



## Anexo 2

### Pruebas para la obtención de títulos de Técnico y Técnico Superior

#### Convocatoria correspondiente al curso académico 2023-2024

(Resolución de 29 de diciembre de 2023 de la Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial)

DATOS DEL ASPIRANTE			FIRMA
Apellidos:			
Nombre:	D.N.I. N.I.E. o Pasaporte:	Fecha:	

Código del ciclo:	Denominación completa del título:
IFCS02	Técnico Superior en Desarrollo de Aplicaciones Multiplataforma
Clave o código del módulo:	Denominación completa del módulo profesional:
0487	Entornos de desarrollo

INSTRUCCIONES GENERALES PARA LA REALIZACIÓN DE LA PRUEBA
<ul style="list-style-type: none"><li>El examen tendrá una duración de 2'5 horas (150 minutos)</li><li>La prueba consta de un examen tipo test con cuatro opciones de las cuales solamente una es correcta.</li><li>Cada pregunta se responderá en el espacio dejado al efecto en la hoja de respuestas. Se rellenarán los recuadros para señalar la respuesta seleccionada.</li><li>Si se quiere rectificar una respuesta contestada, se borrará toda la casilla de la respuesta incorrecta con Tipp-Ex o corrector, tal y como se puede apreciar en el siguiente ejemplo:</li><li>Se elimina la selección de <b>b</b> para seleccionar la opción <b>d</b>: <input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input checked="" type="checkbox"/> d</li><li>Se dispondrá de una hoja para borrador (o de varias si se requieren), que será proporcionada por el centro. Esa hoja se entregará obligatoriamente al final junto con el examen, si bien nada de lo escrito en la hoja de borrador se valorará en la corrección.</li><li>Sólo se utilizará bolígrafo negro o azul y Tipp-Ex o corrector, no permitiéndose usar bolígrafo rojo, lapicero, etcétera.</li><li>No se podrá emplear ningún dispositivo electrónico.</li><li>Cualquier tachadura o borrón en una respuesta podrá invalidar toda la puntuación de la pregunta asociada.</li></ul>

CRITERIOS DE CALIFICACIÓN Y VALORACIÓN
<ul style="list-style-type: none"><li>El test se calificará sobre 10 puntos. Todas las preguntas se calificarán equitativamente con la misma cantidad de puntos. En cada pregunta se plantearán varias respuestas, y se deberá señalar la única que se considere correcta, según el caso. Cada respuesta correcta que se marque se valorará con 0,25 puntos, y si se marca alguna incorrecta, se descontarán 0,075 puntos. Si no se está seguro de si una respuesta es correcta o no, y no se marca, no sumará ni restará puntos.</li><li>Calificación final del módulo profesional:<ul style="list-style-type: none"><li>El alumno obtendrá en el módulo profesional una calificación entera entre 1 y 10.</li><li>Dicha calificación se calculará:<ul style="list-style-type: none"><li>Si la calificación conseguida en la prueba es inferior a 5: se truncará dicha calificación.</li><li>Si la calificación conseguida en la prueba es igual o superior a 5 y los decimales:<ul style="list-style-type: none"><li>Son inferiores a 0'5: se redondeará al entero inferior más próximo.</li><li>Son iguales o superiores a 0'5: se redondeará al entero superior más cercano.</li></ul></li></ul></li><li>La anterior regla tiene una excepción: las notas de examen inferiores a 1 se redondearán a 1.</li></ul></li></ul>

