

Alumno: José Arturo Reyes Roa

ID: 1116842

Materia: IDS347L - LABORATORIO TENDENCIAS EN DESARROLLO DE SOFTWARE

Docente: LORENZO SOLANO MARTINEZ

Requerimientos, Criterios, Casos

1.- ¿Qué es un Coding Dojo?

Coding dojo: es una reunión que hacen varios programadores, en la que se trabaja en conjunto con otras personas en un ámbito colaborativo, a las que se les presentará un problema de programación, y estos tendrán que resolverlos. Estas pueden llegar a durar todo el día, pero es más común que dure algunas horas. Esto se hace con el propósito de poder trabajar con otras personas que, por supuesto tienen diferentes puntos de vista, por ende, diferentes estilos de programación así se pueden aprender de los demás mientras se divierten.

2.- Diferencia entre Requerimientos, Criterios de Aceptación y Escenarios de Prueba. Dar ejemplos a partir de un problema distinto a la referencia.

Los requerimientos son las características que el programa debe cumplir, para poder satisfacer correcta y satisfactoriamente su propósito. Ahora, los criterios de aceptación son las declaraciones o condiciones que debe satisfacer el requerimiento del software para ser aceptado. Y finalmente el escenario de prueba son las formas en las que se pueden probar las funciones del requerimiento del programa desarrollado.

Ejemplo: Un conversor de sistemas numéricos, que se realizo para una tarea de otra materia.

Requerimiento:

- El conversor de sistemas numéricos debe poder recibir un numero decimal para luego convertirlo a sus equivalentes en los sistemas numéricos binario, octal y hexadecimal, para luego mostrarlos por pantalla.

Criterios de aceptación:

- Dado que se recibió un número del sistema numérico decimal, mostrará sus equivalentes en binario, octal y hexadecimal.
- Dado que el programa haya recibido algún dato no numérico, este debe de poder continuar su funcionamiento sin interrupción.

Escenario de prueba:

- Se introduce el numero 348, se devuelven los números, binario: 101011100, octal: 534, y hexadecimal: 15C.
- Se introduce el dato no numérico 15C, se devuelve el mensaje: "Se introdujo un dato no numérico, vuelva a intentar", a continuación, se le permitirá al usuario volver a introducir un dato.

3.- De dos ejemplos de requerimientos no-funcionales, y especifique cuál es su categoría (seguridad, performance, portabilidad, etc.)

- El código debe tener una notación Big O menor a 1000 o poder responder en menos de 1 segundo, para que el programa no sature el sistema. (Performance)
- La información que el programa use solo debe de ser accesible por personas con suficiente autoridad. (Seguridad)

4. ¿Qué es TDD?

El desarrollo guiado por pruebas (Test Driven Development), es una forma de desarrollo o programación en el que se crea un código lo más robusto posible, es decir, resistente a errores, para luego crear varios casos de prueba para cada funcionalidad del programa, que nos permita saber si el código está funcionando correctamente, y en caso contrario, si alguna prueba da un resultado erróneo, se corrigen esos problemas que se presentaron.

5. ¿Que son pruebas unitarias automatizadas?

Las pruebas unitarias son una etapa en la que se segmenta el código para para luego cada segmento ser probado individualmente su correcto funcionamiento. Cuando estas son automatizadas hace referencia al uso de un software que hace estas pruebas de forma automática.

6. ¿Cuál fue el 1er framework de pruebas unitarias y para cual lenguaje fue creado?

El primer framework de pruebas unitarias que se creo fue xUnit, que durante su creación tuvo como referencia los diseños de Kent Beck, y fue implementado inicialmente con el nombre SUnit en Smalltalk que es un lenguaje reflexivo de programación.

7. ¿Describa los componentes de la arquitectura xUnit?

Caso de prueba: se preparan cuales serán los datos de entrada, al igual que el estado ambiental, para luego llamar al programa para ser probado con esos contextos y ver si devuelven los resultados esperados.

Aserciones: estas son afirmaciones que validan si las salidas fueron las esperadas. Normalmente existe un marco de trabajo que define las mismas.

Conjunto de pruebas: son varios casos de pruebas relaciones que al estar bajo el mismo marco de trabajo permite reutilizar el mismo código para cada uno de ellos.

Configuración de prueba: este es el entorno en el que se van a ejecutar las pruebas, y posee características que permiten realizar las pruebas de forma consecutiva, es decir, permite reiniciar el código a un estado base para volver a probar.

