



React Nivel Avanzado

# Consumiendo servicios web



Todas las aplicaciones necesitan mostrar información en diferentes formas y de diferentes fuentes, algunas usan información almacenada de forma local o directamente en el código (a esto se le conoce como código duro o *hardcode*), con el tiempo, han surgido diferentes formas de almacenar la información y compartirla, de manera que una app o plataforma web pueda consultar información almacenada en otra fuente diferente a lo que se describe en el código fuente.

A la acción de consultar información de otro sitio y presentarla al usuario se le llama “consumir”, en el entorno del desarrollo, comúnmente se puede obtener información de diferentes fuentes como servicios web, APIs, plataformas privadas o abiertas, entre muchas más.

La información que se puede conseguir es prácticamente ilimitada y sólo se limita a lo que nuestra aplicación necesite mostrar según las especificaciones.

# Explicación

Comúnmente las aplicaciones que trabajan con internet, utilizan algún servicio web para obtener información y presentarla al usuario, para esto existen diferentes protocolos y métodos que permiten obtener información que posteriormente será mostrada al usuario o procesada de alguna manera por una aplicación.

Esto se realiza a través de peticiones de datos desde la aplicación hacia un servidor que se encuentre, comúnmente en la nube, aunque puede encontrarse en una red privada y protegida por ciertas herramientas.

## Servicios web REST

Cuando se envía una solicitud de datos a una API de REST, se suele hacer a través de un protocolo HTTP. Una vez que reciben la solicitud, las API REST (conocidas también como servicios web) pueden devolver mensajes en distintos formatos como lo son:

- HTML
- XML
- Texto sin formato (texto plano)
- JSON
- BLOB (objetos binarios largos)

El formato preferido para los mensajes es la notación de objetos JavaScript (JSON), ya que, a pesar de su nombre, puede leerlo cualquier lenguaje de programación, es ligero y lo comprenden tanto las personas como las máquinas. De esta forma, las API RESTful son más flexibles y se pueden configurar con mayor facilidad. (Redhat, 2023)

## Servicios con GraphQL

Según su sitio web, GraphQL es un lenguaje de consulta para API que brinda una descripción completa y comprensible de los datos en una API, brinda a los clientes el poder de solicitar exactamente lo que necesitan y nada más. (GraphQL, 2023)

En otras palabras, GraphQL nos permite realizar consultas a los servicios web de otras formas mucho más intuitivas y que de forma habitual en peticiones a servicios REST serían múltiples llamadas o incluso diferentes llamadas a servicios diferentes.

## Realizar peticiones

Para realizar una petición en React, sólo tenemos que hacer uso de la API Fetch, la API Fetch proporciona una interfaz para obtener recursos (incluso a través de la red). Es un reemplazo más poderoso y flexible para XMLHttpRequest. (Mozilla Developer Network, 2023)

La API Fetch nos permite obtener recursos con sólo indicarle la URL y en algunos casos nos puede solicitar autenticación o enviar los datos de cierta forma en específico, esto variará según la implementación que haya realizado el desarrollador del recurso que deseemos acceder.

Ejemplo de uso de fetch:

Suponiendo que queremos consultar una API que nos da los precios actuales de las criptomonedas más populares, podríamos acceder a este con los siguientes datos:

URL: `https://api.coindesk.com/v1/bpi/currentprice.json`

Método: GET

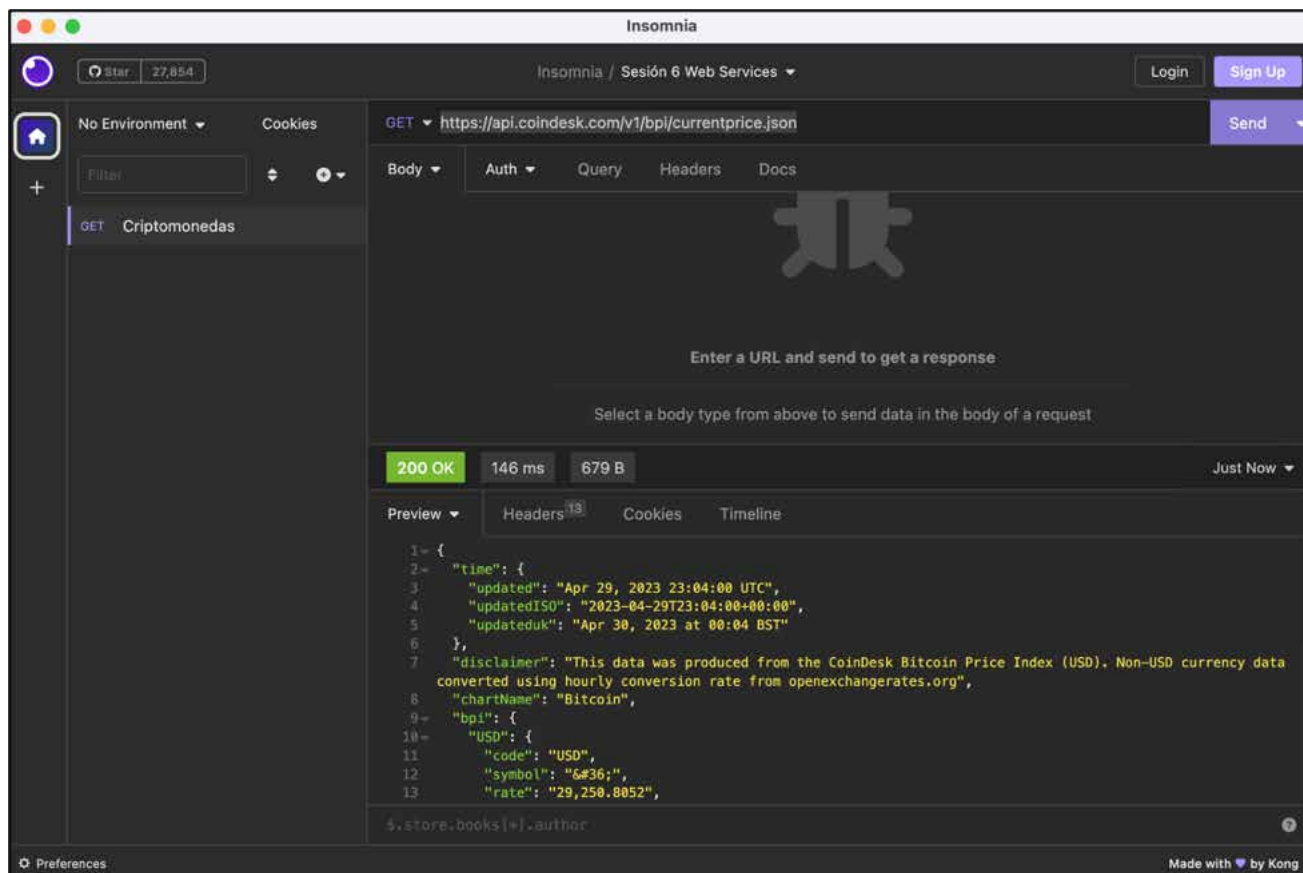
El código anterior al ser un método GET, al cual se puede acceder desde el navegador, ya que tampoco pide una autenticación.

Al acceder desde el navegador, se mostrará lo siguiente:

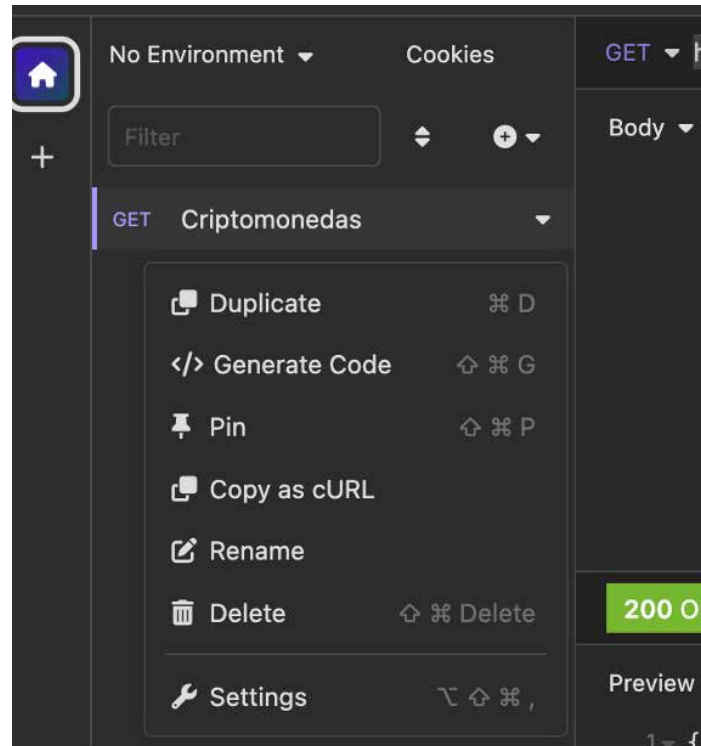


```
{
  "time": {
    "updated": "Apr 29, 2023 23:03:00 UTC",
    "updatedISO": "2023-04-29T23:03:00+00:00",
    "updateduk": "Apr 30, 2023 at 00:03 BST",
    "disclaimer": "This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD currency data converted using hourly conversion rate from openexchangerates.org",
    "chartName": "Bitcoin",
    "bpi": {
      "USD": {
        "code": "USD",
        "symbol": "$",
        "rate": "29,254.2504",
        "description": "United States Dollar",
        "rate_float": 29254.2504
      },
      "GBP": {
        "code": "GBP",
        "symbol": "£",
        "rate": "24,444.6176",
        "description": "British Pound Sterling",
        "rate_float": 24444.6176
      },
      "EUR": {
        "code": "EUR",
        "symbol": "€",
        "rate": "28,497.9110",
        "description": "Euro",
        "rate_float": 28497.911
      }
    }
  }
}
```

Si se consume desde el cliente REST, este es el resultado.



Si se quiere ejecutar desde React, es posible hacerlo generando el código desde el cliente REST, solo se necesita hacer clic junto al nombre de la petición y elegir la opción **Generate Code**.

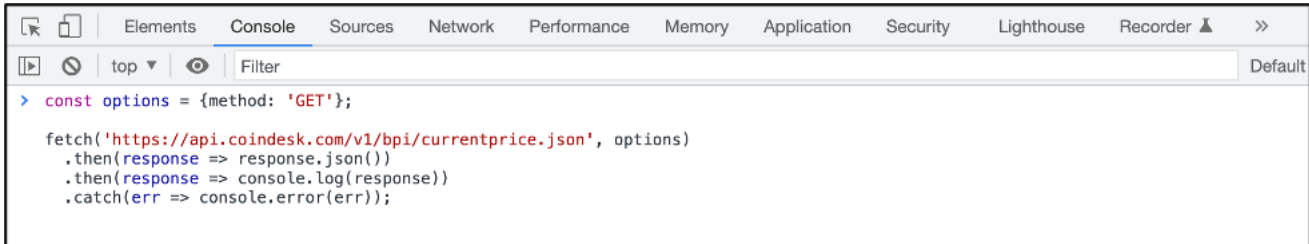


En la ventana que aparece, seleccionar JavaScript y junto a esta opción, fetch, esto dará el código necesario para hacer una petición desde React que reciba los datos y los imprima en consola.

Este código es compatible tanto con Javascript como con React JS y React Native, pudiéndolo incluso ejecutar en la consola de desarrollador de Chrome.



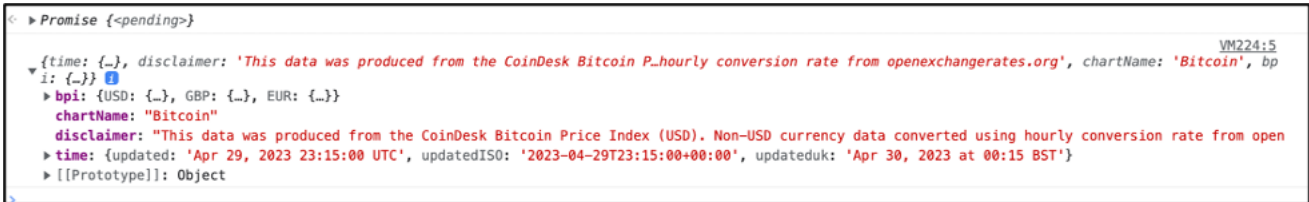
Se puede abrir el navegador y las herramientas de desarrollo, para posteriormente pegar el código anterior en la consola.



```
> const options = {method: 'GET'};

fetch('https://api.coindesk.com/v1/bpi/currentprice.json', options)
  .then(response => response.json())
  .then(response => console.log(response))
  .catch(err => console.error(err));
```

