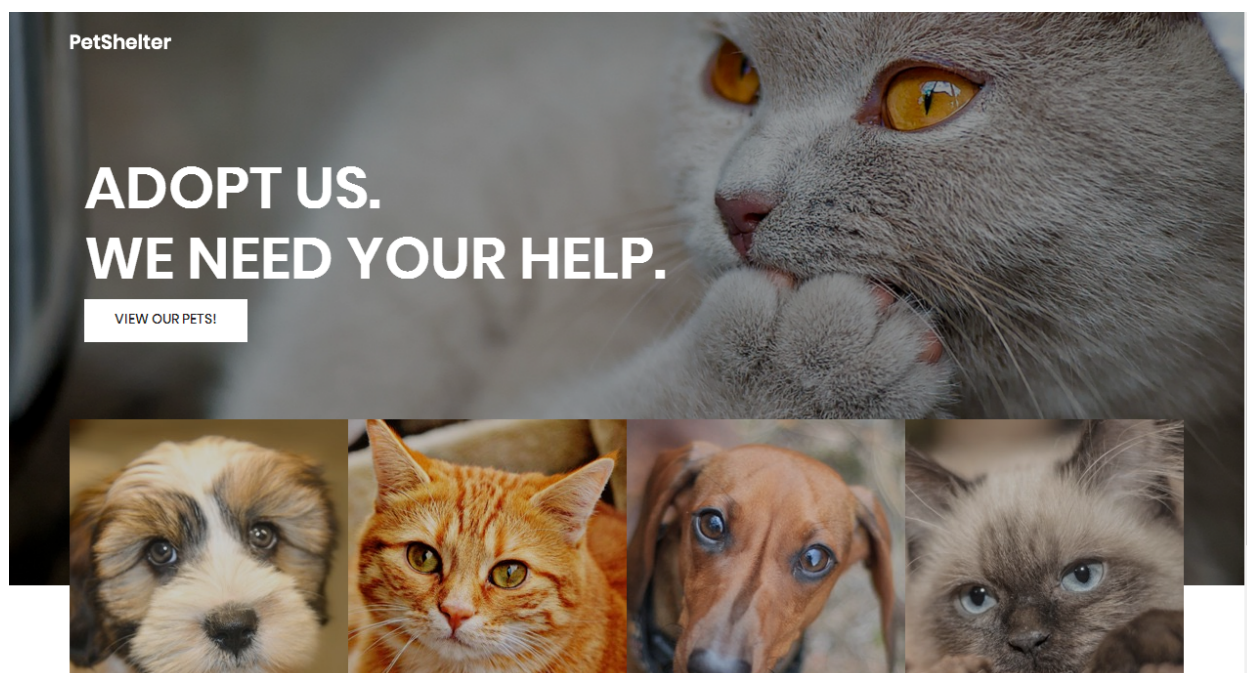


LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

---

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.



Pode ver uma implementação do site no URL

<http://daw.deei.fct.ualg.pt/~a555558/AnimalShelter.vue3/dist/>

O site web consiste no portal de uma instituição amiga dos animais que permite a utilizadores registados adoptar cães e gatos. O site tem as seguintes páginas:

1. **"Home"** é a página de rosto do site;
2. **"Pets"** é a página principal do site;
3. **"Register"** para registo de utilizadores;
4. **"Login"** para o login de utilizadores;
5. **"My Pets"** permite ao utilizador registado ver os animais que adotou;
6. **"Message"** é uma página auxiliar para mostrar mensagens (dado);

É dado acesso a um conjunto de templates HTML construídos em *Bootstrap* realizados pela empresa de web design colorlib.com.

Solicita-se ao aluno a realização do site web no *Framework VueJS 3*, apenas das páginas acima descritas.

Deve ser usado o seguinte mapeamento entre URLs e páginas:

```
const routes = [
  {
    path: '/',
    component: Home
  },
  {
    path: '/pets/:id?',
    component: Pets
  },
  {
    path: '/register',
    component: Register
  },
  {
    path: '/login',
    component: Login
  },
  {
    path: '/mypets',
    component: MyPets
  },
  {
    path: '/message',
    component: Message
  },
]
```

No ficheiro "zip" que é distribuído encontra um esboço das páginas requeridas.

