

JUNIO 2014

1. Según el patrón GRASP Creador, la clase A debe ser encargada de crear una instancia de la clase B si:
 - a. **A contiene o agrega instancias de B**
2. En una arquitectura de 3 capas. ¿Cómo se relacionan las capas de dominio y representación?
 - a. **La capa de dominio nunca debe depender de la capa de presentación**
3. ¿Qué patrón GOF permite crear estructuras jerárquicas de objetos de forma que el cliente pueda manejar objetos simples o compuestos indistintamente?
 - a. **Composite**
4. Cuando se usa el patrón Active Record, ¿dónde se sitúa el código de acceso a la base de datos?
 - a. **en los objetos de dominio**
5. ¿Cuál es la diferencia entre patrones de diseño y frameworks?
 - a. **los frameworks se basan en uno o varios patrones para solucionar problemas concretos**
6. En una arquitectura de capas ¿qué capas se pueden ver afectadas por una nueva funcionalidad?
 - a. **puede haber varias capas afectadas.**
7. El patrón GOF Abstract Factory:
 - a. **Provee un interfaz para crear familias de productos de forma consistente**
8. ¿Cuál de las siguientes afirmaciones es CIERTA? para que un diseño sea fácil de mantener y usar:
 - a. **Debemos mantener un bajo acoplamiento y alta cohesión**
9. ¿Cuál de las siguientes afirmaciones sobre patrones Remote Façade es FALSA?
 - a. **las fachadas remotas no deben tener estado**
10. El uso de interfaces de grano fino mejora el rendimiento de los sistemas distribuidos
 - a. **falso**
11. El patrón GRASP Indirección implica:
 - a. **relacionar dos clases mediante una clase intermedia**
12. ¿Cuál de las siguientes responsabilidades NO corresponde a la capa de presentación?
 - a. **comunicarse con las capas superiores para ofrecer funcionalidades complejas.**
13. Los objetos Data transfer Object:
 - a. **pueden contener datos de varios objetos de dominio para mejorar rendimiento**
14. ¿Qué mide el rendimiento (performance) de un sistema?
 - a. **el tiempo de respuesta**
15. Cuando se desea simplificar el acceso a un subsistema o capa, ¿qué patrón GOF es el más indicado?
 - a. **Façade**

16. ¿Con qué patrón podemos mantener la consistencia cuando se modifican dos instancias de un mismo objeto desde distintas partes del sistema?
 - a. **Unit of Work**
 17. ¿A qué nos referimos normalmente cuando hablamos de la arquitectura de un sistema?
 - a. **a los principales componentes del sistema y sus relaciones**
 18. ¿Con qué patrón de lógica de dominio se combina normalmente el patrón Row Data Gateway?
 - a. **transaction Script**
 19. ¿En qué capa se deben situar las operaciones complejas de la aplicación?
 - a. **En la capa de dominio**
 20. El acoplamiento entre clases es una medida de
 - a. **El grado de dependencia entre las clases del sistema**
 21. ¿Cuándo es conveniente usar un patrón Data Mapper?
 - a. **Cuando el modelo de dominio es complejo**
 22. ¿Cuál de los siguientes patrones no se aplica a la capa de dominio?
 - a. **table data gateway**
 23. ¿Cuál es la diferencia entre los patrones Table Module y Domain Model?
 - a. **normalmente, con Table Module hay un objeto por cada tabla, mientras que con Domain Model hay un objeto por cada fila de tabla**
 24. Las clases de diseño que surgen como resultado del patrón Fabricación Pura suelen aparecer
 - a. **Por descomposición funcional, para dividir responsabilidades**
 25. ¿Cuál es la principal ventaja de usar patrones GOF a la hora de diseñar un sistema?
 - a. **Aumenta la flexibilidad del sistema frente a futuros cambios.**
 26. Al usar el patrón Transaction Script
 - a. **Cada procedimiento representa una acción que el usuario puede ejecutar**
 27. En el patrón Model-View-Controller, cuando hay una serie de operaciones comunes que se deben realizar para cada petición del usuario, es conveniente usar el patrón:
 - a. **Front Controller**
 28. ¿Cuáles son las capas típicas de un sistema de 3 capas?
 - a. **acceso a datos, lógica de dominio y presentación**
 29. ¿Qué problema pretende evitar el patrón lazy load?
 - a. **problemas de rendimiento al cargar objetos relacionados**
 30. En el patrón GOF Observer, ¿Cuál es la clase encargada de notificar un suceso en el sistema?
 - a. **La clase que desempeña el rol Subject**
- JUNIO 2013
31. ¿Cuál de las siguientes historias de usuario está mejor definida?
 - a. **como profesora quiero poder introducir notas con formato coma y dos decimales, y que las notas menores de cinco aparezcan en rojo, con el fin de poder trasladar sin esfuerzo a las actas oficiales de la universidad.**

