

## Bloque I – Lenguaje de programación Java

- 1) ¿Cuál de los siguientes **NO** es un tipo de datos primitivo en Java?
  - a) byte
  - b) short
  - c) char
  - d) decimal
  
- 2) Las Java wrapper classes son:
  - a) Clases que convierten los tipos primitivos de Java en objetos.
  - b) Clases del sistema de permisos de Java que limita el acceso de las clases a los recursos del sistema operativo.
  - c) Clases que facilitan la serialización de objetos complejos para persistirlos utilizando flujos Java.
  - d) Clases que mantienen el estado de los ThreadLocal durante la sesión del usuario.
  
- 3) En una variable de tipo "int" a la que no se le ha dado valor, ¿Cuál es el valor de inicialización por defecto?.
  - a) null
  - b) 0
  - c) void
  - d) -1.
  
- 4) ¿Cuál de las siguientes sentencias devuelve el número de elementos del array myArray?.
  - a) myArray.size()
  - b) myArray.getSize()
  - c) myArray.length
  - d) len(myArray)
  
- 5) En Java, sobre la palabra clave "final": ¿Cuál de las siguientes respuestas es **INCORRECTA**?
  - a) Aplicada sobre un método, previene que las subclasses lo sobrecarguen.
  - b) Aplicada sobre el parámetro de un método, previene que la variable del parámetro sea reasignada o modificada.
  - c) Aplicada sobre una clase, previene que pueda ser extendida.
  - d) Aplicada sobre la propiedad de una clase, establece la variable como estática y compartida por todas las instancias de la clase.
  
- 6) ¿Cuál de las siguientes respuestas es **CORRECTA** respecto de una clase Java sin modificador de acceso?
  - a) Por defecto es pública y visible para todas las clases.
  - b) Es visible únicamente por las clases de su mismo paquete.
  - c) Únicamente se puede acceder a sus métodos static desde fuera de su paquete.
  - d) Es visible fuera de su paquete, pero no se puede extender.

- 7) ¿Cuál de las siguientes afirmaciones, sobre las diferencias entre las clases Vector y ArrayList es **INCORRECTA**?
- a) Vector tiene un tiempo de acceso más lento que ArrayList.
  - b) Tanto Vector como ArrayList incrementan automáticamente su tamaño al añadir un nuevo elemento.
  - c) ArrayList permite acceso sincronizado pero Vector no.
  - d) Vector incrementa su tamaño con un factor del 50% de su tamaño actual y Vector del 100%.
- 8) ¿Cuál de las siguientes afirmaciones sobre la jerarquía de excepciones de Java es **CORRECTA**?
- a) Error y Exception heredan de Throwable.
  - b) Error y Throwable heredan de Exception.
  - c) Throwable y Exception heredan de Error.
  - d) Ninguna de las anteriores es correcta.
- 9) Teniendo en cuenta lo siguiente: A y B son clases, Z e Y son interfaces. ¿Cuál de las siguientes es una definición **VÁLIDA** de clase?:
- a) class C extends A, B implements Z { ....
  - b) class C extends B implements Z, Y { ...
  - c) class C extends Z, Y implements A, B { ...
  - d) class C implements Z extends A { ...
- 10) ¿Cuál de las siguientes **NO** es una forma de prevenir que un método sea sobrecargado (overriding)?
- a) Definir el método como final.
  - b) Definir el método como static.
  - c) Definir el método como private.
  - d) Definir el método como abstract.
- 11) ¿Qué método de la clase Thread debes invocar para detener la ejecución de un hilo por un tiempo determinado?
- a) sleep()
  - b) stop()
  - c) pause()
  - d) wait()

- 12) ¿Cuál de las siguientes afirmaciones sobre mecanismos de Java Generics es **INCORRECTA**?
- a) Type erasure: la información de tipo genérico no está disponible en tiempo de ejecución.
  - b) Bounded type: se puede restringir la sustitución del tipo genérico utilizando la cláusula extends. Ej: `class Class<T extends ClassB>`
  - c) Wildcard Type: pueden utilizarse el operador '\*' para indicar que el tipo genérico puede ser sustituido por cualquier clase. Ej: `metodo1(List<*> list)`
  - d) Multiple bounds: se pueden especificar varias restricciones sobre la clase que puede sustituir el tipo genérico con el operador '&'. Ej: `class ClassA<T extends ClassB & InterfaceZ>`.

