



La vivienda y el agua
son de todos

Minvivienda

PLAN MAESTRO DE PRUEBAS

Mayo de 2021

BOGOTÁ D.C. - COLOMBIA

Contrato No. 001 de 2019 – Arquitectura Empresarial

Objeto del Contrato: Diseñar la arquitectura del Sistema de Información del Subsidio Familiar de Vivienda que comprenda los componentes de oferta y demanda de subsidios de vivienda





CONTROL DE CAMBIOS			
FECHA	VERSIÓN	DESCRIPCIÓN DEL CAMBIO	RESPONSABLE
21/01/2021	0.1	Versión inicial del documento.	UT FONVIVIENDA 2019
01/02/2021	0.2	Se aceptaron los controles de cambio propuestos y se atendieron los comentarios por parte de interventoría. Ajustes Corrección de estilo y forma al documento	UT FONVIVIENDA 2019
01/03/2021	1.0	Versión aprobada	UT FONVIVIENDA 2019

REFERENCIAS CONTRACTUALES DEL ENTREGABLE:

APROBACIÓN				
ACCIÓN	NOMBRE	ROL	FIRMA	FECHA
Elaboro	Cesar Peña	Consultoría		
Revisó	William Palencia	Gerente de Proyecto Interventoría		
Aprobó	Carlos Gutiérrez	Supervisor FONVIVIENDA para el contrato 01 de 2019 Fiduciaria de Occidente y UT FONVIVIENDA 2019		

AVISO DE CONFIDENCIALIDAD



La vivienda y el agua
son de todos

Minvivienda

ARQUITECTURA Y DISEÑO DEL
SISTEMA DE INFORMACIÓN DEL
SUBSIDIO FAMILIAR DE VIVIENDA

PLAN MAESTRO DE PRUEBAS



Versión 0.2

La UT FONVIVIENDA 2019 en aras de preservar la Seguridad de la Información del Ministerio de Vivienda, entrega este documento bajo la condición de confidencialidad mutua, donde las partes deben respetar la información provista. Por lo tanto, la información contenida en este documento y en los medios magnéticos entregados es de carácter reservado y sólo puede ser utilizado por el personal que el Ministerio de Vivienda designe para su revisión, resguardo, manipulación y/o divulgación. Las normas que fundamentan el carácter reservado de la información son los artículos 72 y siguientes de la decisión del acuerdo de Cartagena 344 de 1993, el artículo 238 del Código Penal Colombiano y los artículos 16 y siguientes de la Ley 256 de 1996.



TABLA DE CONTENIDO

	Pág.
1 INTRODUCCIÓN	8
1.1 PROPOSITO DEL DOCUMENTO	8
1.2 ALCANCE DEL PROYECTO	8
1.3 ALCANCE DEL PLAN	8
1.4 STAKEHOLDERS PRINCIPALES	9
1.5 CARACTERÍSTICAS DE LA METODOLOGÍA, TÉCNICAS Y HERRAMIENTAS....	10
1.5.1 Resumen de la Metodología	10
1.5.2 Resumen de técnicas de prueba	11
1.5.3 Herramientas de control.....	11
2 OBJETIVOS Y FACTORES DE MOTIVACIÓN DE PRUEBAS	13
2.1 MOTIVADORES	13
2.2 ELEMENTOS A PROBAR	13
2.2.1 Aspectos técnicos	13
2.3 ELEMENTOS QUE NO SERÁN SUJETOS DE PRUEBA	14
2.3.1 Aspectos funcionales	15
3 ESTRATEGIA DE PRUEBAS	16
3.1 PRUEBAS UNITARIAS (COBERTURAS)	16
3.2 PRUEBAS DE INTEGRACIÓN (COBERTURA).....	17
3.2.1 Pruebas de usabilidad	17
3.2.2 Pruebas de interoperabilidad	17
3.2.3 Pruebas de seguridad	17
3.3 PRUEBAS DE VALIDACIÓN (PRUEBA DE HUMO)	17
3.4 PRUEBAS DE SISTEMA.....	17
3.4.1 Pruebas funcionales	18
3.4.2 Pruebas de Interfaz de Usuario	18
3.4.3 Pruebas de Usuario (UAT).....	18



3.5	PRUEBAS AUTOMATIZADAS	18
3.5.1	Pruebas de Desempeño y Volumen	18
3.5.2	Pruebas de Stress	18
3.5.3	Prueba de concurrencia.....	19
3.5.4	Prueba de Integridad de las bases de datos	19
3.5.5	Pruebas de Regresión	19
4	CRITERIOS DE ACEPTACIÓN Y/O RECHAZO DE “ELEMENTOS” DE PRUEBA 20	
5	CRITERIOS DE suspensión y condiciones DE reanudación PRUEBA.....	21
6	ENTREGABLES DE PRUEBA	22
7	NECESIDADES DE PERSONAL.....	27
7.1	DEFINICIÓN DEL EQUIPO DE TRABAJO REQUERIDO	27
7.2	DISTRIBUCIÓN DEL TRABAJO.....	27
7.2.1	Definición de roles y responsabilidades (Matriz RACI)	27
7.2.2	Definición técnica del equipo de pruebas	28
8	PROGRAMA DE ACTIVIDADES DE PRUEBA	30
9	ALInEAMIENTO A METODOLOGÍAS ÁGILES.....	31
9.1	AUTOMATIZACIÓN DE PRUEBAS.....	31
9.2	AMBIENTES DE PRUEBAS CON DEVOPS	32
9.3	ENFOQUE EJEMPLO TDD.....	32
9.4	CICLOS DE PRUEBAS	33
9.5	VENTAJAS: DISMINUCIÓN DE RIESGOS	34
10	Bibliografía.....	36



LISTADO DE TABLAS

	Pág.
Tabla 1. Stakeholders principales.....	10
Tabla 2. Herramientas propuestas	12
Tabla 3. Funcionalidades a Probar	14
Tabla 4. Criterios de Aceptación.....	20
Tabla 5. Escenarios de prueba 1	23
Tabla 6. Escenarios de prueba 2	24
Tabla 7. Escenarios de prueba 3.....	25
Tabla 8. Escenarios de prueba 4.....	25
Tabla 9. Matriz RACI.....	28



LISTADO DE ILUSTRACIONES

	Pág.
Ilustración 1. Estrategia de Pruebas.....	16
Ilustración 2. Pirámide Automatización de Pruebas.....	31
Ilustración 3. Proceso de Integración continua (Ciclo de Vida del Software).....	32
Ilustración 4. Enfoque TDD.....	33
Ilustración 5. Ciclo de Pruebas.....	34



1 INTRODUCCIÓN

1.1 PROPOSITO DEL DOCUMENTO

El propósito de este documento es establecer una guía con las principales consideraciones que se deben tener en cuenta en la etapa de pruebas del proyecto, con el objetivo de garantizar la calidad del software.

