

Unidad 1. Arquitecturas y lenguajes de programación en clientes web.

Boletín de actividades

En la unidad se diferencian dos partes:

1. Arquitecturas y lenguajes para la programación sobre clientes web, en la que se tratan:
 - A) La arquitectura cliente-servidor.
 - B) Los lenguajes de programación para clientes web.
 - C) Herramientas para la programación.
 - D) La integración de lenguajes para clientes web.
2. Fundamentos de JavaScript, en la que se tratan:
 - A) Los comentarios de código.
 - B) Las variables y constantes.
 - C) Las conversiones entre tipos de datos.
 - D) Las estructuras de control.
 - E) Los bucles.
 - F) La depuración de código.

1. Arquitecturas y lenguajes para la programación sobre clientes web

Apartado A. La arquitectura cliente-servidor.

Actividad 1.

El modelo de funcionamiento de la web se basa en la arquitectura cliente-servidor. En ella están presentes algunos conceptos clave, defínelos:

1. Servidor Web.
2. Cliente Web.
3. Red.

Actividad 2.

Realiza un diagrama de flujo que represente la comunicación entre un cliente y un servidor web.

Apartado B. Los lenguajes de programación para clientes web.

Actividad 1.

Responde a la siguiente pregunta de reflexión en el foro de la unidad 1:

A la hora de escoger un lenguaje de programación para un proyecto, ¿se debería elegir una tecnología que ya se conoce o aquella que mejor se adapte al proyecto?

Actividad 2.

De los siguientes lenguajes, indica cuáles se ejecutan en el lado del cliente (front-end) y cuáles en el lado del servidor (back-end):

JavaScript, Python, PHP, HTML, TypeScript, CSS, SQL, Java

Actividad 3.

No todos los lenguajes de programación para clientes web se utilizan con el mismo propósito. Indica cuál es la finalidad de los lenguajes más populares.

Actividad 4.

Para programar aplicaciones del lado del cliente nos centraremos en el lenguaje JavaScript, ¿crees que es buena idea esforzarse en aprender este lenguaje en concreto?

Echa un vistazo a los resultados de la encuesta anual de *stackoverflow* sobre las principales tecnologías y lenguajes de programación a nivel mundial. (<https://insights.stackoverflow.com/survey/> apartado *Technology*).

Actividad 5.

Indica las principales ventajas y limitaciones de un lenguaje de programación para clientes web como JavaScript.

Apartado C. Herramientas para la programación.

Actividad 1.

Los navegadores son los clientes web por excelencia. Determina cuáles son los navegadores más utilizados en la actualidad y haz una tabla resumen que recoja:

- a) Las plataformas con las que son compatibles y
- b) El motor JavaScript que utilizan.

Fuentes recomendadas:

- [Genbeta](#).
- [PCWorld](#).

Actividad 2.

Los navegadores incluyen herramientas para que los desarrolladores prueben y depuren el código de sus aplicaciones. Indica cómo acceder a estas herramientas en los principales navegadores.

Actividad 3.

Para programar una aplicación web es suficiente con un simple editor de texto pero hoy en día disponemos de programas más avanzados que facilitan la tarea de la programación. Realiza las siguientes actividades sobre los editores de código:

1. ¿Qué características deberían tener estos programas para facilitar la programación web?
2. Indica cuáles son los editores de código más utilizados para la programación web (consulta la encuesta anterior de *stackoverflow*).
3. Evalúa varios editores y escoge uno de ellos para tus desarrollos.

Actividad 4.

Una vez que terminada nuestra aplicación es posible ofuscar el código para protegerlo, ¿qué es ofuscar el código?

Actividad 5.

Responde a la siguiente pregunta de reflexión en el foro de la unidad 1:

En lugar de ofuscar, hay desarrolladores que prefieren proteger el código ofreciéndolo a una comunidad bajo alguna licencia de software libre, ¿qué opción te parece más adecuada?

Apartado D. La integración de lenguajes para clientes web.

Existen varias formas de integrar código JavaScript en documentos HTML:

- a) Insertando código JavaScript en el propio documento HTML.
- b) Haciendo referencia en el HTML a un documento externo con código JavaScript.

Actividad 1.

Crea dos aplicaciones web que incorporen el siguiente código JavaScript siguiendo el primero y el segundo método de integración respectivamente.

```
alert("Hola, soy un mensaje escrito por una función JavaScript");
```

Actividad 2.

De forma general, ¿cuál de los dos métodos de integración es recomendable seguir? ¿Por qué?

2. Fundamentos de JavaScript.

Actividad 1.

Como no podía ser de otra manera, crea una aplicación que muestre por consola el mensaje:
¡Hola mundo!

Apartado A. Los comentarios de código.

Actividad 1.

¿Qué son los comentarios de código? ¿para qué sirven?

Actividad 2.

JavaScript admite comentarios de una o varias líneas. Escribe ambos tipos de comentarios en una aplicación.

