

**UTN.BA** FACULTAD  
REGIONAL  
BUENOS AIRES  
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

# Centro de e-Learning

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // [e-learning@sceu.frba.utn.edu.ar](mailto:e-learning@sceu.frba.utn.edu.ar)



[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)



**UTN.BA** FACULTAD  
REGIONAL  
BUENOS AIRES  
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

**Centro de  
e-Learning**

# Professional Testing Master



## **Presentación:**

En esta Quinta Unidad del curso, completaremos el conocimiento sobre testing con conceptos, metodologías y técnicas complementarios de gran relevancia en el mercado actual así como herramientas que facilitan la tarea del tester.

# **Unidad 5:**

## **Otros conceptos de testing**

## Objetivo:



### Al terminar la Unidad los participantes:

- ☐ Se habrán familiarizado con los diferentes tipos de pruebas que se realizan en distintos momentos del desarrollo.
- ☐ Se habrán introducido en los conceptos fundamentales de planificación de test.
- ☐ Conocerán herramientas que ayudan con la tarea de test, incluyendo automatización de pruebas y nuevas metodologías de desarrollo y testing de software.
- ☐ Conocerán las características de las certificaciones de testing más importantes a nivel internacional.

# Contenido

- 1. Pruebas: regresión, humo, alfa, beta, usabilidad**
- 2. Planificación de test**
- 3. Automatización y herramientas**
- 4. Test Driven Development y Programación Extrema**
- 5. Certificaciones**

# Prueba de regresión



Consiste en un subconjunto de pruebas que se ejecutan cada vez que se produce un cambio en el software o una corrección de un error.

Estas pruebas también se ejecutan cada vez que se integra un nuevo módulo.

El objetivo es verificar que las nuevas funcionalidades no afecten aspectos del software que ya se habían probado y funcionaban correctamente.

# Prueba de humo



La prueba de humo (smoke test) es una estrategia de integración continua.

El software se vuelve a construir con nuevos componentes y se prueba todos los días.

La prueba no es exhaustiva, sino que se intenta detectar errores paralizantes.



## Prueba de humo (cont.)



### Ventajas

- ☐ Se minimiza el riesgo de integración
- ☐ La calidad del producto final mejora
- ☐ El diagnóstico y la corrección de errores se simplifican
- ☐ El progreso es más fácil de evaluar

# Pruebas alfa y beta



Las pruebas de aceptación permiten al cliente validar todos los requerimientos.

Si el software es un producto a ser usado por muchos clientes, no es práctico realizar pruebas de aceptación formales con cada uno.

La prueba alfa se lleva a cabo en las oficinas del desarrollador por un grupo representativo de usuarios finales con la supervisión del desarrollador de software.

## Pruebas alfa y beta (cont.)



La prueba beta se realiza en una o más ubicaciones del usuario final.

A diferencia de la prueba alfa, el desarrollador no está presente.

El cliente registra los problemas encontrados y los reporta periódicamente.

# Prueba de usabilidad



Evalúa la facilidad de uso del sistema.

Se debe tener en cuenta:

- ☐ Perfil del usuario final
- ☐ Resultados claros
- ☐ Mensajes de error comprensibles
- ☐ Interfaz consistente
- ☐ Opciones no sobreabundantes
- ☐ Feedback instantáneo a las acciones del usuario

# Planificación



El plan debe contemplar principalmente:

- ☐ Objetivos
- ☐ Criterios de finalización
- ☐ Cronogramas
- ☐ Responsabilidades
- ☐ Estándares
- ☐ Herramientas

(sigue)

## Planificación (cont.)



El plan debe contemplar principalmente:

- ☐ Tiempo de computadora
- ☐ Configuración de hardware
- ☐ Integración
- ☐ Procedimientos de seguimiento
- ☐ Procedimientos de depuración
- ☐ Pruebas de regresión

