

JavaScript Avanzado

Sesión 14



Universidad
Tecnológica
del Perú

¿Tienen alguna consulta o duda sobre la clase previa?

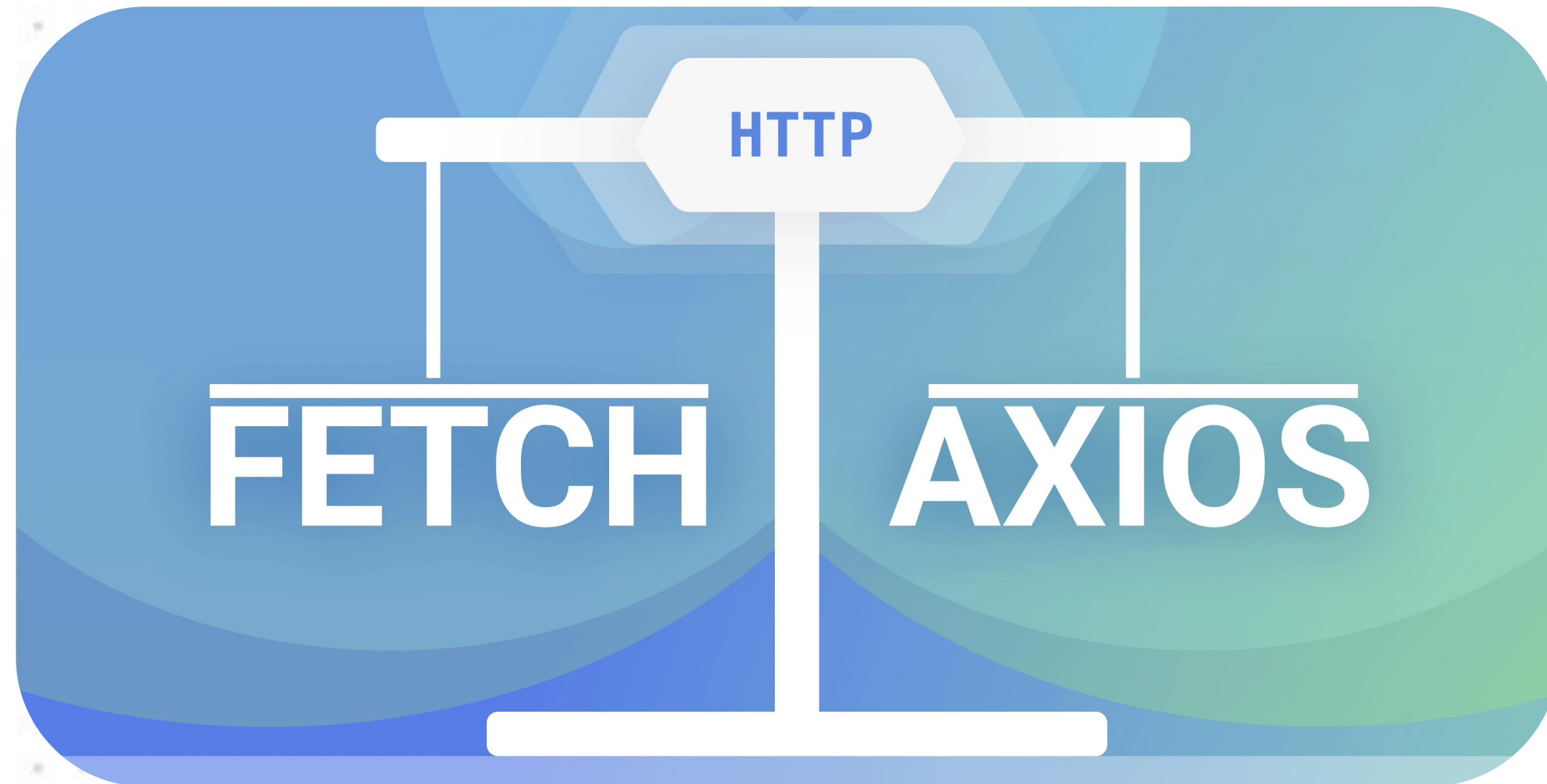


Logro de la sesión

Al finalizar la sesión, el estudiante construye una aplicación web empleando la biblioteca React empleando herramientas y bibliotecas para consumir diversas API REST.

¿Qué es REST? ¿Qué usos tendrá con React?





¿Cuál es la importancia del consumo de servicios web?



