

## TEST

Se debe escoger siempre la respuesta más correcta y sólo disponemos de una

**Puntuación: (8 puntos)**

**Correcta: 0,2**

**Incorrecta: 4 restan una correcta (-0.05)**

**Sin contestar: Ni suma, ni resta**

### 1. En HTML el siguiente código:

```
<dl>
  <dt>Denim (semigloss finish)</dt>
  <dd>Ceiling</dd>

  <dt>Denim (eggshell finish)</dt>
  <dt>Evening Sky (eggshell finish)</dt>
  <dd>Layered on the walls</dd>
</dl>
```

- a. Es incorrecto.
- b. Es incorrecto, representa una lista de términos pero hay un término sin definición y esto hace que el navegador la interprete mal.
- c. Es correcto y se utiliza para almacenar y clasificar términos y definiciones expresadas en texto, pero podemos tener varios términos seguidos con una sola definición.
- d. Ninguna de las anteriores.

### 2. La etiqueta figure en HTML

- a. Permite un marcado enriquecido para dar más relevancia a las imágenes y da la posibilidad de añadir una descripción con figcaption.
- b. Puede sustituir a img.
- c. Está en desuso.
- d. Todas las anteriores.

### 3. El siguiente código CSS

```
p[class~="abajo"] { color: orange; }
```

- a. Coloca de naranja los p con class “abajo” exactamente.
- b. Coloca de naranja los p que no tienen la class “abajo”
- c. Coloca de naranja los p que contienen dentro de su atributo class al menos uno de los valores coincidente con “abajo”.
- d. Ninguna de las anteriores.

### 4. ¿Cuál de las fases de desarrollo de un software no debería tener que repetirse en un proyecto?

- a. Análisis del problema
- b. Codificación
- c. Pruebas
- d. Cualquiera de las anteriores.

## 5. `getElementsByClassName` y `querySelectorAll`, ¿qué diferencias tienen?

- a. Ambos son métodos de acceso al DOM.
- b. Ambos devuelven un array
- c. En ambos podemos usar el nombre de una clase para obtener los nodos del DOM, sólo que en `querySelectorAll` usamos selectores y en `getElementsByClassName` sólo el nombre de la clase.
- d. Todas las anteriores.

## 6. Los ABB son

- a. Estructuras de datos complejas con la finalidad de optimizar búsquedas de elementos que se pueden ordenar.
- b. Equivalentes a una lista o array.
- c. Estructuras de datos complejas con la finalidad de optimizar búsquedas de elementos que se pueden ordenar; está pensado de forma recursiva y esto hace que se incremente la velocidad de búsqueda.
- d. Ninguna de las anteriores.

## 7. El evento `onunload` en JS

- a. No existe.
- b. Detecta si el usuario ha abandonado la página.
- c. Permite detectar si una página no ha cargado.
- d. Ninguna de las anteriores.

## 8. `git checkout -b`

- a. Permite crear una rama y cambiarnos a ella.
- b. Permite cambiarnos a una rama.
- c. Permite borrar una rama
- d. Todas son correctas.

## 9. En POO en JS

- a. Se usa `new` para llamar a los constructores de las clases.
- b. Se detectan los atributos de los que consta la clase por medio de los parámetros pasados al constructor.
- c. Cuando queremos añadir métodos accesorios, se usan las palabras reservadas `get` y `set`.
- d. Todas son correctas.

## 10. El siguiente código CSS

```
#padre{  
  display: flex;  
  justify-content: space-evenly;  
}
```

- a. Da instrucciones acerca del espaciado en vertical.
- b. Da instrucciones acerca del espaciado en horizontal.
- c. Da instrucciones acerca del espaciado en horizontal, de manera que los items tienen el mismo espacio entre ellos, menos el espacio que hay en el inicio y el final.

d. Ninguna de las anteriores.

**11. JSON.stringify()**

- a. Convierte un JSON a String
- b. Convierte un String a JSON
- c. Es incorrecto, no existe este método y no pertenece a la clase JSON en JS
- d. Ninguna de las anteriores.

**12. console.log(Boolean(new String("Hola")));**

- a. Devuelve false
- b. Da una excepción
- c. Muestra false
- d. Ninguna de las anteriores.

**13. Si queremos almacenar información en el lado del cliente usando JS, podemos usar**

- a. El Local Storage
- b. El Session Storage
- c. Las Cookies
- d. Todas las anteriores.

