



TALLER DE TESTING Y CALIDAD DE SOFTWARE

U2. Desarrollo y ejecución de casos de prueba.
Semana 7





ESCUELA DE CONSTRUCCIÓN E INGENIERIA

Director: Marcelo Lucero

ELABORACIÓN

Experto disciplinar: Aída Villamar Gallardo.

Diseño instruccional: Carla Silva Alvarado.

VALIDACIÓN

Experto disciplinar: Andrés del Alcázar

Jefa de Diseño Instruccional: Alejandra San Juan Reyes.

EQUIPO DE DESARROLLO

AIEP

AÑO

2021

Tabla de contenidos

Aprendizaje esperado de la semana.....	5
1. Conceptos asociados a pruebas funcionales y no funcionales, según estándares de la industria.	6
1.1 Conceptos de pruebas funcionales y no funcionales	6
1.2 Estándares de la industria	8
2. Tipos de pruebas funcionales y no funcionales, según estándares de la industria.	9
2.1 Tipos de pruebas funcionales.....	9
2.1.1 Pruebas de humo.....	9
2.1.2 Pruebas de regresión.....	10
2.1.3 Pruebas de aceptación	10
2.1.4 Pruebas alfa testing	11
2.1.5 Pruebas Beta Testing.....	11
2.2 Tipos de pruebas no funcionales.....	12
2.2.1 Pruebas de seguridad:	12
2.2.2 Pruebas de usabilidad.....	13
2.2.3 Pruebas de rendimiento	13
2.2.4 Pruebas de escalabilidad	13
2.2.5 Pruebas de mantenibilidad.....	13
2.2.6 Pruebas de inestabilidad	14
2.2.7 Pruebas de portabilidad	14
3. Pruebas funcionales en proyectos de desarrollo de software, según estándares de la industria.	14
3.1 Aplicación de pruebas funcionales.....	14
4. Pruebas no funcionales en proyectos de desarrollo de software, según estándares de la industria.	15
4.1 Aplicación de pruebas no funcionales	15



Conclusiones	19
Referencias bibliográficas	20



Aprendizaje esperado de la semana

Aplican pruebas funcionales y no funcionales para la medición de calidad de proyectos de desarrollo de software, según estándares de la industria.

Introducción

- ¿Son importantes las pruebas no funcionales?
- ¿Cuáles son los estándares de la industria que debemos tener en consideración?
- ¿Qué diferencia hay entre las pruebas funcionales y las no funcionales?

Esta semana, profundizaremos los conceptos de pruebas que hemos ido conociendo hasta ahora. La idea es que podamos tener una visión general y completa de cada una de ellas, en cada etapa del ciclo de vida de nuestros desarrollos. Incluso, cuando estemos realizando actualizaciones o modificaciones de un sistema existente, no podemos dejar de considerar las pruebas. Finalmente revisaremos los aspectos a considerar para planificar, diseñar y ejecutarlas. Cerrando el proceso con las pruebas de aceptación, con la participación del cliente o usuario.



1. Conceptos asociados a pruebas funcionales y no funcionales, según estándares de la industria.

1.1 Conceptos de pruebas funcionales y no funcionales

Las **pruebas funcionales** se enfocan en asegurar que las funcionalidades de los sistemas que se están probando, se encuentren de acuerdo con los requisitos y especificaciones funcionales.

(Montecinos, 2020) Señala que:

“La funcionalidad representa la capacidad del producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones específicas”.

Las características de funcionalidad según las establece la ISO 25010 son: idoneidad, exactitud, interoperabilidad y seguridad.

En la ISO 25010 se indica que La funcionalidad a su vez, la divide en las siguientes características:

- **Complejidad funcional:** el grado en el que las funcionalidades cubren todas las tareas y objetivos del usuario especificados.
- **Corrección funcional:** capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- **Pertenencia funcional:** capacidad del producto de software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Estas pruebas pueden llevarse a cabo en todos los niveles de prueba, como por ejemplo pruebas de componente basadas en una especificación. Las pruebas funcionales, suelen estar asociadas a las técnicas de diseño de pruebas de caja negra, ya que tienen en cuenta el comportamiento externo del software.

