

Pamplona, 04 de marzo del 2022

PRUEBA PARA DESARROLLADOR VUE.JS

Esta prueba tiene como objetivo valorar el nivel de conocimiento que posee los candidatos que apliquen a la misma, por lo que se evaluarán 3 grandes niveles del desarrollo frontend, los cuáles son:

Nivel #1 – Maquetación con Tailwind CSS o Bootstrap.

Nivel #2 – Implementación del framework de Vue.js en su versión 3 u otro framework.

Nivel #3 – Consumo de API desde Vue.js en su versión 3 o con otro framework.

No importa si haces un solo nivel, ya que el resto lo puedes aprender haciendo carrera profesional con nosotros, siempre y cuando seas aceptado.

INICIO DE LA PRUEBA

Nivel #1 – Maquetación con Tailwind CSS o Bootstrap:

Se da un mayor valor al hacer el proceso con Tailwind CSS, sino cuenta con el conocimiento se puede realizar con Bootstrap, ya que el proceso de transición a Tailwind se podría vivir internamente en la empresa, en caso de que sean elegidos.

En caso de tener la capacidad de entregar un único proyecto corriendo con Vue.js 3, con Tailwind CSS o Bootstrap, incorporado y funcionando, más los siguientes paquetes:

1. Axios.
2. Tailwind CSS versión 3.
3. Vue Select 2.
4. Sweetalert 2.
5. Vue-Axios versión 3.4.0.
6. Vue-Router versión 4.0.12.
7. Vuex Versión 4.

Si no se cuenta con dicha capacidad, se puede implementar Tailwind CSS o Bootstrap, sin Vue.js para poder demostrar que si se tiene el conocimiento del nivel 1.

Realizar la maquetación del inicio de sesión y el registro aplicando las mejores prácticas del frontend, tanto para smartphone como desktop, que demuestre el uso correcto de las herramientas que se tengan a disposición, tratando de rehusar el código lo máximo posible.

Link del figma abierto para su correcta visualización, recuerda registrarte e iniciar sesión para poder usarlo bien:

[https://www.figma.com/file/xgfAwlduvW8nqYoJL6AUq/Sign-Up-Page---Demon-Slayer-\(Community\)?node-id=0%3A1](https://www.figma.com/file/xgfAwlduvW8nqYoJL6AUq/Sign-Up-Page---Demon-Slayer-(Community)?node-id=0%3A1)

Allí encontrarás los diseños de inicio de sesión y registro, tanto en la versión desktop como smartphone, además muestra como la versión desktop tiene variación de imagen y texto por ser un carrusel, el cual se debe implementar a gusto del candidato.

Se sugiere usar **@apply** si se hizo la implementación con Tailwind CSS.

Nivel #2 - Implementación del framework de Vue.js en su versión 3 u otro framework:

Se da un mayor valor al hacer el proceso con Vue.js, sino cuenta con el conocimiento se puede realizar con otros frameworks, ya que el proceso de transición a Vue.js se podría vivir internamente en la empresa, en caso de que sean elegidos.

Requerimos que se haga la implementación de un sistema de autenticación de forma modular, en caso de estar usando Vue.js, emplear Vuex (se debe crear una carpeta de nombre Modulo que contenga los archivos js para el inicio de sesión, registro y cierre de sesión).

Luego de que el inicio de sesión o registro sea exitoso deberá redirigir a un componente de backoffice (Lista de categorías), tener en cuenta la protección de esa ruta. Si no ha iniciado sesión, debe evitar accesos por seguridad.

Nota: (Si deseo ingresar sin iniciar sesión me debe redirigir al componente iniciar sesión).

Las notificaciones deben mostrarse empleando sweetalert2, para mejor experiencia de usuario, los mensajes (error o éxito) deben desaparecer 20s después de haberlo mostrado.

En caso de no contar con conocimientos del paquete sweetalert2, puedes implementar los anteriores puntos de forma nativa o con librerías de js que conozcas, pero valoraremos más todo lo que si se haga con Vue.js.

Nivel #3 – Consumo de API desde Vue.js en su versión 3 o con otro framework:

Deberá consumir la API con AXIOS para inicio de sesión y registro, con los datos que se suministran.

En caso de no dominar AXIOS, puede implementar otras tecnologías de consumo de API, si no se implementó Vue.js para la prueba, adecuarlo al framework elegido.

