# Recopilación DSS

**1. Según el patrón GRASP Creador, la clase A debe ser encargada de crear una instancia de la clase B si:**

- **A contiene o agrega instancias de B**

**2. En una arquitectura de 3 capas. ¿Cómo se relacionan las capas de dominio y representación?**

- **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?**

- **Composite**

**4. Cuando se usa el patrón Active Record, ¿dónde se sitúa el código de acceso a la base de datos?**

- **en los objetos de dominio**

**5. ¿Cuál es la diferencia entre patrones de diseño y frameworks?**

- **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?**

- **puede haber varias capas afectadas.**

**7. El patrón GOF Abstract Factory:**

- **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:**

- **Debemos mantener un bajo acoplamiento y alta cohesión**

**9. ¿Cuál de las siguientes afirmaciones sobre patrones Remote Façace es FALSA?**

- **las fachadas remotas no deben tener estado**

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

- **falso**

**11. El patrón GRASP Indirección implica:**

- **relacionar dos clases mediante una clase intermedia**

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

- **comunicarse con las capas superiores para ofrecer funcionalidades complejas.**

**13. Los objetos Data transfer Object:**

- **pueden contener datos de varios objetos de dominio para mejorar rendimiento**

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

- **el tiempo de respuesta**

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

- **Façade**

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

- **Unit of Work**

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

- **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?**

- **transaction Script**

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

- **En la capa de dominio**

**20. El acoplamiento entre clases es una medida de**

- **El grado de dependencia entre las clases del sistema**

**21. ¿Cuándo es conveniente usar un patrón Data Mapper?**

- **Cuando el modelo de dominio es complejo**

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

- table data gateway

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

- 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**

- Por descomposición funcional, para dividir responsabilidades

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

- Aumenta la flexibilidad del sistema frente a futuros cambios.

**26. Al usar el patrón Transaction Script**

- 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:**

- Front Controller

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

- acceso a datos, lógica de dominio y presentación

**29. ¿Qué problema pretende evitar el patrón lazy load?**

- 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?**

- La clase que desempeña el rol Subject

**31. ¿Cuál de las siguientes afirmaciones es falsa?**

- Un patrón puede incorporar un framework, y un framework puede hacer uso de varios patrones

**32. ¿Cuál de los siguientes grupos de patrones incluye sólo patrones de comportamiento?**

- strategy, observer, command

**33. Uno de los principales inconvenientes del patrón Builder es:**

- **Existe un alto acoplamiento entre el cliente y el constructor del producto**

**34. ¿Cuál de los siguientes grupos de patrones incluye solo patrones estructurales?**

- **façade composite proxy**

**35. ¿qué patrón permitiría crear colas de prioridad o delegar la ejecución de distintas operaciones?**

- **command**

**36. ¿qué patrón sería mejor aplicar cuando no sabemos la clase de los objetos que van a crear las subclases de una clase abstracta?**

- **Factory Method**

**37. El patrón command hace uso, entre otro, de los siguientes patrones GRASP**

- **indirección, fabricación pura y polimorfismo**

**38. 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**

**39. ¿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.**

**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 patrón lo hemos implementado?**

- **Observer**

**47. En el patrón Model-view-controller, el controlador se encarga de:**

- **Procesar la entrada del usuario.**

**48. 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.**

**49. ¿Cuál es el principal inconveniente de un sistema con objetos distribuidos?**

- **Es difícil de modelar**

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

- **Alta**

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

- **lo más bajo posible**

**52. ¿Cómo afecta un número de capas mayor al rendimiento del sistema?**

- **Negativamente, ya que aumenta la comunicación entre capas**

**53. ¿Por qué puede surgir una nueva capa en la arquitectura de un sistema?**

- **por descomprosición de una capa existente, para disminuir su complejidad**

**54. Como se debe documentar la arquitectura de un sistema**

- **Con diagramas de clases**

**55. El uso de patrones GRASP Creador implica**

- **disminuye el acoplamiento entre clases**

**56. En cuantos grupos se dividen los patrones GOF**

- **En tres: creacionales, estructurales y de comportamientos**

**57. ¿Qué desventaja tiene el patron Transaction Script?**

- **Los procedimientos creados pueden tener código duplicado**

**58. En una arquitectura de capas**

- **Depende, es una decisión de los diseñadores**

**59. ¿Qué patrón se usa para organizar la lógica de negocio en procedimientos transaccionales?**

- **Transaction Script**

**60. ¿Con que patrón de lógica de negocio se combina normalmente el patrón Active Record?**

- **Domain Model**

**61. ¿Qué patron GOF define una interfaz para crear un objeto, delegando la decisión de que clase crear en una subclase?**