Las **pruebas no funcionales**, a diferencia de las pruebas funcionales, son aquellas que validan un sistema de acuerdo con sus requerimientos no funcionales. Es decir, no es el que hacen sino el cómo lo hacen. Usualmente encontraremos que estas características son medidas en base a escalas por medio de las cuales, podremos establecer si se ajustan sus resultados a lo establecido inicialmente.

Algo a considerar es que cuando un requerimiento funcional no se cumple como se espera, el sistema se degrada. En cambio, cuando es un requerimiento no funcional normalmente interrumpen su continuidad operativa o se inutiliza.



Como lo menciona (Montecinos, 2020):

“La ISO 25010 también define las características que han de tener estas pruebas que son fiabilidad, facilidad de uso, eficiencia, compatibilidad y seguridad”

Pruebas funcionales	Pruebas no funcionales
Prueba 'qué' hace el producto. Comprueba las operaciones y acciones de una aplicación.	Comprueba el comportamiento de una aplicación.
Las pruebas funcionales se realizan según los requisitos comerciales.	Las pruebas no funcionales se realizan en función de las expectativas del cliente y los requisitos de rendimiento.
Prueba si el resultado real está funcionando de acuerdo con el resultado esperado.	Comprueba el tiempo de respuesta y la velocidad del software en condiciones específicas.
Se realiza de forma manual. Ejemplo: método de prueba de caja negra.	Es más factible realizar pruebas con herramientas automatizadas. Ejemplo: Loadrunner.
Prueba según los requisitos del cliente.	Prueba según las expectativas del cliente.
Los comentarios de los clientes ayudan a reducir los factores de riesgo del producto.	Los comentarios de los clientes son más valiosos para las pruebas no funcionales, ya que ayudan a mejorar y permiten que el evaluador conozca las expectativas del cliente.
Está probando la funcionalidad del software.	Está probando el rendimiento de la funcionalidad del software.
Las pruebas funcionales tienen los siguientes tipos: <ul style="list-style-type: none">•Examen de la unidad•Pruebas de integración• Prueba del sistema•Test de aceptación	Las pruebas no funcionales incluyen: <ul style="list-style-type: none">•Pruebas de rendimiento•Prueba de carga•Pruebas de estrés• Prueba de volumen• Pruebas de seguridad• Prueba de instalación• Prueba de recuperación
Ejemplo: una página de inicio de sesión debe mostrar cuadros de texto para ingresar el nombre de usuario y la contraseña.	Ejemplo: pruebe si una página de inicio de sesión se carga en 5 segundos.

Tabla 1: Diferencia entre pruebas funcionales y no funcionales

Fuente: (myservename, 2021)

1.2 Estándares de la industria

Con respecto a los estándares ampliamente utilizados en la industria al momento de llegar a la etapa de pruebas de sistemas, encontramos a BSI¹
Dentro de estos estándares BSI:

Pruebas de Software	
BS 7925 - 1	BS 7925 - 2
Vocabulario	Pruebas a componentes de software

Tabla 1: estándares BSI

Dentro de IEEE:

IEEE	
estándar 829	estándar 1008
Documentación de la prueba de software	Pruebas de unidad

Tabla 2: Estándares IEEE

Otros estándares ISO	
ISO/IEEC 12207	Information Technology / Software Life Cycle Processes es el estándar para los procesos de ciclo de vida del software de la organización ISO.
ISO/IEEC 15289	Ingeniería de sistemas y software: contenido de los productos de información del proceso del ciclo de vida de los sistemas y del software (documentación)

Tabla 3: Otros estándares ISO

No obstante, como se explica en (Programación, s.f.), estos estándares no cubren ciertos aspectos de la fase de pruebas, como son la organización, el proceso y gestión de las pruebas, presentan pocas pruebas funcionales y no funcionales etc. Ante esta problemática, la industria ha desarrollado la norma ISO/IEC 29119.

La norma ISO/IEC 29119 de prueba de software, pretende unificar en una única norma, todos los estándares, de forma que proporcione vocabulario, procesos, documentación y técnicas para cubrir todo el ciclo de vida del software. Desde estrategias de prueba para la organización y políticas de prueba, prueba de proyecto al análisis de casos de prueba, diseño, ejecución e informe. Con este estándar, se podrá realizar cualquier prueba para cualquier proyecto de desarrollo o mantenimiento de software.

