## PROCESO DE PRUEBAS

### 1. Defina pruebas unitarias, pruebas de integración, pruebas del sistema y pruebas que se realizan en la implementación.

􀂃 **Pruebas unitarias:** El proceso de prueba se centra en los procesos lógicos internos del software, asegurando que todas las sentencias estén probadas y en las funciones externas, buscando que la entrada definida produzca resultados reales de acuerdo con los resultados requeridos. Las pruebas de unidades aseguran que los programas de aplicaciones funcionen de forma adecuada cuando se prueban de forma aislada con respecto a otros programas de aplicación.

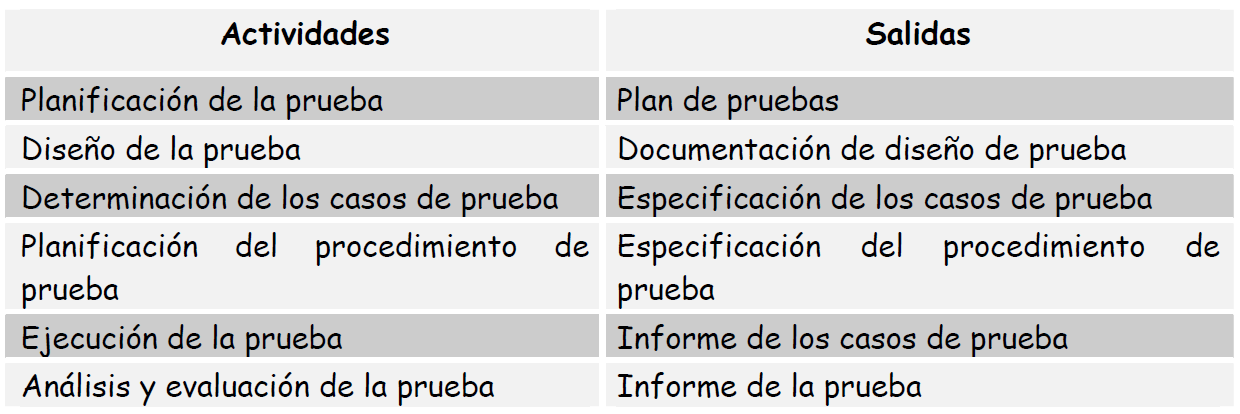
􀂃 **Pruebas de integración:** se buscan errores en la interrelación entre los diferentes componentes del sistema, o relación integral entre los subsistemas

􀂃 **Pruebas del sistema**: Las pruebas de sistemas aseguran que los programas de aplicaciones escritos de forma aislada funcionen adecuadamente cuando se integran en el sistema global. La entrada es la especificación de requerimientos del sistema.

􀂃 **Pruebas que se realizan en la implementación**: Pruebas de usuarios, pruebas de control de calidad, pruebas piloto.

### 2. Fundamente porque la prueba de del software es un elemento crítico para la garantía de calidad

### 3. Enuncie las actividades básicas de un procedimiento de pruebas y la documentación de salida correspondiente.



### 4. Defina planificación de la prueba

Planificación de la prueba: En la documentación para la planificación de la prueba

se deberá indicar:

􀂃 Objetivo de la prueba, detectar fallas,

􀂃 Objetos a probar, funcionalidades, procesos, etc.,

􀂃 Características a probar, dónde se probará, bajo qué entorno, etc.,

􀂃 Características a no probar,

􀂃 Método de prueba, el método utilizado en el desarrollo de la prueba,

􀂃 Tamaño de la prueba, la cantidad de casos de prueba a realizar,

􀂃 Recursos tecnológicos o máquinas. Recursos humanos, mínimo óptimo 3 personas asignadas, un planificador, un diseñador y un testeador,

􀂃 Reparto de responsabilidades, qué tareas tiene asignadas cada integrante,

􀂃 Plan de tiempo, Diagrama de Gantt,

􀂃 Productos a generar en el proceso de prueba, toda la documentación que tendrá la prueba una vez finalizada.

### 5. Defina diseño de la prueba

􀂃 Diseño de la prueba: En el diseño se deberá puntualizar con:

􀂃 Cómo llevar adelante la prueba para alcanzar el objetivo establecido,

􀂃 De qué forma se va a usar el método de prueba, brevemente qué método se utilizará y qué probará ese método,

􀂃 Criterios para pasar la prueba, es el punto más importante e indica qué escala utilizaremos para determinar si algo pasa la prueba o no.

