Plan de pruebas - Actividad N°15

*Eduardo Alvarez Godoy, Marco Bravo Castillo y Álvaro Sepúlveda Becerra*

Docente: Teresa Tapia Soto

# Índice

[Índice 1](#_Toc75866793)

[1. Plan de pruebas 3](#_Toc75866794)

[1.1. Propósito 3](#_Toc75866795)

[1.1. Alcance 3](#_Toc75866796)

[1.2. Vista general del sistema 3](#_Toc75866797)

[2. Estrategia de Pruebas 4](#_Toc75866798)

[2.1. Ciclo del plan de pruebas 4](#_Toc75866799)

[**2.1.1.** Planificación 4](#_Toc75866800)

[**2.1.2.** El diseño de las pruebas 4](#_Toc75866801)

[**2.1.3.** Configuración 4](#_Toc75866802)

[**2.1.4.** Ejecución 4](#_Toc75866803)

[**2.1.5.** Evaluación y cierre 4](#_Toc75866804)

[**2.1.6.** Seguimiento y control 5](#_Toc75866805)

[2.2. Herramientas para pruebas 5](#_Toc75866806)

[**2.2.1.** LAMBDATEST 5](#_Toc75866807)

[**2.2.2.** NETSPARKER 5](#_Toc75866808)

[**2.2.3.** JMETER 5](#_Toc75866809)

[**2.2.4.** VISUAL CODE 5](#_Toc75866810)

[**2.2.5.** XAMPP 5](#_Toc75866811)

[**2.2.6.** Mapas de Experiencia 6](#_Toc75866812)

[2.3. Tipos de pruebas 6](#_Toc75866813)

[2.4. Entregables de pruebas 10](#_Toc75866815)

[2.5. Matriz de tipificación de pruebas 10](#_Toc75866816)

[2.6. Técnicas de ejecución de pruebas 11](#_Toc75866817)

[3. Recursos del plan de pruebas 14](#_Toc75866818)

[3.1. Recurso humano 14](#_Toc75866819)

[3.2. Recurso del sistema 14](#_Toc75866820)

[**3.2.1.** Configuración del ambiente de pruebas 15](#_Toc75866821)

[3.3. Herramientas de reportes y control de incidencias 15](#_Toc75866822)

[3.4. Administración de versiones 15](#_Toc75866823)

[**3.4.1.** Herramientas: 16](#_Toc75866824)

[4. Evaluación de pruebas ejecutadas 16](#_Toc75866825)

[4.1. Criterios de inicio de ejecución 16](#_Toc75866826)

[4.2. Criterios de evaluación 17](#_Toc75866827)

[4.3. Criterio de terminación 20](#_Toc75866828)

[4.4. Criterios de suspensión 21](#_Toc75866829)

[5. Anexos 21](#_Toc75866830)

# Plan de pruebas

## Propósito

Este documento tiene como propósito establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación del plan de pruebas; incluye responsabilidades de cada una de las tareas, los recursos y los prerrequisitos que deben ser considerados en el esfuerzo de cada una de las pruebas, permitiendo garantizar el cumplimiento de los requerimientos planteados en el marco del desarrollo del proyecto denominado “PortArt”.

## Alcance

Este documento de plan de pruebas se convierte en una guía para desarrollar de una forma organizada las diferentes actividades que se realizarán en el proceso del plan de pruebas en el desarrollo del proyecto denominado “PortArt”

## Vista general del sistema

Descripción resumida del contenido de cada una de las secciones que siguen, y explicación de la forma en que está organizado el presente documento:

**Estrategia de Pruebas:**

En este capítulo se presenta una perspectiva general de la estrategia que se va a seguir para analizar, diseñar, implementar y ejecutar las pruebas del proyecto “PortArt”. Así mismo se definirá qué tipos de pruebas se van a realizar y cómo se ejecutarán.

**Recursos del Plan de Pruebas:**

Este capítulo identifica los recursos humanos y no humanos (hardware, software, herramientas de soporte, configuración de entorno de pruebas, entre otros), necesarios para desarrollar el proceso del plan de pruebas de la solución del Sistema de Notificación en Línea.

**Evaluación de Pruebas Ejecutadas:**

En este capítulo se describe de los métodos de evaluación de las pruebas ejecutadas, de tal forma que permitirá evaluar los grados de aceptación de las pruebas.

# Estrategia de Pruebas

## Ciclo del plan de pruebas

El proyecto se dividirá en varias etapas, con el objetivo de cumplir todos los estándares de prueba y acorde al formato establecido para la confección del informe se pasará por un par de etapas las cuáles serán las siguientes:

### Planificación

Esta será una de las etapas más importantes al momento del desarrollo del sistema y las respectivas pruebas que se realizarán, es una parte muy importante y para esta etapa hay que tener en cuenta ciertos puntos para tener claro que abarca exactamente:

* Se planifican pruebas personalizando los estándares específicamente para el proyecto de notificaciones.
* Se definen niveles de pruebas a aplicar.
* Se especifican las técnicas a utilizar.
* Se establece el tiempo para la ejecución de cada una de las pruebas.
* Uso de herramientas.

