



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

**Ingeniería en Tecnologías de la
Información**

Plan de pruebas- Teoría

Asignatura: Control de calidad

Docente: Dr. Juan Manuel González Calleros.

Integrantes:

- **Luis Alberto amador flores.**
- **Luis Adrián Pérez san Martín.**



ÍNDICE

| | | |
|----------|--|--|
| 1. | INTRODUCCIÓN | |
| 1.1. | PROPÓSITO | |
| 1.2. | ALCANCE | |
| 1.3. | DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS | |
| 1.4. | REFERENCIAS | |
| 2. | ESTRATEGIA DE PRUEBAS | |
| 2.1. | TÉCNICAS DE ESPECIFICACIÓN DE LAS PRUEBAS .. | |
| 2.1.1.1. | PLANIFICACIÓN | |
| 2.1.1.2. | DISEÑO DE LAS PRUEBAS | |
| 2.2. | TIPOS DE PRUEBAS | |
| 2.3. | PRUEBAS UNITARIAS | |
| 2.3.1.1. | PRUEBAS DEL SISTEMA | |
| 2.3.1.2. | PRUEBAS DE INTEGRACIÓN | |
| 2.3.1.3. | PRUEBAS DE INTEROPERABILIDAD | |
| 2.3.1.4. | PRUEBAS DE REGRESIÓN | |
| 2.3.1.5. | PRUEBAS FUNCIONALES | |
| 2.3.1.6. | PRUEBAS DE USABILIDAD..... | |
| 2.3.1.7. | PRUEBAS DE SEGURIDAD | |
| 2.3.1.8. | PRUEBAS DE CONFIGURACIÓN | |
| 2.3.10. | PRUEBAS DE RECUPERACIÓN A FALLAS..... | |
| 2.3.11. | PRUEBAS DE ACEPTACIÓN | |
| 2.4. | ENTREGABLES DE PRUEBAS | |
| 3. | EVALUACIÓN DE PRUEBAS EJECUTADAS | |
| 3.1. | CRITERIOS DE INICIO DE EJECUCIÓN | |
| 3.2. | CRITERIOS DE EVALUACION | |
| 4. | ANEXOS | |



INTRODUCCIÓN

1.1. 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 pre-requisitos que deben ser considerados en el esfuerzo de las pruebas, en la búsqueda de garantizar el cumplimiento de los requerimientos planteados en el marco del desarrollo del proyecto.

El documento específicamente:

- Identifica información del proyecto y componentes de software que pueden ser probados.
- Los tipos de pruebas a ser implementados.
- Recomendaciones y descripciones de estrategias de prueba a ser empleadas.

1.2 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 VDALTON. La metodología de pruebas y este documento de plan de pruebas permitirán al equipo evaluar aspectos como: la funcionalidad, la seguridad, la usabilidad, y realizar un seguimiento de las pruebas realizadas, así como el control de los resultados.

1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

El plan de prueba: describe todos los métodos que se utilizarán para verificar que el software satisface la especificación del producto y las necesidades del cliente. Incluye los objetivos de calidad, necesidades de recursos, cronograma, asignaciones, métodos, etc.

Casos de prueba: lista los ítems específicos que serán probados y describe los pasos detallados que serán seguidos para verificar el software.

Reporte de pruebas: describen los problemas encontrados al ejecutar los casos de prueba.

Herramientas de pruebas y automatización: documentación de las herramientas empleadas en el proceso de pruebas.

VDALTON: visión dalton.



| TIPO DE PRUEBA | DEFINICIONES | FASE DEL PROCESO DE DESARROLLO |
|--|---|----------------------------------|
| <ul style="list-style-type: none">• UNITARIAS• INTEGRACIÓN | <p>Unitarias: Permite verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.</p> <p>Integración: Permite verificar el correcto ensamblaje entre los distintos módulos que componen el sistema desarrollado.</p> | DESARROLLO (CONSTRUCCIÓN) |
| <ul style="list-style-type: none">• SISTEMA.• ESTRÉS.• CONCURRENCIA.• INTERFAZ DE USUARIO.• RENDIMIENTO.• SEGURIDAD | <p>Sistema: Estas pruebas buscan diferencias entre la solución desarrollada y los requerimientos, con el fin de identificar errores que se puedan generar entre la especificación funcional y el diseño del sistema.</p> <p>Volumen: Esta prueba somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software.</p> <p>Estrés: Valida aquellos volúmenes de datos máximos que resiste el sistema antes de comenzar con errores.</p> <p>Concurrencia: Valida la capacidad del sistema de atender múltiples solicitudes de parte de los usuarios que acceden a un mismo recurso.</p> <p>Interfaz de usuario: Permite verificar que la navegación a través de los elementos que se están probando, reflejen las funciones del negocio y los requerimientos funcionales.</p> <p>Rendimiento: Permite validar si la aplicación cumple los criterios de tiempos de respuesta establecidos.</p> | DESARROLLO |