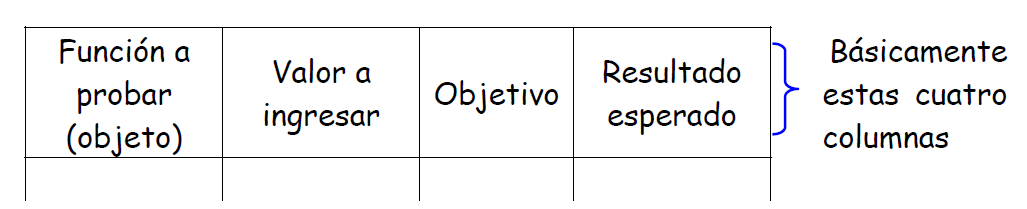
### 6. Defina determinación de los casos de prueba

**Determinación de los casos de prueba:** Determinar puntualmente los casos de

prueba, es decir, enumerar uno a uno los casos de pruebas a probar, indicar el

resultado correcto esperado al realizar algún caso de prueba, pudiéndose

utilizar una tabla de la siguiente forma:



### 7. Defina planificación del procedimiento de prueba

􀂃 **Planificación del procedimiento de prueba**: Se especificarán los procedimientos que es necesario seguir durante la ejecución de la prueba.

### 8. Defina ejecución de la prueba

􀂃 **Ejecución de la prueba:** Es en sí la ejecución de la prueba propiamente dicha, de acuerdo a la planificación del procedimiento de prueba.

### 9. Defina análisis y evaluación de la prueba

### 10. Describa un sistema de su elección y usando las planillas propuestas en el material documente para ese sistema: [a] plan de pruebas, [b] documento de diseño de la prueba, [c] especificación de los casos de prueba, [d] especificación del procedimiento de prueba, [e] informe de los casos de prueba ejecutados y [f] informe de la prueba.

## METODOS DE PRUEBAS

### 11. Defina Método de Prueba

Los métodos de prueba proporcionan un mecanismo de ayuda para asegurar las pruebas sean completas y para conseguir una mayor probabilidad de descubrir errores en el software.

### 12. Clasifique los métodos de prueba y fundamente.

* **Caja blanca**: los que se basan en revisar la codificación del software sin importar los datos de entrada o salida
  + Estilo de codificación (Condiciones – Flujo de datos –Bucles)
  + Caminos independientes
* **Caja negra**: no les interesa la codificación sino los datos de entrada y salida del software.
  + Adivinación de errores
  + Análisis de valores límite y clases de equivalencia (Análisis de valores de frontera)

### 13. Describa la prueba de estilo de codificación

Se tiene en cuenta los parámetros para una buena codificación. La prueba de condición es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. El método de prueba de flujo de datos selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.

### 14. Describa la prueba de Caminos independientes.

Permite garantizar que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo del sistema. Los pasos a seguir son los siguientes:

􀂃 usando el diseño o código como base, se dibuja el correspondiente grafo de flujo

􀂃 se determina la complejidad ciclomática V(G) del grafo de flujo resultante

􀂃 se determina un conjunto básico de caminos linealmente independientes

􀂃 se preparan los casos de prueba que forzarán la ejecución de cada

### camino del conjunto. 15. Describa la prueba de adivinación de errores.

Se supone qué resultados son los que podrían producir errores.

### 16. Describa la prueba de Análisis de valores límite (AVL) y clases de equivalencia (CE).

**Para el AVL se debe seleccionar:** 1 valor mayor y 1 valor menor del campo a analizar.

**Para cada una de las CE se deben elegir:**

• Rangos: 1 caso correcto y 2 casos incorrectos

• Lógicos: 1 caso válido y 1 caso inválido

• Textos: 1 caso correcto y 2 casos incorrectos

## GESTION DE CONFIGURACION

### 17. Defina Gestión de Configuración

Se puede definir como una disciplina cuya misión es controlar la evolución de un sistema software. La Gestión de Configuración se debe realizar a lo largo de todo el ciclo de vida del producto, tanto en el desarrollo como en el mantenimiento, hasta que el producto se retira

### 18. Defina los objetivos de las actividades de Gestión de Configuración del Software.

􀂃 establecer y mantener la integridad de los productos generados durante un proyecto de desarrollo de software y a lo largo de todo el ciclo de vida del producto.