¹ British Standards Institution, BSI o BSI Group, es una multinacional con sede en Londres, cuyos objetivos principales se incluyen la certificación, auditoría y formación en las normas para estandarizar procesos.

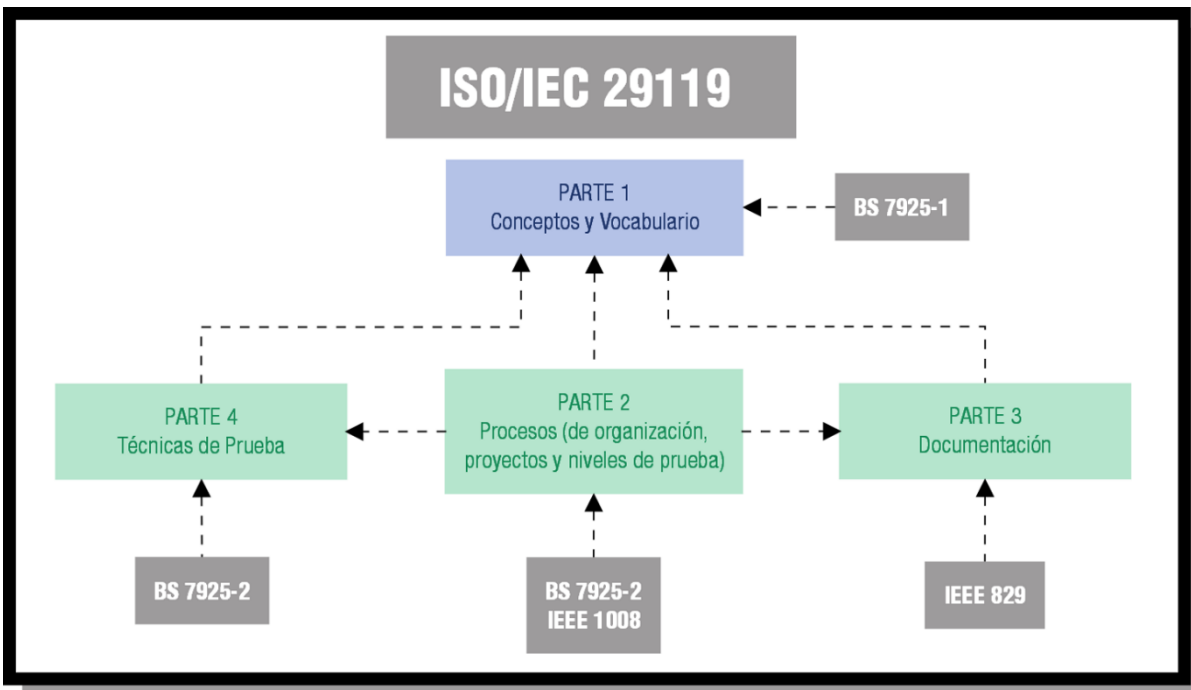


Figura 2: ISO/IEC 29119

Fuente: (Programación, s.f.)

2. Tipos de pruebas funcionales y no funcionales, según estándares de la industria.

2.1 Tipos de pruebas funcionales

2.1.1 Pruebas de humo

Son aquellas pruebas que se realizan para verificar si las funcionalidades con mayor nivel de criticidad funcionan adecuadamente. Mas que aplicar pruebas exhaustivamente lo que se busca es que después de cada compilación sea lo primero que se realice ya que se considera que es una de las pruebas funcionales con mayor importancia.

El resultado de la prueba puede ser exitoso, de ser así quiere decir, que se conseguirá una compilación estable. En caso contrario, debiese ser devuelto al equipo de desarrollo para que solucionen los inconvenientes y generen una nueva compilación.



2.1.2 Pruebas de regresión

Dentro del desarrollo de un sistema, encontraremos a menudo que es necesario modificar o mejorar ciertas funcionalidades. Como consecuencia, de estas intervenciones en el código, es muy posible que se generen errores que afecten su comportamiento de forma inesperada, tanto en lugar del cambio como en otras partes del sistema. La finalidad de las pruebas de regresión apunta a garantizar que se han eliminado los errores y que no han sido introducidos nuevos errores de forma accidental en el sistema como consecuencia del cambio.

