**1. ¿Cuál es el objetivo del Desarrollo Guiado**

**por Pruebas?**

Eliminar el miedo en el desarrollo de aplicaciones

**2. ¿Qué significa TDD?**

Desarrollo Guiado por Pruebas de software o en inglés, Test-driven development

**3. ¿Qué ventaja tiene aplicar un desarrollo**

**guiado por pruebas?**

Generar un código puro, claro y limpio que funcione, evitando el código innecesario además de tener un flujo de información mayor de toda la gran cantidad de documentación, es decir, durante el proceso de creación del código resolvemos dudas que nos proporcionan una preciada documentación de lo que vamos implementado.

**4. ¿Qué es una prueba automatizada?**

Las pruebas automatizadas consisten en la aplicación de herramientas de software para automatizar el proceso manual de revisión y validación de un producto de software.

**5.¿Si se te pide realizar un programa que**

**realice operaciones matemáticas tales como**

**el factorial de un número x, multiplicación,**

**división etc, que es lo primero que realizas?**

Pensar en todos los errores que puede generar que uno o más variables hagan todo eso, después hacer un caso de prueba y codificar.

**6. ¿En que consiste el último paso del TDD?**

Arreglar el código, es decir, el código productivo se pasa a limpio y se perfecciona su estructura para evitar cosas duplicadas en el mismo y se vuelva más profesional. Comprobar que los test sigue pasando correctamente después de haber arreglado el código.

**7. ¿Qué consecuencias traerá el cambiar**

**parte del código?**

Puede afectar a la hora de manejar una base de datos, por ejemplo, una vez que se ha ejecutado una prueba, la base de datos puede quedar en un estado distinto al que se necesita para hacer la siguiente prueba. Además de perder la asimilada a la idea principal

**8. ¿Por qué las pruebas pueden parecer**

**tediosas?**

Porque a la hora de arreglar una cosa puedes romper otra.

**9. ¿Cuáles son los puntos mínimos**

**requeridos para realizar un caso de prueba?**

Hay por lo menos tres cosas que tenemos que escribir la prueba de: positivo, negativo y de excepción. Hay que pensar en la forma en la cual se debe desarrollar cierta tarea dentro de nuestro programa, es decir, pensar en el positivo. Después piensa en lo negativo, es decir, lo que podría salir mal y cómo debe comportarse el código. La excepción es pensar en la posibilidad de secuencia alternativa de eventos que podría suceder y cómo debe comportarse el código para dar cabida a aquellos.

**10. ¿Dónde debe quedar documentado los**

**resultados de los casos de prueba?**

El código de prueba no debe estar dentro del mimo método o clase ya sean públicos, privados o protegidos, pero si debe ser parte del mismo proyecto. En el caso de java debe existir un paquete que contenga el código de prueba.

**11. ¿Es conveniente que el código de**

**producción este junto con el código de**

**prueba?**

No, ya que el otro código podría funcionar como un respaldo además de evitar confusiones.

**12.Realiza y explica un diagrama donde se**

**muestren los pasos para aplicar el TDD en**

**Imagen que contiene dibujo

Descripción generada automáticamenteun Proyecto de software.**