􀂃 evaluar y controlar los cambios sobre ellos, es decir, controlar la evolución del sistema software.

􀂃 facilitar la visibilidad sobre el producto.

### 19. Enuncie condiciones de integridad de un producto software

􀂃 Satisface las necesidades del usuario (cumple todos los requisitos del usuario, tanto los explícitos como los implícitos)

􀂃 Cumple los requisitos de rendimiento.

􀂃 Se puede trazar su evolución desde que se concibió, y a través de todas las fases de su ciclo de vida.

### 20. Enuncie y defina las actividades de Gestión de Configuración del Software propuestas en estándar IEEE y las funciones adicionales que incluyen los sistemas que automatizan dicha gestión.

􀂃 **Identificación de la Configuración**: Consiste en identificar la estructura del producto, sus componentes y el tipo de estos, y en hacerlos únicos y accesibles de alguna forma. Esta tarea consiste en identificar y asignar nombres significativos y consistentes a todos y cada uno de los elementos que forman parte del producto software, en cada fase de su desarrollo, es decir, a cada uno de los Elementos de Configuración del Software.

􀂃 **Control de Cambios en la Configuración:** Consiste en controlar las versiones y entregas de un producto y los cambios que se producen en él a lo largo del ciclo de vida.

􀂃 **Generación de Informes de Estado:** Consiste en informar acerca del estado de los componentes de un producto y de las solicitudes de cambio, recogiendo estadísticas acerca de la evolución del producto. El objetivo es mantener a los usuarios, a los gestores y a los desarrolladores al tanto del estado de la configuración y su evolución. En definitiva, pretende dar respuesta a la pregunta “¿Qué ocurrió?”, y también a la pregunta “¿Cuándo ocurrió?”.

􀂃 **Auditoría de la Configuración:** Consiste en validar la completitud de un producto y la consistencia entre sus componentes, asegurando que el producto es lo que el usuario quiere.

### 21. Fundamente la complejidad de la Gestión de Configuración

* Número elevado de componentes a controlar
* Sin importar en qué momento del ciclo de vida del sistema nos encontremos, el sistema informático cambiará, y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida
* Sin una buena Gestión de Configuración, el mantenimiento de un producto puede ser una verdadera pesadilla.

### 22. Defina configuración del software y elementos de configuración del software.

* **CONFIGURACIÓN DEL SOFTWARE** conjunto de toda la información y productos utilizados o producidos en un proyecto como resultado del proceso de Ingeniería de Software
* ELEMENTO DE CONFIGURACIÓN DEL SOFTWARE (ECS) se le va a llamar A cada uno de los componentes de la configuración del software, El ECS es la unidad de trabajo para la GCS

􀂃 La especificación del sistema.

􀂃 El plan del proyecto software.

􀂃 La especificación de requisitos software.

􀂃 Un prototipo, ejecutable o en papel.

􀂃 El diseño preliminar.

􀂃 El diseño detallado.

􀂃 El código fuente.

􀂃 Programas ejecutables.

􀂃 El manual de usuario.

􀂃 El manual de operación e instalación.

􀂃 El plan de pruebas.

􀂃 Los casos de prueba ejecutados y los resultados registrados.

􀂃 Los estándares y procedimientos de ingeniería de software utilizados.

### 23. Defina línea base y justifique su uso

Para controlar los cambios sin impedir los cambios justificados se utiliza el concepto de LÍNEA BASE o “BASELINE”. Se puede definir una línea base como un punto de referencia en el proceso de desarrollo del software que queda marcado por la aprobación de uno o varios Elementos de Configuración del Software Las líneas base se establecen en hitos previamente especificados a lo largo del proceso de desarrollo

### 24. Liste y defina líneas base más comunes

􀂃 **Línea Base Funcional**, que se establece al finalizar la fase de análisis y especificación de los requisitos del sistema, y comprende todos aquellos documentos en los que se define el problema a resolver, los costes del proyecto, el plan de tiempos, y los diferentes requisitos funcionales, de Inter operatividad y de interfaz del sistema.

􀂃 **Línea Base de Distribución o Asignación de funciones**, que se establece al finalizar la fase de análisis y especificación de requisitos software, y comprende toda la documentación que gobernará el desarrollo de cada uno de los componentes software que se han identificado en la especificación del sistema, y la asignación o reparto de las diferentes funciones entre los distintos componentes del sistema

􀂃 **Línea Base de Diseño Preliminar**, que se establece al finalizar la fase de diseño preliminar. Comprende todos aquellos documentos en los que se define la arquitectura del producto software, así como el Plan de Pruebas.