- **Factory Method**

**62. ¿Qué patrón GOF permite representar operaciones mediante objetos, facilitando llevar un registro de las operaciones realizadas e incluso poder deshacerlas?**

- **command**

**63. ¿Qué cualidad mide la capacidad de un sistema para soportar una mayor carga de trabajo?**

- **Escalabilidad**

**64. El patrón que permite gestionar la navegación entre distintas pantallas de una aplicación es**

- **Aplication controller**

**65. Qué clase de requisitos son los que condicionan la elección de una arquitectura concreta?**

- **los no funcionales**

**66. Con qué patrón de lógica de dominio se combina normalmente el patrón Data Mapper?**

- **Domain Model**

**67. Cuando es preferible usar el patron Domain Model?**

- **cuando la lógica de dominio es compleja**

**68. Qué patrón GOF se usa para definir familias de algoritmos intercambiables?**

- **Strategy**

**69. En un diagrama de clases de UML la composición es:**

- **Una relación sin ambigüedades que sugiere fuertemente una relación de todo/parte.**

**70. El objetivo del patrón Fachada es…**

- **Simplificar el acceso a la interfaz mediante funcionalidades**

**71. El patrón Command permite:**

- **Manipular comandos como objetos.**

**72. Si necesitamos proporcionar una interfaz para crear familias de objetos relacionados dependientes, sin especificar sus clases concretas. ¿Qué patrón debemos aplicar?:**

- **Patrón Abstract Factory**

**72. De los siguientes patrones, indica cuál de ellos es Creacional:**

- **El patrón Sigleton**

**Examen marzo 2015**

**73. Si nos hablan de TDD, BDD, MDD, etc. nos están hablando de...**

- **Métodos de desarrollo software**

**74. Un DSL ...**

**75. eL METAMODELO DE UN DSL define...**

**76. Cuando hacemos refactoring en nuestro código estamos haciendo...**

-

**77. La arquitectura software de un sistema es.**

- **A, B ??**

**78. ¿Cuál de las siguientes NO refleja un problema arquitectural?**

- **Un estudio de logs detecta una serie de accesos no autorizados al sistema**

**79.El patrón Transaction Script...**

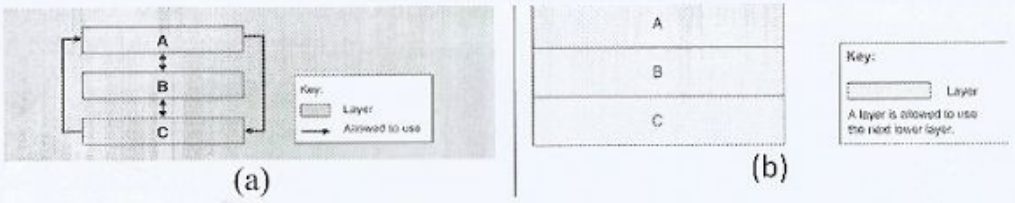
- **Organiza la lógica de negocio en procedimientos, donde cada procedimiento maneja una sola petición de la presentación**

**80. El patrón Domain Model (DM)...**

- **Se puede combinar para el acceso a la BD con un patrón Active Record si el DM es simple, o un patrón Data Mapper si es más complicado**

**81.**

9. Observa las siguientes figuras y elige la afirmación correcta

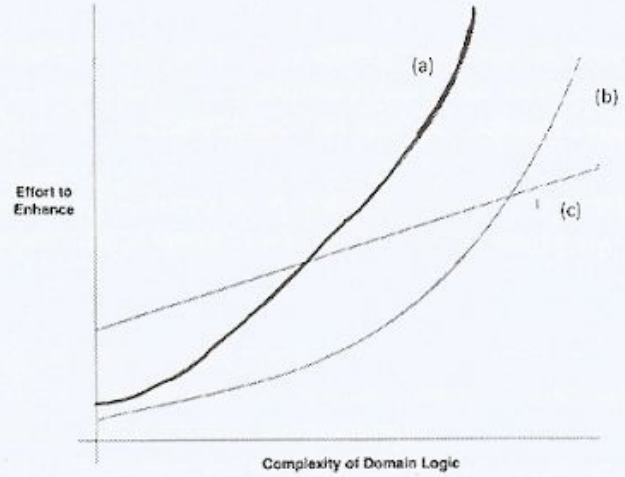



a) La Fig. a refleja una arquitectura de capas abierta, mientras que la Fig. b refleja una arquitectura de capas cerrada
(b) La Fig. a refleja una arquitectura de capas, mientras que la Fig. b no
× c) La Fig. a no refleja una arquitectura de capas, mientras que la Fig. b sí