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

## PRELIMINARES

- **VUEJS**

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

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

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

```
C:\XAMPP\htdocs
```

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

- **VUE STORE**

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

Nos ficheiros `"pets.js"`, `"mypets.js"`, `"adoptions.js"`, `"user.js"` **actualize** os URLs

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

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

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

Na pasta `C:\XAMPP\htdocs\PRACTICE\api` encontra o código de todas as APIs necessárias para acesso à base de dados, bem como encontra no ficheiro `index.html` uma descrição de todas as APIs

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

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

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

---

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

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

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

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

Faça login com o PuTTY no servidor daw2 (IP 10.10.23.184)<sup>2</sup>

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

### crie a pasta PRACTICE

```
a12345@daw2:~/public_html$ mkdir PRACTICE
```

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

- "api"

para dentro da pasta "PRACTICE" no servidor 10.10.23.184<sup>3</sup>

- **ATUALIZAÇÃO DA BASE DE DADOS**

**Atualize** a sua base de dados no servidor de produção 10.10.23.184 com o comando

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

---

<sup>2</sup> Se se encontra fora da Universidade utilize o serviço VPN, ou faça primeiro login com o PuTTY no servidor `ssh.deei.fct.ualg.pt` e em seguida faça `a12345@australia:~$ ssh 10.10.23.184`

<sup>3</sup> Se se encontra fora da Universidade utilize o serviço VPN, ou faça o upload para o servidor `ssh.deei.fct.ualg.pt`

Em alternativa, utilizando o software **phpMyAdmin** disponível no URL

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

importe a estrutura da base de dados necessária para o exame que se encontra no ficheiro `db.SQL` em

`C:\XAMPP\htdocs\PRACTICE\api\db.SQL`

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

Encontra no **ANEXO 2** a descrição das APIs de acesso à base de dados.

## TESTE A INSTALAÇÃO DAS API

A partir do seu browser preferido vá ao seguinte URL

<http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/pets.php>

(substitua "12345" pelo seu número de aluno)

Deverá receber uma página web com dados no formato JSON:

```
[{"cat_id": "1", "cat_name": "Dogs", "id": "1", "name": "Max", "description": "So full of life!", "status": "1", "image": "dg1.jpg"}, {"cat_id": "1", "cat_name": "Dogs", "id": "2", "name": "Buddy", "description": "I love company!", "status": "1", "image": "dg2.jpg"}, {"cat_id": "1", "cat_name": "Dogs", "id": "3", "name": "Charlie", "description": "I like biscuits!", "status": "1", "image": "dg3.jpg"}, {"cat_id": "1", "cat_name": "Dogs", "id": "4", "name": "Rocky", "description": "I love my boss!", "status": "0", "image": "dg4.jpg"}, {"cat_id": "1", "cat_name": "Dogs", "id": "5", "name": "Cooper", "description": "I am bad!", "status": "0", "image": "dg5.jpg"}, {"cat_id": "1", "cat_name": "Dogs", "id": "6", "name": "Bear", "description": "I love going to the beach!", "status": "0", "image": "dg6.jpg"}, {"cat_id": "1", "cat_name": "Dogs", "id": "7", "name": "Doggy", "description": "So sweet!", "status": "1", "image": "dg7.jpg"}, {"cat_id": "1", "cat_name": "Dogs", "id": "8", "name": "Duke", "description": "I am you best friend!", "status": "0", "image": "dg8.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "9", "name": "Kitty", "description": "So shy!", "status": "1", "image": "ct1.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "10", "name": "Lucky", "description": "I love to hunt mice!", "status": "1", "image": "ct2.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "11", "name": "Princess", "description": "I am very sweet!", "status": "1", "image": "ct3.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "12", "name": "Misty", "description": "I love to play!", "status": "0", "image": "ct4.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "13", "name": "Smokey", "description": "Miau!", "status": "0", "image": "ct5.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "14", "name": "Sassy", "description": "I am very curious!", "status": "0", "image": "ct6.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "15", "name": "Simba", "description": "I am bad!", "status": "0", "image": "ct7.jpg"}, {"cat_id": "2", "cat_name": "Cats", "id": "16", "name": "Tareco", "description": "I love sunny days!", "status": "0", "image": "ct8.jpg"}]
```

(um array de objetos com as propriedades dos cães e gatos ...)

