**QA testing strategy- Airline**

Enero, 2024

Contenido

[**Antecedentes** 4](file:///C:\Users\Ruth%20Castro\Desktop\Pros\QA%20testing%20strategy%20-%20Airline.docx#_Toc155142384)

[1 Introducción 4](#_Toc155142385)

[1.1 Objetivos 5](#_Toc155142386)

[1.2 Alcance 5](#_Toc155142387)

[1.3 Glosario de términos. 5](#_Toc155142388)

[1.4 Abreviaturas 5](#_Toc155142389)

[1.5 Prerrequisitos para las pruebas 5](#_Toc155142390)

[**Metodología de Pruebas de QA** 7](file:///C:\Users\Ruth%20Castro\Desktop\Pros\QA%20testing%20strategy%20-%20Airline.docx#_Toc155142391)

[Metodología de pruebas 7](#_Toc155142392)

[Tipo de Pruebas 8](#_Toc155142393)

[7.2 Diseño de Pruebas 12](#_Toc155142394)

[7.3 Ciclo o Iteraciones de Prueba 13](#_Toc155142395)

[7.4 Preparación de las Pruebas 13](#_Toc155142396)

[CREAR ENTORNO DE PRUEBAS 14](#_Toc155142398)

[8.1 Formular Condiciones y Casos de Prueba 14](#_Toc155142399)

[8.2 Identificar Datos de Prueba 14](#_Toc155142400)

[EJECUCCIÓN DE PRUEBAS 15](#_Toc155142401)

[DOCUMENTACIÓN 15](#_Toc155142403)

[3.1 Escenario Pruebas de Carga 21](#_Toc155142404)

[3.2 Escenario Pruebas de Estrés 21](#_Toc155142405)

[3.3 Escenario Pruebas de Resistencia 21](#_Toc155142406)

[3.4 Escenario Pruebas de Picos 21](#_Toc155142407)

[3.5 Escenario Pruebas de Escalabilidad 21](#_Toc155142408)

[3.6 Escenario Pruebas de Volumen 22](#_Toc155142409)

[3.7 SLA 22](#_Toc155142410)

[ Instalación de los componentes de software en el ambiente de QA y que los casos de pruebas de verificación de ambiente ejecutados por el área de Infraestructura, sean exitosos. 24](#_Toc155142411)

[Criterio de suspensión de pruebas y condiciones para reanudarla. 25](#_Toc155142412)

[Criterios de aceptación y/o rechazo de pruebas. 26](#_Toc155142413)

[**1.** **Personal** 30](#_Toc155142414)

[**2.** **Matriz de responsabilidades** 30](#_Toc155142415)

[**3.** **Entornos – Software** 34](#_Toc155142416)

[**4.** **Herramientas de pruebas requeridas** 34](#_Toc155142417)

[2.1 **Funcionalidad no cubierta -** Exclusiones 34](#_Toc155142418)

[8.2 Diferencia entre Error, Defecto y Falla 36](#_Toc155142419)

[8.3 Clasificación de Defectos 36](#_Toc155142420)

[8.4 Estado de Defectos 36](#_Toc155142421)

[Análisis de riesgos 39](#_Toc155142422)

[**4** **Control de historial de cambios** 41](#_Toc155142423)

[**5** **Firmas de revisión y aprobación** 41](#_Toc155142424)

**01.**

# **Antecedentes**

**Estrategia de Pruebas de QA**

Sistema - AEROLÍNEA

# Introducción

[Esta sección debe contener una breve introducción del sistema a probar.]

# Objetivos

[Esta sección debe contener los objetivos del sistema a probar.]

# Alcance

# 1.3 Glosario de términos.

**Prueba:** Es una actividad realizada para evaluar la calidad del producto y mejorarla, identificando defectos y problemas.

**Verificación:** Proceso de evaluación de un sistema o componente para determinar si un producto de una determinada fase de desarrollo satisface las condiciones impuestas al inicio de la fase.

**Validación:** Proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar cuándo se satisfacen los requerimientos especificados.

**Defecto:** Discrepancia entre los resultados obtenidos al ejecutar el software y los resultados que se esperaban.

**Prueba funcional:** Es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas funcionales se hacen mediante el diseño de casos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático.

**Prueba unitaria:** Es una forma de comprobar el correcto funcionamiento un módulo de código, con la finalidad de asegurar que cada uno de los módulos funcione correctamente por separado.

**Prueba de integración:** La prueba de integración es una técnica para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción.

**Pruebas de regresión:** Las pruebas de regresión tienen por objeto descubrir errores con respecto al comportamiento esperado del software, causados por la realización de un cambio en un componente.