- **A (coincide con las definiciones de abierta y cerrada)**

**82.**

12. ¿A qué patrón se corresponde cada línea en la siguiente figura (eje Y: esfuerzo de mejora; eje X: Complejidad de la lógica de Dominio)?




a) (a) Domain Model, (b) Transaction Script, (c) Table Module
b) (a) Domain Model, (b) Table Module, (c) Transaction Script
× c) (a) Transaction Script, (b) Table Module, (c) Domain Model

- **B (mirar tabla Domain Model, en el pdf de la lógica de negocio)**

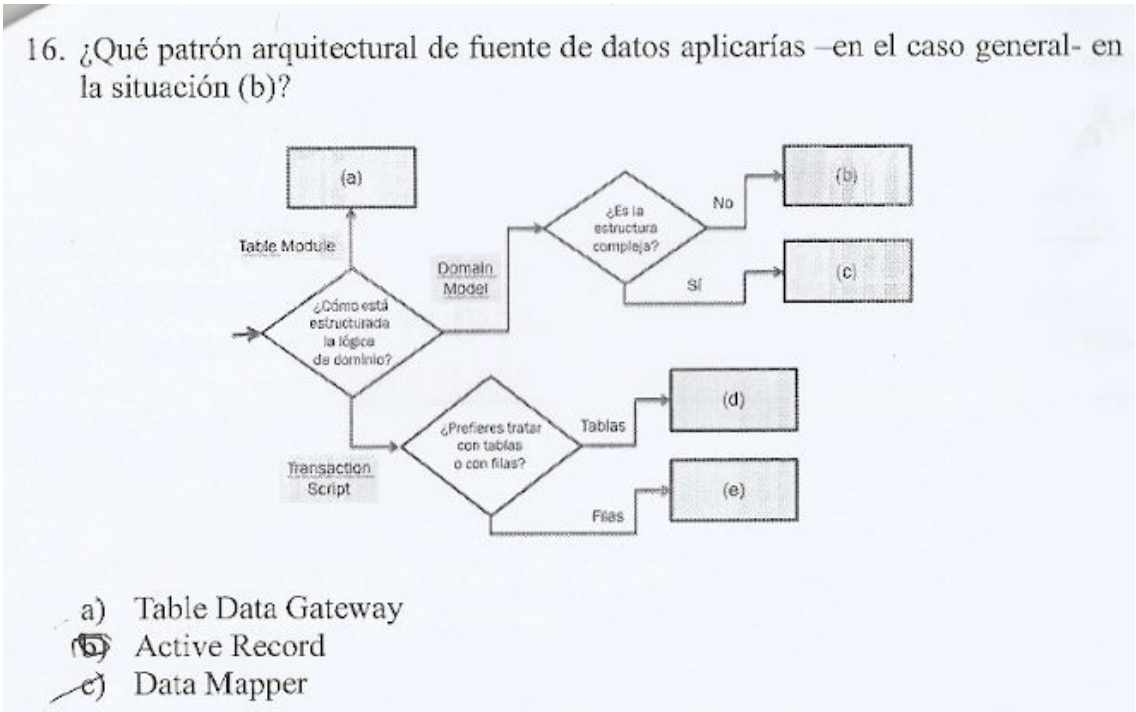
**83. El patrón Table Data Gateway...**

- **Mantiene toda la lógica de acceso a la tabla de la BD encapsulada en una sola clase**

**84. ¿Cual de las siguientes afirmaciones es cierta? En el patrón MVC...**

- **El controlador necesita acceder al modelo, lo que viola el patrón de separación en capas**