**14 . El siguiente código:**

```
function f (x, y = 3, z = 22) {  
    return x + y + z  
}
```

```
console.log(f(3));  
console.log(f(3,0));  
console.log(f(3,0,0));
```

- a. Usa destructuring
- b. Usa una función con parámetros opcionales
- c. Es incorrecto en JS
- d. Usa sobrecarga.

**15 . El siguiente código:**

```
let minimo = Math.min(...miArray);
```

- a. Usa destructuring
- b. Usa spread syntax
- c. Es incorrecto en JS
- d. Usa sobrecarga.

**16 . En el manejo de Excepciones en JS, throw**

- a. Permite capturar una excepción
- b. Ninguna otra opción es correcta
- c. Permite lanzar una excepciones
- d. Se ejecuta siempre, es equivalente a finally

**17. En planificación de procesos, ¿cuál/cuáles es/son los recursos que comparten los procesos?**

- a. E/S
- b. E/S y CPU
- c. CPU
- d. Ninguna de las anteriores.

**18. En una expresión regular el metacaracter “\d” sirve para**

- a. Equivale al patrón [a-zA-Z0-9\_] (se pueden indicar rangos de caracteres), que indica que son válidos cualquier minúscula, mayúscula, cifra o \_
- b. Sirve para escapar el carácter d.
- c. Equivale al patrón [0-9]
- d. Ninguna de las anteriores.

**19. El siguiente pseudocódigo**

```
int sumaDoble ( int n ) {  
    int s;  
    if ( n == 0 )  
        s = 0;  
    else if ( n > 0 )  
        s = sumaDoble( n-1 ) + 2*n;  
    return s;  
}
```

Representa

- a. A una función recursiva
- b. Al sumatorio doble de un número
- c. Equivaldría a devolver directamente 0 o sumaDoble( n-1 ) + 2\*n; sin tener que guardar el valor en s
- d. Todas las anteriores.

**20. El tipo de media speech en Media Queries se usa para referenciar a**

- a. Lectores de pantalla
- b. Es erróneo, no existe
- c. Está en desuso, las aplicaciones actuales usan los lectores de pantalla del SO anfitrión.
- d. Ninguna de las anteriores.

**21. ON UPDATE SET NULL en MySQL indica**

- a. Lo que se hace con el valor de la PK en caso de que se actualice la FK.
- b. Lo que se hace con el valor de la FK en caso de que se actualice la PK a la que apunta.

- c. Lo que se hace con el valor de la FK en caso de que se actualice cualquier campo de la fila a la que apunta.
- d. Ninguna de las anteriores.

## **22. Los parámetros en una función en PLSQL MySQL**

- a. Pueden ser de salida o de entrada/salida
- b. Sólo pueden ser de entrada
- c. No se les pueden pasar parámetros.
- d. Ninguna de las anteriores.

## **23. ¿Bootstrap sirve para aplicaciones mobile- first?**

- a. Sí.
- b. No por defecto.
- c. Sí, pero depende de los media
- d. Ninguna de las anteriores.

## **24. El método race de la clase Promise**

- a. Permite recoger todos los resultados de las operaciones asíncronas al terminar y ejecutar código cuando eso pase.
- b. Permite recoger el resultado de la primera operación asíncrona que termine y ejecutar código cuando eso pase.
- c. Permite recoger el resultado de la última operación asíncrona que termine y ejecutar código cuando eso pase.
- d. Ninguna de las anteriores.

## **25. Firebase incluye:**

- a. BD NoSQL basada en colecciones como hemos visto con MongoDB
- b. Autenticación segura con diferentes plataformas (Google, Facebook, GitHub...)
- c. Almacenamiento en servidores para la información de las aplicaciones que desarrollemos
- d. Todas las anteriores.

## **26. En MongoDB**

- a. La interfaz gráfica oficial es Compass
- b. El servicio en la nube oficial es Atlas
- c. Todas son correctas.
- d. No se puede crear una Base de Datos en Compass sin al menos una colección y se pueden insertar los documentos en las colecciones de uno en uno y por medio de un fichero de texto.