Hace referencia a puntos relevantes de la disciplina de pruebas y buenas prácticas que pueden ser aplicadas durante la ejecución de las pruebas de software y que permitirán a la entidad tener la identificación de los puntos clave de control como son entre otras las siguientes:

- Tipos de pruebas que se recomienda ejecutar para garantizar la calidad del software
- Criterios de suspensión y/o reanudación de las pruebas de software. Principales entregables de las pruebas de software.
- Características del equipo de trabajo

1.2 ALCANCE DEL PROYECTO

El alcance del Objeto Contractual incluye la prestación del servicio de consultoría para diseñar la arquitectura del Sistema de Información del Subsidio Familiar de Vivienda (SISFV). Para dar alcance al presente documento es necesario identificar cual es el ítem sobre el cual se enmarca contractualmente, este corresponde con el literal (d) citado en el documento Anexo1. Plan de gestión del Alcance¹.

“d) Desarrollar una propuesta que integre el marco de intercambio de información del sistema, en el diseño de la arquitectura del SISFV con sus componentes de oferta y demanda, a fin de facilitar el intercambio seguro y transparente de la información que se genera entre las entidades que participan en la política de vivienda.”

1.3 ALCANCE DEL PLAN

Este documento, tiene como finalidad establecer una guía con pautas, estrategias, técnicas, herramientas y actividades que se recomiendan para ser ejecutadas durante la etapa de Construcción del Software, en la fase de ejecución de las pruebas, esto con el objetivo de tener insumos que permitan garantizar el cumplimiento de los requisitos planteados para la certificación del software SISFV.

¹ Anexo 1. Plan de Gestión del Alcance.PDF - ENUNCIADO DETALLADO DEL ALCANCE DEL PROYECTO Alcance funcional, literal d.



1.4 STAKEHOLDERS PRINCIPALES

Para la certificación del Software SISFV, se requiere la participación de los actores funcionales y/o técnicos involucrados en la definición de los diferentes requerimientos del sistema y quienes serán los responsables de dar la aprobación y/o certificación del funcionamiento y cumplimiento de los diversos criterios definidos para su respectiva implementación. (Tabla 1).

ÁREA	FUNCIONES
Subsidio Familiar de vivienda	El equipo de procesos de gestión de subsidio tiene como responsabilidad realizar las pruebas de los siguientes módulos y/o funcionalidades: <ul style="list-style-type: none"> • Gestionar Resoluciones • Gestionar Lista Hogares • Administrar formulario postulación • Generar reportes de Postulación • Gestionar Novedades
Promoción y apoyo técnico	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Promoción y Apoyo técnico. El objetivo principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio
Espacio urbano y territorial	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Licencias Urbanísticas. El objetivo principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio
Acompañamiento social	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Promoción y Acompañamiento. El objetivo principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio
Grupo de titulación y saneamiento predial	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Titulación y saneamiento predial. El objetivo principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio
Oficina asesora de planeación	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Gestión de proyectos. El objetivo principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio
Grupo de contratos	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Gestión de proyectos. El objetivo principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio
Grupo acompañamiento social (Pagos) - Subsidios	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Gestión de Subsidio. El objetivo



	principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio
Grupo vivienda rural	Hacer parte del equipo de Certificación de las funcionalidades correspondientes a los procesos de Focalizar para la asignación de subsidio rural. El objetivo principal en esta parte del proceso es asegurar que las funcionalidades cumplen con los requisitos establecidos y son relevantes para el negocio

Tabla 1. Stakeholders principales

Fuente: UT Fonvivienda 2019, 2020

1.5 CARACTERÍSTICAS DE LA METODOLOGÍA, TÉCNICAS Y HERRAMIENTAS

1.5.1 Resumen de la Metodología

La metodología sugerida estará orientada a la ejecución y documentación de las siguientes etapas:

- **Planeación de Pruebas:** Etapa en la cual se determinan los tipos de prueba a ser aplicados.
- **Preparación del Ambiente:** Con el fin de poder ejecutar las pruebas se debe contar con un ambiente de pruebas adecuado independiente de desarrollo y controlado por el equipo de pruebas, que cuente entre otras con estas características.
 - Este ambiente debe ser lo más parecido al ambiente productivo y se deben tener en cuenta que las aplicaciones instaladas correspondan a las versiones lo más cercanas posibles a las productivas.
Se recomienda realizar la réplica de todos los componentes con los cuales hay interoperabilidad.
Para las herramientas de gestión de casos de prueba y gestión de bugs es recomendable que sean instalados en servidores separados de los ambientes de pruebas de y desarrollo.
- Contar con herramientas de gestión de versiones. **Ejecución de Pruebas de Interfaz:** Se debe contar con el mapa de navegación de la aplicación para identificar el orden funcional en el que se debe ejecutar las pruebas.
- **Ejecución de pruebas de Integridad:** se identifican los accesos y formas como los diferentes servicios pueden modificar los datos y se realizan pruebas para verificar que se mantiene una integridad de la información persistente.
- **Ejecución de pruebas de desempeño:** Se determina el número de usuarios que pueden tener acceso al sistema y los picos de en los cuales el volumen puede ser considerable para medir el tiempo de respuesta.



- **Ejecución de pruebas de carga:** Se determina el número de usuarios que pueden tener acceso al sistema y los picos de en los cuales el volumen puede ser considerable para establecer las características necesarias para que el sistema soporte el volumen de información que puede procesar.
- **Presentación de Resultados:** se debe generar un reporte que permita identificar el resumen de los indicadores medidos en las pruebas y el resultado.

1.5.2 Resumen de técnicas de prueba

De acuerdo con el estándar ISTQB, las técnicas que se deben realizar en cualquier software y que se recomiendan para ser utilizadas en la implementación de este sistema son las siguientes:

- **Pruebas de Caja Negra (Pruebas Funcionales):** Se ejecutan teniendo en cuenta las entradas y las salidas del software, es decir, solo se tienen en cuenta los requisitos funcionales y obvia la estructura de control.
- **Pruebas de Caja Blanca (Pruebas de desarrollo):** El objetivo es realizar las pruebas sobre las funciones internas del software, validando el código y la estructura del software y son utilizadas en las pruebas de unidad.
- **Pruebas de Interfaz de Usuario (Pruebas Funcionales):** Orientadas al uso de las pantallas teniendo en cuenta el flujo de información y la lógica de negocio, con el fin de identificar los posibles errores desde el punto de vista de un usuario final y su percepción sobre la facilidad de uso de la interfaz
- **Pruebas de Desempeño (Pruebas no funcionales):** Orientadas a medir las características del software y su comportamiento frente a los atributos de calidad.

1.5.3 Herramientas de control

A continuación, se presentan algunas características con las cuales deberían contar las herramientas a ser utilizadas para la ejecución, seguimiento y control de la fase de pruebas (Tabla 2). Sin embargo, es importante aclarar que para poder definir las herramientas que se ajustan a las características del proyecto de construcción del software es recomendable realizar en su momento la identificación de características necesarias a nivel de infraestructura, de características de los lenguajes de programación y de los ambientes en los cuales se va a probar para cerrar el espectro de las posibles herramientas y establecer un grupo sobre las cuales poder realizar pruebas de concepto y poder seleccionar las que más se ajusten.



