

TEST-CON-RESPUESTAS.pdf



Akita



Introducción a la ingeniería del



software 2º Grado en Ingeniería



Informática

**Escuela Técnica Superior de Ingeniería
Informática Universidad de Málaga**

Reservados todos los derechos.

No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

1. Según la Ingeniería del Software:
 - a. Es necesario encontrar un equilibrio entre los recursos disponibles para un proyecto y los resultados que se pueden entregar
 - b. Cuando se llega al plazo final de un proyecto hay que cerrarlo en el estado en el que está
 - c. Hay que entregar un sistema completamente libre de errores, aunque implique superar el presupuesto y el plazo
-
2. Una empresa de producción de Software recibe el encargo de realizar un proyecto de gran envergadura que involucra a varios equipos de desarrollo durante un periodo de varios meses de trabajo. Siguiendo un enfoque de ingeniería del Software, ¿cómo deberían de abordar el proyecto?
 - a. Seleccionando un modelo de proceso software y siguiéndolo puntualmente
 - b. Seleccionando un modelo de proceso software y adaptándolo a las necesidades del proyecto y de la empresa
 - c. No es conveniente considerar ningún modelo de proceso software para este tipo de proyecto, pues ralentizaría el proceso de producción
-
3. Una empresa debe estimar el coste de producción de un software personalizado para un cliente particular. Según conversaciones con el cliente, se pretende dar soporte a dicho software durante un periodo a los 5 años. Bajo estas circunstancias, ¿cómo se espera que esté distribuido el coste total?
 - a. Los costes de evolución suelen superar a los costes de desarrollo
 - b. Los costes de desarrollo suelen equipararse a los costes de evolución
 - c. Los costes de desarrollo suelen superar a los costes de evolución
-
- 4.Cuál de las siguientes afirmaciones es cierta:
 - a. Cuando un componente software se deteriora, basta con reemplazarlo por otro igual
 - b. Cada vez que se produce un cambio en el software, puede haber un incremento del índice de fallos
 - c. Los programas software funcionan y son útiles indefinidamente porque no se desgastan ni dependen de su entorno
-
5. En medio de una iteración del proyecto actual, en el que estamos usando Scrum, el cliente ha insistido en incluir tres nuevos requisitos. ¿Qué decisión debe tomar el Dueño del Producto (Product Owner) tras evaluar el coste de los nuevos requisitos?
 - a. Incluir un nuevo desarrollador en el equipo que se encargue de los nuevos requisitos
 - b. Sustituir algunos de los requisitos que estaban planeados y que aún no se habían implementado
 - c. Alargar la duración de la iteración el tiempo necesario
-
6. Un cliente con el que hemos trabajado durante los últimos cinco años quiere ampliar parcialmente el último sistema que le desarrollamos. Según el estudio inicial, harán falta 3 personas durante 2 meses. Por tanto:
 - a. Como es un proyecto que necesita muchas pruebas de aceptación, el Proceso Unificado es el más aconsejable
 - b. Se puede plantear sin problema con una metodología en cascada porque ya tenemos experiencia y no es complejo
 - c. Vamos a escoger una metodología ágil porque no tendremos que ver al cliente durante esos dos meses

7. Las estimaciones sobre el nuevo sistema que se va a desarrollar en la empresa indican que se va a tardar 30 meses en desarrollarlo, que implicará a 5 de los 7 departamentos de la empresa y que va a suponer un cambio estratégico en el funcionamiento del sistema de la empresa. La metodología de desarrollo que se debería usar es:
- Ágil, para poder empezar el desarrollo sin tener que hacer análisis de requisitos
 - Iterativo, porque se pueden incluir mejor los cambios en los requisitos
 - En cascada, para garantizar que se definen correctamente todos los requisitos antes de implementar
-
8. En el siguiente desarrollo vamos a usar la Programación Extrema
- Gracias a la propiedad compartida del código, facilitaremos la comunicación con los clientes
 - No podremos usar la programación extrema porque los clientes no tienen conocimientos técnicos
 - Intentaremos que en las parejas de programadores haya uno más experto y otro más novato
-
9. Si en un diagrama de clases las clases A y B están relacionadas mediante una asociación con cardinales en los extremos de muchos a muchos, quiere decir:
- Que un objeto de la clase B puede estar asociado con varios objetos de la clase A a lo largo de su ciclo de vida, pero con solo uno en cada momento de su ciclo de vida
 - Que un objeto de la clase A solo puede estar asociado con un objeto de la clase B, pero que se puede instanciar muchos objetos de la clase A y de la clase B
 - Que un objeto de la clase A puede estar asociado en un momento dado con varios objetos de la clase B
-
10. En la figura "Diagrama de casos de uso con actor secundario" se muestra un diagrama de casos de uso en el que el actor principal, el Usuario, usa el sistema en un caso de uso en el que el actor "Sistema Externo" es un actor secundario.

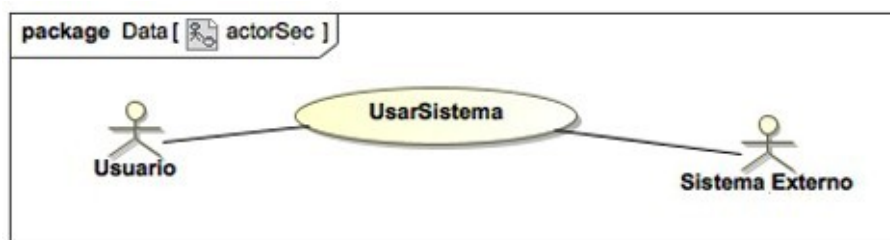
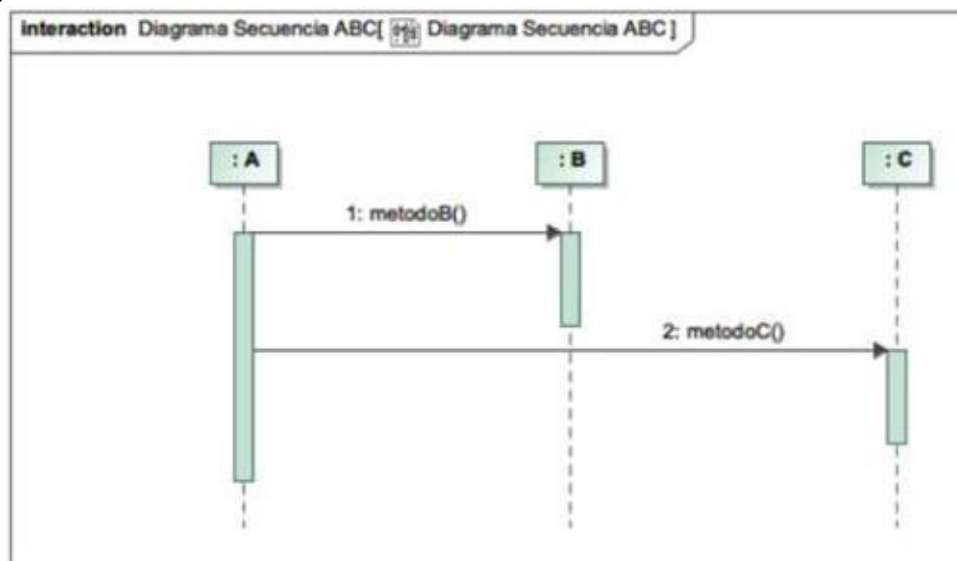


Figura "Diagrama de casos de uso con actor secundario"

- El actor secundario actúa de interfaz entre el actor principal y el sistema
- El actor secundario ayuda al sistema a conseguir el objetivo del actor principal
- El objetivo del actor principal coincide plenamente con el objetivo del actor secundario

11. En el diagrama de secuencia ABC:



- La clase A tiene un atributo de la clase B y otro atributo de la clase C
- Hay una relación de composición entre A y B y otra relación de composición entre B y C
- Las clases B y C especializan a una clase común, por lo que A puede escoger el método al que llamar

12. La figura "Diagrama de secuencia de las clases A y B y mensaje(parámetro)" muestra un diagrama de secuencia donde interactúan un objeto de la clase A y un objeto de la clase B.

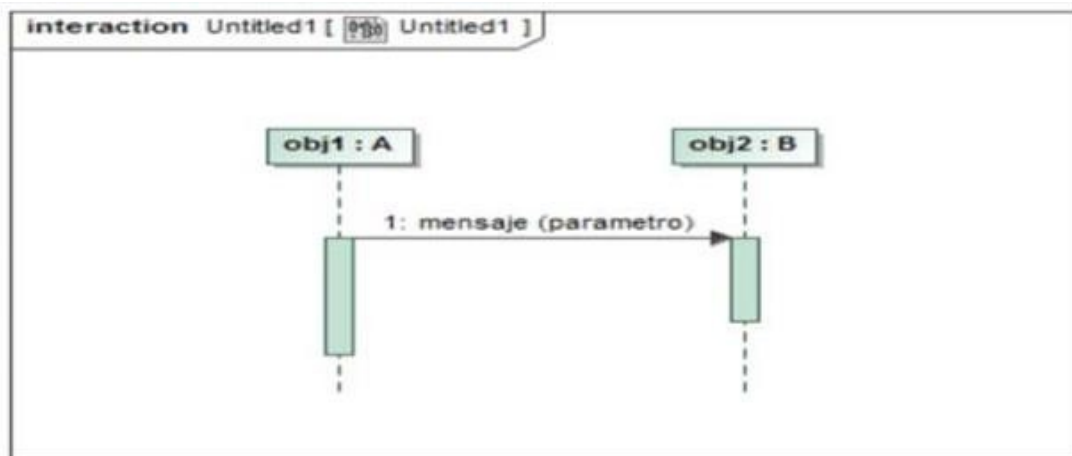
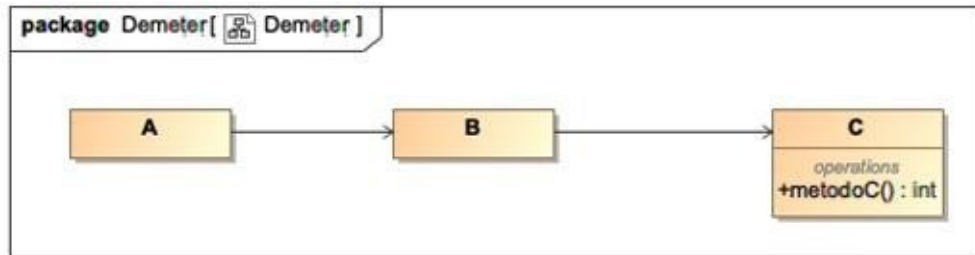


Figura "Diagrama de secuencia de las clases A y B y mensaje(parametro)"

Atendiendo al diagrama de secuencia anterior y sabiendo que mensaje(parámetro) es un método definido en nuestra aplicación, ¿Dónde debe definirse dicho método?