| | | |
|--|---|--|
| <p>INTEROPERABILIDAD.</p> <ul style="list-style-type: none">• DESEMPEÑO.• CONFIGURACIÓN. | <p>Seguridad: Verifica el cumplimiento de las políticas de seguridad acordadas para el sistema.</p> <p>Interoperabilidad: Esta prueba permite verificar todos los artefactos de la solución desarrollada, su arquitectura base, los protocolos de la solución, las interfaces y los módulos del sistema, funcionando en forma conjunta.</p> <p>Desempeño: Este tipo de prueba es un aspecto fundamental en una aplicación pues si ésta no responde en el debido tiempo, se pueden perder clientes, o dañar la imagen ante los usuarios.</p> <p>Configuración: Establece y mantiene la integridad de los productos de software a través del ciclo de vida del proceso del mismo.</p> | |
| <ul style="list-style-type: none">• FUNCIONALES | <p>Funcional: La prueba funcional es un proceso que busca encontrar discrepancias entre el programa y la especificación funcional. Se toma como objeto de prueba cada Caso de Uso.</p> <p>Usabilidad: Esta prueba permite encontrar problemas de factores humanos, en relación con la facilidad de uso, la navegación, diseño amigable, aplicando un conjunto de reglas básicas de usabilidad.</p> | <p>LAS ITERACIONES NECESARIAS</p> |
| <ul style="list-style-type: none">• ACEPTACIÓN | <p>Es la prueba final basada en las especificaciones del usuario o basada en el uso del programa por el usuario final luego de un periodo de tiempo.</p> | <p>IMPLANTACIÓN</p> |



1.4. REFERENCIAS

Algunos documentos del proyecto VDALTON, son la base de este documento inicial de plan de pruebas, que a continuación se especifican:

- Alcances del Proyecto
- Documentos de Especificación de Requerimientos.
- Lista de historias de usuarios.
- Diagramas de clases.

2. ESTRATEGIA DE PRUEBAS

2.1. PLANIFICACIÓN

Se planificarán las pruebas específicas a ser aplicadas, para lo cual se incluye la definición de las pruebas, las estrategias, los recursos y las estimaciones de tiempo.

Esta planificación deberá realizarse para cada sub-sistema en forma separada.

- Se definen las 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.
- Criterios de aceptación.
- Recursos involucrados.

Resultado de la planificación: Se deberán definir los siguientes puntos:

- Cronograma detallado de la ejecución de las pruebas; donde se especifica qué prueba se realiza, cuánto tiempo se estima para su ejecución, recursos a utilizar (humanos y tecnológicos).
- Formatos a utilizar para el diseño de las pruebas.
- Formatos a utilizar para el registro y análisis de los resultados de las pruebas.
- Herramientas a utilizar para la gestión de incidencias.
- Procedimientos para el control de cambios.
- Herramientas a utilizar para la ejecución de las pruebas.



2.2. DISEÑO DE LAS PRUEBAS

A. Definición de Casos de Pruebas: En esta sección se especifica el inventario de las pruebas, el cual permitirá:

- Definir y asignar prioridades como; alta, media o baja.
- Establecer un orden de trabajo.
- Estimación del tiempo en probar cada funcionalidad.

B. Resultados de la ejecución de las Pruebas: En este punto se especifican los resultados de las pruebas, identificando cada prueba y la estimación de esfuerzo

- Inventario de pruebas priorizado.
- Estimación de esfuerzo de cada funcionalidad.
- Plazos previstos para la solución.
- Criterios de Terminación.

2.3. TIPOS DE PRUEBAS

2.3.1. PRUEBAS UNITARIAS

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.