32. ¿qué significa que una historia de usuario debe ser cerrada?
- debe tener output significativo e identificable desde el punto de vista del usuario (se debe poder saber cuando se ha acabado de ejecutar la historia)**
33. los spikes funcionales sirven para...
- Evaluar rendimiento, impacto de carga de una nueva historia. etc.
 - Extraer más información del usuario o de cualquier otro stakeholder.
34. Un punto de historia es una unidad de medida del tamaño de una historia de usuario, y equivale...
- al tiempo que decida el equipo scrum**
35. Cuando, hablamos de requisitos ágiles, distinguimos entre épicas (epics), rasgos(features) e historias de usuario, tenemos que tener en cuenta que...
- las features y las historias normalmente implican la definición de test de aceptación, mientras que las epics no**
36. ¿Cuál de las siguientes afirmaciones es falsa?
- Un patrón puede incorporar un framework, y un framework puede hacer uso de varios patrones**
37. ¿Cuál de los siguientes grupos de patrones incluye sólo patrones de comportamiento?
- strategy, observer, command**
38. Uno de los principales inconvenientes del patrón Builder es:
- Existe un alto acoplamiento entre el cliente y el constructor del producto**
39. ¿Cuál de los siguientes grupos de patrones incluye solo patrones estructurales?
- façade composite proxy**
40. ¿qué patrón permitiría crear colas de prioridad o delegar la ejecución de distintas operaciones?
- command**
41. ¿qué patrón sería mejor aplicar cuando no sabemos la clase de los objetos que van a crear las subclasses de una clase abstracta?
- Factory Method**
42. El patrón command hace uso, entre otro, de los siguientes patrones GRASP
- indirección, fabricación pura y polimorfismo**
43. Cuando necesitamos trabajar con distintos tipos de recursos que dependen de la localización (e.g. diálogos, formatos de fecha, etc.) debemos pensar en la conveniencia de usar el patrón...
- composite**
44. ¿Cuál es el principal inconveniente de la fabricación pura?
- Si se abusa de ello, aparecen clases con un único método (clases "fuctoides"), dando lugar a un diseño centrado en funciones que se implementa sobre un diseño orientado a objetos.**
45. ¿Cuál de las siguientes características NO corresponde a una metodología ágil?
- Prima la creación de código sobre la documentación del sistema.**

46. Hemos implementado un sistema para la agencia F de noticias de manera que la agencia puede informar inmediatamente, cuando ocurre cualquier evento, a cualquiera de sus clientes por Email, sms, twitter o cualquier otro canal que puedan requerir en un futuro ¿con qué patron lo hemos implementado?
- a. **Observer**
47. Una de las principales preocupaciones en scrum es generar código de calidad.
- a. **Cierto. Por eso se hacen reuniones diarias para revisar código**
48. ¿Cuál es el principal objetivo de las reuniones diarias de Scrum?
- a. **Detectar impedimentos en el avance del equipo**
49. En scrum ¿cuál es el rol encargado de mantener las historias de usuario?
- a. **product owner**
50. En scrum, ¿quién es el encargado de dar el visto bueno a las historias de usuario que se han implementado duran un sprint?
- a. **el cliente**
51. ¿Cuál de las siguientes características NO corresponde a un equipo de trabajo scrum?
- a. **El equipo decide cuales son las historias de usuario de mayor valor en cada iteración**
52. ¿Que mide la velocidad de un equipo?
- a. **El numero de puntos de historia que puede completar en una iteración**
53. ¿Cual de las siguientes historias es una mala historia de usuario?
- a. **El programa se conectará a la BD a través de un pool de conexiones**
54. En XP
- a. **Al igual que en Scrum, los clientes ayudan a desarrollar las historias de usuario**
55. El Pair Programming
- a. **No perjudica de manera sustancial la velocidad de desarrollo del equipo y mejora la calidad del código resultante.**
56. ¿Cuál de las siguientes afirmaciones es falsa? El uso de una aproximación XP para el desarrollo software...
- a. **Normalmente requiere que los clientes tengan muy claros los requisitos**
57. En XP
- a. **Antes de comenzar a planificar las releases (entregas) es conveniente realizar una iteración 0 donde se desarrolla un spike arquitectural que permite obtener una metáfora del sistema que guía el resto del desarrollo.**
58. Un componente de entidad de negocio
- a. **Representa a las entidades de dominio localizadas en la capa de lógica de negocio**
59. Al usar la herramienta OOH4RIA, ¿qué es lo que NO sería un error en una relación de herencia?
- a. **La clase hija defina una operación new con el mismo nombre que el padre**
60. El componente de proceso:
- a. **Se encarga de exponer al interfaz de usuario los métodos de mayor nivel de lógica de negocio**

