

Recopilación examen Junio 2014 – Julio 2015 – Junio 2015
Versión 0.4

1. Según el patrón GRASP Creador, la clase A debe ser la encargada de crear una instancia de la clase B si:
 - (A) A contiene o agrega instancias de B
 - (B) B usa instancias de A
 - (C) Las dos son ciertas.

2. En una arquitectura de 3 capas. ¿Cómo se relacionan las capas de dominio y presentación?
 - (A) La capa de dominio nunca debe depender de la capa de presentación.
 - (B) La capa de presentación nunca debe depender de la capa de dominio.
 - (C) Nunca deben estar relacionadas.

3. ¿Qué patrón GOF permite crear estructuras jerárquicas de objetos, de forma que el cliente pueda manejar objetos simples o compuestos indistintamente?
 - (A) Builder
 - (B) Strategy
 - (C) Composite

4. Cuando se usa el patrón Active Record, ¿Dónde se sitúa el código de acceso a la base de datos?
 - (A) En el controlador del sistema.
 - (B) En las clases especializadas de acceso a datos.
 - (C) En los objetos de dominio.

5. ¿Cuál es la diferencia entre los patrones de diseño y los frameworks?
 - (A) Los frameworks se basan en uno o varios patrones para solucionar problemas complejos.
 - (B) Los patrones utilizan frameworks para ofrecer soluciones reutilizables.
 - (C) Ninguna, los frameworks se usan para el diseño detallado del sistema.

6. En una arquitectura de capas, ¿Qué capas se pueden ver afectadas por una nueva funcionalidad?
 - (A) Solamente la capa de dominio.
 - (B) Solamente la capa de presentación.
 - (C) Pueden verse afectadas varias capas.

7. El patrón GRASP Abstract Factory
 - (A) Posee un interfaz para crear familias de productos de forma consistente.
 - (B) Permite controlar el número de instancias de los objetos que se crearán.
 - (C) Disminuye notablemente el número de clases del sistema.

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 una baja cohesión.
 - (B) Debemos mantener un alto acoplamiento y una baja cohesión.
 - (C) Debemos mantener un bajo acoplamiento y una alta cohesión.

9. ¿Cuál de las siguientes afirmaciones sobre el patrón Remote Façade es FALSA?
 - (A) Las fachadas remotas no deben contener lógica de dominio.
 - (B) Las fachadas remotas no deben contener un estado.
 - (C) Las fachadas remotas deben ofrecer un único método para acceder a todas las propiedades de los objetos.

10. El uso de interfaces de grano fino mejora el rendimiento de los sistemas distribuidos:

- (A) Verdadero.
- (B) Sólo si el diseño es orientado a objetos.
- (C) Falso.**

11. El patrón GRASP indirección implica:

- (A) Relacionar dos clases mediante una clase intermedia.**
- (B) Favorecer las relaciones directas entre clases, evitando caminos más largos.
- (C) Asignar menos responsabilidades a las clases que hay en las fronteras con otras capas.

12. ¿Cuál de las siguientes responsabilidades NO corresponde a la capa de presentación?

- (A) Comunicarse con las capas superiores para ofrecer funcionalidades complejas.**
- (B) Manejar la interacción y mostrar información al usuario.
- (C) Comunicarse con la capa de lógica de dominio para transmitir los cambios.

13. Los objetos DATA Transfer Object:

- (A) Solamente pueden contener datos de un objeto de dominio para mantener la consistencia.
- (B) Pueden contener datos de varios objetos de dominio para mejorar el rendimiento.**
- (C) No deben contener datos de los objetos de dominio para aumentar la cohesión.

14. ¿Qué mide el rendimiento (performance) de un sistema?

- (A) El tiempo de respuesta.**
- (B) El tiempo que el sistema está funcionando.
- (C) El tiempo que el sistema funciona sin fallos.

15. Cuando se desea simplificar el acceso a un subsistema o capa, ¿Qué patrón GOF es el más indicado?

- (A) Strategy
- (B) Proxy
- (C) Façade**

16. ¿Con que patrón podemos mantener la consistencia cuando se modifican dos instancias de un mismo objeto desde distintas partes del sistema?