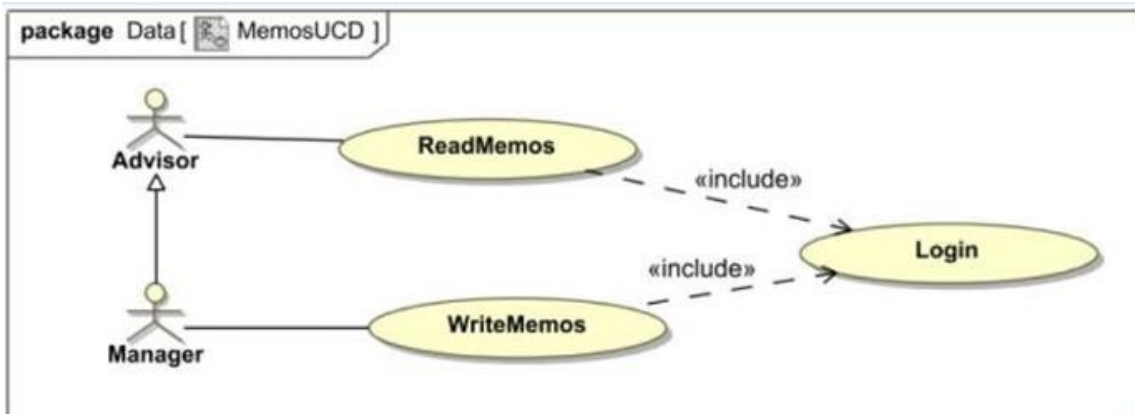
- En la clase B
- En la clase A y en la clase B
- Ni en la clase A ni en la clase B
- En la clase A

13. Si los objetos de la clase A necesitan acceder al valor devuelto por el método metodoC de la clase C, para modificar el diseño cumpliendo la Ley de Demeter:



- La clase B debería ofrecer un método para acceder al valor devuelto por el método metodoC
- La clase B debería ofrecer un método para acceder al objeto de la clase C: getC()
- La clase C debería especializar B

14. En el diagrama de casos de uso MemosUCD:



- El manager puede ejecutar los casos de uso “Escribir Memorando” y “Leer Memorando”
- El caso de uso “Login” no es visible para ninguno de los actores, por lo que no pueden ejecutarlo
- El manager solo tiene que hacer login cuando ejecute el caso de uso “Escribir Memorando”

15. La arquitectura Map-Reduce está especialmente indicada para:

- Trabajar en sistemas con arquitectura Modelo-Vista-Controlador
- Resolver problemas que se pueden resolver con proceso masivo en paralelo
- Reducir el espacio de trabajo de un problema muy complejo

16. Desde el punto de vista arquitectónico, un navegador web:

- Ejecuta la función de controlador, donde la página web es el cliente, el navegador el controlador y la fuente de la URL, el servidor
- Ejecuta la función de cliente, que se conecta a un servidor remoto
- Ejecuta la función de servidor, porque da servicio al usuario, que es el cliente
- Ejecuta la función de cliente cuando accede a la página que le pide el usuario y de servidor cuando le muestra el contenido de la pagina

17. La figura "Diagrama de secuencia con creación de objetos" muestra un diagrama de secuencia en el que un objeto de la clase A crea un objeto de la clase B y, posteriormente, un objeto de la clase C.

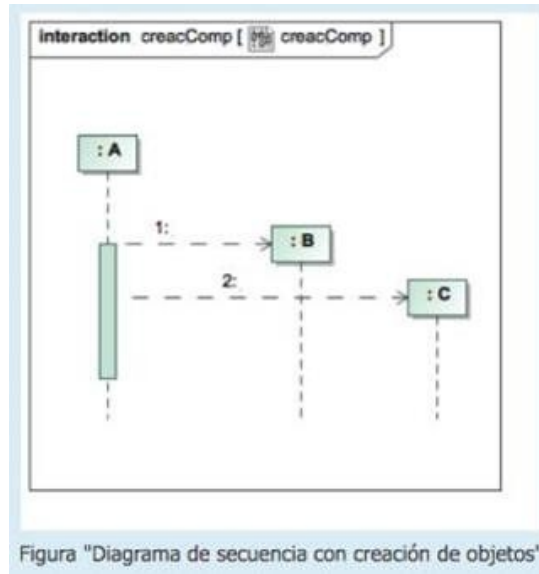


Figura "Diagrama de secuencia con creación de objetos"

Indique para la figura anterior la respuesta que es más probable que sea correcta:

- a. La clase A tiene una relación de composición con las clases B y C. A es el compuesto y B y C los componentes
 - b. Hay una asociación entre la clase A y la B en la que A es cliente de B. La clase C especializa la clase B
 - c. La clase B tiene una relación de composición con las clases A y C. B es el compuesto y A y C los componentes
 - d. La clase A tiene una relación de especialización con las clases B y C. Las clases B y C especializan A
-
18. Un sistema informático de análisis taxonómico de proteínas obtiene datos de un experimento de secuenciación de una muestra metagenómica. Esos datos son pasados al módulo de alineamiento, que las alinea con las proteínas presentes en una base de datos. Los alineamientos resultantes sirven de entrada al módulo de asignación taxonómica, que produce un informe con el análisis taxonómico. Este sistema tiene una arquitectura basada en:
- a. Cliente/servidor
 - b. Filtros y tuberías
 - c. Capas
-
19. En una arquitectura por capas:
- a. Los servicios que ofrece una capa no dependen de los servicios que ofrecen las capas inferiores
 - b. Los elementos de una capa acceden a los servicios de los elementos de las capas inferiores con una arquitectura basada en filtros y tuberías
 - c. Los elementos de una capa ofrecen servicios a los elementos de las capas superiores
-
20. El principio de inversión de dependencias:
- a. Permite reducir el acoplamiento de clases de alto nivel de abstracción respecto a clases más concretas
 - b. Permite cambiar el sentido de una relación de herencia si el acoplamiento entre ambas clases es alto
 - c. Permite cambiar el sentido de una asociación de una clase a otra si el

acoplamiento entre ambas es alto

21. El principio de segregación de interfaces permite:
- Que una clase dependa de menos clases externas
 - Evitar que una clase dependa de métodos que no usa
 - Aumentar el número de métodos que implementa una clase
-
22. El patrón Modelo-Vista-Controlador es un caso particular de:
- Arquitectura basada en Map-Reduce
 - Arquitectura basada en capas
 - Arquitectura basada en filtros y tuberías
-
23. Una forma de extender un módulo que cumple el principio de abierto/cerrado:
- Es añadir una nueva condición a una sentencia if-else encadenada
 - Es añadir una nueva rama a una sentencia switch de un método de una clase
 - Es añadir una nueva clase que implemente un interfaz usado por una clase cliente
-
24. El patrón Estrategia nos permite:
- Evitar tener que cambiar una clase cliente por el hecho de que hayamos extendido el comportamiento de la clase de la que depende
 - Aplicar el principio de refactorización de Liskov
 - Evitar tener que aplicar el principio de segregación de interfaces
-
25. En el patrón Singular:
- El constructor se define público, con un contador de las instancias que ya se han creado
 - El constructor mantiene una lista de las instancias que están activas
 - El constructor se define privado, para que solo pueda instanciar la misma clase
-
26. ¿Cuál es la complejidad ciclomática del siguiente algoritmo?

```
public static int factorial(int n) {  
    int result=-1;           //a  
    int cont =2;             //b  
    if (n >= 0) {             //c  
        result = 1;          //d  
        while (cont <= n) {   //e  
            result = result * cont; //f  
            cont++;           //g  
        }                    //h  
    }                          //i  
    return result;           //j  
}
```

- 8
- 4
- 3
- 5
- 6
- 2
- 1

27. El interruptor de un aire acondicionado está programado para apagarse cuando cae por debajo de 18 (grados) y luego se enciende cuando la temperatura es superior a 21 (grados). Identifica las particiones o clases de equivalencia inválidas. Selecciona una más
- a. Valores entre 18 y 21 ambos incluidos
 - b. Valores entre 18 y 21 ambos no incluidos
 - c. Valores superiores a 21
 - d. Valores inferiores a 18

28. Dado el siguiente código:

```
public final class String2Double{  
    public static boolean esNuloOVacio(String st) {  
        return (st == null || st.compareTo("v") == 0);  
    }  
}
```

Y la siguiente prueba en Junit5:

```
@Test  
public void testesNuloOVacio() {  
    String input = null;  
    assertTrue(String2Double.esNuloOVacio(input));  
  
    input = "abc";  
    assertFalse(String2Double.esNuloOVacio(input));  
  
    input = "";  
    assertTrue(String2Double.esNuloOVacio(input));  
}
```

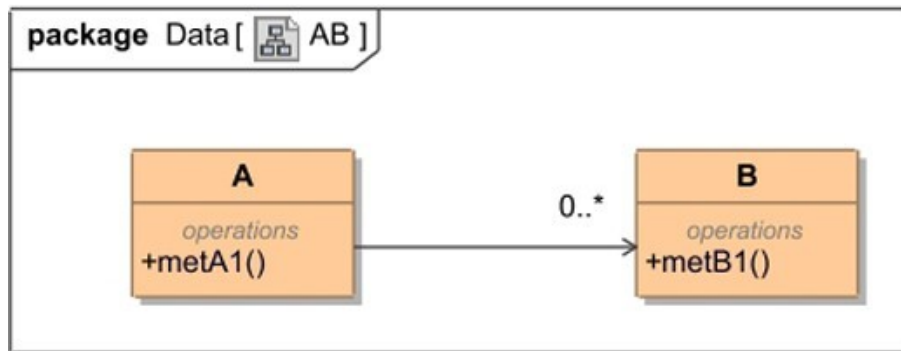
Revisa su funcionamiento y responde a la siguiente pregunta: ¿Cuántos asserts pasan correctamente?

