

# Examen de conocimientos para desarrollar en Cells

**Número de preguntas:** 25

**Tiempo:** 2 horas 30 minutos

**Número de secciones:** 4

Programación Orientada a Objetos 5 Preguntas - Fallos permitidos: 1

JavaScript 10 Preguntas - Fallos permitidos: 2

Polymer 2 10 Preguntas - Fallos permitidos: 3

Sección practica – 1 ejercicio

**Tipo de examen:** Opción múltiple

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

1. La Alta Cohesión debería se acompañada de...
  - A. Polimorfismo
  - B. Instancias
  - C. Bajo acoplamiento
  - D. Patrón de diseño Factory method
  
2. ¿Cuál de las siguientes afirmaciones se corresponde con el concepto de Herencia en POO?
  - A. Evita que otros objetos modifiquen o tengan acceso a las propiedades encapsuladas de un objeto.
  - B. La clase Empleado es la super clase de las clases, Empleado Sindicalizado, Empleado de Confianza, Empleado Externo
  - C. Las funciones se resuelven en tiempo de ejecución en lugar de tiempo de compilación
  - D. Al acoplamiento y baja cohesión son el resultado de una jerarquía de clases bien definidas.
  
3. Cuando dos o más clases sirven como clase base para una clase derivada, la situación se conoce como:
  - A. Polimorfismo
  - B. Herencia múltiple
  - C. Encapsulamiento
  - D. Herencia jerárquica
  - E. Ninguna de las anteriores
  
4. La información que almacena una clase debe ser coherente y estar relacionada lo más posible con la clase misma. ¿Estamos hablando de?
  - A. Alta cohesión
  - B. Especialización
  - C. Monomorfismo
  - D. Bajo acoplamiento
  - E. Polimorfismo
  
5. ¿Cuál de las siguientes afirmaciones es acerca del polimorfismo en POO?
  - A. Es el ocultamiento del estado de un objeto
  - B. Es la capacidad de suministrar datos al código
  - C. Es la capacidad que tienen los objetos de una clase de responder a un evento en función de los parámetros utilizados en su invocación
  - D. Es cuando un objeto adquiere todas las propiedades de otro objeto

6. ¿Cuál es el modelo ó template para creación de objetos?
- A. Lista
  - B. Instancia
  - C. Array
  - D. Clase
7. ¿Cuál de las siguientes afirmaciones se refiere al polimorfismo?
- A. Ocultamiento del estado de un objeto
  - B. Capacidad de suministrar datos al código
  - C. Capacidad que tienen los objetos al responder a un evento en función de los parámetros utilizados en su invocación
  - D. Cuando un objeto adquiere todas las propiedades de otro
8. Es el mecanismo que permite crear nuevas clases a partir de clases existentes extendiendo y refinando sus capacidades
- A. Polimorfismo
  - B. Encapsulamiento
  - C. Diseño de Patrones
  - D. Herencia
9. ¿A Qué se refiere cuando la información de una clase debe ser coherente y relacionada con la misma?
- A. Alta cohesión
  - B. Especialización
  - C. Bajo acoplamiento
10. ¿Cuál es la diferencia entre un objeto y una clase?
- A. Un objeto es una instancia de una clase.
  - B. Un objeto genera funciones, valores para ser usados por una clase.
11. A que se refiere cuando una clase es derivada de otra?
- A. Herencia múltiple
  - B. Herencia jerárquica
12. Método especial de la clase que permite inicializar los valores de un objeto
- A. Objetos
  - B. Constructor

- C. Herencia
- D. Virtual

13. ¿A qué se refiere cuando los detalles internos de una clase pueden ser ocultados del exterior?
- A. Abstracción
  - B. Principio Open Closed
  - C. Encapsulamiento
  - D. Ocultamiento

## Sección II - JavaScript

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

- A. `var obj = Object.parse(json);`
- B. `var obj = JSON.parse(json);`
- C. `var obj = Object.Map(json);`
- D. `var obj = JSON.stringify(json);`

2. ¿Qué obtenemos al ejecutar el siguiente código?

