

## UD1

# Tarea: Selección de arquitectura y herramientas de programación

## Tarea 1: Lenguajes Interpretados Vs Lenguajes Compilados

Realice un **resumen** explicando cada una de ellas y mencione las **ventajas y desventajas** de estos 2 tipos/clasificación de los lenguajes de programación. ¿En qué categoría se sitúa JavaScript?.

Muestre algún ejemplo y trate de que se le queden bien claros y asimilados estos 2 conceptos.

**Lenguaje interpretado:** Es el lenguaje cuyo código no necesita ser procesado por un compilador, es decir, el ordenador es capaz de ejecutar las instrucciones del código sin tener que traducirlo.

Ejemplos de lenguajes interpretados: PHP, ASP, Perl, Python, TCL y Bash.

VENTAJAS	DESVENTAJAS
Son multiplataforma	Se requiere un interprete
Son más sencillos de probar	Son más lentos
Los errores se detectan más fácilmente.	El código fuente es público

**Lenguaje compilado:** Es lo contrario al lenguaje interpretado. Éste necesita de un programa intermediario llamado compilador que traduzca el código para el que ordenador sea capaz de entenderlo y ejecutar las instrucciones.

Ejemplos de lenguajes compilados: ADA, Basic, C, Cobol.

VENTAJAS	DESVENTAJAS
Preparado para ejecutarse	No son multiplataforma
Son más rápidos	Poco flexibles
El código fuente es inaccesible	Se requiere un paso extra (la compilación)

JavaScript es un lenguaje de programación interpretado.

## Tarea 2: Lenguajes Imperativos Vs Lenguajes Declarativos

Realice un **resumen** explicando cada una de ellas y mencione las **ventajas y desventajas** de estos 2 tipos/clasificación de los lenguajes de programación. ¿En qué categoría se sitúa JavaScript?.

Muestre algún ejemplo y trate de que se le queden bien claros y asimilados estos 2 conceptos.

**Nota:** Los declarativos no son tan obvios como los imperativos!!!

**Lenguajes declarativos:** Son aquellos que dan las ordenes de lo que se debe de hacer pero no especifican cómo hacerlo. La solución se se obtiene mediante mecanismos internos de control.

Algunos ejemplos de lenguaje declarativo son: SQL, HTML, XML.

Las ventajas de los lenguajes declarativos depende del tipo de lenguaje que se vaya a utilizar, pero la ventaja más importante es que al decir que es lo que hay que hacer y no como hay que hacerlo, perdura mucho más. También son más compactos y son reutilizables.

La principal desventaja de este tipo de lenguaje es que esta restringido a un subconjunto de problemas para los que el intérprete fue diseñado.

**Lenguajes Imperativos:** son un tipo de lenguaje de programación que tiene como característica es que se basa en procedimientos, o sea, dar instrucciones para que realice algo, su característica es el uso de comandos y variables. Las instrucciones se ejecutan de manera secuencial (una por una).

Algunos ejemplos de lenguajes imperativos son Cobol, Pascal, Algol, Basic.

Sus ventajas son que es relativamente sencillo de implementar, es reutilizable, es muy modular y estructurado y necesita menos memoria.

Sus desventajas son que no tiene seguridad en los datos, tiene dificultad para relacionarse con los objetos del mundo real, es difícil crear nuevos tipos de datos y se da importancia a las operaciones de los datos en vez de a éstos.