2.1.3 Pruebas de aceptación

El concepto de pruebas de aceptación en la Ingeniería de software, se realizan para determinar el nivel de confianza de un sistema, partes de él o bien sus características no funcionales.

El usuario tendrá un nivel de confianza en el sistema que estará relacionado con el grado de concordancia que se logre tanto con las necesidades, requerimientos y procesos de negocio requeridos por él. En base a estas consideraciones el usuario definirá si acepta o no el sistema que se le está entregando.

Por lo mismo, es que estas pruebas por lo general son responsabilidad de los clientes o usuarios del sistema, lo que no impide que otros interesados del proyecto participen.

¿Qué es el Testing de aceptación? Según (PMOinformatica, 2016) encontramos distintos conceptos que pasaremos a revisar:

- En Ingeniería y sus diversas disciplinas, el Testing de aceptación se realiza para determinar si los requerimientos de una especificación o contrato han sido cumplidos.

- En la Ingeniería de sistemas, las pruebas de aceptación de software implican pruebas de caja negra antes de su entrega definitiva.

- Por su parte, el International Software Testing Qualification Board (ISTQB) define la "Aceptación" como: Pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema.

- Las pruebas de aceptación a menudo también se les denominan: Pruebas de aceptación de usuario (UAT), pruebas de usuario final, pruebas de aceptación operacional o pruebas de campo.

- Estas pruebas son fundamentales, para asegurar el éxito de la implementación final de un proyecto de ingeniería de software. Para lo cual, deben incluirse obligatoriamente en el plan de pruebas de software.



2.1.4 Pruebas alfa testing

Según la definición de (Reqtest, 2019):

Son aquellas pruebas internas, en las que se configura un laboratorio de pruebas. Para que, los desarrolladores y evaluadores (que no formaban parte del equipo del proyecto de desarrollo de software) prueben la aplicación. En este tipo de prueba, los desarrolladores toman nota de las acciones del usuario y describen cualquier desviación del comportamiento normal para solucionar problemas.

La prueba alfa es un tipo de prueba en la que:

- Se invita a los usuarios que asistan al laboratorio en que se está desarrollando el software.
- Los desarrolladores anotan las acciones que efectúa el usuario en esta prueba.
- Se presta atención a cualquier anomalía que presente el sistema con respecto al comportamiento normal.
- Los desarrolladores ajustan todos los inconvenientes hallados dentro del sistema.

2.1.5 Pruebas Beta Testing

Según (Reqtest, 2019)

Es un enfoque para obtener comentarios de una muestra de usuarios externos a la empresa. Esta prueba ayuda a que la aplicación se pruebe en un entorno real. Los probadores beta, realizan pruebas de software en un escenario del mundo real. Los desarrolladores solucionan los problemas descubiertos durante las pruebas beta antes de que el software se lance al mercado.

En Beta Testing:

- Con la finalidad de que los usuarios finales del sistema lo utilicen en un entorno real, es que se entrega una versión beta a una cierta cantidad de usuarios que no se encuentran en la empresa.
- A este grupo de usuarios, que se les ha compartido esta versión, se les conoce como probadores beta; prueban el software en un escenario del mundo real.
- Entregan observaciones a los desarrolladores con la finalidad de contribuir a solucionar cualquier inconveniente previo al lanzamiento del software al mercado.

Pruebas alfa vs beta:

Prueba alfa	Prueba Beta
Esta prueba implica pruebas tanto de caja blanca como de caja negra .	Esta prueba solo implica pruebas de caja negra.
Los empleados de la empresa de desarrollo de software donde se está construyendo el software realizan pruebas alfa.	Los usuarios externos que no forman parte de la empresa de desarrollo de software realizan pruebas beta.
La prueba alfa se lleva a cabo al final de los desarrolladores.	La prueba beta se lleva a cabo en el extremo de los usuarios y con la plataforma de su elección.
La confiabilidad y seguridad del software no están cubiertas en las pruebas alfa.	En las pruebas beta, se comprueban la confiabilidad, seguridad y solidez del software.
Las pruebas alfa ayudan a probar la calidad del software antes de aprobarlo para las pruebas beta.	Las pruebas beta ayudan a evaluar el software si está listo para ser lanzado al mercado completo o no.
Para las pruebas alfa, se requiere un laboratorio de pruebas para configurar el entorno de prueba.	En las pruebas beta, no existe tal requisito del entorno de prueba o la configuración del laboratorio.
El ciclo de ejecución de las pruebas alfa puede extenderse a una mayor duración.	Las pruebas beta comparativamente requieren menos tiempo para ejecutarse.
Los desarrolladores pueden abordar rápidamente los problemas o errores encontrados durante las pruebas alfa.	Los problemas encontrados en las pruebas beta se resuelven en la versión futura del software.

Tabla 4: Diferencias comunes entre pruebas alfa y pruebas beta

Fuente: (Reqttest, 2019)

2.2 Tipos de pruebas no funcionales

2.2.1 Pruebas de seguridad:

Son todas aquellas pruebas que en conjunto buscan encontrar fallas y vulnerabilidades como, por ejemplo: accesos no autorizados, ataque de denegación de servicio, entre otros. Lo que buscan es bajar el impacto que puede haber ante ataques a las aplicaciones web y a la pérdida de información valiosa.



2.2.2 Pruebas de usabilidad

Por medio de pruebas de usabilidad por parte del usuario, se establece la facilidad de uso, efectividad y satisfacción. Es una técnica que se realiza por medio del diseño de interacciones, siempre dentro de un grupo específico de usuarios que evalúan el sistema.

2.2.3 Pruebas de rendimiento

El objetivo de las pruebas de rendimiento consiste en establecer el rendimiento del sistema bajo una carga de trabajo determinada empleando distintos tipos de pruebas de rendimiento como son: Pruebas de carga, de estrés y de estabilidad.

En estas pruebas se medirán la velocidad de procesamiento y el tiempo de respuesta del sistema. Fuente: (Montecinos, 2020)

2.2.4 Pruebas de escalabilidad

A diferencia de las pruebas de rendimiento, las pruebas de escalabilidad apuntan al análisis de la forma en que responde un sistema a las variaciones en el número de usuarios simultáneos. Por medio de estas pruebas, se determina que los sistemas amplíen o reduzcan. Y, ajusten la cantidad de recursos (hardware) que se utiliza, para asegurar que los usuarios noten un rendimiento coherente y estable a pesar del número de usuarios simultáneos, sin necesidad de modificar el código.

Un punto importante lo describe (Lee, 2020):

Las pruebas de escalabilidad, también se pueden realizar en hardware, recursos de red y bases de datos. Para ver cómo responden a un número variable de solicitudes simultáneas. A diferencia de las pruebas de carga, donde se analiza cómo responde el sistema a varios niveles de carga, las pruebas de escalabilidad analizan el grado de escala del sistema en respuesta a varios niveles de carga. Este último es especialmente importante en entornos en contenedores.

2.2.5 Pruebas de mantenibilidad

Fundamentalmente, radican en estimar que tan fácil es efectuar el mantenimiento de un sistema o aplicación.

2.2.6 Pruebas de inestabilidad

También conocidas como Soak Testing, se efectúan generalmente para establecer si la aplicación puede soportar una carga esperada continua. Habitualmente, esta prueba se efectúa para averiguar si hay alguna fuga de memoria en la aplicación.

2.2.7 Pruebas de portabilidad

Las pruebas de portabilidad son aquellas que determinan la facilidad o dificultad con la que un producto de software se puede cambiar de un entorno a otro. El entorno está haciendo referencia a distintos sistemas operativos, hardware, bases de datos o navegadores.

3. Pruebas funcionales en proyectos de desarrollo de software, según estándares de la industria.

3.1 Aplicación de pruebas funcionales

Las pruebas funcionales, son conocidas también como pruebas de caja negra o de comportamiento. En este tipo de pruebas, lo que se hace es analizar los datos tanto de entrada como de salida, no centrados en la forma en que se generan los resultados. Los casos de prueba de manera general son definidos con anticipación y previo al comienzo de las pruebas, siguiendo los pasos establecidos o plan de pruebas.