􀂃 **Línea Base de Diseño**, que se establece al finalizar la fase de diseño detallado. Comprende todos aquellos documentos que contienen el diseño detallado del software y el plan de implementación, y también la descripción de los casos de prueba.

􀂃 **Línea Base de Producto**, que se establece al finalizar la fase de pruebas. Comprende los programas creados y todos aquellos documentos que contienen la información relativa a las pruebas realizadas.

􀂃 **Línea Base de Operación**, que se establece al finalizar la fase de implantación. Comprende los manuales de usuario, guías de operación y mantenimiento, manuales de formación, etc.

### 25. Defina Control de Cambios en la Configuración

Es la actividad de Gestión de Configuración más importante y su objetivo es proporcionar un mecanismo riguroso para controlar los cambios, partiendo de la base de que los cambios se van a producir. Normalmente combina procedimientos humanos y el uso de herramientas automáticas.

### 26. Liste y defina tipos de cambios

􀂃 Corrección de un defecto: Los clientes tienden a clasificar todos los cambios en esta categoría.

􀂃 Mejora del sistema: Los programadores, sin embargo, los suelen clasificar aquí.

### 27. Liste y defina niveles de control de cambios

* Control de cambios informal: Antes de que el Elemento de Configuración del Software pase a formar parte de una línea base, aquel que haya desarrollado el Elemento de Configuración del Software podrá realizar cualquier cambio justificado sobre él.
* Control de cambios al nivel del proyecto o semi-formal: Una vez que el Elemento de Configuración del Software pasa la revisión técnica formal y se convierte en una línea base, para que el encargado del desarrollo pueda realizar un cambio debe recibir la aprobación de:
  + El director del proyecto, si es un cambio local
  + El Comité de Control de Cambios, si el cambio tiene algún impacto sobre otros
  + Elementos de Configuración del Software
* Control de cambios formal: Se suele adoptar una vez que se empieza a comercializar el producto, cuando se transfieren los ECS a la Biblioteca Maestra. Todo cambio deberá ser aprobado por el Comité de Control de Cambios.

### 28. Liste, defina y de un esquema de las etapas típicas de un proceso formal para hacer un cambio sobre una línea base

* Iniciación del Cambio: se presenta una solicitud de cambio, que puede venir provocada por un problema que se ha detectado o por un cambio en los requisitos.
* Clasificación y registro de la solicitud de cambio.
* Aprobación o rechazo inicial de la solicitud de cambio. De ello suele ser responsable el Comité de Control de Cambios.
* Evaluación de la solicitud de cambio, si ha s ido aprobada, para calcular el esfuerzo técnico, los posibles efectos secundarios, el impacto global sobre otras funciones del sistema y el coste estimado del cambio. Como resultado se obtiene un Informe de Cambio.
* Se presenta el Informe de Cambio al Comité de Control de Cambios. Si se considera que el cambio es beneficioso se genera una Orden de Cambio (también llamada Orden de Cambio de Ingeniería), que describe el cambio a realizar, las restricciones que se deben respetar y los criterios de revisión y de auditoría. Esta Orden de Cambio es asignada a alguno de los ingenieros de software para que se encargue de llevarlo a cabo. En este momento, el objeto a cambiar se da de baja en la Biblioteca de Soporte al Proyecto.
* Se realiza el cambio, entrando en un proceso de seguimiento y control.
* Una vez finalizado el cambio, se certifica, mediante una revisión, que se ha efectuado correctamente el cambio y con ello se ha corregido el problema detectado o bien se han satisfecho los requisitos modificados. El objeto se devuelve a la Biblioteca de Soporte al Proyecto.
* Se notifica el resultado al originador del cambio.

### 29. Liste y defina tipos de registros asociados a la configuración.

􀂃 Registro de elementos de configuración: Conteniendo toda la información relativa a los diferentes elementos de configuración.

􀂃 Registro de líneas base: Conteniendo toda la información relativa a cada línea base

􀂃 Registro de solicitudes de cambios: El tipo de información que se suele mantener acerca de cada solicitud de cambio es la recogida a través del formulario de Solicitud de Cambio.

