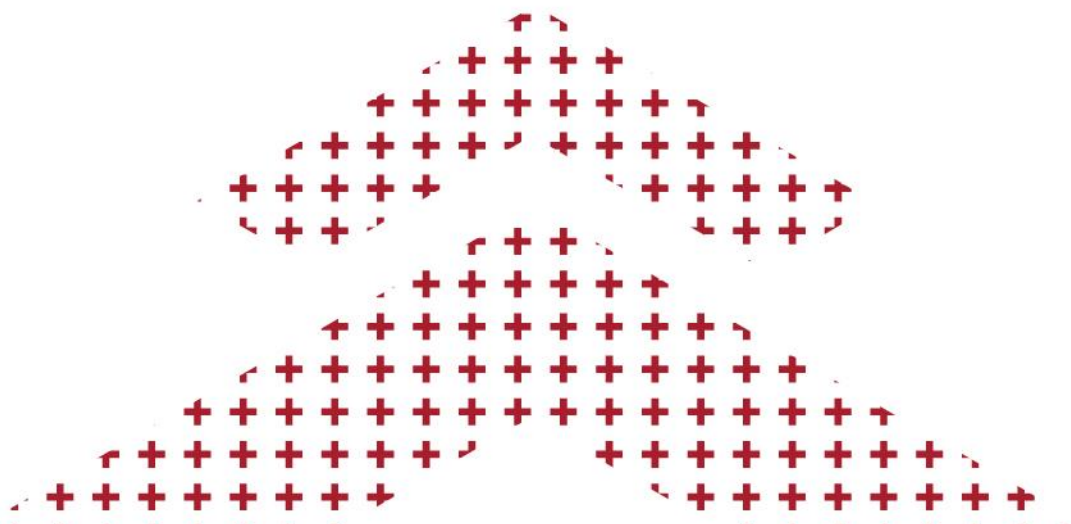


ISID

| Prueba técnica Front-end |





Prueba técnica Desarrollador Front-end

Información de la prueba

En este documento encontrarás la información de tu prueba técnica para el puesto de **Desarrollador Front-end** en **ISID**.

- › Instalación de Nuxt v.2.
- › Conexión API con Axios.
- › Almacenar datos en Vuex.
- › Mostrado de datos.
- › Componentes reutilizables.
- › Refactorización de código.

1. Instalación de Nuxt v.2.

Crear un proyecto con el framework de Vue, **Nuxt**, en su versión 2.

2. Conexión API con Axios.

Deberás realizar una conexión API con la librería **Axios**. Te damos libertad de utilizar cualquier API con la que te sientas cómodo. No obstante, te dejamos la siguiente URL una API de datos de usuario ficticios.

<https://jsonplaceholder.typicode.com/users/>

3. Almacenamiento en Vuex.

Instala la librería **Vuex** y almacena en su store todos los datos de usuario recibidos en la API.

4. Mostrado de datos.

Muestra los datos de usuarios desde la store en una tabla generada con la librería de CSS **Element UI**.

5. Componentes reutilizables.

Crea un componente reutilizable **Button**.

Deberás crear con el dos instancias. Un "editar" y un "eliminar".

La tabla (punto 4) deberá tener una columna llamada "Acciones" en el que utilizarás estas instancias.

6. Refactorización de código.

Explica brevemente **cómo refactorizarías** el siguiente código.

```

1  <template>
2    <div>
3      <!-- Coordinates input -->
4      <el-input placeholder="Latitude" v-model="form.coordinates[0]"></el-input>
5      <el-input placeholder="Altitude" v-model="form.coordinates[1]"></el-input>
6
7      <!-- MAP -->
8      <client-only>
9        <l-map style="height: 350px" :zoom="zoom" :center="center">
10
11          <l-tile-layer :url="url" :attribution="attribution"></l-tile-layer>
12
13          <l-marker draggable :lat-lng="markerLatlng" @dragend="getMarkerPosition">
14            <l-tooltip>Hello!</l-tooltip>
15          </l-marker>
16
17        </l-map>
18      </client-only>
19
20    </div>
21  </template>
22
23  <script>
24    export default {
25      name: "TestComponent",
26      data() {
27        return {
28          form: {
29            name: '',
30            position: {
31              type: '',
32              coordinates: [],
33            }
34          },
35          markerLatlng: [40.4357604633955, -3.6865084995701385], // Example coordinates.
36        }
37      },
38      methods: {
39        getMarkerPosition(e) {
40          const coordinates = e.target.getLatLng();
41          this.markerLatlng[0] = coordinates.lat;
42          this.markerLatlng[1] = coordinates.lng;
43
44          this.form.position.coordinates = this.markerLatlng;
45          console.log(this.markerLatlng); // Detect the change and display the array with the new coordinates.
46          console.log(this.form.position.coordinates); // Detects the change and displays the changed value.
47        },
48      }
49    }
50  </script>

```