# Automatización



Problema: miles de casos de prueba deben ser ejecutados en un tiempo limitado

Solución: test automatizado

Ventajas

- ☐ Velocidad
- ☐ Eficiencia
- ☐ Precisión
- ☐ Resistencia

# Herramientas para testing



## Clasificación

- ☐ Herramientas para planificación y administración de pruebas
- ☐ Herramientas para diseño de casos de prueba
- ☐ Monitores y visores
- ☐ Drivers
- ☐ Stubs
- ☐ Herramientas de carga y stress
- ☐ Herramientas para automatización



# Monitores y visores



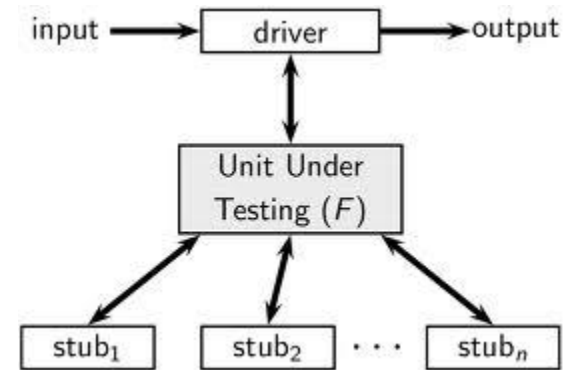
Permiten ver detalles del funcionamiento del software que normalmente no se podrían ver.

Permiten hacer tests de caja blanca.

Ejemplos de este tipo de herramientas son:

- ☐ Analizadores de protocolo
- ☐ Debuggers

# Drivers

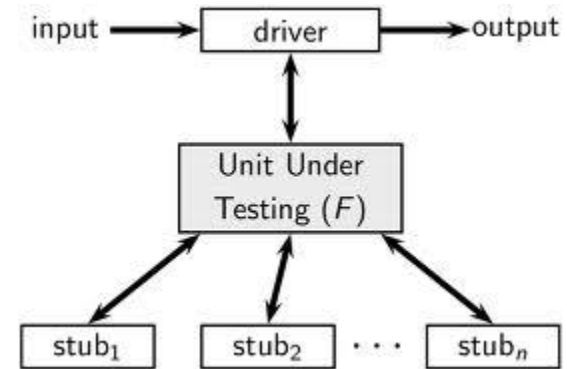


Son herramientas que controlan y ejecutan el software que está siendo probado.

Ejemplos de este tipo de herramientas son:

- ❑ Scripts (ej. Perl, Python, Groovy, PowerShell)
- ❑ Schedulers (ej. cron, windows scheduler)
- ❑ Software simulando entradas de mouse y teclado

# Stubs

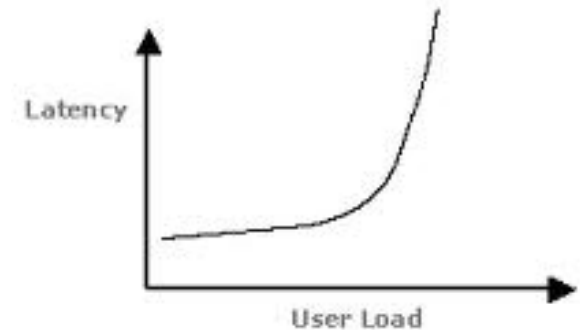


Son lo contrario de los drivers. No controlan la ejecución del software a probar, sino que responden a datos enviados por éste.

Además proveen al tester un modo de visualizar e interpretar los datos recibidos.

Un ejemplo sería un software que simule el comportamiento de una impresora y analice o reporte los datos recibidos.

# Herramientas de carga y stress



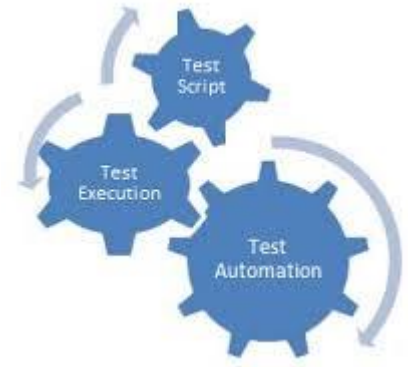
## ☐ Herramientas de stress:

- ☐ Permiten especificar la cantidad de memoria, espacio en disco y otros recursos a utilizar por un programa que se está probando.