## CONTENIDO DE LA PRUEBA

- 1) ¿Qué aportan fundamentalmente las metodologías ágiles e iterativas a los proyectos SW actuales?
  - a. Hacen innecesaria (o al menos irrelevante) la documentación.
  - b. Hacen que los proyectos sean más baratos.
  - c. Hacen que los proyectos duren menos tiempo
  - d. Hacen que los requisitos queden mejor definidos.
- 2) Los requisitos del software incluyen:
  - a. Los funcionales y no funcionales
  - b. Los funcionales y los imperativos
  - c. Los declarativos y los imperativos
  - d. Los imperativos y no imperativos
- 3) Un lenguaje de programación que usa mnemotécnicos, con traducción directa a operaciones soportadas por la CPU. Se trata de...
  - a. Un lenguaje interpretado
  - b. Un lenguaje compilado
  - c. Un lenguaje declarativo
  - d. Un lenguaje ensamblador
- 4) Son lenguajes imperativos:
  - a. Ensamblador, Java, lenguaje máquina
  - b. Todos los lenguajes son imperativos, excepto el lenguaje máquina
  - c. Java, SQL
  - d. Prolog, SQL
- 5) Con el IDE Eclipse parado en un breakpoint (punto de ruptura), puedo:
  - a. Lanzar una excepción desde ese punto de ruptura.
  - b. Avanzar hasta el fin de la función/procedimiento actual
  - c. Deshacer las últimas 3 instrucciones ejecutadas, para repasar cómo se ha llegado al breakpoint.
  - d. Reducir el consumo de memoria de la aplicación que está siendo depurada (garbage collector)
- 6) ¿Por qué el ciclo en cascada NO era apropiado para los proyectos ingenieriles clásicos (arquitectura, obras públicas, industriales, etc.) ?
  - a. Porque tienen requisitos más cambiantes (en comparación a los proyectos Software actuales).
  - b. Porque carecen de etapa de diseño.
  - c. Porque no era tan necesario llevar una metodología tan estricta.
  - d. El ciclo en cascada sí era apropiado para muchos de los proyectos ingenieriles clásicos.
- 7) Una clase de equivalencia no válida
  - a. Sirve para agrupar los casos de prueba que no se tendrán en cuenta
  - b. No existen, ya que todas las clases de equivalencia son válidas para realizar pruebas
  - c. Agrupan los casos de prueba en los que la aplicación debe producir un error determinado
  - d. Agrupan los casos de prueba en los que la aplicación tiene un comportamiento sin definir

- 8) El objetivo del cubrimiento en las pruebas de caja blanca es:
- Establecer casos de prueba con valores en el límite del rango.
  - Comprobar que todos los caminos se pueden ejecutar.
  - Obtener casos de prueba representativos de las condiciones normales de operación
  - Establecer clases de equivalencia que disminuyan el número de pruebas.
- 9) Durante la depuración de un programa, se utiliza la posibilidad del IDE para cambiar el valor de una variable del proceso en ejecución. Esta funcionalidad se incluye en:
- Los puntos de ruptura condicionales
  - La refactorización de código
  - La inspección de variables ("watch" o "inspect")
  - No se puede cambiar el valor de una variable una vez lanzado el programa, es necesario hacerlo antes con un "conditional value"
- 10) La realización de pruebas \_\_\_\_\_ nos permite detectar errores de cada parte del programa por separado.
- de validación
  - semánticas
  - de integración
  - unitarias
- 11) Durante la depuración de un programa ¿para qué se utilizan las clases de equivalencia?
- Para determinar los puntos interesantes del código donde colocar puntos de ruptura
  - Para descartar algunos errores comunes que no son interesantes en el proceso de verificación
  - No se utilizan durante la depuración, sino en una fase anterior del desarrollo del software
  - Para que, una vez encontrado el error, el IDE encuentre automáticamente el origen del mismo, posiblemente en una función anterior
- 12) El código ...
- ```
class Test {  
    @Test  
    public void f() throws Exception {  
        assertThrows(Exception.class, ()->{throw new Exception();}, "");  
    }  
}
```
- ... produciría en Eclipse, al ejecutarse con el *runner* de Junit, un reporte de...
- Color azul (assertion exception)
  - Color rojo (unhandled exception)
  - Color verde (ok)
  - Ninguno de los anteriores
- 13) Un fichero .class ¿cómo puede ejecutarse?
- En Java, los ficheros .class no se ejecutan, sino los .jar
  - Utilizando directamente el intérprete de java
  - Utilizando el intérprete de java, tras procesarlos con la utilidad jar
  - En Java, los ficheros .class no se ejecutan, sino los .project



- 14) En los diagramas de casos de uso, un actor se representa con...
- Un óvalo
  - Una línea que une un óvalo con un rectángulo
  - Una estrella
  - Un monigote
- 15) Ciclo de vida en el que no está previsto repetir alguna fase o tarea
- En espiral
  - En cascada
  - Iterativo incremental
  - Basado en prototipos
- 16) ¿Cómo se llama el cambio consistente en mejorar la legibilidad de programas, sin alterar la funcionalidad de este?
- Evolución
  - Refactorización
  - Corrección
  - Versión "release"
- 17) Un lenguaje de programación procesa todo el código fuente previamente a su ejecución. Durante la ejecución, ya no se utiliza el código fuente, sino el resultado de dicho proceso. Es un sistema:
- Sandbox
  - Transpilado
  - Interpretado
  - Compilado
- 18) Con las clases de equivalencia diseñamos casos de prueba
- Con valores en el límite del rango admitido
  - Con valores fuera del rango admitido
  - Con valores fuera y dentro del rango admitido
  - Con valores dentro del rango admitido
- 19) ¿Cuáles de los siguientes entornos de desarrollo son propietarios? (su código fuente no es público)
- Eclipse
  - IntelliJ Idea Ultimate Edition
  - Microsoft Visual Studio Code
  - Apache Netbeans
- 20) De todos los tipos de software, es el encargado de gestionar los recursos hardware de un computador.
- el código objeto
  - la aplicación
  - el compilador
  - el sistema operativo

