|  |
| --- |
|  |
| Software Testing |
|  |

|  |
| --- |
|  |

# Integrantes: Falcón Camila, Marrero Facundo, Paredero Balda Sebastian

Indice

**Historia3**

**Pruebas estáticas3**

**Pruebas dinámicas4**

**Pruebas de compatibilidad4**

**Pruebas de regresión4**

**Pruebas de integración5**

**Tipos de prueba de ejecucion5**

Pruebas manuales5

Pruebas dinamicas5

**Enfoques de prueba4**

Caja blanca4

Caja negra4

**Niveles de prueba5**

Pruebas unitarias5

Pruebas de integrales5

Pruebas de sistemas 5

**Pruebas funcionales5**

Pruebas de humo5

Prueba de regresion5

Prueba de aceptacion5

Alpha testing6

Beta testing6

**Pruebas no funcionales6**

Pruebas de usabilidad6

Pruebas de rendimiento6

Pruebas de escabilidad6

Pruebas de mantenimiento6

Prueba de seguridad web6

# Historia

El objetivo de las pruebas es presentar información sobre la calidad del producto a las personas responsables de éste. Las pruebas de calidad presentan los siguientes objetivos: encontrar defectos o bugs, aumentar la confianza en el nivel de calidad, facilitar información para la toma de decisiones, evitar la aparición de defectos.

Teniendo esta afirmación en mente, la información que puede ser requerida es de lo más variada. Esto hace que el proceso de *testing* sea completamente dependiente del contexto en el que se desarrolla.

El ambiente ideal de las pruebas de testing es aquel que es independiente del desarrollo del software, de esta manera se logra objetividad en las pruebas.

A pesar de lo que muchos promueven, no existen las "mejores prácticas" como tal. Toda práctica puede ser ideal para una situación pero completamente inútil o incluso perjudicial en otra.

Por esto, las actividades, técnicas, documentación, enfoques y demás elementos que condicionarán las pruebas a realizar, deben ser seleccionadas y utilizadas de la manera más eficiente según contexto del proyecto.

# Pruebas estáticas

Son el tipo de pruebas que se realizan sin ejecutar el código de la aplicación.

Puede referirse a la revisión de documentos, ya que no se hace una ejecución de código. Esto se debe a que se pueden realizar "pruebas de escritorio" con el objetivo de seguir los flujos de la aplicación.

# Pruebas dinámicas

Todas aquellas pruebas que para su ejecución requieren la ejecución de la aplicación.

Las pruebas dinámicas permiten el uso de técnicas de caja negra y caja blanca con mayor amplitud. Debido a la naturaleza dinámica de la ejecución de pruebas es posible medir con mayor precisión el comportamiento de la aplicación desarrollada.

# Pruebas de compatibilidad

Son las pruebas que se realizarán en un software o aplicación determinado y que comprobarán que tu desarrollo es compatible con todos los navegadores de Internet y todos los sistemas convenientes. Estas pruebas son realmente importantes para que tu producto llegue a todos los usuarios que deberían de llegar y que todo el mundo pueda utilizarlo con lo que disponga en su equipo informático.

# Pruebas de regresión

Se evalúa el correcto funcionamiento del software desarrollado frente a evoluciones o cambios funcionales. El propósito de éstas es asegurar que los casos de prueba que ya habían sido probados y fueron exitosos permanezcan así. Se recomienda que este tipo de pruebas sean automatizadas para reducir el tiempo y esfuerzo en su ejecución.

# Pruebas de integración

Es el nivel de pruebas posterior a las pruebas modulares de los componentes de un sistema. Se centra principalmente en probar la comunicación entre los componentes de un mismo sistema, comunicación entre sistemas o entre hardware y software.

# Tipos de prueba de ejecución

#### Pruebas manuales

Las pruebas manuales son ejecutadas por un usuario. La fiabilidad siempre está en duda ya que al ser realizadas por usuarios estos siempre puede cometer errores. Se llevan a cabo sin ayuda de herramientas de evaluación automatizadas.

#### Pruebas automáticas

Es el uso de software especial para controlar la ejecución de pruebas y la comparación entre los resultados obtenidos y los resultados esperados. La automatización de pruebas permite incluir pruebas repetitivas y necesarias dentro de un proceso formal de pruebas ya existente o bien adicionar pruebas cuya ejecución manual resultaría difícil.

# Enfoques de prueba

#### Prueba de caja blanca

Se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. El testeador escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecucion del programa y cerciorarse de que se devuelven los valores de salida adecuados.