Es una Prueba técnica que permitirá:

- Verificar que los módulos del sistema estén libres de errores.
- Que todos los caminos lógicos principales deben ejecutarse correctamente en cada módulo de la aplicación.
- Todas las transacciones deben ser probados.
- Todos los tipos de registro de entrada válidos deben ser procesados
- Todos los tipos de registro de entrada inválidos deben ser procesados correctamente □Códigos de vuelta no nulos.
- Excepciones a tratamiento normal.
- Todas las salidas válidas son procesadas.
- Rasgos de Control son probados y documentados.



| Objetivo de la Prueba: Validar las piezas individuales del software como una unidad independiente. | |
|--|--|
| Estrategia: | <ul style="list-style-type: none">• Se efectúan para los servicios del negocio y para la lógica de beans en capa Web que tengan complejidad alta.• Generar casos de pruebas necesarios que permitan identificar: o Que al menos cada sentencia o instrucción del programa se ejecute al menos una vez correctamente.• Que cada condición tenga por lo menos una vez un resultado positivo y/o negativo.• Que cada bucle del sistema se pueda probar considerando: - ignorar el bucle, pasar una vez, pasar n veces. |
| Herramienta: | <ul style="list-style-type: none">• JUNIT |
| Observaciones | <ul style="list-style-type: none">• La prueba se realizará por Módulo entendiéndose por tal:• Bloque básico de programa• Implementa función independiente y simple• Puede probarse por separado. |

2.3.2. PRUEBAS DEL SISTEMA

Las pruebas de sistema buscan diferencias entre la solución desarrollada y los requerimientos, enfocándose en la identificación de los errores que se puedan generar entre la especificación funcional y el diseño del sistema, así como, el negocio objeto de la aplicación.

| | |
|------------------------|---|
| Objetivo de la Prueba: | <ul style="list-style-type: none">• Validar aquellos volúmenes de datos máximos (por lo general las transacciones o informes) que pueden ser completados dentro de un período específico en el tiempo, y con un nivel de concurrencia dado (carga, concurrencia y desempeño).• Validar los requerimientos no funcionales de cada proyecto. |
|------------------------|---|



| | |
|----------------|--|
| Estrategia: | <ul style="list-style-type: none">• Realizar Set de Pruebas a partir de los Requerimientos no funcionales.• Realizar pruebas de rendimiento básico. Consiste en probar la aplicación simulando la carga esperada en el entorno de producción.• Realizar las pruebas de concurrencia: verificar el comportamiento de la aplicación en condiciones de sobrecarga de usuarios, que supone permitirá identificar potenciales problemas de rendimiento o cuellos de botella, antes de su pase a producción.• Realizar pruebas de requerimientos no funcionales: Consiste en probar la aplicación con cada uno de los requerimientos no funcionales establecidos en el proyecto.• Identificar posibles cuellos de botella o problemas de rendimiento.• Realizar pruebas de carga: Altos volúmenes de información. |
| Herramientas: | <ul style="list-style-type: none">• JUNIT• HTTPUNIT• JMETER o THE GRINDER 3 |
| Observaciones: | La elección de la herramienta de prueba será a discreción del grupo de |

2.3.2. PRUEBAS DE INTEGRACIÓN

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos módulos que componen la solución una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces internas y externas, que cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

En esta prueba se comprueba la compatibilidad y funcionalidad de los interfaces entre las distintas 'partes' que componen el desarrollo de la solución. Estas partes pueden ser módulos, aplicaciones individuales, es decir esta prueba válida la



integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta, teniendo en cuenta los siguientes temas técnicos:

- El funcionamiento integrado de módulos interdependientes debe estar libre de errores
- Probar todas las dependencias entre módulos
- Probar el flujo de control y el flujo de datos a través de todas las capas

| | |
|---------------------------|--|
| Objetivo de la Prueba: | <ul style="list-style-type: none">• Validar la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta |
| Estrategia: | <ul style="list-style-type: none">• Pruebas de Integración Incremental Ascendente Automática |
| Herramientas requeridas: | <ul style="list-style-type: none">• NetBeans |
| Criterios de Terminación: | <ul style="list-style-type: none">• Ausencia de error |

