

■ ENTREGABLE 4

Visualización de los datos dinámicamente mediante obtención remota de una API.

Fecha: 24/05/2021

1 ■ DATOS CON FETCH

Consultar "FETCH"

- Obtener datos con fetch.
- Modificar JS de síncrono a asíncrono.
- Añadir loader mientras carga los datos.

Estimación **8 puntos**

2 ■ REFACTOR - DRY

- Pasar loops a Array Methods, siempre que sea posible.
- Juntar funciones similares pasando más parámetros.
- Eliminar el máximo de variables globales posibles.

Estimación **4 puntos**



FETCH

Plan de ataque

- Solicitar datos a través de `FETCH`.
- Añadir el header:
`"X-Auth-Token": "${tu API KEY}"`
- Acceder a las promise devueltas con el `.then()`
- Tener en cuenta las funciones que necesitan los datos, y no llamarlas hasta obtenerlos.

EJEMPLO FETCH

```
function getDataFetch() {
  const url = "https://api.football-data.org/v2/competitions/2014/standings";
  fetch(url, {
    method: "GET",
    headers: {
      "X-Auth-Token": "3bcd6e5663a94bba833d420110684b01"
    }
  }).then(function (response) {
    if (response.ok == true) {
      return response.json();
    }
  }).then(function (data) {
    console.log(data);
  });
}
```

```
function getDataFetchAbbr() {
  const url = "https://api.football-data.org/v2/competitions/2014/standings";
  fetch(url, {
    method: "GET",
    headers: {
      "X-Auth-Token": "3bcd6e5663a94bba833d420110684b01"
    }
  }).then(response => {
    if (response.ok) response.json();
  }).then(data => {
    console.log(data);
  });
}
```

```
async function getData() {
  const url = "https://api.football-data.org/v2/competitions/2014/standings";
  const response = await fetch(url, {
    method: "GET",
    headers: {
      "X-Auth-Token": "3bcd6e5663a94bba833d420110684b01"
    }
  });
  const data = await response.json();
  console.log(data);
};
```

ASYNC / AWAIT
nivel avanzado