Tipo Herramienta	DESCRIPCIÓN
Pruebas Funcionales	Herramienta que permita la creación de casos de prueba para aplicaciones web y la automatización de pruebas de rendimiento, así como para la prueba de la interfaz web.
Pruebas de Carga y Stress	Herramienta que permita la automatización de pruebas de rendimiento y carga. flexibilidad para utilizar los lenguajes de programación más populares (C #, Java, Perl, PHP, Python, entre otros), compatibilidad con la mayoría de los navegadores web y sistemas operativos, permita ejecutar pruebas remotas
Gestión de Casos de prueba	Herramienta que permita crear y gestionar casos de prueba, organizar por planes de prueba, facilitar la realización del seguimiento de los resultados y la trazabilidad de los casos de prueba con los requisitos.
Gestión de Bugs	Herramienta que permita realizar seguimiento de los errores, que incluya sistema de usuarios para que múltiples usuarios puedan interactuar y realizar varios proyectos, compatible con opciones de control de versiones

Tabla 2. Herramientas propuestas

Fuente: UT Fonvivienda 2019, 2020



2 OBJETIVOS Y FACTORES DE MOTIVACIÓN DE PRUEBAS

El propósito principal de las Pruebas de Software es el encontrar los posibles fallos de implementación, calidad o usabilidad del sistema, desde el punto de vista funcional no funcional y de negocio, con el fin de garantizar que cumple con las expectativas de los interesados y con los requerimientos y especificaciones funcionales definidas por el cliente.

2.1 MOTIVADORES

Proporciona una breve descripción que define la misión para el esfuerzo de prueba. Esta descripción puede incorporar una o más preocupaciones como:

- Encontrar los errores que sea posible.
- Encontrar los problemas importantes.
- Evaluar si los riesgos percibidos han sido mitigados.
- Certificar un estándar.
- Verificar una especificación (requerimientos, diseño o pretensión).
- Notificar acerca de la calidad del producto.
- Validar el cumplimiento de las reglas de negocio.
- Validar el comportamiento del negocio conforme a lo especificado.
- Validar que los flujos de negocio están contemplados y funcionan correctamente
- Asegurar la usabilidad del sistema y el buen funcionamiento de la interfaz de usuario.
- Las respuestas del sistema son las esperadas conforme con las definiciones y requerimientos.

2.2 ELEMENTOS A PROBAR

2.2.1 Aspectos técnicos

La base principal para determinar los aspectos técnicos a probar, son los requerimientos no funcionales para poder encontrar problemas en el sistema en cuanto al comportamiento futuro de la solución. Se debe considerar: (1) los escenarios derivados de los requerimientos no funcionales. (2) Consideraciones de volúmenes en el momento de hacer las pruebas y (3) las condiciones de ambientes que se tengan durante la ejecución de estas. Es importante que estas pruebas estén fundamentadas en las características de infraestructura que se tengan en el momento de las pruebas.



Con base en lo anterior, se relacionan las funcionalidades a ser probadas que se detallan en la Tabla 3.

PROCESO	SUBPROCESO	CASOS DE USO
GESTION SUBSIDIO	ABONO CUENTA CAP	4
	APERTURA CUENTA	3
	ASIGNACION SUBSIDIO	34
	AUTORIZACION PAGO 20%	7
	LEGALIZACION TRANSFERENCIA PVG	3
	MOVILIZACIONES	3
	PROCEDIMIENTO PAGO MI CASA YA	2
	PROCESO FIDUCIARIO	12
	DISTRIBUCION RECURSOS SFM	7
	ASIGNACION SUBSIDIO RURAL	2
	FOCALIZAR SUBSIDIO RURAL	2
LICENCIAS URBANISTICAS	LICENCIAS URBANISTICAS	21
	CANCELACION DE GRAVAMENES	8
	IDENTIFICACION DE BIENES INMUEBLES	5
	PROMOCION Y ACOMPAÑAMIENTO TITULACION	5
TRANSVERSALES	TRANSVERSALES	78
GESTION DE PROYECTOS	SUPERVISION Y SEGUIMIENTO PROYECTOS	6
Total		202

Tabla 3. Funcionalidades a Probar

Fuente: UT Fonvivienda 2019, 2020

2.3 ELEMENTOS QUE NO SERÁN SUJETOS DE PRUEBA

En esta etapa no es conveniente descartar las pruebas de ninguna funcionalidad, dado que no se conocen a priori las características que tendrá el proceso de desarrollo del software; sin embargo, es importante tener las siguientes consideraciones y su verificación en la fase de ejecución de las pruebas del software.

- Se debe considerar la estimación en tiempo de la ejecución de pruebas para un universo de casos de prueba como el relacionado que dependiendo de la complejidad se



puede estimar que el esfuerzo relativo de pruebas funcionales puede corresponder a un 15 % del total del esfuerzo realizado en Desarrollo.

- Las pruebas No funcionales pueden estimarse en un 10 % del total del esfuerzo realizado en desarrollo.
- Si existen restricciones de tiempo es necesario priorizar las funcionalidades conforme con su impacto en el negocio y este trabajo se debe realizar en conjunto con los usuarios funcionales y los interesados para acordar las prioridades, tanto en desarrollo como en pruebas.
- Contar con un equipo de pruebas experimentado que garantice la atención en cantidad que pueda atender del volumen de las pruebas en tiempo y esfuerzo. La cantidad de personas en el equipo de pruebas se vuelve trascendental y debe estar contemplado en razón a la granularidad que se pueda hacer de los casos de prueba puesto que existen razones funcionales y técnicas por las cuales una prueba no se puede dividir para que varias personas la ejecuten.
- Se deben considerar también las herramientas tecnológicas con las que cuente el equipo de pruebas, en la medida que se deban realizar tareas manuales se hace más difícil la labor de las pruebas y los resultados en tiempos cortos.

2.3.1 Aspectos funcionales

Desde el punto de vista funcional no se descarta la prueba de ninguna funcionalidad; sin embargo; en el proceso de detallado de los casos de uso, así como en el proceso de diseño técnico es posible identificar escenarios comunes que permitan factorizar unas funcionalidades y reutilizar otras, con lo cual se podría inferir que la cantidad de casos de uso a probar puede disminuir.



3 ESTRATEGIA DE PRUEBAS

La estrategia de pruebas está encaminada principalmente a la ejecución de cuatro (4) tipos de pruebas, como se presenta a continuación (ISO, ISO / IEC 29119-2: Procesos de prueba, 2013).

