

## Laboratorio 03

### Tema: JavaScript

Docente	Escuela	Asignatura
M.Sc. Ing. Richart Smith Escobedo Quispe rescobedoq@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web Semestre: III Código: 1702122

Laboratorio	Tema	Duración
03	JavaScript	06 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	13 Mayo 2024	17 Mayo 2024

## Índice

<b>1. Especificaciones del Laboratorio</b>	<b>2</b>
1.1. Objetivos del curso . . . . .	2
1.2. Objetivos del laboratorio . . . . .	2
1.3. Equipos, materiales y temas . . . . .	2
<b>2. Marco teórico</b>	<b>2</b>
2.1. JavaScript . . . . .	2
<b>3. Guía de laboratorio</b>	<b>6</b>
3.1. Directorio de trabajo personal . . . . .	6
3.2. Repositorio local . . . . .	6
3.3. Directorio del laboratorio . . . . .	6
3.4. Archivo .gitignore . . . . .	7
<b>4. Tarea</b>	<b>7</b>
4.1. Descripción . . . . .	7
4.2. Pregunta . . . . .	8
4.3. Entregables . . . . .	10
<b>5. Rúbricas</b>	<b>11</b>
5.1. Sobre el Informe . . . . .	11
5.2. Contenido del Informe . . . . .	11
<b>6. Referencias</b>	<b>13</b>

## 1. Especificaciones del Laboratorio

### 1.1. Objetivos del curso

- Proporcionar los conocimientos y habilidades para el uso de las principales metodologías de análisis y diseño de sistemas.
- Brindar los conocimientos para la utilización de técnicas para el análisis y diseño de sistemas web.
- Proporcionar conocimientos y habilidades en el manejo de herramientas para el desarrollo de sistemas Web.
- Desarrollar sistemas de información dentro de una arquitectura cliente servidor.

### 1.2. Objetivos del laboratorio

- Utilizar el sistema de control de versiones Git.
- Implementar ejercicios utilizando JavaScript puro.

### 1.3. Equipos, materiales y temas

- Sistema Operativo (GNU/Linux de preferencia).
- Editor de texto plano (GNU Vim de preferencia).
- Navegador Web: Chrome, Firefox, Edge, Brave, Opera, etc.
- Documentación y librería JQuery.
- Git.
- Cuenta en GitHub asociada al correo institucional.

## 2. Marco teórico

### 2.1. JavaScript

- JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo *just – in – time* con funciones de primera clase.

#### focus()

- El método focus() da foco a un elemento (si se puede enfocar).
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_html\\_blur](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_html_blur)

#### click()

- El método click() simula un clic del mouse en un elemento.
- Este método se puede utilizar para ejecutar un clic en un elemento como si el usuario hiciera clic manualmente en él.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_html\\_click](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_html_click)

## innerHTML

- La propiedad InnerHTML establece o devuelve el contenido HTML (HTML interno) de un elemento.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_element\\_innerhtml\\_ul](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_element_innerhtml_ul)

## setAttribute()

- El método setAttribute() establece un nuevo valor para un atributo.
- Si el atributo no existe, se crea primero.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_element\\_setattribute1](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_element_setattribute1)

## getElementById()

- El método getElementById() devuelve un elemento con un valor especificado.
- El método getElementById() devuelve nulo si el elemento no existe.
- El método getElementById() es uno de los métodos más comunes en HTML DOM. Se utiliza casi cada vez que desea leer o editar un elemento HTML.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_document\\_getelementbyid1](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_document_getelementbyid1)

## addEventListener()

- El método addEventListener() adjunta un controlador de eventos a un documento.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_document\\_addeventlistener2](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_document_addeventlistener2)

## createAttribute()

- El método createAttribute() crea un atributo y lo devuelve como un objeto Attr.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_document\\_createattribute1](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_document_createattribute1)

