

Pruebas de software

Generalidades



1 • FUNDAMENTOS DE LAS PRUEBAS

CAPITULO 1 • FUNDAMENTOS DE PRUEBAS DE SOFTWARE

Existen **2 errores comunes sobre las pruebas** de software:

1. Sólo consisten en ejecutar el software y comprobar resultados.
2. Se enfocan en validar que el software cumpla con la verificación de requerimientos.



**Probar no es sólo
correr pruebas.**

Verificación:

¿Construimos bien el
software según los
requerimientos?

Validación:

¿Construimos el software
adecuado basado en las
necesidades reales del
cliente?

PROCESO DE PRUEBAS



La definición de Pruebas de Software la dividimos en 2 partes: Proceso y Objetivos.

El Proceso

- Probar es un proceso no una actividad, se realiza durante todo el Ciclo de vida de desarrollo, abarcando las pruebas Estáticas y Dinámicas.
- Las pruebas se deben preparar. Se planifica antes, durante y después de la ejecución de las Pruebas.
- Los resultados de las pruebas del Código y los Productos de Trabajo relacionados deben ser verificados.

Objetivos

- Se **verifica** y se **valida** el cumplimiento de los requisitos.
- Al Detectar defectos comprendemos riesgos, mejoramos la calidad de los productos.
- Y cuando analizamos la **causa raíz** mejoramos los Procesos de Desarrollo

Alguna forma de prueba implica la ejecución del componente o sistema que se está probando; esta prueba se denomina prueba dinámica. Otras formas de prueba no implican la ejecución del componente o sistema que se está probando; estas formas de pruebas se denominan pruebas estáticas.

OBJETIVOS DE LAS PRUEBAS

- Evaluar que todos los productos de trabajo estén correctos
- Verificar que todos los requerimientos especificados han sido satisfechos
- Validar que todos los requerimientos especificados son correctos, incluidos los de ámbito legal
- Generar confianza en el software
- Prevenir defectos
- Encontrar fallas y defectos
- Reducir el riesgo de fallas
- Proporcionar información para la toma de decisiones
- Reducir el nivel de riesgo de software de baja calidad
- Cumplir con estándares legales o comerciales



Los objetivos de la prueba varían dependiendo del sistema que se esté probando, en qué fase del ciclo de desarrollo se está, o del nivel de prueba.

Producto de trabajo

Un producto de trabajo es cualquier documento, informe, reporte, diagrama e inclusive el código fuente o el sistema terminado.



Probar es distinto de depurar

Debugging o Depuración es el proceso de desarrollo que encuentra, analiza y repara el defecto.

Probar es verificar que el software funciona como se espera.

El ciclo de depuración implica que el desarrollador depura el código y realiza sus pruebas.

Luego del ciclo de depuración, un probador independiente ejecuta **las pruebas de confirmación**.

Las pruebas de verificación se realizan para garantizar que se cumpla lo establecido en los requerimientos.

¿POR QUÉ ES NECESARIO PROBAR EL SOFTWARE?

1. Todo el mundo comete errores, los errores introducen defectos que pueden generar fallas.
2. Reduce el riesgo y aumenta la confianza.
3. Algunos defectos son difíciles de identificar debido a que se originan en suposiciones erradas o puntos ciegos.
4. Algunas pruebas son obligatorias debido a estándares legales o comerciales.

¿ Cómo contribuye la prueba al éxito de un proyecto?

1. Cuando los probadores y diseñadores trabajan en equipo se puede aumentar la comprensión del diseño del sistema y cómo probarlo.
2. Cuando los probadores y desarrolladores trabajan en equipo mientras el código se está desarrollando, le permite al probador ubicar segmentos de la aplicación susceptibles a error.

3. Tener probadores verificando y validando el software antes de su liberación aumenta la probabilidad de que el software cumpla con las expectativas del usuario.
4. Que el software cumpla con las expectativas del usuario aumenta la confianza en el producto, y la confianza en el producto genera fidelidad al proveedor.



***Aseguramiento de la Calidad
y Pruebas son conceptos
distintos***

El Aseguramiento de la Calidad se centra en el cumplimiento de los procesos adecuados para generar productos de trabajo de mayor calidad.

