1. Según el patrón GRASP Creador la clase A debe ser la encargada de crear una instancia de la clase B si:
2. A contiene o agrega instancias de B
3. B usa instancias de A
4. Las dos son ciertas
5. En una arquitectura de 3 capas. ¿Cómo se relacionan las capas de dominio y presentación?
6. La capa de dominio nunca debe depender de la capa de presentación.
7. La capa de presentación nunca debe depender de la capa de dominio.
8. Nunca deben estar relacionadas.
9. ¿Qué patrón GOF permite crear estructuras jerárquicas de objetos, de forma que el cliente pueda manejar objetos simples o compuestos indistintamente?
10. Builder
11. Strategy
12. Composite
13. Cuando se usa el patrón Active record, ¿dónde se sitúa el código de acceso a la base de datos?
14. En el controlador del sistema
15. En las clases especializadas de acceso a datos
16. En los objetos de dominio
17. ¿Cuál es la diferencia entre los patrones de diseño y los frameworks?
18. Los frameworks se basan en uno o varios patrones para solucionar problemas complejos.
19. Los patrones utilizan frameworks para ofrecer soliciones reutilizables.
20. Ninguna, los frameworks se usan para el diseño detallado del sistema.
21. En una arquitectura de capas, ¿Qué capas se pueden ver afectadas por una nueva funcionalidad?
22. Solamente la capa de dominio
23. Solamente la capa de presentación
24. Puede haber varias capas afectadas
25. El patrón GRASP Abstract Factory:
26. Posee un interfaz para crear familias de productos de forma consistente
27. Permite controlar el número de instancias de los objetos que se crearán
28. Disminuye notablemente el número de clases del sistema
29. ¿Cuál de las siguientes afirmaciones es CIERTA? Para que un diseño sea fácil de mantener y usar:
30. Debemos mantener un bajo acoplamiento y una baja cohesión
31. Debemos mantener un alto acoplamiento y una baja cohesión
32. Debemos mantener un bajo acoplamiento y una alta cohesión
33. ¿Cuál de las siguientes afirmaciones sobre el patrón Remote Façade es FALSA?
34. Las fachadas remotas no deben contener lógica de dominio.
35. Las fachadas remotas no deben tener estado
36. Las fachadas remotas deben ofrecer un único método para acceder a todas las propiedades de un objetos
37. El uso de interfaces de grano fino mejora el rendimiento de los sistemas distribuidos:
38. Verdadero
39. Solo si el diseño es orientado a objetos
40. Falso
41. El patrón GRASP indirección implica:
42. Relacionar dos clases mediante una clase intermedia
43. Favorecer las relaciones directas entre clases, evitando caminos más largos
44. Asignar menos responsabilidades a las clases que hay en las fronteras con otras capas
45. ¿Cuál de las siguientes responsabilidades NO corresponde a la capa de presentación?
46. Comunicarse con las capas superiores para ofrecer funcionalidades complejas
47. Manejar la interacción y mostrar información al usuario
48. Comunicarse con la capa de lógica de dominio para transmitir los cambios realizados por el usuario
49. Los objetos Data Transfer Object:
50. Solamente pueden contener datos de un objeto de dominio para mantener la consistencia
51. Pueden contener datos de varios objetos de dominio para mejorar el rendimiento
52. No deben contener datos de los objetos de dominio para aumentar la cohesión
53. ¿Qué mide el rendimiento (performance) de un sistema?
54. El tiempo de respuesta
55. El tiempo que el sistema está funcionando
56. El tiempo que el sistema funciona sin fallos
57. Cuando se desea simplificar el acceso a un ¿subsistema? o capa, ¿Qué patrón GOF es el más indicado?
58. Strategy
59. Proxy
60. Façade
61. ¿Con que patrón podemos mantener la consistencia cuando se modifican dos instancias de un mismo objeto desde distintas partes del sistema?
62. Unit of Work
63. Identity Map
64. Lazy Road
65. ¿A que nos referimos normalmente cuando hablamos de la arquitectura de un sistema?
66. A la infraestructura de hardware que dará soporte al sistema
67. A los principales componentes del sistema y sus relaciones
68. A la estructura de bajo nivel de cada componente del sistema
69. ¿Con qué patrón de lógica de dominio se combina normalmente el patrón Row Data Gateway?
70. Table Data Gateway
71. Domain model
72. Transaction Script
73. ¿En que capa se deben situar las operaciones complejas de la aplicación?
74. En la capa de dominio
75. En la capa de servicio
76. En la capa de presentación
77. El acoplamiento entre clases es una medida de:
78. El grado de dependencia entre las clases del sistema
79. Cuantas funcionalidades distintas se asigna a cada clase
80. Cuántas clases intermedias hay que recorrer para llegar de una clase A a una clase B
81. ¿Cuándo es conveniente usar el patrón Data Mapper?
82. Cuando el modelo de dominio es sencillo
83. Cuando el modelo de dominio es complejo
84. Cuando se usa una base de datos orientada a objetos
85. ¿Cuál de los siguientes patrones no se aplica a la capa de Dominio?
86. Transaction Script
87. Table Module
88. Table Data Gategay
89. ¿Cuál es la diferencia entre patrones Table Module y Domain Model?
90. Table Module pertenece a la capa de acceso a datos y Domain Model a la capa de lógica de dominio
91. Normalmente con Table Module hay un objeto por cada tabla, mientras con Domain Model hay un objeto por cada fila de la tupla
92. Los dos son ciertas
93. Las clases de dominio que surgen como resultado del patrón Fabricación Pura aparecer:
94. Para representar objetos de dominio
95. Por descomposición funcional, para dividir responsabilidades
96. Por la aparición de jerarquías de herencia
97. ¿Cuál es la principal ventaja de usar patrones GOF a la hora de diseñar un sistema?
98. Permite disminuir el número de clases del sistema
99. Hace que el sistema sea más fácil de comprender, a cambio de disminuir las posibilidades de reutilización de código
100. Aumenta la flexibilidad del sistema frente a futuros cambios
101. Al usar el patrón Transaction Script:
102. Cada procedimiento creado es independiente del resto de capas del sistema
103. 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
104. Cada procedimiento representa una acción que el usuario puede ejecutar
105. En el patrón Model-View-Controler, cuándo hay una serie de operaciones comunes que se deben realizar para cada petición del usuario, es conveniente usar el patrón:
106. Page Controller
107. Front Controller
108. Application Controller
109. ¿Cuáles son las capas típicas de un sistema de 3 capas?
110. Acceso a datos, servicio y lógica
111. Servicio, lógica de dominio y presentación
112. Acceso a datos, lógica de dominio y presentación
113. ¿Qué problema pretende evitar el patrón Lazy Road?
114. Problemas de integridad referencial al cargar objetos relacionados
115. Problemas de rendimiento al cargar objetos relacionados
116. Problemas por el uso excesivo de herencia en la lógica de dominio
117. En el patrón GOF Observer, ¿cuál es la clase encargada de notificar un suceso en el sistema?
118. La clase que desempeña el rol Observer
119. La clase que desempeña el rol Concrete Observer
120. La clase que desempeña el rol Subject

RESPUESTAS

1-A

2-A

3-C

4-C

5-A

6-C

7-A

8-C

9-B

10-C

11-A

12-A

13-B

14-A

15-C

16-A

17-B

18-C

19-B

20-A

21-B

22-C

23-B

24-B

25-C

26-C

27-B

28-C

29-B

30-C