## Bloque II – Arquitectura j2EE. Arquitectura SOA. Paradigma REST. Estándares y protocolos de comunicación entre sistemas.

- 13) ¿Cuál de las siguientes sentencias provocaría que el cliente web de un servlet recibiese un código HTTP 301 ó 302?
- a) `response.close()`
  - b) `response.setHeader("status",302)`
  - c) `response.sendRedirect(url);`
  - d) `requestDispatcher.forward(request, response);`
- 14) ¿En qué fichero se definen por defecto los servlets en una aplicación J2EE?
- a) `WEB-INF/web.xml`
  - b) `web/classpath.xml`
  - c) `web/config.xml`
  - d) `web/app.xml`
- 15) ¿Qué tipo de objeto debes utilizar para llamar a un procedimiento almacenado desde una conexión jdbc?
- a) `Statement`.
  - b) `PreparedStatement`.
  - c) `DynamicStatement`.
  - d) `CallableStatement`.
- 16) Partiendo de un api REST sobre un recurso "Usuario". Según el paradigma REST, ¿Cuál es la operación HTTP y la ruta más adecuada para crear un recurso de tipo Usuario?
- a) POST sobre el recurso `/usuarios/`.
  - b) PUT sobre el recurso `/usuarios/<userid>` indicando en la ruta el id del usuario.
  - c) CREATE sobre el recurso `/usuarios/`.
  - d) GET sobre el recurso `/usuarios/` para obtener el id del nuevo recurso y después utilizar CREATE `/usuarios/<userid>`.

- 17) Tras crear el nuevo recurso usuario, ¿Qué deberá devolver el servidor en la cabecera "location"?
- a) El host y puerto del servidor que aloja el recurso.
  - b) El URI del recurso que se acaba de crear.
  - c) La IP del cliente que ha creado el recurso o los posibles proxies intermedios.
  - d) En el caso de tratarse de un recurso geolocalizado, las coordenadas del mismo y el sistema de referencia utilizado (SRID).
- 18) Según el paradigma REST, ¿Cómo puede el cliente de un API REST indicar al servidor en qué formato quiere recibir la representación del recurso que solicita (json, xml, etc.)?
- a) El cliente no debe poder seleccionar nunca el tipo de representación.
  - b) Estableciendo el tipo de contenido en una cookie asociada al dominio del servidor.
  - c) Utilizando la cabecera content-type.
  - d) Enviando una petición HTTP de tipo OPTIONS previa a la consulta del recurso.

## Bloque III – Frameworks y herramientas de desarrollo J2EE. Herramientas de gestión de configuraciones.

- 19)Cuál de los siguientes **NO** es un módulo de spring:
- a) spring-core
  - b) spring-jdbc
  - c) spring-beans
  - d) spring-sharp
- 20) ¿Cuál es el scope, por defecto, para un componente de spring?
- a) singleton
  - b) request
  - c) session
  - d) application
- 21) ¿Cuál de las siguientes es una condición imprescindible para definir una clase como Entity de Hibernate?
- a) Debe establecer una propiedad como identificador que represente una clave primaria de la tabla subyacente.
  - b) Debe tener un constructor sin argumentos.
  - c) Se debe utilizar la anotación @Table para indicar la tabla o vista que almacenará la entidad.
  - d) Debe implementar el interfaz Serializable.
- 22) De los siguientes métodos, ¿Qué tipo **NO** puede ser testado de forma directa utilizando un test Junit?
- a) Método con el calificador de acceso static.
  - b) Método con el calificador de acceso final.
  - c) Método con el calificador de acceso private.
  - d) Método con el calificador de acceso protected.