El Control de la Calidad implica varias actividades, incluidas las actividades de prueba, lo que apoya el logro de niveles apropiados de calidad.

La Gestión de Calidad une a ambos conceptos: Aseguramiento y el Control de la calidad.

¿ Por qué ocurren errores?

- Trabajo bajo presión
- Falibilidad humana
- Falta de experiencia del personal
- Complejidad del proyecto
- Falta de comunicación

Causa Raíz de un defecto

Cuando se detecta una falla, debemos rastrearla para determinar la razón por la cual ocurrió. Los defectos deben ser analizados para identificar la raíz de su origen, y así reducir la posibilidad de que se repita en el futuro.

ERROR, DEFECTO Y FALLA



¡Recuerda no confundir conceptos!

Error: Error, Mistake

Defecto: Falta, Defect, Bug, fault, fail.

Falla: Failure.



Un humano comete un **ERROR** que puede introducir un **DEFECTO** que podría generar una **FALLA**

Oráculo de prueba es la fuente para determinar resultados esperados para compararlos con los resultados reales del software en pruebas.

- No todos los defectos causarán fallas.
- No todas las fallas son de origen humano, pueden ser de origen ambiental.
- No todos los resultados inesperados de las pruebas son fallas.

Existen **falsos positivos** y **falsos negativos** en los resultados de las pruebas.

Si la prueba dice que hay un error cuando en realidad no lo hay, es un **Falso positivo**.

Si la prueba no detecta defectos que debió haber identificado, es un **Falso negativo**.

SIETE PRINCIPIOS DE LAS PRUEBAS

1. Las pruebas demuestran que hay errores, no su ausencia
2. Es imposible probarlo todo
3. Probar desde el inicio del ciclo ahorra tiempo, esfuerzo y dinero
4. Los defectos se agrupan
5. Los casos de prueba deben actualizarse para encontrar nuevos defectos
6. Las pruebas dependen del sistema que se esté probando
7. Es imposible que no hayan errores



La creencia de que el software debe ser liberado sin errores y con absoluta correctitud es algo imposible de lograr. Sin embargo se puede reducir el riesgo de falla en ambientes operativos haciendo pruebas con estándares altos de calidad.

PROCESO DE PRUEBAS: Actividades



Planificación y Control: ¿Cuales son los objetivos y cómo se alcanzan?

En la **planificación** se debe asegurar la comprensión de las expectativas del usuario, cuáles son sus metas, objetivos y los riesgos que se deberán considerar. Se establecen las bases de prueba y el plan de pruebas a seguir. El **control o monitoreo** verifica que las actividades se están llevando a cabo según el plan y se realiza durante todo el proceso de pruebas.

Análisis: Se analizan los requerimientos levantados a fin de verificar entre otras cosas, factibilidad de la prueba, tiempo y recursos necesarios. Se responde a la pregunta: **¿Que probar?**

Diseño de los casos de prueba de alto nivel. Un caso de prueba es un grupo de valores, precondiciones y resultados esperados. Se debe responder **¿Cómo probar?**

Implementación: Aquí se crean o se completan los productos de prueba necesarios para la ejecución de la prueba.

A menudo implementación y ejecución son etapas combinadas, en esta etapa se debe preguntar **¿Está todo listo para ejecutar las pruebas?**

Ejecución: Se ejecutan los casos de prueba, se registran y reportan los resultados de la prueba.

Cierre o compleción de la prueba: Finalmente, se libera el software cuando los criterios de aprobación han sido satisfechos y se ejecutan las actividades de cierre del proyecto.

PROCESO DE PRUEBAS: Actividades

Planificación → Análisis → Diseño → Construcción → Pruebas → Cierre



Aunque parezca que el proceso de pruebas sigue una serie secuencial de actividades, algunas de ellas pueden llevarse a cabo iterativamente, como por ejemplo la implementación y la ejecución de las pruebas.



