



Tester de Aplicaciones

Profesor Matias Ruiz

Veamos un Video y comenzamos con la clase!

[Que es elUser Story](#)



<https://products.office.com/excel>



<https://www.openoffice.org/>



- Como vimos en el primer ejercicio, encontrar un error es una cosa, pero documentarlo es igual de importante. El informe de errores demuestra un problema de desarrollo y brinda a sus desarrolladores un lugar para comenzar a solucionarlo.
- Puede ser tentador escribir un informe de 10 páginas sobre lo descubierto, pero un informe más simple y más enfocado es mucho mejor y hará felices a todo el equipo.

¿Qué es un requerimiento de software?

El **requerimiento** de software es una descripción de un sistema de software a desarrollar.

- Requisitos funcionales y no funcionales
- Conjunto de casos de uso.
- Definido por el “Business Analyst” o el “Product Owner”

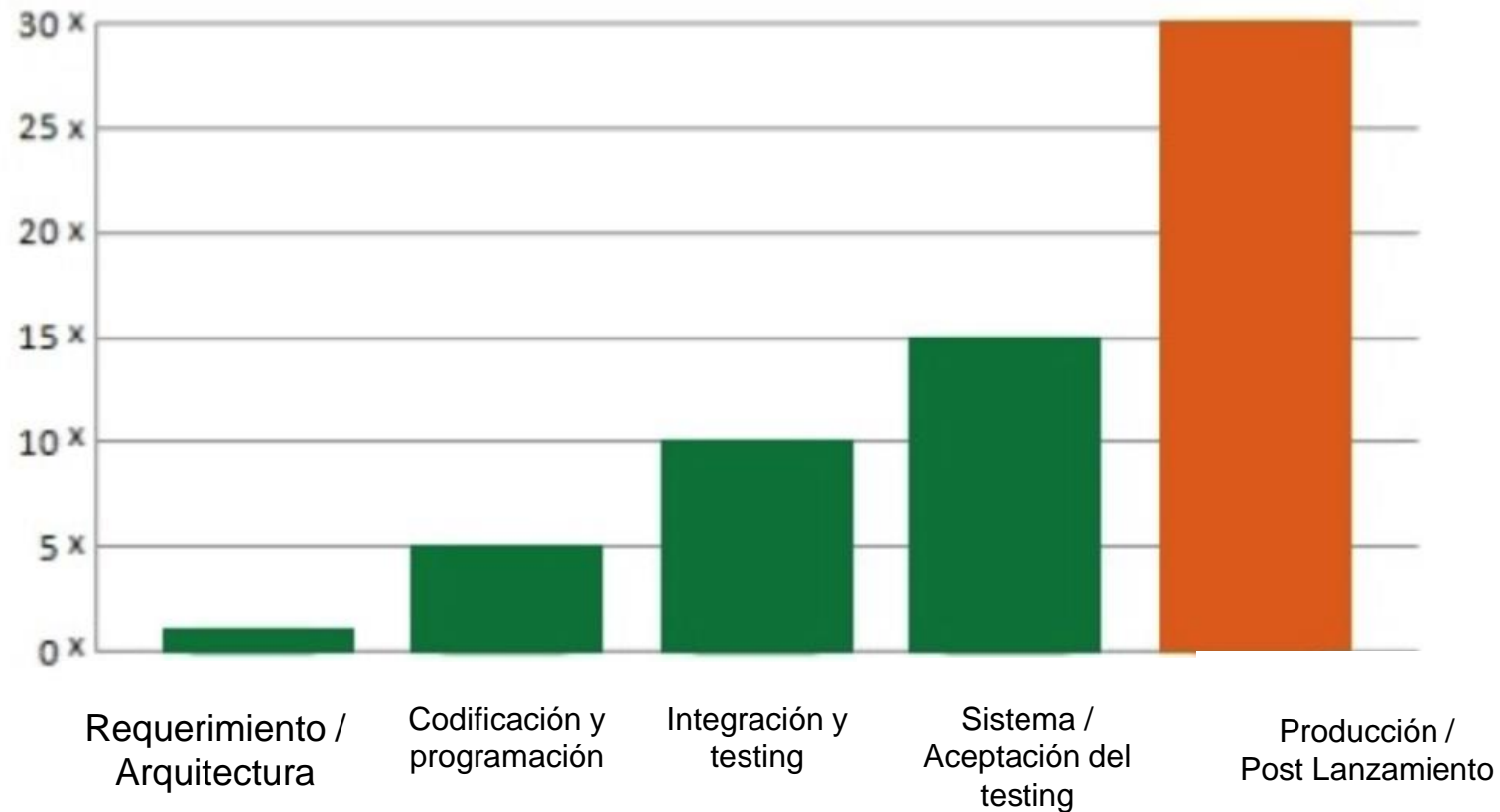
Ejemplo de **requerimiento software** :

- Un usuario puede hacer una transferencia de dinero entre cuentas.
- Un usuario puede hacer transferencias de dinero a otras cuentas
- Un usuario puede recibir dinero de otras cuentas
- Un usuario puede recibir dinero de cuentas propias de otros bancos

Algunas de las tareas del tester de software durante la fase de requisitos del software:

- Leer/probar los requisitos del software
- Colaborar con los business analysts / Product Owners (PO)
- Detectar el comportamiento incorrecto de la aplicación en una etapa temprana

Costo relativo de reparación, basado en el tiempo de detección



Fuente: Instituto nacional de estándares y tecnología de Estados Unidos

Que es un Test Case(Caso de Prueba)?