**Iteración:** Es el acto de repetir un proceso con la intención de alcanzar una meta deseada, objetivo o resultado.

# 1.4 Abreviaturas

**Q.C**= Control de Calidad

**Q.A**= Aseguramiento de la Calidad

**TI** = Tecnologías de la Información

# 1.5 Prerrequisitos para las pruebas

Para dar inicio a la ejecución de las pruebas se solicita el cumplimiento de las siguientes condiciones:

* Un desarrollo estable de la aplicación correspondiente al sprint que se va a probar, instalada en el entorno de QA lo más parecido al ambiente de producción.
* Preparación del entorno de prueba la configuración del entorno de prueba se realiza en base a la lista de requisitos de hardware y software que se tengan.
* Verificación  de Herramientas de control de calidad instaladas y configuradas
* Aceptación por parte de los interesados de este documento de estrategia de pruebas de QA
* Definición de criterios de done

**02.**

# **Metodología de Pruebas de QA**

# Metodología de pruebas

Considerando que la Institución desarrolla sistemas informáticos los mismos que deben implementarse de acuerdo a cronogramas establecidos y optimizando recursos, en base a la metodología de desarrollo que se aplica actualmente se ha establecido un método ágil aplicable para las pruebas de los aplicativos de software. (Scrum).

Breve Descripción del Proceso de Pruebas:

**Fase: Inicio**

**Fase: Planificación y estimación**

**Fase: Implementación**

**Fase: Revisión y retrospectiva**

**Fase: Lanzamiento**

## Tipo de Pruebas

En la siguiente tabla se definen los tipos de pruebas que serán aplicados en los respectivos ambientes:

| **Ambiente** | **Tipos de pruebas** | **Descripción / Objetivo** | **Responsable** |
| --- | --- | --- | --- |
| Pruebas | Certificación de ambiente de prueba | Certificar la correcta instalación y configuración del ambiente de pruebas, garantizando que el ambiente será idéntico al ambiente de Producción en cuanto a software base y configuración. | Miembros del equipo de Infraestructura |
| Desarrollo | Pruebas Unitarias | Realizadas para verificar el correcto funcionamiento de los requerimientos establecidos por el usuario final, este tipo de pruebas son realizadas exclusivamente en el ambiente de desarrollo antes de envío a pruebas de certificación. | Miembros del equipo de Desarrollo |
| Pruebas | Pruebas de Integración | Se verifica la relación de integración de funcionalidad que se da entre los diferentes módulos, procesos y sistemas. | Miembros del equipo de Desarrollo |
| Pruebas | Pruebas Funcionales | Su objetivo principal es probar y verificar la funcionalidad correcta del sistema, a partir de los requerimientos funcionales. | Miembros del equipo de QA |
| Pruebas | Pruebas de Regresión | Su objetivo es validar que ante un cambio realizado en el sistema, las partes del mismo que no hayan sido afectadas, continúen funcionando correctamente, luego de implementada la modificación. | Miembros del equipo de QA |
| Pruebas | Pruebas de Rendimiento | La finalidad de realizar pruebas de Rendimiento (tiempo de respuesta adecuada), volumen (Funcionamiento de grandes volúmenes de datos) y sobrecarga (Funcionamiento en el umbral límite de los recursos), es predecir anticipadamente problemas de rendimiento y degradación de recursos del sistema antes de su paso a Producción para facilitar su corrección. | Miembros del equipo de QA |
| Pruebas | Vulnerabilidades | Consiste en identificar vulnerabilidades que puedan ser explotadas y afecten el sistema o aplicativo, mediante análisis automatizados y manuales. | Miembros del equipo de QA |
| Pruebas | Pruebas de Aceptación de usuario (UAT) | Su objetivo principal es probar y verificar la funcionalidad correcta del sistema, a partir de los requerimientos funcionales, sin tomar en cuenta la forma en cómo éste resuelve internamente las especificaciones.  El éxito de este tipo de pruebas se basa en generar y mantener una clara trazabilidad entre los requisitos y las historias de usuario asociadas. Como parte de estas pruebas, se debe verificar la integridad de la información. | Miembros del equipo de QA y Usuario Funcional |

Tabla 1: Tipo de Pruebas

**03.**

**Pruebas Funcionales**

## Diseño de Pruebas

A continuación se describe brevemente y de manera general el diseño de las pruebas.

|  |  |
| --- | --- |
| **Tipo de Prueba** | **Diseño de Prueba** |
| Pruebas Funcionales | Estas pruebas se ejecutarán en el ambiente de QA, con la URL y usuarios entregados por el área de desarrollo de software.  Durante el diseño de la prueba se buscará obtener un conjunto amplio de casos de prueba (evitando la redundancia), para chequear que toda la información incluida en los requisitos funcionales efectivamente esté implantada en la aplicación bajo prueba.  Los casos de prueba deberán incluir:   * Proceso de negocio * Flujo Normal * Flujos Alternativos * Flujos que generen condiciones de error. * Excepciones * Validaciones de estándares de presentación y diseño. * Validaciones de políticas de Seguridad. * Validaciones que se puedan hacer en el sistema que correspondan a documentos emitidos por entidades regulatorias. * Validaciones de Integridad * Campos obligatorios * Campos opcionales * Tipos de datos, longitudes y formatos * Campos calculados |
| Pruebas de Integración | Los casos de prueba a ejecutar en las pruebas funcionales de integración, se diseñan de la siguiente manera:   * Se enlazan casos de prueba de los distintos módulos que tienen dependencia directa (la función de un módulo llama a la de otro). * Se determinan los riesgos de cada caso de prueba en función de: importancia de la funcionalidad y probabilidad de uso. |

