Modulo 3. Taller 1.

Informe de Investigación: Generalidades del Lenguaje JavaScript (2 puntos)

JavaScript(Js) fue creado por Brendan Eich en 1995, en solo 10 días, como un lenguaje de scripting para Netscape Navigator. Aunque fue diseñado inicialmente para agregar interactividad básica a los sitios web.

Anterior a la implementación de Js, cada vez que se refrescaba la pagina se tenía que solicitar al servidor toda la información para

que desplegase en el navegador. Dicho proceso era lento y muy demandante para el servidor.

El código escrito en Js como tal es ejecutado en modo interpretado mediante el mismo navegador utilizando un motor de JavaScript, sin necesidad de incluir herramientas adicionales.

Js se puede ejecutar en el navegador para dar interactividad y funcionalidad a la pagina web como en el servidor, en el cual permite desarrollo de aplicaciones back end utilizando Node.Js

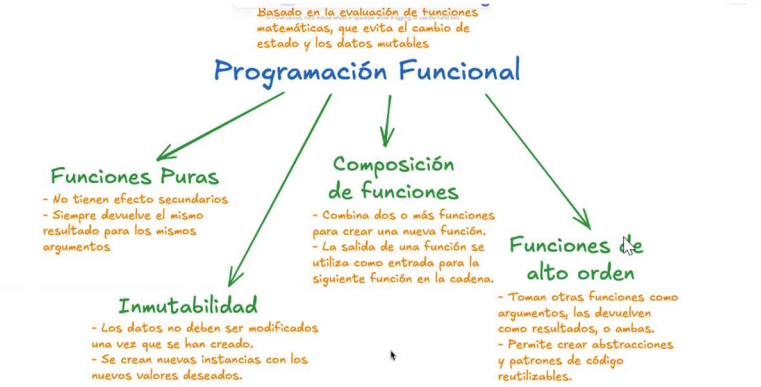
La diferencia entre JS y otros lenguajes se puede ver en la siguiente tabla

Caracteristica	JavaScript (Js)	C++	Python
Compilación	Interpretado	Compilado	Interpretado
Hilos	Monohilo	Multihilo	Multihilo
Paradigmas	Multiparadigma	Orientado a Objetos	Multiparadigma
Uso	Web, Servidores	Web, Servidores, Compiladores	IA, Ciencia de datos.

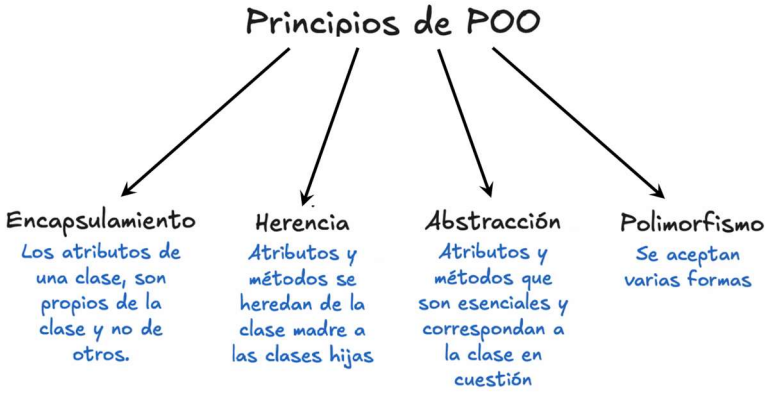
Dentro de los paradigmas que soporta Js están :

Imperativo. Secuencia de instrucciones paso a paso.

Funcional. Funciones como valores de primera clase y funciones puras.



Orientado a Objetos. Uso de clases para organizar el código. El cual se basa en los siguientes principios.



Dentro de las fortalezas se pueden mencionar.

Flexibilidad, facilidad para interactuar con el DOM, soporta la programación asincrónica, lo que permite operaciones sin bloquear el hilo principal.

Dentro de las debilidades.

Es un lenguaje no fuertemente tipado, lo cual puede generar error que sólo se detectan en ejecución del código.