PROCESO DE PRUEBAS: Productos de Trabajo



PROCESO DE PRUEBAS: Productos de Trabajo



Condiciones de Prueba

Diseño de Pruebas

- Escenarios
- Suites de Prueba
- Data de Prueba

Insumo

Escenarios
Suites de Prueba
Data de Prueba

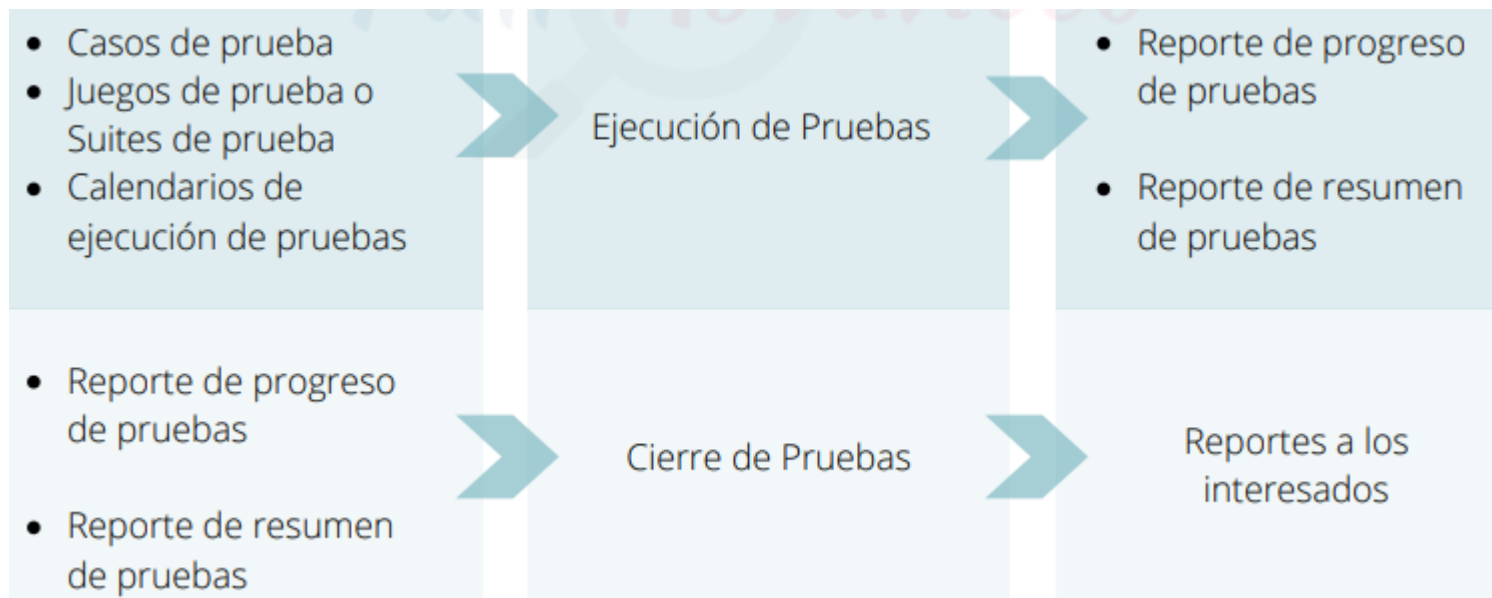
Actividad

Implementación de
Pruebas

Salida

- Procedimientos
- Casos de prueba
- Juegos de prueba
- Calendarios de ejecución de pruebas

Insumo	Actividad	Salida



Trazabilidad entre Bases de Prueba y Productos de Trabajo

Etapa	Entradas	Objetivos	Actividades	Salidas	Trazabilidad
Planificación de pruebas	Bases de Prueba	<p>Definir los objetivos de las pruebas</p> <p>Establecer la estrategia de pruebas: técnicas, tareas, fechas de entrega</p>	<p>Definir alcance, riesgos, enfoque y calendario de pruebas</p> <p>Establecer cómo medir el avance, quién hace qué y cómo lo hace</p> <p>Integrar las pruebas al ciclo de vida</p>	<p>Estrategia</p> <p>Plan de pruebas</p>	A partir del plan de pruebas se genera trazabilidad hacia la totalidad de los productos de trabajo, ya que esta es la piedra angular del proyecto de pruebas.