Tabla 2: Diseño de Pruebas

## Ciclo o Iteraciones de Prueba

Para cada tipo de prueba se realizará el re-test respectivo si el caso lo amerita, antes de iniciar con las pruebas de aceptación de usuario.

Los defectos resueltos por el área de desarrollo serán verificados por los miembros del equipo de QA y marcados como Resuelto.

Tabla 3: Ciclos de prueba

|  |  |
| --- | --- |
| **SPRINT** | |
| **Iteración 1** | |
| Pruebas funcionales | Primera ejecución de pruebas. |
| Re-Test (en caso de que se haya reportado incidencias, se debe volver a probar esos casos). |
| La prueba termina hasta que se cumplan los criterios de aceptación de la user story. |
| Pruebas de rendimiento | Primera ejecución de pruebas. |
| Re-Test (en caso de que se haya reportado incidencias, se debe volver a probar esos casos). |
|  | La prueba termina hasta que se cumplan los criterios de aceptación la user story. |
|  | Primera ejecución de pruebas. |
| Pruebas de vulnerabilidades | Re-Test (en caso de que se haya reportado incidencias, se debe volver a probar esos casos). |
|  | La prueba termina hasta que se cumplan los criterios de aceptación la user story. |

# Preparación de las Pruebas

Esta etapa inicia con la versión y análisis de requerimientos funcionales levantados para el desarrollo del aplicativo a ser probado, en base a esto se estable casos y condiciones de prueba y se define las características hardware y software requeridas para el ambiente de pruebas.

# 

# CREAR ENTORNO DE PRUEBAS

Para la ejecución de las pruebas se deberá contar con un ambiente de pruebas con las mismas características en cuanto a configuración y software base con las que contará el ambiente de Producción.

Para la creación del ambiente de pruebas se deberá contar con:

Versión liberada para pruebas del aplicativo de software

Manual de instalación del aplicativo

## Formular Condiciones y Casos de Prueba

Un caso de prueba es un conjunto de condiciones o variables bajo las cuales es posible determinar si un sistema de software o una funcionalidad específica son satisfactorios.

Las condiciones y casos de prueba se deberán detallar en el documento Plan de casos de prueba.

Al diseñar los casos de prueba, se debe considerar la elaboración de casos tanto alternos como de excepción. Los casos de excepción permiten validar el comportamiento del sistema ante situaciones anómalas.

Cada caso de prueba tiene la siguiente estructura:

Pasos y valor esperado: En todos los casos de pruebas se detalla claramente los pasos a seguir y el valor esperado por cada paso.

Después de ejecutar los casos de prueba estos pueden tener los siguientes estados:

**Cumple:** Si la aplicación muestra el valor esperado por cada paso definido en el caso de prueba.

**No cumple:** Si la aplicación no muestra el valor esperado para algún paso definido en el caso de prueba.

**Bloqueado:** Si no es posible ejecutar el caso de prueba por falta de implementación o falla de algún caso de prueba.

**No aplica:** Si luego de realizar una revisión, se determina que el caso de prueba elaborado no aplica para la funcionalidad del sistema.

## Identificar Datos de Prueba

Es necesario definir cuáles son los datos de pruebas requeridos para la ejecución de los casos de pruebas diseñados.

Se especificarán conjuntos de datos de prueba tomando las muestras necesarias para la ejecución de las pruebas.

# EJECUCCIÓN DE PRUEBAS

La ejecución de pruebas arranca con la creación de los datos de prueba necesarios para ejecutar los casos de pruebas diseñados en el ambiente correspondiente.

La ejecución de los casos de prueba, implicará el registro de evidencia de la prueba ejecutada, así como la documentación de los defectos encontrados para la gestión de su corrección.

Una vez el defecto ha sido corregido por el equipo de desarrollo, será necesario realizar una nueva prueba que permita confirmar que le defecto fue solucionado de manera exitosa.



# DOCUMENTACIÓN

Como parte del Proceso de pruebas se generará la siguiente documentación para cada fase del proyecto:

**Fase: Inicio**

**Fase: Planificación y estimación**

**Fase: Implementación**

**Fase: Revisión y retrospectiva**

**Fase: Lanzamiento**

**04.**

**Pruebas de Vulnerabilidad**

Se realizarán pruebas de vulnerabilidades (caja blanca) al código del sistema de manera automática, para identificar vulnerabilidades de seguridad en el código y verificar la calidad del mismo.