## forms

- La propiedad de formularios devuelve una colección de todos los elementos <form> de un documento.
- La propiedad de formularios devuelve una HTMLCollection.
- La propiedad de formularios es de solo lectura.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_doc\\_forms5](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_doc_forms5)

## onclick

- El evento onclick ocurre cuando el usuario hace clic en un elemento HTML.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_onclick](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_onclick)

## fetch()

- El método `fetch()` inicia el proceso de obtener un recurso de un servidor.
- El método `fetch()` devuelve una Promesa que se resuelve en un objeto Respuesta.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_api\\_fetch\\_easy](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_api_fetch_easy)

## CANVAS

- El elemento `<canvas>` define un área de mapa de bits en una página HTML.
- La API de Canvas permite que JavaScript dibuje gráficos en el lienzo.
- La API Canvas puede dibujar formas, líneas, curvas, cuadros, texto e imágenes, con colores, rotaciones, transparencias y otras manipulaciones de píxeles.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryhtml5\\_canvas\\_stroke](https://www.w3schools.com/jsref/tryit.asp?filename=tryhtml5_canvas_stroke)

## Input Date Object

- El objeto Fecha de entrada representa un elemento HTML `<input>` con `tipo="fecha"`.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_input\\_date\\_get](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_input_date_get)

## Input Range Object

- El objeto Rango de entrada representa un elemento HTML `<input>` con `tipo="rango"`.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_range\\_get](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_range_get)

## Input Week Object

- El objeto Input Week representa un elemento HTML `<input>` con `tipo="semana"`.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_week\\_create](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_week_create)

## length

- La propiedad de longitud devuelve el número de elementos de una `HTMLCollection`.
- La propiedad de longitud es de solo lectura.
- La propiedad de longitud es útil cuando desea recorrer una `HTMLCollection`.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_htmlcollection\\_length\\_loop](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_htmlcollection_length_loop)

## forEach()

- El método `forEach()` llama a una función para cada elemento de una matriz.
- El método `forEach()` no se ejecuta para elementos vacíos.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_foreach](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_foreach)

### **map()**

- `map()` crea una nueva matriz llamando a una función para cada elemento de la matriz.
- `map()` no ejecuta la función para elementos vacíos.
- `map()` no cambia la matriz original.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_map2](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_map2)

### **eval()**

- El método `eval()` evalúa o ejecuta un argumento.
- Si el argumento es una expresión, `eval()` evalúa la expresión. Si el argumento es una o más declaraciones de JavaScript, `eval()` ejecuta las declaraciones.
- Ejemplo: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_eval](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_eval)

### 3. Guía de laboratorio

#### 3.1. Directorio de trabajo personal

- Cree su directorio de trabajo personal. (Debe utilizar su usuario institucional.)
- Luego, diríjase a este directorio.

Listing 1: Creando directorio personal

```
$ mkdir -p $HOME/rescobedoq/
```

Listing 2: Dirigiéndonos al directorio personal

```
$ cd $HOME/rescobedoq/
```

#### 3.2. Repositorio local

- Clone su repositorio remoto.
- Luego, diríjase a este directorio.

Listing 3: Clonando repositorio remoto

```
$ git clone https://github.com/rescobedoulasalle/pw2-24a.git
```

Listing 4: Dirigiéndonos al directorio del repositorio local

```
$ cd $HOME/rescobedoq/pw2-24a
```

#### 3.3. Directorio del laboratorio

- Cree un directorio para el laboratorio correspondiente, dentro del directorio para el repositorio local. (Utilizar el estándar **lab03**)
- Luego, diríjase a este directorio.

