




# Evaluación y mejora para el desarrollo de Software

ISC. Laura Castillo Salzar

## UNIDAD II

# Tipos de pruebas de software


II. Herramientas para la ejecución de pruebas



Todos los tipos de pruebas de software que existen, básicamente, se pueden agrupar en dos grupos: las pruebas funcionales y las pruebas no funcionales.

Sin embargo, seguramente has escuchado hablar de más tipos de pruebas, por ejemplo, pruebas unitarias, pruebas de integración o pruebas de aceptación, pero estos tipos se pueden agrupar dentro de los dos grupos anteriores.

.



# Pruebas Funcionales

- Pruebas unitarias.
- Pruebas de aceptación.
- Pruebas de integración.
- Pruebas de regresión.



# Pruebas No Funcionales

- Pruebas de carga.
- Pruebas de estrés.
- Pruebas de escalabilidad.
- Pruebas de portabilidad.



## PRUEBAS UNITARIAS (FUNCIONALES)

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños *tests* que validan el comportamiento de un objeto y la lógica.

El *unit testing* suele realizarse durante la fase de desarrollo de aplicaciones de software o móviles. Normalmente las llevan a cabo los desarrolladores, aunque en la práctica, también pueden realizarlas los responsables de QA (Quality Assurance).

Hay una especie de mito respecto a las pruebas unitarias. Algunos desarrolladores están convencidos de que son una pérdida de tiempo y las evitan buscando ahorrar tiempo. Nada más alejado de la realidad. Con ellas se detectan antes errores que, sin las pruebas unitarias, no se podrían detectar hasta fases más avanzadas como las pruebas de sistema, de integración e incluso en la beta.

# IMPORTANCIA

Las *unit testing* permiten al desarrollador refactorizar el código más adelante y tener la garantía de que el módulo sigue funcionando correctamente. Para ello se escriben casos de prueba para todas las funciones y métodos, para que cada vez que un cambio provoque un error, sea posible identificarlo y repararlo rápidamente.

La calidad final del código mejorará ya que, al estar realizando pruebas de manera continua, al finalizar el código será limpio y de calidad.

Como las pruebas unitarias dividen el código en pequeños fragmentos, es posible probar distintas partes del proyecto sin tener que esperar a que otras estén completadas.



# Las 3 A's del unit testing

Describe un proceso compuesto de tres pasos.

**Arrange (organizar)**. Es el primer paso de las pruebas unitarias. En esta parte se definen los requisitos que debe cumplir el código.

**Act (actuar)**. Es el paso intermedio de las pruebas, el momento de ejecutar el *test* que dará lugar a los resultados que deberás analizar.

**Assert (afirmar)**. En el último paso, es el momento de comprobar si los resultados obtenidos son los que se esperaban. Si es así, se valida y se sigue adelante. Si no, se corrige el error hasta que desaparezca.

## PRUEBAS DE ACEPTACIÓN (FUNCIONALES)

Las pruebas de aceptación son las últimas pruebas realizadas donde el cliente prueba el software y verifica que cumpla con sus expectativas. Estas pruebas generalmente son funcionales y se basan en los requisitos definidos por el cliente y deben hacerse antes de la salida a producción.

Son fundamentales por lo cual deben incluirse obligatoriamente en el plan de pruebas de software.

Estas pruebas se realizan una vez que ya se ha probado que cada módulo funciona bien por separado, que el software realice las funciones esperadas y que todos los módulos se integran correctamente.



El **test de aceptación** termina de definir el nivel de calidad de la aplicación y le permite conocer al equipo qué tan bien supo interpretar correctamente los requerimientos del usuario o Product owner.

Es el cliente quien tendrá la decisión final de aprobar o no el producto, como también de solicitar modificaciones.



## ¿Cuál es la base para definir las pruebas de aceptación de software?

Según los estándares establecidos por el ISTQB, las pruebas de aceptación de software son diseñadas a partir de:

- Requerimientos del usuario.
- Requerimientos de sistema.
- Procesos de negocio.

Las pruebas de aceptación Se enfocan en verificar si el sistema es “apto para el uso”. Se diseñan principalmente a partir de las especificaciones de requerimientos, casos de uso y de los procesos de negocio definidos.

Los usuarios y clientes suelen involucrarse en la ejecución de las pruebas de aceptación de software, siendo esto un mecanismo muy útil para ganar su confianza en el nuevo sistema o funcionalidad.

En algunos casos, el cliente le pide al equipo, generalmente es el tester, que realice una demo y valide todos los requerimientos definidos.

Para que el cliente apruebe las pruebas de aceptación no deben haber errores, como máximo algún error con criticidad muy baja.