## TESTE A INSTALAÇÃO DO FRAMEWORK

- Execute o comando

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

- Execute o comando

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

A partir do seu browser preferido vá ao seguinte URL

```
http://localhost:8080/~a12345/PRACTICE/dist/
```

Deverá receber uma página web de boas-vindas



**Welcome to Your PRACTICE Vue App**

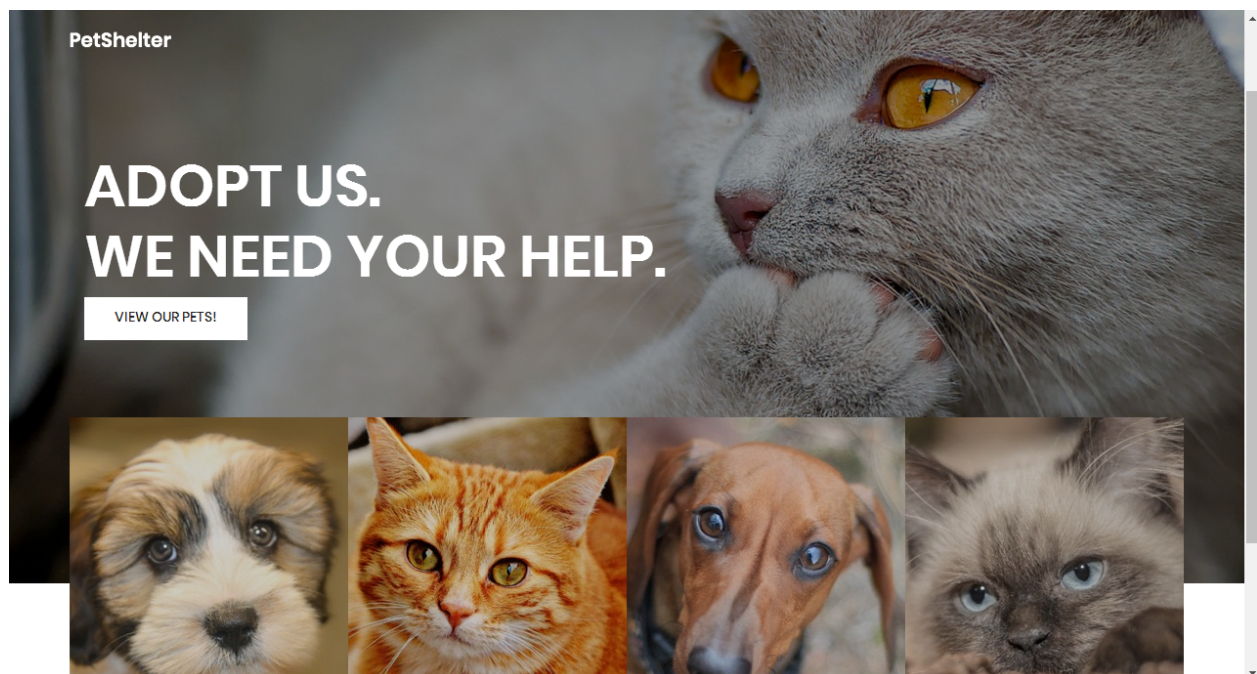
## FUNCIONALIDADE “HOME”

A funcionalidade “Home” é página de entrada do site.

### 1. [2.5 valores]

Adapte o template “index.html” fornecido pela empresa de web design.  
Encontra este template na pasta “demo” (C:\XAMPP\htdocs\PRACTICE\demo).  
Pode ver este template no browser no URL

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



Construa o template no ficheiro

C:\XAMPP\htdocs\PRACTICE\src\views\Home.vue

O botão "VIEW OUR PETS!" é um link para a página "pets"



## NOTA IMPORTANTE:

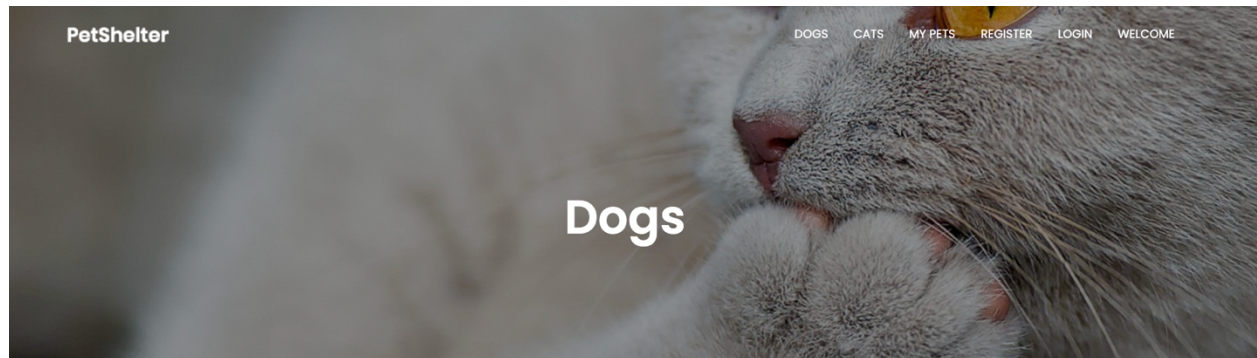
- **os hyperlinks** para os recursos estáticos css e javascript utilizados no template (`<link rel="stylesheet" href="...">` `<script src="...">`) **estão já colocados** no ficheiro "index.html" (**dado**)
- **os recursos estáticos** utilizados no template (img, css, fonts, js) **estão já colocados** 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`.