## ☐ Herramientas de carga:

- ☐ Permiten especificar los parámetros de carga de una aplicación, como por ejemplo cantidad de accesos y usuarios simultáneos en una aplicación web

# Herramientas para automatización



## Clasificación

- ☐ Grabación y reproducción de macros
- ☐ Macros programables
- ☐ Herramientas de testing totalmente programables
- ☐ Herramientas de testing aleatorio

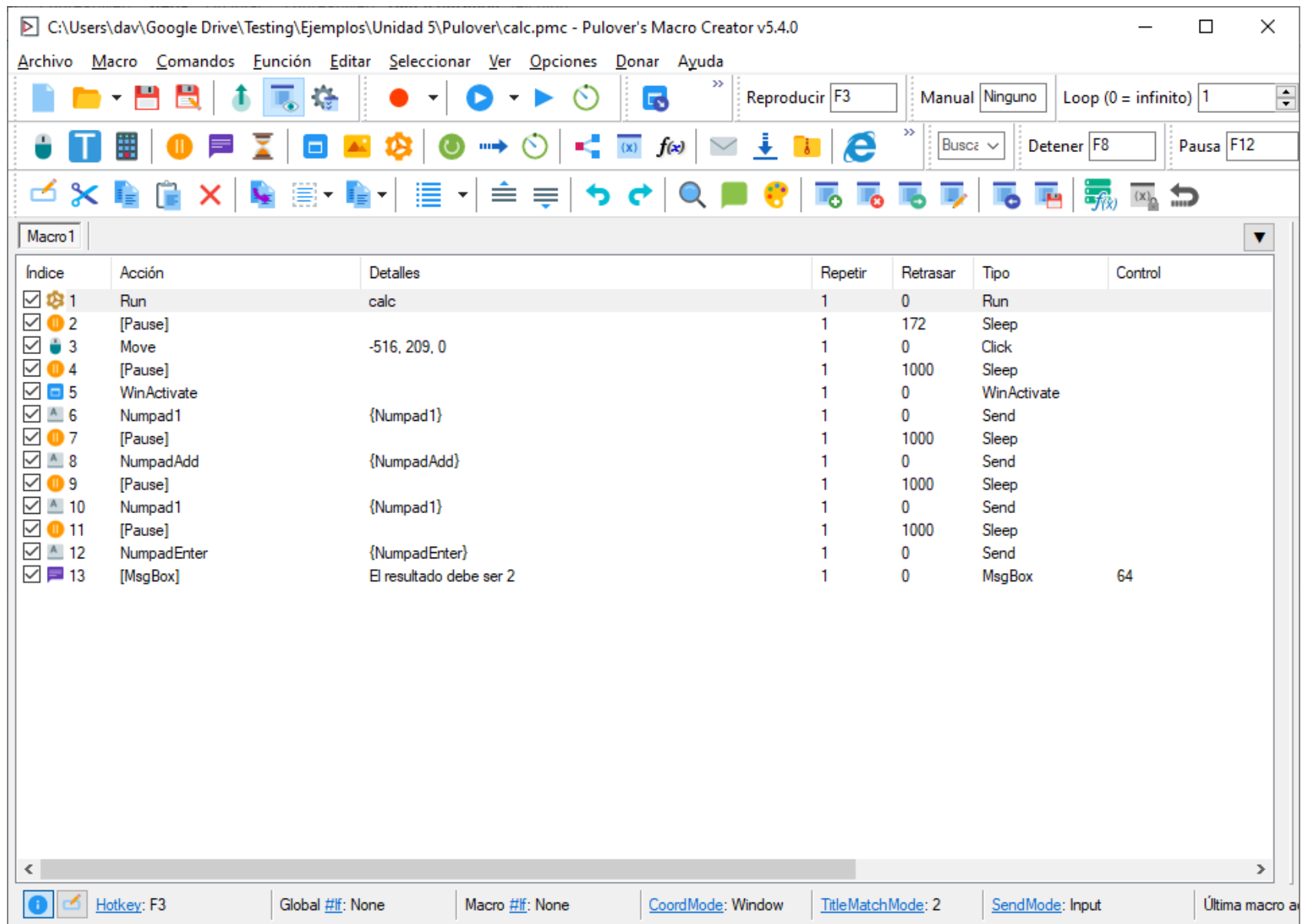
# Herramientas de grabación y reproducción de macros