## Tarea 3: Varios códigos fuentes .js !!!

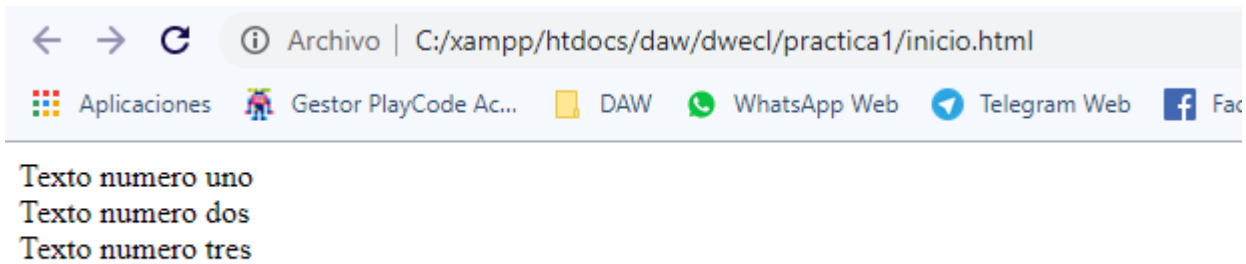
Realice una prueba de cómo cargar código javascript desde más de un archivo .js en una misma página HTML. Muestre capturas de pantalla de los códigos y empleados y que verifiquen su funcionamiento en la web.

```
<> inicio.html JS uno.js × JS dos.js JS tres.js
daw > dwed > practica1 > JS uno.js
1 document.write('Texto numero uno');
2
```

```
<> inicio.html JS uno.js JS dos.js × JS tres.js
daw > dwed > practica1 > JS dos.js
1 document.write('Texto numero dos');
2
```

```
<> inicio.html JS uno.js JS dos.js JS tres.js ×
daw > dwed > practica1 > JS tres.js
1 document.write('Texto numero tres');
2
```

```
<> inicio.html × JS uno.js JS dos.js JS tres.js
daw > dwed > practica1 > <> inicio.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title></title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <link href="css/style.css" rel="stylesheet">
8   </head>
9   <body>
10    <script src="uno.js"></script>
11    <br>
12    <script src="dos.js"></script>
13    <br>
14    <script src="tres.js"></script>
15  </body>
16 </html>
17
```



```
<script type="text/javascript" src="jsfile.js"></script>
```

#### Tarea 4: ¿Dónde es más recomendable incluir código javascript y por qué?

Trate de buscar en foros, discusiones, artículos información que se discuta las ventajas y desventajas de incluir código javascript en el head o en el body. Trate de razonar y ver los pros y contras, y/o los diversos casos en los que un programador se puede ver en esta disyuntiva.

Hay que evitar poner los scripts en el header. El renderizado e interpretación del html se realiza a medida que el navegador encuentra los elementos en el documento html. Por lo tanto si encuentra en la cabecera una etiqueta <script>, tendrá que esperar a que el navegador cargue el script para continuar con el renderizado de la página por lo que se vería el navegador con la pantalla en blanco, comportamiento que uno desea evitar.

#### Tarea 5: ¿Cómo podemos depurar / debugear un código JavaScript haciendo uso de Mozilla Firefox? “Ejercicio Estrella”

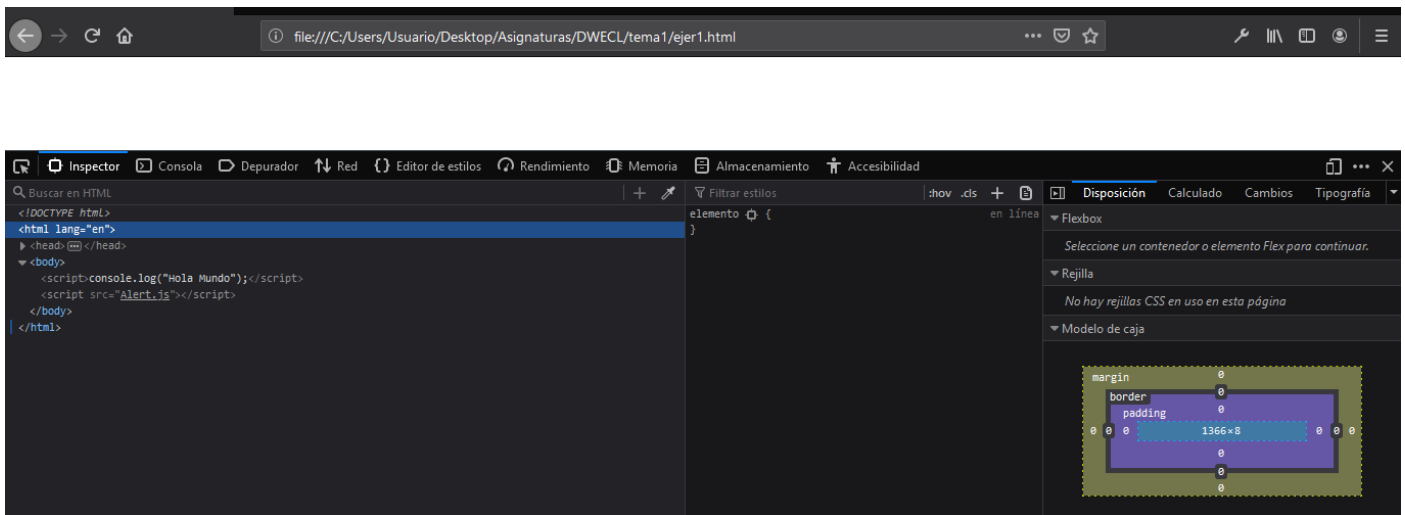
Encuentre información y realice un ejercicio práctico para demostrar que haya adquirido esta destreza. Añada alguna captura. Le recomiendo que para hacer las capturas busque un código Javascript que contenga alguna variable, y cambios en ella, de forma que pueda ver y mostrar por capturas como en la pestaña de depuración del Firefox se va modificando el valor. ¿Podemos cambiar el valor de una variable en tiempo de ejecución haciendo uso del depurador?