JULIO 2014

52. En el patron Model-view-controller, el controlador se encarga de:
- Procesar la entrada del usuario.**
53. Cuando se accede desde varias partes del sistema a un objeto guardado en la base de datos ¿qué patrón nos garantiza que siempre exista una única instancia de ese objeto para evitar problemas de consistencia
- Unit of work.**
54. ¿Qué patron GOF permite crear estructuras jerárquicas de objetos, de forma que el cliente pueda manejar objetos simples o compuestos indistintamente
- Composite**
55. ¿Cuál es el principal inconveniente de un sistema con objetos distribuidos?
- Es difícil de modelar**
56. Para que un diseño de clases sea más fácil de entender y usar, la cohesión debe ser:
- Alta**
57. Para disminuir la dependencia entre clases y favorecer la reutilización de código el acoplamiento debe ser:
- lo más bajo posible**
58. ¿Cómo afecta un número de capas mayor al rendimiento del sistema?
- Negativamente, ya que aumenta la comunicación entre capas**
59. ¿Por qué puede surgir una nueva capa en la arquitectura de un sistema?
- por descomposición de una capa existente, para disminuir su complejidad**
60. Como se debe documentar la arquitectura de un sistema
- Con diagramas de clases**
61. El uso de patrones GRASP Creador implica
- disminuye el acoplamiento entre clases**
62. En cuantos grupos se dividen los patrones GOF
- En tres: creacionales, estructurales y de comportamientos**
63. ¿Qué desventaja tiene el patron Transaction Script?
- Los procedimientos creados pueden tener código duplicado**
64. En una arquitectura de capas
- Depende, es una decisión de los diseñadores**
65. ¿Qué patrón se usa para organizar la lógica de negocio en procedimientos transaccionales?
- Transaction Script**
66. ¿Con que patrón de lógica de negocio se combina normalmente el patrón Active Record?
- Domain Model**
67. ¿Qué patron GOF define una interfaz para crear un objeto, delegando la decisión de que clase crear en una subclase?
- Factory Method**

68. ¿Qué patrón GOF permite representar operaciones mediante objetos, facilitando llevar un registro de las operaciones realizadas e incluso poder deshacerlas?
- command**
69. ¿Qué cualidad mide la capacidad de un sistema para soportar una mayor carga de trabajo?
- Escalabilidad**
70. El patrón que permite gestionar la navegación entre distintas pantallas de una aplicación es
- Application controller**
71. ¿Qué clase de requisitos son los que condicionan la elección de una arquitectura concreta?
- los no funcionales**
72. Con qué patrón de lógica de dominio se combina normalmente el patrón Data Mapper?
- Domain Model**
73. Cuando es preferible usar el patrón Domain Model?
- cuando la lógica de dominio es compleja**
74. ¿Qué patrón GOF se usa para definir familias de algoritmos intercambiables?
- Strategy**
75. En un diagrama de clases de UML la composición es:
- Una relación sin ambigüedades que sugiere fuertemente una relación de todo/parte.
76. El objetivo del patrón Fachada es...
- Simplificar el acceso a la interfaz mediante funcionalidades**
77. El patrón Command permite:
- Manipular comandos como objetos.**
79. Si necesitamos proporcionar una interfaz para crear familias de objetos relacionados dependientes, sin especificar sus clases concretas. ¿Qué patrón debemos aplicar?:
- Patrón Abstract Factory**
80. De los siguientes patrones, indica cuál de ellos es Creacional:
- El patrón Singleton**