Algunos pasos generales para realizar pruebas funcionales son:

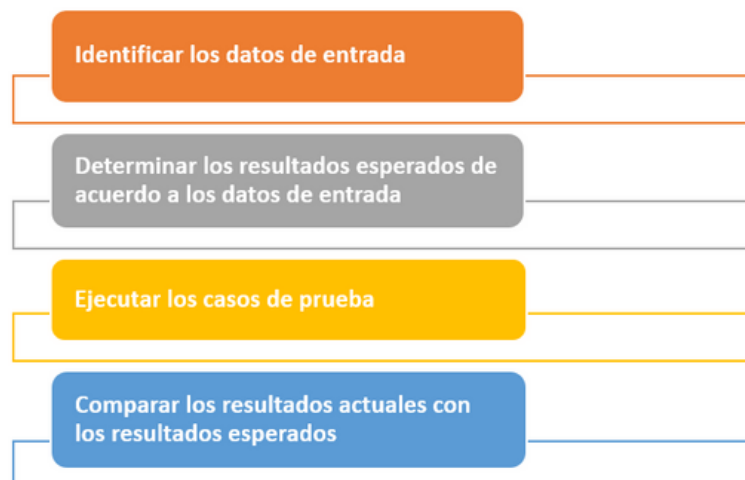


Figura 1: Pasos generales para realizar pruebas funcionales

Fuente: (<https://www.diariodeqa.com/>, 2020)

4. Pruebas no funcionales en proyectos de desarrollo de software, según estándares de la industria.

4.1 Aplicación de pruebas no funcionales

Las pruebas no funcionales están relacionadas con la verificación de los requisitos, características generales y restricciones de una aplicación.

Existen muchas características, que un software debe poseer para que sea considerado de calidad, dentro de ellas podemos encontrar tales como:



Figura 2: Algunas pruebas no funcionales

Fuente: (Gomez, s.f.)

Principales actividades que se deben realizar en la ejecución de las pruebas:

- Establecer su alcance.
- Diseñar el Plan de Prueba (estrategia, condiciones de término de cada ciclo, parámetros de calidad, etc.).
- Diseñar los casos de pruebas y los escenarios.
- Ejecutar las pruebas.
- Realizar los informes con los resultados de la ejecución de las pruebas.
- Posterior a un análisis de los resultados, se genera un informe con todas aquellas incidencias que han sido detectadas.

Pasaremos a revisar algunas pruebas no funcionales y como se ejecutarían en base a (Greensqa, 2018):

1. Carga – Load test

Con esta prueba, la empresa puede conocer la cantidad de usuarios que soporta su producto, en un lapso determinado, según los usuarios que espera tender el cliente en su sistema, sin forzarlo a una capacidad mayor a la esperada.



Figura 3: Pruebas de carga
Fuente: (Greensqa, 2018)

Por ejemplo: Se necesita probar un puente para saber su comportamiento con 100 personas sobre él (el dato de la cantidad es determinado por el cliente). Los arquitectos de pruebas, lo que hacen es dividir este grupo total en grupos más pequeños, que vayan ingresando al puente de manera paulatina en un lapso (5 personas inician, al minuto ingresan otras 5 y así sucesivamente hasta completar las 100 personas) y con una actividad específica como “saltar”. Al final, permite evaluar el comportamiento del puente, si su estructura y estado final tuvieron el comportamiento saludable para soportar esta actividad.

2. Capacidad o rendimiento – Capacity Test

El objetivo de las pruebas de capacidad es evaluar el punto de quiebre de la aplicación. Es determinar la carga máxima de usuarios, que puede soportar la aplicación (escalabilidad).

Tomando nuevamente el ejemplo del puente, se agregaría más personas por intervalo de tiempo, superando las 100 personas que inicialmente esperábamos que soportara el puente. Ejemplo 150 personas en total, siempre haciendo pausas entre cada grupo que se suman a las que ya están saltando en el puente, llegando hasta el punto puente colapse o se vea seriamente afectado. De esta manera nos daríamos cuenta de que capacidad tiene el sistema para atender usuarios de manera concurrente.