# PRUEBAS DE INTEGRACIÓN (FUNCIONALES)

Las pruebas integrales se tienen que aplicar **justo después de haber llevado a cabo cada prueba unitaria** con la intención de probar los métodos aplicados en el desarrollo. Si no existen ningún problema de código y las pruebas unitarias han terminado de forma exitosa se podrá pasar al test integral para asegurarse de que en este punto no se produce ningún tipo de problema en la combinación de elementos unitarios. El motivo principal se encuentra en que el test integral lleva a cabo la revisión conjunta de los diferentes elementos que están presentes con el objetivo de formar el software. Se realiza la comprobación para ver que todo funciona de una manera adecuada en conjunto, dado que no es extraño que se produzcan alteraciones en el rendimiento.



Con esta comprobación representada por la prueba de integración podremos ver si **la comunicación entre los distintos componentes presentes** en el software es funcional. Se comprueban también las comunicaciones de forma invariable tanto si están representadas con software o con hardware. En el caso de ser necesario ir más allá por la existencia de subsistemas los profesionales que estén al cargo de este software también tendrán que hacer la prueba específica de subsistemas, que viene a ser una variación de la de integración pero profundizando en los elementos que están incluidos dentro de cada sistema. Durante este proceso en el cual se verifican los distintos tipos de integración, los especialistas tendrán que ensamblar los módulos independientes, dar forma al software al completo y verificar el proceso a conciencia. Una de las ventajas de este protocolo se encuentra en la oportunidad de llevar a cabo pruebas de una manera paralela, lo que aporta flexibilidad extra en el proceso de calendarización. Para ello se optará por **la elección de frameworks en los que las pruebas se puedan combinar** con el desarrollo y con la supervisión simplificada de los procesos, especialmente en aquellos casos en los que las pruebas puedan ser un poco más complejas. El resultado garantizará que el proyecto de software pueda avanzar hacia su siguiente fase antes de darse por finalizado.

# PRUEBAS DE REGRESIÓN (FUNCIONALES)

Pruebas de regresión son pruebas de un programa previamente probado que ha sufrido modificaciones, para asegurarse que no se han introducido o descubierto defectos en áreas del software que no han sido modificadas como resultado de los cambios realizados. Se realiza cuando el software o su entorno han sido modificados.

Las pruebas de regresión no necesitan ser escritas como tal, sino que se pueden crear a partir de pruebas que hemos ido elaborando durante el desarrollo. La idea es que cada vez que modificas el código fuente, ejecutes una selección de pruebas relevantes que ya tenías para ver si todo funciona como debería. Pruebas funcionales, pruebas unitarias, pruebas de integración (cualquier cosa que haya verificado durante el proceso de desarrollo que ciertos componentes del software funcionan como deberían) pueden actuar como pruebas de regresión.

## “Implementación de un cambio”

¿Qué constituye una variación en nuestro software? **Toda aquella modificación, por mínima que sea, necesitará**, ya sea en mayor o menor medida, **de pruebas de regresión**. Debemos tener en cuenta, además, que un cambio puede ir desde cualquier adición de funcionalidad, hasta una pequeña actualización de configuración.

Asimismo, cabe decir que el tamaño de la alteración realizada no tiene porque estar directamente relacionada con la cantidad de Regression Testing a realizar. Al fin y al cabo, la “cantidad” estará íntimamente relacionada con otro de los conceptos claves del mundo de la calidad: el riesgo. O lo que es lo mismo, a mayor riesgo ,mayor necesidad de ejecutar tests de regresión, el riesgo puede revelarse en el más pequeño e insospechado de los cambios.

# Tipos de pruebas funcionales

## para el aseguramiento de la calidad



### ¿Qué son?

Son un proceso de control de calidad para asegurar el cumplimiento de un sistema o componente con requerimientos funcionales.

Estas pruebas pueden realizarse durante:

- la fase de desarrollo: individualmente para secciones específicas desarrolladas por el equipo
- al final: cuando las diferentes secciones del proyecto están unidas.



### Unitarias

Asegura que cada célula del código desarrollado en un componente brinde los resultados adecuados.

Se evalúan:

- interfaz
- la especificación de un componente

# Tipos de pruebas funcionales para el aseguramiento de la calidad



## de Integración

Prueba componentes individuales con el objetivo de verificar cómo los módulos, que trabajan de forma individual, funcionan cuando están integrados.



## de Regresión

Asegura que los cambios o adiciones en el desarrollo no hayan alterado ni eliminado las funcionalidades existentes. Su objetivo es encontrar errores que puedan haber sido introducidos accidentalmente en la compilación



## de Cordura

Si tienes una compilación con modificaciones menores, en vez de ejecutar las pruebas de regresión, realizamos una prueba de cordura.

Determina que las modificaciones realmente hayan solucionado los problemas, y que dichas correcciones no hayan generado ningún problema.



## de Aceptación

Los usuarios reales del software lo usan para verificar que cumpla con las tareas requeridas en un ambiente 'real'. En ocasiones se realiza como punto de control final entre todos los tipos de pruebas funcionales



**GRACIAS POR SU ATENCIÓN !!!**



¿Dudas o Preguntas?