Inicio de sesión

URL

<https://texxt.tk/api/v1/login>

HEADER

KEY	VALUE
Accept	application/json

BODY

Seleccione: raw

Tipo: JSON

CONTENIDO


```
{
  "email": "randommail2@mail.com",
  "password": "strongkey1232",
  "apiKey": "RtYujtghf4534TThglUY6556tyuGHJGHHGH",
  "utcTimeStamp": "2022-01-20T09:39:38Z",
  "signature": "756403b52606111ee553e75e927bc0d92cc376d2aa63d469ee6d851e2cc04e9a"
}
```

Donde la SIGNATURE se compone por:

PUBLIC_KEY: VBNfgfTYrt5666FGHFG6FGH65GHFGHF656g

PRIVATE KEY: DGDFGDbnbTRTEfg67hgyTYRTY56gfhdR6

HORA ACTUAL: En formato "Y-m-d\TH:i:s\Z"

Siguiendo el parámetro:

Signature = PRIVATE KEY + “,” + PUBLIC KEY + “,” + HORA ACTUAL

Luego la deberá Hashear con formato sha256:

```
SignatureHash = hash("sha256", signature)
```

Nota: SignatureHash es un ejemplo de hasheado en php deberá buscar el equivalente al lenguaje utilizado.

- Donde la APIKEY equivale a **PUBLIC_KEY**
- Donde la utcTimeStamp equivale a la **HORA_ACTUAL** con la cual creo la signatura.
- Obtendrá como respuesta un **TOKEN**, el cual deberá usar para consumir las demás APIS

RESPONSE

```
{
  "token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvdGV4eHQudGtCL2FwaVwvdjFCL2xvZ2luliwiaWF0IjoxNjQ2MzQ5NjI0LCJleHAiOiJlE2NDYzNTMyMjQsIm5iZiI6MTY0NjM0OTYyNCwianRpljoiUjF3THNWMFNyY0pxenRhSCIsInN1Yil6MiwiwCj2IjoiMjNiZDVjODk0OWY2MDBhZGlzOWU3MDFjNDAwODcyZGI3YTU5NzZmNyJ9.G5xLiXILLYhx1MNCaFOF2u3NUgrJqY1kZbDsZtYf4JM"
}
```


