

# *Ingeniería de Software*

Unidad N°  
Testing

Juan José Vanzetti



- Calcular

$$850 * 77.1 =$$

$$1700 * 38,55 =$$

$$425 * 154,20 =$$



¿Por qué deben hacerse las pruebas?

- En sistemas críticos la falla del sistema puede traer como consecuencia pérdidas de vidas humanas y/o monetarias, y repercusiones económicas (incluso penales) muy severas.



Que se dice en el mercado si no hay pruebas adecuadas

- **Consorcio de Informática Sostenible** reportó que podían existir tanto **como 20 o 30** fallos o bugs por cada **1.000** líneas de código en la mayoría de las aplicaciones de software.
- El **Consorcio Cutter** comentó que de las firmas de software que encuestaron,
  - el **32 %** admitió haber lanzado software con una gran cantidad de defectos
  - el **38%** dijo que su software carecía de un programa de garantía adecuado,
  - y el **27 %** comentó que no realizaban revisiones formales de calidad de su software



Que se dice en el mercado si no hay pruebas adecuadas

## **Impacto económico de una infraestructura inadecuada para testing de software. NIST mayo de 2002**

- El total de las ventas del año 2000 fueron de \$ 180 mil millones de dólares.
- Los bugs de software cuestan \$59,5 mil millones de dólares anuales a la economía de EE.UU.
- La mitad es afrontada por los usuarios y la otra mitad por los constructores.



- **Un poco de historia:**
  - Inicialmente , el testing era debugging.
  - En los '80... aparecen estándares.
  - En los '90... aparecen las herramientas.
  - En nuestros días, aún es un campo inmaduro.



No!

es depurar código.

es simplemente verificar que las funciones del software se implementen.

es demostrar de que no hay errores.



- ES ejercitar el software para detectar errores.
- ES detectar desviaciones desde la especificación.
- ES ejercitar el software para verificar que satisface los requerimientos funcionales y no funcionales.
- ES someter un software a ciertas condiciones que puedan demostrar si es o no válido, a los requerimientos planteados.
- ES verificar que el software se ajusta a los requerimientos y además validar que las funciones se implementan correctamente.
- Es agregar valor a un producto, y no solamente al producto, sino al proceso de desarrollo, siempre que se consideren los productos generados.





*“...el proceso de ejercitar el software para detectar errores y verificar que satisface los requisitos funcionales y no-funcionales especificados.”*

El testing es una forma de medir la calidad.



## Visión más apropiada del Testing

- Proceso DESTRUCTIVO que trata de encontrar defectos, cuya presencia se asume en el programa.
- El tester toma una actitud negativa para demostrar que algo es incorrecto.



## ¿Qué es Software Testing?

- Testing es el proceso de establecer confianza en que el programa o sistema hace lo que se supone que debería hacer. **Hetzel, 1973**
- Testing es el proceso de ejecutar un programa o sistema con la intención de encontrar errores. **Myers, 1979**
- Testing es cualquier actividad enfocada hacia la evaluación de un atributo o capacidad de un programa o sistema para determinar que cumple con los resultados esperados. **Hetzel, 1983**
- La medida de la calidad en el software. Hetzel, 1988