􀂃 Registro de cambios: El tipo de información que se suele mantener acerca de cada cambio es la recogida a través de el Informe de Cambio,la Orden de Cambio, el proceso de Gestión de Problemas, etc.

􀂃 Registro de modificaciones del código

􀂃 Registro de modificaciones sobre bases de datos

􀂃 Registro de modificaciones sobre documentación

􀂃 Registro de instalaciones: Su objetivo es mantener información acerca de todos los lugares en los que se ha instalado un producto software.

􀂃 Actas de las reuniones del Comité de Control de Cambios

### 30. Defina tipos de informes y de ejemplos

􀂃 Planificados

􀂃 Bajo demanda

### 31. Defina Auditoria de la Configuración

Una auditoría es una verificación independiente de un trabajo o del resultado de un trabajo o grupo de trabajos para evaluar su conformidad respecto de especificaciones, estándares, acuerdos contractuales u otros criterios. La auditoría de la Configuración es la forma de comprobar que efectivamente el producto que se está construyendo es lo que pretende ser.

### 32. Liste y defina tipos de actividades de Auditoria de la Configuración

􀂃 Revisiones de fase: Se realizan al finalizar cada fase del desarrollo y su objetivo es examinar los productos de dicha fase. Las revisiones propias de la Gestión de configuración son aquellas en las que se establecerán las líneas base. El objetivo de esta revisión es descubrir problemas, no comprobar que todo está bien. Hay que ser capaz de desenmascarar los problemas ocultos y sutiles, no sólo los que son obvios.

􀂃 Revisiones de cambios: Se realizan para comprobar que los cambios aprobados sobre una línea base se han realizado correctamente.

􀂃 Auditorías: Se realizan al final del proceso de desarrollo de software y su objetivo es examinar el producto en su conjunto.

### 33. Enuncie las funciones que implica la tarea de revisión de cambios

􀂃 Verificar que la configuración actual del software se corresponde con lo que era en fases anteriores. Debe haber correspondencia y trazabilidad entre los elementos de configuración que aparecen en una línea base y los que aparecen en las líneas base que la preceden y que la siguen. La verificación se realiza con respecto a la línea base precedente.

􀂃 Validar que la configuración actual del software satisface la función que se esperaba del producto en cada hito del proceso de desarrollo. La validación se realiza con respecto a los requisitos del sistema.

􀂃 Valorar si una determinada línea base, teniendo en cuenta los resultados de la verificación y validación, y otro tipo de comprobaciones, se debe considerar aceptable o no.

### 34. Liste y tipos de auditorías de configuración

􀂃 Auditoría Funcional: Cuyo objetivo es comprobar que se han completado todos los tests necesarios para el Elemento de Configuración auditado, y que, teniendo en cuenta los resultados obtenidos en los tests, se puede afirmar que el Elemento de Configuración satisface los requisitos que se impusieron sobre él.

􀂃 Auditoría Física: Cuyo objetivo es verificar la adecuación, completitud y precisión de la documentación que constituye las líneas base de diseño y de producto. Se trata de asegurar que representa el software que se ha codificado y probado. Tras la auditoría física se establece la línea base de Producto. Tiene lugar inmediatamente después de haberse superado la auditoría Funcional.

􀂃 Revisión Formal de Certificación: Cuyo objetivo es certificar que el Elemento de Configuración del Software se comporta

## AUDITORIA

### 35. Defina auditoria en informática

es la revisión y evaluación de los controles, sistemas, procedimientos de informática y de los equipos de cómputo, su utilización, eficiencia y seguridad, a fin de que por medio del señalamiento de cursos alternativos se logre una utilización más eficiente y segura de la información que servirá para una adecuada toma de decisiones.

### 36. Defina el alcance de la auditoria en informática

la evaluación de los equipos de cómputo o de un sistema o procedimiento específico, además habrá de evaluar los sistemas de información en general desde sus entradas, procedimientos, controles, archivos, seguridad y obtención de información

### 37. Fundamente la concepción de examen que se le da a la auditoria informática

para concluir en un conjunto de recomendaciones y un plan de acción. Es la elaboración de este plan de acción lo que diferencia la auditoría informática de una auditoría de gestión. La auditoría tradicional concluye emitiendo un juicio del estado de todo aquello que se ha verificado, la auditoría informática avanza un paso más y se atreve a elaborar un plan de acción.