El análisis del código fuente se realizará con las herramientas de scaneo de vulnerabilidades con que cuente el área.

Una vez ejecutadas las pruebas de caja blanca, se identificará:

* Vulnerabilidades conocidas de seguridad.
* Errores bloqueantes, críticos y mayores.
* Detección de código duplicado.
* Falta de comentarios.
* Cálculo de la deuda técnica.
* No adecuación a estándares y convenciones de código.
* Código spaghetti.

Se evaluarán todos los componentes del sistema, los resultados obtenidos, serán entregados a los desarrolladores para la optimización y mejora del código.

El análisis del código fuente, se ejecutará en un ambiente de Seguridad Informática destinado para este tipo de pruebas.

El área de desarrollo de software, proporcionará el acceso al repositorio del código fuente del sistema.

Las pruebas se harán a todos los módulos del sistema.

**05.**

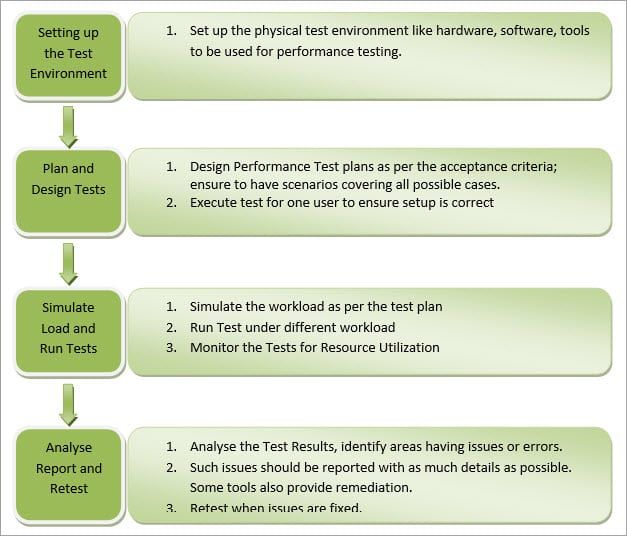
**Pruebas de Rendimiento**

Las pruebas de rendimiento de software se centran en determinar la velocidad con la que el sistema realiza tareas específicas bajo condiciones particulares, tomando en cuenta el tiempo desde que se hacen las solicitudes al servidor de aplicación hasta que se recibe una respuesta por parte de este. El tiempo de respuesta es un factor importante en la fase de inspección y análisis ya que sirve para detectar en cuales operaciones/funcionalidades pudiera existir cuellos de botella y que además no cumplen con los niveles de servicio acordados.

Cave recalcar, que las pruebas de rendimiento se podrán realizar siempre y cuando se cuente con el hardware y software necesario, caso contrario se informará oportunamente las limitaciones y los tipos de pruebas a ejecutar.

Es recomendable hacer pruebas de rendimiento en servidores con las características más próximas a los de producción, simulando escenarios reales o de ser el caso se las debe ejecutar directamente en el ambiente de producción tomando las medidas preventivas necesarias.

Reunión con el arquitecto de software para definición de los escenarios a probar según se lo haya solicitado en la historia de usuario correspondiente a requerimientos no funcionales, en donde se debe especificar el número de visitantes que espera el sitio web o aplicación durante un período de tiempo específico.



Las métricas que deben ser consideradas en este tipo de pruebas son:

* Tiempo de carga
* Tiempo de respuesta
* Tiempo de espera
* Tiempo máximo de respuesta
* Tasa de errores
* [Usuarios simultáneos](https://www.loadview-testing.com/es/blog/pruebas-de-carga-http-simultaneo-frente-a-navegadores-simultaneos-frente-a-usuarios-simultaneos/)
* Solicitudes por segundo
* Transacciones superadas/fallidas
* Rendimiento del servidor web
* Utilización de la CPU
* Utilización de la memoria
* E/S de disco

Durante la ejecución de estas pruebas se realizará la monitorización de los servidores donde se está ejecutando la prueba.

# Escenario Pruebas de Carga

|  |  |  |
| --- | --- | --- |
|  | **Primera prueba** | |
| **Carga:** | 1 | |
| **Tiempo de ejecución:** | 1 | 0:05:00 |
| **Subida de usuarios:** |  | Simultáneo |
| **Bajada de usuarios:** |  | Simultáneo |

# Escenario Pruebas de Estrés

|  |  |  |
| --- | --- | --- |
|  | **Primera prueba** | |
| **Carga:** | 1 | |
| **Tiempo de ejecución:** | 1 | 0:05:00 |
| **Subida de usuarios:** |  | Simultáneo |
| **Bajada de usuarios:** |  | Simultáneo |

# Escenario Pruebas de Resistencia

|  |  |  |
| --- | --- | --- |
|  | **Primera prueba** | |
| **Carga:** | 1 | |
| **Tiempo de ejecución:** | 1 | 0:05:00 |
| **Subida de usuarios:** |  | Simultáneo |
| **Bajada de usuarios:** |  | Simultáneo |

