

LER O ENUNCIADO ATÉ AO FIM ANTES DE COMEÇAR!

O objectivo do exame é construir um site web 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 escola que permite a utilizadores registados inscreverem-se em cursos online. O site tem as seguintes páginas:

1. **"Home"** é a página de rosto do site;
2. **"Courses"** é a página principal do site;
3. **"Register"** para registo de utilizadores;
4. **"Login"** para o login de utilizadores;
5. **"My Courses"** mostra uma lista dos cursos a que o utilizador está inscrito;
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. É obrigatório em qualquer circunstância utilizar os templates dados.

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

Deve ser **UTILIZADO** o seguinte mapeamento entre URLs e páginas:

```
const routes = [  
  {  
    path: '/',  
    component: Home  
  },  
  {  
    path: '/courses',  
    component: Courses  
  },  
  
  {  
    path: '/register',  
    component: Register  
  },  
  {  
    path: '/message',  
    component: Message  
  },  
  {  
    path: '/login',  
    component: Login  
  },  
  {  
    path: '/mycourses',  
    component: MyCourses  
  },  
]
```

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 de acesso às APIs

## PRELIMINARES

- **VUEJS**

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

<https://github.com/jmatbastos/RECURSOVue2/archive/refs/heads/master.zip>

e expanda o ficheiro `RECURSOVue2-master.zip` para dentro pasta

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

- Se necessário mude o nome da pasta de `RECURSOVue2-master` para `RECURSOVue2`

- **VUE STORE**

Na pasta `C:\XAMPP\htdocs\RECURSOVue2\src\store` encontra o código de todas as acções necessárias para acesso AJAX à base de dados.

Nos ficheiros `"categories.js"`, `"courses.js"`, `"enrolls.js"`, `"user.js"` **actualize** os URLs

```
http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/...
```

e substitua "12345" pelo seu nº de aluno

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

Faça login com o PuTTY no servidor daw2 (IP 10.10.23.184)

```
a12345@daw2:~$ cd public_html
a12345@daw2:~/public_html$
git clone https://github.com/jmatbastos/RECURSOVue2.git
RECURSOVue2
```

Abra o ficheiro `RECURSOVue2/api/db.php`

---

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

```
C:\XAMPP\htdocs> git clone https://github.com/jmatbastos/RECURSOVue2.git
```

```
a12345@daw2:~/public_html$ cd RECURSOVue2/api
a12345@daw2:~/public_html/RECURSOVue2/api$ nano db.php
```

e atualize com as suas credenciais de acesso à base de dados

```
$hostname = "localhost";
$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...)

## • ATUALIZAÇÃO DA BASE DE DADOS

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

```
a12345@daw2:~/public_html/RECURSOVue2/api$
mysql -ua12345 -pPASS db_a12345 < ~/public_html/RECURSOVue2/api/db.SQL
```

Encontra no **APÊNDICE 1** a descrição da estrutura da base de dados, bem como um método alternativo de criação das tabelas.

## TESTE O FUNCIONAMENTO DA API E O ACESSO À BASE DE DADOS

A partir do seu browser preferido vá ao seguinte URL

<https://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/courses.php>

(substitua '12345' pelo seu nº de aluno)

Deverá receber dados no formato JSON:



```
{
  "cat_name": "Featured",
  "teacher_name": "Patty Smith",
  "teacher_image": "teacher_2.jpg",
  "id": "1",
  "name": "Literature Course",
  "description": "The best!",
  "price": "40",
  "image": "course_1.jpg",
  "sales": "345",
  "cat_name": "New",
  "teacher_name": "Julia Williams",
  "teacher_image": "teacher_5.jpg",
  "id": "2",
  "name": "Social Media Course",
  "description": "The best Social in the Internet!",
  "price": "35",
  "image": "course_2.jpg",
  "sales": "233",
  "cat_name": "Best Seller",
  "teacher_name": "Mary Kinget",
  "teacher_image": "teacher_4.jpg",
  "id": "3",
  "name": "Android Course",
  "description": "Best Mobile app development online!",
  "price": "50",
  "image": "course_3.jpg",
  "sales": "124",
  "cat_name": "Classic",
  "teacher_name": "Julia Williams",
  "teacher_image": "teacher_5.jpg",
  "id": "4",
  "name": "Literature Course",
  "description": "The best present day literature course!",
  "price": "30",
  "image": "course_4.jpg",
  "sales": "456",
  "cat_name": "Legacy",
  "teacher_name": "Julia Williams",
  "teacher_image": "teacher_5.jpg",
  "id": "5",
  "name": "Marketing Course",
  "description": "The latest economy theories!",
  "price": "60",
  "image": "course_5.jpg",
  "sales": "896",
  "cat_name": "Featured",
  "teacher_name": "Peter Frampton",
  "teacher_image": "teacher_3.jpg",
  "id": "6",
  "name": "Javascript Course",
  "description": "The most complete!",
  "price": "55",
  "image": "course_6.jpg",
  "sales": "753",
  "cat_name": "Best Seller",
  "teacher_name": "Julia Williams",
  "teacher_image": "teacher_5.jpg",
  "id": "7",
  "name": "Music Course",
  "description": "Let your inspiration be free!",
  "price": "20",
  "image": "course_7.jpg",
  "sales": "345",
  "cat_name": "Featured",
  "teacher_name": "Patty Smith",
  "teacher_image": "teacher_2.jpg",
  "id": "8",
  "name": "Vue.JS Course",
  "description": "Be an expert in frontend programming!",
  "price": "45",
  "image": "course_7.jpg",
  "sales": "245",
  "cat_name": "New",
  "teacher_name": "Paulo Nabo da Horta",
  "teacher_image": "teacher_6.jpg",
  "id": "9",
  "name": "Advanced Symphony",
  "description": "Simply the best!",
  "price": "60",
  "image": "course_8.jpg",
  "sales": "644"
}
```



## TESTE A INSTALAÇÃO DO FRAMEWORK

- Execute o comando

```
C:\XAMPP\htdocs\RECURSOVue2> npm install2
```

- Execute o comando

```
C:\XAMPP\htdocs\RECURSOVue2> npm run serve
```

A partir do seu browser preferido vá ao seguinte URL

**<http://localhost:8080/RECURSOVue2/dist/>**

Deverá receber uma página web de boas-vindas



Welcome to Your RECURSO Vue App

---

<sup>2</sup> tem de ter instalado node.js versão 16.18.0 ou 16.18.1  
Recurso IPM 30.01.2023

## FUNCIONALIDADE “HOME”

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

### 1. [3 valores]

Adapte o template “index.html” fornecido pela empresa de web design. Encontra este template na pasta “demo”. Pode ver este template em funcionamento no URL

`http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/demo/index.html`

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



Construa o template no ficheiro

`C:\XAMPP\htdocs\RECURSOVue2\src\views\Home.vue`

"SEE ALL COURSES" é um link para a página/vista "courses"

## NOTA IMPORTANTE:

- os hyperlinks para os recursos css e javascript utilizados no template (`<link rel="stylesheet" href="...">` `<script src="...">`) estão já colocados no ficheiro "public/index.html" (**dado**)
- os recursos estáticos utilizados no template (images, css, fonts, js, etc) estão já colocados em pastas (images, js, plugins, styles) dentro da pasta "public" (**dado**)



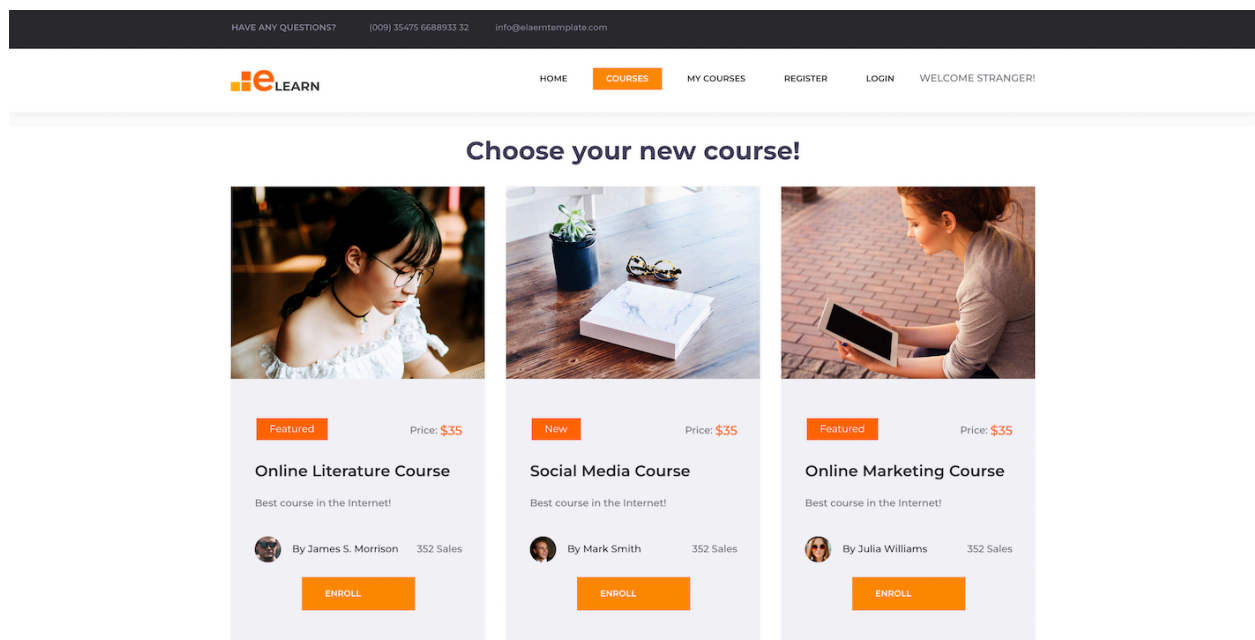
## FUNCIONALIDADE “COURSES”

A funcionalidade “Courses” é página principal do site e contem uma lista dos cursos online disponíveis na escola.

### 2. [7 valores]

Adapte o template “courses.html” fornecido pela empresa de web design. Encontra este template na pasta “demo”. (Pode ver este template em funcionamento no URL

<http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/demo/courses.html>)



Construa o template e o controlador no ficheiro

C:\XAMPP\htdocs\RECURSOVue2\src\views\Menu.vue

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

C:\XAMPP\htdocs\RECURSOVue2\src\components\Header.vue (dado parcial)

C:\XAMPP\htdocs\RECURSOVue2\src\components\Footer.vue (dado)

Recurso IPM 30.01.2023

O controlador utiliza a acção