2.3.4. PRUEBAS DE INTEROPERABILIDAD

En esta prueba se valida que el sistema se comuniquen de manera exitosa con los sistemas externos con que se requiera, de acuerdo con los requerimientos no funcionales. Estos sistemas pueden ser sistemas propios de cada entidad, servicios de estampado de tiempo, servicios para la validación de certificados digitales, servicios publicados en el tramitador transaccional, servicios de envío de SMS, servicios de envío de correo electrónico.

| | |
|------------------------|---|
| Objetivo de la Prueba: | <ul style="list-style-type: none">• Validar la interoperabilidad de la solución con sistemas externos. |
| Estrategia: | <ul style="list-style-type: none">• Prueba directa con los servicios que se encuentren disponibles en un ambiente de pruebas controlado suministrado por Agenda de conectividad.• Para los servicios que no estén disponibles para prueba se desarrollarán sistemas DUMMIE que garanticen que la integración funcionará. |



| | |
|-------------------------|---|
| | <ul style="list-style-type: none">• Para los archivos XML generados por la aplicación se verificará que sean válidos según la definición de los esquemas xsd definidos para cada uno. La verificación de los archivos GEL- XML la realizará cada desarrollador, aprovechando las características de la herramienta utilizada para estas pruebas.• La autenticación de los sistemas externos se probará mediante un DUMMIE; el cual llevará el usuario y contraseña de autenticación frente al sistema de notificaciones. Se probarán 3 escenarios uno con el usuario incorrecto, otro con la clave incorrecta y finalmente un escenario con el usuario y clave correctos.• Las pruebas de conexión con el tramitador de gobierno en línea se verificará mediante una prueba técnica utilizando y siguiendo los pasos del Wizard de conexión java con el tramitador de gobierno en línea; de esta prueba se espera que la conexión quede establecida.• Se expondrá un servidor de FTP externo, con el cual se realizarán transferencias masivas de documentos históricos, se probaran tres escenarios: (Correcto, Incorrecto e incompleto). |
| Herramienta requeridas: | <p>DUMMIES de los servicios que no están contruidos o no están disponibles:</p> <ul style="list-style-type: none">• Autenticación• Firma Digital• Estampado de Tiempo• Sistema Emisor <p>XSD de cada archivo XML generado. XMLBuddy</p> |
| Observaciones: | |

2.3.5. PRUEBAS DE REGRESIÓN En esta prueba se valida que el sistema mantenga su correcta funcionalidad después de la incorporación de un ajuste, corrección o nuevo requerimiento. Es una prueba funcional y técnica que valida que el sistema siga funcionando perfectamente después de que las correcciones sean aplicadas.



| | |
|------------------------|--|
| Objetivo de la Prueba: | <ul style="list-style-type: none">• Validar que el sistema siga funcionando perfectamente después de que las acciones correctivas sean aplicadas. |
| Estrategia: | <ul style="list-style-type: none">• Repetir las pruebas (unitarias, de integración, funcionales y de carga) que se hicieron antes de corregir defectos o de añadir nuevas funcionalidades, para comprobar que las modificaciones no provocan errores donde antes no los había. |
| Herramientas: | <ul style="list-style-type: none">• Utilizar las mismas herramientas usadas para las pruebas según sea el caso. |
| Observaciones | <ul style="list-style-type: none">• Los responsables de las Pruebas de Regresión se establecen dependiendo del momento en el que se realicen las modificaciones. |

2.3.6. PRUEBAS FUNCIONALES

La prueba funcional es un proceso para procurar encontrar discrepancias entre el software desarrollado y la especificación funcional. La prueba funcional normalmente es una actividad de caja negra. Esta prueba permite validar:

- Los procesos y reglas de negocio establecidas,
- Que se cumplan los requerimientos funcionales establecidos

En esta prueba se validan los Casos de Uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los set de pruebas correspondientes. Se deben elaborar los casos de pruebas necesarios que permitan asegurar el funcionamiento de todos los flujos normales y alternos de dichos casos de uso.