- (A) Unit of Work**
- (B) Identity Map
- (C) Lazy Road

17. ¿A qué nos referimos normalmente cuando hablamos de la arquitectura de un sistema?

- (A) A la infraestructura de hardware que dará soporte al sistema.
- (B) A los principales componentes del sistema y sus relaciones.**
- (C) A la estructura de bajo nivel de cada componente del sistema.

18. ¿Con qué patrón de lógica de domino se combina normalmente el patrón Row Data Gateway?

- (A) Table Date Gateway
- (B) Domain Model
- (C) Transaction Script.**

19. En qué capa se deben situar las operaciones complejas de aplicación?

- (A) En la capa de dominio.
- (B)** En la capa de servicio.
- (C) En la capa de presentación.

20. El acoplamiento entre clases es una medida de:

- (A)** El grado de dependencia entre las clases del sistema.
- (B) Cuantas funcionalidades distintas se asigna a cada fase.
- (C) Cuántas clases intermedias hay que recorrer para llegar de una clase A a una clase B.

21. Cuando es conveniente usar el patrón Data Mapper?

- (A) Cuando el módulo de dominio es sencillo.
- (B)** Cuando el modelo de dominio es complejo.
- (C) Cuando se usa una base de datos orientada a objetos.

22. ¿Cuál de los siguientes patrones no se aplica a la capa de Dominio?

- (A) Transaction Script.
- (B) Table Module.
- (C)** Table Data Gateway.

23. ¿Cuál es la diferencia entre patrones Table Module y Domain Model?

- (A) Table Module pertenece a la capa de acceso a datos y Domain Model a la capa de lógica de dominio.
- (B)** Normalmente con Table Module hay un objeto por cada tabla, mientras con Domain Model hay un objeto por cada fila de la tupla.
- (C) Las dos son ciertas.

24. Las clases de dominio que surgen como resultado del patrón de Fabricación Pura aparecen:

- (A) Para representar objetos de dominio.
- (B)** Por descomposición funcional, para dividir responsabilidades.
- (C) Por la aparición de jerarquías de herencia.

25. ¿Cuál es la principal ventaja de usar patrones GOF a la hora de diseñar un sistema?

- (A) Permite disminuir el número de clases del sistema.
- (B) Hace que el sistema sea más fácil de comprender a cambio de disminuir las posibilidades de reutilización de código.
- (C)** Aumenta la flexibilidad del sistema frente a futuros cambios.

26. Al usar el patrón Transaction Script:

- (A) Cada procedimiento creado es independiente del resto de capas del sistema.
- (B) Surgen dos tipos de procedimientos: los que acceden a la base de datos y los que se comunican con el resto de capas del sistema.
- (C)** Cada procedimiento representa una acción que el usuario debe ejecutar.

27. En el patrón Model-View-Controller, cuándo hay una serie de operaciones comunes que se deben realizar para cada petición del usuario, es conveniente usar el patrón:

- (A) Page Controller.
- (B)** Front Controller.
- (C) Application Controller.

28. ¿Cuáles son las capas típicas de un sistema de 3 capas?

- (A) Acceso a datos, servicio y lógica.
- (B) Servicio, lógica de dominio y presentación.
- (C)** Acceso a datos, lógica de servicio y presentación.

29. ¿Qué problema pretende evitar el patrón Lazy Road?

- (A) Problemas de integridad referencial al cargar objetos relacionados.
- (B)** Problemas de rendimiento al cargar objetos relacionados.
- (C) Problemas por el uso excesivo de herencia en la lógica del dominio.

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 Observer.
- (B) La clase que desempeña el rol Concrete Observer.
- (C)** La clase que desempeña el rol Subject.

31. Para que un diseño de clases sea más fácil de entender y usar la cohesión debe ser:

- (A) Alta
- (B)** Baja
- (C) La cohesión no influye.