Contenido

Servicios en React

- Fetch API
- Axios
- Alova



Universidad
Tecnológica
del Perú

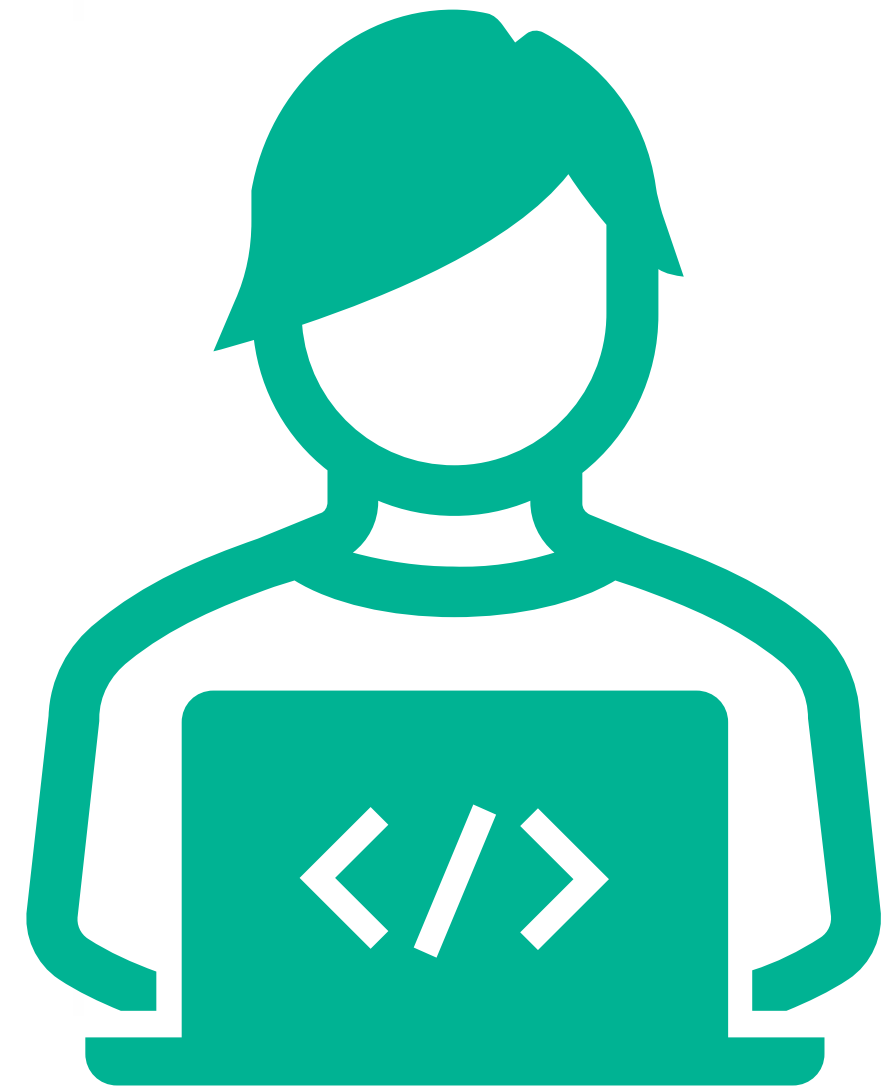
React con Fetch API

```
1  import { useState, useEffect, FC } from 'react';
2
3  const FetchExample: FC = () => {
4    const [data, setData] = useState<any>(null);
5    useEffect(() => {
6      fetch('http://localhost:3005/clientes')
7        .then(response => response.json())
8        .then(data => setData(data))
9        .catch(error => console.error('Error:', error));
10   }, [data]);
11   return (
12     <div>
13       {data
14         ? <pre>
15           {JSON.stringify(data, null, 2)}
16         </pre>
17         : 'Cargando...'}
18     </div>
19   );
20 };
21 export default FetchExample;
```

- **Fetch API** es una interfaz nativa de **JavaScript** para realizar **solicitudes HTTP**.
- Es simple y está disponible en todos los navegadores modernos.