Ejecutor de pruebas: este es el programa que ejecuta las pruebas en sí, y dice si la prueba fue exitosa o no, en varios formatos, ya sea texto legible humano, como textos que pueden leer otro tipo de software.

8. Indique al menos tres desventajas de las pruebas unitarias automatizadas

- No aseguran verdaderamente que el código este libre de errores, ya que estan diseñados para probar lo que ya se conoce, y no la forma en la que el usuario final haría.

- Si se define mal como será la prueba y por ende su programación, puede tirar falsas alarmas de error.
- Al ser programas, estos también requieren mantenimiento, lo cual se traduce a más tiempo gastado.

9. Indique al menos tres ventajas de las pruebas unitarias automatizadas

- Permite probar una gran variedad de casos que un tester manual pudo no haber pensado.
- Vuelve más eficiente esta etapa, ya que de forma manual es muy lenta.
- Son fáciles de implementar, lo que permite que su realización sea más económica.

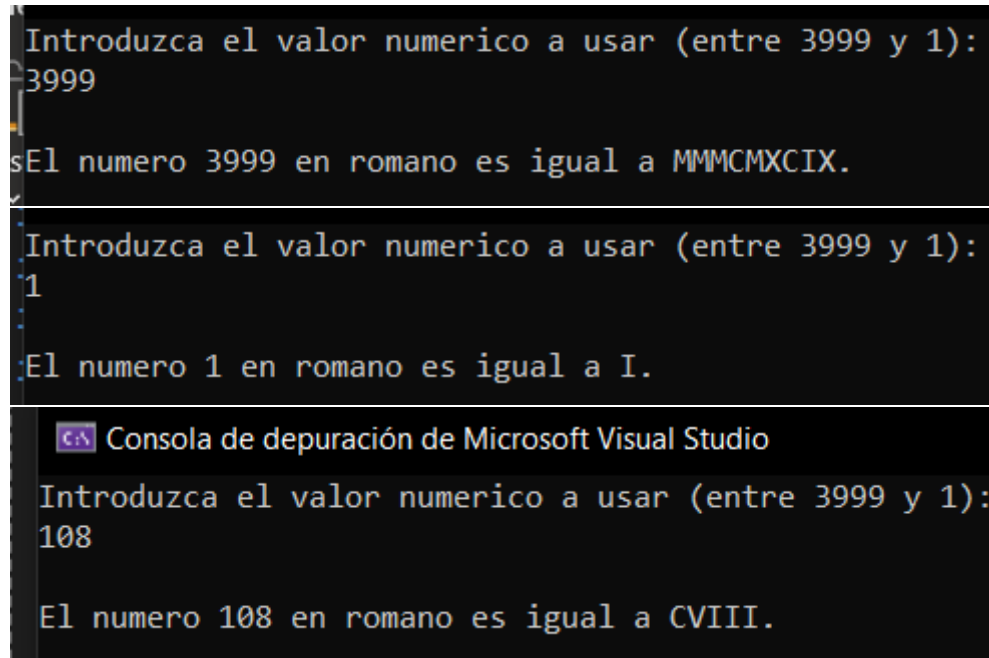
10. Tomando el algoritmo de conversión de números arábigos o "decimales" a números Romanos:

Ver el documento Read me

11. Utilizando el lenguaje de su preferencia cree cinco o más casos de prueba unitarios automatizados utilizando un framework de automatización de pruebas para el algoritmo en cuestión

Pruebas Unitarias Manuales

Casos de prueba del requerimiento 1



```

Introduzca el valor numerico a usar (entre 3999 y 1):
3999
El numero 3999 en romano es igual a MMMCMXCIX.

Introduzca el valor numerico a usar (entre 3999 y 1):
1
El numero 1 en romano es igual a I.

Consola de depuración de Microsoft Visual Studio
Introduzca el valor numerico a usar (entre 3999 y 1):
108
El numero 108 en romano es igual a CVIII.

```

Introduzca el valor nuerico a usar (entre 3999 y 1):
4000

El dato que usted ha introducido no es valido, vuelva a intentar.

Introduzca el valor nuerico a usar (entre 3999 y 1):

Introduzca el valor nuerico a usar (entre 3999 y 1):
toif

El dato que usted ha introducido no es valido, vuelva a intentar.

Introduzca el valor nuerico a usar (entre 3999 y 1):
_

Pruebas Unitarias Automatizadas