Listing 5: Creando directorio para este laboratorio

```
$ mkdir -p $HOME/rescobedoq/pw2-24a/lab03/exercises/
```

Listing 6: Dirigiéndonos al directorio del laboratorio

```
$ mkdir -p $HOME/rescobedoq/pw2-24a/lab03
```

### 3.4. Archivo .gitignore

- Siempre evalúe utilizar el archivo `.gitignore` para no hacer seguimiento a archivos innecesarios, esto es muy importante sobre todo para ignorarlos en el repositorio GitHub.
- Pueden haber varios de estos archivos y estar ubicados estratégicamente en algún directorio; por ejemplo sólo para el laboratorio o para todo el repositorio local.

## 4. Tarea

### 4.1. Descripción

- Programar en JavaScript sobre una página web html básica.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!-- no html, generate it with javascript -->
  <script src="script_ejercicio_01.js"></script>
</body>
</html>
```

- Ejercicio 01: Cree un teclado random para banca por internet.



 **Usted se encuentra en una zona segura**

Selección:  

Número de tarjeta:

Tipo y N° Documento:  

Ingresar tu clave usando el teclado virtual:

9	4	1
7	2	5
6	3	0
8	LIMPIAR	

 [Genera tu Clave de Internet](#)

Ingresar tu Clave de Internet (06 dígitos)

 [Olvidé mi clave](#)

Ingresar el texto de la imagen:



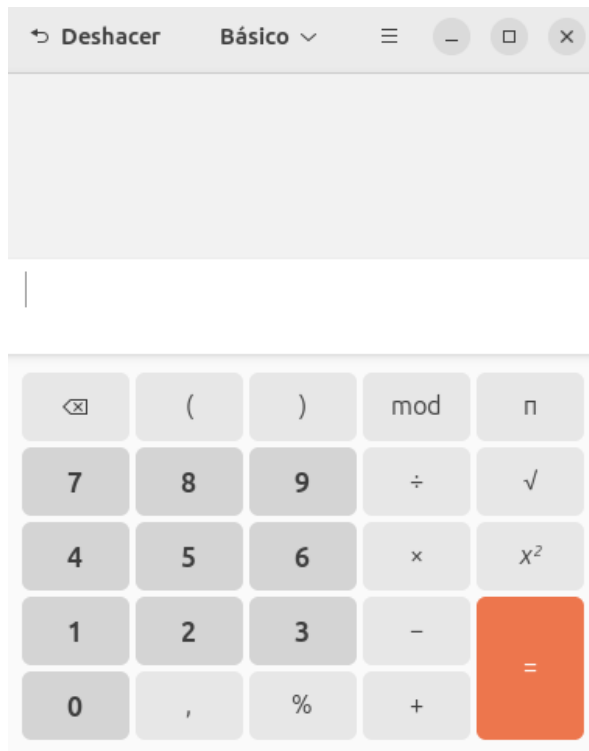
 [Cambiar texto](#)

**INGRESAR**

**Banco de la Nación | Ministerio de Economía y Finanzas**

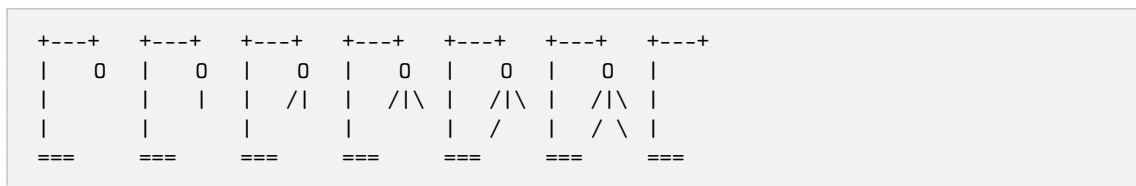
Oficina Principal: Av. Javier Prado Este 2499, San Borja, Central Telefónica: 519 2000.  
Atención en Oficinas Administrativas: Lunes a Viernes de 08:30 a 17:30, Refrigerio de: 13:00-14:00.  
Atención en Oficina de Trámite Documentario: Lunes a Viernes de 8:30 a 16:30 (horario corrido).

- **Ejercicio 02:** Cree una calculadora básica como la de los sistemas operativos, que pueda utilizar la función `eval()` y que guarde todas las operaciones en una pila. Mostrar la pila al pie de la página web.



14
$(x*x)+10$
$x=2$
4.5
$(1+8)/2$

- **Ejercicio 03:** Cree una versión de el juego 'el ahorcado' que grafique con canvas paso a paso desde el evento `onclick()` de un botón.



## 4.2. Pregunta

- Explique una herramienta para ofuzcar código JavaScript.



- Muestre un ejemplo de su uso en uno de los ejercicios de la tarea.
- Adjunte a su repositorio ambas versiones:
  - script\_ejercicio\_01.js (**development**).
  - script\_ejercicio\_01.min.js (**production**).

### 4.3. Entregables

- URL al directorio específico del laboratorio en su repositorio GitHub privado donde esté todo el código fuente y otros que sean necesarios. Evitar la presencia de archivos: binarios, objetos, archivos temporales, cache, librerías, entornos virtuales. Si hay configuraciones particulares puede incluir archivos de especificación como: packages.json, requirements.txt o README.md.
- No olvide que el profesor debe ser siempre colaborador a su repositorio que debe ser privado (Usuario del profesor **@rescobedoulasalle**).
- El informe debe estar elaborado en  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- Usted debe describir sólo los **commits** más importantes que marcaron hitos en su trabajo, adjuntando capturas de pantalla, del commit, porciones de código fuente, evidencia de sus ejecuciones y pruebas.
- En el informe siempre se debe explicar las imágenes (código fuente, capturas de pantalla, commits, ejecuciones, pruebas, etc.) con descripciones puntuales pero precisas.
- Agregar la estructura de directorios y archivos de su laboratorio hasta el nivel 4.