BODY

Seleccione: raw

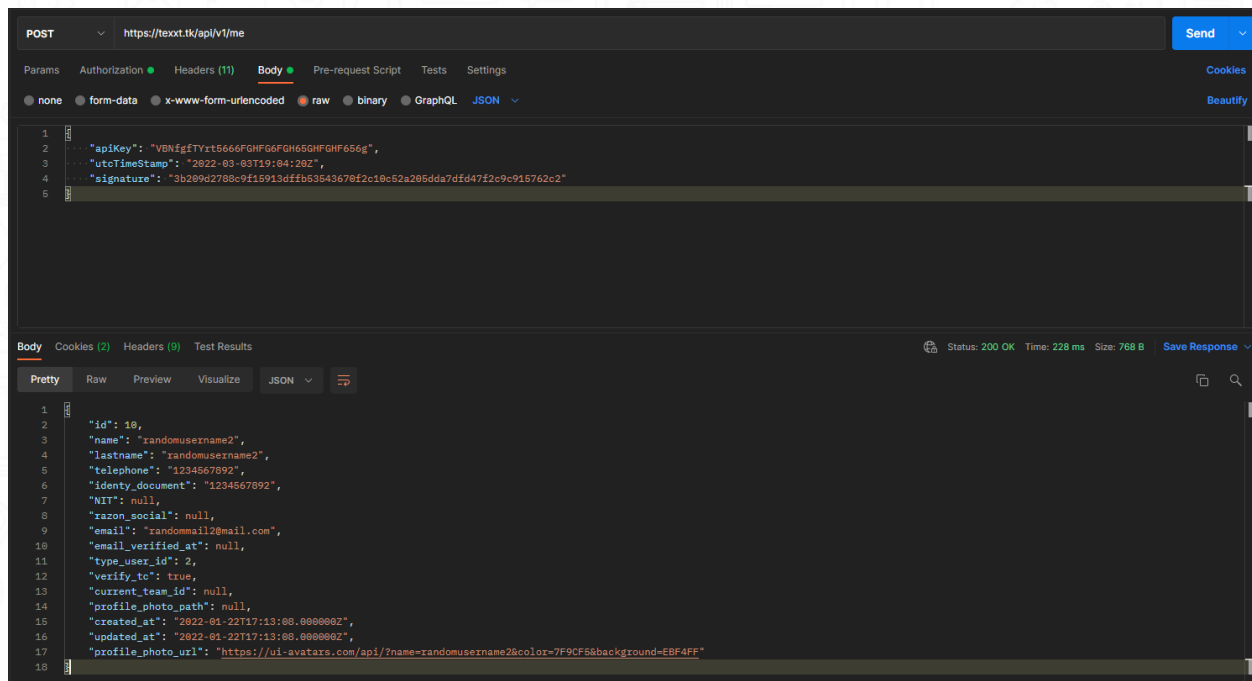
Tipo: JSON

CONTENIDO

```
{  
  "apiKey": "RtYujtghf4534TThglUY6556tyuGHJGHHGH",  
  "utcTimeStamp": "2022-01-20T09:39:38Z",  
  "signature": "756403b52606111ee553e75e927bc0d92cc376d2aa63d469ee6d851e2cc04e9a"  
}
```

RESPONSE

```
{  
  "id": 10,  
  "name": "randomusername2",  
  "lastname": "randomusername2",  
  "telephone": "1234567892",  
  "identity_document": "1234567892",  
  "NIT": null,  
  "razon_social": null,  
  "email": "randommail2@mail.com",  
  "email_verified_at": null,  
  "type_user_id": 2,  
  "verify_tc": true,  
  "current_team_id": null,  
  "profile_photo_path": null,  
  "created_at": "2022-01-22T17:13:08.000000Z",  
  "updated_at": "2022-01-22T17:13:08.000000Z",  
  "profile_photo_url": "https://ui-avatars.com/api/?name=randomusername2  
    &color=7F9CF5&background=EBF4FF"  
}
```



Registro PERSONA JURIDICA

URL

<https://texxt.tk/api/v1/register>

HEADER

KEY	VALUE
Accept	application/json

BODY

Seleccione: raw

Tipo: JSON

CONTENIDO

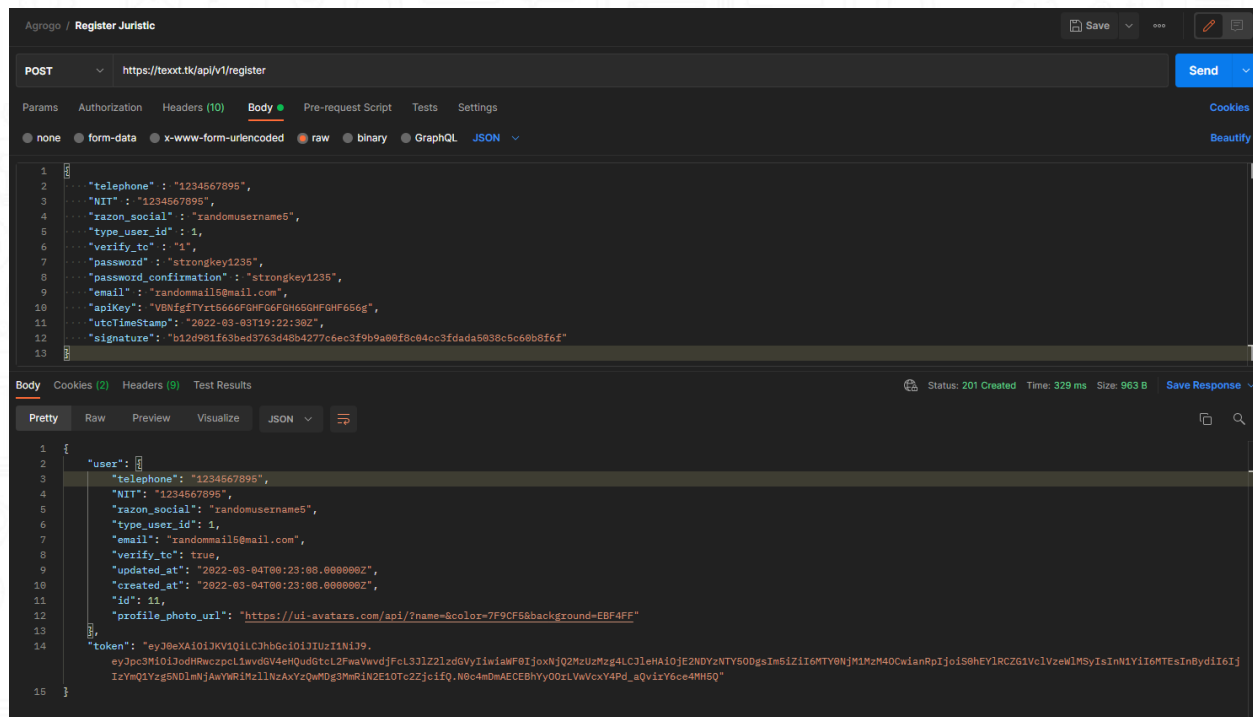
```
{
  "telephone": "1234567895",
  "NIT": "1234567895",
  "razon_social": "randomusername5",
  "type_user_id": 1, //Aquí se indica el tipo de persona
  "verify_tc": "1",
  "password": "strongkey1235",
  "password_confirmation": "strongkey1235",
```



```
"email": "randommail5@mail.com",  
"apiKey": "VBNfgfTYrt5666FGHFG6FGH65GHFGHF656g",  
"utcTimeStamp": "2022-03-03T19:22:30Z",  
"signature": "b12d981f63bed3763d48b4277c6ec3f9b9a00f8c04cc3fdada5038c5c60b8f6f"  
}
```