- a. 3
 - b. Pasan todos los asserts correctamente
 - c. 1
 - d. 2
29. Si un programa pasa todas las pruebas a las que se le somete:
- a. No podemos afirmar que esté libre de errores
 - b. Es una demostración de que el conjunto de pruebas se ha diseñado bien
 - c. Se puede asegurar que todos los requisitos funcionales se han implementado correctamente
30. En una prueba de caja blanca:
- a. El usuario sólo prueba los requisitos funcionales de alto nivel
 - b. El sistema tiene que proporcionar el resultado antes de un límite de tiempo
 - c. Se prueba el sistema sabiendo cuál es su estructura interna y su código

31. Los ingenieros de tres empresas de producción de software tienen formas de trabajar diferentes. ¿Cuál de ellas sigue un enfoque de ingeniería del software?
- a. Los ingenieros no siguen ningún proceso concreto, simplemente consultan a su supervisor por la tarea que tienen asignada
 - b. Los ingenieros ejecutan las tareas que les asignan diariamente los clientes del proyecto
 - c. Los ingenieros adoptan una forma de trabajo sistemática y organizada
-
32. La ingeniería del software cobra especial interés:
- a. Cuando se desarrolla software para un cliente particular
 - b. Cuando se desarrolla software destinado para un mercado general
 - c. Es igualmente necesaria independientemente del destinatario del software
-
33. El cliente del nuevo proyecto nos ha avisado de que no será posible reunirse personalmente con frecuencia con los desarrolladores para detallar las historias de usuario cuando empiecen a implementarlas. En ese caso:
- a. Se puede usar una metodología iterativa usando herramientas alternativas para detallar las historias de usuario, como casos de uso o reuniones telefónicas
 - b. Lo más apropiado es seguir la Programación Extrema, para que puedan ser los desarrolladores quienes definan los casos de prueba unitarios y los de aceptación
 - c. Es más conveniente usar una metodología en cascada, con todo el análisis de requisitos al inicio
-
34. Hemos desarrollado un prototipo rápido para acordar la interfaz gráfica de usuario para algunos de los requisitos de la siguiente iteración al cliente le gusta el aspecto resultante y nos pide incluir el prototipo en el sistema final:
- a. Le aconsejamos al cliente no usar el prototipo porque no cumple los criterios de calidad del proyecto
 - b. Lo incluimos porque el cliente siempre tiene razón
 - c. Lo incluimos, aunque no cumpla los criterios de calidad si el proyecto va con retraso
-
35. Como los flujos de trabajo que se hacen en cada iteración coinciden con los flujos de trabajo de la metodología en cascada, el jefe de proyecto ha decidido hacer dos iteraciones de un año en el nuevo proyecto. Esta decisión:
- a. Es adecuada porque permite definir con más calma las pruebas de integración y de aceptación
 - b. No es adecuada porque aumenta los riesgos porque las iteraciones se aplican a una parte mucho mayor del sistema
 - c. Es adecuada porque reduce el tiempo de comunicación y coordinación con el cliente
-
36. Un cliente que quiere que se le construya un sistema informático para una entidad financiera, ha empleado una fórmula matemática en la definición del requisito funcional del cálculo del riesgo a la hora de conceder un préstamo.
- a. Es una buena práctica usar esa fórmula, ya que de esa forma se tiene una definición más precisa del requisito
 - b. No es una buena idea usar esa fórmula. Las fórmulas se deben reservar para describir requisitos no funcionales detallados
 - c. Es muy posible que el equipo de desarrollo no estará familiarizado con ese tipo de fórmulas, por lo que esa fórmula no se puede tener en cuenta al implementar el requisito

37. La mejor forma de describir el siguiente requisito no funcional es:
- “El sistema debe poder dar una respuesta en menos de un segundo”, porque permite definir pruebas para validarlo
 - “El sistema debe responder rápido”, porque de esa forma lo entiende mejor el cliente
 - “El tiempo de respuesta del sistema puede variar en función del tamaño de la entrada”, porque permite un abanico más amplio de soluciones
-
38. El cliente para el que se va a desarrollar el software ha pedido el aseguramiento de la calidad del software se siga la norma de ISO 9126. Sin embargo, el jefe del equipo de desarrollo mantiene que van a usar la norma IEEE 730-1998 porque es la que ellos usan habitualmente en sus proyectos. ¿Cuál debe ser la norma que se siga?
- Como los requisitos de proyecto no pueden contradecir a los de la organización, se usará la norma ISO 9126
 - Como los requisitos de proyecto son prioritarios, se usará la norma IEEE 730-1998
 - Como ambas normas son similares, se pueden usar tanto una como la otra
-
39. En el análisis de requisitos del nuevo sistema, cuyo desarrollo se va a hacer de manera iterativa, el cliente ha pedido un requisito que es “Manejo de los pagos a proveedores”. ¿Qué debe hacer el analista?
- Pedirle que detalle todos los requisitos de usuario relacionados con ese requisito, para poder hacer una planificación global del desarrollo
 - Proponerle al cliente cuáles serán los requisitos de nivel de usuario relacionados con ese requisito
 - Si el cliente lo ha puesto en la categoría de requisitos de poca importancia, se puede iniciar el desarrollo y detallarlo más tarde, en una iteración posterior
-
40. Si el analista detecta que dos requisitos incoherentes entre sí:
- Se implementará el requisito que pase antes las pruebas
 - Debe discutir con el cliente cuál será el requisito que se elimine
 - Debe escoger el requisito que sea más rápido de implementar
-
41. En un sitio web se permiten varios tipos de autenticación: por DNI electrónico, por usuario/contraseña y por referencia enviada al móvil.
- El caso de uso “Autenticación” tendrá una relación “extends” con los otros tres casos de uso para asegurar que el usuario puede escoger su opción preferida
 - Es necesario definir tres tipos de usuario, uno por cada tipo de autenticación, para evitar de ese modo accesos no permitidos
 - Si el caso de uso “Autenticación” no tiene muchas acciones por sí mismo, los casos de uso de cada tipo de autenticación deben especializar el caso de uso “Autenticación”

42. En el diagrama de clases de la figura:



- En los objetos de la clase A habrá una colección de referencias a objetos de la clase B
- Hay que introducir una clase intermedia para normalizar el diagrama de clases
- Los objetos de la clase B se encargan de crear los objetos de la clase A

43. La figura "Diagrama de casos de uso de web de recetas" representa una parte de la funcionalidad para una web de recetas de cocina. Esta web tiene la particularidad de que incluye programas descargables para diferentes máquinas (robots de cocina), pero sólo los usuarios identificados pueden descargar los programas asociados a las recetas.

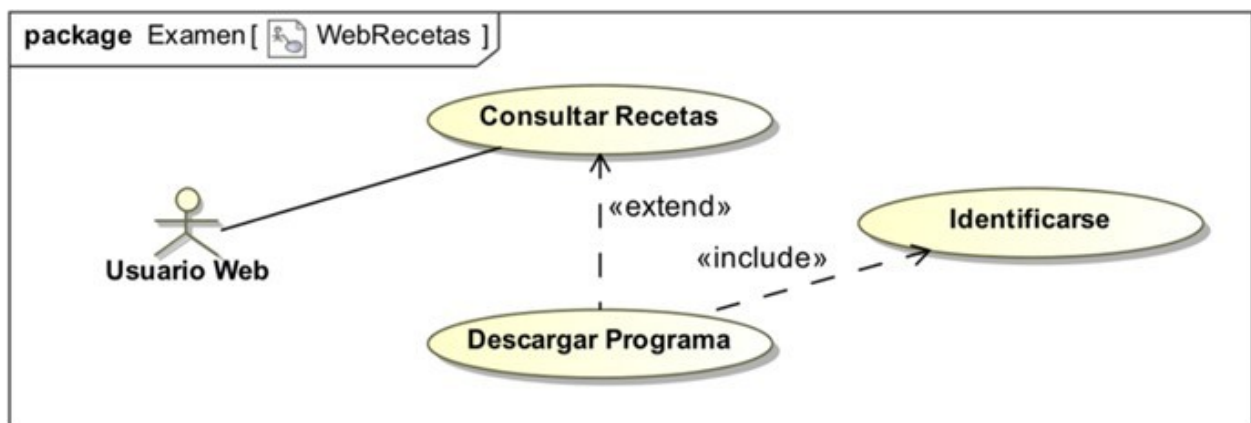
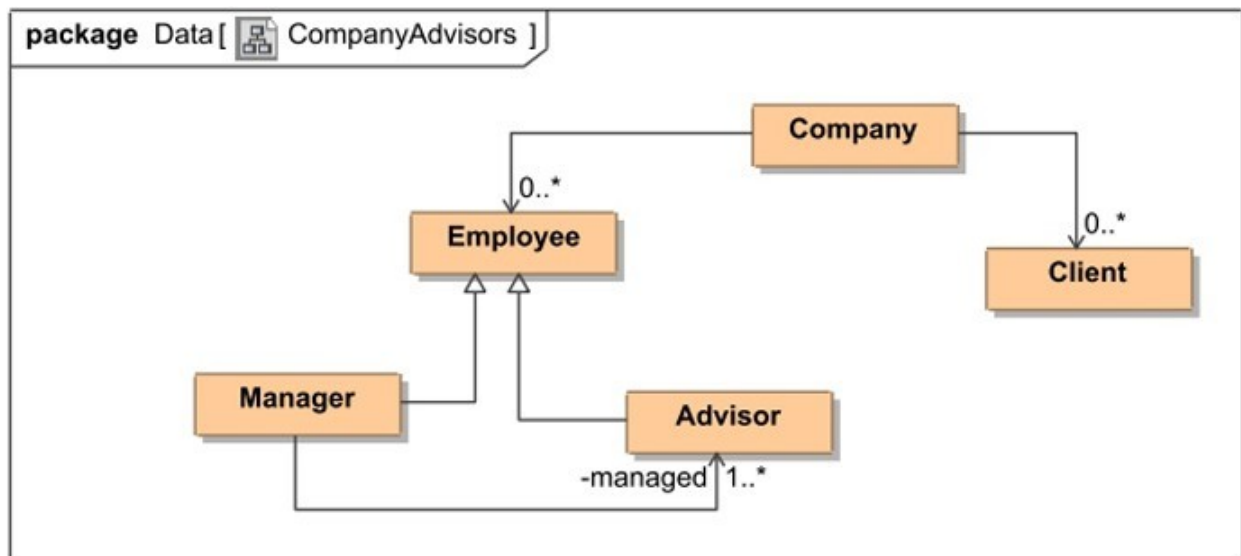


