

Asocia los conceptos a términos

1. Antes de realizar ninguna refactorización debemos tener un conjunto completo de ...  
**Tests unitarios**
2. Código duplicado **Código maloliente/sospechoso**
3. Hemos cometido un error en el diseño de la jerarquía de clases por entender mal el problema. Para corregirlo añadimos nuevas clases. **Cambios estructurales**
4. Sentencias 'switch' **Código maloliente/sospechoso**
5. Métodos muy largos **Código maloliente/sospechoso**
6. Teníamos un código probado y funcionando correctamente. Tras unos cambios en el código derivados por una petición de nuestro cliente hay cosas que han dejado de funcionar. **Error de regresión**
7. Atributos, métodos, variables no usadas **Código maloliente / sospechoso**
8. Para minimizar los errores y simplificar su corrección cuando hacemos cambios en mantenimiento, o bien lo refactorizamos, o bien añadimos una nueva funcionalidad, pero no ambas cosas al mismo tiempo. **Metáfora de los dos sombreros**
9. Clase con demasiados métodos y/o atributos **Código maloliente/sospechoso**

Indica cuáles de las siguientes son propiedades que deberían cumplir las pruebas unitarias.

Seleccione una o más de una:

- ☐ no es necesario que sean completas
- ☒ deben tener una cobertura del 100% del código que se prueba
- ☐ deben poder ejecutarse manualmente
- ☒ deben ser automatizables

Retroalimentación

Respuesta correcta

Enunciado de la pregunta

Indica cuáles de las siguientes son propiedades que deberían cumplir las pruebas unitarias.

Seleccione una o más de una:

- ☒ deben ser repetibles siempre en las mismas condiciones (por ejemplo, siempre con el mismo estado de persistencia)
- ☐ es recomendable que las diferentes pruebas unitarias dependan unas de otras para poder capturar posibles errores al integrar diferentes partes de una aplicación.
- ☐ deben poder repetirse en diferentes condiciones (por ejemplo, con diferentes estados de persistencia)
- ☒ deben ser independientes entre sí

Retroalimentación

Respuesta correcta

Enunciado de la pregunta

Indica en qué se ocupa la mayor parte del tiempo de desarrollo de una aplicación:

Seleccione una:

- ☐ a. en programar el código de la aplicación
- ☒ b. en encontrar los errores en el código de la aplicación
- ☐ c. en arreglar los errores del código de la aplicación

Retroalimentación

Respuesta correcta

Indica cuáles de las siguientes son propiedades que deberían cumplir las pruebas unitarias.

Seleccione una o más de una:

- ☒ cada prueba unitaria debería diseñarse para probar un único caso de uso de un único método.
- ☐ resultan muy útiles para capturar posibles errores de integración entre diferentes módulos de una aplicación.
- ☒ Las pruebas deben considerarse como parte esencial del desarrollo de software, equiparándolas en importancia al propio código para el cual se diseñan.
- ☐ las pruebas son una parte opcional del desarrollo de aplicaciones, lo importante es centrarse en el desarrollo del código que finalmente formará parte de la aplicación.

#### Retroalimentación

#### Respuesta correcta

Indica cuándo es conveniente refactorizar el código.

Seleccione una o más de una:

- ☐ cuando el código original es tan 'malo' (por diseño, múltiples errores de implementación, etc.) que merece la pena reescribirlo desde cero.
- ☒ cuando hay código maloliente.
- ☒ antes de añadir una nueva funcionalidad.
- ☐ una vez implementado y probado el código, poco antes de que se cumplan los plazos de entrega.
- ☒ al encontrar un error (bug) en el código

#### Retroalimentación

#### Respuesta correcta