| | |
|------------------------|---|
| Objetivo de la Prueba: | Se asegura el trabajo apropiado de los requisitos funcionales, Incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. |
|------------------------|---|



| | |
|--------------------------|--|
| Estrategia: | Validación y ejecución de Set de Pruebas y escenarios definidos, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos para verificar lo siguiente: <ul style="list-style-type: none">○ Los resultados esperados ocurren cuando se usan datos validos. ○ Se despliegan mensajes de error cuando se usan datos inválidos.○ Cada regla de negocio es propiamente aplicada.○ Realizar set de pruebas de los requerimientos mínimos para el adecuado funcionamiento de la aplicación |
| Herramientas Requeridas: | <ul style="list-style-type: none">• HTTPUNIT cuando aplique• Formato de casos de prueba funcionales |
| Observaciones: | Para el reporte de incidencias se utilizará una herramienta para el registro |

2.3.7. PRUEBAS DE USABILIDAD

Las pruebas de usabilidad son una forma de medir que tan bien puede una persona usar un objeto hecho por el hombre, como puede ser una página web, una interfaz de usuario, un documento o un dispositivo.

Las pruebas de usabilidad consisten en seleccionar a un grupo de usuarios de una aplicación y solicitarles que lleven a cabo las tareas para las cuales fue diseñada, en tanto el equipo de diseño, desarrollo y otros involucrados toman nota de la interacción, particularmente de los errores y dificultades con las que se encuentren los usuarios.

No es necesario que se trate de una aplicación completamente terminada, pudiendo tratarse de un prototipo

| | |
|------------------------|--|
| Objetivo de la Prueba: | Validar el grado de usabilidad empírico del sistema. |
|------------------------|--|



| | |
|-------------|--|
| Estrategia: | <p>Se usarán cuatro métricas principales para medir la usabilidad del sistema</p> <ul style="list-style-type: none">○ Exactitud: Número de errores cometidos por los sujetos de prueba y si estos fueron recuperables o no al usar los datos o procedimientos adecuados.○ Tiempo requerido para concluir la actividad.○ Recuerdo: Qué tanto recuerda el usuario después de un periodo sin usar la aplicación.○ Respuesta emocional: Cómo se siente el usuario al terminar la tarea (bajo tensión, satisfecho, molesto, etcétera). |
|-------------|--|

2.3.8. PRUEBAS DE SEGURIDAD

Estas pruebas tienen dos enfoques:

- Pruebas de seguridad de la aplicación; donde se verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.
- Pruebas de seguridad del sistema; donde se verifica que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla.

| | |
|------------------------|---|
| Objetivo de la Prueba: | <ul style="list-style-type: none">• Que los usuarios están restringidos a funciones específicas o su acceso está limitado únicamente a los datos que está autorizado a acceder.• Que solo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones del sistema.• Que el cortafuego oculte apropiadamente la aplicación.• Que los puertos clave solo puedan ser accedidos de la forma establecida para su acceso.• Que los puertos restringidos efectivamente no se encuentren accesibles. |
| Estrategia: | <ul style="list-style-type: none">• Identificar cada tipo de usuario y las funciones y datos a los que se debe autorizar.• Crear pruebas para cada tipo de usuario y verificar cada permiso, creando transacciones específicas para cada tipo de usuario. |



| | |
|--------------------------|--|
| | <ul style="list-style-type: none">• Modificar tipos de usuarios y volver a ejecutar las pruebas. |
| Herramientas Requeridas: | <ul style="list-style-type: none">• Nessus• Pruebas funcionales de seguridad. |
| Observaciones: | Para el reporte de incidencias se utilizará una herramienta para el registro y seguimiento. |

2.3.9. PRUEBAS DE CONFIGURACIÓN

El propósito de esta prueba es establecer y mantener la integridad de los productos de software a través del ciclo de vida del proceso del mismo. Esta prueba implica la identificación de la Configuración del software en puntos dados en el tiempo, el control sistemático de los cambios en la Configuración y el mantenimiento de la integridad y trazabilidad de la Configuración a través del ciclo de vida del software.

| | |
|--------------------------|---|
| Objetivo de la Prueba: | Validar la integridad de los productos de software. |
| Estrategia: | <ul style="list-style-type: none">• Validación y ejecución de Set de Pruebas que representen un ciclo del proceso de negocio principal de principio a fin.• Validación de la integridad de la configuración de todos los sistemas involucrados en puntos datos en el tiempo.• Realizar trazabilidad de los cambios de configuración realizados para puesta a punto. |
| Herramientas Requeridas: | HTTPUNIT |
| Observaciones: | Para el reporte de incidencias se utilizará una herramienta para el registro y seguimiento. |

2.3.10. PRUEBAS DE RECUPERACIÓN A FALLAS

Estas pruebas aseguran que el software pueda recuperarse a fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos.