Figura "Diagrama de casos de uso de web de recetas"

Indique la respuesta correcta para el diagrama anterior:

- Para "Consultar Recetas" siempre hay que "Identificarse" porque ese caso de uso está relacionado mediante una relación de "Include"
- Para "Descargar Programa" hay que "Consultar Recetas" porque ese caso de uso está relacionado mediante una relación de "Extend"
- Para "Descargar Programa" hay que hacerlo mediante "Consultar Recetas" porque no hay asociación de uso directa entre el "Usuario Web" y "Descargar Programa"

44. En el diagrama de clases de la figura:



- a. Solo es posible saber los clientes de los managers
- b. No es posible saber directamente cuáles son los clientes de un asesor en una empresa
- c. Para saber cuáles son los clientes de un asesor basta con saber la empresa en la que trabaja

45. Un sistema para compartir archivos como Dropbox:

- a. Tiene una arquitectura cliente/servidor
- b. Tiene una arquitectura basada en capas
- c. Tiene una arquitectura basada en filtros y tuberías

46. Si pensamos en la búsqueda de información en la web en un buscador como Google como un problema MapReduce:

- a. La operación reduce consiste en sumar el número de veces que aparecen los términos buscados en las páginas indexadas por el buscador
- b. La operación map consiste en buscar la ocurrencia de los términos buscados en las páginas indexadas por el buscador
- c. La operación map consiste en ordenar las páginas indexadas en las que han aparecido los términos de búsqueda en base a alguna función de pertinencia

47. Un cuadro de texto acepta valores numéricos en el rango de 18 a 25 ambos incluidos. Identifique cuáles de los siguientes valores pertenecen a la misma clase de partición de equivalencia. Seleccione una o más:

- a. {17, 15, -5}
- b. {18, 19, 23}
- c. {17, 18, 19}
- d. {15, 14, 25}

48. ¿Cuál de las siguientes pruebas de caja blanca sirve para verificar el camino base más

```
public static boolean esPrimo(int num) {  
    int cnt = 0;           //a  
    int div = 1;           //b  
    boolean result = false; //c  
    while (div <= num){    //d  
        if((num % div) == 0) { //e  
            ++cnt;         //f  
        }                 //g  
        ++div;            //h  
    }                     //i  
    if(cnt == 2){         //j  
        result = true;    //k  
    }                     //l  
    return result;        //m  
}
```

largo?

- a. Entrada: num = 0
- b. Entrada: num = 3
- c. Entrada: num = 4
- d. Entrada: num = 1

49. Dado el siguiente código:

```
public final class String2Double{  
    public static Double toDouble(String str){  
        if(str==null){  
            return null;  
        }  
        return Double.valueOf(str);  
    }  
}
```

Y la siguiente prueba en Junit5:

```
@Test  
public void testConvertToDoubleOK() {  
    String age = "1990";  
    Double expAge = Double.valueOf(age);  
    Double actual = String2Double.toDouble(age);  
    assertNotNull(actual, "The actual is null");  
    assertEquals(expAge, actual);  
}
```

Revisa su funcionamiento y responde a la siguiente pregunta: ¿Pasa el código descrito esta prueba?

- a. Si, pasan todos los asserts correctamente
- b. No, el primer assert falla
- c. No, fallan todos los asserts
- d. No, el segundo assert falla

50. Dado el siguiente código:

```
public final class String2Double{
    public static String devolverPorDefectoSiNULL(final String st, final String
defaultSt) {
        String resultado = defaultSt;
        if (st!=null) resultado = st;
        return resultado;
    }
}
```

Y la siguiente prueba en JUnit5:

```
@Test
public void testDevolverPorDefectoSiNULL() {
    String st = null;
    String defaultSt = "pordefecto";

    String actual = String2Double.devolverPorDefectoSiNULL(st, defaultSt);
    assertEquals(defaultSt, actual);

    st = "def";
    actual = String2Double.devolverPorDefectoSiNULL(st, defaultSt);
    assertNotSame(defaultSt, actual);

    st = "";
    actual = String2Double.devolverPorDefectoSiNULL(st, defaultSt);
    assertNotSame(defaultSt, actual);
}
```

Revisa su funcionamiento y responde a la siguiente pregunta: ¿Pasa el código descrito esta prueba?

- a. No, falla el tercer assert
- b. No, el segundo assert falla
- c. Si, pasan todos los asserts correctamente
- d. No, el primer assert falla

51. Respecto a los desarrollos industriales, en los desarrollos software:

- a. El porcentaje de trabajo que se emplea en hacer pruebas es mayor porque el software es etéreo y no se puede probar fácilmente
- b. El coste unitario de cada copia del software es mucho menor porque es más fácil hacer copias
- c. Los cambios introducidos en un producto existente están libres de errores

52. ¿Por qué es necesaria la ingeniería del software?

53. En el desarrollo dirigido por pruebas, por norma general:

- a. Dependiendo del requisito a implementar, las pruebas se codifican antes o después que el código
- b. Las pruebas se codifican después que el código
- c. Las pruebas se codifican antes que el código

54. En una prueba de caja negra solo:

- a. Se ejecutan pruebas unitarias que se puedan automatizar
- b. Se aprueban los módulos que interaccionan directamente con el usuario
- c. Se ejecutan pruebas sin conocer el diseño ni la implementación

55. ¿Cuál es la complejidad ciclomática del siguiente algoritmo?

```
public static int maxArray(int arr[]) {  
    int resultado = Integer.MIN_VALUE; //a  
    if (arr.length > 0) { //b  
        resultado = arr[0]; //c  
        int i = 1; //d  
        while (i < arr.length) { //e  
            if (arr[i] > resultado) { //f  
                resultado = arr[i]; //g  
            } //h  
            i++; //i  
        } //j  
    } //k  
    return resultado; //l  
}
```

- a. 8
- b. 1
- c. 3
- d. 5
- e. 2
- f. 6
- g. 4

56. La cohesión de un sistema:

- a. Es aconsejable mantenerla baja para que se cumplan los criterios de calidad del sistema
- b. Es difícil mantener alta porque los elementos de un módulo no deben interaccionar entre sí
- c. Es conveniente mantenerla alta para que los módulos tengan una función específica

57. ¿Cuál es la complejidad ciclomática del siguiente algoritmo?

```
public static int maxElevado(int n1, int n2, int e) {  
    int resultado=1;    //a  
    int cont=0;         //b  
    int max =n2;        //c  
  
    if (n1 > n2){        //d  
        max = n1;        //e  
    }                   //f  
  
    while (cont < e){    //g  
        resultado *=max; //h  
        cont++;         //i  
    }                   //j  
    return resultado;   //k  
}
```

- a. 3
- b. 2
- c. 6
- d. 4
- e. 8
- f. 5
- g. 1

58. Un software de reconocimiento facial usa una cámara RGB-D. Cuando la persona está a menos de 1.8 metros lanza una excepción notificando que la persona está muy cerca, cuando una persona está a una distancia superior a 2.4 metros, lanza una excepción notificando que la persona está muy lejos. Entre estos valores, la detección facial se realiza correctamente. Identifique cuales de los siguientes valores pertenecen a la misma clase de partición de equivalencia. Seleccione una o más:

- a. {1.79, 1.92, 2.05}
- b. {1.81, 2.39, 2.4}
- c. {-1.81, 2.05, 2.41}
- d. {1.80, 2.05, 2.39}

59. Un módulo con un acoplamiento alto con otros muchos módulos:

- a. Posibilita que ese módulo sea más versátil
- b. Permite implementar el sistema con un número menor de módulos
- c. Hace más difícil que se pueda reutilizar ese módulo en otro entorno

60. Para que una clase B que especializa a otra clase A cumpla el principio de sustitución de Liskov:

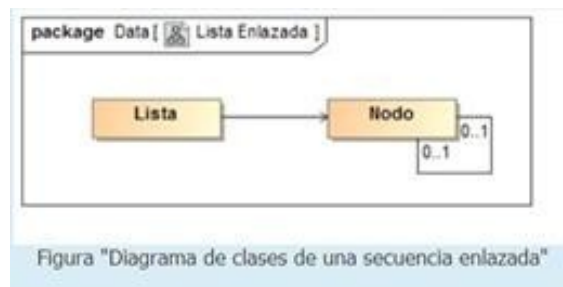
- a. La clase B no puede usar los métodos que se han definido en la clase A
- b. La clase B no puede cambiar la signatura y comportamientos de los métodos de la clase A
- c. La clase B no puede añadir nuevos métodos respecto a la clase A

61. En la arquitectura basada en llamada y retorno funcional:

- a. El código de las funciones son los conectores del sistema
- b. Las funciones deben conectarse mediante variables globales
- c. Las llamadas a funciones son los conectores del sistema

62. En una arquitectura basada en objetos, los objetos se comunican enviándose mensajes. La implementación más habitual de estos envíos de mensajes es:
- Definición de interfaces
 - Llamadas a métodos
 - Elevación de excepciones

63. La figura "Diagrama de clases de una secuencia enlazada" muestra un diagrama de clases que representa una lista enlazada.