## FUNCIONALIDADE “PETS”

A funcionalidade “Pets” é página principal do site e contem uma lista dos animais disponíveis para adopção.

### 2. [7.5 valores]

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



Construa o template e o controlador no ficheiro

```
C:\XAMPP\htdocs\PRACTICE\src\views\Pets.vue
```

Sugere-se que o conteúdo "Header" e "Footer" deste template, que é comum aos outros templates, seja colocado nos componentes

```
C:\XAMPP\htdocs\PRACTICE\src\components\Header.vue (dado)
```

```
C:\XAMPP\htdocs\PRACTICE\src\components\Footer.vue (dado)
```

O controlador utiliza a acção

```
this.petsStore.getPetsDB()
```

para fazer o download de todos os animais existentes na base de dados

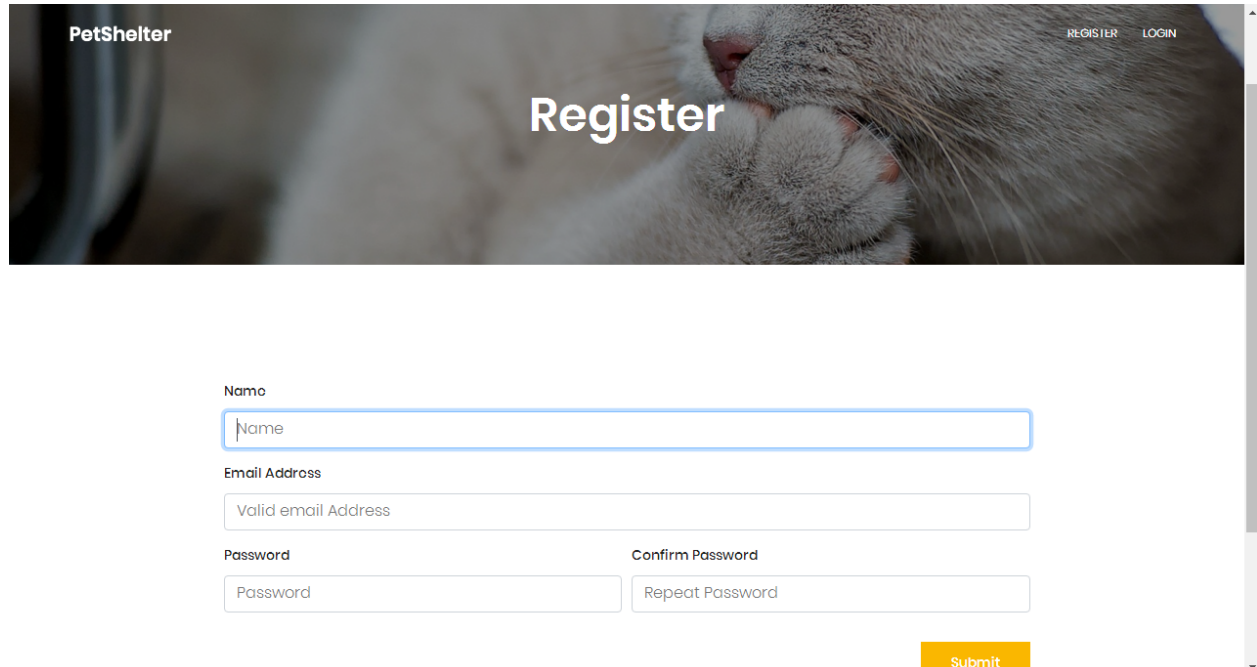
- No componente "Header", "REGISTER", "LOGIN" são links ("/register", "/login") activos caso o utilizador seja anonimo. Caso o utilizador tenha feito login, transformam-se em "LOGOUT", "MY PETS" ("/logout", "/mypets"), bem como aparece o texto "Welcome *user*" onde *user* é o login do utilizador
- O componente "Header" tem dois links "/pets/1" e "/pets/2" e idealmente existe uma só vista "/pets" comum aos "gatos" e aos "cães".
- Para cada animal mostre a imagem e o nome ("name", "image", existentes na tabela "pets")
- A vista APENAS mostra os animais que ainda não foram adotados ("status"=0)
- Caso o utilizador tenha feito login, na foto de cada animal deverá aparecer um link/botão que permite ao utilizador adotar o animal

## FUNCIONALIDADE “REGISTER”

A funcionalidade “Register” permite registar um utilizador do site.

