

Consumo de un webservice JSON con JavaScript

Código JavaScript para consultar un URI que devuelve una salida JSON (Fig. 1), el URI es <http://huasteco.tiburcio.mx/pmarvel>:

Código 1

```
fetch('http://huasteco.tiburcio.mx/pmarvel')
  .then(response => response.json())
  .then(function(result) { for(let i = 0; i < result.length; i++)
    console.log(result[i].nombreP+" "+result[i].foto)
  })
```

Configurar CORS (Cross-Origin Resource Sharing)

Fue necesario configurar CORS en el servidor Apache2 en UBUNTU 18.04 modificando el siguiente archivo:

nano /etc/apache2/apache2.conf

A continuación se muestra la línea agregada (en negro) en el apartado *Directory* correspondiente al directorio donde reside el sitio Web:

Código 2

```
<Directory /var/www/>
  Options Indexes FollowSymLinks
  AllowOverride None
  Require all granted
  Header set Access-Control-Allow-Origin "*"
</Directory>
```

Se tiene también que habilitar el módulo de headers con el comando:

a2enmod headers

Y reiniciar el servidor Apache2 con:

systemctl restart apache2

Código en JavaScript que recibe un URI con respuesta JSON (Fig. 1) que contiene URLs de imágenes y visualiza las imágenes en un elemento *div* (Fig. 2). El URI es:

<http://huasteco.tiburcio.mx/pmarvel>

Código 3

```
document.write("<div id=contenedor2></div>")

var sal=""

fetch('http://huasteco.tiburcio.mx/pmarvel')
  .then(response => response.json())
  .then(function(result) { for(let i = 0; i < result.length; i++)
    sal+="
```

Nota: Los códigos mostrados en este documento han sido ejecutados desde la consola (Fig. 4) de un navegador web dentro de la opción de *inspeccionar* (Fig. 3) que proveen las mayoría de los navegadores.