El diagrama es:

- Incorrecto porque en UML no hay relaciones recursivas
 - Correcto
 - Incorrecto porque la relación recursiva de la clase Nodo no nombra los roles
 - Incorrecto porque la relación recursiva de Nodo no tiene caso base
64. El cliente ha pedido como nuevo requisito no funcional de seguridad que se respete la privacidad de los datos que se almacenen de los pacientes de la clínica. Como consecuencia:
- Se añaden nuevos requisitos funcionales, como el inicio de sesión autenticado, y se adaptan otros, como la anonimización de los datos a la hora de generar informes estadísticos
 - Se añade una nueva iteración al final del desarrollo dedicada en exclusiva a implementar ese requisito no funcional
 - El analista renegocia los requisitos funcionales con el cliente para eliminar otros requisitos funcionales que tengan una complejidad similar
65. ¿Cuál de las siguientes pruebas de caja blanca sirve para verificar el camino base más largo?

```
public static int maxArray(int arr[]) {  
    int resultado = Integer.MIN_VALUE; //a  
    if (arr.length>0){ //b  
        resultado = arr[0]; //c  
        int i = 1; //d  
        while (i<arr.length){ //e  
            if (arr[i]>resultado){ //f  
                resultado = arr[i]; //g  
            } //h  
            i++; //i  
        } //j  
    } //k  
    return resultado; //l  
}
```

- Entrada: arr = {3, 2}
- Entrada: arr = {}
- Entrada: arr = {2, 2}
- Entrada: arr = {1, 2}

66. Un ejemplo de requisito no funcional es:

- a. El estilo de las páginas web seguirá las guías de estilo de la empresa
- b. Si el usuario publica un comentario ofensivo, se bloqueará su cuenta
- c. Hay tres tipos de envíos: normal, urgente y exprés

67. En los diagramas de clases de UML, para indicar en un extremo de una asociación que la relación es opcional se usa:

- a. 1...?
- b. 1...*
- c. 0...1

68. Un diagrama de secuencia en UML:

- a. Modela el comportamiento de una determinada clase
- b. Modela la interacción entre componentes del sistema
- c. Modela la estructura estática del sistema

69. ¿Cuál es la complejidad ciclomática del siguiente algoritmo?

```
public static int minElevado(int n1, int n2, int e) {  
    int resultado=1; //a  
    int cont=0; //b  
    int min = n1; //c  
  
    if (n1 > n2){ //d  
        min = n2; //e  
    } //f  
  
    while (cont < e){ //g  
        resultado *= min; //h  
        cont++; //i  
    } //j  
    return resultado; //k  
}
```

- a. 2
- b. 5
- c. 8
- d. 6
- e. 4
- f. 1
- g. 3

70. El principal objetivo de los ingenieros de software es producir software de calidad. Esto quiere decir que:

- a. Tienen producir un software que además de cumplir con las funciones del cliente tenga otras características tales como: mantenibilidad, fiabilidad, eficiencia y adaptación
- b. Tienen que procurar producir un software lo más completo posible para minimizar posibles actualizaciones futuras
- c. Los ingenieros determinarán cuáles son los criterios de calidad que tiene que cumplir el sistema requerido por el cliente

71. El uso del patrón Método Fábrica (Factory Method):
- Permite a una clase cliente crear instancias de una clase implementada con el patrón Singular
 - Permite a una clase cliente construir instancias de un interfaz o de una clase abstracta
 - Permite a una clase cliente no estar acoplada con una clase concreta a la hora de crear nuevas instancias
-
72. En el análisis de requisitos de un sistema, el analista ha detectado que un cliente ha pedido “que el inicio de sesión en el sistema se haga exclusivamente mediante certificado digital”, mientras que otro cliente ha pedido “que el inicio de sesión en el sistema se haga exclusivamente con el sistema Active Directory Domain que ya tiene implementado la empresa”. Para resolver el conflicto, el analista debe:
- Plantear el conflicto a los clientes y proporcionarles la información necesaria para que ellos decidan
 - Incluir ambos requisitos funcionales en el sistema
 - Elegir la opción “Active Directory Domain” porque ya está implementada
-
73. Cuando un caso de uso siempre necesita la realización de otro caso de uso, la relación que existe entre ellos es de:
- Extensión (extend)
 - Generalización
 - Inclusión (include)
-
74. En el segundo día de la iteración actual, uno de los tres desarrolladores del grupo ha caído enfermo, y tardará un mes en volver. ¿Qué decisión debe tomar el jefe de proyecto?
- Retrasar la iteración hasta que se incorpore el desarrollador enfermo
 - Informar al cliente y negociar una reducción del alcance de la iteración
 - Incorporar dos nuevos desarrolladores al equipo para suplir al miembro enfermo
-
75. En un proceso de producción de software:
- Si el programa resultante no contiene errores, no es necesaria ninguna documentación
 - Son importantes tanto el programa generado como la documentación correspondiente al mismo
 - Los ingenieros de software tienden a escribir grandes cantidades de documentación que no sirve para nada
-
76. Dividir un sistema en módulos:
- Es aconsejable solo para arquitecturas Modelo-Vista-Controlador
 - Permite definir el espacio de diseño del sistema
 - Facilita la resolución del problema
-
77. En un diagrama de clases de UML:
- Hay que indicar la información adecuada según el nivel de abstracción
 - Hay que dar siempre el máximo detalle posible
 - Hay que indicar siempre los principales campos y métodos

78. Dado el siguiente código:

```
public final class String2Double{
    public static boolean esNuloOVacio(String st) {
        return st == null || st.trim().length() == 0;
    }
}
```

Y la siguiente prueba de JUnit5:

```
@Test
public void testIsNullOrBlank() {
    String input = null;
    assertTrue(String2Double.esNuloOVacio(input));

    input = " ";
    assertTrue(String2Double.esNuloOVacio(input));

    input = "abc";
    assertFalse(String2Double.esNuloOVacio(input));
}
```

Revisa su funcionamiento y responde a la siguiente pregunta: ¿Pasa el código descrito esta prueba?

- a. No, el segundo assert falla
- b. No, el tercer assert falla
- c. Si, pasan todos los asserts correctamente
- d. No, el primer assert falla

79. En los diagramas de clases de UML, una asociación de composición se representa con:

- a. Un rombo blanco
- b. Un rombo negro
- c. Un triángulo

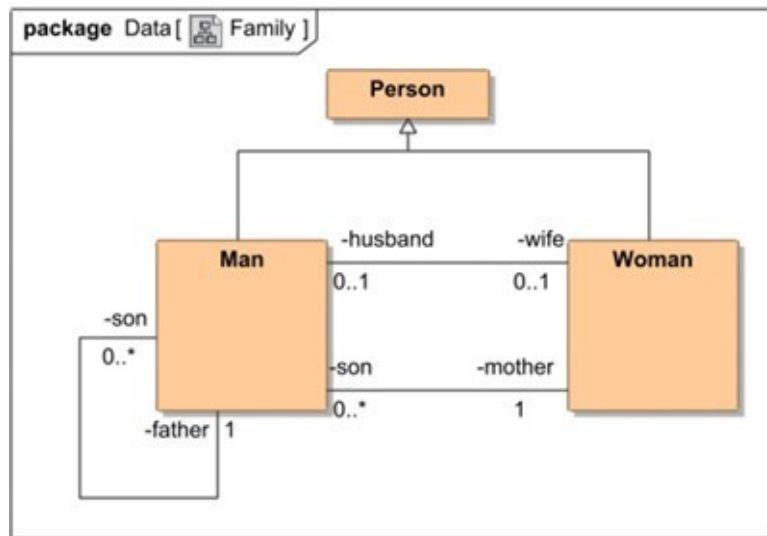
80. ¿Cuál de los siguientes analistas lo ha hecho mejor?

- a. El analista C ha hecho una primera descripción de los requisitos funcionales con lenguaje natural y una descripción posterior con lenguaje natural estructurado
- b. El analista B ha descrito tanto los objetivos globales del sistema como los requisitos funcionales detallados a nivel de usuario con lenguaje natural para que el usuario los entienda mejor
- c. El analista A ha descrito los objetivos globales del sistema en lenguaje natural estructurado y los requisitos funcionales detallados a nivel de usuario los ha descrito como lenguaje natural

81. El patrón Estrategia se basa en:

- a. Usar la vinculación dinámica para escoger la versión adecuada de un algoritmo en tiempo de ejecución
- b. Dividir la responsabilidad del método en varias clases usando la estrategia de "divide y vencerás"
- c. Probar varias estrategias hasta conseguir la respuesta que necesita la clase cliente

82. En el diagrama de clases de la figura:



- a. Las personas solo pueden tener hijos varones
- b. Solo las personas casadas pueden tener hijos
- c. El diagrama es incorrecto porque no puede haber dos asociaciones entre dos clases

83. Un cuadro de texto acepta los valores numéricos de rango de 18 a 25 (18 y 25 también son parte de la clase). Identifica las particiones o clases de equivalencia válidas.

- a. Valores entre 18 y 25
- b. Valores mayores que 25
- c. Valores menores que 18

84. En los diagramas de clases de UML, una asociación de agregación se representa con:

- a. Un triángulo
- b. Un rombo negro
- c. Un rombo blanco

85. Uno de los inconvenientes de la arquitectura de tuberías y filtros es:

- a. Que es difícil conectar un filtro con el siguiente mediante tuberías
- b. Que los filtros solo pueden procesar archivos de texto
- c. Que es difícil tratar una excepción que ocurra en mitad de la cadena de filtros

86. MagicDraw es una potente herramienta que da soporte para el modelado de sistemas software:

especificación de requisitos, diseño del sistema... Como tal, podemos decir que:

- a. Debido a la gran cantidad de prestaciones que ofrece, no encaja en ninguna de las dos categorías
- b. Es una herramienta de tipo Lower-CASE
- c. Es una herramienta de tipo Upper-CASE

87. En el pliego de contratación de la nueva web del Senado de España se incluye el requisito "Dada la naturaleza de este sistema de información, es requisito indispensable que la solución trabaje en alta disponibilidad".
- Es un buen requisito, permite a los desarrolladores escoger la solución que crean más conveniente
 - Es un mal requisito, es ambiguo, por lo que la solución dada puede no gustar al cliente
 - Es un requisito que se definirá de forma más detallada creando nuevos requisitos funcionales

-
88. Según el principio de segregación de interfaces:
- Las clases deberían estar abiertas a la extensión y cerradas a la modificación
 - Los clientes no deben depender de métodos que no utilizan
 - Una clase debería tener un solo motivo para cambiar

-
89. Dado el siguiente código:

```
public final class String2Double{
    public static String devolverPorDefectoSiNULL(final String st, final String
defaultSt) {
        String resultado = defaultSt;
        if ((st!=null) || (st.compareTo("")==0)) resultado = st;
        return resultado;
    }
}
```

Y la siguiente prueba Junit5:

```
@Test
public void testDevolverPorDefectoSiNULL() {
    String st = null;
    String defaultSt = "pordefecto";

    String actual = String2Double.devolverPorDefectoSiNULL(st, defaultSt);
    assertEquals(defaultSt, actual);

    st = "def";
    actual = String2Double.devolverPorDefectoSiNULL(st, defaultSt);
    assertEquals(defaultSt, actual);

    st = "";
    actual = String2Double.devolverPorDefectoSiNULL(st, defaultSt);
    assertEquals(defaultSt, actual);
}
```

Revisa su funcionamiento y responde a la siguiente pregunta: ¿Pasa el código descrito esta prueba?