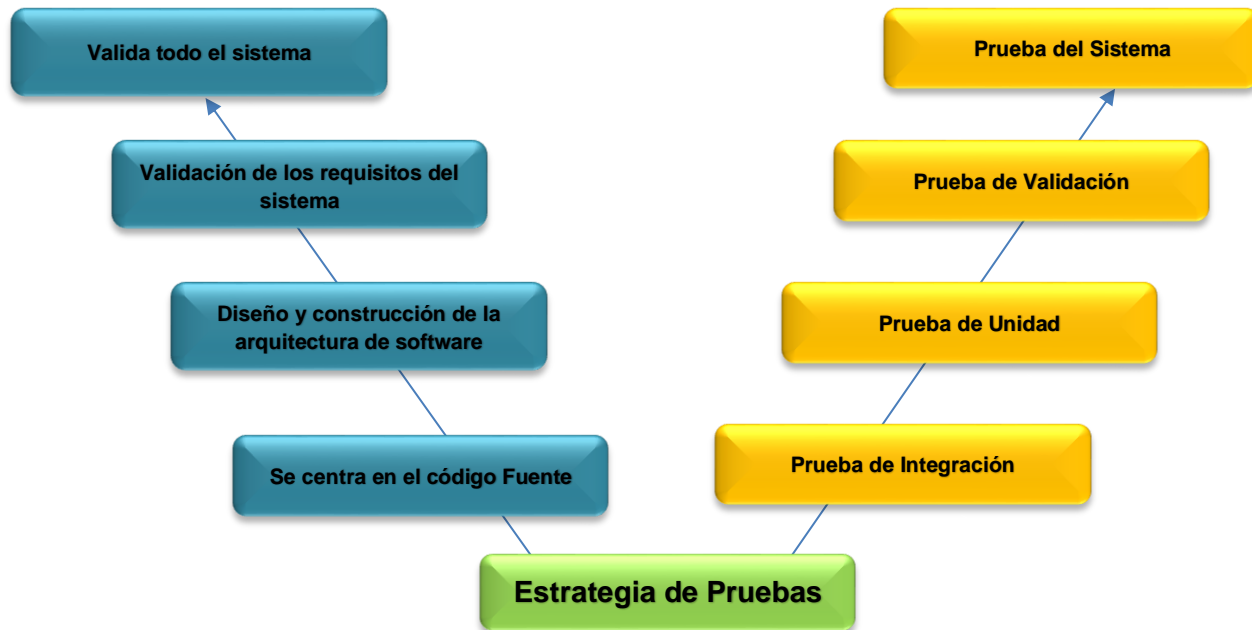


Ilustración 1. Estrategia de Pruebas

Fuente: UT Fonvivienda 2019, 2020

Para poder dar cumplimiento a la premisa anterior se identifican los tipos de pruebas necesarios.

- Pruebas Unitarias
- Pruebas de integración
- Pruebas de Validación (pruebas de Humo)
- Pruebas Sistema (Funcionales)

3.1 PRUEBAS UNITARIAS (COBERTURAS)

Permite verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado, estas pruebas deben estar a cargo del Desarrollador y se debe garantizar la correcta ejecución y documentación, con el objetivo de que la labor de pruebas

funcionales sea más eficiente y la calidad del software se pueda garantizar en el proceso de desarrollo (EAFIT, 2010).

En este punto también es importante establecer procesos para la prueba del código y la implementación de buenas prácticas.

3.2 PRUEBAS DE INTEGRACIÓN (COBERTURA)

Permite verificar el correcto ensamblaje entre los distintos módulos que componen el sistema desarrollado.

3.2.1 Pruebas de usabilidad

Permite evaluar la interacción del usuario con la aplicación, la facilidad uso de forma intuitiva, si necesidad de ser un usuario experto.

Desde el punto de vista de usabilidad es importante incorporar desde etapas muy tempranas la visión de experiencia de usuario, con el objetivo de mejorar la interacción del usuario con la funcionalidad y optimizar rutinas que a nivel de desarrollo pueden implicar más esfuerzo.

3.2.2 Pruebas de 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.

3.2.3 Pruebas de seguridad

Verifica el cumplimiento de las políticas de seguridad acordadas para el sistema.

3.3 PRUEBAS DE VALIDACIÓN (PRUEBA DE HUMO)

Permiten hacer una inspección del software y verificar de forma general su funcionamiento, se caracterizan por permitir una identificación temprana de comportamientos erróneos antes del inicio formal de las pruebas funcionales.

3.4 PRUEBAS DE 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.

En esta etapa, se debe contar con el diseño de los casos de prueba que deben ser ajustados al cumplimiento de los criterios de calidad definidos y al cumplimiento de las reglas de negocio.

3.4.1 Pruebas funcionales

Permiten comprobar que el software cumple con los requisitos y funciones para las cuales fue desarrollado, se validan las entradas y las salidas de información, con el fin de establecer que el resultado es el esperado.

3.4.2 Pruebas de 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.

3.4.3 Pruebas de Usuario (UAT)

Estas pruebas buscan identificar la brecha que existen entre el entendimiento técnico y funcional del software y el conocimiento y experiencia desde el negocio y permiten la certificación del software por parte del cliente.

3.5 PRUEBAS AUTOMATIZADAS

3.5.1 Pruebas de Desempeño y Volumen

Esta prueba somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software y Permite validar si la aplicación cumple los criterios de tiempos de respuesta establecidos.

Este tipo de prueba es fundamental en una aplicación ya que, si ésta no responde en el debido tiempo, se pueden perder clientes, o dañar la imagen ante los usuarios.

3.5.2 Pruebas de Stress

Valida aquellos volúmenes de datos máximos que resiste el sistema antes de comenzar con errores.



3.5.3 Prueba de concurrencia

Valida la capacidad del sistema de atender múltiples solicitudes de parte de los usuarios que acceden a un mismo recurso.

3.5.4 Prueba de Integridad de las bases de datos

Verifica el cumplimiento de las políticas de seguridad acordadas para el sistema.

3.5.5 Pruebas de Regresión

Se pueden hacer en una etapa posterior del software para identificar alteraciones en las funcionalidades por cambios realizados después de la liberación parcial o total del sistema. (Este tipo de prueba es opcional y por eso se recomienda que sea automatizada para que garantice la ejecución de rutinas sobre funcionalidades ya estables y que se encuentren en producción)



4 CRITERIOS DE ACEPTACIÓN Y/O RECHAZO DE “ELEMENTOS” DE PRUEBA

Los criterios de aceptación y/o rechazo, se deben definir en términos de indicador que permitan la calificación de la calidad del software y el nivel de tolerancia a fallos. De tal manera que es necesario definir unos indicadores que puedan ser medibles y verificables para cada uno de los componentes individuales del software, pero también en su globalidad.

Para esta etapa se proponen los indicadores que se listan en la Tabla 4, orientados a identificar y medir los fallos desde el punto de vista del impacto en la operatividad del software (Fallos Críticos, Fallos Mayores, Fallos Medios y Fallos Cosméticos).

NOMBRE INDICADOR	MEDICIÓN	UMBRAL DE TOLERANCIA
Fallos Críticos	Cantidad de errores donde los usuarios no pueden utilizar las funcionalidades principales del sistema.	Aceptación Mínima del 85%, que se calcula aplicando la fórmula: (Cantidad de casos de Uso con defectos críticos / Cantidad de Casos de Uso implementados en la solución) * 100.
Fallos Mayores	Cantidad de errores donde el sistema opera con restricciones que impiden completar la operación de negocio que define el requerimiento.	(Cantidad de casos de Uso con defectos mayores / Cantidad de Casos de Uso implementados en la solución) * 100.
Fallos Medios	Cantidad de errores donde no se encuentran disponibles algunas funciones o componentes del sistema, que generan un impacto mínimo para los usuarios del sistema.	Aceptación Mínima del 85%, que se calcula aplicando la fórmula: (Cantidad de casos de Uso con defectos menores / Cantidad de Casos de Uso implementados en la solución) * 100
Fallos Cosméticos	Cantidad de errores de la interfaz de usuario, que no impiden la correcta ejecución del sistema	Aceptación Mínima del 85%, que se calcula aplicando la fórmula: (Cantidad de casos de Uso con defectos cosméticos / Cantidad de Casos de Uso implementados en la solución) * 100

Tabla 4. Criterios de Aceptación

Fuente: UT Fonvivienda 2019, 2020



5 CRITERIOS DE SUSPENSIÓN Y CONDICIONES DE REANUDACIÓN PRUEBA.

Normalmente los criterios de suspensión de las pruebas se basan en la identificación de riesgos que no han sido mitigados, que no fueron identificados o que se materializaron.