RESPONSE

```
{  
  "user": {  
    "telephone": "1234567895",  
    "NIT": "1234567895",  
    "razon_social": "randomusername5",  
    "type_user_id": 1,  
    "email": "randommail5@mail.com",  
    "verify_tc": true,  
    "updated_at": "2022-03-04T00:23:08.000000Z",  
    "created_at": "2022-03-04T00:23:08.000000Z",  
    "id": 11,  
    "profile_photo_url": "https://ui-avatars.com/api/?name=&color=7F9CF5&background=EBF4FF"  
  },  
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpCL1wvdGV4eH  
    QudGtCL2FwaVwvdjFcl3JlZ2ZldGVyYiwiWF0ljoXNjQ2MzUzMzg4LCJleHAiOiJlE2ND  
    YzNTY5ODgslm5iZiI6MTY0NjM1MzM4OCwianRpljoiS0hEYlRCZG1VclVzeWlMSyls  
    InN1Yil6MTEsInBydiI6IjIyYmQ1Yzg5NDImNjAwYWVWRiMzIlInAxYzQwMDg3MmRiN  
    2E1OTc2ZjcfQ.N0c4mDmAEECEBhYyOOLVwVcxY4Pd_aQvirY6ce4MH5Q"  
}
```

Registro PERSONA NATURAL

URL

<https://texxt.tk/api/v1/register>

HEADER

KEY	VALUE
Accept	application/json

BODY

Seleccione: raw

Tipo: JSON

CONTENIDO

```
{
  "name": "randomusername12",
  "lastname": "randomusername12",
  "telephone": "12345678932",
  "identity_document": "12345678921",
  "type_user_id": "2",
  "verify_tc": "1",
  "password": "strongkey12321",
  "password_confirmation": "strongkey12321",
```

```
"email": "randommail21@mail.com",  
"apiKey": "VBNfgfTYrt5666FGHFG6FGH65GHFGHF656g",  
"utcTimeStamp": "2022-03-03T19:43:50Z",  
"signature": "b20cf1f851072d00ca9b99eb00c495f71eb74ef63db3229e58c46401ded3aec8"
```