- **Test case** es un documento que tiene un conjunto de datos de prueba, condiciones previas, resultados esperados, condiciones posteriores
- Verificar el **cumplimiento** de un requisito específico.

Parámetros típicos del **caso de prueba (Test Case)**:

- Escenario de prueba
- Descripción de la prueba
- Pasos de la prueba
- Condiciones previas
- Datos de la prueba
- Resultados esperados
- Prioridad
- Information del ambiente

Veamos un ejemplo de **test case**:

Number	Nombre de la prueba	Descripción (Acción/Pasos)	Condición previa	Datos de prueba	Prioridad	Resultado Esperado	Tester	Status
1	Acceso de Usuario a Menús de Candy Store	Fallas en el acceso del Usuario a los menús de la página	El usuario debe tener acceso a la página	Ninguno	High	El usuario podrá tener acceso y verá el contenido de los menús de la página	Enrique	Failed

Que es un checklist (lista de verificación)?



- **El Checklist** es similar al caso de prueba pero tiene menos información.
- Es beneficioso usar la lista de verificación cuando el equipo de pruebas tiene más **experiencia**.

Veamos un ejemplo de **checklist**:

Number	Checklist Descripción	Status
1	Acceso a la Página	Passed
2	Ver alineación de imágenes y texto	Passed
3	Menú Birthdays	Failed
4	Menú Special Events	Failed
5	Menú Breakfast	Failed
6	Menú Open House	Failed
7	Menú Restaurants	Failed
8	Menú Takeaway	Failed

Test case vs Checklist

Es mejor usar **test cases** cuando:

- Trabajas en un equipo más grande
- Trabajas junto con colegas inexpertos
- Comunicación formal en la oficina.

Es mejor usar **checklist** cuando :

- Trabajas en equipo pequeño
- Todo el equipo tiene un buen conocimiento del producto.
- Entorno colaborativo
- Los productos completos no están disponibles

En estos casos, las **checklist** son más beneficiosas que los **test cases**:

- Si no te estás enfocando en la documentación
- Si necesita comenzar a testear rápidamente
- Si tenes menos tiempo para mantener y actualizar el **Test Cases**

Veamos un Video y luego seguimos.

[¿Que es caja negra?](#)

Pueden tomar apuntes y preparar el Mate o café.



¿Qué es la prueba de caja negra (Black box testing)??



- **Black box testing** es un método de prueba de software, en el que el probador no conoce la implementación interna de la función.
- Este método se denomina "Caja negra", porque el programa de software, a los ojos del probador, es como una caja negra donde el interior **no es visible**.

Los métodos de prueba de caja negra intentan encontrar errores en:

- Función incorrecta o faltante
- Interfaz / Errores visuales
- Problemas de comportamiento
- Errores de inicialización y terminación.

Tipos de pruebas de caja negra

- Prueba funcional: este tipo de prueba de caja negra está relacionada con los requisitos funcionales de un sistema; lo hacen los probadores de software.
- Pruebas no funcionales: se define como un tipo de prueba de software para verificar aspectos no funcionales, como el rendimiento, la usabilidad, la confiabilidad, etc.
- Regression testing - es donde estamos probando las aplicaciones de software existentes para asegurarnos de que un cambio o adición no haya roto ninguna funcionalidad existente.

Pruebas funcionales	Pruebas no funcionales
1 En las pruebas funcionales, el probador prueba qué tan bien se desempeña el sistema	1 En las pruebas no funcionales, los testers comprueban qué tan bien responde el sistema.
2 Las pruebas funcionales se basan en los requisitos del cliente.	2 Las pruebas no funcionales se basan en las expectativas del cliente
3 significa probar la aplicación según los requisitos comerciales	3 Las pruebas no funcionales significan probar la aplicación frente a los clientes y los requisitos de rendimiento.
4 Es parte de una prueba del sistema.	4 También es parte de las pruebas del sistema.
5 Cubre la validación del comportamiento de una aplicación. Esta prueba cubre pruebas unitarias, pruebas de integración,	5 Pruebas no funcionales que validan el rendimiento de una aplicación Las pruebas no funcionales admiten pruebas de carga/rendimiento,
6 smoke testing, sanity testing, pruebas de regression, etc.	6 pruebas de volumen de estrés, pruebas de seguridad, pruebas de instalación, etc.
7 Siempre se concentra en los requisitos del cliente.	7 Siempre se concentra en las expectativas del cliente
Las pruebas funcionales significan cómo se está haciendo el sistema.	Las pruebas no funcionales significan qué tan bien está funcionando su sistema.
8	8 Ejemplos de pruebas de usabilidad, rendimiento y estrés.



source: gauss development



source: testBytes

Si bien el tester de software realiza pruebas de caja negra, existen varias técnicas que facilitan la vida de los testers y que ayudan a detectar **errores importantes**.

- **Equivalence Class Testing (Pruebas de clase de equivalencia):** Una técnica de prueba de caja negra en la que los casos de prueba están diseñados para ejercitar un miembro representativo de cada partición.
- **Boundary Value Testing (Prueba de valor límite):** La prueba de valores límite se centra en los valores al límite.
- **Decision Table Testing (Prueba de tabla de decisiones):** Es una representación visual de las acciones a realizar en función de unas condiciones dadas.



Cuando creamos nuestro **test case** o **checklist**, tenes que tener en cuenta estas técnicas y agregar casos para poder asegurarte de que tenes suficiente cobertura de pruebas de la aplicación.

- **Test Coverage (Cobertura de prueba)** significa que vamos a pasar la mayor cantidad de tiempo posible en una función para testear y cubrir las partes más **críticas** para asegurarnos de que no se entreguen **errores** importantes a los clientes.

