

# Ejercicio de Entrevista - Frontend

## Introducción

El propósito de este ejercicio es evaluar los conocimientos generales y algunos puntuales de la persona. El ejercicio presenta pequeños retos de diferentes niveles que nos sirven para esta evaluación.

La duración del ejercicio es de una hora.

## Nota

La duración reducida del ejercicio nos sirve para medir el desempeño de la persona bajo un tiempo limitado. La idea es que la práctica sea terminada en este tiempo, pero no es algo sumamente importante.

No requerimos diseño de ningún tipo. HTML5 puro y plano sin CSS es más que suficiente. Si termina la prueba antes de la hora y desea estilizar, lo puede hacer.

No obstante, si la persona no termina el ejercicio en el tiempo estipulado él mismo podrá terminarla desde su hogar con entrega límite de la medianoche del mismo día.

## Ejercicio

El ejercicio consta de un microsistema que tendrá:

1. Registro de usuario
2. Login de usuario
3. Listado del recurso de usuario

Para lograr esto se estará consumiendo un RESTful API.

**URL Base del API:** <https://apidev.kanvas.dev/v2>

## ¡Importante!

Este API utiliza la tecnología de [JWT](#) como medio de identificación del usuario una vez este se haya identificado con el sistema.

Toda llamada al API (exceptuando el registro de usuario) tiene que enviar en el encabezado de las peticiones el token (generado al momento del registro o login del usuario) de la siguiente forma:

```
Authorization : <token>
```

## API

Estos son los endpoints con los cuales la persona estará trabajando.

### Registro de usuario

```
POST /v1/users
```

```
{
  "firstname": "Nombre",
  "lastname": "Apellido",
  "email": "ejemplo@email.com",
  "password": "ejemplo",
  "verify_password": "ejemplo",
  "default_company": "Empresa Ejemplo"
}
```

```
RESPUESTA
```

```
{
  "user": {
    "firstname": "Nombre",
    "lastname": "Apellido",
    "email": "ejemplo@email.com",
    "password": "ejemplo",
    "verify_password": "ejemplo",
    "default_company": "Empresa Ejemplo"
  },
  "session": {
```

```
{
  "id": 23,
  "token": <token>,
  "time": "2019-07-31 15:57:26",
  "expires": "2019-08-31 15:57:26"
}
```

## Login de usuario

**POST** /v1/auth

```
{
  "email": "ejemplo@email.com",
  "password": "ejemplo"
}
```

RESPUESTA

```
{
  "id": 23,
  "token": <token>,
  "time": "2019-07-31 15:57:26",
  "expires": "2019-08-31 15:57:26"
}
```

## Listado de usuarios

**GET** /v1/users

RESPUESTA

```
[
  {
    "id": 23,
    "firstname": "Nombre A",
    "lastname": "Apellido A",
    "email": "ejemploa@email.com",
    "default_company": "Empresa Ejemplo A",
    "created_at": "2019-07-31 15:57:26",
    "updated_at": "2019-08-05 09:11:36"
  }
]
```

```
    },  
    {  
      "id": 24,  
      "firstname": "Nombre B",  
      "lastname": "Apellido B",  
      "email": "ejemploB@email.com",  
      "default_company": "Empresa Ejemplo B",  
      "created_at": "2019-08-02 12:43:56",  
      "updated_at": null  
    }  
  ]
```

## Funcionamiento

Luego de un usuario registrarse el mismo deberá ser marcado como identificado y debe ser enviado a la pantalla del listado de usuarios.

Si un usuario intenta acceder al listado de usuario sin estar identificado, se deberá redireccionar a la pantalla de login.

Cuando un usuario se identifica satisfactoriamente debe ser enviado a la pantalla del listado de usuario.

Deben mostrarse los mensajes de error que el API envía. Por ejemplo: falló en el intento de login.

## Conocimientos

Esperamos ver el uso de interceptors para la aplicación de los token de autorización. (Ya sea con Axios o Fetch).

Esperamos ver un código limpio, legible, bien estructurado y con buenas prácticas de código.