Etapa	Entradas	Objetivos	Actividades	Salidas	Trazabilidad
Monitoreo y Control de pruebas	<p>Cronograma de planificación</p> <p>Resultado de pruebas</p> <p>Reportes de análisis de riesgo</p>	<p>Verificar que el plan de pruebas cumpla con las fechas, estándares de calidad y demás especificaciones establecidas</p> <p>Tomar acciones correctivas en caso de desvíos en la planificación</p>	<p>Revisar plan de pruebas vs progreso real</p> <p>Revisar resultados de pruebas vs. Criterios de salida (definición de listo)</p> <p>Determinar si más pruebas son necesarias</p> <p>Comunicar resultados a las partes interesadas</p>	<p>Reporte de progreso de pruebas</p> <p>Reporte resumen de pruebas</p>	<p>Casos de pruebas Vs Calendario de pruebas</p> <p>Casos de pruebas Vs Riesgos identificados</p> <p>Casos de prueba vs Cobertura establecida</p>

Etapa	Entradas	Objetivos	Actividades	Salidas	Trazabilidad
Análisis de pruebas	Bases de Prueba	Evaluar las bases de prueba	Definir y priorizar las condiciones de prueba	Condiciones de prueba Niveles de riesgo	Bases de prueba vs. Condiciones de prueba
Diseño de pruebas	Condiciones de Prueba Estrategia de prueba Plan de prueba	Diseñar el ambiente Identificar data, infraestructura y herramientas necesarias	Definir y priorizar casos de prueba Identificar data, herramientas e infraestructura Diseñar el ambiente	Casos de prueba en alto nivel (Lógicos)	Condiciones de prueba vs. Casos de prueba de alto nivel

Implementación de pruebas	Casos de prueba	<p>Desarrollar el ambiente de pruebas</p> <p>Crear las suites de pruebas</p>	Desarrollar y priorizar procedimientos de prueba Test procedures	<p>Casos de prueba de bajo nivel (concretos)</p> <p>Escenarios de prueba</p> <p>Suites de prueba</p> <p>Cronograma de ejecución de pruebas</p>	<p>Procedimientos Escenarios</p> <p>Vs.</p> <p>Casos de prueba de bajo nivel</p>
Ejecución de pruebas	Suites/Juegos de prueba	Ejecutar las pruebas	Comparar resultados obtenidos con los esperados: Oráculo de prueba	<p>Reporte de resultado de pruebas</p> <p>Reporte de defectos encontrados</p>	<p>Defectos encontrados Vs. Cobertura establecida</p> <p>Defectos encontrados Vs. Riesgos identificados</p>
Cierre de pruebas	Reporte de Resultados	Recopilar datos de actividades realizadas	<p>Revisar que todos los defectos reportados hayan sido cerrados</p> <p>Almacenar los productos de trabajo que puedan ser reutilizables</p>	Reporte a los interesados Stakeholders	Pruebas realizadas Vs. Definición de hecho

El sesgo de confirmación es la dificultad para aceptar información que difiere con nuestras creencias actuales.

Es decir, es la tendencia a dar un mayor valor a las ideas u opiniones que confirman nuestras creencias.



Sesgo de confirmación

Es común que algunas personas pueden percibir el proceso de prueba como una actividad destructiva, por esta razón probadores y jefes de prueba deben saber comunicarse de forma asertiva los resultados de las pruebas y construir relaciones positivas con sus colegas.



Los desarrolladores y probadores tienen esquemas de pensamiento diferente



Desarrollador | Developer: Se enfoca en diseñar y construir un producto.



Probador | Tester: Se enfoca en verificar y validar el producto, encontrando tantos defectos como sea posible antes de la liberación a producción.

Pruebas independientes

Los probadores independientes son personas distintas al autor del producto de trabajo.

Hay niveles de independencia, de menor a mayor:

- Pruebas realizadas por la persona que creó el producto bajo prueba
- Pruebas realizadas por otra persona del mismo equipo
- Pruebas realizadas por una persona de un equipo diferente
- Pruebas realizadas por una persona de una empresa diferente

-
- Recordar que es una colaboración, no una batalla entre bandos
 - Enfatizar los beneficios de las pruebas
 - Comunicar los defectos encontrados de una manera neutral, sin criticar al creador del producto de trabajo defectuoso
 - Tratar de entender cómo se siente la otra persona
 - Confirmar que el interlocutor ha entendido lo que se ha dicho, y viceversa

1. Introducción

Introducción sobre la temática a trabajar, se deberá dar una introducción corta sobre que tiene el documento y una contextualización del proyecto a construir.