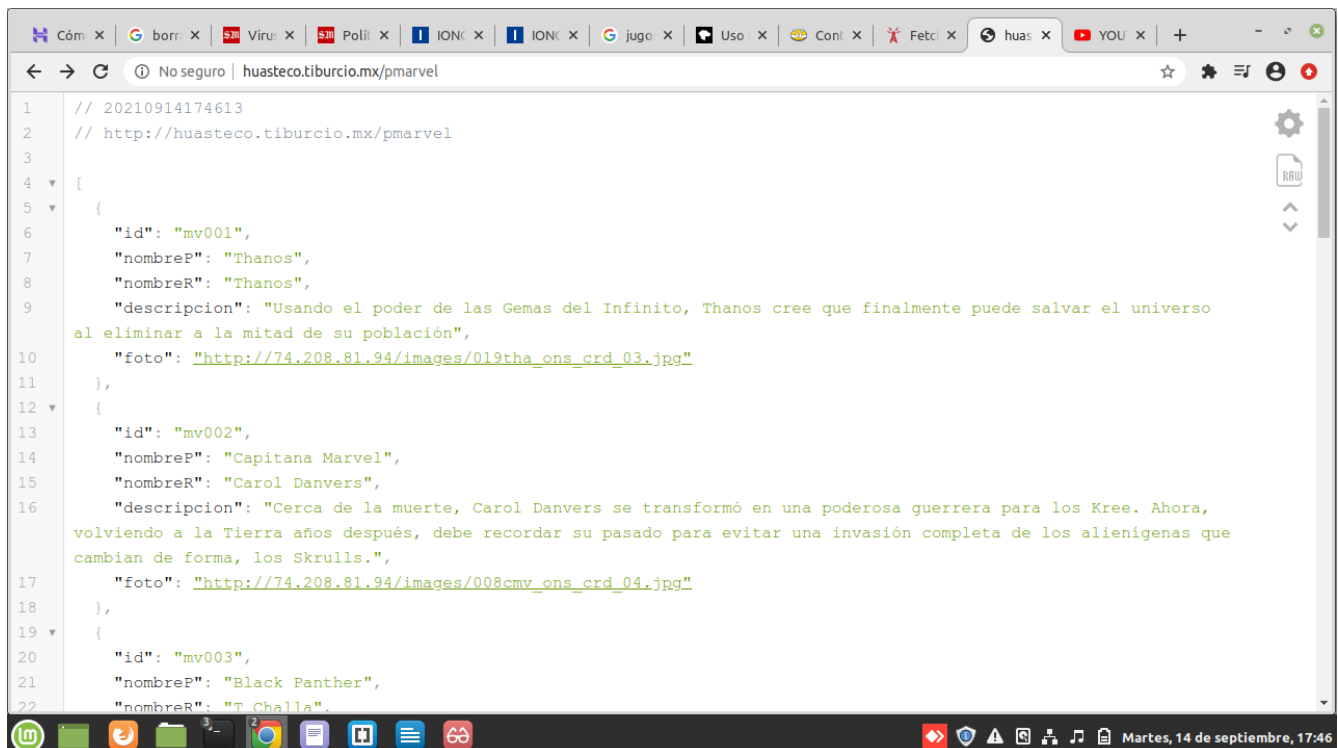


Figura 1 Salida JSON de <http://huasteco.tiburcio.mx/pmarvel>

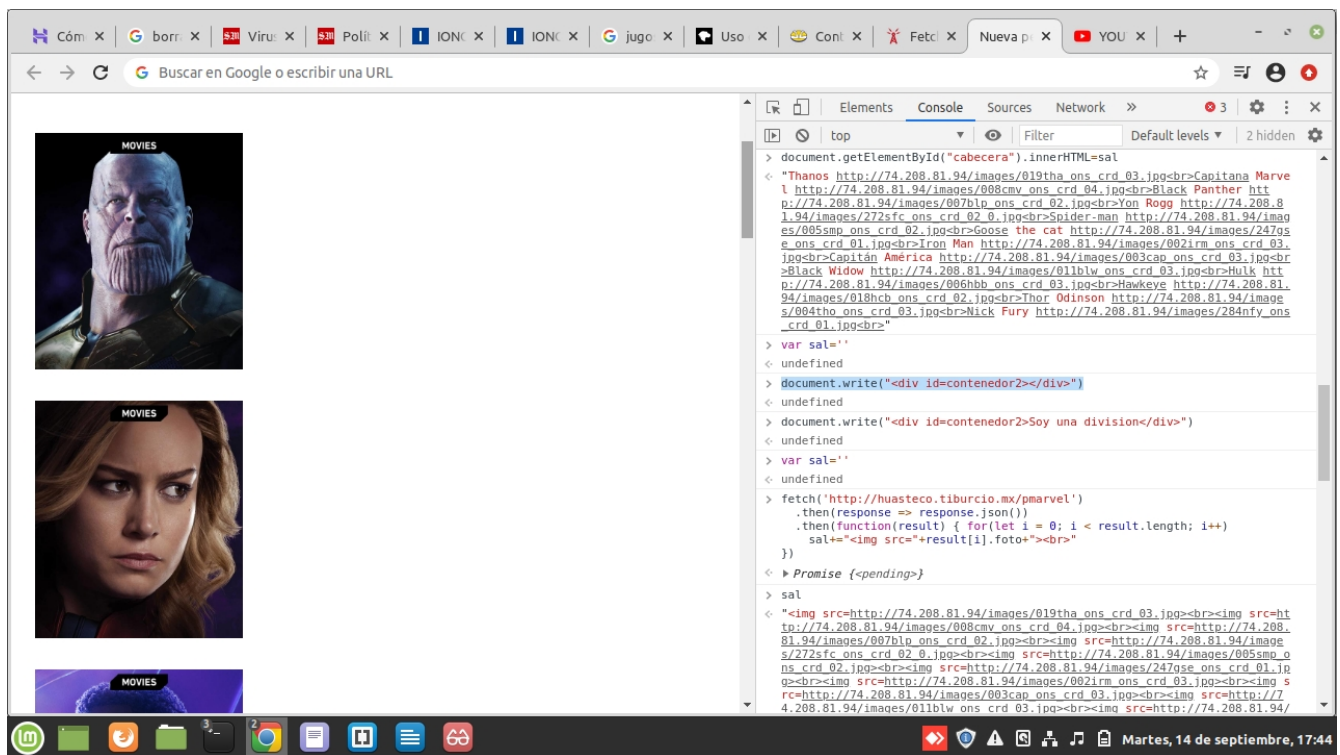