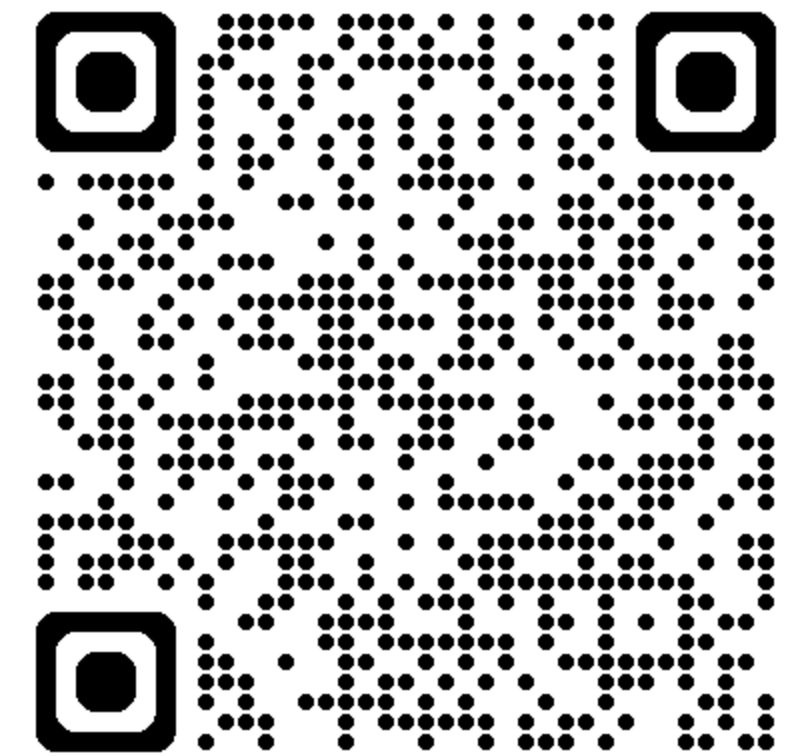
Programando

- Consumimos un **servicio web** empleando **Fetch** y **React**.



Axios

- Es un **cliente HTTP** basado en **promesas** para **node.js** y el **navegador**.
- Es **isomórfico** (puede ejecutarse en el navegador y node.js con la misma base de código).
- En el lado del servidor utiliza el módulo node.js nativo, mientras que en el cliente (**navegador**) utiliza **XMLHttpRequests**.



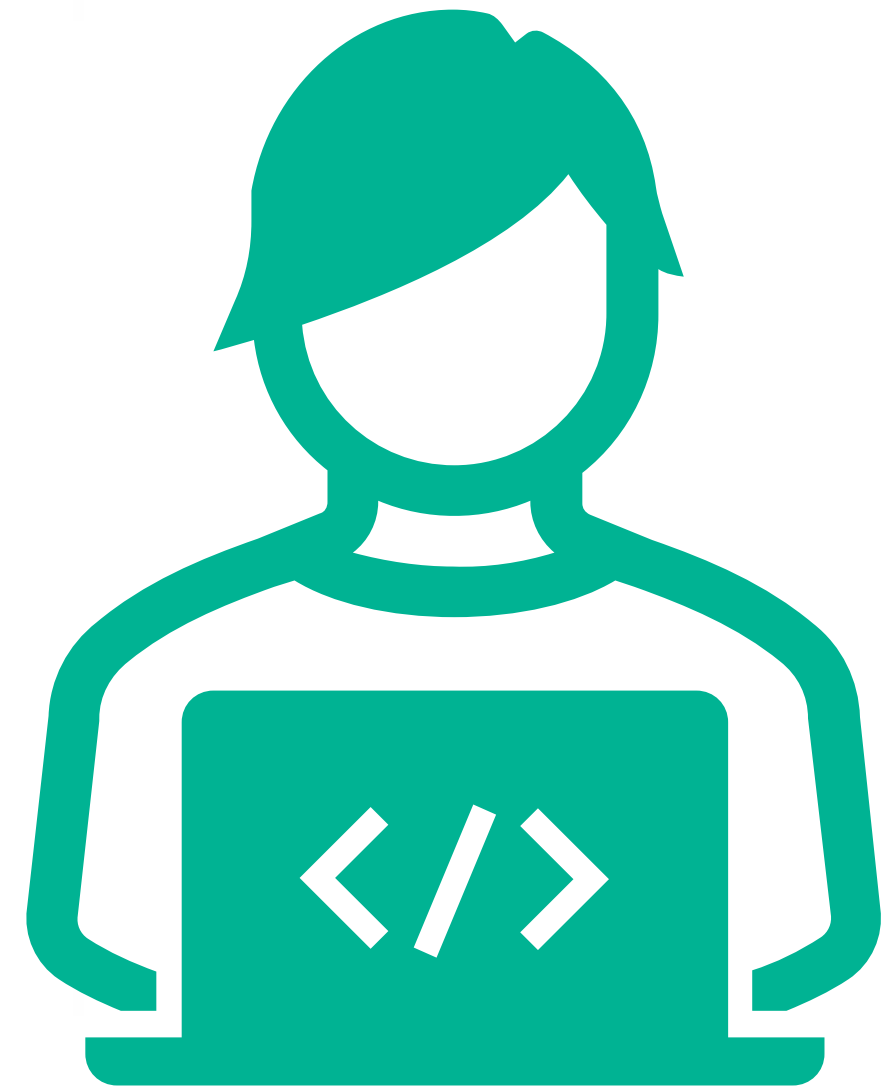
React con Axios

```
1  import React, { useState, useEffect } from 'react';
2  import axios from 'axios';
3
4  const AxiosExample: React.FC = () => {
5    const [data, setData] = useState<any>(null);
6    useEffect(() => {
7      axios.get('http://localhost:3005/clientes')
8        .then(response => setData(response.data))
9        .catch(error => console.error('Error:', error));
10   }, [data]);
11   return (
12     <div>
13       {data
14         ? <pre>{JSON.stringify(data, null, 2)}</pre>
15         : 'Cargando...'}
16     </div>
17   );
18 };
19
20
21 export default AxiosExample;
```

