

# INFORME SOBRE LA APLICACIÓN DE LOS DIFERENTES TIPOS DE TESTING EN WWW.ELIDEALGALLEGO

**QA Tester:** Humberto López Rodríguez

Lunes, 11 de marzo de 2024

## **Introducción**

En el mundo del desarrollo de software, asegurar la calidad del producto es fundamental. Una de las prácticas más importantes para garantizar esta calidad es el testing o pruebas del software. En este informe, exploraremos los diferentes tipos de testing que existen y cómo se aplican en el desarrollo de software.

## **Manual vs Automated Testing**

El testing puede ser llevado a cabo de dos maneras principales: manual y automatizado. Las pruebas manuales las ejecutan personas, que interactúan con el software siguiendo pasos predefinidos. Las pruebas automatizadas las realizan máquinas, ejecutando scripts de prueba escritos.

Un ejemplo de pruebas manuales podría ser la verificación de la funcionalidad de inicio de sesión en el sitio web de El Ideal Gallego. Un evaluador humano podría intentar iniciar sesión con diferentes combinaciones de nombre de usuario y contraseña para asegurarse de que el sistema responde correctamente en cada caso.

Las pruebas automatizadas podrían consistir en escribir scripts que simulan el proceso de navegación por el sitio web, haciendo clic en diferentes enlaces y botones para verificar que todas las funcionalidades estén operativas. Por ejemplo, se podría escribir un script que automatice el proceso de búsqueda de noticias recientes en la página de inicio de El Ideal Gallego y verifique que se muestran correctamente.

## **Los Diferentes Tipos de Tests**

### **Unit Tests**

Las pruebas unitarias se enfocan en probar unidades individuales de código, como funciones o métodos. Estas pruebas son de bajo nivel y su objetivo es verificar el correcto funcionamiento de componentes específicos del software.

Por ejemplo, en el desarrollo del backend del sitio web de El Ideal Gallego, se podrían escribir pruebas unitarias para verificar que la función que recupera las noticias más recientes de la base de datos devuelve los resultados esperados.

### **Integration Tests**

Las pruebas de integración verifican la interacción entre diferentes módulos o servicios del software. Se aseguran de que todas las partes funcionen en armonía cuando se utilizan juntas.

Un ejemplo sería verificar que el sistema de comentarios integrado en las noticias de El Ideal Gallego funcione correctamente al publicar un nuevo comentario. Esto implicaría probar la interacción entre la base de datos, el backend y la interfaz de usuario.

### **Functional Test**

Estas pruebas se centran en verificar que el software cumpla con los requisitos de negocio establecidos. Se evalúa la salida de una acción sin prestar atención a los estados intermedios del sistema.

Un ejemplo sería probar la funcionalidad de compartir noticias en redes sociales desde el sitio web de El Ideal Gallego. Se verificaría que, al hacer clic en el botón de compartir, se abra una ventana emergente con las opciones de compartir en diferentes plataformas.

### **End-to-end Tests**

Las pruebas de punta a punta replican el comportamiento de los usuarios con el software en un entorno completo de la aplicación. Verifican que los flujos de usuario funcionen como se espera, desde la carga de una página web hasta acciones más complejas como realizar pagos en línea.

Un ejemplo sería simular el proceso completo de compra de una suscripción premium en el sitio web de El Ideal Gallego, desde la selección del plan hasta el pago exitoso y la activación de la cuenta premium.

### **Regression Testing**

Estas pruebas se centran en verificar que las funcionalidades que funcionaban correctamente en el pasado continúen haciéndolo después de la introducción de nuevos cambios o características en el software.

Por ejemplo, después de realizar una actualización en el diseño del sitio web de El Ideal Gallego, se realizarían pruebas de regresión para asegurarse de que todas las funcionalidades existentes, como la navegación y la búsqueda, sigan funcionando correctamente.

### **Smoke Testing**

Las pruebas de humo verifican la funcionalidad básica de una aplicación, asegurando que las características más importantes del sistema funcionen como se espera. Son pruebas rápidas y se ejecutan después de la construcción de una nueva versión del software o después de un despliegue.

Un ejemplo sería realizar pruebas de humo después de una actualización importante del sitio web de El Ideal Gallego, verificando que la página de inicio cargue correctamente y que las noticias más recientes se muestren sin errores.

## Acceptance Testing

Estas pruebas son ejecutadas para verificar si el sistema cumple con los requisitos de negocio establecidos. Se realizan después de que una fase de desarrollo haya finalizado y se centran en replicar el comportamiento de los usuarios.

Por ejemplo, se podrían realizar pruebas de aceptación en el sitio web de El Ideal Gallego para verificar que todas las funcionalidades requeridas por los usuarios, como la búsqueda de noticias por categoría o la suscripción a boletines informativos, estén implementadas correctamente.

## Performance Testing

Las pruebas de rendimiento evalúan cómo responde el sistema bajo una carga determinada. Se centran en la fiabilidad, estabilidad y disponibilidad del sistema, verificando los tiempos de respuesta y el comportamiento del sistema ante una carga significativa.

Por ejemplo, se podrían realizar pruebas de rendimiento en el sitio web de El Ideal Gallego para verificar cómo responde el servidor ante un alto número de solicitudes simultáneas, asegurando que el sitio pueda manejar picos de tráfico sin experimentar tiempos de inactividad o ralentización significativa.

## ¿Por qué y cómo automatizar nuestras pruebas?

La automatización del testing es fundamental para garantizar la calidad del software y acelerar el proceso de desarrollo. Es importante escribir pruebas automatizadas utilizando frameworks de testing adecuados para el lenguaje de programación utilizado en el proyecto. Estas pruebas pueden ejecutarse automáticamente con herramientas de integración continua o servicios en la nube.

## Exploratory Testing

Aunque la automatización del testing es crucial, las pruebas manuales también son importantes, especialmente en el caso de pruebas exploratorias. Estas pruebas permiten descubrir errores que pueden no ser evidentes y ayudan a comprender cómo se comporta el sistema en diferentes escenarios.

## Conclusión

En resumen, el testing es una parte fundamental del desarrollo de software, y existen diferentes tipos de pruebas que se pueden aplicar según las necesidades del proyecto. Tanto las pruebas manuales como las automatizadas son importantes para garantizar la calidad del software y deben formar parte del proceso de desarrollo desde el principio. La combinación de ambos tipos de pruebas permite detectar errores de manera eficiente y asegurar que el software cumpla con los requisitos de negocio establecidos.

Este informe proporciona una visión general de los diferentes tipos de testing en el desarrollo de software y destaca la importancia de cada uno en el aseguramiento de la calidad del producto final. Los ejemplos específicos tomados del sitio web de El Ideal Gallego ilustran cómo se pueden aplicar estos tipos de pruebas en un contexto real.

## Referencias

- "Different types of testing explained"
- "The different types of software testing"
- "Testing" - El Ideal Gallego