Permiten grabar las acciones de mouse y teclado que realiza el tester y luego reproducirlas automáticamente.

Se puede configurar la cantidad de veces que se repite la macro y la velocidad.

Principales desventajas

- ☐ Pueden tener problemas de sincronización
- ☐ No permiten verificación



Herramienta Pulover's Macro Creator para Windows

# Macros programables

Se generan en forma de código, aunque no es necesario escribirlo manualmente.

Permiten sincronizarse al poder esperar por eventos como aparición de determinadas ventanas.

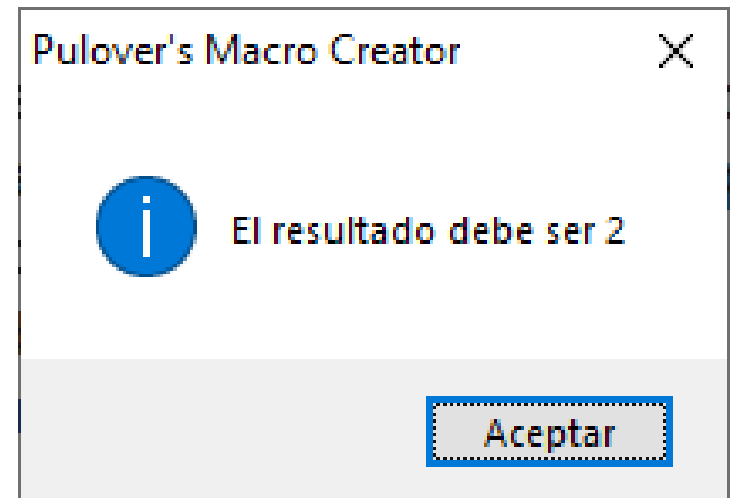
También pueden detener la ejecución mostrando un mensaje para que el usuario confirme si determinado resultado es válido.

Desventaja: No permiten verificación



## Macros programables (cont.)

```
1 Run, calc
2 WinActivate, Calculadora
3 Sleep, 1000
4 Send, {Numpad1}
5 Sleep, 1000
6 Send, {NumpadAdd}
7 Sleep, 1000
8 Send, {Numpad1}
9 Sleep, 1000
10 Send, {NumpadEnter}
11 Sleep, 1000
12 MsgBox, 64, , El resultado debe ser 2
13 WinClose, Calculadora
```



# Macros totalmente programables

Combinan la capacidad de macros de simular acciones con la potencia de un lenguaje de programación.

De esta forma permiten tomar decisiones y verificar los resultados mediante:

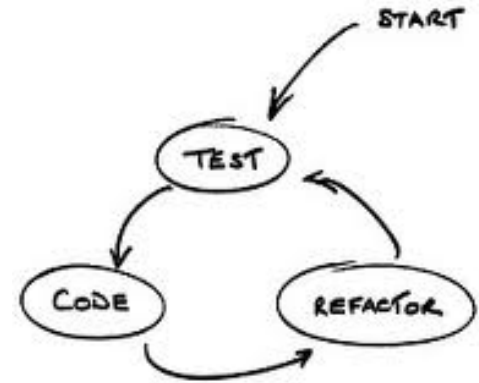
- ☐ Capturas de pantalla totales o ciertas porciones o rasgos
- ☐ Valores de control (de ciertos elementos en la ventana)
- ☐ Archivos o dispositivos de salida (red, modem etc.)

