

Preguntas de examen para la certificación de Cells V3

Programación Orientado a Objetos

La Alta Cohesión debería se acompañada de...

☐ Polimorfismo

☐ Instancias



☒ Bajo acoplamiento

☐ Patrón de diseño Factory method

Programación Orientada a Objetos - Fundamentos
iii. Alta cohesión y bajo acoplamiento [\[Link\]](#)

¿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 superclase 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 jerarquía de clases bien definidas.

Programación Orientada a Objetos - Programación basada en prototipos
iv. Conceptos (herencia) [\[Link\]](#)

JavaScript - POO (ES6)
xi. Herencia (Extend y super, mixins) [\[Link\]](#)

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

☐ Polimorfismo



☒ Herencia múltiple

☐ Encapsulamiento

☐ Herencia jerárquica

☐ Ninguna de las anteriores

Programación Orientada a Objetos - Programación basada en prototipos
iv. Conceptos (herencia) [\[Link\]](#)

JavaScript - POO (ES6)
xi. Herencia (Extend y super, mixins) [\[Link\]](#)

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

- ☒ Alta cohesión
- ☐ Especialización
- ☐ Monomorfismo
- ☐ Bajo acoplamiento

Programación Orientada a Objetos - Fundamentos
iii. Alta cohesión y bajo acoplamiento [\[Link\]](#)

¿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

Programación Orientada a Objetos - Programación basada en prototipos
iv. Conceptos (polimorfismo) [\[Link\]](#)

JavaScript - POO (ES6)
xi. Herencia (Extend y super, mixins) [\[Link\]](#)

Cuando 2 o más clases sirven como clase base para una clase derivada la situación se conoce como:

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

Programación Orientada a Objetos - Programación basada en prototipos
iv. Conceptos (herencia) [\[Link\]](#)

JavaScript - POO (ES6)
xi. Herencia (Extend y super, mixins) [\[Link\]](#)

¿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.
- ☐ Un objeto es aquel que tiene muchos métodos

Programación Orientada a Objetos - Fundamentos
i. Fundamentos de POO [\[Link\]](#)

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

- ☐ Herencia múltiple



☒ Herencia jerárquica

☐ Polimorfismo

☐ POO

Programación Orientada a Objetos - Programación basada en prototipos
iv. Conceptos (herencia) [\[Link\]](#)

JavaScript - POO (ES6)
xi. Herencia (Extend y super, mixins) [\[Link\]](#)

Método especial que permite iniciar valores de un objeto



☒ constructor

☐ instancia

☐ get/set

Programación Orientada a Objetos - Fundamentos
ii. Mi primera clase [\[Link\]](#)

JavaScript - POO (ES6)
iii. Constructor [\[Link\]](#)

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

☐ Protected



☒ Encapsulamiento

☐ Polimorfismo

☐ Herencia múltiple

Programación Orientada a Objetos - Programación basada en prototipos
iv. Conceptos (encapsulamiento) [\[Link\]](#)

JavaScript - POO (ES6)
vii Métodos privados y x. Propiedades privadas [\[Link\]](#)

En una clase para definir un constructor que inicialice su valores se hace como?

☐ Poniendo el nombre de la clase precedido por el la palabra constructor



☒ Se crea un método llamado constructor

☐ Se crea un método con el nombre de la clase

☐ Se usa el método ready

Programación Orientada a Objetos - Fundamentos
ii. Mi primera clase [\[Link\]](#)

JavaScript - POO (ES6)
iii. Constructor [\[Link\]](#)

¿Cuál es el modelo para crear un objeto?

☐ inicialización

☐ array



☐ clase

Programación Orientada a Objetos - Fundamentos

ii. Mi primera clase [\[Link\]](#)

Programación Orientada a Objetos - Programación basada en prototipos

iv. Conceptos (clase) [\[Link\]](#)

JavaScript - POO (ES6)

ii. Clases (clase, instanciar clase) [\[Link\]](#)

JavaScript

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);`

JavaScript - Shallow copy and Deep copy

iii. Deep copy (JSON) [\[Link\]](#)

¿Qué obtenemos al ejecutar el siguiente código?