### 3. [2 valores]

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



The screenshot shows a web registration form for 'PetShelter'. The page has a dark header with the site name 'PetShelter' on the left and 'REGISTER LOGIN' on the right. The main content area features a large, light-colored image of a cat's face with the word 'Register' overlaid in white. Below the image is a registration form with the following fields:

- Name:** A text input field with a light blue border and a placeholder text 'Name'.
- Email Address:** A text input field with a placeholder text 'Valid email Address'.
- Password:** A text input field with a placeholder text 'Password'.
- Confirm Password:** A text input field with a placeholder text 'Repeat Password'.

At the bottom right of the form is a yellow 'Submit' button.

O template deve ser colocado no ficheiro

C:\XAMPP\htdocs\PRACTICE\src\views\Register.vue

O controlador da acção valida os dados introduzidos.

Adicione o código necessário para:

- Em caso de insucesso
  - o formulário apresenta os campos correctos já preenchidos e o motivo do insucesso no registo
- Em caso de sucesso
  - o controlador utiliza as acções `this.userStore.userExistsDB(user)` para verificar se o utilizador já existe na base de dados e, se não existir,

`this.userStore.addUserDB()` para inserir o novo utilizador na base de dados

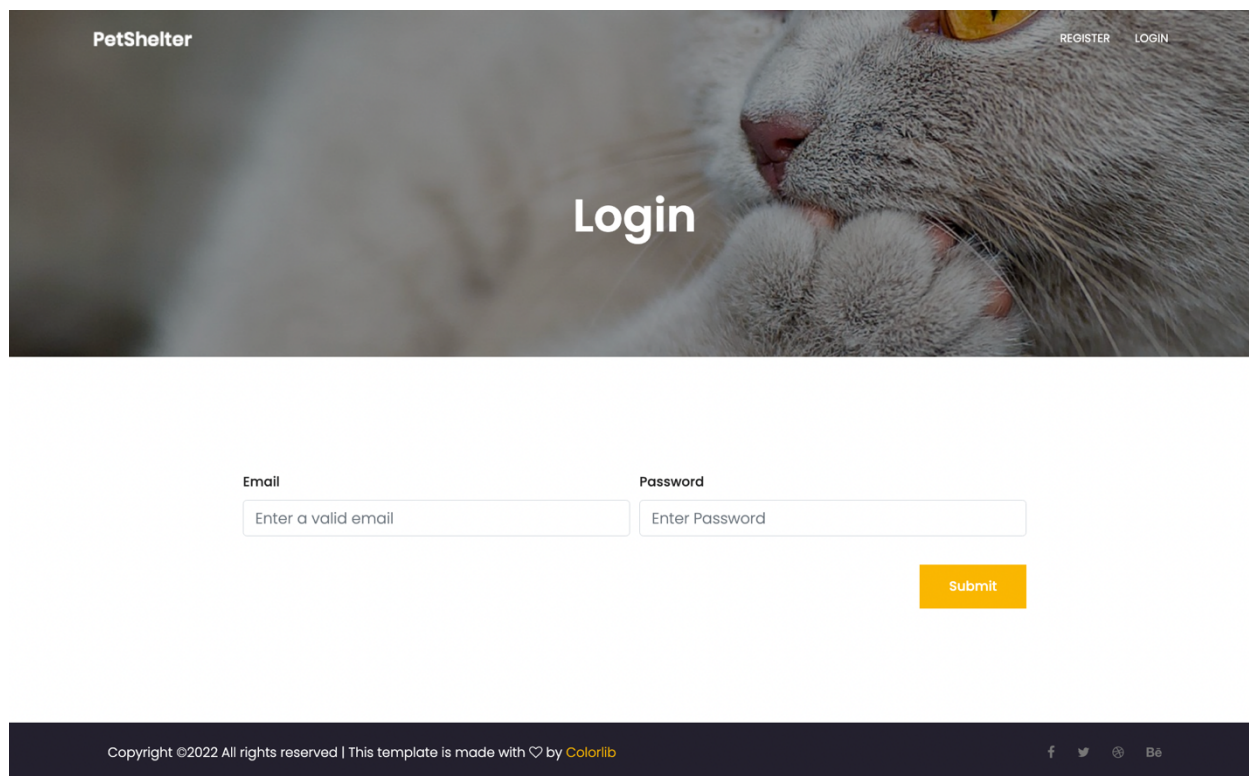
- o controlador redireciona para a "vista" `Message.vue`, que mostra a mensagem "Welcome! You can now login". A "vista" `Message.vue` redireciona automaticamente para a página de "pets" do site passados 2 segundos

## FUNCIONALIDADE “LOGIN” & “LOGOUT”

A funcionalidade “Login” permite autenticar um utilizador.

