

Tipos de tests



Veamos de qué tratan los tres tipos de tests más conocidos: unitarios, integración, end to end (E2E). Durante la cursada trabajaremos con tests unitarios pero es importante saber que hay varias maneras de testear y de qué trata cada una.



Pruebas Unitarias

Están destinadas a verificar si una parte pequeña y aislada del código (unidad) se comporta como se pretendía. Consiste en un fragmento de código que prueba el código de producción de alguna manera, y luego verifica si el resultado coincide con lo esperado. Es decir, una prueba unitaria adecuada es una herramienta fantástica para obtener una retroalimentación súper precisa.

Pruebas Unitarias

Una prueba no es una prueba unitaria si:

- Habla con la base de datos ya que depender de estructuras como la base de datos o el sistema de archivos las hace lentas.
- Se comunica a través de la red. Una prueba no debería empezar a fallar si antes no fallaba por ende no debería depender de dependencias externas.
- Toca el sistema de archivos porque si este tiene algún error podría trasladarse a nuestros tests.
- No puede ejecutarse al mismo tiempo que cualquiera de tus otras pruebas unitarias,
- Tenemos que hacer cosas especiales en nuestro entorno para ejecutarlas.

Pruebas de Integración

A diferencia de las pruebas unitarias, en las que se prueba una pequeña unidad aislada, las pruebas de integración suelen implicar la comprobación de una funcionalidad concreta, normalmente denominada módulo, que tiene dependencias de otra funcionalidad. El objetivo de estas pruebas es comprobar la conectividad y la comunicación entre los distintos componentes de la aplicación. También ayudan a validar las interfaces de la aplicación para que los datos que fluyen de un módulo a otro sean los adecuados.

Pruebas de Integración

Para implementarlas podemos elegir entre 3 enfoques según el tamaño de nuestro proyecto, equipo, presupuesto y otros factores:

1. Enfoque ascendente: los tester se centran en la integración de los módulos más pequeños. La principal ventaja de este enfoque es la posibilidad de fallar rápidamente, ya que los errores en los módulos más pequeños son más fáciles de detectar y más rápidos de solucionar.
2. Enfoque descendente: aquí se da prioridad a los módulos más grandes y complejos, y se simulan los de menor nivel.
3. Enfoque Big Bang: con este enfoque, todos los módulos se prueban juntos al mismo tiempo. La detección de errores puede ser un poco más difícil en comparación con los otros enfoques, pero en una aplicación pequeña puede ser una gran opción.

Pruebas **End to End (E2E)**

Cuando se hacen pruebas E2E, se busca probar el producto de la misma manera que lo experimenta una persona usuaria real. Su objetivo es simular la experiencia paso a paso de una persona usuaria y permitir validar diferentes subsistemas y capas de la aplicación. Hay dos maneras de llevarlas a cabo:

1. Pruebas Horizontales: replican los casos de uso típico de las aplicaciones y los recorren de principio a fin. Por ejemplo el proceso de registro de un nuevo usuario. Es la modalidad más común.
2. Pruebas Verticales: tienen como objetivo probar los componentes del sistema siguiendo los datos a través de las capas de la interfaz de usuario, la API y la base de datos. Por ejemplo el proceso de compra en un ecommerce.

Pruebas **End to End (E2E)**

Los beneficios de este tipo de tests van desde asegurar que todo funciona como se espera, yendo desde la interfaz de usuario hasta la capa de la base de datos, aumentar la cobertura general de las pruebas y ayudar a detectar errores antes de que lo hagan las personas usuarias finales.

Son grandes beneficios pero sepamos también que las pruebas E2E son, claramente, las más costosas y complejas de realizar y que, además, llevan más tiempo.

DigitalHouse>
Coding School