

# Desarrollo de una App que Accede a la API FORTNITE

Miembros del grupo:

Elena Díaz Pérez, Pablo Álvaro Serrano y Alejandro Navarro Fernández

S2DAW – 09/12/2024

## Índice

Introducción .....	3
Descripción de la API .....	3
Acceso a la API.....	3
Funcionalidad de la Aplicación más importantes .....	4
Funcionalidad de la Aplicación a nivel de usuario .....	5
Porcentaje de participación de los integrantes del grupo .....	7

## Introducción

Hemos desarrollado una aplicación web que interactúa con la API de Fortnite pública, con la cual podemos acceder a datos y verlos en una interfaz atractiva. Para ello, hemos hecho filtros de búsqueda por varios parámetros, y podemos añadir los cosméticos a favoritos.

## Descripción de la API

La API pública elegida es la de [Fortnite](#), tiene varios endpoints AES, Banners, Cosmetics, Creator Code, Map, News, Playlists, Shop y Stats. También está en varios lenguajes, en nuestro caso pondremos: **?language=es**. Los endpoints que hemos utilizado son Cosmetics y News.

### Acceso a la API

Para acceder a los datos hacemos un **fetch** y la API nos devuelve los datos en formato JSON. Convertimos la respuesta en un objeto JSON para poder procesarlos en JavaScript. La función **response.json()** devuelve una promesa que contiene los datos en formato JSON. Una vez convertida a JSON, se ejecuta el bloque de procesar los datos.

También manejamos los errores, si hay un error en el proceso mostramos que error es.

```
function fetchNews() {  
  fetch('https://fortnite-api.com/v2/news?language=es')  
    .then(response => response.json()) // Primero conviertes la respuesta a JSON  
    .then(jsondata => {  
      procesarJSON(jsondata);  
      procesarJSON2(jsondata);  
    })  
    .catch(error => {  
      console.error('Error al cargar los datos:', error);  
    });  
}
```

Para escribir y guardar los favoritos cogemos los datos de ese cosmético de la API y cuando le das a favoritos ese cosmético se guarda en un JSON.

Con este fetch guardamos los datos en favoritos.

```
// Verificar si el cosmético ya existe en la lista de favoritos
fetch('http://localhost:3000/favoritos?id=${cosmetico.id}')
.then(response => response.json())
.then(data => {
  if (data.length > 0) {
    alert('El cosmético ya está en la lista de favoritos.');
```

```
  } else {
    // Si no existe, agregarlo a la lista de favoritos
    fetch('http://localhost:3000/favoritos', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(favorito)
    })
    .then(response => response.json())
    .then(data => {
      console.log('Favorito guardado:', data);
    })
    .catch(error => {
      console.error('Error al guardar el favorito:', error);
    });
  }
})
.catch(error => {
  console.error('Error al verificar el favorito:', error);
});
```

Y con esto visualizamos los datos de favoritos

```
cosmeticosFav.js > fetchFavoritos
function fetchFavoritos() {
  fetch('http://localhost:3000/favoritos')
    .then(response => response.json())
    .then(jsondata => {
      mostrarFavoritos(jsondata);
    })
    .catch(error => {
      console.error('Error al cargar los datos:', error);
    });
}
```

## Funcionalidad de la Aplicación más importantes

Tenemos **addEventListener** para escucha el evento **DOMContentLoaded**, que se activa cuando el documento HTML ha sido completamente cargado y parseado por el navegador. Así nos aseguramos de que el script se ejecute **después de que el DOM esté listo**, evitando errores que ocurrirían si el script intenta manipular elementos HTML que aún no existen en la página.