Y en esta etapa se valoran cosas como el alcance que tendrá la aplicación, la complejidad de los procesos y las plataformas en las que será utilizable el sistema. Además de los conocimientos y la formación de quienes realizarán las respectivas pruebas.

También hay que tener en cuenta como las pruebas estarán presente durante todo el ciclo de vida del desarrollo de la solución, siempre existirán errores y que probar exhaustivamente es algo imposible, lo que entorpecería enormemente el proyecto. Y, obviamente, que el programador no sea quien realice las pruebas.

### El diseño de las pruebas

Para el diseño de las pruebas se tendrán en cuenta los aspectos que permitan la detección de errores o defectos en el periodo de desarrollo de software, la realización de pruebas propias de verificación y validación de datos.

A grandes rasgos estará conformado por el Alcance, Inventario de pruebas, Resultado de la ejecución de pruebas, Ciclo de la Prueba.

### Configuración

El sistema a desarrollar no necesita una configuración enorme a comparación otros servicios que podrían utilizarse o diseñarse, mientras los usuarios cuenten con un navegador en las versiones más actualizadas como se nombra previamente en los requerimientos No Funcionales.

### Ejecución

En la ejecución a medida se vayan realizando las pruebas, estas serán documentadas como es correspondiente según la metodología seleccionada que fue SCRUM, en las reuniones diarias se mostrarán los avances, errores que se encuentren en el camino los desarrolladores. Con la facilidad de en conjunto poder encontrar una fácil solución para los errores detectados.

Una vez finalizada una prueba, esta será documentada para futuros cambios y que puedan aplicarse correctamente.

### Evaluación y cierre

Para la evaluación y cierre, se mostrarán los resultados finales de todas las pruebas.

### Seguimiento y control

Se realiza el seguimiento correspondiente a las pruebas realizadas.

## Herramientas para pruebas

Las herramientas utilizadas serán:

### LAMBDATEST

**Características** de LambdaTest: LambdaTest es una plataforma de prueba de navegadores basada en la nube que ayuda en las pruebas de compatibilidad, tanto manuales como de navegador. Los usuarios pueden realizar pruebas interactivas en vivo de su sitio web o aplicación web con una combinación de más de 2000 navegadores y sistemas operativos diferentes sin salir de su propio navegador.

### NETSPARKER

**Características** de Netsparker: Netsparker Security Scanner es una solución precisa que identifica automáticamente XSS, Inyección de SQL y otras vulnerabilidades en aplicaciones web.

La tecnología de análisis basada en pruebas única de Netsparker te permite dedicar más tiempo a corregir las fallas informadas explotando automáticamente las vulnerabilidades identificadas de forma segura y de solo lectura, y también presenta una prueba de explotación. Por lo tanto, puedes ver de inmediato el impacto de la vulnerabilidad y no tienes que verificar manualmente.

### JMETER

**Características** de Jmeter: es una herramienta de carga para llevar a cabo simulaciones sobre cualquier recurso de Software.

Inicialmente diseñada para pruebas de estrés en aplicaciones web, hoy en día, su arquitectura ha evolucionado no sólo para llevar a cabo pruebas en componentes habilitados en Internet (HTTP), sino además en Bases de Datos, programas en Perl, requisiciones FTP y prácticamente cualquier otro medio.

Posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de requisiciones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción.

En este sentido, simula todas las funcionalidades de un Navegador ("Browser"), o de cualquier otro cliente, siendo capaz de manipular resultados en determinada requisición y reutilizarlos para ser empleados en una nueva secuencia.

### VISUAL CODE

**Características** de VisualCode: Visual Studio Code es un editor de código fuente que permite trabajar con diversos lenguajes de programación, admite gestionar tus propios atajos de teclado y refactorizar el código. Es gratuito, de código abierto y nos proporciona una utilidad para descargar y gestionar extensiones con las que podemos personalizar y potenciar esta herramienta.

### XAMPP

**Características** de XAMPP: XAMPP es una herramienta de desarrollo y servidor independiente de software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.   
Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

### Mapas de Experiencia

Características de los mapas de experiencia: los mapas de experiencia son una herramienta de Design Thinking que permite plasmar en un mapa, cada una de las etapas, interacciones, canales y elementos por los que atraviesa nuestro cliente desde un punto a otro de nuestro servicio, de modo en que nos dará a conocer el cómo se sintió tras utilizar el sistema.