**85.**




- **(B) Active Record**

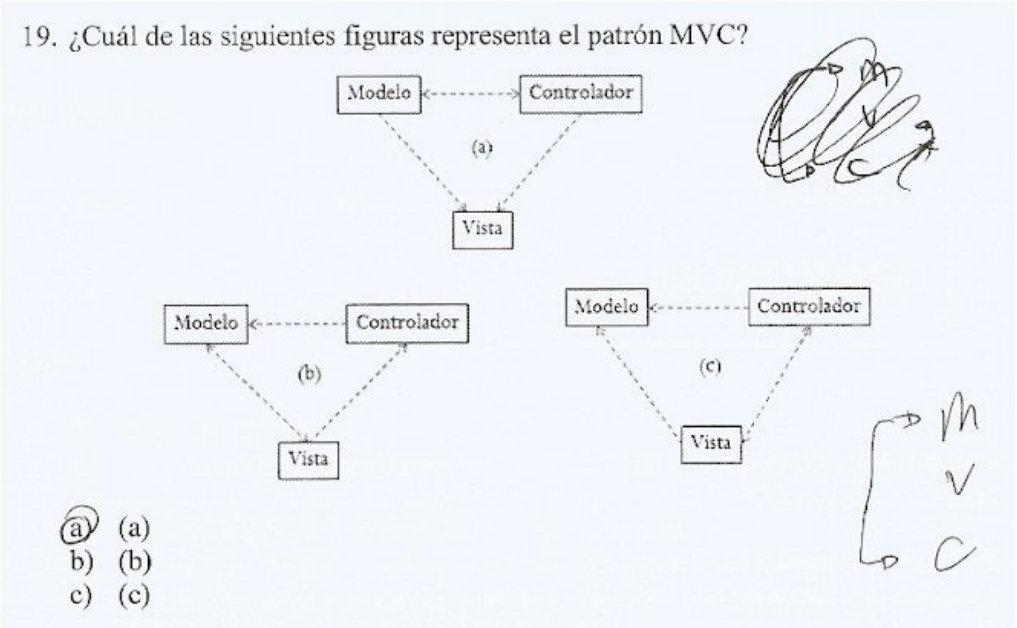
**86. El patrón Unit of Work se ocupa de que ...**

- **La integridad referencial sea preservada cuando se crean y modifican múltiples objetos dentro de una transacción de negocio o de manera concurrente, al mismo tiempo se limitan los accesos a la BD**

**87. ¿Cual de las siguientes afirmaciones NO es cierta? El controlador del patrón MVC en una aplicación Web ...**

- **Si adopta la forma de Controlador de Aplicación, delega tanto la decisión sobre qué lógica de dominio debe ejecutarse como la decisión sobre la vista que debe mostrar la respuesta**

**88.**




- **(C), el dibujo está en los apuntes Capa de Presentación**

**Examen julio 2015**

**89. Al usar el patrón Active Record, ¿dónde debe situarse el código para acceder a la base de datos?**

- **En las clases de dominio**

**90. ¿Qué patrón está pensado para poder intercambiar distintos comportamientos en tiempo de ejecución?**

- **Strategy**

**91. ¿Cual de los siguientes patrones NO es de comportamiento?**

- **Proxy**

**92.¿Qué inconveniente tiene el uso del patrón Composite?**

- **Resulta complicado imponer restricciones sobre la estructura de los objetos compuestos**

**93.¿Qué inconveniente tiene el uso de patrón Abstract Factory?**

- **No es fácil de añadir nuevos productos**

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

- **Proteger al sistema frente a posibles variaciones**

**95. En el patrón Command, ¿qué información necesitan los comandos para ejecutarse?**

- **Que clase es la receptora de la acción**

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

- **Proxy**

**97. En el patrón Observer,¿qué rol es el encargado de llevar un registro de los objetos que deban ser notificados cuando hay cambios?**

- **Subject**

**Examen Junio 2016**

**98. Cuándo conviene aplicar el patrón GRASP "Experto en Información" en cascada?**

- **Cuando queremos prevenir la aparición de nuevas dependencias entre clases**

**99. ¿Qué ventaja NO podemos conseguir con el patrón GOF "Proxy"?**

- **Proporcionar una implementación alternativa para un objeto**

**100. ¿Quién debe ser el encargado de crear los objetos de transferencia de datos (DTO)?**

- **Un objeto Assembler que tiene acceso a los objetos de dominio**

**101. ¿Qué inconveniente tiene el uso del patrón GOF "Composite"?**

- **Resulta complicado imponer restricciones sobre la estructura de los objetos compuestos**

**102. ¿Qué tipo de interfaces son deseables para llamadas entre objetos distribuidos?**

- **Interfaces de grano grueso**

**103. ¿Qué inconveniente tiene el uso de patrón GOF "Abstract Factory"?**

- **No es fácil de añadir nuevos productos**

**104. ¿Qué patrón es más recomendable para añadir nuevas funcionalidades a un objeto?**

- **Decorator**

**105. ¿Qué patrón permite reutilizar un mismo objeto de acceso a datos para distintas vistas?**

- **Model View Presenter**

**106. ¿En qué se basa el patrón GRASP “Polimorfismo”?**

- **Las instancias de clases hijas se pueden comportar como si se tratase de la clase padre**