```
rescobedoq/  
|--- pw2-24a  
| |--- README.md  
| |--- .gitignore  
| |--- lab03  
| |   |--- README.md  
| |   |--- ejercicio01.html  
| |   |--- ejercicio02.html  
| |   |--- ejercicio03.html  
| |   |--- .gitignore  
| |   |--- css  
| |   |   |--- style.css  
| |   |--- js  
| |   |   |--- script.js  
| |   |--- exercises  
| |   |   |--- onclick.html  
| |   |   |--- map.html  
| |   |   |--- date.html  
| |   |--- latex  
| |       |--- rescobedoq_pw2_24a_lab03.tex  
| |       |--- rescobedoq_pw2_24a_lab03  
.pdf
```

## 5. Rúbricas

### 5.1. Sobre el Informe

Tabla 1: Rúbrica para el Informe

Informe		Cumple	No cumple
<b>L<sup>A</sup>T<sub>E</sub>X</b>	El informe está en formato PDF desde L <sup>A</sup> T <sub>E</sub> X, con un formato limpio (buena presentación) y fácil de leer.	20	0
<b>Observaciones</b>	Respetar la estructura de organización para ubicación de los entregables. Por cada observación dentro del informe se le descontará puntos. Se debe incluir el código fuente latex del informe (*.tex)	-	-

### 5.2. Contenido del Informe

- El alumno deberá autocalificarse, marcando o dejando en blanco las celdas de la columna **Checklist**, de acuerdo a si cumplió o no con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación siempre será sobre la nota mínima aprobatoria, siempre y cuando cumpla con todos los ítems. (Máximo 24 horas)
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la tabla de calificación de niveles de desempeño:

Tabla 2: Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
<b>2.0</b>	0.5	1.0	1.5	2.0
<b>4.0</b>	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y evidencias

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
<b>1. GitHub</b>	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o omisión)	4			
<b>2. Commits</b>	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4			
<b>3. Ejecución</b>	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4			
<b>4. Pregunta</b>	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2			
<b>7. Ortografía</b>	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2			
<b>8. Madurez</b>	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4			
<b>Total</b>		20			

## 6. Referencias

- <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>
- <https://www.w3schools.com/jsref/>