## Tipos de pruebas

Las pruebas que se realizarán serán algunas de tipo Documentación, Pruebas de Carga, Pruebas de Estrés, Pruebas de compatibilidad, Pruebas de Portabilidad, Pruebas de Interfaz de Usuario, Pruebas de Rendimiento y un par más. Con el fin de poder detectar los errores durante el desarrollo y tomar las correspondientes medidas. A continuación, se explicará brevemente cada tipo de pruebas:

-Prueba de documentación.

-Prueba de carga.

-Prueba de estrés.

-Prueba de compatibilidad.

-Prueba de usabilidad.

-Prueba de integridad de datos y base de datos.

-Prueba de seguridad y control de acceso

### Prueba de documentación.

Las pruebas de documentación tienen como objetivo el mantener la documentación de una forma ordenada y organizada desde el inicio hasta el mantenimiento del software, de modo de que se obtiene una disponibilidad continua de los documentos que se necesitan en el momento en que se les necesita.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Mantener la información ordenada y organizada para cuando esta sea necesaria. |
| Estrategia: | -Se efectuarán al finalizar cada proceso o desarrollo de alguna actividad.  -Se mantendrá la documentación clasificada y ordenada dependiendo de la función que cumpla dentro del proyecto.  -Se solicitará a los desarrolladores y analistas que detallen la información de los procesos y funciones realizadas. |
| Herramientas requeridas: | -VisualCode.  -Gestor de archivos y documentos.  -GitHub. |
| Observaciones | -VisualCode se utilizará de modo de que los desarrolladores muestren sus códigos y los guarden con comentarios que expliquen su uso y funcionamiento. |

* + 1. Prueba de Carga.

Éste es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad esperada de peticiones. Esta carga puede ser el número esperado de usuarios concurrentes, utilizando la aplicación que realizan un número específico de transacciones, durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | -Medir la cantidad de personas que pueden interactuar con el sistema al mismo tiempo. |
| Estrategia: | -Agregar en JMeter 100 números de subprocesos con recuento de bucles de 10 y periodo de aceleración de 100.  -Agregar los elementos de JMeter con los valores predeterminados.  -Agregar los resultados gráficos.  -Interpretar la prueba y los resultados.  -En caso de no funcionar, probar de nuevo con cifras más bajas. |
| Herramientas requeridas: | -JMeter |
| Observaciones |  |

* + 1. Prueba de Estrés.

Se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | -Medir el rendimiento de la aplicación, de modo que se sobrecargará el sistema para saber cuándo y cómo fallará. |
| Estrategia: | -Se comienza agregando 10 usuarios en grupo de hilos en JMeter.  -En contador de bucle se pone la opción “Sin fin”, para que este no termine hasta que nosotros lo indiquemos.  -En número de hilos pondremos 10.  -Agregamos la petición HTTP.  -Agregamos como forma de medición de las peticiones un árbol de resultados, un gráfico de resultados y un informe agregado para monitorear los resultados.  -Iniciar la prueba.  -Ver los resultados y saber en qué momento el sistema se detuvo. |
| Herramientas requeridas: | JMeter. |
| Observaciones |  |

* + 1. Prueba de Compatibilidad.

Las pruebas de compatibilidad son pruebas que verifican el funcionamiento del sistema en diferentes entornos. Por ejemplo, se puede probar un sistema en los distintos navegadores o en distintas versiones de máquina virtual o en distintos sistemas operativos o en distintos dispositivos, en nuestro caso las pruebas serán orientadas a Windows 10, Windows 8 y Windows 7, y en los navegadores Chrome, Opera, Internet Explore, Microsoft Edge y Mozilla Firefox.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Comprobar que la aplicación, junto con todas sus características y funciones sean compatibles con todos los tipos de navegadores web. |
| Estrategia: | -Ingresar a LambdaTest, y colocar la URL de la aplicación web.  -Seleccionar los sistemas operativos con los navegadores más conocidos.  -Iniciar la prueba y corroborar que el sistema funcione correctamente.  -En caso de error, revisar y corregir el código y volver a probar. |
| Herramientas requeridas: | -LambdaTest |
| Observaciones |  |

* + 1. Prueba de Usabilidad.

Son un método para evaluar la experiencia del usuario de un producto o sitio web. Al probar la usabilidad con un grupo representativo de usuarios o clientes se determina si los usuarios reales pueden usar el sitio web de manera fácil e intuitiva.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | -Validar que el sistema sea fácil e intuitivo de usar. |
| Estrategia: | -Se usarán cuatro métricas principales para medir la usabilidad del sistema, la exactitud, el tiempo requerido, lo que recordó el usuario del sistema y su respuesta emocional.  -Estos puntos serán implementados en cada uno de los aspectos utilizados en la prueba.  -Se realizarán encuestas para conocer la experiencia del usuario tras probar el prototipo del sistema.  -En caso de que algún aspecto o característica del sistema sea criticado negativamente en varias ocasiones se realizarán cambios adecuándose a los requerimientos de los usuarios. |
| Herramientas requeridas: | -Encuestas.  -Mapas de empatía y de experiencia del usuario.  -Prototipos. |
| Observaciones |  |

* + 1. Prueba de Integridad de datos y base de datos.