A diagram consisting of three purple circles with white numbers 1, 2, and 3 inside. Red arrows point from the text above to each circle: one from "temática a trabajar" to circle 1, one from "se deberá dar una introducción" to circle 2, and one from "una contextualización del proyecto a construir" to circle 3.

1.1 Propósito

Propósito del documento, para que se realizará, esto complementa lo que se mencionó en la introducción.

1.2 Alcance

Describir el alcance, mencionar los proyectos asociados y determinar que se ve afectado por este documento, hasta qué punto se llegará con el desarrollo del sistema, se pueden contemplar aspectos geográficos, si el sistema consta de varios sistemas incluirlo en el alcance.

1.3 Definiciones, Acrónimos y Abreviaturas

En este apartado se debe mostrar las definiciones de todos los términos, siglas y abreviaciones requeridas para entender este documento, será el glosario de este.

1.4 responsables e involucrados

En esta sección deben indicar el o los involucrados en el desarrollo del proyecto



Nombre	Tipo (<u>Responsable</u> / Involucrado)	Rol
<nombre>	<tipo>	<rol>



1.5 Referencias (bibliografía o web grafía)

En este apartado se debe mostrar las referencias bibliográficas o web grafía consultadas en el proceso de análisis del sistema, adicionalmente si las referencias fueron obtenidas gracias al proceso de levantamiento de información con el cliente, esto se debe mencionar previamente.

2. Descripción general

2.1 Perspectiva del producto

En este apartado se debe brindar una descripción general del sistema a construir, se debe dar la perspectiva del producto explicando en que consiste la herramienta, esta sección brinda una visión general del software como complemento a información que se brindará en secciones anteriores.

Ejemplo de perspectiva: El sistema SIS-I será un producto principalmente una aplicación para dispositivos móviles con el fin de llevar un registro en cada dispositivo que cuente con dicha aplicación de las rutas accesibles y disponibles en diferentes horarios de la semana. Contraría con alertas y/o notificación si el usuario así lo desea para recordar cierta ruta.

