



Practica No. 4

Funciones en Javascript

Nombre(s):

Martin Isai Nuñez Villeda

Paola Monserrat Osorio

Objetivo:

Con la realización de esta práctica se pretende: definir e implementar funciones en una página web mediante el lenguaje de programación Javascript.

Fundamento Teórico:

Funciones en JavaScript

Las funciones en JavaScript son una manera de escribir una acción de la que se obtiene un resultado según ciertos parámetros, como su nombre indica, las funciones permiten obtener un resultado en función de lo que se defina, entonces, con este tipo de elemento, se pueden definir unos valores (**in**) que pueden transformarse con la función para obtener un resultado (**out**).

En JavaScript, hay dos tipos de funciones, las funciones de tipo void son aquellas que, como su propio nombre indica, están vacías y no tienen parámetros in. Esto hace que las funciones no devuelvan nada, pues no tienen nada con lo que trabajar.

Por ello, no estará presente la palabra clave return. Las demás funciones sí tienen contenido, por lo que devuelven algún elemento. Entonces es posible, por ejemplo, tener dos parámetros in, como nombre y empleo, que se transforman dentro de la función y devuelven como resultado un string.

Escribir funciones en JavaScript es muy útil para simplificar acciones que deben repetirse varias veces en un proyecto. Sin embargo, es recomendable usar funciones en toda línea de código que genere una acción porque, en el peor de los casos, solamente se usan una vez, la misma vez que se programaría la acción sin ser función.

Esto, aparte de funciones en JS, también ayuda a crear trozos de código más cortos, que sean más manejables, tendrán menos errores y, además, serán menos costosos.

Formas de escribir funciones en JS:

Forma canónica

Se utiliza la palabra clave **function** seguida de un espacio y, a continuación, el nombre de la función. Después de definir su nombre, se escribe entre paréntesis las variables o parámetros de dicha función, después de esto, se tienen lo que sería el **body** de la función, introducido por las llaves { }.

```
1 function name(parameter1, parameter2, ... parameterN) {  
2   // body  
3 }
```



Funciones por medio de variables

Otra manera de escribir funciones en JavaScript es por medio de las palabras clave que se usan para declarar variables.

Entonces, en vez de utilizar la palabra clave function de función en JavaScript al inicio, se puede usar cualquiera de las palabras **var**, **let** y **const**.

Después de esta palabra, se inserta el nombre de la función en JavaScript y se le asigna un valor function.

A partir de la asignación de este valor, el resto de las líneas de código funcionan de la misma manera que en la forma canónica.

```
var time = new Date();  
console.log(time.getHours() + ":" + time.getMinutes() + ":" + time.getSeconds());
```

Para utilizar las funciones en JavaScript, basta con llamar a la función usando su nombre y determinar los valores de cada parámetro entre paréntesis.

Forma de trabajo:

Colaborativa en equipos de 3 personas

Material:

1. Computadora
2. IDE Netbeans

Procedimiento:

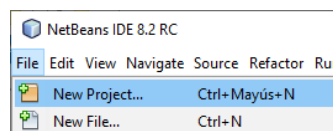
Se va a crear una página web a partir del IDE Netbeans que ejecute una serie de operaciones específicas mediante funciones.

El objetivo es crear una aplicación web que incluya la implementación y el uso de funciones para la ejecución de operaciones varias y familiarice al alumno con la declaración y ejecución de funciones en javascript.

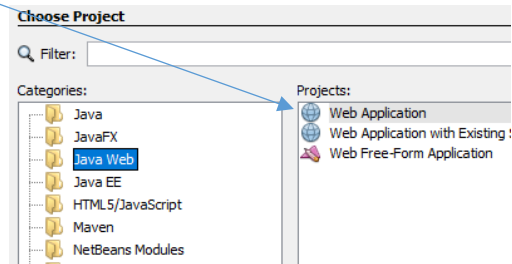
Desarrollo

Para la creación de la página web deberán realizarse los siguientes pasos:

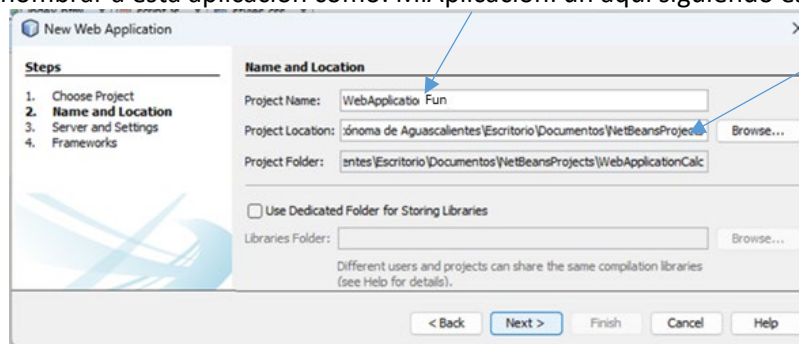
1. Abrir el IDE Netbeans.
2. crear un proyecto nuevo



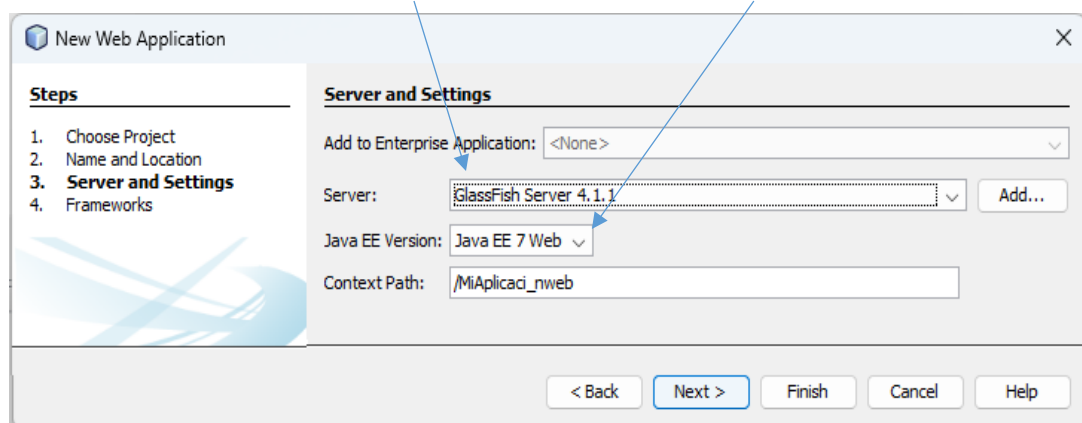
3. Seleccionar Java Web > Web Application



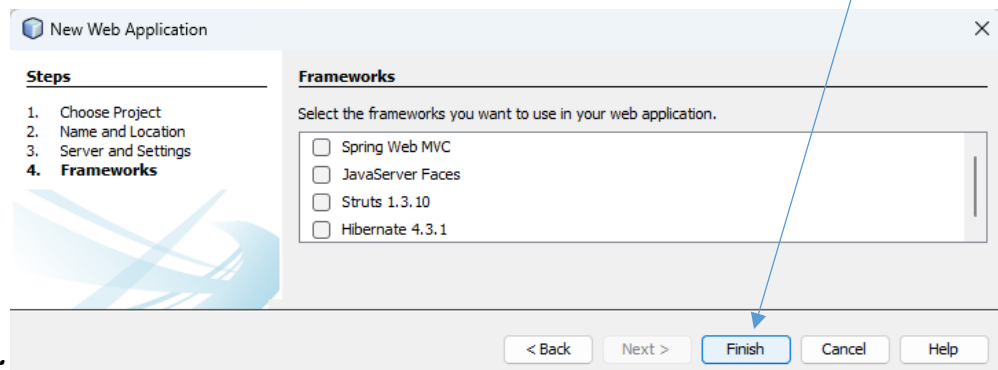
4. Se va a nombrar a esta aplicación como: MiAplicacionFun aquí siguiendo este camino



5. Se debe seleccionar al servidor GlassFish y la versión Java EE 7 web.



6. Para esta aplicación no se va a seleccionar una estructura y se termina:



7. A continuación, agrega las siguientes funciones a tu aplicación, considerando los datos de cada uno de los integrantes del equipo:

```
function miNombre (nombre, apellido, edad) {  
    return 'Me llamo ' + nombre + ' ' + apellido + ' y tengo ' + edad +  
        ' ' + 'años';  
}  
  
console.log( miNombre('Germán', 'Cutraro', 17) ); // Imprimimos el resultado  
  
// → Me llamo Germán Cutraro y tengo 17 años
```