Pruebas de integridad de base de datos son pruebas de los métodos y procesos utilizados para acceder y gestionar datos (base de datos), para asegurar que los métodos de acceso, los procesos y las reglas de los datos funcionan como se espera y que, durante el acceso a la base de datos, los datos no se corrompan, sean borrados, modificados o creados de forma inesperada y que se encuentren disponibles al menos un 99.96% del tiempo.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | -Que los datos de la base de datos sean actualizados y guardados en forma correcta y segura. |
| Estrategia: | -Se realizarán pruebas con Netsparker para probar la seguridad de la base de datos.  -Se probarán las funciones del CRUD en el prototipo conectado con XAMPP, ingresando todo tipos de datos y se irán revisando si es que los datos aparecen actualizados o no en la base de datos.  -En caso de que existiera algún problema se revisa el código, evaluando y corrigiendo los errores, para posteriormente ejecutar de nuevo la prueba. |
| Herramientas requeridas: | -Netsparker  -Pruebas funcionales del prototipo conectadas con XAMPP |
| Observaciones |  |

* + 1. Prueba de Seguridad y control de acceso.

La prueba de seguridad y control de acceso se encarga de verificar que solo un usuario o actor sea capaz de acceder a las funciones y los datos que su usuario tiene permitido y en caso de que exista algún usuario que rompa los términos y condiciones de uso, serán baneados permanente o temporalmente dependiendo de la gravedad de la falta.

|  |  |
| --- | --- |
| Objetivo de la Prueba: | -Que los usuarios solo puedan realizar las funciones respectivas al tipo de usuario al cual pertenecen. |
| Estrategia: | -Identificar los tipos de usuarios y las funciones que pueden realizar.  -Crear pruebas para los distintos tipos de usuarios.  -Modificar los usuarios y repetir las pruebas. |
| Herramientas requeridas: | -Netsparker.  -Pruebas funcionales de seguridad. |
| Observaciones |  |

## Entregables de pruebas

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **ENTREGABLES** |
| -Pruebas de documentación. | * Se entregará un documento de pruebas de documentación, la cual incluye los resultados y la verificación de los documentos. |
| -Pruebas de carga. | * Se entregará un documento de pruebas de carga, la cual incluye los resultados en modo de gráficos otorgando la información otorgada por JMeter |
| -Pruebas de estrés. | * Se entregará un documento de pruebas de estrés, la cual incluye los resultados en modo de gráficos otorgando la información otorgada por JMeter |
| -Pruebas de compatibilidad. | * Se entregará un resumen acerca de la funcionalidad y compatibilidad del software con cada uno de los navegadores y sistema operativos utilizados. |
| -Pruebas de usabilidad. | * Se entregará un documento con las recomendaciones y críticas otorgadas por los usuarios tras la realización de encuestas. * Mapa de experiencia del usuario. |
| -Pruebas de integridad de datos y base de datos. | * Se entregará un documento con la información otorgada por Netsparker y con un resumen acerca de las funcionalidades de la base de datos. |
| -Pruebas de seguridad y control de acceso | * Informes de vulnerabilidades. * Resultado de pruebas funcionales de seguridad. |

## Matriz de tipificación de pruebas

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **TIPO DE PRUEBA** |
| -Pruebas de documentación. | Manual. |
| -Pruebas de carga. | Automática |
| -Pruebas de estrés. | Automática |
| -Pruebas de compatibilidad. | Manual y automática |
| -Pruebas de usabilidad. | Manual. |
| -Pruebas de integridad de datos y base de datos. | Manual y automática. |
| -Pruebas de seguridad y control de acceso | Manual y automática. |