Apartado B. Variables y constantes

Actividad 1.

¿Para qué sirven las variables en un programa?

Actividad 2.

En JavaScript pueden crearse dos tipos de variables utilizando `var` o `let` para declararlas, ¿en qué se diferencian unas de otras?

Actividad 3.

Crea una aplicación en la que se observe claramente la diferencia entre declarar una variable con `var` y `let`.

Nota: los *identificadores* en JavaScript deben empezar con una letra, dólar o guión bajo y pueden contener letras, números, el signo de dólar o el de guión bajo. Los identificadores no pueden ser palabras reservadas.

Actividad 4.

En JavaScript también pueden crearse constantes, ¿en que se diferencia una variable de una constante?

Actividad 5.

Crea una aplicación en la que observe claramente la diferencia entre una variable y una constante.

Apartado C. Las conversiones entre tipos de datos.

Actividad 1.

En ocasiones el tipo de los datos puede dar lugar a errores en las operaciones. JavaScript proporciona dos funciones para convertir cadenas a números: `parseInt` y `parseFloat`. Convierte a un número decimal y a un número entero respectivamente las siguientes cadenas: “11.1” y “22”.

Apartado D. Las estructuras de control.

Actividad 1.

Implementa una aplicación que muestre un mensaje al usuario si dos notas de un cuestionario cumplen las siguientes condiciones:

- a) Son iguales o distintas.
- b) Las dos son mayores o iguales a 100.
- c) alguna es menor que 50.

Actividad 2.

Implementa una aplicación que muestre un mensaje al usuario con su nivel de inglés en función del código del mismo: A2 (Elementary), B1 (Lower Intermediate), B2 (Intermediate), C1 (Upper Intermediate), C2 (Advanced).

Apartado E. Los bucles.

Actividad 1.

Implementa una aplicación que muestre los números del 1 al 50 de la siguiente forma:

- a) Con un bucle `for` separando los números por una coma.
- b) Con un bucle `while` separando los números por un espacio.
- c) Con un bucle `do-while` empezando cada número en una nueva línea.

Apartado F. La depuración de código.

Actividad 1.

Para depurar programas podemos usar las herramientas de depuración que ofrecen los navegadores. La instrucción `debugger` es útil para crear *breakpoints* en los que el programa se detendrá.

Ejecuta paso a paso (F10) el siguiente ejemplo:

```
let nombre = "Ada";
let i;
for (i = 1; i < 10; i++) {
    document.write(nombre + ", ");
}
```

3. Ejercicios.

Actividad 1. NIF/NIE

Implementa una aplicación que, dado un número de NIF español o un NIE de extranjeros residentes en España, determine el dígito de control que le corresponde.

En esta [web del ministerio](#) puedes encontrar información sobre cómo se obtiene el dígito de control.

Actividad 2. Calculadora.

Implementa una aplicación que actúe como una calculadora para operaciones aritméticas básicas: suma, resta, multiplicación y división. El programa solicita al usuario dos números y la operación a realizar e informa del resultado obtenido.

Actividad 3. Vocales.

Implementa un juego en el que el usuario tenga que adivinar una letra vocal (a, e, i, o, u) en tres intentos. El programa tendrá tres palabras predefinidas para cada vocal, por ejemplo, para la vocal “a”, podría tener las palabras “informática”, “javaScript” y “programar”.

La aplicación elegirá automáticamente una vocal y mostrará la primera palabra asociada. Si el usuario acierta la vocal, gana el juego, en caso contrario le mostrará la segunda palabra. Si en tres intentos el usuario no adivina la vocal, pierde el juego. Se debe informar al usuario de los turnos empleados.

Actividad 4. Suma20.

Implementa un juego en el que el usuario, partiendo de 0, tenga que sumar un número entre 1 y 5 para llegar al número 20. Éste es el ejemplo de una partida:

¡Juguemos a Suma20!

Partimos del número 0.

Número elegido por el usuario: 3

Número elegido por la máquina: 7

Número elegido por el usuario: 12

Número elegido por la máquina: 13

Número elegido por el usuario: 18

Número elegido por la máquina: 20

¡Gana la máquina!

Ejercicio 5. Fibonacci.

Desarrolla un juego en el que el usuario tenga que introducir términos de la sucesión de Fibonacci sin equivocarse. Un ejemplo sería el siguiente:

Sucesión: 0 1 1 2 3 5 8 13 21 34 46
Error. Términos correctos: 4

Se puede jugar en dos modos, “fácil” partiendo del 0 y 1 o “difícil” partiendo del 987 y 1597. El usuario elegirá el modo de juego al inicio de la partida.

Ejercicio 6. Adivina 100

Implementa un juego que adivine un número elegido por el usuario de 1 a 100. En cada turno el usuario indicará si el número elegido por el juego es mayor, menor o igual al número que tiene que adivinar. El juego debe emplear el menor número de turnos posibles.