# Escenario Pruebas de Picos

|  |  |  |
| --- | --- | --- |
|  | **Primera prueba** | |
| **Carga:** | 1 | |
| **Tiempo de ejecución:** | 1 | 0:05:00 |
| **Subida de usuarios:** |  | Simultáneo |
| **Bajada de usuarios:** |  | Simultáneo |

# Escenario Pruebas de Escalabilidad

|  |  |  |
| --- | --- | --- |
|  | **Primera prueba** | |
| **Carga:** | 1 | |
| **Tiempo de ejecución:** | 1 | 0:05:00 |
| **Subida de usuarios:** |  | Simultáneo |
| **Bajada de usuarios:** |  | Simultáneo |

# Escenario Pruebas de Volumen

|  |  |  |
| --- | --- | --- |
|  | **Primera prueba** | |
| **Carga:** | 1 | |
| **Tiempo de ejecución:** | 1 | 0:05:00 |
| **Subida de usuarios:** |  | Simultáneo |
| **Bajada de usuarios:** |  | Simultáneo |

# SLA

**Successses Number**

**Failures Number**

**Response time sec**

**Uptime %**

**Target Uptime**

**06.**

**Criterios de pruebas de QA**

Criterios de inicio de pruebas

Los criterios de inicio de pruebas solicitadas por el área de QA son los siguientes:

|  |  |
| --- | --- |
| **Tipo de prueba** | **Criterio de Inicio de pruebas** |
| Pruebas funcionales | Se procederá con el inicio de pruebas funcionales si se cumple las siguientes condiciones:   * Se deberá contar con un ambiente de pruebas con características lo más cercanas posibles al ambiente de Producción donde estará instalada la aplicación a probar.  Instalación de los componentes de software en el ambiente de QA y que los casos de pruebas de verificación de ambiente ejecutados por el área de Infraestructura, sean exitosos. |
| Pruebas de vulnerabilidad | Se procederá con el inicio de pruebas de vulnerabilidades por parte del equipo de seguridad de la información si se cumple las siguientes condiciones:   * El área de desarrollo de software proporcionará el acceso al repositorio del código fuente del sistema en GitLab al área de aseguramiento de la información. * La infraestructura donde se encuentre instalado la herramienta de scaneo del código fuente Fortify, debe ser la adecuada para que el proceso de scaneo de vulnerabilidades sea óptimo. |
| Pruebas de rendimiento | Se procederá con el inicio de pruebas de rendimiento si se cumple las siguientes condiciones:   * El sistema al cual se realizará las pruebas de rendimiento, en el caso que se amerite, deberá estar parametrizado para que el robot que realiza la grabación del script de pruebas no tenga impedimento en la grabación del mismo. Por ejemplo, no deberá contener Captcha o factor de doble autenticación. * Las pruebas de rendimiento no podrán iniciarse si se están ejecutando pruebas funcionales en el mismo sistema, debido a la interferencia que tendrá, no será posible realizar el análisis de manera completa y confiable. |
| Pruebas de Aceptación | Se procederá con el inicio de pruebas de aceptación de usuario si se cumple las siguientes condiciones:   * El área requirente deberá recoger los criterios de aceptación de las historias de usuario que están listas para entrar en funcionamiento. * Las pruebas deberán comenzar cuando el porcentaje de incidencias reportadas por QA sean muy pocas o ninguna o según los criterios de aceptación que se hayan acordado. |

# Criterio de suspensión de pruebas y condiciones para reanudarla.

Existen factores que impiden la ejecución normal de las pruebas de manera estable y continua, o no permiten realizarlas de manera exitosa. Por tal motivo, se establecen criterios que pueden determinar la suspensión de un ciclo de pruebas.

Los criterios de suspensión de pruebas, establecidos por el área de QA son los siguientes:

|  |  |
| --- | --- |
| **Tipo de prueba** | **Criterio de Suspensión de pruebas** |
| Pruebas funcionales | Se suspenderá la ejecución de pruebas funcionales si se detectan anomalías tales como:   * Si no se tiene la certificación por parte del área de   Infraestructura de que el sistema ha sido instalado en el ambiente de QA en su totalidad.   * Si en alguno de los casos de prueba se detecta un defecto bloqueante, que impida la continuidad del flujo de prueba. * Presencia de gran cantidad de defectos de severidad alta y crítica, lo que demuestra la necesidad de una revisión completa por parte del equipo de desarrollo. * No se cuenta con datos para pruebas en el ambiente de QA. |
| Pruebas de integración | Se suspenderá la ejecución de pruebas de integración si se detectan anomalías tales como:   * Si alguno de los módulos del sistema no está disponible, lo que impide concluir con un flujo completo de pruebas entre módulos. * En caso de producirse nuevas versiones del aplicativo se detendrán las pruebas y deberán iniciarse otra vez. |
| Pruebas de rendimiento | Se suspenderá la ejecución de pruebas de rendimiento si se detectan anomalías tales como:   * Que la base datos se encuentre llena. * Que se pierda la conexión entre el sistema a probar y el servidor donde se encuentra la aplicación. * Los logs del servidor se encuentren llenos. |
| Pruebas de vulnerabilidades | Se suspenderá la ejecución de pruebas de vulnerabilidades si se detectan anomalías tales como:   * Los servidores donde está instalado el Sistema Fortify presentan anomalías y no permite la correcta ejecución de las pruebas. * No se ha dado acceso al código fuente del sistema a probar. |
| Pruebas de Aceptación | Se suspenderá la ejecución de las pruebas de aceptación si el Equipo de Q.A determina que los errores (altos y críticos) detectados en las pruebas funcionales aún no han sido corregidos en su totalidad. |