## Técnicas de ejecución de pruebas

|  |  |  |
| --- | --- | --- |
| **TIPO DE PRUEBAS** | **TÉCNICA DE EJECUCIÓN** | **HERRAMIENTAS A UTILIZAR** |
| Pruebas de documentación. | La prueba será ejecutada tras la finalización de cada actividad del proyecto, y la técnica a utilizar será la siguiente:   1. Crear un documento en Word, detallando los detalles de la actividad realizada. 2. En caso de que la documentación sea acerca de algún desarrollo de código, se especificará dentro del mismo código a través de un comentario la función que cumple. 3. Los archivos serán guardados en carpetas dentro de las ramas respectivas en GitHub. | -**VisualCode.**  **-Gestor de archivos y documentos.**  **-GitHub.** |
| Pruebas de carga. | Las pruebas de cargas involucran los siguientes pasos:   1. Tras abrir Jmeter elegimos la opción de ‘plan de prueba’ y luego agregamos subprocesos. 2. Escribimos el número de usuarios que queremos que hagan las peticiones, en este caso 10 y en periodo de aceleración 120, con un recuento de bucles 3. Agregamos los elementos de Jmeter con los valores predeterminados y la dirección de nuestra aplicación web. 4. Agregar los resultados gráficos, para poder ver los resultados de la prueba. 5. Revisar e interpretar los resultados otorgados. | **-Jmeter.** |
| Pruebas de estrés. | Las pruebas de estrés involucran los siguientes pasos:   1. Tras abrir Jmeter elegimos la opción de ‘plan de prueba’ y luego agregamos subprocesos. 2. Escribimos el número de usuarios que queremos que hagan las peticiones, en este caso 20 y en periodo de aceleración 120, y la cuenta de bucles la configuramos de modo que se encuentre ‘Marcado para siempre’ 3. Agregamos los elementos de Jmeter con los valores predeterminados y la dirección de nuestra aplicación web. 4. Agregar los resultados gráficos, para poder ver los resultados de la prueba. 5. Revisar e interpretar los resultados otorgados. | **-Jmeter.** |
| Pruebas de compatibilidad. | Las pruebas de estrés involucran los siguientes pasos a través de LambdaTest:   1. En el apartado de ‘Real Time Test’ podemos hacer la prueba en tiempo real de la aplicación. 2. Ingresamos el tipo de dispositivo en el cual nosotros queremos hacer la prueba. 3. Seleccionamos el navegador junto a la versión deseada. 4. Y seleccionamos el sistema operativo con la resolución de pantalla e iniciamos la prueba. 5. Revisamos las distintas funcionalidades y páginas de nuestro sistema y comprobamos que todo esté funcionando correctamente. 6. Podemos escribir detalladamente y sacar capturas de pantallas de los errores y problemas que podamos encontrar y poder guardarlos dentro de las funciones que la herramienta nos entrega 7. Revisar y corregir los errores y volver a realizar las pruebas de compatibilidad. | **-LambdaTest** |
| Pruebas de usabilidad. | Las pruebas de usabilidad aplicadas con el prototipo y las encuestas involucran los siguientes pasos:   1. Elaboración del prototipo. 2. Elaboración de encuestas. 3. Pruebas del prototipo mediante la utilización de usuarios comunes para que puedan comprobar sus funciones y los estilos de las páginas. 4. Recopilación de los datos de las encuestas. 5. Análisis de los resultados de las encuestas. 6. Correcciones de errores y sugerencias. | **-Prototipos**  **-Encuestas.** |
| Pruebas de integridad de datos y base de datos. | Las pruebas de integridad de datos y base datos involucran los siguientes pasos:   1. Se realizan pruebas con Netsparker para probar la seguridad de la base de datos. 2. Se prueban las funciones de CRUD con el prototipo conectado a XAMPP y se ingresan datos con los distintos usuarios y se revisa si se actualizaron. 3. Recopilación de los resultados. 4. Correcciones de errores e incidencias 5. Repetición de prueba. | **-Netsparker.**  **-Pruebas con el prototipo conectado a XAMPP** |
| Pruebas de seguridad y control de acceso | Las pruebas de seguridad y control de acceso fueron hechas mediante la utilización del software Netsparker para probar la seguridad del sistema y pruebas funcionales de seguridad hechas con distintos usuarios, y los pasos fueron los siguientes:   1. Configuración del ambiente de pruebas. 2. Ingresar la dirección de la aplicación web. 3. Se definen los parámetros de escaneo y lo ponemos en predeterminado. 4. Ejecución de los casos automáticos de pruebas de seguridad. 5. Recopilación de resultados. 6. Revisión manual de los permisos de los usuarios en el prototipo. 7. Correcciones de errores e incidencias 8. Repetición de prueba. | **-Netsparker.**  **-Pruebas Funcionales de seguridad.** |

# Recursos del plan de pruebas

## Recurso humano

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **PERFIL DEL RECURSO HUMANO** |
| Pruebas de documentación. | Desarrollador.  Analista. |
| Pruebas de carga. | Desarrollador.  Analista. |
| Pruebas de estrés. | Desarrollador.  Analista. |
| Pruebas de compatibilidad. | Desarrollador.  Analista. |
| Pruebas de usabilidad. | Desarrollador.  Analista.  Usuario funcional. |
| Pruebas de integridad de datos y base de datos. | Desarrollador.  Analista.  Encargado de base de datos. |
| Pruebas de seguridad y control de acceso | Desarrollador.  Analista.  Encargado de base de datos. |