```
this.$store.dispatch('courses/getCoursesFromDB')
```

para fazer o download de todos os cursos existentes na base de dados.

Deve aparecer a seguinte informação:

- nome do curso
- categoria
- imagem do curso
- preço
- descrição
- nome do instrutor
- imagem do instrutor
- total de vendas do curso
- botão "ENROLL" — apenas para utilizadores que fizeram login

## FUNCIONALIDADE “REGISTER”

A funcionalidade “Register” permite registar um utilizador.

### 3. [2 valores]

Adapte o template “register.html” fornecido pela empresa de web design. Encontra este template na pasta “demo”. Pode ver este template em funcionamento no URL <http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/demo/register.html>.

The screenshot shows a web registration form for 'eLEARN'. At the top, there is a dark header with contact information: 'HAVE ANY QUESTIONS?', '(009) 35475 6688933 32', and 'info@elaerntemplate.com'. Below the header is a navigation bar with 'HOME', 'COURSES', 'REGISTER' (highlighted in orange), and 'LOGIN'. The main content area contains a registration form with four input fields: 'Name:', 'Email:', 'Password:', and 'Repeat Password:'. Below these fields is an orange 'Register' button. At the bottom of the page, there is a light gray footer with the 'eLEARN' logo.

O template deve ser colocado no ficheiro

`C:\XAMPP\htdocs\RECURSOVue2\src\views\Register.vue`

O controlador do formulário 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.$store.dispatch('user/userExists', user)` para verificar se o utilizador já existe na base de dados e, se não existir,

`this.$store.dispatch('user/addUser')` para inserir o novo utilizador na base de dados

- o 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 "courses" 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”. Pode ver este template em funcionamento no URL <http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/demo/login.html>.

The screenshot shows a web application interface for a login page. At the top, there is a dark navigation bar with the text "eLEARN" on the left and "HOME", "COURSES", "REGISTER", and "LOGIN" on the right. The "LOGIN" button is highlighted in orange. Below the navigation bar is a light gray background area containing a login form. The form has two input fields: "Email:" and "Password:". Below the "Password:" field is an orange "Login" button. At the bottom of the page, there is a footer with the "eLEARN" logo.

O template e o controlador deverão encontrar-se em

```
C:\XAMPP\htdocs\RECURSOVue2\src\views>Login.vue
```

Adicione o código necessário para utilizar a "acção"

```
this.$store.dispatch('user/loginUser', 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 Orders" e "Logout".
- Em caso de sucesso deve aparecer um botão "ENROLL" na descrição do curso
- Em caso de sucesso utilizar a "vista" `Message.vue`, com a mensagem "Welcome back " + `this.$store.getters['user/getUser'].name` + "!"
- Em caso de insucesso no formulário de login é apresentada a mensagem de erro "Login failed: wrong email or password".