El objetivo de esta prueba es verificar que los procesos de recuperación (manual o automáticos) se realice apropiadamente: las base de datos, las aplicaciones y el sistema a un estado conocido y deseado.

En la prueba se incluyen los siguientes tipos de condiciones:

- Interrupción de energía al cliente
- Interrupción de energía al servidor
- Interrupción de comunicaciones mediante los servidores de la red
- Interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores
- Ciclos incompletos (procesos de filtro de datos interrumpidos, procesos de sincronización de datos interrumpidos)

| | |
|------------------------|--|
| Objetivo de la Prueba: | <ul style="list-style-type: none">• Validar la capacidad de recuperación a fallas de: Hardware o Software Mal funcionamiento de Red. |
| Estrategia: | <ul style="list-style-type: none">• Interrumpir la energía del cliente: apagar el PC.• Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.• Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red. |

2.3.11. PRUEBAS DE ACEPTACIÓN

El objetivo de las pruebas de aceptación es validar que la solución desarrollada cumpla con el funcionamiento esperado y permitir al usuario de dicho sistema determine su aceptación, desde el punto de vista de su funcionalidad y de su rendimiento. Estas pruebas son realizadas por el cliente, donde comprueba que el sistema cumple con lo definido y se obtiene la conformidad del cliente. Esta prueba se realiza mediante el proceso de validación de caja negra.

Estas pruebas corresponden a la ejecución de las siguientes pruebas por parte de los usuarios funcionales o cliente:

- Pruebas Funcionales.
- Pruebas de Usabilidad.



- Pruebas de Configuración

2.4. ENTREGABLES DE PRUEBAS

De acuerdo con el tipo de pruebas ejecutadas puede que el entregable del mismo sea diferente, en el siguiente cuadro se señalan los diferentes entregables por tipo de prueba.

| TIPO DE PRUEBAS | ENTREGABLES |
|------------------------------|--|
| Pruebas Unitarias | Resumen de validación de la prueba. |
| Pruebas de Sistema | Se entregará un documento de resultados de las pruebas realizadas, que incluya resultados de la ejecución de los scripts de pruebas y análisis de los errores o defectos encontrados durante el proceso de realización de las pruebas. |
| Pruebas de Integración | Se entregará un documento de pruebas de integración que incluye resultados de la ejecución de los scripts de pruebas y análisis de los defectos encontrados durante el proceso de pruebas |
| Pruebas de Interoperabilidad | Se entregará un documento de pruebas de interoperabilidad que |

3. EVALUACIÓN DE PRUEBAS EJECUTADAS

3.1. CRITERIOS DE INICIO DE EJECUCIÓN

A continuación se señalan las condiciones mínimas que se deben presentar para iniciar la ejecución de las pruebas:

- Se poseen los set de pruebas aprobados con escenarios claros.
- El entorno de pruebas es el adecuado para el tipo de pruebas a iniciar.
- Todos los artefactos requeridos se encuentran disponibles.
- Se recibió la Versión del Software para pruebas con su correspondiente Release Note y Lista de Chequeo cuando esta aplique.
- Todos los recursos humanos y técnicos necesarios se encuentran disponibles.

3.2. CRITERIOS DE EVALUACION



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 EVALUACION |
|------------------------------|---|
| Pruebas Unitarias | Detectar errores en la ejecución de las pruebas. |
| | El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de Sistema | Detectar errores en la ejecución de las pruebas |
| | Que los reportes generados por las herramientas de automatización de las pruebas contengan las mínimas variables que permitan un análisis acertado de cada una de las pruebas realizadas. |
| | Tener en cuenta todos los escenarios posibles. |
| | El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de Integración | La totalidad de los puntos de control probadas debe ser mayor al 75% del total de los componentes que integran la solución. |
| | Detectar errores en la ejecución de las pruebas |
| | El 90% de las pruebas realizadas deben ser exitosas. |
| Pruebas de Interoperabilidad | Detectar errores en la ejecución de las pruebas |
| | El 90% de las pruebas realizadas deben ser exitosas contra los servicios del tramitador. |
| | El 90% de las pruebas realizadas deben ser exitosas contra los |