# Herramientas de testing aleatorio

Estas herramientas se conocen como monkeys (monos) ya que se basan en el concepto de millones de monos escribiendo durante millones de años.

De esta forma, al realizar una gran cantidad de acciones al azar durante algún tiempo podrían descubrir errores inimaginables.

# Test Driven Development (TDD)

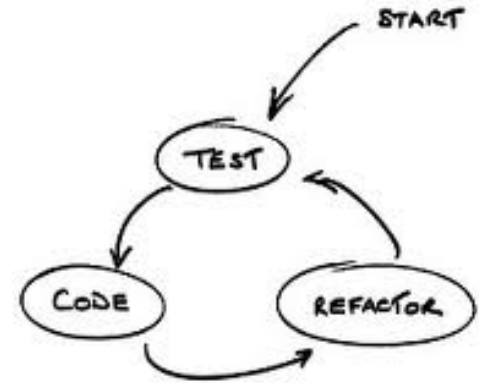


En Test Driven Development (TDD, Desarrollo Basado en Pruebas), los requerimientos para un componente de software son la base para la creación de casos de prueba, que se crean antes de escribir el código fuente.

Cuando una prueba falla se corrige el código y se repiten todas las pruebas, hasta que todas pasen.

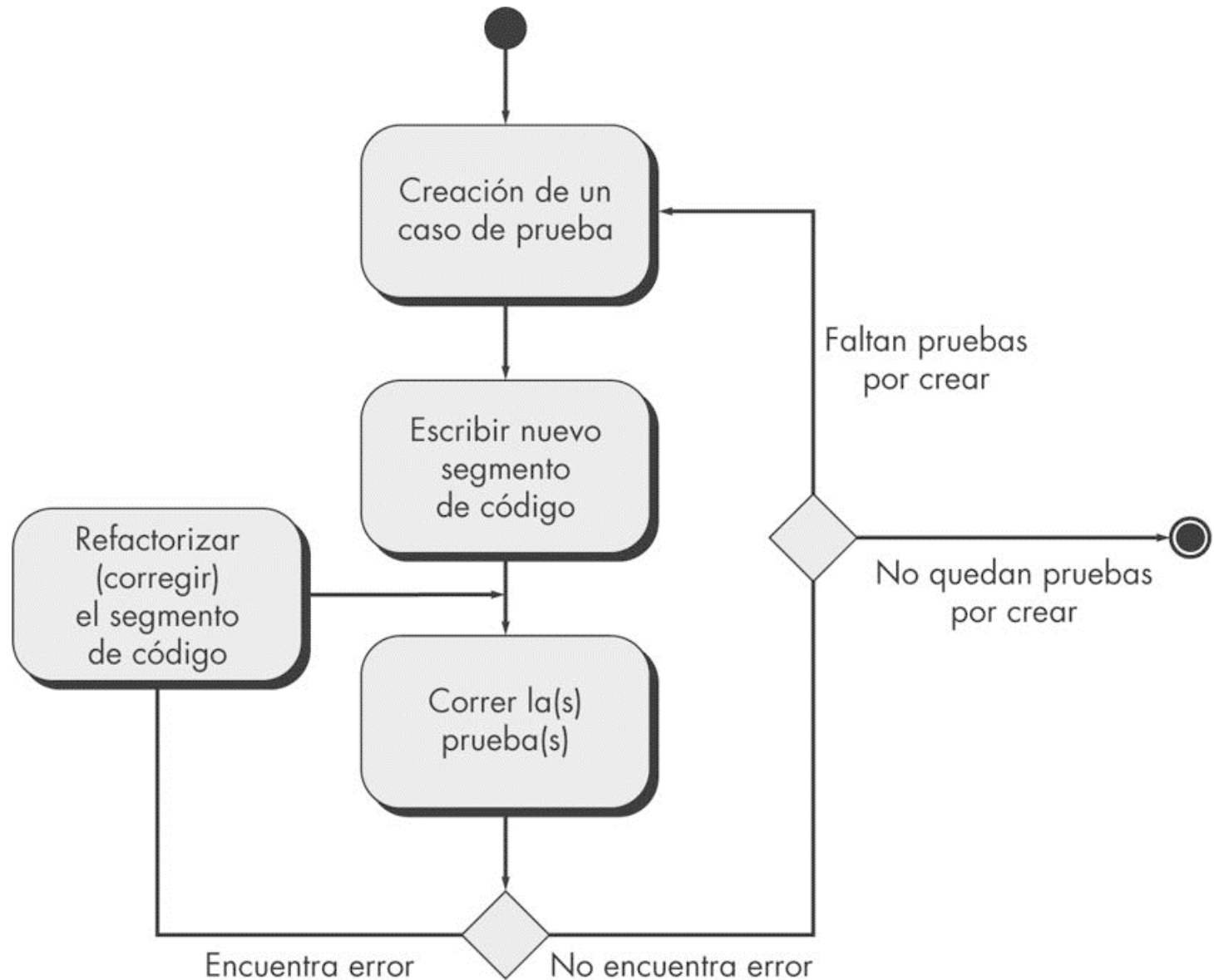
Luego se agregan nuevas pruebas y se agrega el código fuente para que el componente las pase.