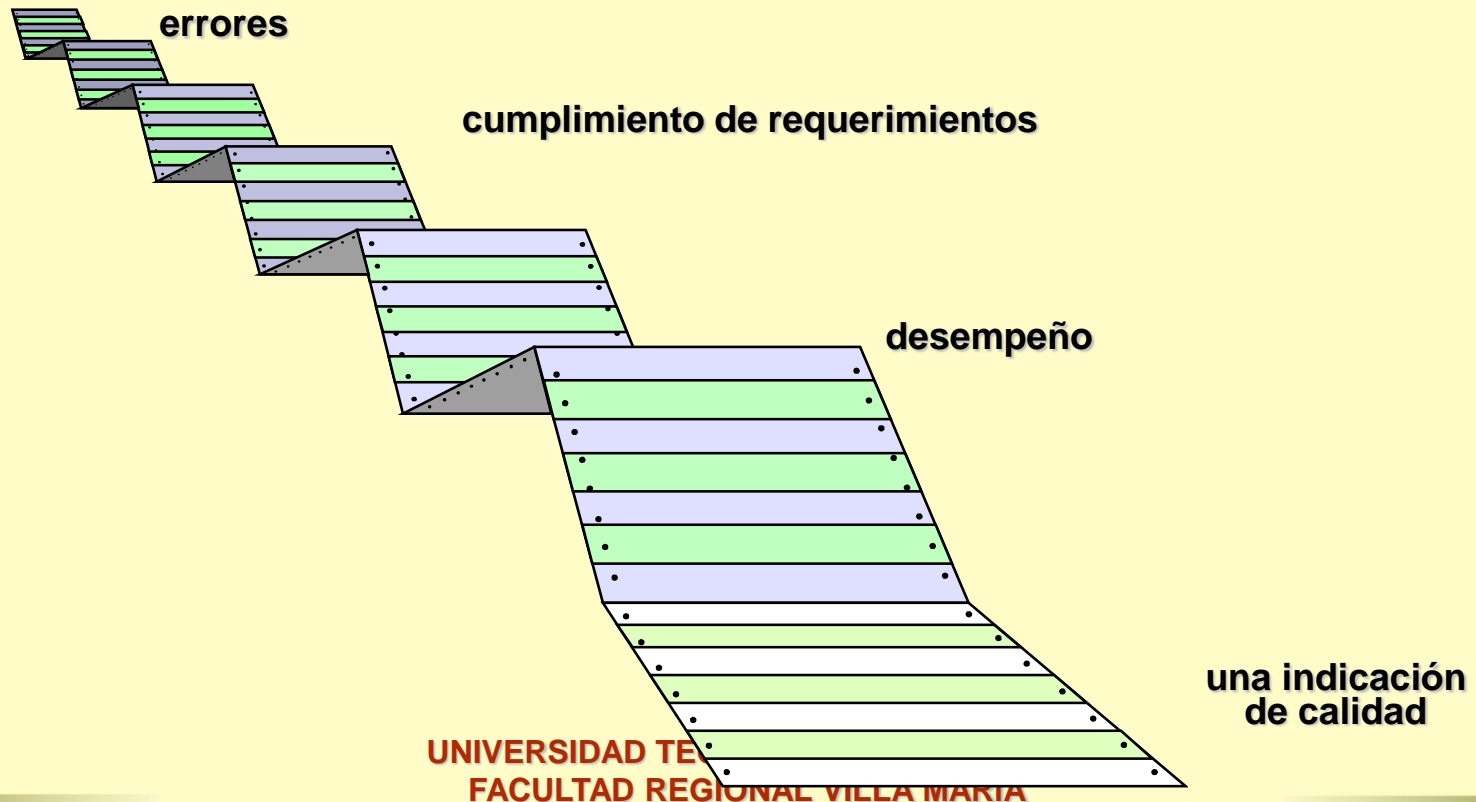


## Qué es Software Testing para el Practicante

- Chequear el programa contra las especificaciones.
- Encontrar errores en el programa.
- Determinar el grado de aceptabilidad para el usuario.
- Asegurarse de que un sistema está listo para usarse.
- Ganar confianza de que el programa funciona.
- Mostrar que un programa funciona correctamente.
- Demostrar que los errores no están presentes.
- Entender los límites del rendimiento.
- Aprender lo que el sistema no puede hacer.
- Evaluar las capacidades de un sistema.
- Verificar la documentación.
- Convencerse a uno mismo de que el trabajo ya está terminado.



- La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión final de los requerimientos, del diseño y de la codificación



## La Importancia de la Definición de Testing

- **La mayoría de la gente tiene una visión incorrecta de lo que es testing, y esa es la principal causa de una mala prueba del software.**
- Si la meta es demostrar que el programa no tiene errores, entonces subconscientemente estamos atraídos hacia esa meta y tendemos a seleccionar datos de prueba que tengan una baja probabilidad hacer que el programa falle.



## Concepto de testing

- Cualquier actividad dirigida a evaluar un atributo de un sistema.
- Una medida de la calidad del software.
- El proceso de evaluar un programa o sistema.
- Confirmar que un sistema ejecuta la funcionalidad pretendida correctamente.
- El proceso de ejecutar un programa utilizando pruebas construidas expresamente para asegurar que la conducta exhibida es la esperada.



- Note las diferencias
  - El proceso de encontrar
  - El proceso de evaluar
  - El proceso de medir
  - El proceso de mejorar
- Por un cuarto de siglo, como testers, se han enfocado en el proceso interno del testing y en general hemos omitido su verdadero propósito.





## **El verdadero propósito de testing es: Generar Información**

*“La principal razón por la que los testers existen es para proveer información que pueda ser utilizada por otros y éstos la utilicen para generar cosas de valor “.*

*James Bach*



**Concepto de testing:** Testing es una disciplina profesional que requiere personas capacitadas y entrenadas. Edward Kit

IEEE 829-1998- Standard for Software Test Documentation

El proceso de analizar un ítem de software para detectar la diferencia entre las condiciones existentes y las condiciones requeridas( bug) y evaluar las características de los items del software.