Equivalencias:

```
var valor = 15.4351953;  
var dolarAPesos = function (dolar) {  
    return dolar * valor;  
};  
  
console.log( dolarAPesos(40) ); // Imprimimos el resultado  
  
// → 617.407812
```

índice de masa corporal IMC:

```
function bmi(weight, height) {  
    return weight / height**2  
}  
  
console.log("Tu IMC es: " + bmi(80, 1.8));
```

Definir tamaño de cadenas:

```
let nombre = 'Jose';  
console.log(nombre.length); //retorna un 4
```

Concatenar cadenas:

```
let nombre = 'Jose';  
console.log(nombre.concat(' ', 'Aguilar')); //retorna Jose Aguilar
```

Convertir a mayúsculas:

```
let nombre = 'Jose';  
console.log(nombre.toUpperCase()); //Resultado: JOSE
```

Dividir cadenas:

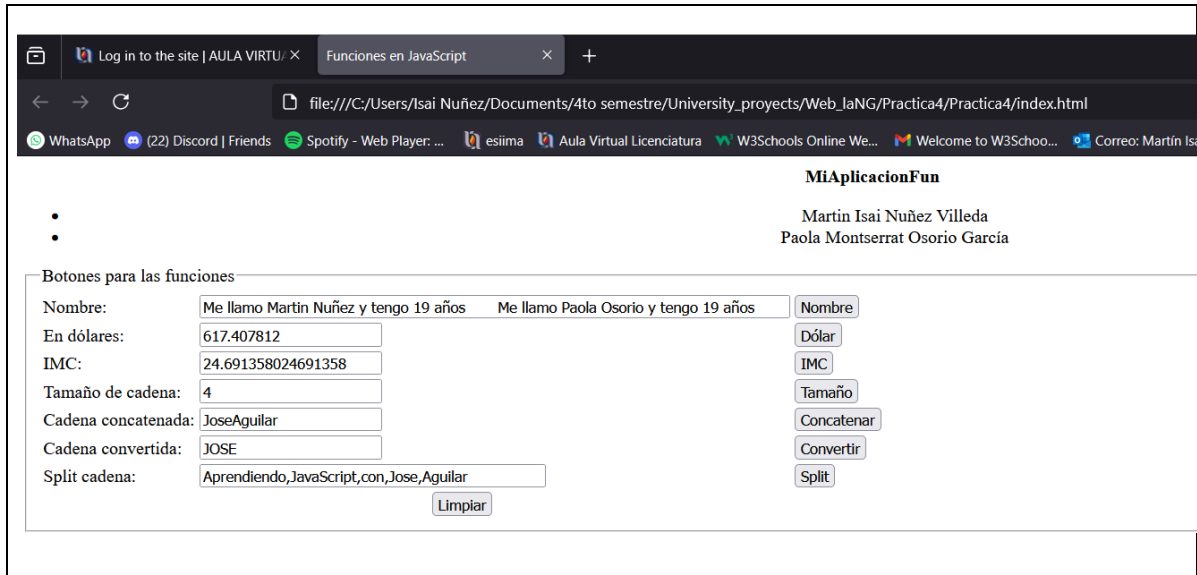
```
let mensaje = "Aprendiendo JavaScript con Jose Aguilar";  
console.log(mensaje.split(' '));
```

8. Invoca las funciones definidas.



Resultados:

Realizar una corrida de prueba y mostrar cómo se ve la página en el navegador.



Una vez terminado el programa debe subirse el archivo index.html a la plataforma de **aulavirtual** junto con este reporte.

Conclusiones:

En esta práctica, exploramos la integración de HTML con JavaScript para el desarrollo backend, enfocándonos en la gestión de funciones dentro de la página.

La implementación de funciones podría mejorar la interactividad y eficiencia del sitio web.

En conclusión, combinar estas tecnologías nos permite crear aplicaciones dinámicas y funcionales.