32. ¿Qué patrón está pensado para poder intercambiar distintos comportamientos en tiempos de ejecución?

- (A)** Strategy.
- (B) Proxy.
- (C) Command.

33. ¿Cuál de los siguientes patrones NO es de comportamiento?

- (A) Observer
- (B)** Proxy
- (C) Command

34. El uso del patrón GRASP Creador implica:

- (A)** Disminuye el acoplamiento entre clases.
- (B) Aumenta el acoplamiento entre clases.
- (C) No afecta al acoplamiento entre clases.

35. ¿Qué inconveniente tiene el uso del patrón Composite?

- (A)** El cliente no distingue entre clases simples y compuestas.
- (B) No permite añadir nuevas clases simples.
- (C) Resulta complicado imponer restricciones sobre la estructura de objetos compuestos.

36. ¿Qué inconveniente tiene el uso del patrón Abstract Factory?

- (A) Los productos de distintas familias comparten el mismo interfaz.
- (B)** No es fácil añadir productos nuevos.
- (C) Para el cliente no es fácil seleccionar la familia de productos a crear.

37. ¿Con qué patrón podemos reducir el acoplamiento entre dos partes de un sistema cuando una clase usa un conjunto de clases de otra?

- (A) Composite
- (B)** Façade
- (C) Proxy

38. ¿Qué ventaja tiene el patrón Builder?

- (A) El director no necesita conocer los pasos del proceso de construcción.
- (B) Cada constructor tiene un interfaz distinto.
- (C)** Permite crear distintos productos siguiendo el mismo proceso.

39. ¿Cuál es la principal motivación de los patrones GRASP?

- (A) Crear modelos que no cambian a lo largo del proyecto.
- (B)** Proteger al sistema frente a las posibles variaciones.
- (C) Identificar las clases del modelo de dominio que se comunican con el resto del sistema.

40. En el patrón Command, ¿Qué información necesitan los comandos para ejecutarse?

- (A) Qué clase ha instanciado la acción.
- (B)** Qué clase ha invocado la acción.
- (C) Qué clase es la receptora de la acción.

41. Qué patrón sería más adecuado para llevar un recuento de número de veces que se llama a cada método de un objeto?

- (A) Strategy.
- (B) Observer.
- (C)** Proxy.

42. En el patrón Observer, ¿Qué rol es el encargado de llevar un registro de los objetos que deben ser notificados cuando hay un cambio?

- (A)** Subject.
- (B) Observer.
- (C) Client.

43. ¿Cuál de los siguientes tipos de asociación implica un menor grado de acoplamiento?

- (A) Uso
- (B) Herencia.
- (C)** Implementación de interfaces.

44. El patrón GRASP indirección implica:

- (A)** Relacionar dos clases mediante otra intermedia.
- (B) Favorecer las relaciones directas entre clases.
- (C) Asignar menos responsabilidades a las clases que hay en al frontera con otras capas.

45. Para disminuir la dependencia entre clases y favorecer la reutilización de código, el acoplamiento debe ser:

- (A)** Lo más bajo posible.
- (B) Lo más alto.
- (C) El acoplamiento no influye.

46. GRASP es el acrónimo de patrones generales de software para la asignación de responsabilidades. Con respecto a dicha responsabilidades, ¿Cuáles de las siguientes afirmaciones es falsa?

- (A) Los métodos se implementan para cubrir responsabilidades.
- (B) Cada responsabilidad se traduce en un método que se debe asignar alguna clase software.
- (C)** Los métodos pueden colaborar con otros métodos u objetos para cubrir una determinada responsabilidad.

