

# **Ejercicios API FETCH**

## **CFGs 2º DAW**



**Curso: 2022/2024**

**Departamento de Informática**

## Licencia:



**Reconocimiento – No Comercial – Compartir Igual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## Índice de Contenidos

1. Ejercicio .....	4
2. Ejercicio .....	5
3. Ejercicio .....	6
4. Ejercicio .....	7
5. Ejercicio .....	8
6. Ejercicio .....	11

**hImportante 1:** no intentes copiar ejercicios ni tan siquiera “ver un poco” código de otros compañeros. Es el mayor error de quien empieza a programar, ya que luego no sabe resolver problemas por sí mismo y da una falsa sensación de aprendizaje.

**hImportante 2:** si en programación algo no sale a la primera... es totalmente normal. Es parte del aprendizaje. ¿Cómo crees que aprendieron los mejores programadores?

## 1. Ejercicio.

Haz la siguiente petición:

<https://swapi.dev/api/people/>

Y recorre el resultado para mostrar una lista con los datos recuperados.

## 2. Ejercicio.

Mediante la dirección web:

<https://apis.datos.gob.ar/georef/api/provincias>

Se nos retorna un objeto JSON con la información de los nombres de las provincias de Argentina, su id etc.

Recuperar mediante el API fetch los datos y mostrarlos.

La estructura del archivo JSON es:

```
{
  "provincias": [
    {
      "nombre": "Santiago del Estero",
      "id": "86",
      "centroide": {
        "lat": -27.782412,
        "lon": -63.252387
      }
    },
    {
      "nombre": "Córdoba",
      "id": "23",
      "centroide": {
        "lat": -25.72323412,
        "lon": -63.2523322387
      }
    }
    //Aquí las otras provincial
  ],
  "cantidad": 23,
}
```

### 3. Ejercicio.

Mediante la dirección web:

<https://freegeoip.app/json/>

freegeoip.app nos proporciona una API de geolocalización IP. Utiliza una base de datos de direcciones IP asociadas a ciudades junto con otra información relevante como zona horaria, latitud y longitud.

La estructura del archivo JSON es:

```
{  
  "ip":"186.123.122.56",  
  "country_code":"AR",  
  "country_name":"Argentina",  
  "region_code":"X",  
  "region_name":"Córdoba",  
  "city":"Córdoba",  
  "zip_code":"5000",  
  "time_zone":"America/Argentina/Cordoba",  
  "latitude":-31.4015,  
  "longitude":-64.1803,  
  "metro_code":0}
```

Mostrar en la página el nombre del país, ciudad, código postal e ip.

## 4. Ejercicio.

Mediante la dirección web:

[https://jsonplaceholder.typicode.com/users?id=\[número de 1 a 10\]](https://jsonplaceholder.typicode.com/users?id=[número de 1 a 10])

Recuperamos los datos de un usuario en particular. Ingresar en un formulario el id de un usuario y al presionar un botón mostrar los datos del mismo.

La estructura del archivo JSON es:

```
[
  {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  }
]
```

## 5. Ejercicio.

Mediante la dirección web:

<https://randomuser.me/api/>

Se nos retorna un objeto JSON con la información de un usuario.

Recuperar mediante el API fetch los datos y mostrar el primer y segundo nombre.

La estructura del archivo JSON es:

```
{
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "mr",
        "first": "brad",
        "last": "gibson"
      },
      "location": {
        "street": "9278 new road",
        "city": "kilcoole",
        "state": "waterford",
        "postcode": "93027",
        "coordinates": {
          "latitude": "20.9267",
          "longitude": "-7.9310"
        },
        "timezone": {
          "offset": "-3:30",
          "description": "Newfoundland"
        }
      },
      "email": "brad.gibson@example.com",
      "login": {
```



```
"uuid": "155e77ee-ba6d-486f-95ce-0e0c0fb4b919",
"username": "silverswan131",
"password": "firewall",
"salt": "TQA1Gz7x",
"md5": "dc523cb313b63dfe5be2140b0c05b3bc",
"sha1": "7a4aa07d1bedcc6bcf4b7f8856643492c191540d",
"sha256":
"74364e96174afa7d17ee52dd2c9c7a4651fe1254f471a78bda0190135dcd3480"
},
"dob": {
  "date": "1993-07-20T09:44:18.674Z",
  "age": 26
},
"registered": {
  "date": "2002-05-21T10:59:49.966Z",
  "age": 17
},
"phone": "011-962-7516",
"cell": "081-454-0666",
"id": {
  "name": "PPS",
  "value": "0390511T"
},
"picture": {
  "large": "https://randomuser.me/api/portraits/men/75.jpg",
  "medium": "https://randomuser.me/api/portraits/med/men/75.jpg",
  "thumbnail": "https://randomuser.me/api/portraits/thumb/men/75.jpg"
},
"nat": "IE"
}
],
```

```
"info": {  
  "seed": "fea8be3e64777240",  
  "results": 1,  
  "page": 1,  
  "version": "1.3"  
}  
}
```

Luego según el valor que almacena la propiedad "gender" que puede ser "male" o "female" proceder a recuperar nuevamente con el API fetch una imagen llamada 'hombre.png' o 'mujer.png', pero que se encuentra en nuestro servidor.

Mostrar entonces en la página el primer nombre, segundo nombre y la imagen respectiva. Hacer esto inmediatamente se carga la página web.

Tener en cuenta que dentro del método 'then' podemos hacer la llamada nuevamente al método 'fetch'.

## 6. Ejercicio.

Mediante un formulario, insertar contenido dentro de un JSON y mostrar en la parte inferior todo el contenido del JSON en forma de tabla, tal como aparecen en la siguiente imagen:

Descripción del artículo

Precio:

Nuevo artículo

Código	Descripción	Precio
1	papas	23
2	peras	18
3	manzanas	55
4	naranjas	30
5	melón	120
6	sandia	19