### 4. [3 valores]

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



O template e o controlador deverão encontrar-se em

```
C:\XAMPP\htdocs\PRACTICE\src\views>Login.vue
```

Adicione o código necessário para utilizar a "acção"

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

- Em caso de sucesso, guardar na "store" no objecto "user" o id, nome, e email do utilizador validado
- Em caso de sucesso, os links no componente "Header" devem mudar de "Login" "Register" para "My Pets" e "Logout".
- Em caso de sucesso deve aparecer um botão "Adopt Me!" na imagem do animal na vista "/pets"
- Em caso de sucesso utilizar a "vista" Message.vue, com a mensagem

```
`Welcome back ${this.userStore.user.name}!`
```

- Em caso de insucesso no formulário de login é apresentada a mensagem de erro "Login failed: wrong email or password".

O link "Logout" no componente "Header" utiliza a acção `this.userStore.logoutUser()` para fazer logout do utilizador

## FUNCIONALIDADE “My Pets”

A funcionalidade “My Pets” permite mostrar os pets que o utilizador adoptou até ao momento

### 5. [5 valores]

No ficheiro

```
C:\XAMPP\htdocs\PRACTICE\src\views\Pets.vue
```

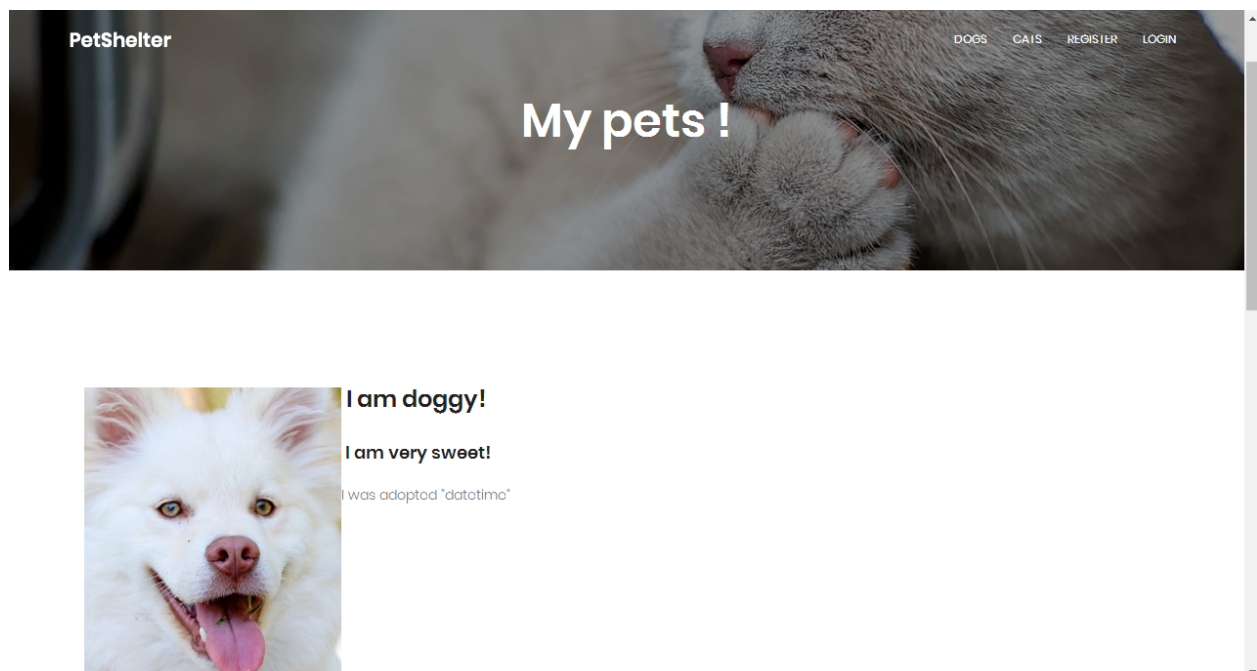
utilize a acção `this.adoptionsStore.newAdoptionDB(newAdoption)`

para guardar na base de dados a adopção do amigo dos animais, sempre que este carregar no link/botão "Adopt Me!" de um animal.

Utilize a "vista" `Message.vue`, com a mensagem "Success! Pet adopted!"

Adapte o template “`mypets.html`” fornecido pela empresa de web design.