Utilize `this.$store.commit('user/logoutUser')`  
para fazer logout do utilizador

## FUNCIONALIDADE “My Courses”

A funcionalidade “My Courses” permite mostrar uma lista dos cursos em que o utilizador se inscreveu até ao momento

### 5. [5 valores]

No ficheiro

```
C:\XAMPP\htdocs\RECURSOVue2\src\views\Courses.vue
```

utilize a acção `this.$store.dispatch('enrolls/newEnroll', newEnroll)`

para guardar na base de dados o curso em que o utilizador se inscreve, sempre que este carregar no botão “ENROLL”. O objecto `newEnroll` tem duas propriedades: `user_id`, `course_id`.

Utilize a “vista” `Message.vue`, com a mensagem “Enroll success!”

Adapte o template “`myCourses.html`” fornecido pela empresa de web design. Encontra este template na pasta “demo”. Pode ver este template em funcionamento no URL <http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/demo/myCourses.html>.

HAVE ANY QUESTIONS? (009) 35475 6688933 32 info@elaerntemplate.com

**ie** LEARN HOME COURSES **MY COURSES** LOGOUT WELCOME STRANGER!

Enroll Date: April 02, 2018  
**Online Literature Course**  
By James S. Morrison

---

Enroll Date: April 02, 2018  
**Social Media Course**  
By Mark Smith

Construa o template e o controlador no ficheiro

```
C:\XAMPP\htdocs\RECURSOVue2\src\views\MyCourses.vue
```

O controlador utiliza a acção

```
this.$store.dispatch('enrolls/getMyEnrollsFromDB',  
this.$store.getters['user/getUser'].id)
```

para fazer o download da base de dados dos cursos em que o utilizador se inscreveu

Deve aparecer a seguinte informação:

- data da inscrição
- nome do curso
- nome do instrutor

# UPLOAD

**Atualize** o “publicPath” no ficheiro

C:\XAMPP\htdocs\RECURSOVue2\vue.config.js  
com o conteúdo

```
// vue.config.js
module.exports = {
  publicPath: '/~a12345/RECURSOVue2/dist/'
}
```

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

**Execute o comando**

```
C:\XAMPP\htdocs\RECURSOVue2> npm run build
```

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

- "src"
- "dist"

para a pasta “RECURSOVue2” no seu site web pessoal no servidor de produção

**Teste** o funcionamento do site no URL

<http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/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

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/RECURSOVue2/api/db.SQL
```

- ou 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\RECURSOVue2\api\db.SQL)

```
--  
-- Table structure for table `users`  
--  
  
CREATE TABLE `users` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(255) default NULL,  
  `email` varchar(255) NOT 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 (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Table structure for table `teachers`  
--
```

```

CREATE TABLE `teachers` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(255) default NULL,
  `email` varchar(255) NOT NULL,
  `image` varchar(255) default NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `teachers`
--

INSERT INTO `teachers` VALUES (1,'James S.
Morrison','James.Morrison@gmail.com','teacher_1.jpg'),(2,'Patty
Smith','Patty.Smith@gmail.com','teacher_2.jpg'),(3,'Peter
Frampton','Peter.Frampton@gmail.com','teacher_3.jpg'),(4,'Mary
Kinet','Mary.Kinet@berkeley.edu','teacher_4.jpg'),(5,'Julia
Williams','Julia.Williams@mit.edu','teacher_5.jpg'),(6,'Paulo
Nabo da Horta','Paulo.Horta@ualg.pt','teacher_6.jpg');

--
-- Table structure for table `coursecategories`
--

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

--
-- Dumping data for table `coursecategories`
--

INSERT INTO `coursecategories` VALUES
(1,'Featured',NULL),(2,'New',NULL),(3,'Best
Seller',NULL),(4,'Legacy',NULL),(5,'Classic',NULL);