- No, el tercer assert falla
- No, el primer assert falla
- Si, pasan todos los asserts correctamente
- No, el segundo assert falla

-
90. Un ejemplo típico de requisito del dominio puede ser:
- Permitir que un usuario autenticado acceda a sus mensajes
 - Seguir las normas contables nacionales en un programa de contabilidad
 - Construir un modelo del dominio con clases que entienda el cliente

91. ¿Cuál de las siguientes pruebas de caja blanca sirve para verificar el camino base más largo?

```
public static int minElevado(int n1, int n2, int e) {  
    int resultado=1;    //a  
    int cont=0;         //b  
    int min = n1;       //c  
  
    if (n1 > n2){        //d  
        min = n2;       //e  
    }                   //f  
  
    while (cont < e){    //g  
        resultado *=min; //h  
        cont++;         //i  
    }                   //j  
    return resultado;   //k  
}
```

- a. Entrada: n1 = 4, n2 = 3, e = 0
- b. Entrada: n1 = 4, n2 = 3, e = 2
- c. Entrada: n1 = 2, n2 = 2, e = 1
- d. Entrada: n1 = 1, n2 = 3, e = 2

92. La solución que aporta un método ágil para la impredecibilidad que afecta el desarrollo de un proyecto software es:

- a. Adaptabilidad
- b. Usar herramientas de modelado gráficas (UML)
- c. Aplicación de métodos formales

93. Scrum es una metodología ágil:

- a. Que se usa marginalmente
- b. De uso frecuente en la práctica
- c. Es un modelo de proceso software de tipo teórico

94. ¿Qué actividad no forma parte de la gestión de proyectos?

- a. Selección y evaluación del personal
- b. Planificación y programación temporal
- c. Modelado del sistema a desarrollar

95. Si la tecnología en la que se basa un proyecto es superada por otra nueva, el riesgo de que esto ocurra afecta al:

- a. Producto
- b. Negocio
- c. Proyecto

96. La interfaz debe seguir la normativa de colores e imagen corporativa de la empresa.

- a. Requisito funcional
- b. Requisito no funcional del producto
- c. Requisito no funcional de la organización
- d. Requisito no funcional externo

97. Durante un proyecto de desarrollo del software se detectaron dos errores de los requisitos. Uno se detectó durante la fase de requisitos y el otro durante la fase de implementación. ¿Cuál de las siguientes afirmaciones es probablemente más cierta?
- El error más costoso de corregir es el detectado durante la fase de implementación
 - No hay relación entre la fase en la que se detecta un error y su coste de reparación
 - El error más costoso de corregir es el detectado durante la fase de requisitos
 - El coste de arreglar cualquiera de los errores será aproximadamente similar
-
98. De los siguientes problemas indique cuáles se producen durante la obtención de requisitos al tratar con los participantes (stakeholders). Seleccione una o más de una:
- Hay requisitos que entran en conflicto, según el grupo de la organización que lo solicite o defina
 - Durante el propio proceso de análisis, hay requisitos que cambian porque cambia el entorno del negocio
 - Expresan los requisitos en su jerga del dominio
 - Conocen realmente lo que desean
-
99. Durante la obtención de los requisitos para un sistema, los clientes plantean que los usuarios tengan identificador y clave de acceso. Suponen, porque es lo habitual, que las claves se almacenen encriptadas en el sistema, pero nunca lo expresan ni comunican a los analistas de la empresa que desarrollará el software. El requisito de almacenar las claves encriptadas es un...
- Requisito externo
 - Requisito implícito
 - Requisito de la organización
-
100. Para añadir un detalle a un caso de uso, se debe utilizar:
- Un diagrama de estados
 - Un diagrama de secuencia
 - Un diagrama de clases
-
101. La ingeniería de requisitos trata de:
- Entender el problema
 - Plantear una primera solución al problema (prototipo)
 - Plantear un plan para resolver el problema
-
102. El tiempo de respuesta es un requisito de tipo:
- No funcional
 - Del dominio
 - Funcional
-
103. ¿En qué actividad del proceso de desarrollo software se establecen las restricciones en el funcionamiento del sistema?
- Evolución del software
 - Validación del software
 - Diseño e implementación del software
 - Especificación software

104. ¿Qué características son aplicables al modelo en cascada? Seleccione una o más de una:
- a. El cliente ve rápidamente un resultado aproximado de su producto
 - b. Los proyectos rara vez siguen un modelo secuencial
 - c. Es el más antiguo
 - d. El cliente debe exponer sus requisitos al principio del proceso
-
105. De los siguientes modelos, ¿cuál no es Evolutivo?
- a. Modelo en cascada
 - b. Modelo espiral
 - c. Scrum
-
106. A diferencia de la Ingeniería de Caminos, en la Ingeniería del Software:
- a. Los equipos de trabajo están compuestos de gente con distinto perfil
 - b. El producto resultante no se desgasta con el uso
 - c. El tamaño importa
-
107. Cuando se habla de fiabilidad del software se hace referencia a:
- a. Los recursos que utiliza el software no son malgastados
 - b. Es aceptado por los usuarios finales, no los desarrolladores
 - c. Puede evolucionar hacia nuevas necesidades
 - d. Está libre de errores
-
108. Un diagrama de estados muestra:
- a. Las interacciones entre el sistema y su entorno
 - b. Cómo parte del sistema reacciona ante eventos internos o externos
 - c. Las interacciones entre los actores y el sistema
-
109. Para especificar los escenarios de un caso de uso en UML podemos usar. Seleccione uno o más:
- a. Un diagrama de secuencia
 - b. Un diagrama de estados
 - c. Un diagrama de casos de uso
 - d. Un diagrama de actividad
-
110. Los diagramas de actividad:
- a. Muestran las interacciones entre el sistema y su entorno
 - b. Muestran interacciones entre los actores, el sistema y los componentes
 - c. Muestran cómo el sistema reacciona ante eventos internos o externos
 - d. Muestran las actividades implicadas en un proceso o en un procesamiento de datos
-
111. Los requisitos deben ser consistentes, lo que implica que:
- a. La descripción de todos los servicios y funciones deben estar definidos
 - b. No deben existir conflictos o contradicciones en la descripción de las funciones del sistema
 - c. Todos los componentes del sistema software deben estar bien especificados

112. De las siguientes actividades indique cuáles se producen durante el proceso de especificación del software (Ingeniería de requisitos). Seleccione una o más de una:

- a. Pruebas de aceptación
- b. Diseño arquitectónico
- c. Mantenimiento del Software
- d. Validación de requisitos
- e. Elicitación de requisitos
- f. Estudio de viabilidad
- g. Programación y depuración del código
- h. Diseño de interfaz

113. ¿Cuáles de las siguientes pueden ser relaciones entre las clases de un diagrama de clases? Seleccione una o más de una.

- a. Asociación y enlace
- b. Agregación y Generalización
- c. Asociación y Agregación

114. Las cuatro claves de la programación extrema (XP) son:

- a. Comunicación, simplicidad, retroalimentación y coraje
- b. Comunicación, simplicidad, trabajo en equipo y rigor
- c. Comunicación, flexibilidad, coraje y sincronización

115. Normalmente esto es lo que describe el cliente cuando habla sobre el sistema.

- a. Camino Principal
- b. Requisito
- c. Caso de uso

116. Algo que el sistema debe realizar para funcionar correctamente:

- a. Camino principal
- b. Requisito
- c. Caso de uso

117. Ayuda a obtener buenos requisitos. Cuenta una historia sobre cómo funciona el sistema:

- a. Camino principal
- b. Requisito
- c. Caso de uso

118. Los clientes no deben aprender de métodos que no utilizan.

- a. Principio de responsabilidad única
- b. Principio de segregación de interfaces
- c. Principio de sustitución de Liskov
- d. Principio abierto-cerrado
- e. Principio de inversión de dependencias

119. Las clases deben estar abiertas a la extensión y cerradas a la modificación.
- a. Principio de responsabilidad única
 - b. Principio de segregación de interfaces
 - c. Principio de sustitución de Liskov
 - d. Principio abierto-cerrado
 - e. Principio de inversión de dependencias
-
120. Depende de abstracciones; no dependas de implementaciones.
- a. Principio de responsabilidad única
 - b. Principio de segregación de interfaces
 - c. Principio de sustitución de Liskov
 - d. Principio abierto-cerrado
 - e. Principio de inversión de dependencias
-
121. Las subclasses deben poder sustituir a las clases sin que el código cliente lo note.
- a. Principio de responsabilidad única
 - b. Principio de segregación de interfaces
 - c. Principio de sustitución de Liskov
 - d. Principio abierto-cerrado
 - e. Principio de inversión de dependencias
-
122. Una clase debería tener un solo motivo para cambiar:
- a. Principio de responsabilidad única
 - b. Principio de segregación de interfaces
 - c. Principio de sustitución de Liskov
 - d. Principio abierto-cerrado
 - e. Principio de inversión de dependencias
-
123. @AfterClass
- ```
public static void method()
```
- a. Ejecuta method() antes de cada test de la clase
  - b. Ejecuta method() después de ejecutar todos los test de la clase
  - c. Ejecuta method() después de cada test de la clase
  - d. Ejecuta method() antes de ejecutar todos los test de la clase
- 
124. @Before
- ```
public void method()
```
- a. Ejecuta method() antes de cada test de la clase
 - b. Ejecuta method() después de ejecutar todos los test de la clase
 - c. Ejecuta method() después de cada test de la clase
 - d. Ejecuta method() antes de ejecutar todos los test de la clase

125. @BeforeClass

public static void method()

- a. Ejecuta method() antes de cada test de la clase
 - b. Ejecuta method() después de ejecutar todos los test de la clase
 - c. Ejecuta method() después de cada test de la clase
 - d. Ejecuta method() antes de ejecutar todos los test de la clase
-

126. @After

public void method()

- a. Ejecuta method() antes de cada test de la clase
 - b. Ejecuta method() después de ejecutar todos los test de la clase
 - c. Ejecuta method() después de cada test de la clase
 - d. Ejecuta method() antes de ejecutar todos los test de la clase
-

127. Un diseño de alta calidad para un componente software (módulo, paquete, subsistema), ¿cuáles de las siguientes características deberá tener? Selecciona una o más.