## Recurso del sistema

Las pruebas se realizarán en un ambiente controlado por el equipo de pruebas, a continuación, se describirán las características de la infraestructura del ambiente de pruebas.

|  |  |  |
| --- | --- | --- |
| **DESCRIPCIÓN** | **FUNCIONALIDAD** | **CANTIDAD** |
| **Estaciones de trabajo.** | Con acceso a GitHub, para poder mantener actualizada y compartida la información de las pruebas con todos los integrantes del grupo. | **4** |
| **Software.** | Herramientas para lo codificación, en nuestro caso Visual Code, y XAMPP para controlar las bases de datos. | **1** |
| **Herramientas de pruebas.** | LambdaTest, Netsparker, Jmeter, XAMPP. | **1** |
| **Repositorio de Documentos.** | GitHub. | **1** |

### Configuración del ambiente de pruebas

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **Configuración** | **Software Instalado y configurado** | **Cantidad** |
| Estaciones de Trabajo | Equipos con mínimo:  Procesador Intel Pentium G4560  Memoria RAM: 8 GB  RED: Acceso a la red local (Para los equipos dentro del área de trabajo de la UT), Acceso a Internet (Para los equipos fuera del área de trabajo de la UT). | Sistema Operativo: Windows 10  Internet Explorer 9.0, Mozilla 2.0, Office 2016. | 4 |

## Herramientas de reportes y control de incidencias

La herramienta a utilizar para la realización del reporte, seguimiento y control de errores es Sentry, la cual proporciona un rastreador de errores de código abierto para supervisar y responder a errores y fallas en cualquier lugar de tu aplicación en tiempo real. Sentry agrega y suma seguimientos "del contexto a la aplicación" importantes e informa a los usuarios, servicios y servidores que se vieron afectados. Por primera vez, cada desarrollador puede obtener una visibilidad instantánea de cómo el código de producción afecta a los usuarios reales y automáticamente selecciona, clasifica y resuelve problemas como parte del flujo de trabajo existente.

## Administración de versiones

La administración de versiones que se probarán será el mecanismo ideal, para controlar los release de pruebas y los cambios que estos sufrirán en la etapa de corrección de incidencias reportadas y contemplarán las siguientes etapas:

1. Entrega de la versión para las pruebas.
2. Creación de incidencias en la herramienta.
3. Análisis y desarrollo de incidencias.
4. Pruebas de correcciones de incidencias.
5. Aprobación del release.

Y se deben considerar algunas buenas prácticas para poder mantener una buena comunicación acerca de los cambios que se han realizado dentro del proyecto, las cuales son:

* Decidir cuántas versiones del archivo deben mantenerse, cuáles, durante cuánto tiempo y cómo serán organizadas.
* Identificar las versiones más importantes que deben ser guardadas.
* Registrar las versiones y estado de un archivo, tales como “borrador”,” Provisorio”,” Final”, etc.
* Registrar las relaciones entre ítems cuando sea necesario.
* Sincronizar los archivos ubicados en diferentes locaciones de forma regular.
* Mantener un único archivo o rama maestra, en un formato apropiado para evitar problemas de control de versiones que puedan surgir cuando se trabaja en diferentes archivos de forma paralela.
* Identificar una única ubicación para el almacenamiento de los archivos que constituyen archivos maestros.

### Herramientas:

**GIT**

Es un sistema de control de versiones distribuido de código abierto desarrollado por Linus Torvalds, el creador de Linux. El control de versiones distribuido permite a los desarrolladores descargar un software, realizar cambios y subir la versión que han modificado.

**Ventajas de GIT:**

**Trabajo en equipo.** Permite que varios desarrolladores trabajen al mismo tiempo y en paralelo en un proyecto con un acceso compartido, pero sin estar físicamente cerca, así como identificar qué usuario y cuándo ha realizado cada modificación.

**Mayor autonomía**. Cada desarrollador cuenta con una copia local de todo el proyecto y de los cambios generados, lo que le permite trabajar de forma individual y a su propio ritmo, en cualquier momento y lugar.

**La velocidad** de Git, ya que necesita menos capacidad de procesamiento y gestión al poder realizar las operaciones en local.

**Sin conexión a la red**. El desarrollador puede trabajar o enviar cambios de código al sistema sin necesidad de estar conectado a Internet o a cualquier otra red, al disponer de un repositorio local.

**Estructura en árbol**. Esta característica hace posible que los desarrolladores puedan trabajar en diferentes ramas de un proyecto, pero sin modificar en el código base principal.

**Permite comparar,** fusionar o restaurar versiones de una aplicación y contar con una copia del código fuente para volver atrás ante cualquier imprevisto.

**Es software libre y open source.**

# Evaluación de pruebas ejecutadas

En este capítulo se mostrarán los criterios de ejecución, evaluación, terminación y suspensión de las pruebas.