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

☐ "function"

☐ "undefined"



☒ "number"

☐ Error

JavaScript - Funciones JavaScript

iv. Funciones anónimas [\[Link\]](#)

JavaScript - Tipo de Valores

ii. Valores primitivos (number) [\[Link\]](#)

En JavaScript ¿que hace la siguiente línea 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

JavaScript - Funciones JavaScript

vi. Funciones flecha (Sintaxis básica, primeros pasos, el this, casos especiales) [\[Link\]](#)

En JavaScript ¿que hace la siguiente línea de código?

```
[1,2,3,4].map(number => number * number);
```

☐ Nada, hay error de sintaxis



☒ Itera un arreglo que retorna el cuadrado de cada número en un arreglo nuevo

☐ Itera el arreglo y retorna la multiplicación de 1*2*3*4 en un arreglo nuevo

☐ Itera el arreglo que retorna el cuadrado de cada número en el mismo arreglo

JavaScript - Funciones JavaScript

vi. Funciones flecha (Sintaxis básica, primeros pasos, el this, casos especiales) [\[Link\]](#)

JavaScript - Loops

xii. array.map [\[Link\]](#)

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

☐ JavaScript es sensible a mayúsculas y minúsculas

☐ JavaScript es un lenguaje debidamente tipado

☐ "do" es una palabra reservada del lenguaje



☒ Al final de una sentencia en JavaScript es obligatorio usar ";"

JavaScript - Tipo de Valores

i. Types (valores primitivos y complejos) [\[Link\]](#)

¿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

JavaScript - Extended Parameter Handling

i. Parámetros por default [\[Link\]](#)

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);

JavaScript - Shallow copy and Deep copy

iii. Deep copy (JSON) [\[Link\]](#)

Al ejecutar las siguientes líneas 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

JavaScript - Array

i. Métodos transformadores (push, pop, shift, unshift, splice, reverse, sort) [\[Link\]](#)

¿Cuál es el resultado del siguiente código?

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

☐ Juan

☐ Jose

☐ error

☐ undefined

JavaScript - Scoping

iii. const [\[Link\]](#)

Considera el siguiente código en JavaScript: `function foo() { return 5; }`

¿Qué es lo que haría el código `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

JavaScript - Funciones JavaScript

i. Introducción (función global, de objeto y constructor) [\[Link\]](#)

En JavaScript ¿cuál es el tipo de dato de null?

☐ undefined

☐ integer

☐ null

☐ object

JavaScript - Tipo de Valores
ii. Valores primitivos (null) [\[Link\]](#)

JavaScript - Tipo de Valores
iv. Casos especiales (null) [\[Link\]](#)

Resultado que imprime en consola:

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

☐ ["perro", "canario"]

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

☐ error

☐ undefined

JavaScript - Funciones JavaScript
vi. Funciones flecha (Sintaxis básica, primeros pasos, el this, casos especiales) [\[Link\]](#)

JavaScript - Loops
xii. array.map [\[Link\]](#)

¿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'

☐ Nunca entra

JavaScript - Ecuaciones Complejas
ii. Coacción explícita e implícita (string, boolean, number) [\[Link\]](#)

¿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;

JavaScript - Array [\[Link\]](#)

¿Cuál es un estado de promesa?

☐ pendiente (pending)

☐ cumplida (fulfilled)

☐ rechazada (rejected)

☒ todas las anteriores

Javascript - Callbacks y Promises (Básico)
iii. Promesas [Link](#)

¿Cual de las siguientes líneas nos da el resultado `['H', 'e', 'l', 'l', 'o']`?

☐ `['H', 'e'].merge(['l', 'l', 'o'])`

☐ `['H', 'e'].splice(1, ['l', 'l', 'o'])`



☒ `['H', 'e'].concat(['l', 'l', 'o'])`

☐ `['H', 'e'].join(['l', 'l', 'o'])`

JavaScript - Array
ii. Métodos accesorios (concat) [Link](#)

¿Cual es la propiedad que regresa el tipo de dato?

☐ `typeof`

☐ `typeOf`

☐ `type`

☐ `instanceof`

JavaScript - Tipo de Valores
ii. Valores primitivos [Link](#)

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

☐ `on-focus`

☐ `on-leave`

☐ `lost-focus`

☐ `on-blur`

JavaScript - Document (DOM)
v. Tipos de evento (mouse, teclado, otros) [Link](#)

¿Qué seleccionas con esta línea de código?