A continuación, se relacionan a manera de ejemplo, algunas situaciones que pueden obligar la suspensión de las pruebas de forma parcial o total y que pueden hacer parte de la matriz de riesgos del proyecto en la cual se deben incluir los correspondientes a las pruebas, para poder establecer los planes de mitigación y tenerlos controlados:

- No se cuenta con el Software disponible en las fechas establecidas para el inicio de la etapa de pruebas. La condición para reanudación obedecerá a que el equipo de pruebas cuente con las funcionalidades dispuestas en el ambiente de pruebas establecido.
- Las funcionalidades se encuentran incompletas impidiendo poder ejecutar un flujo básico de información. La condición para reanudación obedecerá a poder contar con las funcionalidades y la evidencia de las pruebas unitarias por parte del equipo de desarrollo.
- Existe un fallo crítico que impide ejecutar la funcionalidad y bloquea el proceso de pruebas. La condición para reanudación obedecerá a la resolución de fallo.
- Existen cambios en la definición del alcance en las pruebas. La condición para reanudar obedecerá al cierre del alcance de las pruebas.



6 ENTREGABLES DE PRUEBA

La Etapa de definición de los entregables de pruebas es muy importante y debe quedar plenamente establecido el proceso de identificación, documentación, y gestión de los errores del software, para tal fin es necesario generar los siguientes entregables

Plan de prueba: El plan de pruebas debe estar ajustado a la realidad del momento en el que se inicia la etapa de pruebas (Proceso de construcción del producto de software), y debe considerar las decisiones tomadas al respecto de los numerales 1 al 4, ya que en este momento no es posible determinar, pero como se indicó en el documento se puede recomendar, es de aclarar que este documento debe ser actualizado y completado en la etapa de construcción del software y es allí donde se tomarán las decisiones que correspondan de acuerdo con las características del proyecto en ese momento.

- **Escenarios de prueba:** en el proceso de testeo del software y la verificación de la calidad del mismo, los casos de prueba deben estar enmarcados en evaluación y verificación de los puntos de control de las especificaciones funcionales. (Es necesario contar con una herramienta que garantice el seguimiento y control de los errores encontrados y la gestión para su resolución).

Es importante destacar, que la identificación de los escenarios de prueba debe garantizar el cubrimiento de las funcionalidades desde el punto de vista de negocio y de los requerimientos identificados por el cliente.

Desde esa perspectiva, se proponen los escenarios que se detallan de la Tabla 5 a la Tabla 8.

TIPO DE USUARIOS DEL SISTEMA		MACROPROCESO		GESTIÓN DE PROYECTOS					
		PROCESO		Elegibilidad de Proyectos	Incumplimientos	Oferentes	Postventa	Promoción y Acompañamiento	Supervisión y Seguimiento Proyectos
ESCENARIOS DE PRUEBAS DESDE EL PUNTO DE VISTA DE LAS FUNCIONES DE NEGOCIO	ADMINISTRACIÓN DEL SISTEMA	Configuración y del parametrización sistema	CREAR	✓	✓	✓	✓	✓	✓
			CONSULTAR	✓	✓	✓	✓	✓	✓
			MODIFICAR	✓	✓	✓	✓	✓	✓
			ELIMINAR	✓	✓	✓	✓	✓	✓
			ADMINISTRAR NOTIFICACIONES	✓	✓	✓	✓	✓	✓
			ADMINISTRAR ALERTAS	✓	✓	✓	✓	✓	✓
			ADMINISTRAR TAREAS	✓	✓	✓	✓	✓	✓
			ADMINISTRAR CARGA DE ARCHIVOS	✓	✓	✓	✓	✓	✓
	FUNCIONAL		CREAR	✓	✓	✓	✓	✓	✓



TIPO DE USUARIOS DEL SISTEMA		MACROPROCESO		GESTIÓN DE PROYECTOS					
		PROCESO		Elegibilidad de Proyectos	Incumplimientos	Oferentes	Postventa	Promoción y Acompañamiento	Supervisión y Seguimiento Proyectos
		Uso y gestión del sistema, desde el punto de vista de usuario final	CONSULTAR	✓	✓	✓	✓	✓	✓
			MODIFICAR	✓	✓	✓	✓	✓	✓
			ELIMINAR	✓	✓	✓	✓	✓	✓
ESCENARIOS DE PRUEBAS DESDE EL PUNTO DE VISTA DE LAS FUNCIONES DE NEGOCIO	ADMINISTRADOR FUNCIONAL	Gestión de las características administrativas que se requieren por área funcional, así como las configuraciones de las reglas de negocio	GESTIONAR ARCHIVOS	✓	✓	✓	✓	✓	✓
			GESTIONAR RESOLUCIÓN	✓	✓	✓	✓	✓	✓
			GESTIONAR CORREOS	✓	✓	✓	✓	✓	✓
			GESTIONAR DASHBOARD	✓	✓	✓	✓	✓	✓
			GESTIONAR INFORMES Y REPORTES	✓	✓	✓	✓	✓	✓
			GESTIONAR ESTADÍSTICAS	✓	✓	✓	✓	✓	✓
			CONFIGURAR REGLAS	✓	✓	✓	✓	✓	✓

Tabla 5. Escenarios de prueba 1

Fuente: UT Fonvivienda 2019, 2020

TIPO DE USUARIOS DEL SISTEMA		MACROPROCESO		GESTIÓN SUBSIDIO											
		PROCESO		Asignación Subsidio	Autorización abono CAP	Autorización Apertura Cuenta	Autorización Pago 20%	Distribución Recursos SEM	Focalización SFVR	Legalización y transferencia PVG	Movilización SFV	Procedimiento pago SFV MCV	Procedimiento Sancionatorio Rev	Proceso Fiduciario	
ESCENARIOS DE PRUEBAS DESDE EL PUNTO DE VISTA DE LAS FUNCIONES DE NEGOCIO	ADMINISTRACIÓN DEL SISTEMA	Configuración y parametrización del sistema	CREAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
			CONSULTAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
			MODIFICAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			ELIMINAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			ADMINISTRAR NOTIFICACIONES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			ADMINISTRAR ALERTAS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			ADMINISTRAR TAREAS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	FUNCIONAL	Uso y gestión del sistema, desde el punto de vista de usuario final	CREAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
			CONSULTAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
			MODIFICAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
			ELIMINAR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	ADMINISTRADOR FUNCIONAL	Gestión de las características	GESTIONAR ARCHIVOS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		