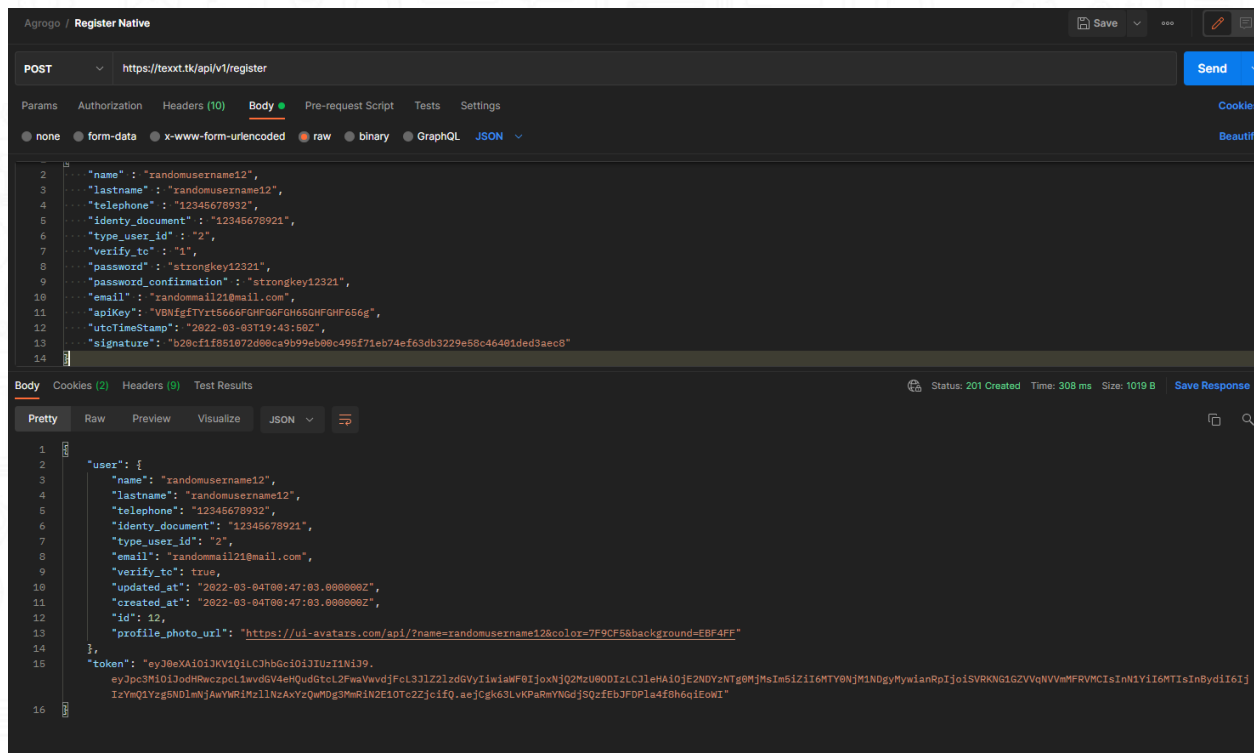
```
}
```

RESPONSE

```
{
```

```
  "user": {  
    "name": "randomusername12",  
    "lastname": "randomusername12",  
    "telephone": "12345678932",  
    "identity_document": "12345678921",  
    "type_user_id": "2",  
    "email": "randommail21@mail.com",  
    "verify_tc": true,  
    "updated_at": "2022-03-04T00:47:03.000000Z",  
    "created_at": "2022-03-04T00:47:03.000000Z",  
    "id": 12,  
    "profile_photo_url": "https://..."  
  },  
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvdGV4eH  
    QudGtcl2FwaVwvdjFcl3JlZ2lzdGVyIiwiaWF0IjoxNjQ2MzUzMzg4LCJleHAiOiJlE2ND  
    YzNTY5ODgsIm5iZiI6MTY0NjM1MzM4OCwianRpljoiS0hEYlRCZG1VclVzeWlMSyls  
    InN1Yil6MTEsInBydiI6IjIyYmQ1Yzg5NDImNjAwYWRiMzllInAxYzQwMDg3MmRiN  
    2E1OTc2ZjcfQ.N0c4mDmAEECBhYyOORLVwVcxY4Pd_aQvirY6ce4MH5Q"
```

```
}
```



Categoría: En el backoffice para implementar la lista de categorías.

URL

<https://texxt.tk/api/v1/categoria>

HEADER

KEY	VALUE
Accept	application/json

AUTHORIZATION

KEY	VALUE
Bearer Token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpCL1wvdGV4eHQudGtCL2FwaVwvdjFCL2xvZ2lulwIWF0IjoxNjQ2MzQ5NjI0LCJleHAiOiJlY2NDYzNTMyMjQ5NjI0MTY0NjM0OTYyNiwianRpIjoifjF3THNWMTFyY0pxenRhSCIsbnN1Yil6MiwiwiczHJ2ljoimjNiZDVjODk0OWY2MDBhZGZlOWU3MDFjNDAwODcyZGI3YTU5NzZmNyJ9.G5xLiXILLYhx1MNCaFOF2u3NUgrJqY1kZbDsZtYf4JM