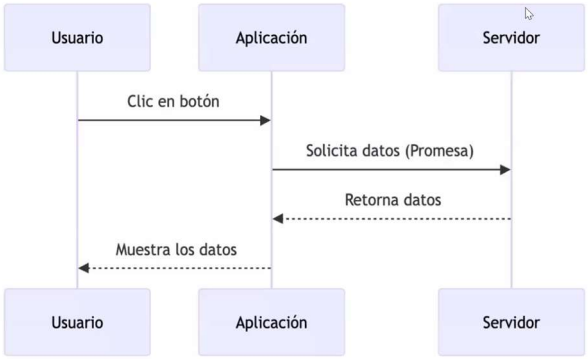
La forma de implementar la programación asincrónica es mediante callbacks, promesas y async/await.

Los callbacks son funciones que se pasan como argumentos a otras funciones, ejecutándose una vez que la operación termina.

```
function obtenerDatos(callback) {
  setTimeout(() => {
    callback("Datos obtenidos");
  }, 1000);
}

obtenerDatos((resultado) => {
  console.log(resultado); // "Datos obtenidos"
});
```

Las promesas representan el resultado futuro de operación asíncrona. Async/Await simplifica su uso, permitiendo un flujo más lineal y fácil de entender.



Evolución del Lenguaje JavaScript y el Estándar ECMAScript (2 puntos)

Js es un lenguaje interpretado, vale decir se ejecuta línea a línea por el motor de javascript del navegador o del servidor. Distinto a lenguajes como c++ que mediante un proceso se llega a código máquina, el cual se puede ejecutar

Js esta basado en el estándar ECMAScript gestionado por ECMA International

Js es a implementación del estándar ECMAScript. ECMAScript define el comportamiento y las funcionalidades del lenguaje.

La evolución de ECMAScript se puede resumir en:

ES3 (1999). Versión base de los navegadores.

ES5 (2009). Introdujo JSON nativo, strict mode y mejoras en la manipulación de objetos.

ES6 (2015). Introdujo let, const, clases, modulo, promesas y funciones flecha.

ES9 (2018). Mejora la gestión asíncrona promise, finally() y operaciones spread/rest.

TypeScript es un superconjunto de Js que añada tipado estático opcional, lo que facilita el desarrollo de aplicaciones de mayor envergadura al detectar errores en tiempo de desarrollo.

Dentro de las ventajas se encuentran.

1.-Tipado estático. Lo que permite detectar errores en desarrollo y no en ejecución.

2.-Legibilidad mejorada en el código. La declaración explicita, facilita a los desarrolladores comprender el tipo de datos esperado.

3.-Mantenimiento más sencillo. Debido al tipado estático y legibilidad el mantenimiento de los proyectos se vuelve más sencillo.
4.-Amplia comunidad. Este lenguaje cuenta con una gran comunidad activa de desarrolladores, lo cual también implica que existe una gran cantidad de recursos, librerías y herramientas para ayudar al desarrollador.
Dentro de las desventajas.
1.- Curva de aprendizaje. Se requiere un tiempo para los desarrolladores Js puros para que se acostumbren a la forma de programar.
2.- Necesidad de transpilacion. Es necesario llevar el código typescript a Js para que pueda ser ejecutado.
3.- Documentación más limitada que en Js, lo cual puede implicar que encontrar soluciones a problemas específicos sea más desafiante.

Análisis de la Pertinencia de Integrar JavaScript Avanzado o TypeScript en el Proyecto (3 puntos).

La Web del hospital tiene varias secciones que hacen pertinente incorporar codificación, a saber.
Formulario de contacto. Implica validar datos y poder guardarlo en una base de datos para poder realizar un seguimiento.
Los médicos y su especialización. En este caso, también es necesario disponer de dicha información en un servidor de base de datos y poder recuperarlo mediante el uso de Apis. Recomendaría una base de datos Sql.
Los comentarios de pacientes. En este también es necesario disponer de dicha información en un servidor de base de datos y poder recuperarlo mediante el uso de Apis. Una alternativa podría ser una base de datos NoSql.
En relación con el uso de Javascript avanzado vs TypeScript. De base recomendaría el uso de TypeScript ya que ayudaría a detectar error de tipo y asignación al momento de codificar y no al momento de ejecución. Si bien es cierto se necesita una pequeña curva de aprendizaje. El tiempo invertido vale la pena.

Dentro de las posibles dificultades esta en que es necesario disponer de una solución que no sólo involucra la interfaz (front end) sino que también un servidor de base de datos (back end) y un desarrollo que permite disponibilizar los datos mediante APIs.