- **Axios** es una biblioteca de terceros que facilita las solicitudes **HTTP**.
- Ofrece una **API** amigable y con **características** como interceptores y cancelación de solicitudes.

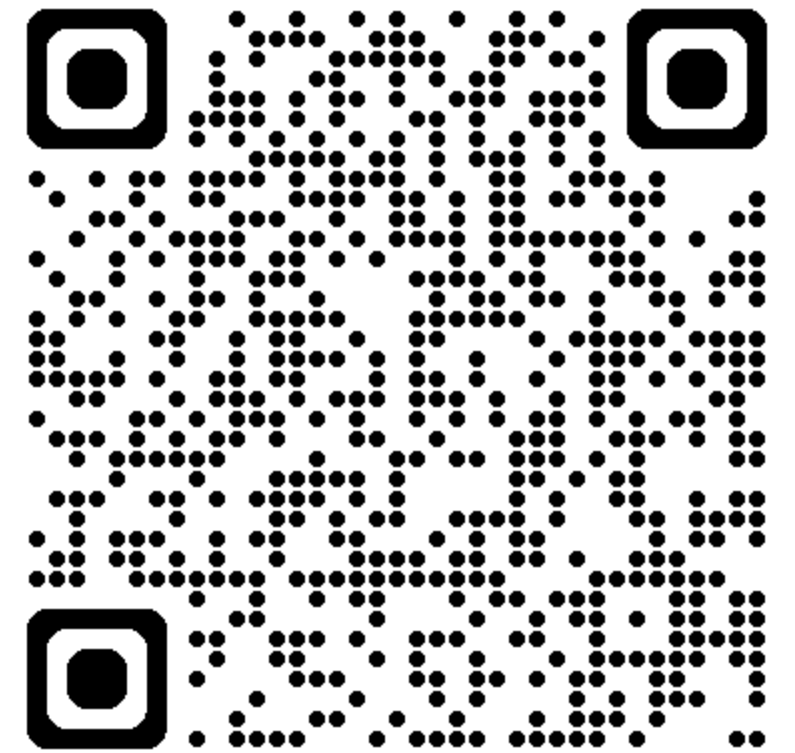
Programando

- Consumimos un **servicio web** empleando **Axios** y **React**.



Alova

- Es una herramienta de solicitud de próxima generación **optimizada** para el **flujo de trabajo**.
- Hace que la integración de una solicitud sea simple y mantiene una interacción de datos **cliente-servidor** más eficiente.



