 23:25:53','You will work alongside some of
the best developers','Faro',70,'2.jpg'),(3,1,3,'FullStack Developer in
Shopify','Facebook','2019-12-18 23:30:51','Do you look beyond the code and
back-end solution?','Brussels',80,'3.jpg'),(4,4,4,'Dropbox Product
Designer','Yahoo','2019-12-18 23:30:51','Back-end developer with expertise

```

```
in Java','Dublin',60,'4.jpg'),(5,5,5,'Front End Developer in  
Vue.js','Ebay','2019-12-18 23:33:15','Front-end developer who masters  
JavaScript, Angular, React, Vue','London',50,'5.jpg');
```

## ANEXO 2. Descrição dos métodos disponíveis na API de acesso à base de dados

Todos os dados recebidos da API ou enviados à API estão no formato JSON. Pode testar os métodos disponíveis com o comando "curl", por exemplo

```
curl -X GET 'http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/publishers.php?email=jbastos@ualg.pt'  
curl -X POST 'http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/publishers.php' -d '{"name":"Mary Stevens","email":"stevens@gmail.com", "password":"sWd356"}'
```

ou utilizando a extensão "THUNDER " do Visual Studio Code, ou utilizando o site <https://postman.com>

### API publishers.php

- **Verifica se um utilizador já se encontra registado na base de dados:**  
**GET** `http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/publishers.php?email=jbastos@ualg.pt`  
retorna: JSON string `{"email":"jbastos@ualg.pt"}` ou null
- **Registo de um utilizador:**  
**POST** `http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/publishers.php`  
Body: JSON string `{"name" : "Jose Bastos", "email": "jbastos@ualg.pt", "password": "segredo"}`  
retorna: JSON string `{"id":"1", "name": "Jose Bastos", "email": "jbastos@ualg.pt"}` ou null
- **Login de um utilizador:**  
**GET** `http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/publishers.php?email=jbastos@ualg.pt&password="pass"`  
retorna: JSON string `{"id":"1", "name": "Jose Bastos", "email": "jbastos@ualg.pt", "password_digest": "827ccb0eea8a..."}` ou null
- **Logout de um utilizador:**  
**GET** `http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/publishers.php?session_id=827ccb0eea8a...`  
retorna: JSON string `{"message": "Session destroyed"}` ou null

## API jobcategories.php

- **Todas as categorias:**

**GET** <http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/jobcategories.php>  
retorna: JSON string [{"id": "1", "name": "Full-time", "description": null, "image": null}, ...]

## API jobs.php

- **Todos os anuncios de emprego:**

**GET** <http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/jobs.php>  
retorna: JSON string [{"title": "FullStack Developer in Shopify", "id": "3", "image": "3.jpg", "location": "Brussels", "salary": "80000", "description": "Do you look beyond the code and back-end solution?", "company": "Facebook", "created\_at": "2019-12-18 23:30:51", "publisher": "Neil Diamond", "category": "Full-time"}, ...]

- **Apenas os anuncios de uma categoria:**

**GET** [http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/jobs.php?cat\\_id=2](http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/jobs.php?cat_id=2)  
retorna: JSON string [{"title": "Senior Programmer in Laravel", "id": "2", "image": "2.jpg", "location": "Faro", "salary": "70000", "description": "You will work alongside some of the best developers", "company": "Spotify", "created\_at": "2019-12-18 23:25:53", "publisher": "Peter Smith", "category": "Part-time"}, ...]

- **Registo de um novo anuncio de emprego:**

**POST** [http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/jobs.php?session\\_id=827ccb0eea8a...](http://daw.deei.fct.ualg.pt/~a12345/EXAME/api/jobs.php?session_id=827ccb0eea8a...)  
Body: JSON string {"company": "Google", "image": "1.jpg", "cat\_id": "1", "location": "California", "title": "Vue developer", "description": "seeking junior programmer", "salary": "10000"}  
retorna: JSON string {"title": "Vue developer", "id": "13", "image": "1.jpg", "location": "California", "salary": "10000", "description": "seeking junior programmer", "company": "Google", "created\_at": "2024-01-02 18:18:16", "publisher": "abx", "category": "Full-time"}