47. ¿Cuál de las siguientes afirmaciones relacionadas con la herencia es verdadera?
- (A)** La herencia escala bien cuando aumenta el número de variaciones del sistema.
 - (B) Siempre que tenemos una jerarquía de herencia, es posible sustituirla por una agregación o composición (delegación), lo cual aumenta la flexibilidad del sistema.
 - (C) La herencia mejora el acoplamiento del sistema.
48. ¿Cuál de los siguientes efectos se puede deber al alto acoplamiento entre clases?
- (A) Mayor cohesión de las clases fuertemente acopladas.
 - (B) Mayor facilidad de reuso.
 - (C)** Alta probabilidad de propagación inadvertida de errores ante modificaciones en el código.
49. ¿Qué beneficio obtenemos con el patrón GRASP creador?
- (A) Reducir cohesión
 - (B)** Reducir acoplamiento.
 - (C) Evitar la necesidad de inicializar la clase creada.
50. ¿Cuál es la principal motivación de los patrones GRASP?
- (A)** Proteger el sistema de posibles variaciones.
 - (B) Elaborar modelos que no cambien a lo largo del proyecto.
 - (C) Identificar las clases del modelo de dominio que se comunican con el resto del sistema.
51. ¿Con qué patrón GOF se relaciona el patrón GRASP controlador?
- (A)** Fachada
 - (B) Fabricación pura
 - (C) Factory Method.
52. ¿Cuáles de estas clases es más probable que requieran la aplicación de un patrón singleton?
- (A) Los comandos de un patrón command.
 - (B) Los componentes de un patrón composite.
 - (C)** Las factorías completas de un patrón abstract factory.
53. Hemos implementado un sistema al que queremos añadir ahora la posibilidad de leer una serie de datos de configuración desde distintas fuentes, como por ejemplo ficheros XML, base de datos, etc ¿Qué patrón GOF se adapta mejor a este problema?
- (A) Command.
 - (B) Strategy.
 - (C)** Façade.
54. En el patrón OBSERVER...
- (A) La lista de observadores concretos puede incluir objetos de distinto tipo.
 - (B) El sujeto puede ser que envíe la observación necesaria al observador mediante el método update().
 - (C)** Todas son ciertas.

55. Nos han pedido diseñar un sistema de aprendizaje... En el, vamos a tener distintos objetos de aprendizaje que se deben instanciar en función del nivel de experiencia del alumno (De momento tres: K12, Adulto, Profesional). ¿Qué patrón aplicarías para solucionar este problema de manera flexible y extensible a nuevos niveles de experiencia? (e.g alumnos universitarios).

- (A) Builder.
- (B)** Abstract Factory.
- (C) Proxy

56. En el sistema de aprendizaje uno de los requisitos es la construcción de guías docentes. Estas guías están compuestas de un contexto, una serie de objetivos, contenidos, una descripción de la metodología docente, un cronograma, y unos criterios de evaluación. ¿Qué patrón GOFT es el más adecuado para diseñar esta construcción?

- (A)** Builder.
- (B) Factory method.
- (C) Abstract factory.

57. La clase fachada del patrón fachada.

- (A) Se limita a transmitir invocaciones desde el exterior hacia los objetos del sistema que oculta, preservando la interfaz de estos últimos.
- (B)** Realiza la adaptación entre su interfaz y la interfaz de los objetos del sistema que este oculta.
- (C) Es una clase abstracta de la que heredan las fachadas concretas de cada componente interno del sistema.

58. El patrón builder se utiliza para...

- (A) Permitir la variación de la representación interna de un producto.
- (B) Simplificar el código cliente que crea objetos complejos.
- (C)** Todas son ciertas.

59. En patrón OBSERVER ...

- (A) La clase SUBJECT debe tener acceso de algún modo a los datos del OBSERVER.
- (B) La clase OBSERVER debe tener acceso de algún modo a los datos del SUBJECT.
- (C)** No existe ningún acoplamiento entre las clases SUBJECT Y OBSERVER.

60. El patrón Abstract Factory contribuye a preservar el principio OPEN-CLOSED de Bertrant Merrey (Diseños abiertos a la extensión y cerrados al cambio) cuando...

- (A) El punto más probable de variación supone la adición de nuevas factorías concretas
- (B) El punto más probable de variación supone la adición de nuevas familias de productos.
- (C)** Los puntos más probables de variación suponen (a) la adición de nuevas familias de producto y/o (b) la adición de nuevas factorías concretas .