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

- A. "function"
- B. "undefined"
- C. "number"
- D. Error

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

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

- A. Nada, hay error de sintaxis
- B. Define una función sin parámetros que retorna el cuadrado de un número
- C. Define una función con dos parámetros que retorna el cuadrado de un número
- D. Define una función que retorna el cuadrado de un número

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

- A. JavaScript es sensible a mayúsculas y minúsculas
- B. JavaScript es un lenguaje débilmente tipado
- C. "do" es una palabra reservada del lenguaje
- D. Al final de una sentencia en JavaScript es obligatorio usar ";;"

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);
```

- A. undefined
- B. NaN
- C. true
- D. 10

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

- A. var json = Object.parse(obj);
- B. var json = JSON.parse(obj);
- C. var json = Object.Map(obj);
- D. var json = JSON.stringify(obj);

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

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

- A. El arreglo letras se ordena de forma ascendente
- B. El arreglo letras se ordena de forma descendente
- C. Error, sort no es un método para los arreglos de JavaScript
- D. Error, el método sort requiere obligatoriamente una función que defina la manera

8. ¿Cuál es el resultado del siguiente código?

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

- A. Juan
- B. Jose
- C. error
- D. undefined

9. Considera el siguiente código en JavaScript

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

¿Qué es lo que haría el código **let myVar = foo** ?

- A. Asigna el valor entero 5 a la variable myVar.
- B. Lanza una excepción
- C. Asigna una referencia hacia función foo en la variable myVar
- D. Nada
- E. Null

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

- A. undefined
- B. integer
- C. nill
- D. object

12. Resultado que imprime en consola

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

- A. ["perro", "canario"]
- B. ["perro" , "gato", "loro", "canario"]

13. ¿Cuándo se ejecuta el siguiente código?

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

- A. Cuando las dos variables son iguales se imprime 'Hola mundo'
- B. Cuando las variables y tipos son iguales se imprime 'Hola mundo'

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

- A. `arr[arr.length + 1] = value;`
- B. `arr[arr.length] = value;`
- C. `arr[arr.length - 1] = value;`
- D. `arr = arr + value;`

15. ¿Cuál es un estado de promesa?

- A. pendiente (pending)
- B. cumplida (fulfilled)
- C. rechazada (rejected)
- D. todas las anteriores

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

- A. `this.set('persons.4.lastName', "juan");`
- B. `this.set('persons.4.lastname', "juan");`
- C. `this.set('persons.4.last-name', "juan");`
- D. `this.set('persons[4]lastame', "juan");`
- E. `this.persons[4].lastname = "juan";`

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

- A. `on-focus`
- B. `on-leave`
- C. `lost-focus`
- D. `on-blur`

18. ¿Qué seleccionas con esta linea de código?

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

- A. El primer elemento p con la clase five
- B. Todos los elementos de la clase p
- C. Todos los elementos de p con la clase five
- D. Ninguno, este no es un selector valido

19. ¿Qué método borra el elemento avion?

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

- A. `transportes.shift();`
- B. `transportes.push('avion');`
- C. `transportes.pop();`
- D. `transportes.unshift();`

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

- A. Simple
- B. Doble
- C. Complejo
- D. Binario

21. ¿Cómo se declaran variables de bloque?

- A. `let`
- B. `var`
- C. `const`
- D. `set`

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

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

- A. false
- B. undefined
- C. null
- D. NaN

23. ¿Qué imprime en consola?

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

- A. 49 445
- B. 49 13
- C. 445 85
- D. 445 13

### Sección III- Polymer

1. Para notificar a Polymer del cambio hecho solamente en una propiedad de un objeto debes invocar esta función

- A. this.push
- B. this.notify
- C. this.notifyPath
- D. this.set
- E. Ninguno de los anteriores

2. Si en un componente tienes la siguiente propiedad

```
name: {  
  type: 'String',  
  value: '';  
  notify: true  
}
```

¿Qué evento es lanzado al ejecutar la siguiente línea de código?