TIPO DE USUARIOS DEL SISTEMA		MACROPROCESO		GESTIÓN SUBSIDIO										
		PROCESO		Asignación Subsidio	Autorización abono CAP	Autorización Apertura Cuenta	Autorización Pago 20%	Distribución Recursos SEM	Focalización SFVR	Legalización y transferencia PVG	Movilización SFV	Procedimiento pago SFV MCV	Procedimiento Sancionatorio Rev	Proceso Fiduciario
		administrativas que se requieren por área funcional, así como las configuraciones de las reglas de negocio	GESTIONAR RESOLUCIÓN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			GESTIONAR CORREOS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			GESTIONAR DASHBOARD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ESCENARIOS DE PRUEBAS DESDE EL PUNTO DE VISTA DE LAS FUNCIONES DE NEGOCIO	ADMINISTRADOR FUNCIONAL	Gestión de las características administrativas que se requieren por área funcional, así como las configuraciones de las reglas de negocio	GESTIONAR INFORMES Y REPORTES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			GESTIONAR ESTADÍSTICAS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			CONFIGURAR REGLAS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabla 6. Escenarios de prueba 2

Fuente: UT Fonvivienda 2019, 2020

TIPO DE USUARIOS DEL SISTEMA		MACROPROCESO		TITULACIÓN Y SANEAMIENTO PREDIAL					
		PROCESO		Cesión Bienes Vocación uso Publico	Cesión título gratuito	Enajena bienes instituciones religiosas	Identificación bienes inmuebles	Promoción y Acompañamiento titulación	Transferencia dominio bienes inmuebles
ESCENARIOS DE PRUEBAS DESDE EL PUNTO DE VISTA DE LAS FUNCIONES DE NEGOCIO	ADMINISTRACIÓN DEL SISTEMA	Configuración y parametrización del sistema	CREAR	✓	✓	✓	✓	✓	✓
			CONSULTAR	✓	✓	✓	✓	✓	✓
			MODIFICAR	✓	✓	✓	✓	✓	✓
			ELIMINAR	✓	✓	✓	✓	✓	✓
			ADMINISTRAR NOTIFICACIONES	✓	✓	✓	✓	✓	✓
			ADMINISTRAR ALERTAS	✓	✓	✓	✓	✓	✓
			ADMINISTRAR TAREAS	✓	✓	✓	✓	✓	✓
			ADMINISTRAR CARGA DE ARCHIVOS	✓	✓	✓	✓	✓	✓
	FUNCIONAL	Uso y gestión del sistema, desde el punto de vista de usuario final	CREAR	✓	✓	✓	✓	✓	✓
			CONSULTAR	✓	✓	✓	✓	✓	✓
			MODIFICAR	✓	✓	✓	✓	✓	✓
			ELIMINAR	✓	✓	✓	✓	✓	✓
	ADMINISTRADOR FUNCIONAL	Gestión de las características administrativas que se requieren por área funcional, así como	GESTIONAR ARCHIVOS	✓	✓	✓	✓	✓	✓
			GESTIONAR RESOLUCIÓN	✓	✓	✓	✓	✓	✓
			GESTIONAR CORREOS	✓	✓	✓	✓	✓	✓
GESTIONAR DASHBOARD			✓	✓	✓	✓	✓	✓	
GESTIONAR INFORMES Y REPORTES			✓	✓	✓	✓	✓	✓	



TIPO DE USUARIOS DEL SISTEMA	MACROPROCESO		TITULACIÓN Y SANEAMIENTO PREDIAL					
	PROCESO		Cesión Bienes Vocación uso Público	Cesión título gratuito	Enajena bienes instituciones religiosas	Identificación bienes inmuebles	Promoción y Acompañamiento titulación	Transferencia dominio bienes inmuebles
	las configuraciones de las reglas de negocio	GESTIONAR parámetros	✓	✓	✓	✓	✓	✓
		CONFIGURAR REGLAS	✓	✓	✓	✓	✓	✓

Tabla 7. Escenarios de prueba 3

Fuente: UT Fonvivienda 2019, 2020

TIPO DE USUARIOS DEL SISTEMA	MACROPROCESO		LICENCIAS URBANÍSTICAS	
	PROCESO		Licencias Urbanísticas	
ESCENARIOS DE PRUEBAS DESDE EL PUNTO DE VISTA DE LAS FUNCIONES DE NEGOCIO	ADMINISTRACIÓN DEL SISTEMA	Configuración y parametrización del sistema	CREAR	✓
			CONSULTAR	✓
			MODIFICAR	✓
			ELIMINAR	✓
			ADMINISTRAR NOTIFICACIONES	✓
			ADMINISTRAR ALERTAS	✓
			ADMINISTRAR TAREAS	✓
	FUNCIONAL	Uso y gestión del sistema, desde el punto de vista de usuario final	ADMINISTRAR CARGA DE ARCHIVOS	✓
			CREAR	✓
			CONSULTAR	✓
	ADMINISTRADOR FUNCIONAL	Gestión de las características administrativas que se requieren por área funcional, así como las configuraciones de las reglas de negocio	MODIFICAR	✓
			ELIMINAR	✓
			GESTIONAR ARCHIVOS	✓
			GESTIONAR RESOLUCIÓN	✓
			GESTIONAR CORREOS	✓
		GESTIONAR DASHBOARD	✓	
		GESTIONAR INFORMES Y REPORTES	✓	
		GESTIONAR ESTADÍSTICAS	✓	
		CONFIGURAR REGLAS	✓	

Tabla 8. Escenarios de prueba 4

Fuente: UT Fonvivienda 2019, 2020

- **Evidencia Casos de prueba:** Durante la etapa de construcción del software, para cada uno de los diseños de los escenarios de prueba propuestos, se debe establecer la



forma en la que se recoge la evidencia (Videos, imágenes, y archivos de documentación de las pruebas), en cualquiera de las formas establecidas para generar las evidencias, las herramientas a ser utilizadas juegan un papel muy importante, puesto que facilitarán la labor de pruebas y permitirán que el avance en la ejecución sea más o menos (anteriormente se recomendó contar con herramientas robustas que permitan llevar toda la trazabilidad y eviten trabajos en exceso que recortan los tiempos de ejecución).

- **Reportes de defectos:** La gestión de los fallos o defectos en el software es muy relevante en la etapa de pruebas, permite tener el control de la información y realizar una gestión adecuada, se recomienda la generación de reportes con indicadores de avance en la resolución de los defectos, por módulo funcional y por caso de uso, así como por equipo técnico a cargo (Desarrollo o Pruebas).



7 NECESIDADES DE PERSONAL

7.1 DEFINICIÓN DEL EQUIPO DE TRABAJO REQUERIDO

Gran parte del éxito en la producción de software de calidad se centra en el equipo de pruebas, su experiencia, su conocimiento y su habilidad para diseñar, analizar y probar el software.