#### Prueba de caja negra

En pruebas de software, conociendo una función específica para la que fue diseñado el producto, se pueden diseñar pruebas que demuestren que dicha función está bien realizada. Dichas pruebas son llevadas a cabo sobre la interfaz del software, es decir, de la función, actuando sobre ella como una caja negra, proporcionando unas entradas y estudiando las salidas para ver si concuerdan con las esperadas.

# Niveles de prueba

#### Pruebas unitarias

En programación, una prueba unitaria es una forma de comprobar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las pruebas de integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

#### Pruebas integrales

Pruebas de integración son aquellas que se realizan en el ámbito del [desarrollo de software](https://es.wikipedia.org/wiki/Desarrollo_de_software) una vez que se han aprobado las [pruebas unitarias](https://es.wikipedia.org/wiki/Prueba_unitaria). Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez.

Consiste en realizar pruebas para verificar que un gran conjunto de partes de [software](https://es.wikipedia.org/wiki/Software) funcionan juntos.

#### Pruebas de sistema

Evalúan el sistema en su conjunto. El objetivo de este tipo de pruebas es comprobar que se cumplen los requisitos funcionales y las especificaciones técnicas del software. Suele ser habitual que se prueba el sistema en un entorno similar al de producción.

# Pruebas funcionales

#### Pruebas de humo

Son aquellas pruebas que pretenden evaluar la calidad de un producto de [software](https://es.wikipedia.org/wiki/Software) previo a una recepción formal, ya sea al equipo de pruebas o al usuario final, es decir, es una revisión rápida del producto de software para comprobar que funciona y no tiene defectos que interrumpan la operación básica del mismo. Se hace la analogía al humo, puesto que en [bienes raíces](https://es.wikipedia.org/wiki/Bienes_ra%C3%ADces) se inyecta humo en las tuberías de agua para validar que no tengan fugas, evitando provocar inundaciones.

#### Pruebas de regresión

Cualquier tipo de pruebas de [software](https://es.wikipedia.org/wiki/Software) que intentan descubrir errores ([bugs](https://es.wikipedia.org/wiki/Error_de_software" \o "Error de software)), carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa.

Este tipo de cambio puede ser debido a prácticas no adecuadas de [control de versiones](https://es.wikipedia.org/wiki/Control_de_versiones), falta de consideración acerca del ámbito o contexto de producción final y extensibilidad del error que fue corregido (fragilidad de la corrección), o simplemente una consecuencia del rediseño de la aplicación.

Por lo tanto, en la mayoría de las situaciones del desarrollo de software se considera una buena práctica que cuando se localiza y corrige un bug, se grabe una prueba que exponga el bug y se vuelvan a probar regularmente después de los cambios subsiguientes que experimente el programa.

#### Pruebas de aceptación

Las pruebas de aceptación evalúan que el sistema cumple con los requisitos del cliente, para ello se cuenta con su participación y por norma general la superación de este tipo de pruebas significa que el cliente acepta el sistema y éste queda validado.

#### Alpha testing

Es el testeo general para corregir errores básicos de un programa. Esta no es la versión final de software y cierta funcionalidad puede ser añadido al software incluso después de la prueba alfa.

#### Beta testing

Se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. El desarrollador no está presente normalmente. La prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador. Es la última fase de las fases de prueba.

# Pruebas no funcionales

#### Pruebas de usabilidad

Consisten en seleccionar a un grupo de usuarios y solicitarles que lleven a cabo las tareas para los cuales fue diseñado el programa, en tanto el equipo de diseño, desarrollo y otros involucrados toman nota de la interacción, particularmente de los errores y dificultades con las que se encuentren los usuarios.

#### Pruebas de rendimiento

Se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos.

#### Prueba de escabilidad

Son aquellas que permiten determinar el grado de escabilidad que tiene un sistema. Se entiende como escalable a la capacidad que tiene el sistema para que, sin aplicar cambios drásticos en su configuración, pueda soportar el incremento de demanda en la operación.

#### Pruebas de mantenibilidad

Miden el grado de facilidad con la que el software puede ser analizado, modificado y probado. Miden el tiempo empleado en diagnosticar y arreglar los problemas identificados en el sistema. Miden el esfuerzo en realizar cambios al software.

#### Pruebas de seguridad

Este servicio está basado en la metodología y guías propuestas por OWASP (proyecto abierto de seguridad en aplicaciones web). Entre las pruebas de seguridad se incluyen pruebas de intrusión y análisis de vulnerabilidades, así como el asesoramiento y revisión de resultados tras el análisis realizado.