## Criterios de inicio de ejecución

Las condiciones mínimas que se deben presentar para iniciar la ejecución de las pruebas son:

* El entorno de pruebas es el adecuado para el tipo de pruebas al iniciar
* Todas las herramientas se encuentran disponibles para el equipo de pruebas.
* Todos los recursos humanos se encuentran disponibles.
* El sistema se encuentra funcional

## Criterios de evaluación

Los criterios de evaluación estarán dados de forma independiente para cada tipo de pruebas; el siguiente cuadro muestra los criterios de evaluación generales de las pruebas ejecutadas.

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **CRITERIOS DE EVALUACIÓN** |
| Pruebas de Documentación. | * Detectar errores en la ejecución de las pruebas. * El 90% de las pruebas realizadas deben ser exitosas. * Deben generar informes continuamente, incluso si el |
| Pruebas de carga. | * Detectar errores en la ejecución de las pruebas. * Que los informes y reportes generados por las herramientas contengan las mínimas variables que permitan un análisis acertado. * El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de estrés. | * Detectar errores en la ejecución de las pruebas. * Que los informes y reportes generados por las herramientas contengan las mínimas variables que permitan un análisis acertado. * Tener en cuenta todos los escenarios posibles. * El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de compatibilidad. | * Detectar errores en la ejecución de las pruebas. * El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. * Encontrar fallas al ejecutar los distintos casos de pruebas. * La aplicación cumple con los requerimientos mínimos para el funcionamiento. |
| Pruebas de usabilidad. | * El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. * La aplicación debe cumplir con al menos el 90% de los requerimientos funcionales especificados en la fase de análisis. * La aplicación debe cumplir con al menos el 90% de los requerimientos no funcionales especificados en la fase de análisis.   La aplicación cumple con los requerimientos mínimos para el funcionamiento. |
| Pruebas de integridad de datos y base de datos. | * Detectar errores en la ejecución de las pruebas. * El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. * La aplicación debe cumplir con los requerimientos mínimos de seguridad. |
| Prueba de seguridad y control de acceso. | * Detectar errores en la ejecución de las pruebas. * El resultado de cada caso de prueba debe ser igual al resultado de salida esperado. * La aplicación debe cumplir con los requerimientos mínimos de seguridad. |

Para cada una de las pruebas se tendrá en cuenta:

**-Prueba de documentación.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Elemento a Revisar | SI | NO | No Aplica | Observaciones |
| ¿Se realizaron las Pruebas Unitarias con alguna herramienta especializada? |  |  |  |  |
| ¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema? |  |  |  |  |
| ¿Existe constancia de la realización de las pruebas mencionadas? |  |  |  |  |
| ¿El funcionamiento de la prueba unitaria respeta el diseño establecido? |  |  |  |  |
| ¿Existe un manejo de errores adecuado? |  |  |  |  |
| ¿Se cumplió con la estrategia de ejecución de la prueba? |  |  |  |  |

**-Prueba de carga:** El resultado de las pruebas de carga se verá reflejado en el siguiente informe:

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** |  | **Descripción del escenario** |  |
| **Número de pruebas exitosas** |  | **Número de pruebas Fallidas** |  |
| **Tiempo máximo de ejecución de una prueba** |  | **Tiempo Promedio de ejecución de las pruebas** |  |
| **Número de peticiones exitosas** |  | **Número de Peticiones Fallidas** |  |
| **Número de Errores** |  | **Tipo de errores** |  |
| **% de Utilización del Procesador** |  | **Cantidad de Memoria utilizada** |  |
| **Promedio de bytes enviados** |  | **Promedio de bytes recibidos** |  |

**-Prueba de estrés:** El resultado de las pruebas de estrés se verá reflejado en el siguiente informe:

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** |  | **Descripción del escenario** |  |
| **Número de pruebas exitosas** |  | **Número de pruebas Fallidas** |  |
| **Tiempo máximo de ejecución de una prueba** |  | **Tiempo Promedio de ejecución de las pruebas** |  |
| **Número de peticiones exitosas** |  | **Número de Peticiones Fallidas** |  |
| **Número de Errores** |  | **Tipo de errores** |  |
| **% de Utilización del Procesador** |  | **Cantidad de Memoria utilizada** |  |
| **Promedio de bytes enviados** |  | **Promedio de bytes recibidos** |  |

**-Prueba de compatibilidad:** El resultado de las pruebas de compatibilidad se verá reflejado en el siguiente informe:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Elemento a Revisar | SI | NO | No Aplica | Observaciones |
| ¿Se realizaron las Pruebas de compatibilidad con alguna herramienta especializada? |  |  |  |  |
| ¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema? |  |  |  |  |
| ¿Existe constancia de la realización de las pruebas mencionadas? |  |  |  |  |
| ¿El funcionamiento de la prueba compatibilidad respeta el diseño establecido? |  |  |  |  |
| ¿Existe un manejo de errores adecuado? |  |  |  |  |
| ¿Se cumplió con la estrategia de ejecución de la prueba? |  |  |  |  |