Los roles principales a considerar son:

- Líder de Pruebas
- Analista de pruebas
- Tester

Es muy común que con un mismo perfil se puedan cubrir varios de estos roles; sin embargo, es importante señalar que el rol de Líder de pruebas es fundamental y debe estar dedicado solo a la gestión de las pruebas

7.2 DISTRIBUCIÓN DEL TRABAJO

7.2.1 Definición de roles y responsabilidades (Matriz RACI)

FUNCIONES	DESARROLLADOR	LÍDER DE PRUEBAS	ANALISTA	TESTER
Realizar las pruebas Unitarias	R	I	I	I
Realizar las revisiones de código conforme con los estándares	R	I	I	I
Realizar el ajuste de los errores reportados por el equipo de pruebas	R	CI	I	I
Capacitar y velar por que el equipo tenga un entendimiento de producto	I	R	I	I
Identificar Riesgos en el proceso de ejecución y escalar cuando sea necesario	I	R	I	I
Gestionar la asignación de tareas dentro del equipo de pruebas	I	RA	I	I
Administra las herramientas de gestión de Pruebas	I	RA	I	I
Delega las funciones a los demás integrantes del equipo	I	RA	CI	I
Valida constantemente el avance del proceso	I	RA	I	I



FUNCIONES	DESARROLLADOR	LÍDER DE PRUEBAS	ANALISTA	TESTER
Asegura la calidad de los entregables de pruebas	I	RA	R	R
Asegura el cumplimiento de los indicadores definidos para el proceso de pruebas	I	R	R	R
Segura que los requerimientos funcionales definidos están plenamente identificados en los escenarios de pruebas	I	A	R	R
Liderar el análisis desde el punto de vista funcional y garantizar que los entregables de desarrollo estén bien definidos y en condiciones óptimas para ser atendidos por el equipo de pruebas	I	AR	R	I
Participar activamente en el proceso de desarrollo desde el punto de vista del entendimiento funcional para identificar de forma temprana riesgos y ajustes necesarios al software	I	R	I	I
Diseñar los escenarios y casos de prueba que garanticen la calidad del software y el cumplimiento de los requisitos	I	AR	R	I
Probar cada uno de los casos de prueba diseñados y ajustarlo cuando se requiera	I	CI	R	R
Documentar adecuadamente conforme con los estándares definidos los casos fallidos del software, así como como los exitosos	I	CI	R	R

Tabla 9. Matriz RACI

Fuente: UT Fonvivienda 2019, 2020

7.2.2 Definición técnica del equipo de pruebas

El equipo de Pruebas debe contar con las siguientes características:

- Líder de Pruebas:
 - Tener conocimientos técnicos de la disciplina de pruebas
 - Conocer las herramientas de pruebas dispuestas para el proyecto
 - Habilidades de gestión para hacer un buen seguimiento y control de los defectos
 - Habilidades de comunicación con el equipo de desarrollo y equipo funcional para poder gestionar adecuadamente la resolución de los defectos



- Conocimientos en las metodologías de estimación, para evaluar el impacto y el esfuerzo de la etapa de pruebas, así como poder hacer la asignación efectiva del trabajo.
- Identificar las necesidades que el equipo tenga de capacitación a nivel del negocio o a nivel técnico y metodológico.
- Analista de pruebas
 - Capacidad de análisis y comprensión del negocio para poder identificar fallos en el software.
 - Habilidad para concentrarse en los detalles.
 - Capacidad de comunicación tanto escrita como hablada para poder transmitir la información suficiente y necesaria que permita entender el fallo identificado.
 - Capacidad de trabajo bajo presión.



8 PROGRAMA DE ACTIVIDADES DE PRUEBA

El programa o planificación del cronograma de pruebas, debe estar enmarcado en la planificación global del proyecto y debe tener fechas y tiempos establecidos con claridad, conforme con la evolución de las etapas anteriores.

Es posible, que como estrategia se inicien tareas de pruebas tempranas, que esté orientadas a identificar inconsistencias de integración de los componentes y que pueden ayudar a optimizar la salida de funcionalidades con mayores índices de calidad desde la etapa de desarrollo.

Se puede recomendar iniciar con pruebas de escritorio sobre los casos de uso y sobre los mockups para establecer los posibles eventos a considerar durante el proceso de pruebas formales y puntos de control que el software debe haber tenido en cuenta (esta actividad es opcional y es a criterio del equipo de proyecto y la gerencia).

9 ALINEAMIENTO A METODOLOGÍAS ÁGILES

9.1 AUTOMATIZACIÓN DE PRUEBAS

La automatización de pruebas se origina en la necesidad de las compañías y de los equipos de desarrollo de software de facilitar la labor de las pruebas y reducir los tiempos de ejecución; esto no es funcional en todos los casos, ni en todo el proceso de pruebas, por tal razón es importante tener claridad en el contexto de los proyectos para que la decisión de automatización corresponda con la realidad del proyecto y efectivamente sea una oportunidad y no se convierta en un problema.

La automatización de las pruebas de software *“persigue el objetivo de simplificar el trabajo dispendioso, repetitivo o complejo, haciéndolo efectivo y más productivo”* (Abstracta, n.d.)

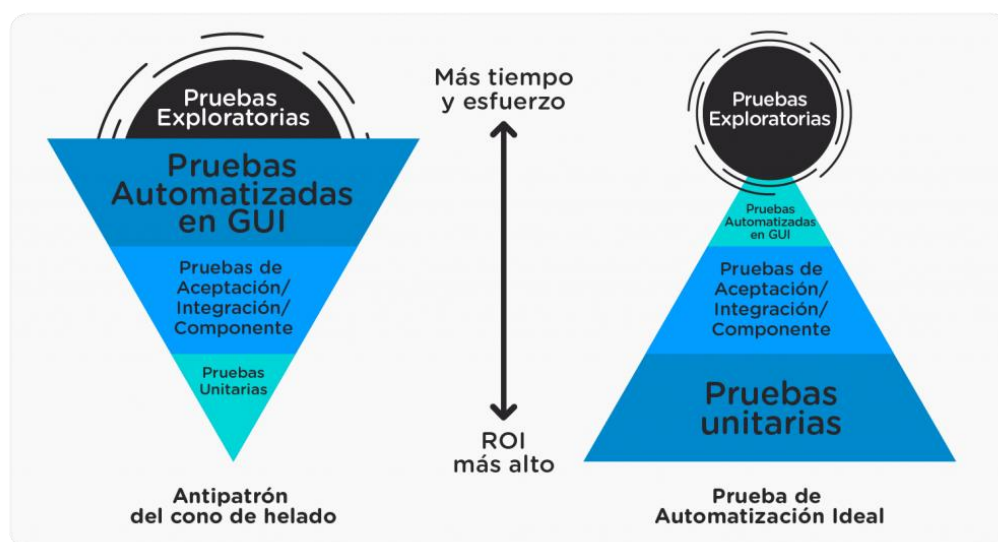


Ilustración 2. Pirámide Automatización de Pruebas

Fuente: Abstracta, Chile

Para tomar la decisión de automatización de las pruebas se deben considerar las características del proyecto y el costo de maduración del producto, pues los beneficios más altos se obtienen cuando el conocimiento del sistema es suficiente y profundo, de tal manera que se puedan identificar las rutinas que se pueden automatizar, tanto a nivel funcional como No funcional.

La automatización de las pruebas implica un cambio cultural y organizacional, por lo que es importante que, si el MVCT decide optar por metodologías ágiles, fortalezca sus procesos de comunicación y colaboración con el fin de reducir al máximo los reprocesos.



En cada una de las iteraciones que se pueden establecer en una metodología ágil, se pueden crear o ir adicionando completitud a las pruebas definidas para automatizar, teniendo en cuenta el esfuerzo que sea necesario. También se debe considerar que el desarrollo de un sistema desde cero como el SISFV, se debería planear el esfuerzo de la creación y mantenimiento de estas pruebas y ver el beneficio a largo plazo.

9.2 AMBIENTES DE PRUEBAS CON DEVOPS

Dentro de la cultura Devops, los equipos de trabajo están preparados para, construir, probar y desplegar productos mínimos viables en cortos periodos de tiempos, pero esto solo es posible si se acompaña de procesos y herramientas colaborativas y de integración continua.