23) ¿Qué se entiende por test Fixture?

- a) El conjunto de clases que son objeto de prueba por el test.
- b) El estado fijo de un conjunto de objetos que se utiliza como punto de partida del test.
- c) El conjunto de objetos creados durante la ejecución del test.
- d) La parte de código del test JUnit anotada con @TearDown.

## Bloque IV: Frameworks de desarrollo web-Javascript.

24) ¿Qué mostrará por consola este código?

```
var a = '3'; var b = 3; console.log((a == b) + " " + (a === b));
```

- a) true true
- b) true false
- c) false true
- d) false false

25) ¿Qué mostrará por consola este código?

```
var user= {age:12, name:"Billy Batson"};
delete user.age;
console.log(user)
```

- a) {age: null, name: "Billy Batson"}
- b) {age: 12, name: "Billy Batson"}
- c) {name: "Billy Batson"}
- d) No imprime nada, se produce un error en la sentencia delete user.age.

26) ¿Cuál de las siguientes es una definición válida de un string JSON?

- a) var jsonStr = { "name": "John" };
- b) var jsonStr = { 'name' , 'John' };
- c) var jsonStr = ("name": "John" );
- d) var jsonStr = [ name: 'John' ];

27) ¿Cuál es el resultado de la ejecución del siguiente código?

```
'use strict';
name = "Bruce Banner";
console.log(name)
```

- a) Se ejecuta sin errores, mostrará el texto "Bruce Banner".
- b) Provocará una excepción "name is not defined". El uso de la sentencia "use strict" obliga a declarar las variables antes de utilizarse.
- c) Provocará una excepción "type not defined for variable 'name'". El uso de la sentencia "use strict", obliga a definir el tipo de dato de las variables.
- d) Ninguna de las anteriores.

28) ¿Cuál es el resultado de la ejecución del siguiente código?

```
function counter() {  
  var _value = 0;  
  return {  
    add: function(increment) { _value += increment; },  
    get: function() { return _value; }  
  }  
}  
var c1 = counter();  
c1.add(3);  
var c2 = counter();  
c2.add(5);  
c2.add(7);  
console.log(c1.get() + " " + c2.get())
```

- a) 7 7
- b) 3 7
- c) 3 12
- d) NaN NaN

29) ¿Cuál es el resultado de la ejecución del siguiente código?

```
var values = ["one", "two", "three"];  
values.push("four");  
var last = values.pop();  
console.log(values);
```

- a) "one", "two", "three".
- b) "four", "one", "two".
- c) "two", "three", "four".
- d) "one", "two", "three", "four".

30) ¿Cuál de las siguientes **NO** se una forma válida de crear un objeto genérico en javascript?

- a) var a = new Object();
- b) var a = {};
- c) var a = "";
- d) var a = Object.create({});

31) ¿Qué es ESLint?

- a) Una librería de test unitarios para Javascript.
- b) Una librería para empaquetado de código Javascript.
- c) Una librería para detección de bugs en código Javascript.
- d) Una librería para ofuscar código Javascript.