**27. ¿Qué podemos decir del siguiente código?**

```
    async function getUserAsync2(name) {  
    try {  
        let response = await fetch(`https://api.github.com/users/${name}`);  
        let data = await response.json()  
        return data;  
    }  
    catch (error) {  
        console.log(`ERROR: ${error.stack}`);  
    }  
}  
getUserAsync2('daviniathebridge')  
.then(data=>console.log(data))
```

- a. Las tareas asíncronas las marca el async
- b. Las tareas asíncronas las marca el await
- c. error puede ser tanto de la llamada a .json() como del fetch.
- d. b y c son correctas

**28. La palabra reservada import en JS...**

- a. Permite usar recursos de un js dentro de otro
- b. Permite usar recursos de un js dentro de otro solamente en Node
- c. Permite usar recursos de un js dentro de otro solamente en el Front
- d. Ninguna de las anteriores

**29 En versionado semántico**

- a. La versión MAYOR cuando realizas un cambio incompatible con lo anterior (un cambio mayor. Ej: Añadir una funcionalidad nueva en Whatsapp que no existía: Tratamiento de Audios)
- b. La versión MENOR cuando añades funcionalidad que compatible con versiones anteriores (un cambio menor. Ej: Que en Whatsapp los audios ahora se puedan borrar)
- c. La versión PARCHE cuando reparas errores compatibles con versiones anteriores (un correctivo. Ej: Los usuarios de Whatsapp de iPhone tienen un problema con los audios mayores de 5 min)
- d. Todas las anteriores.

**30. app.listen(3000); indica**

- a. Que app ha sido construido como un servidor de Express
- b. Que el servidor se encuentra esperando peticiones en el puerto 3000
- c. Todas son correctas
- d. a y b son correctas

**31. ¿De la librería de Node mongodb qué método permite hacer búsquedas?**

- a. findOne
- b. find
- c. a y b son correctas
- d. Ninguna de las anteriores

**32. En Moongose Schema.Types.Mixed**

- a. Indica que el valor de la clave puede ser cualquier tipo de datos (flexible). (Ojo: Con este tipo de cambios no se confirman los cambios sin llamar a una función propia de Mongoose.)
- b. No existe este tipo de dato
- c. Indica que el valor de la clave es de tipo binario como imágenes, pdfs, etc.
- d. Todas las anteriores.

**33. Sequelize en Node**

- a. Sirve para BD MySQL
- b. Sirve para BD Relacionales
- c. Se puede usar en MongoDB
- d. Todas las anteriores

**34. El siguiente código en PUG**

**a#boton.btn(href="http://www.google.com" target="\_blank") Google**

**¿Qué salida HTML produce?**

- a. <a class="boton" id="btn" href="http://www.google.com" target="\_blank">Google</a>
- b. <a class="btn" id="boton" href="http://www.google.com" target="\_blank">Google</a>

c. `<a href="http://www.google.com" target="_blank">#boton.btn</a>`

d. Todas las anteriores.

**35. La palabra reservada en EJS para incluir archivos parciales dentro de las vistas es...**

a. import

b. require

c. include

d. Ninguna de las anteriores

**36. En un módulo Node module.exports**

a. Permite exportar un valor final, función o JSON

b. Sólo puede estar una vez

c. Conecta con los require que hagamos en otros ficheros y dependiendo del formato de lo devuelto, tendremos que usarlo correctamente (Las claves con su nombre en el JSON, por ejemplo)

d. Todas las anteriores.

**37. ¿Cuál de los siguientes puntos no forma parte del manifiesto ágil?**

a. Las personas y las interacciones son secundarias, los procesos y herramientas van antes.

b. Que el producto funcione de forma perfecta es más importante que contar con una documentación extensa.

c. La colaboración con el cliente es más beneficiosa que establecer una relación contractual.

d. El proceso de trabajo debe tener capacidad de responder ante el cambio, y no obcecarse en mantener el statu quo anterior.

**38. El diagrama de secuencia en diseño de proyectos software sirve para**

a. Mostrar las opciones de los diferentes roles dentro de la aplicaciones.

b. Describir el flujo de una tarea concreta.

c. Indicar las clases y cómo se relacionan los objetos de la aplicación.

d. Ninguna de las anteriores.

**39. Un middleware se usa principalmente para**

a. Validar datos

b. Validar los endpoints

c. Validar permisos

d. Todas las anteriores



#### **40.HTTP PUT**

- a. Equivale a POST
- b. No existe
- c. Crea un nuevo elemento o reemplaza una representación del elemento de destino con los datos de la petición.
- d. Ninguna de las anteriores

#### **DESARROLLO**

##### **1. Describe línea a línea el siguiente código (1.5 puntos):**

```
const form = document.getElementById("task-form");
form.addEventListener("submit", e => {
    e.preventDefault();
    const task = {
        title: form["task-title"].value,
        description: form["task-description"].value
    }

    insertTask(task);
})
```

##### **2. Indica la salida por consola del siguiente programa (0.5 puntos)**

```
function sum(a, b) { return a + b; }
console.log(sum(1, "2"));
```