- 21) ¿En qué momento se suelen realizar las pruebas Beta Test?
- a. Antes de las pruebas de integración.
  - b. En el mismo entorno productivo de la aplicación, una vez instalada y desplegada
  - c. A lo largo de la etapa de mantenimiento.
  - d. Antes de realizar las pruebas unitarias.
- 22) Tipo de pruebas realizado paralelamente al desarrollo, incluso sin haber obtenido aún un prototipo de programa
- a. Las pruebas de integración
  - b. No se pueden realizar pruebas si el proyecto realizado no puede siquiera ejecutarse
  - c. Las pruebas unitarias
  - d. Las pruebas preliminares de integración
- 23) Indica la afirmación correcta
- a. Un intérprete puede estar implementado en un lenguaje ensamblador, e incluso en lenguaje máquina
  - b. Tras la compilación de un programa se obtiene el código que siempre es directamente ejecutable por el procesador.
  - c. Los programas intérpretes sólo se compilan una vez, y el fichero resultado es un intérprete que sirve para todas las plataformas
  - d. Los microprocesadores pueden ejecutar directamente dos tipos de código: el código máquina y el código ensamblador.
- 24) La extensión \_\_\_\_\_ es característica de proyectos empaquetados en Java
- a. .jar
  - b. .class
  - c. .project
  - d. .java
- 25) En GIT, la operación push, se realiza...
- a. entre la copia de trabajo y el repositorio local
  - b. entre el repositorio local y el repositorio remoto
  - c. entre la copia de trabajo y el repositorio remoto
  - d. ninguna de las anteriores
- 26) ¿Qué utilidad reúne todos los componentes requeridos y crea un archivo ejecutable?
- a. Compilador
  - b. Ensamblador
  - c. Intérprete
  - d. Enlazador

- 27) Un programa necesita que su código fuente esté disponible durante su ejecución. Se deduce que es un programa
- Transpilado
  - Compilado
  - Todos los programas necesitan de su código fuente mientras se están ejecutando
  - Interpretado, quizás mediante un JIT (Just in time compiler)
- 28) Los siguientes lenguajes se consideran orientados a objetos
- Java y C++
  - C++ y C
  - Java y SQL
  - C++ y HTML
- 29) Tener deficiencias en la fase de \_\_\_\_ es la principal causa del gran porcentaje de fracasos de los proyectos software
- Depuración
  - Instalación
  - Documentación
  - Análisis
- 30) Cuáles de las siguientes son herramientas de control de versiones de código fuente:
- Mercurial, Subversion
  - GIT, javadoc
  - Subversion, javadoc
  - GIT, JUnit
- 31) En GIT, el comando pull equivale aproximadamente a
- Un clone y un commit
  - Un fetch y un merge
  - Un cambio de rama (checkout)
  - Un cambio de rama (checkout) y un merge
- 32) En GIT, ¿cómo se relaciona un repositorio remoto (tipo github) con los repositorios locales?
- El repositorio local puede tener una copia de todas las ramas y todas las versiones de cada rama.
  - El repositorio local sólo puede tener una de las ramas del repositorio remoto, pero todas las versiones de dicha rama.
  - El repositorio local tiene todas las ramas existentes en el remoto, pero sólo la última versión (HEAD) de cada una.
  - Uno de los repositorios locales llamado "maestro" tiene todas las ramas y sus versiones, y puede invitar a colaboradores a participar en una rama específica.



e.

33) Para integrar los cambios de la rama B en la rama A se puede usar la siguiente secuencia de comandos

- a. git pull A; git merge B
- b. git merge origin/A origin/B
- c. git checkout A; git merge B
- d. git merge B A; git confirm -f

34) Para comenzar a usar la rama A de un repositorio remoto cuya URL es "https://github.com/remoto.git" que aún no se tiene localmente se pueden usar los comandos

- a. git clone "https://github.com/remoto.git" refs/A
- b. git fetch "https://github.com/remoto.git"; git checkout A
- c. git clone "https://github.com/remoto.git" repo; cd repo; git checkout A
- d. git fetch "https://github.com/remoto.git" A

35) La refactorización del código fuente

- a. Elimina funcionalidad antigua.
- b. Puede conseguir nueva funcionalidad.
- c. Puede conseguir que el código sea más rápido.
- d. Puede cambiar la cantidad de métodos y clases

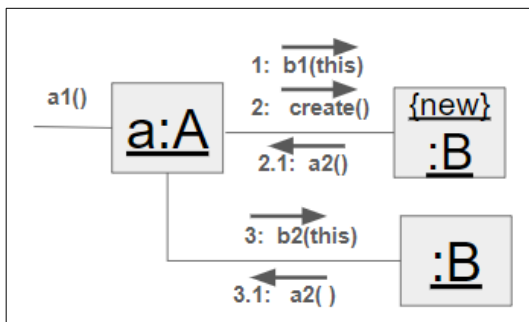


36) ¿Cuál de los siguientes diagramas de colaboración describiría el comportamiento del método `a1()` de la clase `A`, detallado en el siguiente código de JAVA?