BODY

Selección: raw

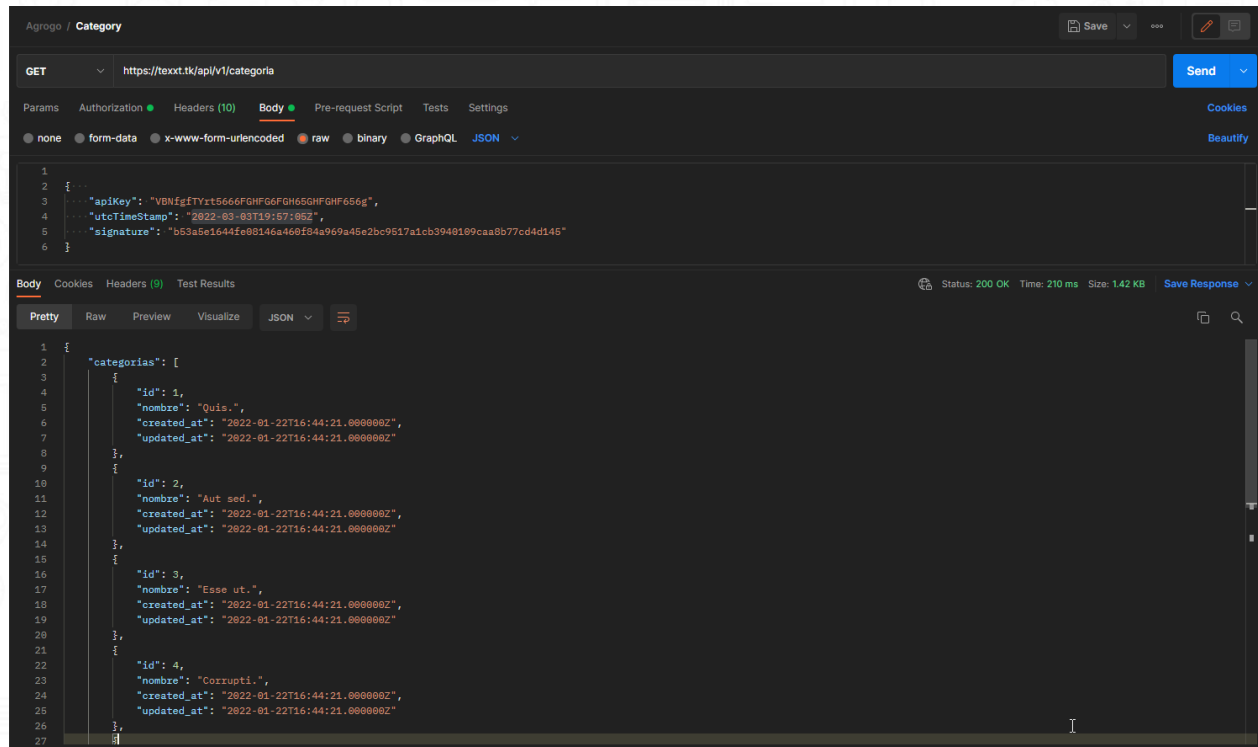
Tipo: JSON

CONTENIDO

```
{
  "apiKey": "RtYujtghf4534TThglUY6556tyuGHJGHHGH",
  "utcTimeStamp": "2022-01-20T09:39:38Z",
  "signature": "756403b52606111ee553e75e927bc0d92cc376d2aa63d469ee6d851e2cc04e9a"
}
```

RESPONSE

```
{
  "categorias": [
    {
      "id": 1,
      "nombre": "Quis.",
      "created_at": "2022-01-22T16:44:21.000000Z",
      "updated_at": "2022-01-22T16:44:21.000000Z"
    },
    {
      "id": 2,
      "nombre": "Aut sed.",
      "created_at": "2022-01-22T16:44:21.000000Z",
      "updated_at": "2022-01-22T16:44:21.000000Z"
    },
    {
      "id": 3,
      "nombre": "Esse ut.",
      "created_at": "2022-01-22T16:44:21.000000Z",
      "updated_at": "2022-01-22T16:44:21.000000Z"
    }
  ]
}
```

Categoría - CREAR: En el backoffice para implementar el crear categoría.