**107. En una arquitectura de tipo Microkernel...**

- **se pueden añadir nuevas funcionalidades en tiempo de ejecución**

**108. En una arquitectura en capas abiertas ...**

- **las capas superiores pueden saltarse algunas de las capas inferiores**

**109. ¿Con qué patrón podemos recibir notificaciones cuando cambia el estado de un objeto?**

- **Observer**

**110. En el patrón GOF “Factory Method”, la clase Creator proporciona una funcionalidad genérica independientemente del tipo de producto que se quiera crear**

- **Verdadero**

**111. En el patrón GOF “Builder”, ¿que clase debe conocer la secuencia de pasos necesaria para construir un producto?**

- **La clase Director**

**112. ¿Qué beneficio obtenemos al usar patrones de software?**

- **Se simplifica la introducción de nuevas funcionalidades en el futuro**

**113. ¿Cuál de los siguientes tipos de acoplamiento es más fuerte?**

- **Cuando hay una jerarquía de herencia**

**114. ¿Con que otro patrón GRASP está relacionado el patrón “Creator”?**

- **Bajo acoplamiento, ya que disminuye el número de dependencias del sistema**

**115. ¿Qué patrón permite centralizar en una sola clase las comprobaciones comunes en una implementación del patrón MVC( seguridad, personalización, etc)?**

- **Front Controller**

**116. ¿En qué se diferencian un modelo de dominio y un diagrama de diseño de clases?**

- El diagrama de diseño de clases se deriva a partir del modelo de dominio

**117. ¿En qué caso está más indicado aplicar el patrón GRASP "Creator"?**

- Cuando queremos almacenar instancias del objeto creado

**118. ¿Cuándo es más necesario introducir una clase "Fabricación Pura"?**

- Cuando el aumento de responsabilidades de una clase pone en peligro su cohesión

**Examen 2017**

**119. ¿Quién debe ser el encargado de crear los objetos Data Transfer Object (DTO)?**

**120. ¿Qué patrón GOF permite limitar el número de instancias que se crean de una clase?**

- Singleton

**121. ¿Cuándo es preferible usar una clase Factoría en lugar de aplicar el patrón GRASP Creator?**

-

**122. ¿Que patrón GOF nos permitirá saber cuántas veces hemos accedido a un objeto?**

- Proxy

**123. ¿Cuál de las siguientes NO es una ventaja de usar un framework arquitectural?**

- Podemos cambiar fácilmente la arquitectura del sistema

**124. ¿Qué tipo de acoplamiento debemos evitar para facilitar los cambios de tecnología en la capa de presentación?**

- Que la capa de lógica de negocio tenga como dependencias clases de la capa de presentación

**125. ¿En qué situación NO podemos reemplazar una jerarquía de herencia por una solución distinta basada en composición?**

- Cuando necesitamos usar el polimorfismo

**126. ¿Qué patrón GOF permite implementar de forma sencilla la funcionalidad "deshacer"?**

- Command

**127. ¿Cuándo es necesario dividir una clase en dos o más clases distintas?**

- **Cuando la cohesión de la clase es demasiado baja**

**128. En el patrón State, ¿Qué clase es la encargada de decidir cuál es el siguiente estado cuando hay un cambio de estado?**

- **Solamente la clase <<Context>> puede tener esa información**

**129. Según el principio de inversión de dependencias...**

- **Los módulos de alto y bajo nivel deben depender de abstracciones**

**130. ¿Con qué patrón GRASP está relacionado el patrón GOF Facade?**

- **Controlador y bajo acoplamiento**

**131. En una arquitectura en capas, ¿por qué la capa de lógica de negocio se sitúa por encima de la capa de acceso de datos?**

- **Para desacoplar a las capas superiores de los detalles de acceso a la base de datos**

**132. ¿Qué objetos colaboran con el Front Controller para realizar comprobaciones de seguridad?**

- **Middleware**

**133. ¿Cuál es el principal problema del patrón “Class Table Inheritance”?**

- **Las operaciones join necesarias pueden afectar al rendimiento**

**134. ¿Qué problema se puede derivar de un excesivo acoplamiento en el diseño de un sistema?**

- **Los cambios en una clase pueden afectar a un gran número de clases distintas**

**135. ¿Qué técnica podemos usar para evitar acoplar una clase con la implementación concreta de una funcionalidad?**

- **Inyección de dependencia**

**136. ¿Qué técnica de diseño de interfaces consiste en realizar un diseño separado de un sitio web para dispositivos móviles?**

- **Adaptive web design**