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

☐ El primer elemento `p`` con la clase `five``

☐ Todos los elementos de la clase `p``

☐ Todos los elementos de `p`` con la clase `five``

JavaScript - Document (DOM)
ii. Acceso a elementos DOM (querySelector) [Link](#)

¿Qué método borra el elemento avión?

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



☒ `transportes.shift();`

☐ `transportes.push('avion');`

- ☐ transportes.pop();
- ☐ transportes.unshift();

JavaScript - Array

i. Métodos transformadores (push, pop, shift, unshift, splice, reverse, sort) [\[Link\]](#)

¿Cómo se declaran variables de bloque?



☒ let

☐ var

☐ const

JavaScript - Scoping

ii. Let [\[Link\]](#)

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

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



☒ false

☐ true

☐ undefined

☐ error

JavaScript - Ecuaciones Complejas

ii. Coacción explícita e implícita (string, boolean, number) [\[Link\]](#)

¿Qué imprime en consola?

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

☐ 13 13



☒ 445 85

☐ 85 445

☐ 85 85

JavaScript - Ecuaciones Complejas

iv. Ecuaciones complejas [\[Link\]](#)

Encontrar el índice de 'o' ,

```
let array= ('w','o','r','l','d');
```

☐ array.indexOf('o')



☒ array.indexOf('o')

☐ array.slice('o')

JavaScript - Array

ii. Métodos accesorios (concat, join, slice, toString, indexOf, lastIndexOf) [\[Link\]](#)

¿Qué es JSON?

☐ Un lenguaje de programación orientado a objetos basado en JavaScript

☐ Un servidor de aplicaciones basado en JavaScript



☒ Un formato ligero para el intercambio de datos empleado en JavaScript

☐ Las anteriores respuestas no son correctas

JavaScript - Shallow copy and Deep copy

ii. Deep copy [\[Link\]](#)

En JavaScript, ¿cómo se llama el método que comprueba una expresión regular y devuelve true si se cumple?

☐ check()

☐ exec()

☐ run()



☒ test()

En JavaScript, ¿cómo se llama el objeto que representa una expresión regular?

☐ No hay ningún objeto

☐ ExpReg



☒ RegExp

☐ Rexp

Polymer

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";

Work with object and array data

Specifying paths [\[Link\]](#)

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

- ☐ this.push
- ☐ this.notify
- ☒ this.notifyPath
- ☐ this.set
- ☐ Ninguno de los anteriores

Notify Polymer of a subproperty change
Specifying paths [\[Link\]](#)