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



Construa o template e o controlador no ficheiro



C:\XAMPP\htdocs\PRACTICE\src\views\MyPets.vue

O controlador utiliza a acção

```
this.adoptionsStore.getMyadoptionsDB(this.userStore.user.id)
```

para fazer o download da base de dados dos pets adoptados pelo amigo dos animais

- Para cada animal mostre a imagem, o nome, e a descrição (“name”, “image”, “description” existentes na tabela “pets”)

# COMPILAÇÃO/BUILD LOCALHOST OPCIONAL

**Atualize** o “publicPath” no ficheiro

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

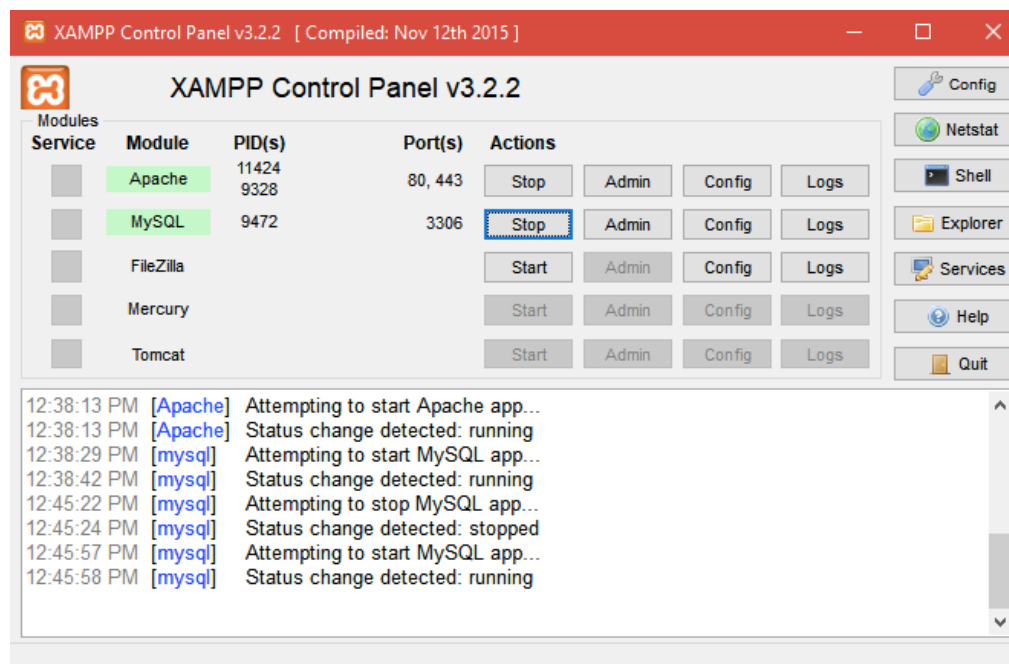
com o conteúdo

```
base: '/PRACTICE/dist/'
```

**Execute o comando**

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

Arranque o serviço XAMPP/Apache no seu portátil



**Teste** o funcionamento do site no URL

<http://localhost/PRACTICE/dist/>

# COMPILAÇÃO/BUILD PRODUÇÃO UPLOAD

**Atualize** o “publicPath” no ficheiro

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

com o conteúdo

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

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

**Execute o comando**

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

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

- "src"
- "dist"

para a pasta “PRACTICE” no seu site web pessoal no servidor de produção 10.10.23.184<sup>4</sup>

**Teste** o funcionamento do site no URL

<http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/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"!!**

---

<sup>4</sup> Se se encontra fora da Universidade utilize o serviço VPN, ou faça o upload para o servidor ssh.deei.fct.ualg.pt  
PRACTICE IPM 00.00.2024

# ANEXO 1. Estrutura da base de dados

A estrutura da base de dados pode ser criada com

- na shell do servidor de produção (IP 10.10.23.184) com o comando

```
a12345s@daw2:~$  
mysql -ua12345 -pPASS db_a12345 < ~/public_html/PRACTICE/api/db.SQL
```