--
-- Table structure for table `courses`
--

```

```

CREATE TABLE `courses` (
  `id` int(11) NOT NULL auto_increment,
  `cat_id` int(11) NOT NULL,
  `teacher_id` int(11) NOT NULL,
  `name` varchar(255) default NULL,
  `description` varchar(255) default NULL,
  `price` int(5) default NULL,
  `image` varchar(255) default NULL,
  `sales` int(5) default NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT FOREIGN KEY (`cat_id`) REFERENCES `coursecategories`
(`id`),
  CONSTRAINT FOREIGN KEY (`teacher_id`) REFERENCES `teachers`
(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

--
-- Dumping data for table `courses`
--

```

```

INSERT INTO `courses` VALUES (1,1,2,'Literature Course','The
best!',40,'course_1.jpg',345),(2,2,5,'Social Media Course','The
best Social in the
Internet!',35,'course_2.jpg',233),(3,3,4,'Android Course','Best
Mobile app development
online!',50,'course_3.jpg',124),(4,5,5,'Literature Course','The
best present day literature
course!',30,'course_4.jpg',456),(5,4,5,'Marketing Course','The
latest economy
theories!',60,'course_5.jpg',896),(6,1,3,'Javascript
Course','The most
complete!',55,'course_6.jpg',753),(7,3,5,'Music Course','Let
your inspiration be free!',20,'course_7.jpg',345),(8,1,2,'Vue.JS
Course','Be an expert in frontend
programming!',45,'course_7.jpg',245),(9,2,6,'Advanced
Symphony','Simply the best!',60,'course_8.jpg',644);

```

```

--
-- Table structure for table `enrolls`
--

```

```

CREATE TABLE `enrolls` (
  `id` int(11) NOT NULL AUTO_INCREMENT,

```

```
`user_id` int(11) NOT NULL,  
`course_id` int(11) NOT NULL,  
`enroll_date` datetime NOT NULL,  
PRIMARY KEY (`id`),  
CONSTRAINT FOREIGN KEY (`user_id`) REFERENCES `users` (`id`),  
CONSTRAINT FOREIGN KEY (`course_id`) REFERENCES `courses`  
(`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

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

### *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/RECURSOVue2/api/users.php?email=jbastos@ualg.pt'  
curl -X POST 'http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/users.php' -d '{"name":"Mary Stevens", "email":"stevens@gmail.com", "password":"sWd356"}'
```

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

### Tabela users

- **Verifica se um utilizador já se encontra registado na base de dados:**  
**GET** `http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/users.php?email=jbastos@ualg.pt`  
retorna: JSON string `{"id":"1","name":"Jose Bastos","email":"jbastos@ualg.pt"}` ou null
- **Registo de um utilizador:**  
**POST** `http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/users.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/RECURSOVue2/api/users.php?email=jbastos@ualg.pt&password="segredo"`  
retorna: JSON string `{"id":"1","name":"Jose Bastos","email":"jbastos@ualg.pt"}` ou null

## Tabela coursecategories

- **Todas as categorias:**

**GET** <http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/coursecategories.php>

retorna: JSON string [{"id": "1", "name": "Featured", "description": null}, {"id": "2", "name": "New", "description": null}, ...]

## Tabela courses

- **Todos os cursos:**

**GET** <http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/courses.php>

retorna: JSON string [{"cat\_name": "Featured", "teacher\_name": "Patty Smith", "teacher\_image": "teacher\_2.jpg", "id": "1", "name": "Literature Course", "description": "The best!", "price": "40", "image": "course\_1.jpg", "sales": "345"}, ...]

## Tabela enrolls

- **Todas as inscrições do utilizador:**

**GET** [http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/enrolls.php?user\\_id=1](http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/enrolls.php?user_id=1)

retorna: JSON string [{"course\_name": "Javascript Course", "teacher\_name": "Peter Frampton", "id": "3", "enroll\_date": "2022-12-18 20:38:19"}, ...] ou null

- **Registo de uma nova ordem do utilizador:**

**POST** <http://daw.deei.fct.ualg.pt/~a12345/RECURSOVue2/api/enrolls.php>

Body: JSON string {"user\_id": "1", "course\_id": "6"}

retorna: JSON string {"course\_name": "Javascript Course", "teacher\_name": "Peter Frampton", "id": "3", "enroll\_date": "2022-12-18 20:38:19"}