2.1.1 Interfaces del usuario

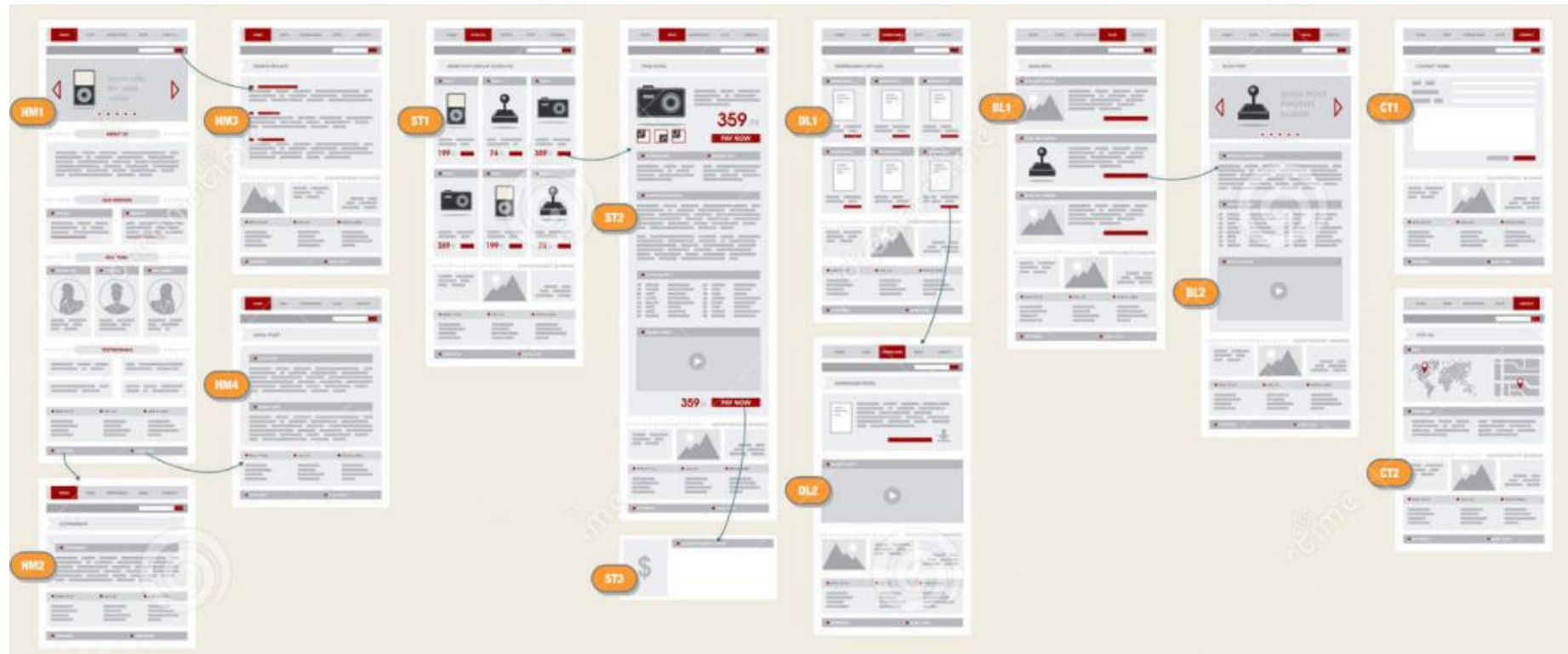
Descripción de cada una de las interfaces (no imágenes) se deberá definir en ítems el nombre de las interfaces de usuario a construir y una pequeña descripción de esta.

Ej.: Ventana login: Esta ventana permitirá al usuario iniciar sesión en el sistema mediante un rol definido.

interfaz al conjunto de elementos de la pantalla que permiten al usuario realizar acciones sobre el Sitio Web que está visitando.

2.1.2 Mapa de Navegación.

Se deberá construir un mapa de navegación donde se muestre de forma ordenada las interfaces de usuario descritas en la sección anterior, se mostrará una imagen tipo mapa conceptual.



4 2.2. Características del producto

Descripción de las principales características del sistema ej: el sistema será desarrollado en el lenguaje de programación xxxxxx, será una herramienta web, permitirá realizar procesos de gestión de usuarios, entre otras funcionalidades a nivel general, esto se realizará mediante ítems.

2.2.1 Funciones del producto

Descripción de cada uno de los módulos del sistema, un módulo puede considerarse como un conjunto de funcionalidades, se deberá reemplazar la palabra “módulo” por el nombre del módulo y debajo su descripción. ej:

2.2.1.1 Gestión de usuarios./

Este módulo permitirá realizar la gestión de usuarios mediante los procesos de registro, consulta, actualización y eliminación.

2.3 Características del usuario

Descripción de cada uno de los usuarios del sistema, un usuario del sistema será el tipo de rol con el que las personas pueden ingresar a la herramienta, se deberá reemplazar la palabra “tipo de usuario” por el nombre del usuario y debajo su descripción ej:

2.3.1 Administrador

Este usuario será el encargado de realizar los procesos de xxxxx y xxxxx dentro del sistema, este podrá tener acceso total al mismo etc etc etc

3. Especificación de requisitos

3.1 Requisitos funcionales

*Descripción de cada uno de los requisitos funcionales del sistema, estos requisitos giran en torno a lo que el sistema debe hacer (**funcionalidad**) se deberá indicar el código del requisito acompañado de su **nombre**, así como asignarle el tipo de requisito definiendo su prioridad. Ej: RF01 – Registro de usuario.*

FUNCIONALIDAD	TIPO (esencial, ideal, opcional)
RF01 -	
RF02 -	
RF0n -	

3.1.1 Clasificación de requisitos funcionales

Se deberá realizar un cuadro descriptivo por cada uno de los requisitos especificados anteriormente, en caso de que existan requisitos dependientes de otros, se podrá realizar solo un cuadro con el requisito principal y en la descripción se clasificarán los requisitos implícitos
Ej.: RF01 - Gestionar usuarios, este requisito es el principal, dentro de la descripción se explicará en que consiste y a su vez se debe explicar sus requisitos implícitos RF1.1 - Registrar Usuario, RF1.2 - Consultar usuario...