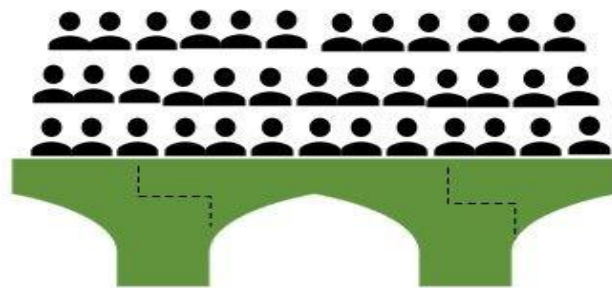


Figura 4: Pruebas de rendimiento

Fuente: (Greensqa, 2018)

3. Estrés – Stress Test

Acá colocamos a prueba la robustez y la confiabilidad del software probado, sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen, el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan bugs (defectos) potencialmente peligrosos y verificar si los mismos pueden recuperar sus recursos físicos de forma autónoma, sin requerir la intervención humana.

Continuando con el ejemplo de personas saltando sobre el puente, en esta prueba colocamos a las mismas 150 personas que participaron en la prueba de capacidad, pero sin pausa entre salto y salto, con el objetivo de evidenciar si el puente puede recuperar su forma después de alterar su estructura durante la prueba.

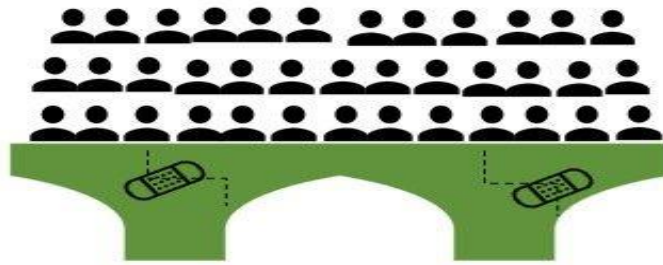


Figura 5: Pruebas de estrés
Fuente: (Greensqa, 2018)

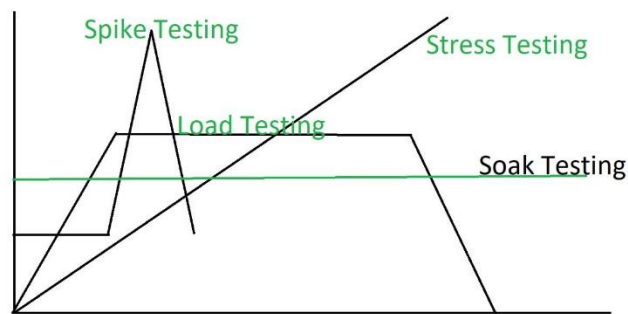


Figura 6: Cuadro comparativo pruebas no funcionales de rendimiento
Fuente: (Geeksforgeeks, 2019)

Stress Testing: Prueba de estrés
Spike Testing: Pruebas de picos
Load Testing: Pruebas de carga
Soak Testing: Prueba de estabilidad



Conclusiones

Como hemos podido revisar esta semana y semanas anteriores, existen muchas pruebas que podemos aplicar a nuestros desarrollos. Cada una con un enfoque y en etapas del proyecto muy diferentes, que nos dan la certeza de que cada uno de los elementos ha sido probado. Hemos detectado y solucionado todas las incidencias o resultados no esperados, tanto en el comportamiento como en las salidas de los procesos que entrega el sistema. Finalizando con las pruebas de aceptación, ya que serán éstas por medio de las cuales el cliente o usuario podrá definir si acepta o no el producto final.



Referencias bibliográficas

Roger S. Pressman, P. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO*. McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.

Federico Toledo Rodríguez, *INTRODUCCIÓN A LAS PRUEBAS DE SISTEMAS DE INFORMACIÓN*. Abstracta, Montevideo, Uruguay, 2014

J. A. Mera-Paz, "ANÁLISIS DEL PROCESO DE PRUEBAS DE CALIDAD DE SOFTWARE", *Ingeniería Solidaria*, vol. 12, no. 20, pp. 163-176, oct. 2016.
doi: <https://repository.ucc.edu.co/handle/20.500.12494/962>