- 32) ¿Cuál de las siguientes es una declaración **VÁLIDA** de la importación de un módulo en Javascript?
- a) `source * from modules.my-module;`
  - b) `import * as myModule from '/modules/my-module.js';`
  - c) `include * from '/modules/my-module.js';`
  - d) `requirement { myFunction as mf } from '/modules/my-module.js';`
- 33) ¿Qué se entiende por promesas (“promises”) en Javascript?
- a) Un requerimiento que se establece en la cabecera de un módulo para importar automáticamente otros módulos.
  - b) Funciones asignadas para ejecutar código en respuesta a los eventos de interacción del usuario.
  - c) Funciones de callback que se pueden invocar durante la inicialización de un módulo.
  - d) Un mecanismo para la ejecución asíncrona de código.
- 34) ¿Qué se entiende por “reactividad” en el ámbito de las aplicaciones web-Javascript?
- a) La aplicación es capaz de sobreponerse a pérdidas de la conexión y enviar los datos en espera cuando se restablece la conectividad.
  - b) El renderizado de las vistas es gestionado por el framework javascript, de forma que éste se desencadena automáticamente ante un cambio en los datos de los que depende la vista.
  - c) La visualización de la aplicación se ajusta de forma dinámica a las dimensiones del dispositivo del usuario.
  - d) El cliente web recibe desde el servidor paquetes de datos mediante web-sockets sin que sea necesario hacer pooling desde el cliente.
- 35) ¿Cuál de los siguientes **NO** es un mecanismo de almacenamiento estandarizado por la w3c que puede ser utilizado en aplicaciones Javascript?
- a) IndexedDB.
  - b) Local Storage.
  - c) Request Storage.
  - d) Session Storage.
- 36) ¿Cuál de los siguientes puntos **NO** se define en el fichero “package.json” en una aplicación nodejs? (Ej: Vue, Angular, etc.)
- a) Nombre y versión de la aplicación.
  - b) Las dependencias de la aplicación con otros módulos.
  - c) Los perfiles de ejecución de la aplicación: dev, start, build, etc.
  - d) Los plugins y configuración específica de webpack.
- 37) ¿Qué hook de Vue se ejecuta justo después del renderizado inicial de un componente?
- a) created
  - b) mounted
  - c) updated
  - d) installed

38) ¿Cuál de las siguientes **NO** es una directiva de Vue?

- a) v-show
- b) v-if
- c) v-model
- d) v-console

39) ¿Cuál es el resultado en consola de la ejecución del siguiente código?

```
f = function () {  
    this.a = 1;  
}  
let o = new f();  
f.prototype.b = 2;  
console.log(o.a + " " + o.b);
```

- a) 1 2
- b) 1 undefined
- c) 1 null
- d) Se producirá un error al intentar fijar una nueva propiedad en el prototipo de la función "f".

40) ¿Cuál de las siguientes afirmaciones es **INCORRECTA** sobre las restricciones de ejecución de un "web worker" Javascript?

- a) Los web workers tienen un contexto global diferente al del navegador.
- b) Los web workers no pueden modificar el árbol DOM.
- c) Los web workers tienen un acceso limitado al objeto window.
- d) Los web workers no tienen acceso al almacenamiento IndexedDB.

## Preguntas de reserva

- 41) ¿Cuál de las siguientes implementaciones de la interfaz Map, permite recuperar los elementos almacenados ordenados por la clave principal? (Bloque I)
- a) TreeMap.
  - b) LinkedHashMap.
  - c) HashMap.
  - d) WeakHashMap.
- 42) ¿Cuál de los siguientes métodos HTTP **NO** es idempotente? (Bloque II)
- a) GET.
  - b) POST.
  - c) PUT.
  - d) HEAD.
- 43) ¿Cuál de los siguientes **NO** es un scope válido de Spring? (Bloque III)
- a) interaction
  - b) session
  - c) global-session
  - d) application
- 44) ¿Cuál de las siguientes **NO** es una anotación válida para registrar una clase como componente de Spring? (Bloque III)
- a) @Controller
  - b) @LogicController
  - c) @Component
  - d) @Repository
- 45) ¿Cuál es la función de un “service worker” en Javascript? (Bloque IV)
- a) Proporcionar a los módulos Javascript acceso a APIs del dispositivo como la geolocalización, cámara, etc.
  - b) Ejecutar trabajos en segundo plano para evitar que la experiencia del usuario sufra por tiempos de espera altos.
  - c) Hacen de proxy entre el navegador y la red para permitir que la aplicación web trabaje de forma desconectada.
  - d) Se encargan de la carga dinámica de módulos Javascript bajo demanda.