Hemos creado una clase objeto para los cosméticos, para poder usarlo a la hora de coger los datos y enseñarlos, esto hace de que haya menos errores.

```

class Objeto {
  constructor(
    id,
    name,
    type,
    description,
    rarity,
    series,
    set,
    introduction,
    images,
    added
  ) {
    this.id = id;
    this.name = name;
    this.type = type;
    this.description = description;
    this.rarity = rarity;
    this.series = series;
    this.set = set;
    this.introduction = introduction;
    this.images = images;
    this.added = added;
  }
}

```

Para los filtros hemos accedido a la API de cosméticos. Al filtrar por el primer filtro por ejemplo por rareza, coges los datos y los filtras otra vez por Epic por ejemplo (opciones de segundo filtro) y directamente solo mostramos los Epic.

Otra cosa muy importante es la función LazyLoad, la hemos configurado para que los cosméticos se vayan añadiendo de 10 en 10 y así no se sature.

```

// Función para cargar más cosméticos cuando el usuario se acerque al final
function lazyLoadCosmeticos() {
  const loading = document.getElementById("loading-indicator");

  const options = {
    rootMargin: '200px', // Se activa antes de llegar al final de la página
    threshold: 1.0 // Cargar cuando el 100% del objetivo sea visible
  };

  const observer = new IntersectionObserver(entries => {
    entries.forEach(entry => {
      if (entry.isIntersecting) {
        if (currentBatch * batchSize < cosmeticData.length) {
          mostrarCosmeticos(); // Si hay más datos, mostramos el siguiente lote
        } else {
          loading.textContent = "No hay más cosméticos"; // Cuando ya no hay más elementos
        }
      }
    });
  }, options);

  observer.observe(loading); // Observar el indicador de carga
}

```

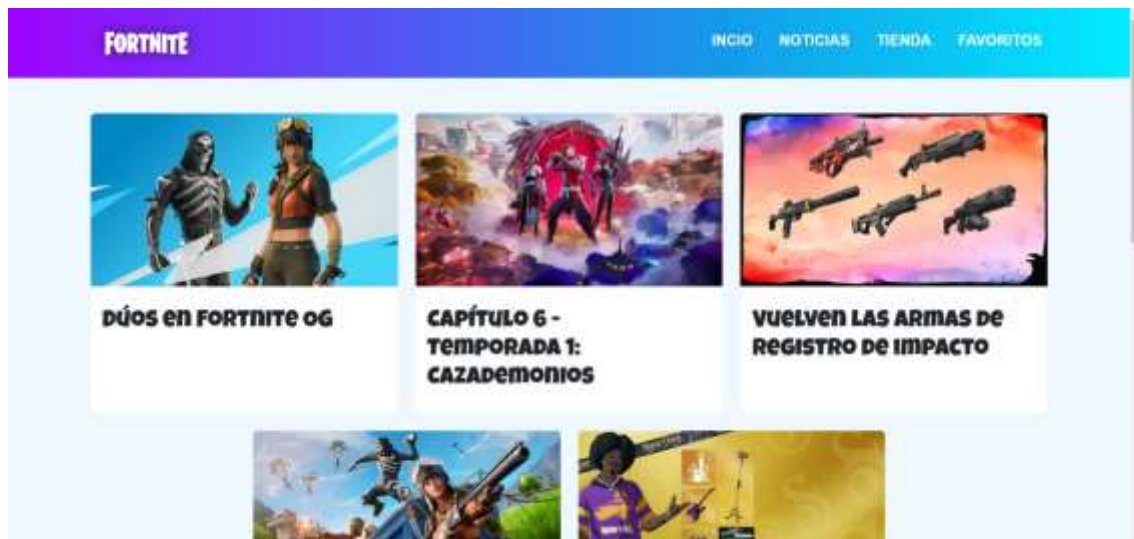
## Funcionalidad de la Aplicación a nivel de usuario

En la página de inicio el usuario puede descargarse el juego pulsando al botón.