ID del requerimiento	<i>Colocar el ID del requerimiento funcional</i>
Nombre del requerimiento	<i>Colocar el nombre del requerimiento funcional</i>
Descripción	<i>Aquí se debe realizar una descripción del requerimiento funcional. Se debe colocar información suficiente de tal manera que sirva de ayuda para el desarrollador del sistema. Cualquier representación gráfica debe ser anexada en este apartado.</i>
Prioridad	<i>Alta / media / baja</i>

3.2 Requisitos Técnicos

Descripción de cada uno de los requisitos técnicos del sistema, estos requisitos giran en torno a los aspectos técnicos que se deben tener en cuenta para el correcto funcionamiento del sistema (no confundir con los aspectos técnicos donde se mencionan temas de capacidad en cuanto a hardware, memoria entre otros), hay que contemplar aspectos como acceso a internet, almacenamiento de información, versiones entre otros.

Se deberá indicar el código del requisito acompañado de su nombre, así como asignarle el tipo de requisito definiendo su prioridad. Ej: RT01 – Acceso a Internet.

FUNCIONALIDAD	TIPO (esencial, ideal, opcional)
RT01 -	
RT02 -	
RT0n -	

4 3.2.1 Clasificación de requisitos Técnicos.

Se deberá describir cada uno de los requisitos especificados anteriormente, en caso de que existan requisitos dependientes de otros, se podrá realizar solo un cuadro con el requisito principal y en la descripción se clasificarán los requisitos implícitos tal como en el punto anterior.

ID del requerimiento	<i>Colocar el ID del requerimiento Técnico</i>
Nombre del requerimiento	<i>Colocar el nombre del requerimiento Técnico</i>
Descripción	<i>Aquí se debe realizar una descripción del requerimiento Técnico. Se debe colocar información suficiente de tal manera que sirva de ayuda para el desarrollador del sistema. Cualquier representación gráfica debe ser anexada en este apartado.</i>
Prioridad	<i>Alta / media / <u>baja</u></i>

3.3 Requisitos no funcionales

Describa los requerimientos no funcionales para este documento. Los requerimientos no funcionales tienen que ver con las características que de una u otra forma puedan limitar el sistema como son: el rendimiento (en tiempo y espacio), confiabilidad, interfaces, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc. Se deberá indicar el código del requisito acompañado de su nombre ej:

Usabilidad

RNF01 – Aprendizaje del sistema: El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.

RNF02 – Manuales de usuario: El sistema debe contar con manuales de usuario estructurados adecuadamente dentro de la aplicación.



Código – Nombre	Descripción
Usabilidad <i>En este apartado se debe incluir la lista de todos los requerimientos que afecten la usabilidad. Esto debe incluir: el tiempo que se tomará un usuario en aprender a utilizar el sistema y se podría explicar por qué debe ser rápido el aprendizaje, los tiempos medibles de tarea para las tareas típicas y los requerimientos para concordar con estándares.</i>	
RNF01 -	
RNF02 -	

4. Aspectos legales (normas o leyes)

Este punto define las cuestiones legales que pudieran afectar esta entrega. No considerar cuidadosamente estas cuestiones puede poner a la organización de desarrollo en riesgo de una acción legal. Busque consejo profesional si es necesario.

5. Restricciones del software

En esta sección se definen los problemas, restricciones o inconvenientes encontrados en todo el proceso

6. Anexos

En esta sección se especifican los anexos recopilados en el proceso (pruebas, entrevistas, encuestas, documentación)

impedimento o limitación a la hora de realizar alguna tarea, como ser la concreción de algún proyecto o trabajo



G R A C I A S

Línea de atención al ciudadano: 018000 910270
Línea de atención al empresario: 018000 910682



@SENAcomunica

www.sena.edu.co