Figura 2 Ejecución del código 3

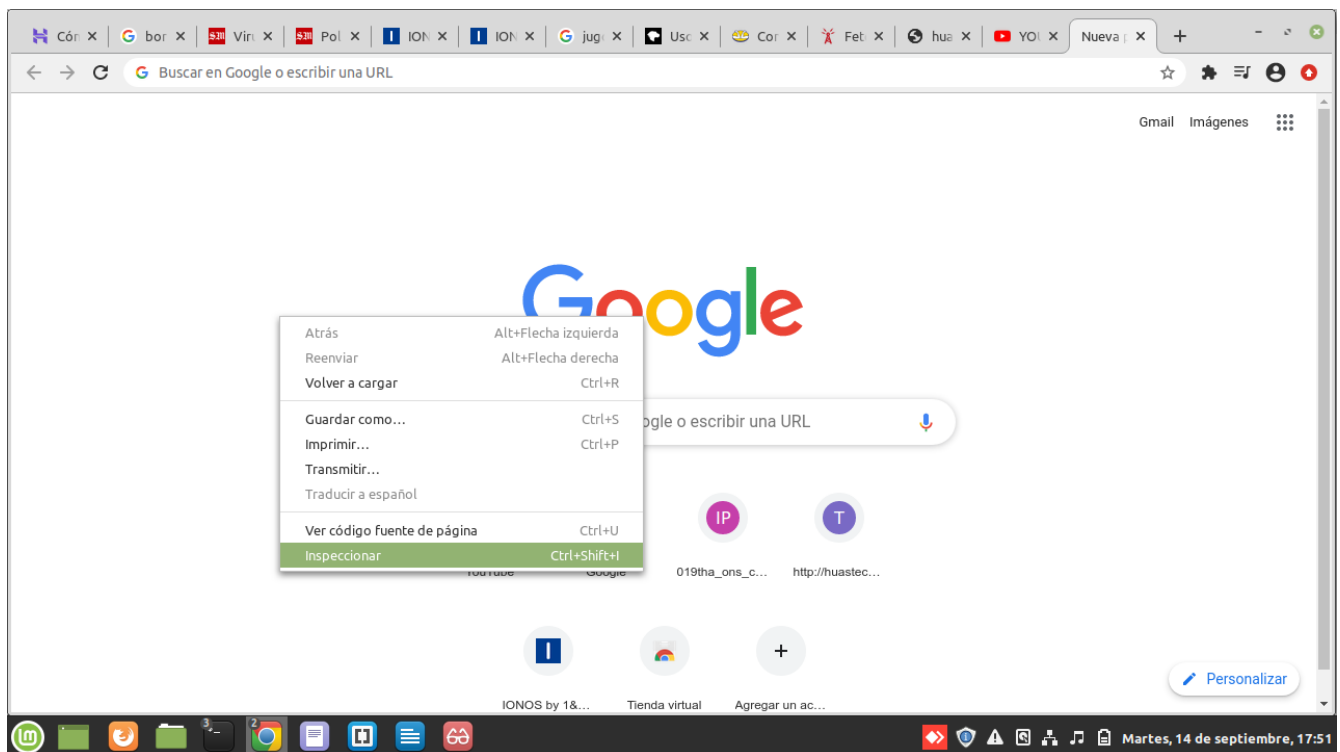


Figura 3 Opción de “Inspeccionar” (botón derecho del navegador).