De presentar alguna de las anomalías descritas anteriormente las pruebas serán suspendidas temporalmente, y serán reanudadas una vez que estas condiciones ya no existan (sean solventadas), entonces se procederá con la reanudación.

# Criterios de aceptación y/o rechazo de pruebas.

El elemento que se utilizará para la ejecución de las pruebas de QA son las User Stories, el área de QA deberá recoger los criterios de aceptación de las historias de usuario que estén listas para ser probadas.

La historia debe estar “lista” cuando el usuario puede completar la tarea descrita, es decir cuando cumple completamente su funcionalidad.

Se ha definido una clasificación de defectos encontrados durante la ejecución de pruebas, en la cual se especifica el porcentaje para la aceptación de cada una de ellas, a continuación se detallan según su severidad y estatus dependiendo del tipo de prueba:

**Niveles de severidad para pruebas funcionales:**

|  |  |  |
| --- | --- | --- |
| **Severidad** | **Descripción** | **% para aceptación** |
| **Crítico** | Es un problema que bloquea el progreso.  En esta categoría estarán todas las historias de usuarios.  El sistema deberá cumplir todas las historias de usuario solicitadas por el área requirente. | 100 % |
| **Alto** | Problema grave que podría bloquear el progreso. | 100 % |
| **Medio** | Tiene el potencial de afectar el progreso. | % |
| **Bajo** | Problema menor bajo o fácil de solucionar. | % |
| **Más bajo** | Trivial con poco o ningún impacto en el progreso. | % |

**Niveles de severidad para pruebas de vulnerabilidad**

Para calcular la severidad global de riesgo, Fortify, tiene una escala que va del 0 al 9 dividido en 4 categorías de probabilidad de ocurrencia e impacto estimado, estas son: BAJA, MEDIA, ALTA o CRÍTICA, su detalle se muestra a continuación:

Owasp top 10 2021

|  |  |  |
| --- | --- | --- |
| **Severidad** | **Descripción** | **% para aceptación** |
| **Crítico** | Los problemas críticos son fáciles de descubrir y explotar para que el atacante provoque un gran daño a los activos. | % |
| **Alto** | Los problemas de alta prioridad a menudo son difíciles de descubrir y explotar, pero pueden provocar daños importantes a los activos. | % |
| **Medio** | Los problemas de prioridad media son fáciles de descubrir y explotar, pero a menudo resultan en poco daño a los activos. | % |
| **Bajo** | Los problemas de baja prioridad son difíciles de descubrir y explotar y, por lo general, provocan poco daño a los activos. | % |

|  |  |
| --- | --- |
| **Estado** | **Descripción** |
| A1 Injection |  |
| A2 Broken Authentication |  |
| A3 Sensitive Data Exposure |  |
| A4 XML External Entities (XXE) |  |
| A5 Broken Access Control |  |
| A6 Security Misconfiguration |  |
| A7 Cross-Site Scripting (XSS) |  |
| A8 Insecure Deserialization |  |
| A9 Using Components with Known Vulnerabilities |  |
| A10 Insufficient Logging and Monitoring |  |

**Niveles de severidad para pruebas de rendimiento**

|  |  |  |
| --- | --- | --- |
| **Severidad** | **Descripción** | **% para aceptación** |
| **Crítico** | Tasa de transacciones con error | % |
| **Alto** |  | % |
| **Medio** |  | % |
| **Bajo** |  | % |

**SLA**

|  |  |  |
| --- | --- | --- |
| **Transacciones** | **Descripción** | **Criterio para aceptación** |
| Carga de página | Tiempo que se demora en cargar cada página del sistema. | 2 segundos |
| Login | Tiempo que se demora en ejecutar la transacción de Login. | 2 segundos |
| Logout | Tiempo que se demora en ejecutar la transacción de Logout. | 2 segundos |
| Uso de CPU | El uso del CPU no debe superar el 80% | No debe sobrepasar el 80% |
| Uso de Memoria | El uso de la memoria no debe superar el 80% | No debe sobrepasar el 80% |