```
this.set('name', 'Juan Pérez');
```

- A. name-modified
- B. name-notified
- C. name-changed
- D. Ningún evento es lanzado



3. Selecciona la opción que contenga sólo tipos de datos validos en Polymer

- A. Boolean, Number, String, Array, Object
- B. Boolean, Number, String, Enum, Array
- C. Boolean, Computed, Number, String, Class
- D. Observer, Number, String, Array, Object

4. ¿Qué tipo de “binding” es el siguiente?

```
<my-component attribute="[ [property]]"></my-component>
```

- A. Upward
- B. Two-way
- C. One-way
- D. Lazy

5. Si queremos que nuestro componente exponga la posibilidad de modificar su color de texto teniendo como valor por default el azul ¿cuál opción cubriría el requerimiento?

```
:host { color: var(--my-text-color, blue) }  
:body { color: var(blue, --my-text-color) }  
:html { color: var(blue, --my-text-color) }  
:p, :label, :span { color: var(--my-text-color,  
blue) }
```

6. Son propiedades virtuales cuyo valor se determina basado en otras propiedades

- A. Observers
- B. Computadas
- C. Objects
- D. Behaviors

7. Supongamos la siguiente condición para un requerimiento:

Si la edad del usuario es mayor o igual a dieciocho el color de texto de su nombre debe ser azul, si no debe ser rojo. ¿Cómo se podría implementar dicho requerimiento?

- A. Usando un dom-if para cada una de las condiciones
- B. Usando un dom-if-else
- C. Usando un dom-if para una condición y un dom-else para la otra
- D. Todas las anteriores son válidas

8. Supongamos que tienes un arreglo llamado persons: el cual tiene objetos como

```
{name: '', lastName: '' }
```

La longitud del arreglo es 10. Tú necesitas remover el tercer elemento del arreglo

- A. `this.splice(persons, 2, 1);`
- B. `this.splice('persons', 2, 1);`

- C. `this.[2] = null;`
- D. Ninguna de las anteriores

9. Permite el uso de una función **filter** como una de sus propiedades

- A. dom-if
- B. dom-repeat
- C. dom-module
- D. dom-filter

10. ¿Cuál es el nombre correcto del método para lanzar eventos en Polymer 2?

- A. `launchEvent`
- B. `fireEvent`
- C. `fire`
- D. `dispatchEvent`

11. De forma declarativa ¿cómo haces el binding de la propiedad `firstName`?

- A. `<my-component firstName="Joe"></my-component>`
- B. `<my-component first.Name="Joe"></my-component>`
- C. `<my-component first-name="Joe"></my-component>`
- D. `<my-component first-Name="Joe"></my-component>`

12. Los observers se ejecutan de forma:

- A. Asíncrona
- B. Síncrona
- C. Síncrona y Asíncrona
- D. Los observers son valores de propiedades, por lo tanto no se ejecutan

13. ¿Con que medios Polymer monitorea cambios en las propiedades...?

- A. Return, New Properties, Data Bindings
- B. Observers, Molecule Elements, Template Repeater
- C. Observers, Computed Properties, Data Bindings
- D. Interfaces, Clases, Prototypes

14. En polymer ¿cómo puedes añadir un listener a un campo de texto para detectar cuando se pierde el focus en un input?

- A. `<input type="text" on-blur="myListener">`
- B. `<input type="text" on-focus="myListener">`
- C. `<input type="text" on-leave="myListener">`
- D. `<input type="text" lost-focus="myListener">`

15. A la posición 4 de un arreglo de objetos cambiar el valor de la propiedad `lastname`.

- A. `this.set('persons.4.lastname', "juan");`
- B. `this.set('persons.[4].lastname', "juan");`
- C. `this.set('persons[4]lastname', "juan");`
- D. `this.persons[4].lastname = "juan";`

16.Cuál de los siguientes métodos no forma parte del ciclo de vida de polymer?

- A. `constructor()`
- B. `connectedCallback()`
- C. `disconnectedCallback()`
- D. `connectedEvent()`
- E. `attributeChangedCallback()`