React con Alova

- **Alova** es una **biblioteca** enfocada en la eficiencia y la gestión de estado de las solicitudes **HTTP**.
- Provee gestión automática del **estado del caché** de datos y de **las solicitudes**.
- Para una integración con React se requiere:
 1. Instalar y crear una instancia de Alova
 2. Invocar la instancia para las peticiones HTTP


```

1  import { createAlova } from 'alova';
2  import adapterFetch from 'alova/fetch';
3  import reactHook from 'alova/react';
4
5  // create a alova instance
6  export const alovaInstance = createAlova({
7    baseUrl: 'http://localhost:3005',
8    statesHook: reactHook,
9    requestAdapter: adapterFetch(),
10   responded: response => {
11     if (response.status !== 200) {
12       throw new Error('request error');
13     }
14     return response.json();
15   }
16 });

```

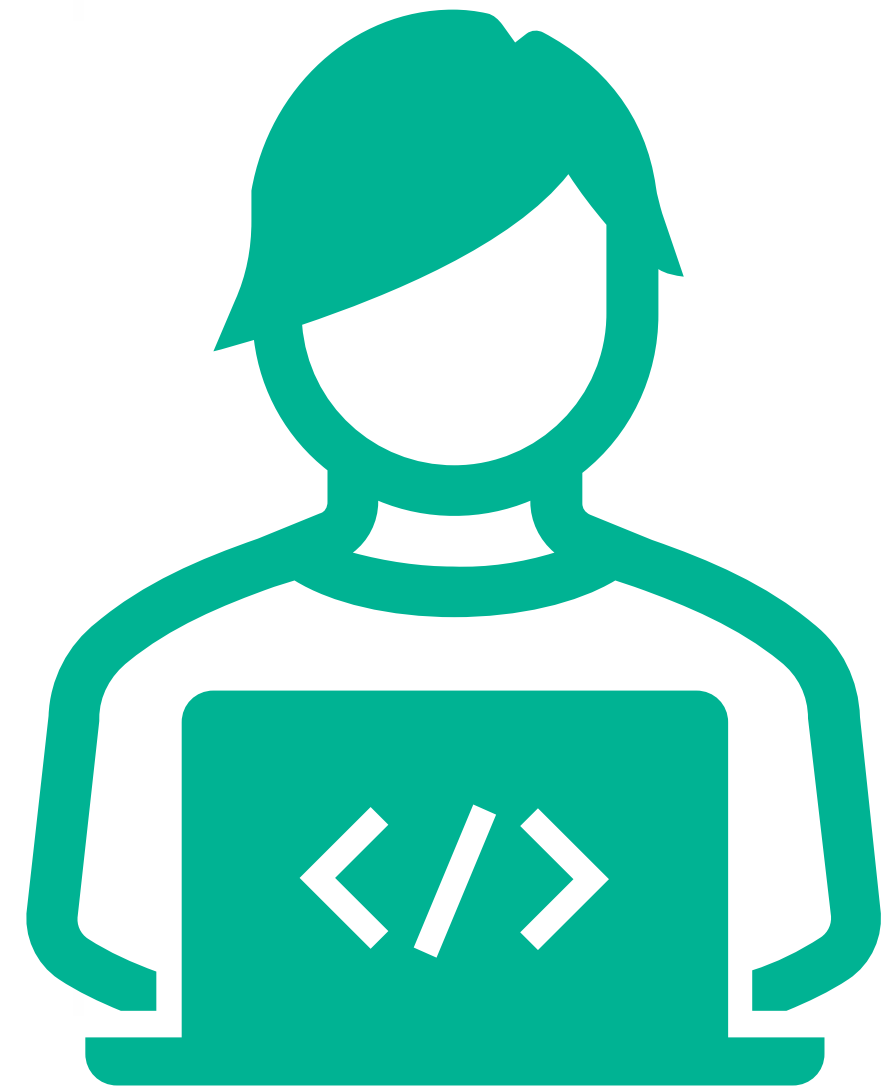
```

1  import { useRequest } from 'alova/client';
2  import { alovaInstance } from '../util/alovaHelper';
3  import { FC } from 'react';
4
5  const AlovaExample: FC = () => {
6    const { data, loading, error } = useRequest(
7      alovaInstance.Get('/clientes'), {
8        initialData: []
9      });
10
11    if (loading) return <div>Cargando...</div>;
12    if (error) return <div>Error: {error.message}</div>;
13
14    return (
15      <div>
16        {data
17          ? <pre>{JSON.stringify(data, null, 2)}</pre>
18          : 'Sin datos'}
19      </div>
20    );
21  };
22  export default AlovaExample;

```

Programando

- Consumimos un **servicio web** empleando **Alova** y **React**.



¿Tienen alguna consulta o duda?



Actividad



Resolver la actividad planteada en la plataforma.

Cierre

¿Qué hemos aprendido hoy?



Elaboramos nuestras conclusiones sobre el tema tratado



**Universidad
Tecnológica
del Perú**