- a. El conjunto de tareas dentro del componente está lógicamente relacionado
 - b. El acoplamiento del componente con otros componentes es bajo
 - c. El componente es fácil de probar
-

128. Si queremos hacer un test en el que intervenga una interfaz hemos de...

- a. Usar mocking
 - b. Usar JUnit
 - c. Crear una clase que implemente la interfaz
-

129. La idea básica de las pruebas de caja blanca es...

- a. Asegurar que todas las sentencias y condiciones han sido ejecutadas al menos una vez
 - b. Asegurar que no existen bucles sin fin ni interbloqueos en el caso de programas concurrentes
 - c. Comprobar que al menos el 90% de las sentencias y condiciones han sido ejecutadas al menos una vez
-

130. Los componentes de una arquitectura software son:

- a. Componentes, patrones y conectores
 - b. Cliente, servidor y capas
 - c. Componentes, conectores y configuración
-

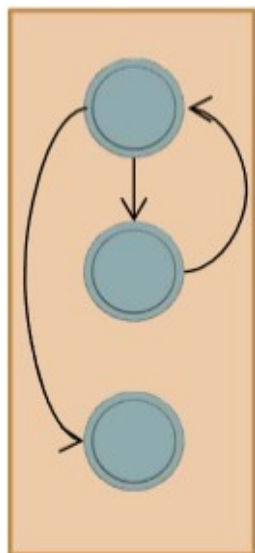
131. Tres ejemplos de arquitectura software son:

- a. Capas, orientado a objetos, maestro/esclavo
- b. Cliente/servidor, MapReduce, capas
- c. Cliente/servidor, Singleton, orientado a objetos

132. En arquitectura de tuberías, la información se procesa según un esquema de:
- Flujo de tareas
 - Flujo de control
 - Flujo de datos
-
133. El esquema de MapReduce se usa para:
- Procesamiento masivo de datos
 - Procesamiento en la nube
 - Procesamiento masivo de procesos
-
134. Los patrones de diseño se pueden clasificar en tres tipos, que son:
- Creación, estructurales y de comportamiento
 - Creación, arquitectura y de comportamiento
 - Estructurales, creación y de pruebas
-
135. ¿Qué definición se ajusta mejor a patrón de diseño?
- Son bibliotecas de clases probadas y sin errores aplicables a un determinado tipo de problemas
 - Una solución probada que se puede aplicar a un determinado tipo de problemas que se repiten en el desarrollo software
 - Son los puntos de acceso a los módulos o sistemas de comunicación
 - Es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el contexto, y los principios que orientan su diseño y evolución
-
136. La siguiente lista es una serie de Estilos arquitectónicos y Patrones de diseño. Indique cuáles son Patrones de Diseño.
- Tubería filtro
 - Peer to peer
 - Por capas
 - Delegación
 - Cliente-servidor
 - Fachada
 - Factoría abstracta
 - Singleton
-
137. El diseño por contrato es una técnica orientada a asegurar el principio SOLID...
- Principio de segregación de interfaces
 - Principio de responsabilidad única
 - Principio Abierto-Cerrado
 - Principio de sustitución de Liskov
-
138. ¿Cuáles de las siguientes características son indicativas de un buen diseño? Seleccione una o más de una.
- Presenta un fuerte acoplamiento entre sus módulos
 - Incluye casos de prueba para todos los componentes
 - Implementa todos los requisitos del modelo de análisis
 - Se puede adaptar a nuevas situaciones con facilidad

139. Un escenario de un caso de uso:
- Indica cuáles son las acciones en una de las posibles situaciones que se ejecute en un caso de uso
 - Detalla los tipos de datos y las relaciones entre los distintos tipos de datos
 - Especifica cuáles son las partes interesadas y sus necesidades e intereses

140. ¿Cuál es la complejidad ciclomática del siguiente grafo de ejecución?



- 3
- 2
- 1

141. El software:

- Se desgasta por el uso de los clientes porque encuentran errores no detectados en las pruebas anteriores
- Se desgasta por el uso de los clientes, igual que los productos hardware industriales
- No se desgasta por el uso, pero puede dejar de ser útil por cambios en los requisitos de los clientes

142. Una de las ventajas de la arquitectura basada en objetos sobre la arquitectura de llamada y retorno funcional es que:

- Las funciones representan elementos del mundo real
- Los datos están mejor encapsulados
- Los objetos no dependen de otros objetos

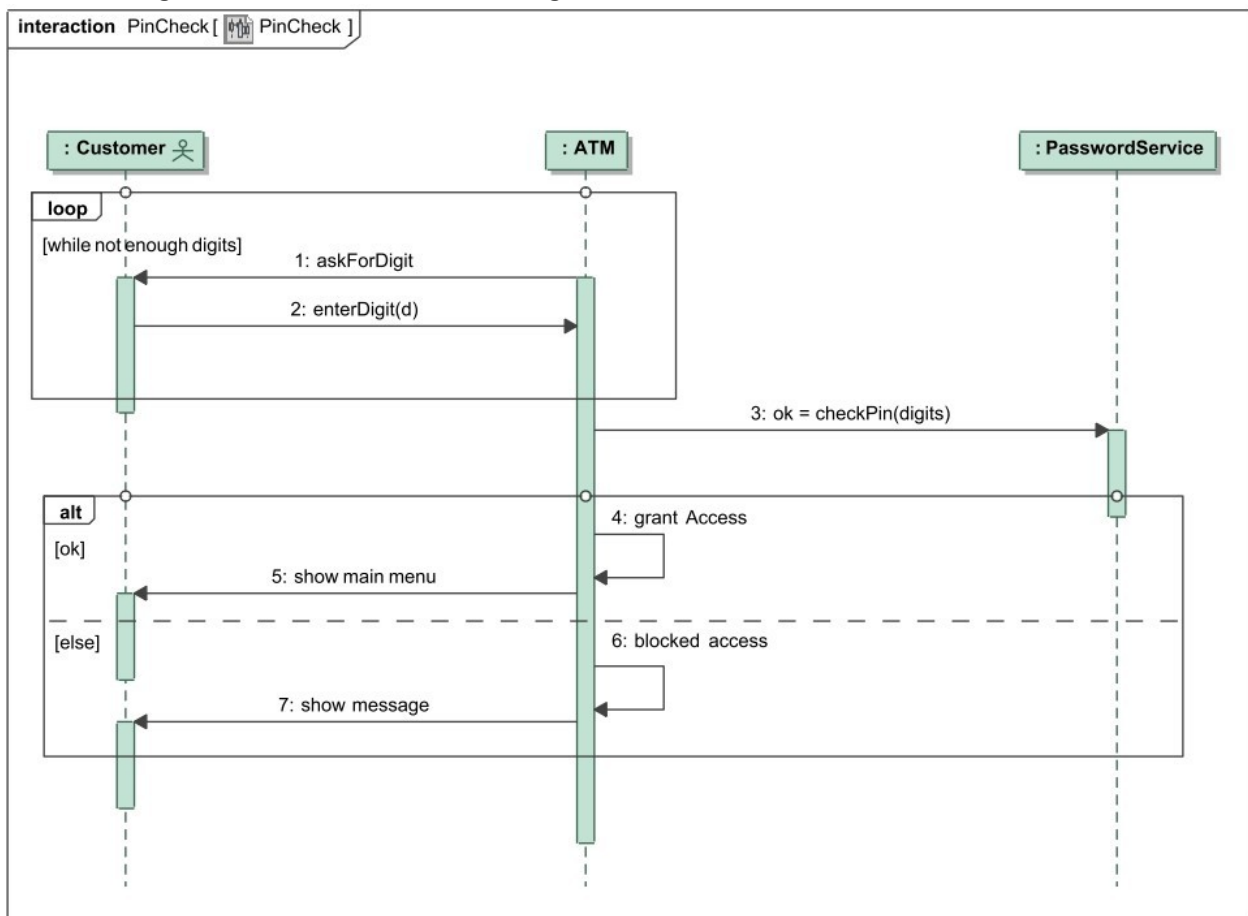
143. Entre las actividades habituales de un proceso de desarrollo están:

- Recepción, habilitación y manufactura
- Especificación, desarrollo y validación
- Arranque, estabilización y parada

144. El principio de sustitución de Liskov permite:

- Redefinir en una clase B que especializa a otra clase A el código de métodos privados de la clase A
- Garantizar que una clase B que especializa a otra clase A implementa todos los interfaces que implementa la clase A
- Evitar problemas en este supuesto: una clase cliente A tiene un método con un parámetro formal de la clase B, pero a veces se le llama con objetos de subclases de B como parámetros actuales

145. En el diagrama de secuencia de la figura:



- El usuario introduce su pin tantas veces como sea necesario hasta que el sistema ATM valida que es correcto
- El usuario introduce su pin. Si el pin es correcto, el sistema ATM permite el acceso. Si se equivoca varias veces al introducir el pin, se bloquea el acceso
- El usuario introduce dígitos hasta llegar al número necesario. El sistema ATM comprueba si son válidos. Si lo son, da acceso al menú principal. Si no lo son, bloquea el acceso

146. La figura "Diagrama de casos de uso de Web de Fotos" presenta el diagrama de casos de uso de una web para almacenar fotos. La web saca periódicamente copias de seguridad de las fotos almacenadas, sin intervención del usuario. El diagrama es:

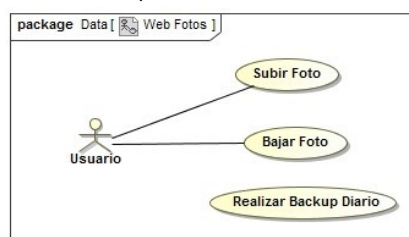
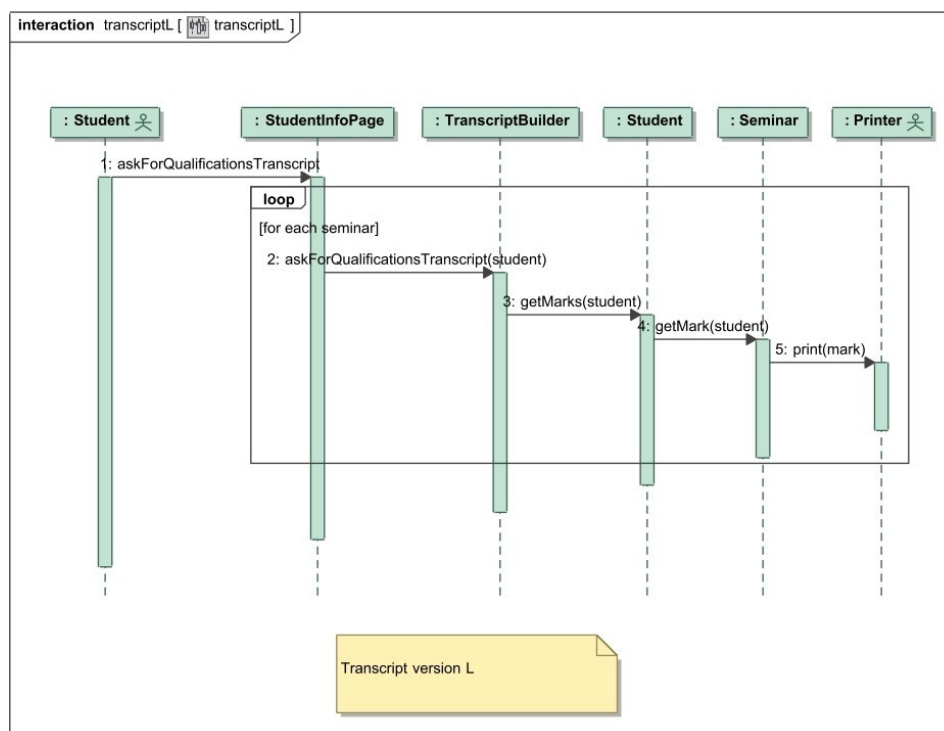
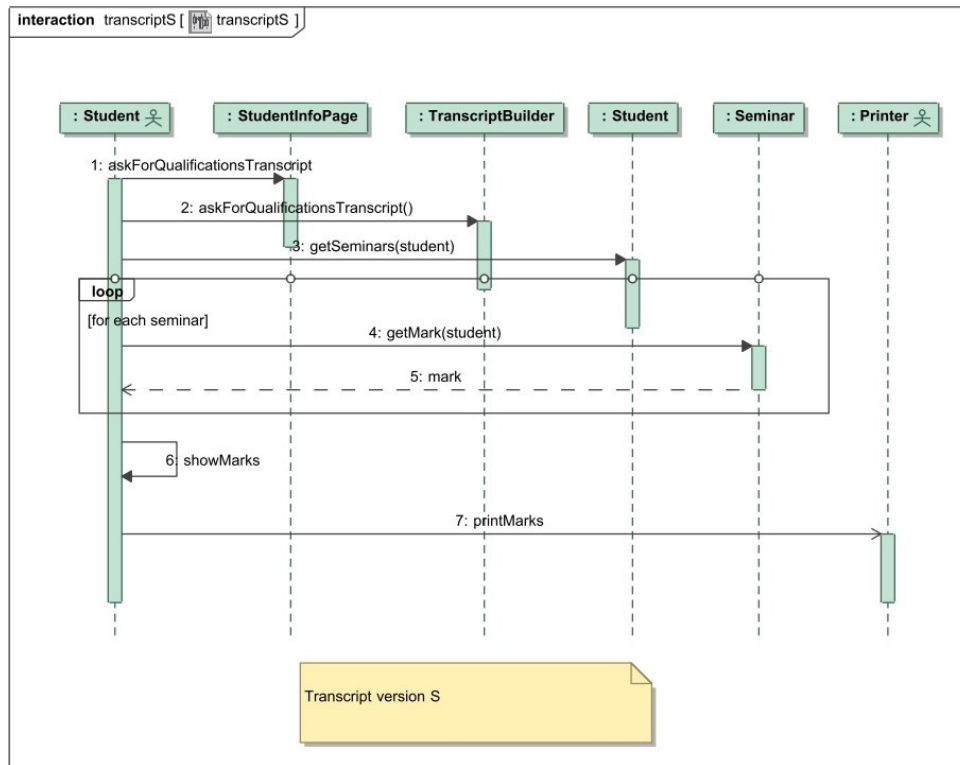
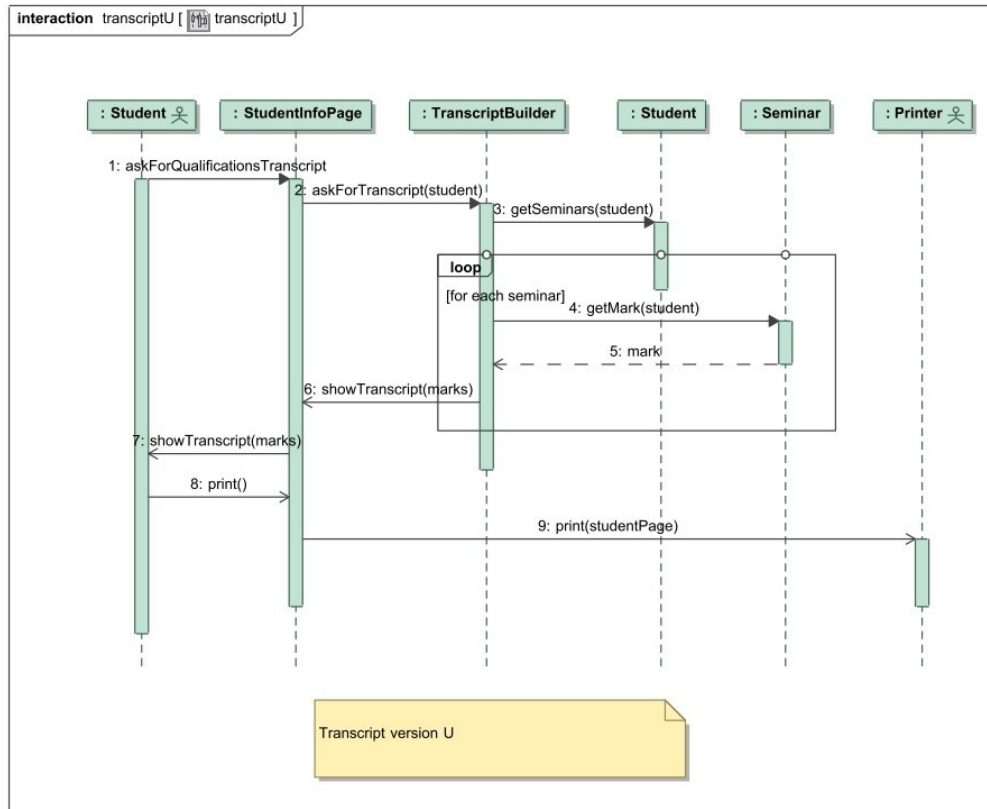


Figura "Diagrama de casos de uso de Web de Fotos"

- Incorrecto porque la web no aparece como actor secundario
- Incorrecto porque no muestra la frontera del sistema
- Correcto
- Incorrecto porque hay un caso de uso que no está asociado a ningún actor

147. Para obtener una copia de sus notas, un estudiante se conecta a una página de información y le solicita un informe de notas. La página web le pasa la petición al generador de informes. El generador busca en la información sobre el estudiante cuáles son sus asignaturas. Para cada asignatura el generador consulta su nota para generar el informe, que devuelve a la página de consulta. La página se la muestra al estudiante que puede solicitar su impresión. ¿Cuál de los diagramas de secuencia corresponde a esa descripción?





- Transcript versión L
- Transcript versión U
- Transcript versión S

148. La figura "Diagrama de clases de web de recetas" representa el modelo estructural para una web de recetas de cocina. Esta web tiene la particularidad de que incluye programas descargables para diferentes máquinas (robots de cocina). Se desea que en la web se puedan hacer búsquedas de recetas de cocina por sus ingredientes, por su dificultad, por las máquinas para las que hay programa y por las técnicas culinarias usadas en la receta.

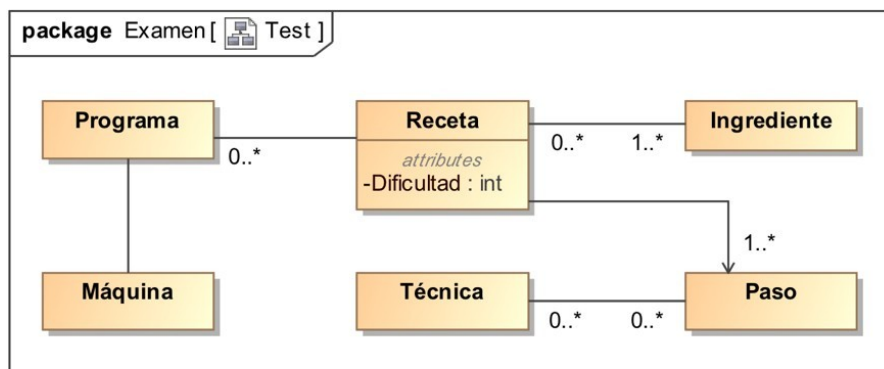


Figura "Diagrama de clases de web de recetas"

Escoja la respuesta correcta para el sistema anterior:

- Una búsqueda de receta por "Máquina" puede dar como resultado 5 "Recetas"
- No se puede buscar por "Técnica" porque la asociación entre "Paso" y "Receta" es direccional
- No se puede buscar por "Técnica" porque no hay una asociación directa con "Receta"