**Estado de reporte de las incidencias.**

|  |  |
| --- | --- |
| **Estado** | **Descripción** |
| Abierto | El problema está abierto y listo para que el cesionario comience a trabajar en él. |
| En progreso | La cesionaria está trabajando activamente en esta cuestión. |
| Re - Abierto | Este problema se resolvió una vez, pero la resolución se consideró incorrecta. A partir de aquí, los problemas se marcan como asignados o resueltos. |
| Resuelto | Se ha tomado una resolución, y está a la espera de la verificación del periodista. A partir de aquí, los problemas se vuelven a abrir o se cierran. |
| Cerrado | El problema se considera terminado, la resolución es correcta. Los problemas que están cerrados se pueden reabrir. |
| Construcción | El código fuente se ha confirmado y JIRA está esperando a que se construya el código antes de pasar al siguiente estado. |
| Construcción rota | El código fuente confirmado para este problema posiblemente haya roto la compilación. |

Entre las partes involucradas en el proceso de pruebas de cada aplicativo de software, se definirá bajo qué condiciones se puede considerar que la prueba fue realizada. Por cada tipo de prueba se definirá los criterios de cierre.

Los criterios de cierre de pruebas solicitados por el área de QA son los siguientes:

Porcentaje de casos de prueba ejecutados satisfactoriamente

Porcentaje de cubrimiento de funcionalidades en los casos de pruebas ejecutados

Porcentaje de cubrimiento de funcionalidades críticas para el sistema en los casos de pruebas ejecutados.

Número de defectos críticos y mayores pendientes

Número de defectos menores pendientes

Se cuenta con el aval de la Unidad Interna de Seguridad Informática luego de las pruebas de Vulnerabilidades

Se cuenta con el acta de aceptación de usuario firmada en donde consta que el Sistema cumple con lo solicitado.

# **Personal**

El personal a cargo de la ejecución de las pruebas de QA es:

|  |  |
| --- | --- |
| **Nombres** | |
| Verónica Mejía |  |

# **Matriz de responsabilidades**

|  |  |
| --- | --- |
| **TAREA** | **RESPONSABLE** |
| Documento de Estrategia de pruebas |  |
| Plan de pruebas Sistema de Reclutamiento. |  |
| Elaboración de casos de pruebas. |  |
| Creación del proyecto y carga del plan de pruebas en la herramienta TestLink. |  |
| Creación del proyecto en Mantis. |  |
| Ejecución de pruebas funcionales. |  |
| Reporte de avance de las pruebas. |  |
| Reporte de incidencias al equipo de desarrollo. |  |
| Cierre de defectos. |  |
| Elaboración de informe de pruebas funcionales. |  |
| Configuración de equipos para pruebas de rendimiento. |  |
| Ejecución de pruebas de rendimiento. |  |
| Elaboración de informe de pruebas de rendimiento. |  |
| Pruebas aceptación de usuario |  |

# 

**04.**

**Pruebas de aceptación de usuario**

**05.**

**Infraestructura y Software**

**Entornos – Hardware - Pruebas Funcionales**

Equipos con los que cuenta el personal de QA para la ejecución de pruebas funcionales:

Equipo miembro 1 de QA:

|  |  |
| --- | --- |
| **PC DE ESCRITORIO** | |
| Procesador | Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19 GHz |
| RAM | 16,0 GB (15,8 GB usable) |
| Tipo de sistema | 64-bit operating system, x64-based processor |
| SO | Windows 11 Pro |

Servidores en los que se encuentra instalado el “Sistema de Reclutamiento” en el ambiente de QA.

|  |
| --- |
| **SERVIDOR DE BASE DE DATOS** |
| IP: 172.16.2.250  SO: Linux Centos 7  RAM: 16 Gb  CPU: 10 CPU  Espacio en Disco: 76 GB  Motor de Base de Datos: Postgresql 13  Puerto: 5432  Base de Datos: reclutamiento |

|  |
| --- |
| **SERVIDOR DE APLICACIONES** |
| IP: 172.16.2.133  SO: Windows Server  2019 Standart  RAM: 8 Gb  CPU: 4 CPU  Espacio en Disco: 100 GB  Framework de Desarrollo: .Net Framework 4.5  Servidor Web: IIS  Puerto: 80  URL: <http://172.16.2.133/Default.aspx> |

**Entornos – Hardware - Pruebas de Rendimiento**

**SERVIDOR DE CARGA**

Equipos para la ejecución de pruebas de rendimiento:

|  |  |
| --- | --- |
| **SERVIDOR DE CARGA** | |
| Procesador | Intel® Xeon(r) cpu E5-470 2.3GHz |
| RAM | 12 |
| IP | 172.16.2.85 |
| SO | Microsoft Windows Server 2012 Standard |

**GENERADORES DE CARGA**

Al momento nos asignaron 5 PCs, de las cuales dos presentan problemas técnicos y las restantes tienen 4 G de RAM.

