

Blue Tech Test - Examen de conocimientos para desarrollar en Cells

- **Número de preguntas:** 25
- **Tiempo:** 2 horas 30 minutos
- **Número de secciones:** 3
 - Programación Orientada a Objetos
 - 5 Preguntas - Fallos permitidos: 1
 - JavaScript
 - 10 Preguntas - Fallos permitidos: 2
 - Polymer 2.
 - 10 Preguntas - Fallos permitidos: 3
- **Tipo de examen:** Opción multiple
- *Para obtener una nota aprobatoria de todo el examen debes aprobar cada una de las secciones, lo que implica que si una de las secciones rebasa su número máximo permitido de fallos automáticamente habrás reprobado el examen independientemente del hecho de que las secciones restantes no tengan error alguno.*
- **El examen se ejecuta en pantalla completa de la cual no debes salir hasta terminar el examen, los cambios de pestaña o ventana son registrados en la plataforma así como la apertura de la consola de desarrollador, si dichos eventos son registrados el examen será anulado y por ende**

no acreditado.

Sección I - POO

Buenas prácticas

1. La Alta Cohesión debería ser acompañada de...

- ☐ Polimorfismo
- ☐ Instancias
- ☒ Bajo acoplamiento
- ☐ Patrón de diseño Factory method

Herencia

2. ¿Cuál de las siguientes afirmaciones se corresponde con el concepto de Herencia en POO?

- ☐
Evita que otros objetos modifiquen o tengan acceso a las propiedades encapsuladas de un objeto.
- ☒
La clase Empleado es la super clase de las clases, Empleado Sindicalizado, Empleado de Confianza, Empleado Externo
- ☐

Las funciones se resuelven en tiempo de ejecución en lugar de tiempo de compilación

- ☐

Al acoplamiento y baja cohesión son el resultado de una jeraquia de clases bien definidas.

Padres e hijos

3. Cuando dos o mas clases sirven como clase base para una clase derivada, la situación se conoce como.

- ☐ **Polimorfismo**
- ☒ **Herencia múltiple**
- ☐ **Encapsulamiento**
- ☐ **Herencia jerárquica**
- ☐ **Ninguna de las anteriores**

Responsabilidades

4. La información que almacena una clase debe ser coherente y estar relacionada lo mas posible con la clase misma. ¿Estamos hablando de?

- ☒ **Alta cohesion**
- ☐ **Especialización**
- ☐ **Monomorfismo**

- ☐ Bajo acoplamiento

Polimorfismo

5. ¿Cuál de las siguientes afirmaciones es acerca del polimorfismo en POO?

- ☐ Es el ocultamiento del estado de un objeto
- ☐ Es la capacidad de suministrar datos al código
- ☒

Es la capacidad que tienen los objetos de una clase de responder a un evento en función de los parámetros utilizados en su invocación

- ☐
Es cuando un objeto adquiere todas las propiedades de otro objeto

Sección II - JavaScript

JSON

1. De esta manera podemos convertir un texto JSON en un objeto de JavaScript

- ☐ `var obj = Object.parse(json);`
- ☒ `var obj = JSON.parse(json);`
- ☐ `var obj = Object.Map(json);`

- ☐ `var obj = JSON.stringify(json);`

Funciones

2. ¿Qué obtenemos al ejecutar el siguiente código?

```
(function f(f) {  
    return typeof f();  
})(function(){ return 1; })
```

- ☐ “function”
- ☐ “undefined”
- ☒ “number”
- ☐ Error

¿Qué es lo que hace?

3. En JavaScript ¿qué hace la siguiente linea de código?

```
var square = number => number * number;
```

- ☐ Nada, hay error de sintaxis
- ☐
Define una función sin parámetros que retorna el cuadrado de un número
- ☐

Define una función con dos parámetros que retorna el cuadrado de un número

- ☒

Define una función que retorna el cuadrado de un número dado

Verdades y mentiras

4. ¿Cuál de los siguientes enunciados **no es cierto** acerca de Javascript?

- ☐ JavaScript es sensible a mayúsculas y minúsculas
- ☐ JavaScript es un lenguaje debilmente tipado
- ☐ “do” es una palabra reservada del lenguaje
- ☒

Al final de una sentencia en JavaScript es obligatorio usar “;”

¿Qué imprime?

5. ¿Qué imprime en consola la siguiente porción de código?

```
function f(x, y=2, z=7) {  
    return x + y + z;  
};  
console.log(f(1) === 10);
```

- ☐ **undefined**
- ☐ **NaN**
- ☒ **true**
- ☐ **10**

6. De esta manera podemos convertir un objeto de JavaScript en un texto JSON

- ☐ `var json = Object.parse(obj);`
- ☐ `var json = JSON.parse(obj);`
- ☐ `var json = Object.Map(obj);`
- ☒ `var json = JSON.stringify(obj);`

Ordenando

7. Al ejecutar las siguientes lineas de código ¿Qué resultado obtendremos?

```
let letras = ['Lambda', 'Alfa', 'Gamma', 'Beta'];  
letras.sort();
```

- ☒ **El arreglo letras se ordena de forma ascendente**
- ☐ **El arreglo letras se ordena de forma descendente**
- ☐ **Error, sort no es un método para los arreglos de JavaScript**
- ☐

Error, el método sort requiere obligatoriamente una función que defina la manera

Nombres

8. ¿Cuál es el resultado del siguiente código?

```
function getName() {  
    const name = 'Juan';  
    name = 'Jose';  
    return name;  
};  
getName();
```

- ☐ Juan
- ☐ Jose
- ☒ error
- ☐ undefined

9. Considera el siguiente código en JavaScript

```
function foo() {  
    return 5  
}
```

¿Qué es lo que haria el codigo **let myVar = foo** ?