URL

<https://texxt.tk/api/v1/categoria/create>

HEADER

KEY	VALUE
Accept	application/json

AUTHORIZATION

KEY	VALUE
Bearer Token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvdGV4eHQudGtcL2FwaVwdjFcL2xvZ2luliwiaWF0IjoNjQzMzQ5NjI0LCJleHAiOjE2NDYzNTMyMjQsIm5iZil6MTY0NjM0OTYyNCwianRpljoiUjF3THNWMFNyY0pxenRhSCIsInN1Yil6MiwicHJ2IjoiaWJNiZDVjODk0OWY2MDBhZGlzOWU3MDFjNDAwODcyZGI3YTU5NzZmNyJ9.G5xLiXILLYhx1MNcCaFOF2u3NUgrJqY1kZbDsZtYf4JM

BODY

Seleccione: raw

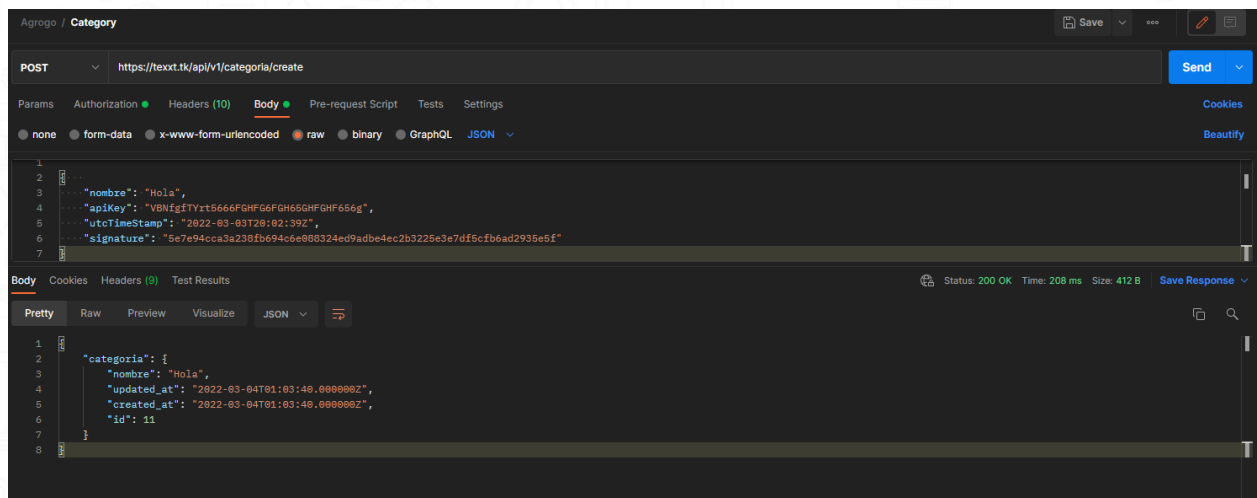
Tipo: JSON

CONTENIDO

```
{
  "nombre": "Hola",
  "apiKey": "RtYujtghf4534TThglUY6556tyuGHJGHHGH",
  "utcTimeStamp": "2022-01-20T09:39:38Z",
  "signature": "756403b52606111ee553e75e927bc0d92cc376d2aa63d469ee6d851e2cc04e9a"
}
```

RESPONSE

```
{
  "categoria": {
    "nombre": "Hola",
    "updated_at": "2022-03-04T01:03:40.000000Z",
    "created_at": "2022-03-04T01:03:40.000000Z",
    "id": 11
  }
}
```



Categoría - ACTUALIZAR: En el backoffice para implementar el actualizar categoría.

URL

<https://texxt.tk/api/v1/categoria/update/{id}>

HEADER

KEY	VALUE
Accept	application/json

AUTHORIZATION

KEY	VALUE
Bearer Token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvdGV4eHQudGtcL2FwaVwvdjFcl2xvZ2luliwiaWF0IjoxNjQ2MzQ5NjI0LCJleHAiOjE2NDYzNTMyMjQsIm5iZil6MTY0NjM0OTYyNCwianRpljoiUjF3THNWMFNyY0pxenRhSClInN1Yil6MiwicHJ2IjoiaWJNiZDVjODk0OWY2MDBhZGZlOWU3MDFjNDAwODcyZGI3YTU5NzZmNyJ9.G5xLiXILLYhx1MNCaFOF2u3NUgrJqY1kZbDsZtYf4JM