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');
```

- ☐ name-modified
- ☐ name-notified
- ☒ name-changed
- ☐ Ningún evento es lanzado

Declare Properties
Property change notification events (notify) [\[Link\]](#)

Selecciona la opción que contenga sólo tipos de datos válidos en Polymer

- ☒ Boolean, Number, String, Array, Object
- ☐ Boolean, Number, String, Enum, Array
- ☐ Boolean, Computed, Number, String, Class
- ☐ Observer, Number, String, Array, Object

Declare Properties [\[Link\]](#)


¿Qué tipo de "binding" es el siguiente?

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

- ☐ Upward
- ☐ Two-way
- ☒ One-way
- ☐ Lazy

Data binding
Anatomy of a data binding [\[Link\]](#)

Si queremos que nuestro componente exponga la posibilidad de modificar su color de texto teniendo como valor por default el azul ¿cuál opción cubrirá 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) }`

Step 5: Theming with custom CSS properties [\[Link\]](#)
Use custom properties [\[Link\]](#)

Son propiedades virtuales cuyo valor se determina basado en otras propiedades

☐ Observers

☐ Computadas

☐ Objects

☐ Behaviors

Observers and computed properties
Computed properties [\[Link\]](#)

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?

- ☐ Usando un dom-if para cada una de las condiciones
- ☐ Usando un dom-if-else
- ☐ Usando un dom-if para una condición y un dom-else para la otra
- ☐ Todas las anteriores son válidas

Data binding helper elements
Conditional templates (dom-if) [\[Link\]](#)

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:

- ☐ `this.splice(persons, 2, 1);`
- ☐ `this.splice('persons', 2, 1);`
- ☐ `this.[2] = null;`
- ☐ Ninguna de las anteriores

Work with object and array data
Specifying paths [\[Link\]](#)

Permite el uso de una función filter como una de sus propiedades

☐ dom-if

☒  dom-repeat

☐ dom-module

☐ dom-filter

Data binding helper elements

Template repeater (dom-repeat) [Link](#)

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

☐ launchEvent

☐ fireEvent

☐ fire



☒ dispatchEvent

Handle and fire events

Fire custom events [Link](#)

De forma declarativa ¿cómo hacer el binding de la propiedad firstName?

☐ <my-component firstName="Joe"></my-component>

☐ <my-component first.Name="Joe"></my-component>



☒ <my-component first-name="Joe"></my-component>

☐ <my-component first-Name="Joe"></my-component>

Data binding

Anatomy of a data binding [Link](#)

Los observers se ejecutan de forma:

☐ Asíncrona



☒ Síncrona

☐ Síncrona y Asíncrona

☐ Los observers son valores de propiedades, por lo tanto no se ejecutan

Observers and computed properties

Observers are synchronous [Link](#)

Si estamos usando un observer para monitorear dos propiedades, estamos hablando de un observer tipo:

☐ Simple

☐ Doble



☒ Complejo

☐ Binario

Observers and computed properties

Complex observers [Link](#)

¿Con qué medios Polymer monitorea cambios en las propiedades...?

☐ Return, New Properties, Data Bindings

☐ Observers, Molecule Elements, Template Repeater



☒ Observers, Computed Properties, Data Bindings

☐ Interfaces, Clases, Prototypes

Observers and computed properties [\[Link\]](#)

Data binding [\[Link\]](#)

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

☐ `<input type="text" on-blur="myListener">`

☐ `<input type="text" on-focus="myListener">`

☐ `<input type="text" on-leave="myListener">`

☐ `<input type="text" lost-focus="myListener">`

Handle and fire events [\[Link\]](#)

Mantener un valor de atributo HTML sincronizado con un valor de propiedad

☐ `reflecttoattribute`

☐ `computed`

☐ `observer`

☐ ninguna de las anteriores

Declare Properties [\[Link\]](#)

Cuantos `<slot>` puede haber en un componente

☐ No puede haber

☐ 42

☐ Tantos como sean necesarios

☐ 1

Shadow DOM concepts [\[Link\]](#)

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

☐ `constructor()`

☐ `connectedCallback()`

☐ `disconnectedCallback()`

☐ `connectedEvent()`

☐ `attributeChangedCallback()`

Custom element concepts

Custom element lifecycle [\[Link\]](#)

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

☐ Simple

☐ Doble

☒ Complejo

☐ Binario

Observers and computed properties
Complex observers [Link](#)

¿De qué forma polymer puede compartir los estilos entre componentes?

☐ Por medio de un style general

☐ Por medio de un componente que tenga estilos



☒ Por medio de apply

☐ No se puede compartir por el shadow dom

Use custom properties
Create custom properties [Link](#)

¿Cuál es la forma correcta de dejar abierta la modificación de un estilo de una clase en un componente?

☐ apply(--nombre-de-la-clase)

☐ var(--nombre-de-la-clase)



☒ @apply(--nombre-de-la-clase)

☐ #apply(--nombre-de-la-clase)

Use custom properties
Create custom properties [Link](#)

Si se tiene una propiedad computada que depende de dos propiedades, de que forma aseguras que en polymer solo se ejecute la función una vez al setear los dos valores

☐ this.set

☐ this.setComputed



☒ this.setProperties

☐ this.setPropertie

Work with object and array data
Batch multiple property changes [Link](#)

Cual de los siguientes elementos en polymer acepta el atributo filter

☐ dom-if



☒ dom-repeat

☐ dom-filter

☐ dom-switch

Data binding helper elements
Template repeater (dom-repeat) [Link](#)

¿Cómo se bindea una variable con el valor de un input con Polymer?



☒ <input value="{{first::input}}">

☐ <input type="{{first::input}}">
☐ <input value="{{input::first}}">
☐ <input value="{{first--input}}">

Data binding
Two-way bindings [link](#)

Cual de las siguientes propiedades, no pertenecen a una propiedad en polymer

☐ notify



☐ readyValue

☐ reflectToAttribute

☐ readOnly

Declare Properties [link](#)

Falta la de un componente de un click