- ☐ Asigna el valor entero 5 a la variable myVar.
- ☐ Lanza una excepción
- ☒
Asigna una referencia hacia función foo en la variable myVar
- ☐ Nada

Null

10. En JavaScript ¿cuál es el tipo de dato de **null**?

- ☐ undefined
- ☐ integer
- ☐ nill
- ☒ object

11. ¿Cuál es el modelo que se usa para crear un objeto?

- ☐ inicializacion
- ☐ array
- ☒ clase

12. Resultado que imprime en consola

```
let mascotas = ["perro" , "gato", "loro", "canario"];  
mascotas.filter(mascota => mascota.length >= 5);
```

- ☒ ["perro", "canario"]
- ☐ ["perro" , "gato", "loro", "canario"]

13. ¿Cuándo se ejecuta el siguiente código?

```
if(x === y) {  
  console.log('Hola mundo');  
}
```

- ☐
Cuando las dos variables son iguales se imprime 'Hola mundo'
- ☒
Cuando las variables y tipos son iguales se imprime 'Hola mundo'

14. ¿Cómo hacemos para agregar un **value** al final de un arreglo?

- ☐ **arr[arr.length + 1] = value;**
- ☒ **arr[arr.length] = value;**
- ☐ **arr[arr.length - 1] = value;**
- ☐ **arr = arr + value;**

15. ¿Cuál es un estado de promesa?

- ☐ **pendiente (pending)**
- ☐ **cumplida (fullfilled)**
- ☐ **rechazada (rejected)**

- ☒ **todas las anteriores**

16. A la posición 5 del arreglo de objetos llamado **persons** cambiar el valor de la propiedad lastName.

- ☒ **this.set('persons.4.lastName', "juan");**
- ☐ **this.set('persons.4.lastname', "juan");**
- ☐ **this.set('persons.4.last-name', "juan");**
- ☐ **this.set('persons[4]lastame', "juan");**
- ☐ **this.persons[4].lastname = "juan";**

17. ¿Qué evento se usa para detectar cuando se pierde el focus en un input?

- ☐ **on-focus**
- ☐ **on-leave**
- ☐ **lost-focus**
- ☐ **on-blur**

18. ¿Qué seleccionas con esta linea de código?

```
document.querySelector("p.five")
```

- ☐ **El priimer elemento **p** con la clase **five****
- ☐ **Todos los elementos de la clase **p****
- ☒ **Todos los elementos de **p** con la clase **five****

19. ¿Qué método borra el elemento avion?

```
let transportes = ['avion', 'camion', 'barco', 'tren'];
```

- ☒ `transportes.shift();`

El método `shift()` elimina el primer elemento del array y devuelve dicho elemento. Este método modifica la longitud del array.

- ☐ `transportes.push('avion');`
- ☐ `transportes.pop();`
- ☐ `transportes.unshift();`

El método `unshift()` agrega uno o más elementos al inicio del array, y devuelve la nueva longitud del array.

Otros

2. Cuando 2 o mas clases sirven como clase base para una clase derivada la situacion se conoce como:

- ☐ **Polimorfismo**
- ☒ **Herencia multiple**
- ☐ **Encapsulamiento**
- ☐ **Herencia jerarquica**
- ☐ **Ninguna de las anteriores**

3. Cuando se asigna un observer a 2 propiedades ¿Qué tipo de observer es?

- ☐ Simple
- ☐ Doble
- ☒ Complejo
- ☐ Binario

4. ¿Cuál de las siguientes afirmaciones se refiere al polimorfismo?

- ☐ Ocultamiento del estado de un objeto
- ☐ Capacidad de suministrar datos al código
- ☒
Capacidad que tienen los objetos al responder a un evento en función de los parámetros utilizados en su invocación
- ☐ Cuando un objeto adquiere todas las propiedades de otro

5. ¿A que se refiere cuando la información de una clase debe ser coherente y relacionada con la misma?

- ☒ Alta cohesión
- ☐ Especialización
- ☐ Bajo acoplamiento

7. ¿Cómo se declaran variables de bloque?

- ☒ let
- ☐ var
- ☐ const

8. ¿Cuál es el valor de **y** en el siguiente bloque de código?

```
var x;  
var y = x === null;
```

- ☒ **false**

9. ¿Qué imprime en consola?

```
let x = "4" + 4 + 5;  
let y = 4 + 4 + "5";  
console.log(x + " " + y);
```

- ☒ **445 85**

10. ¿Cuál es la diferencia entre un objeto y una clase?

- ☒ **Un objeto es una instancia de una clase.**
- ☐

Un objeto genera funciones, valores para ser usados por una clase.

11. A que se refiere cuando una clase es derivada de otra?

- ☐ **Herencia multiple**
- ☒ **Herencia jerarquica**

12. Metodo especial que permite inicial valores de un objeto

- ☒ **Constructor**

13. ¿A que se refiere cuando los detalles internos de una clase pueden ser ocultados del exterior?

- ☒ **Encapsulamiento**