BODY

Selezione: raw

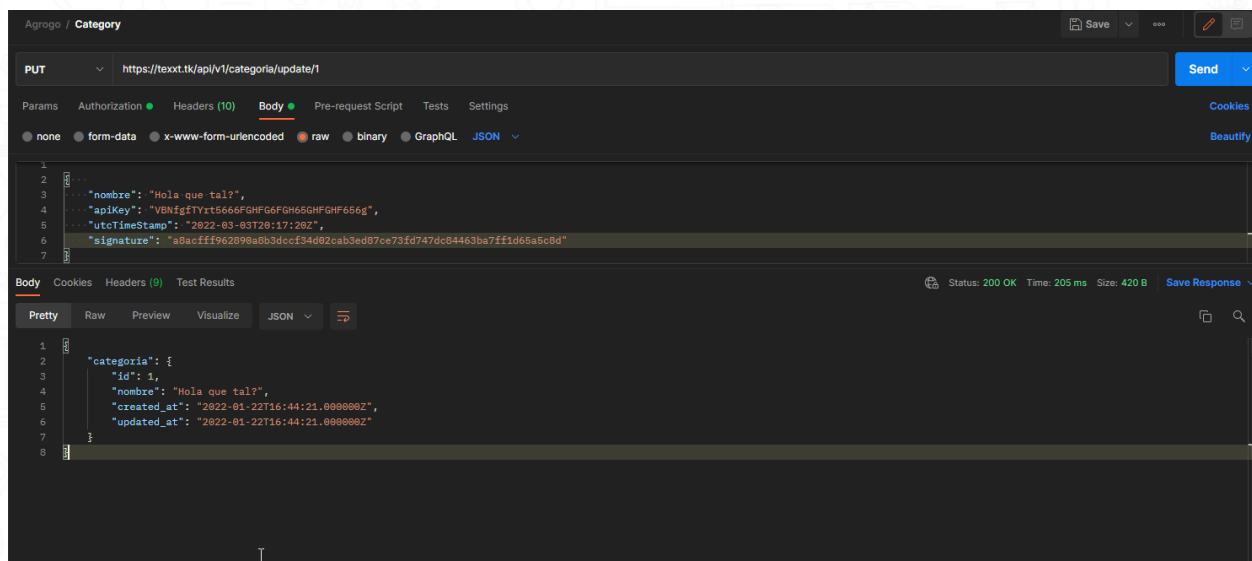
Tipo: JSON

CONTENIDO

```
{
  "nombre": "Hola que tal?",
  "apiKey": "RtYujtghf4534TThglUY6556tyuGHJGHHGH",
  "utcTimeStamp": "2022-01-20T09:39:38Z",
  "signature": "756403b52606111ee553e75e927bc0d92cc376d2aa63d469ee6d851e2cc04e9a"
}
```

RESPONSE

```
{
  "categoria": {
    "id": 1,
    "nombre": " Hola que tal?",
    "created_at": "2022-01-22T16:44:21.000000Z",
    "updated_at": "2022-01-22T16:44:21.000000Z"
  }
}
```



Se debe crear una variable de entorno, la cual se encargará de gestionar los datos como: nombre de la aplicación, url API. Esto con el fin de tener un único archivo desde el cual manipular los datos globales del sistema. El valor de la variable debe enviarse como dato adjunto a los datos de inicio de sesión y registro.

Al implementar axios en las peticiones, debe llamarse la variable de entorno seguido de la ruta API en las peticiones https, ejemplo:

process.env.vue_app_url_api_backend/login

Esto con el fin de poder cambiar la url del API cada vez que se necesite para toda la aplicación, sin necesidad de cambiarla en todas las vistas.

Evitar llamar las librerías en cada componente de lo posible llamarlas desde un archivo global. Ejemplo: (main.js).

El token de inicio de sesión debe almacenarse en Local Storage, y al actualizar la página, el token no debe desaparecer, debe ser accesible desde cualquier componente para envío de datos.

Al expirar el token en la aplicación debe redirigir al inicio de sesión, para regenerarse y almacenarse un nuevo token al iniciar sesión en el Local Storage.

En el dashboard debe aparecer el botón cerrar sesión, que eliminara los datos del Local Storage y del token en la API.

Buenas prácticas a tomar en cuenta:

- Los registros de formularios y visualización de los datos deben ser reactivos, usando Vuex.
- Evitar en lo posible usar pros y emit en Vue.js. En caso de ser necesario la comunicación entre componentes, usar composición de API de Vue.js 3.