Se procederá a hacer la revisión de estos equipos para determinar si son aptas para la ejecución de pruebas de rendimiento.

# **Entornos – Software**

Los navegadores web en los cuales se harán las pruebas son:

|  |  |
| --- | --- |
| **NAVEGADOR** |  |
| Firefox | Version 99.0.1 (64-bit) |
| Google Chrome | Version 102.0.5005.115 (Official Build) (64-bit) |

# **Herramientas de pruebas requeridas**

**Jira:**

**LoadRunner:** Es su solución de pruebas de rendimiento que se utiliza para simular transacciones de usuario y medir el comportamiento y el rendimiento del sistema bajo carga, lo que ayuda a DevOps e ingenieros a descubrir errores y cuellos de botella.

**Fortify:**

# **Funcionalidad no cubierta -** Exclusiones

**06.**

**Métricas**

## Diferencia entre Error, Defecto y Falla

Error: acción humana que produce un resultado incorrecto, por ejemplo un error de programación.

Defecto: Imperfección en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas. Por ejemplo, si se localiza un defecto durante una ejecución puede causar un fallo en el componente o sistema, por ejemplo una sentencia o una definición de datos incorrecta.

Fallo: Manifestación física o funcional de un defecto, por ejemplo, desviación de un componente o sistema respecto de la presentación, servicio o resultado esperado.

## Clasificación de Defectos

Se ha definido una clasificación de defectos encontrados durante la ejecución de pruebas, a continuación se detalla cada uno:

| **Clasificación Incidencias** | **Descripción** |
| --- | --- |
| Bloqueantes | Son defectos del producto del mal funcionamiento del sistema que impiden continuar con la ejecución de pruebas. |
| Critico | Son defectos que afectan los requisitos del negocio o degradan el rendimiento, la funcionalidad o la capacidad de uso. |
| Mayor | Son incidentes que pueden afectar el funcionamiento del sistema pero no impiden continuar con la ejecución de las pruebas. |
| Menor | Son incidentes que no afectan la funcionalidad del sistema, pero si la forma, es decir la interfaz gráfica o la usabilidad. |
| Sugerencias | Son incidentes que se pueden realizar por parte del Analista Q.A para el mejor funcionamiento del Sistema. |

Tabla 4: Clasificación de Defectos

## Estado de Defectos

Se manejarán los siguientes estados para los defectos encontrados durante la ejecución de los distintos tipos de prueba, su corrección, reprueba y posterior cierre de defecto reportado.

|  |  |
| --- | --- |
| **Estado Defectos** | **Descripción** |
| Abierto | Estado del defecto una vez que ha sido identificado por el equipo de pruebas. Los defectos en este estado son asignados al equipo de desarrollo para su corrección. |
| En corrección | En este estado se encuentran los defectos sobre los cuales el equipo de desarrollo se encuentra trabajando para su solución. |
| En reprueba | Una vez que el equipo de desarrollo ha concluido la corrección, el defecto pasa a este estado, es decir estará disponible para una nueva prueba para verificar que ha sido solucionado. |
| Cerrado | Una vez verificado el correcto funcionamiento de una funcionalidad que anteriormente se reportó con errores, el defecto pasará a estado “Cerrado”. |
| Reabierto | Si durante la nueva prueba para verificar la corrección de un defecto, persiste los errores anteriormente cerrados, este defecto pasa a estado “reabierto”, con lo cual será nuevamente asignado al equipo de desarrollo. |

**08.**

**Riesgos**

# Análisis de riesgos

Procedimientos:

* De acuerdo con el procedimiento que maneja DIRAD, no se cuenta con un área específica de QA, por lo que, se está proponiendo en la nueva estructura de DITIC, se incluya esta área dentro de la Gestión de Seguridad Informática, Interoperabilidad y Riesgos.

Recursos humanos:

* El área de QA cuenta con dos integrantes, encargados de la ejecución de pruebas de los sistemas a ser utilizados en el Censo 2022. Lo que provoca una alta carga de trabajo al equipo.  Para esto, se ha dividido actividades buscando que la carga de trabajo sea equitativa y además se priorizará los casos de pruebas de mayor impacto del sistema a probar.

Tiempo:

* Para la ejecución de pruebas por parte de QA, no se cuenta con el plazo acordado del 30% del tiempo asignado para el desarrollo. Las pruebas no se ejecutaron en conjunto con el desarrollado del software, con entregables parciales, sino fueron ejecutadas en la fase previa a la puesta en producción del sistema.

**09.**

**Versiones**

# **Control de historial de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha actualización** | **Versión** | **Autor** | **Descripción del cambio** |
|  |  |  |  |

# **Firmas de revisión y aprobación**

|  |  |  |  |
| --- | --- | --- | --- |
| **Acciones** | **Nombre** | **Cargo** | **Firma** |
| **Elaboración de la estrategia de pruebas de QA** |  |  |  |