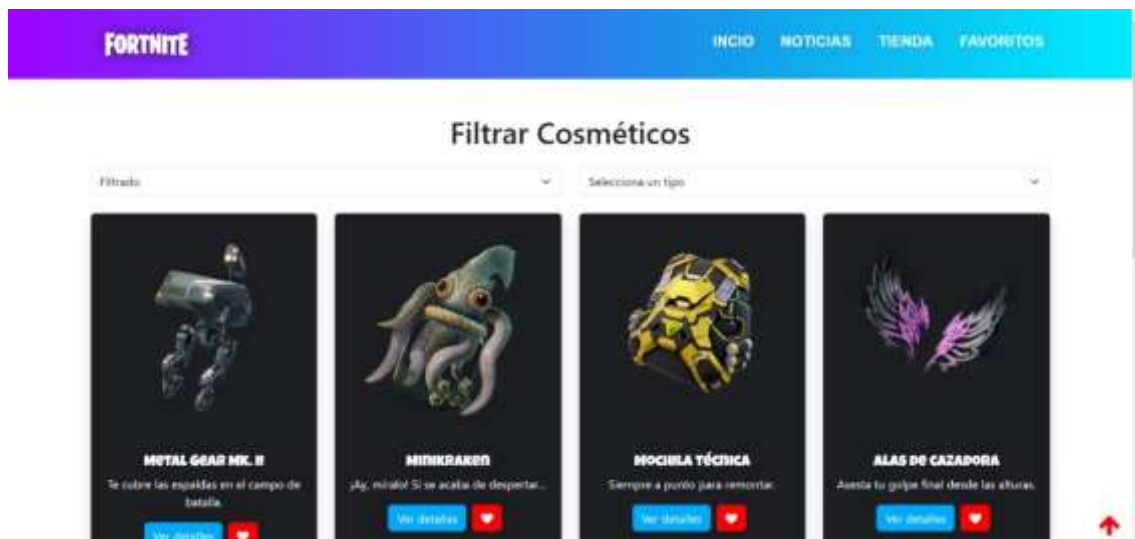
Tanto en el header y el footer hay enlaces a las demás páginas y en el footer hay iconos de las redes sociales que te llevan a las redes sociales de Fortnite.



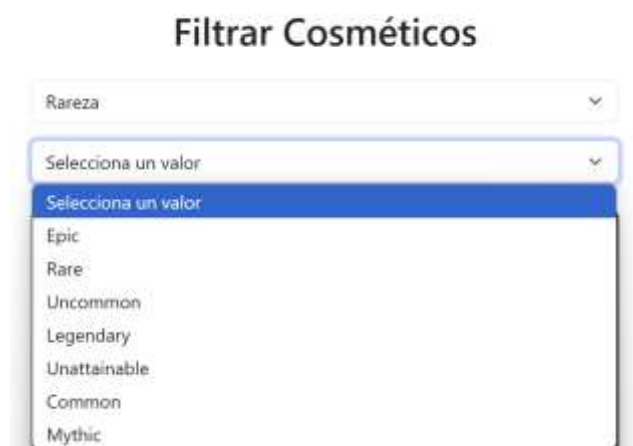
Si vamos a la página de noticias, podemos ver todas las noticias y el mensaje del día. Y si pulsamos a la noticia tenemos más detalles sobre la noticia.



En la tienda tendremos todos los productos nuevos. En cada objeto hay un botón para ver detalles del producto, y un botón de me gusta, donde el usuario puede ir almacenando los productos que le gustan, y al inicio de la página, en el menú hay un enlace a todos tus favoritos, donde puede ver todos los productos favoritos que tiene guardados y eliminarlos si quiere. Y cuenta con un botón en forma de flecha en la parte inferior para mayor comodidad para el usuario, el cual sube al principio de la página.



Además, la aplicación permite al usuario realizar búsquedas utilizando filtros. Permite realizar búsquedas de cosméticos en dos categorías diferentes: Rareza o capítulo y según cual elijas puedes filtrar entre las opciones de cada uno.



Porcentaje de participación de los integrantes del grupo

Nombre	Participación
Elena Díaz Pérez	33%
Alejandro Navarro Fernández	33%
Pablo Álvaro Serrano	33%