*Sin duda, todo programador debe de aprender a depurar bien su técnica de depuración, valgase de la redundancia.*

1º Paso:

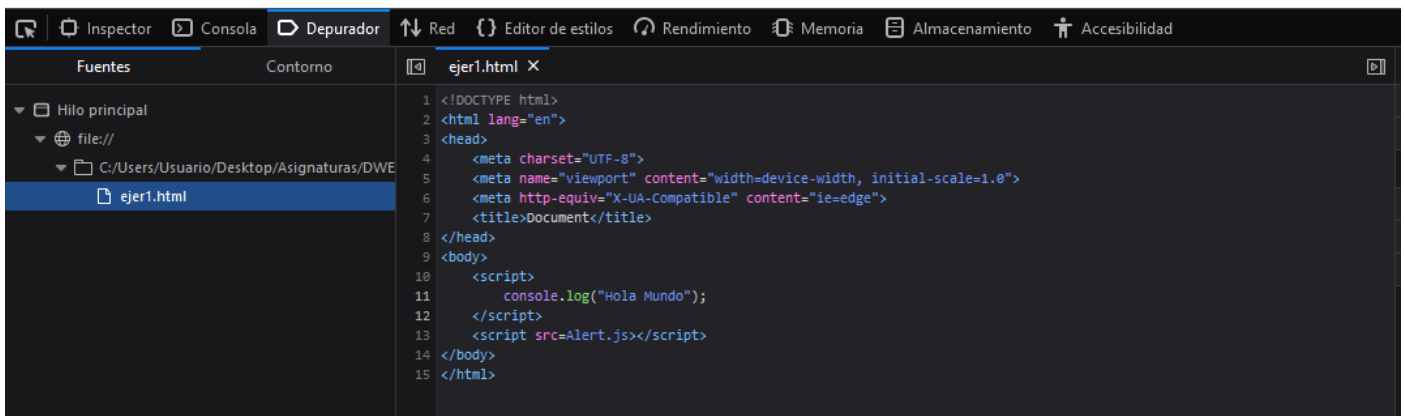
Abrimos la página web con mozilla y una vez abierta damos click derecho → Inspeccionar elemento.

Nos aparecerá lo siguiente:



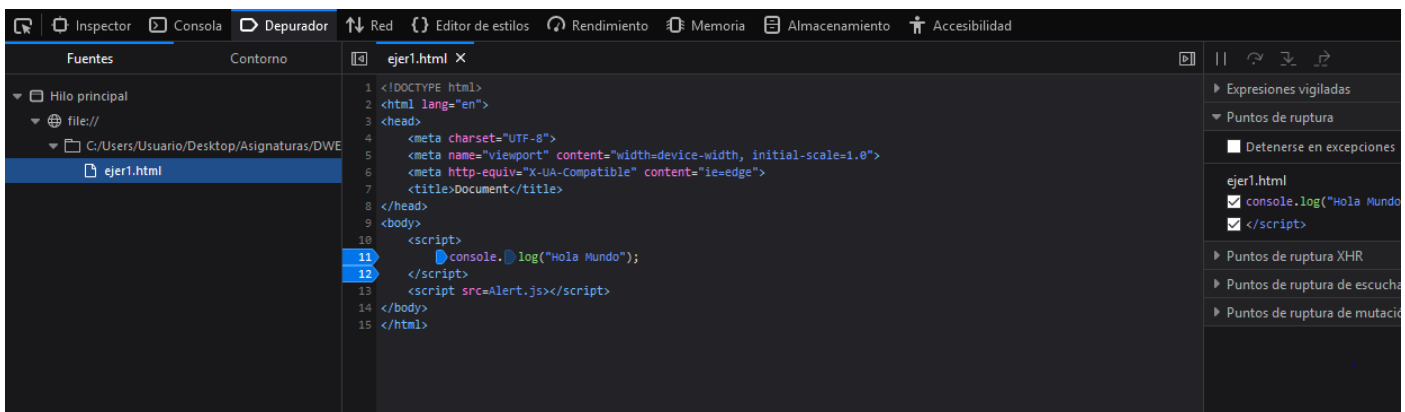
## 2º Paso:

En el menú que nos aparece (donde esta seleccionado Inspector), nos aparece la opción de Depurador. Clickamos en ella y se nos cambiará la ventana de abajo.



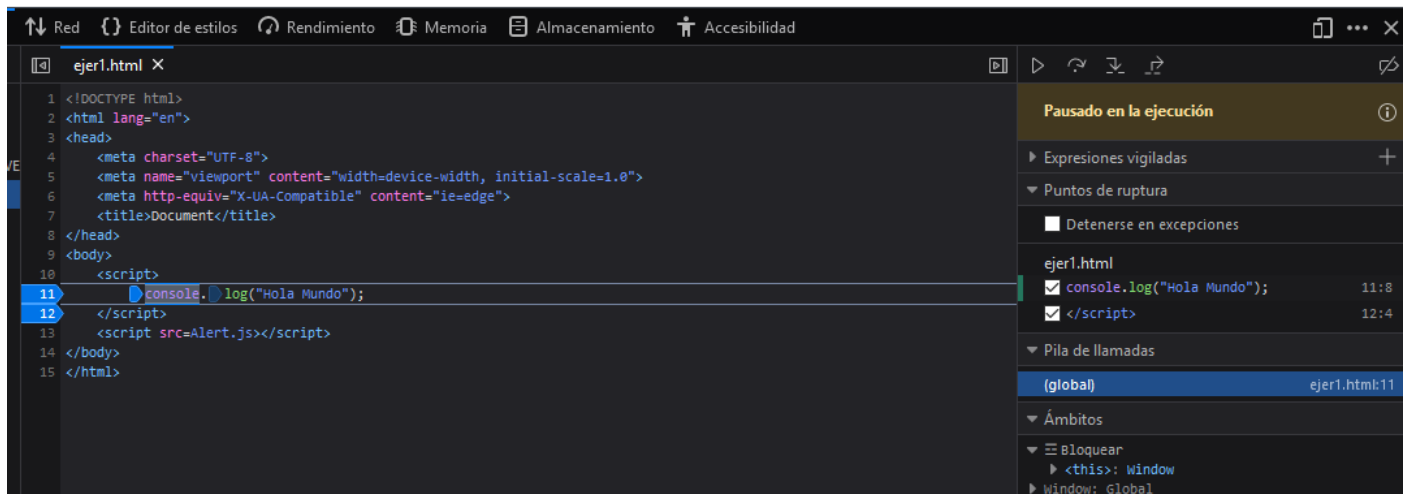
## 3º Paso:

Debemos de seleccionar el fichero donde se encuentra el código y a continuación colocamos los breakpoints (delimitaciones del código a debuggear).



#### 4º Paso:

Recargamos la página para que empiece el debuggeo y listo.



### Tarea 6: Busque información sobre los distintos y más usados frameworks/librerías utilizados hoy día para desarrollar funcionalidad propia de JavaScript.

Hay mucha información sobre estos frameworks en la web. De lo que trata la tarea es de alcanzar una visión global y observar cuales son tendencia en el mundo actual. Lo discutiremos en clase.

**Angular.js:** Ordena las aplicaciones JavaScript y potencia las Arquitecturas SPA. (Google)

**React.js :** La librería de Facebook orientada a la gestión de del UI.

**Meteor.js :** Apoya el concepto de aplicaciones JavaScript Isomórficas, aquellas aplicaciones que pueden ejecutar su código JavaScript en cliente y servidor.

**Moment.js :** La librería de Javascript orientada a la gestión de fechas .

**jQuery.js :** Permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

**Modernizr:** Se encarga de validar si nuestro navegador soporta una u otra característica de HTML 5, permitiendo a los desarrolladores actuar en consecuencia.

**Video.js :** Framework de referencia para la gestión de Videos a través de JavaScript.