¿ Qué es probar ? ANSI-1990-Std 610.12

El proceso de ejecutar un sistema o componente bajo condiciones específicas, observando o registrando sus resultados, y realizando una evaluación de algún aspecto del sistema o componente.



**Falla ( failure)**

- Diferencia entre los resultados esperados y los reales.
- Una desviación en la respuesta, servicio o resultado esperado de un componente o sistema.
- Ocurre cuando un programa no se comporta adecuadamente.
- Representa una propiedad del programa en ejecución.

**Defecto (fault) (bug ) :**

- Si se ejecuta el defecto puede causar una falla.
- El desperfecto en un componente o sistema, causa que el componente o sistema falle al ejecutar su función requerida.
- Puede haber defecto y el sistema no produce una falla.

**Error(mistake, error) :**

- Una acción humana que produce resultado incorrecto.
- Un error lleva a uno o más defectos, que están presentes en un producto de software.
- Un error puede ocurrir en cualquier momento, durante el ciclo de desarrollo.
- Un error ocasiona 0, una o más fallas.



Un **ERROR**  
GENERA  
0,1 o más **DEFECTOS**  
GENERA  
0,1 o más **FALLAS**



## Ejemplo de Error, Defecto y Falla

```
int doblar (int param ) {  
    int resultado;  
    resultado = param * param;  
    return (resultado );  
}
```

- Una llamada a doblar (3) retoma 9
- El resultado 9 representa un falla
- Esta falla se debe a un defecto en la línea 3
- El error es tipográfico



- Testing muestra la presencia de defectos
- Testing puede mostrar la presencia de defectos, pero no puede probar que no hay defectos.
- Testing reduce la probabilidad de existencia de defectos no descubiertos en el software.
- No es una prueba de correctitud si no se encuentran defectos.



- **¿Cuánto tiempo tomaría el testing exhaustivo del siguiente programa?**

`imprimirSuma (int a, int b)`

- $2^{32} \times 2^{32} = 2^{64} = \text{aprox } 1,8 \times 10^{19} \text{ pruebas}$
- Asumiendo 1 nanosegundo por prueba (  $10^9$  pruebas/segundo)
- Tenemos  $1,8 \times 10^{10}$  segundos....

**Son aproximadamente 600 años**



- El Testing :
- **NO** puede asegurar
  - **NI** calidad en el software
  - **NI** software de calidad





## **Testing es dependiente del contexto**

- Testing se realiza en forma diferente en contextos diferentes.
- Por ejemplo, un sistema crítico se prueba diferente de un sitio de comercio electrónico



## **Referidos a quién realiza la tarea:**

- Un programador debería evitar probar su propio código.
- Un equipo de programación no debería probar sus propios desarrollos.
- El Testing es una disciplina profesional que requiere gente entrenada y competente.
- Se debe cultivar una actitud de equipo positiva para la destrucción creativa.



## **Referidos a la tarea en general:**

- El testing es una tarea extremadamente creativa e intelectualmente desafiante.
- Nunca planificar el esfuerzo de prueba suponiendo que no se encontrarán defectos.
- Ni en los casos más triviales es posible probar un programa para encontrar todos los defectos.



## Principios de la prueba

1. A todas las pruebas se les debería poder hacer un seguimiento hasta los requerimientos del cliente/usuario. El error mas grave desde el punto de vista del cliente son aquellos que impiden al programa cumplir con sus requerimientos.
2. Las pruebas deberían planificarse antes de que empiecen. La planificación de las pruebas pueden empezar cuando este completo el modelo de requerimientos. (antes de generar código)
3. Las pruebas deberían empezar de lo individual a lo general.
  - A) Módulos individuales
  - B) Grupos de módulos integrados
  - C) Sistema total
4. Para ser mas efectiva las pruebas deberían ser conducidas por un equipo independiente.. El equipo que creo el sistema no es el mas adecuado para llevar acabo las pruebas



## Testing Dinámico

- Pruebas que implica la ejecución del software de un componente o sistema.
- Ej. pruebas unitarias sobre un componente

## Testing Estático

- El proceso de evaluar un componente o un sistema sin ejecutar el objeto de prueba
- Ej. Revisiones o análisis estático de código.

*Nota: Algunos autores indican que el testing estático no es realmente testing, pero este proceso es generalmente llevado a cabo por el área de testing.*



**Un propósito de testing es encontrar defectos**  
**Encontrar defectos destruye la confianza**

**ENTONCES**

**Un propósito de testing es destruir la confianza**

**PERO**

**Otro propósito de testing es dar confianza**



# CONCLUSIÓN

**“La mejor forma de dar confianza es destruyéndola “**