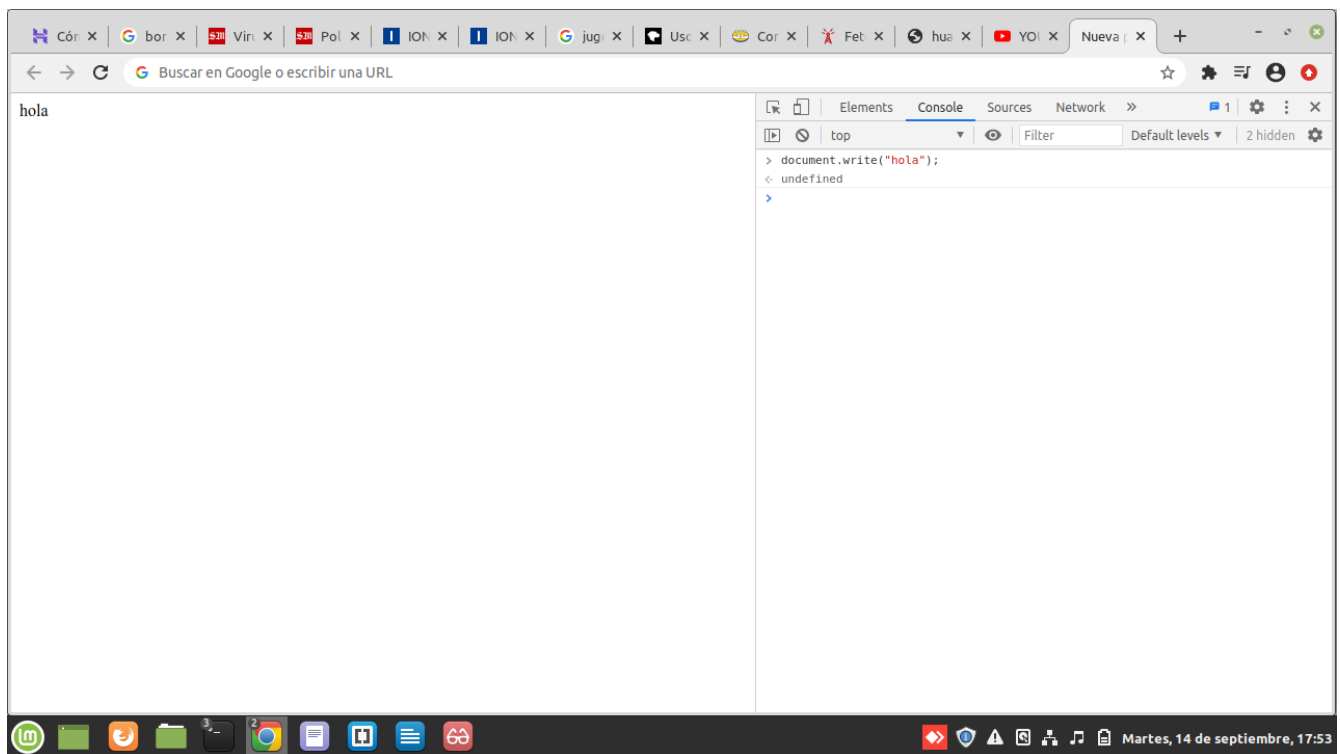


Figura 4 Consola de JavaScript del navegador.

El código siguiente permite probar si la operación *fetch* es satisfactoria. De no serlo se dispara una alerta.

Código 5

```
try {  
  await fetch('http://example.com');  
}  
catch(err) {  
  alert(err); // Failed to fetch  
}
```