## Test Driven Development (cont.)



El código se desarrolla en incrementos muy pequeños (una sub función a la vez) y no se escribe código hasta que exista una prueba que lo verifique.

La gran ventaja es que se garantiza que el código pasa las pruebas, ya que se escribe para que lo haga.



Flujo del proceso TDD

# Programación Extrema (XP)



Es una metodología de desarrollo ágil de software que implica el uso de TDD.

Funciona bien en proyectos pequeños a medianos donde los cambios son frecuentes.

Requiere comunicación fluida con los clientes.

Su modelo de testing se denomina Testing Extremo (XT)

# Testing Extremo (XT)

Se basa en

- ☐ Pruebas unitarias

- ☐ Se realizan a nivel clase o métodos de clase

- ☐ Pueden ser automatizadas

- ☐ Pruebas de aceptación

- ☐ Determinan si la aplicación cumple con los requisitos como funcionalidad y usabilidad

- ☐ Las llevan a cabo los clientes



# Certificaciones para testing

Dos de las instituciones de certificación más importantes a nivel internacional son:

- ☐ International Institute of Software Testing® (IIST®)
- ☐ International Software Testing Qualifications Board® (ISTQB®)



# Certificaciones más importantes del IIST

- ☐ Certified Software Test Professional – Associate Level (CSTP-A)
- ☐ Certified Software Test Professional – Practitioner Level (CSTP-P)
- ☐ Certified Software Test Professional – Master Level (CSTP-M)
- ☐ Certified Test Manager (CTM)
- ☐ Certified Software Test Automation Specialist (CSTAS)

# Certificaciones más importantes del ISTQB

- ☐ Foundation Level
- ☐ Advanced Level
  - ☐ Technical Test Analyst
  - ☐ Test Analyst
  - ☐ Test Manager
- ☐ Expert Level



## Consignas para discutir en el foro:

Reflexione a partir de las siguientes preguntas:

- ☐ ¿En qué casos tiene sentido utilizar pruebas alfa y beta?
- ☐ ¿Provea ejemplos de aspectos de testing que se pueden automatizar y otros que no.
- ☐ Con sus palabras, explique las ventajas de metodologías como TDD y XP.
- ☐ ¿Por qué son importantes las certificaciones en testing?

Vierta sus respuestas en el foro y establezca un intercambio de opiniones con sus compañeros de curso.

**Esperamos hayan  
disfrutado y aprovechado  
del estudio y las  
actividades propuestas en  
esta unidad!!!!!!!!!!!!**



**UTN.BA** FACULTAD  
REGIONAL  
BUENOS AIRES  
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

# Centro de e-Learning

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // [e-learning@sceu.frba.utn.edu.ar](mailto:e-learning@sceu.frba.utn.edu.ar)



[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)