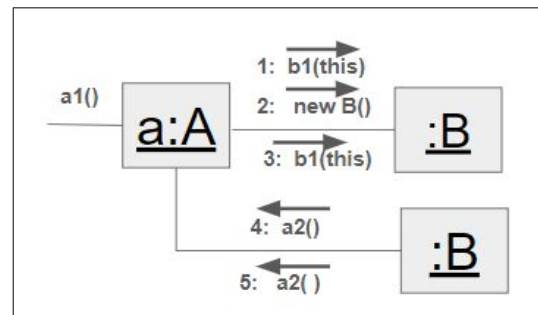
```
public class A {
    private B b;
    public void a1() {
        b.b1(this);
        B b2 = new B();
        b2.b1(this);
    }
    public void a2() {}
}
```

```
public class B {
    private A a;
    public void b1(A a) {
        a.a2();
    }
    public void b2() {
        a.a2();
    }
}
```

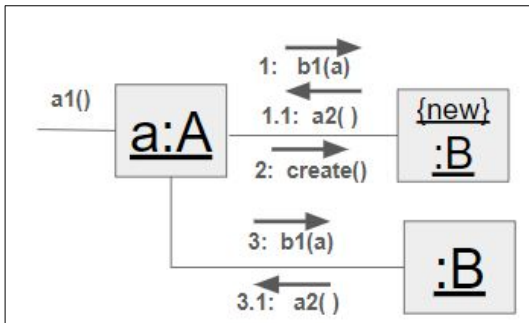
a.



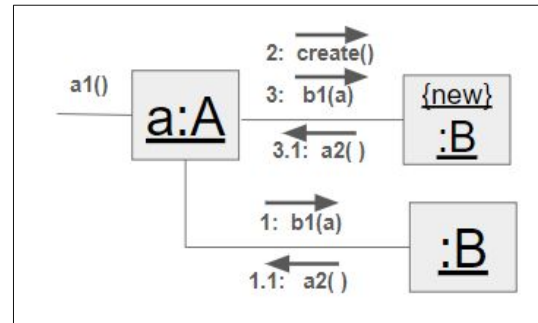
b.



c.



d.



37) La operación GIT para pasar un fichero de estado "staged" a "unmodified" es

- a. add
- b. commit
- c. checkout
- d. push

38) El tag de javadoc `@param` continúa inmediatamente con ...

- a. el identificador del parámetro
- b. el tipo del parámetro
- c. una descripción
- d. nada

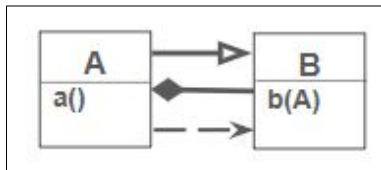




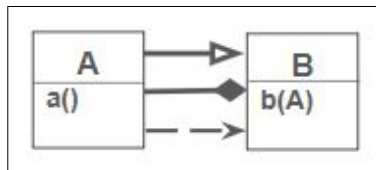
39) Dado el siguiente código JAVA, ¿cuál de los 4 diagramas de clases lo describiría?

```
public class A {  
    private Collection<B> b;  
  
    public void a() {  
        b.add(new B());  
    }  
}
```

a.

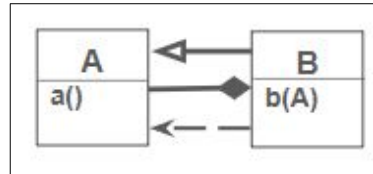


c.

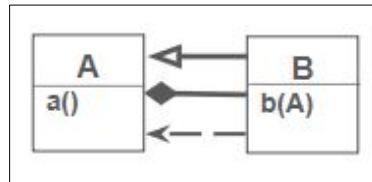


```
public class B extends A {  
  
    public void b(A a) {  
        a.a();  
    }  
}
```

b.



d.



40) En el contexto de procesos de refactorización, ¿qué *BAD SMELL* intenta solventar el principio DRY?

- a. Métodos con mucho código
- b. Clases con mucho código
- c. Métodos con muchos parámetros
- d. Código duplicado