CICLO 2

[FORMACIÓN POR CICLOS]

Introducción a las pruebas UNITARIAS

Ingeni@ Soluciones TIC

UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

En sus inicios, la programación se concibió como una actividad experimental, con unos procesos que, si bien requerían de conocimiento muy especializado, no iban de la mano de prácticas que garantizaran su calidad. Posteriormente, a lo largo de los años, se han propuesto diferentes prácticas, *frameworks*, herramientas y metodologías que buscan un mayor énfasis del desarrollo de software en la calidad del producto y de los procesos.

Es así como dos conceptos han ganado importancia en los últimos años en el desarrollo de software. Estos son, la verificación y la validación. La **verificación**, busca evaluar un sistema en construcción para determinar si cumple con los requisitos y especificaciones del diseño de etapas de construcción previas. La **validación**, por su parte, busca evaluar un sistema en construcción para determinar si cumple con los requisitos y expectativas de los interesados, especialmente, de los usuarios o clientes. Cada vez más, estos dos conceptos se implementan mediante el *testing*.

El *testing* (o las pruebas de software) se utilizan para verificar y/o validar que un sistema de software en construcción cumpla con las especificaciones deseadas y haga lo que se supone que debe hacer. Para lo anterior, contamos con diferentes niveles o tipos de pruebas. Si bien se proponen constantemente diferentes clasificaciones de pruebas, la mayor parte de estas incluyen los siguientes niveles:

Pruebas de aceptación

Las pruebas de aceptación (o *acceptance tests*) están orientadas a la validación del sistema de software, y buscan evaluar la conformidad de este con los requisitos de negocio, valorando si es aceptable para ser entregado a clientes y usuarios.

Pruebas de sistema

Por otro lado, las pruebas de sistema buscan evaluar un sistema en su totalidad. El sistema suele estar terminado o próximo a su finalización, y estas pruebas comprenden acciones que realizaría un usuario, además de la interacción del sistema con otros sistemas asociados. Las pruebas de sistema por lo general son pruebas de **caja negra**, es decir, se realizan sin que se conozca el código fuente ni el funcionamiento interno del sistema en gran detalle, y se pueden dividir, a su vez, en diferentes tipos:

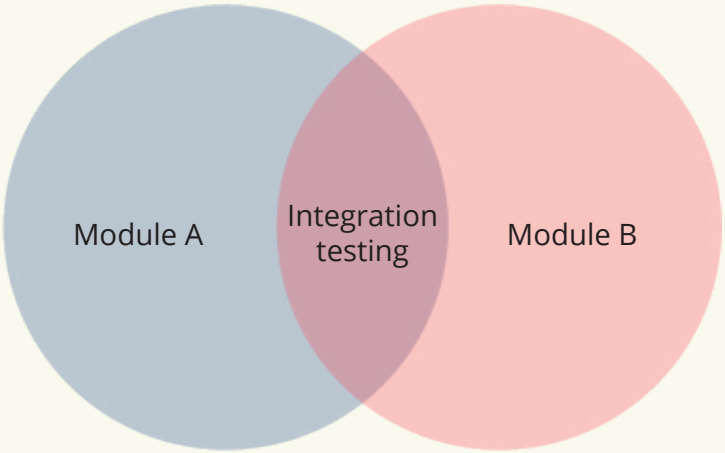
System testing

- Pruebas de estrés o de carga.
- Pruebas de usabilidad.
- Pruebas de compatibilidad.

Entre otros tipos.

Pruebas de integración

Las pruebas de integración, por su parte, buscan evaluar la integración, interacción o combinación entre dos o más componentes de código de un sistema de software. Esto, debido a que es común que dos o más componentes de software trabajen juntos dentro de una aplicación y se generen dependencias entre ellos. Estas dependencias se deben verificar a medida que el sistema de software evoluciona.



Pruebas unitarias

Finalmente, tenemos las pruebas unitarias, también denominados tests unitarios. Una prueba unitaria consiste en evaluar el comportamiento de una clase, una función o un componente de manera independiente al resto del sistema de software o aplicación. Por lo general, suelen ser pruebas de **caja blanca**, es decir, pruebas en las cuales se tiene conocimiento del código fuente de la aplicación, así como de su funcionamiento interno detallado. Las características de estas pruebas han permitido que sean ampliamente automatizadas.