**-Prueba de usabilidad:** El resultado de las pruebas de usabilidad se verá reflejado en el siguiente informe:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Elemento a Revisar | SI | NO | No Aplica | Observaciones |
| ¿Se realizaron las Pruebas de usabilidad con alguna herramienta especializada? |  |  |  |  |
| ¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema? |  |  |  |  |
| ¿Existe constancia de la realización de las pruebas mencionadas? |  |  |  |  |
| ¿Qué paginas cubrió la prueba? |  |  |  |  |
| ¿se estableció un criterio para la ejecución de pruebas? |  |  |  |  |
| ¿Se cumplió con la estrategia de ejecución de la prueba? |  |  |  |  |

-**Prueba de integridad de datos y base de datos:** El resultado de las pruebas de integridad de datos y base de datos se verá reflejado en el siguiente informe:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Elemento a Revisar | SI | NO | No Aplica | Observaciones |
| ¿Se realizaron las Pruebas de integridad de datos y base de datos con alguna herramienta especializada? |  |  |  |  |
| ¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema? |  |  |  |  |
| ¿Existe constancia de la realización de las pruebas mencionadas? |  |  |  |  |
| ¿El funcionamiento de la prueba de integridad de datos y base de datos respeta el diseño establecido? |  |  |  |  |
| ¿Existe un manejo de errores adecuado? |  |  |  |  |
| ¿Se cumplió con la estrategia de ejecución de la prueba? |  |  |  |  |

**-Prueba de seguridad y control de acceso:** El resultado de las pruebas de seguridad y control de acceso se verá reflejado en el siguiente informe:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Elemento a Revisar | SI | NO | No Aplica | Observaciones |
| ¿Se realizaron las Pruebas de seguridad con alguna herramienta especializada? |  |  |  |  |
| ¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema? |  |  |  |  |
| ¿Existe constancia de la realización de las pruebas mencionadas? |  |  |  |  |
| ¿El funcionamiento de la prueba de seguridad respeta el diseño establecido? |  |  |  |  |
| ¿Existe un manejo de errores adecuado? |  |  |  |  |
| ¿Se cumplió con la estrategia de ejecución de la prueba? |  |  |  |  |

## Criterio de terminación

A continuación, se señalan los criterios de terminación de las pruebas a ejecutar.

* Se ejecutaron todas las pruebas del sistema.
* Todas las pruebas se ejecutaron de acuerdo con los criterios de evaluación.
* Las pruebas de carga demuestran que se posee un grado satisfactorio de capacidad operativa y funcional.
* Los incidentes encontrados en las pruebas fueron corregidos y probados.

## Criterios de suspensión

Los criterios de suspensión impiden la iniciación y/o continuación de las pruebas ante cualquier situación de improvisto que hace que la ejecución de las pruebas no logre grados satisfactorios de probabilidad de éxito.

* Después de la instalación y configuración del sistema, se evidencia problemas o situaciones anormales en cualquiera de sus componentes.
* Después de la instalación y configuración del sistema, se evidencia que el ambiente de pruebas no es lo suficientemente estable para la ejecución de las pruebas.
* Discrepancia entre la documentación (Set de Pruebas, Casos de Uso) y el sistema.

# Anexos

I. Mapa de empatía



II. Mapa de experiencia de usuario



III. Cuestionario de evaluación del servicio

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Totalmente de acuerdo** | **Ni de acuerdo ni en desacuerdo** | **Totalmente en desacuerdo** |
| *El acceso a la Página Web resultó fácil.* |  |  |  |
| *Pudo entrar a la Página Web desde el primer intento, sin necesidad de insistir en más de una ocasión.* |  |  |  |
| *El diseño de la Página Web es lo suficientemente atractivo como para desear mantenerme en ella.* |  |  |  |
| *Navegar dentro de la Página Web resulta una experiencia fácil.* |  |  |  |
| *Los procesos de búsquedas de información dentro de la Página Web, así como la navegación dentro de ella ocurren de manera rápida y ágil.* |  |  |  |
| *La información publicada en la Página Web es de alta calidad y confiable.* |  |  |  |
| *Encontró todas las informaciones que buscaba.* |  |  |  |
| *El modo en que las informaciones estaban organizadas dentro de la Página Web resultó adecuado y de fácil búsqueda.* |  |  |  |
| *Pensando en la experiencia que acaba de vivir, le gustaría volver a entrar a esta Página Web.* |  |  |  |
| *Está dispuesto a recomendar esta Página Web a un relacionado suyo.* |  |  |  |