Después de haber pegado el código, presionar **enter** para obtener el resultado impreso justo debajo del código mostrando lo siguiente:



```
< ▶ Promise {<pending>}

{time: {_, disclaimer: 'This data was produced from the CoinDesk Bitcoin P...hourly conversion rate from openexchangerates.org', chartName: 'Bitcoin', bp
i: {_}}
  ▶ bpi: {USD: {_, GBP: {_, EUR: {_}}}
    chartName: "Bitcoin"
    disclaimer: "This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD currency data converted using hourly conversion rate from open
  ▶ time: {updated: 'Apr 29, 2023 23:15:00 UTC', updatedISO: '2023-04-29T23:15:00+00:00', updateduk: 'Apr 30, 2023 at 00:15 BST'}
  ▶ [[Prototype]]: Object
```

Este mismo código se puede agregar en la aplicación y hacer uso de estados para guardar el resultado.



El uso de servicios web forma una parte esencial en cómo se intercambia información entre las aplicaciones y los diferentes servicios que se pueden usar durante el desarrollo de la aplicación, el comprender su funcionamiento permite desarrollar mejor los servicios y encontrar nuevas formas de optimizar el funcionamiento de la aplicación y también, en ciertos casos el uso de llamadas a los servicios.

Los servicios web se utilizan en una amplia variedad de aplicaciones, desde sistemas empresariales complejos hasta aplicaciones móviles y sitios web. Algunos ejemplos de servicios web incluyen servicios de geolocalización, servicios de pago en línea y servicios de redes sociales.

El intercambio de información entre aplicaciones puede mejorar la eficiencia, la flexibilidad y la escalabilidad de los sistemas de información y permitir una mayor integración entre diferentes aplicaciones y plataformas.

## Referencias bibliográficas

- GraphQL. (2023). *A query language for you API*. Recuperado de <https://graphql.org>
- Mozilla Developer Network. (2023). *Fetch API*. Recuperado de [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)
- Redhat. (2023). *Diferencias entre REST y SOAP*. Recuperado de <https://www.redhat.com/es/topics/integration/whats-the-difference-between-soap-rest#:~:text=REST%20es%20un%20conjunto%20de,recomendaciones%20depende%20de%20los%20desarrolladores>.



## Para saber más

### Lecturas

Para conocer más acerca de **API Fetch**, te sugerimos revisar lo siguiente:

- Mozilla Developer Network. (2023). *Fetch() Global function*. Recuperado de <https://developer.mozilla.org/en-US/docs/Web/API/fetch>
- React Native Docs. (2023). *Using the Fetch API*. Recuperado de <https://reactnative.dev/docs/network#using-fetch>

### Videos

Para conocer más acerca de **cómo consumir una API**, te sugerimos revisar lo siguiente:

- Midulive. (2022, 17 de julio). *Cómo CONSUMIR una API REST con JAVASCRIPT y Fetch + Promises con gestión de Errores* [Archivo de video]. Recuperado de [https://www.youtube.com/watch?v=FJ-w0tf3d\\_w](https://www.youtube.com/watch?v=FJ-w0tf3d_w)
- ColorCode. (2023, 5 de marzo). *Fetch API - JavaScript Tutorial for beginners* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=ubw2hdQIl4E>

## Checkpoint

Asegúrate de:

- Comprender cómo consumir APIs mediante un cliente REST.
- Manipular los resultados obtenidos tras una consulta a un servicio web.
- Entender el funcionamiento de los diferentes métodos HTTP.

## Requerimientos técnicos

- Android Studio
- React
- Xcode
- Insomnia
- VS Code

## Pework

- Deberás revisar:
  - Tema 1. Fundamentos de React.
  - Tema 2. Primeros pasos con React Native.
  - Tema 3. Componentes Core y Nativos.
  - Tema 4. Maquetando nuestra app.
  - Tema 5. Eventos, estados y contexto.
  - Tema 6. Consumiendo servicios web



La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.