- com o "tab" "IMPORT" em

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

onde db.SQL é o ficheiro que contem a estrutura da base de dados: (encontra este ficheiro em C:\XAMPP\htdocs\PRACTICE\api\db.SQL)

```
--  
-- Table structure for table `petlovers`  
--  
  
CREATE TABLE `petlovers` (  
  `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_users_on_email` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Table structure for table `petcategories`  
--
```

```

CREATE TABLE `petcategories` (
  `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 `petcategories`
--

INSERT INTO `petcategories` VALUES
(1, 'Dogs', NULL, NULL), (2, 'Cats', NULL, NULL);

--
-- Table structure for table `pets`
--

CREATE TABLE `pets` (
  `id` int(11) NOT NULL auto_increment,
  `cat_id` int(11) NOT NULL,
  `name` varchar(255) default NULL,
  `description` varchar(255) default NULL,
  `status` int(5) default NULL,
  `image` varchar(255) default NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`cat_id`) REFERENCES `petcategories`
(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `pets`
--

INSERT INTO `pets` VALUES (1,1,'Max','So full of
life!',0,'dg1.jpg'), (2,1,'Buddy','I love
company!',0,'dg2.jpg'), (3,1,'Charlie','I like
biscuits!',0,'dg3.jpg'), (4,1,'Rocky','I love my
boss!',0,'dg4.jpg'), (5,1,'Cooper','I am
bad!',0,'dg5.jpg'), (6,1,'Bear','I love going to the
beach!',0,'dg6.jpg'), (7,1,'Doggy','So
sweet!',0,'dg7.jpg'), (8,1,'Duke','I am you best
friend!',0,'dg8.jpg'), (9,2,'Kitty','So
shy!',0,'ct1.jpg'), (10,2,'Lucky','I love to hunt
mice!',0,'ct2.jpg'), (11,2,'Princess','I am very
sweet!',0,'ct3.jpg'), (12,2,'Misty','I love to

```

```
play!',0,'ct4.jpg'),(13,2,'Smokey','Miau!',0,'ct5.jpg'),(14,2,'S  
assy','I am very curious!',0,'ct6.jpg'),(15,2,'Simba','I am  
bad!',0,'ct7.jpg'),(16,2,'Tareco','I love sunny  
days!',0,'ct8.jpg');
```

```
--  
-- Table structure for table `adoptions`  
--
```

```
CREATE TABLE `adoptions` (  
  `id` int(11) NOT NULL auto_increment,  
  `petlover_id` int(11) NOT NULL,  
  `pet_id` int(11) NOT NULL,  
  `created_at` datetime NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uniq_key` (`petlover_id`, `pet_id`),  
  CONSTRAINT FOREIGN KEY (`petlover_id`) REFERENCES `petlovers`  
  (`id`),  
  CONSTRAINT FOREIGN KEY (`pet_id`) REFERENCES `pets` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

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

Na pasta C:\XAMPP\htdocs\PRACTICE\api pode consultar o código de todas as APIs utilizadas no acesso à base de dados.

Pode testar todos os métodos disponíveis com o comando "curl", por exemplo<sup>5</sup>

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

ou utilizando o site <https://postman.com>

### Tabela petlovers

- **Verifica se um amigo dos animais já se encontra registado na base de dados:**  
**GET** http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/petlovers.php?email=jbastos@ualg.pt  
retorna: JSON string {"id":"1","name":"Jose Bastos","email":"jbastos@ualg.pt"} ou null
- **Registo de um amigo dos animais:**  
**POST** http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/petlovers.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 amigo dos animais:**  
**GET** http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/petlovers.php?email=jbastos@ualg.pt&password="segredo"  
retorna: JSON string {"id":"1","name":"Jose Bastos","email":"jbastos@ualg.pt"} ou null

---

<sup>5</sup> NOTA: as APIs que utilizam o comando HTTP GET podem ser testadas diretamente no browser!

## Tabela petcategories

- **Todas as categorias:**  
**GET** <http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/petcategories.php>  
retorna: JSON string [{"id": "1", "name": "Dogs", "description": "", ...}, ...]

## Tabela pets

- **Todos os pets:**  
**GET** <http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/pets.php>  
retorna: JSON string [{"cat\_id": "1", "cat\_name": "Dogs", "id": "1", "name": "Max", "description": "So full of life", "image": "dg1.jpg", ...}, ...]

## Tabela adoptions

- **Todas as adoções do amigo dos animais:**  
**GET** [http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/adoptions.php?petlover\\_id=35](http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/adoptions.php?petlover_id=35)  
retorna: JSON string [{"id": "1", "created\_at": "2021-12-03 18:20:31", "pet\_name": "Max", "pet\_image": "dg1.jpg", ...}] ou null
- **Registo de uma nova adoção do amigo dos animais:**  
**POST** <http://daw.deei.fct.ualg.pt/~a12345/PRACTICE/api/adoptions.php>  
Body: JSON string {"petlover\_id": "19", "pet\_id": "3"}  
retorna: JSON string {"id": "5", "created\_at": "2021-12-03 18:20:31", "pet\_name": "Max", "pet\_image": "dg1.jpg"}