Para cada una de las etapas del ciclo de desarrollo es necesario contar con herramientas que permitan integrar el proceso.



Ilustración 3. Proceso de Integración continua (Ciclo de Vida del Software)

Fuente: Pragma

De esta forma se pueden mitigar los riesgos antes de pasar a las siguientes etapas del ciclo de vida de desarrollo de software.

En la imagen se muestra las pruebas aplicables al proceso de integración continua donde se puede ver que se deben planear, dentro de los ciclos o iteraciones de una metodología ágil, los puntos correctos para incluir los diferentes tipos de pruebas.

Lo anterior quiere decir que, dentro de las definiciones propias de DevOps, los diferentes equipos, tendrán que estar sincronizados y estos tiempos tenerlos estimados y planeados.

9.3 ENFOQUE EJEMPLO TDD

TDD (Test-Driven Development), desarrollo dirigido por pruebas, es una metodología de desarrollo, que es tendencia actual, en la cual se hace el diseño de pruebas antes de hacer la codificación (aplica especialmente para las pruebas unitarias que son las que hace el



desarrollador), esta técnica le permite poder identificar con anterioridad a la implementación los puntos de control a ser considerados durante la implementación, facilitando al desarrollador el entendimiento del proceso.



Ilustración 4. Enfoque TDD

Dentro de las definiciones de la metodología de desarrollo, utilizar una metodología ágil como TDD, es muy útil si la implementación y en general el proceso de construcción del software se orientan a la entrega de valor constante, detección temprana de errores y enfoque de validación funcional o de requerimientos.

9.4 CICLOS DE PRUEBAS

El enfoque de pruebas en metodologías ágiles, implica hacer ciclos cortos pero que se repiten muchas veces y están orientados a producir software de alta calidad por medio de la construcción de componentes o funciones pequeñas, de forma incremental.

En términos generales se deberían cumplir los siguientes pasos:

- Selección de un requisito.
- Diseñar los casos de prueba orientados al fallo.
- Construcción de la funcionalidad mínima.
- Ejecución de todos los casos de prueba.
- Refactorización.
- Actualización de la lista de requisitos.

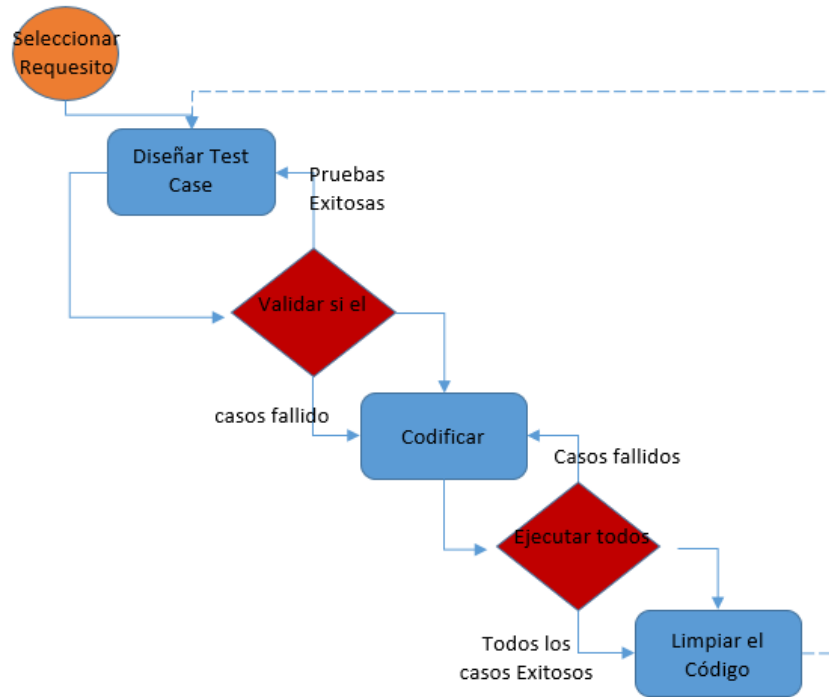


Ilustración 5.Ciclo de Pruebas

Fuente: Paradigma Digital

De la misma forma como se planean los sprint o iteraciones, está en la responsabilidad del marco de la administración del proyecto, incluir las pruebas dependiendo de la metodología de desarrollo y poder cumplir con las definiciones del aseguramiento de calidad.

9.5 VENTAJAS: DISMINUCIÓN DE RIESGOS

Las principales ventajas que se pueden desprender de implementar proyectos con metodologías ágiles y características de DevOps en relación con las pruebas son:

- Promover la comunicación y la colaboración entre el aseguramiento de calidad (pruebas de software), el desarrollo y los diferentes ambientes.
- Ayudar a los proyectos de desarrollo de software a producir con mayor calidad, dando valor constante, con disminución de esfuerzo y tiempo, con lo cual se puede lograr eficiencia en los costos.
- Integración y entrega continua por medio del uso de herramientas de gestión, permitiendo la delimitación y planeación controlada de las pruebas.
- Permitir flexibilidad en la definición de los requerimientos y como resultado facilitar



cambios (cambios en el negocio) de forma viable y con menor impacto.

- Facilita la preparación de entornos de desarrollo y pruebas con características similares a (WEB, n.d.) los entornos productivos.



10 BIBLIOGRAFÍA

Abstracta. (s.f.). *¿Porque Automatizar Pruebas de Software?*

AENOR, G. D. (2018). *ISO/IEC/IEEE 29119 - EL NUEVO ESTANDAR INTERNACIONAL DE PRUEBAS*. MADRID.

EAFIT, U. (2010). *Metodología para testing de software basado en componentes*. Obtenido de <https://core.ac.uk/download/pdf/47237302.pdf>

ECURED. (2018). Obtenido de https://www.ecured.cu/Estrategia_de_prueba_de_software

IDG, C. F. (s.f.). *Gartner avisa de los riesgos de implantar DevOps*. Obtenido de <https://www.ciospain.es/gobierno-ti/gartner-avisa-de-los-riesgos-de-implantar-devops>

IEEE. (1983). *Documentation., IEEE Standard for Software Test*. ANSI/IEEE Std 829-1983,.

IEEE. (1983,). *IEEE Standard Glossary of Software Engineering Terminology*. ANSI/IEEE Std 729.

ISO. (2013). ISO / IEC 29119-1: Conceptos y definiciones.

ISO. (Septiembre de 2013). *ISO / IEC 29119-2: Procesos de prueba*.

ISO. (2014). ISO / IEC 29119-4: Técnicas de prueba.

ITCA FEPAE - ESCUELA ESPECIALIZADA EN INGENIERIA. (2016). *ITCA FEPAE*. Obtenido de ITCA FEPAE:

https://virtual.itca.edu.sv/Mediadores/stis/42estrategias_de_prueba_para_software.html

UNIVERSIDAD POLITECNICA DE MADRID. (Junio de 2015). *Archivo Digital UPM*. Obtenido de http://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf

WEB, R. C. (s.f.). *DevOps: la mejor manera de ganar tiempo y evitar riesgos*. Obtenido de <https://www.relacioncliente.es/devops-ganar-tiempo-evitar-riesgos/>