



# Buenas prácticas de software



## Unidad 4. Buenas prácticas para los modelos de prueba de software.



# Buenas prácticas de software



## 4.1 Método verificar y validar.





# Buenas prácticas de software



## 4.1 Método verificar y validar.

La verificación y validación de software es un enfoque disciplinado para evaluar los **productos de software** a lo largo del ciclo de vida del producto.

La verificación y validación de software emplea técnicas de revisión, análisis y prueba para determinar si un **sistema de software** y sus productos intermedios cumplen con los requisitos.

Estos requisitos incluyen capacidades funcionales y atributos de calidad y la planificación del esfuerzo empleado en el proceso de V&V se realiza desde una etapa temprana del proyecto.



# Buenas prácticas de software



## Verificación y Validación (V&V)

Es el conjunto de procesos de comprobación y análisis, que aseguran que el software que se desarrolla está acorde a las especificaciones y cumple las necesidades de los clientes.



# Buenas prácticas de software



## En la verificación.

Se comprueba que el software cumple los requisitos **funcionales y no funcionales** definidos en la etapa de **análisis de requerimientos**.

Es el conjunto de actividades que aseguran que el software desempeña correctamente las funciones.

Considerando la gestión de desarrollo del software, la verificación hace referencia al **proceso para determinar si el flujo de trabajo se ha llevado** a cabo conforme lo especificado en el manual de procedimientos de trabajo.

El proceso de verificación debe aplicarse en cada etapa del software.



# Buenas prácticas de software



## La validación.

Es un procedimiento que garantiza que el software o sistema cumple con las buenas prácticas de desarrollo y construcción, garantizando la calidad de los procesos establecidos para el desarrollo.

Es un conjunto de actividades que se realizan para asegurar que el software construido cubre y corresponde a los requisitos del cliente.

El proceso de validación debe aplicarse en cada etapa del software, aunque en muchos casos se hace al final del desarrollo y antes de entregar el producto.





# Buenas prácticas de software



## Calidad del software.

La calidad del software se determina por el grado o medida en que el producto cumple con las especificaciones.

La calidad hace referencia a la excelencia, la que permite identificar que el software funciona correctamente y cumple con todos los requisitos del cliente.



# Buenas prácticas de software



En el proceso de **verificación y validación**, se utilizan dos técnicas de comprobación y análisis de software o sistema:

1. La Inspección del software: En donde se analizan y comprueban los requerimientos del sistema mediante la revisión del documento de requerimientos, los diagramas de diseño y el código fuente del programa.
2. Las inspecciones de software y los análisis automatizados son técnicas estáticas ya que no se requiere que el sistema se ejecute o esté en operación.
3. Las pruebas del software: Se llevan a cabo mediante la implementación del software o sistema y se utilizan los datos de prueba y se revisan las salidas del software y su comportamiento operacional, para comprobar que se desempeña conforme al modelado de requerimientos.

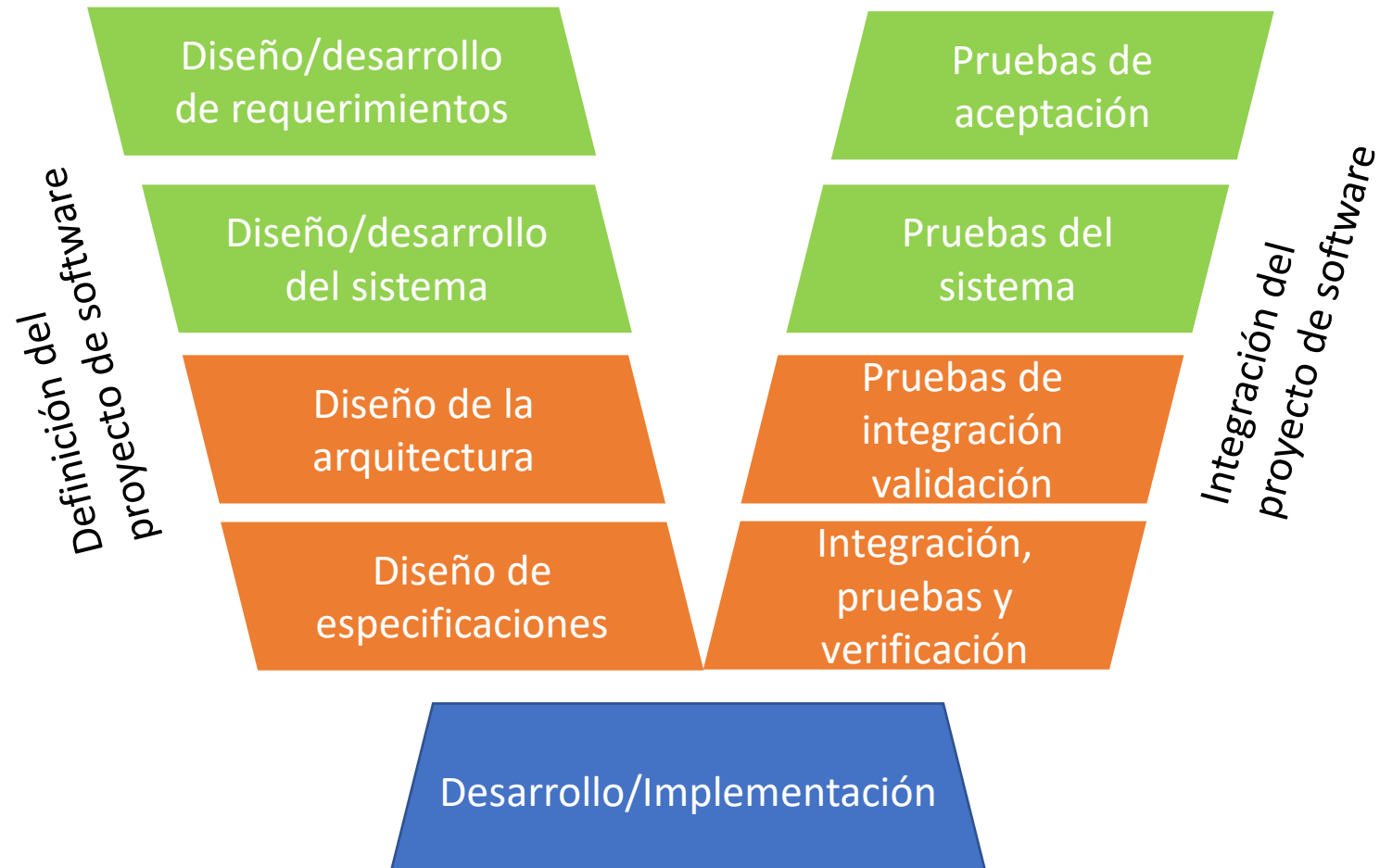




# Buenas prácticas de software



## Método verificar y validar (V&V)





# Buenas prácticas de software



## 4.1.1 Casos de prueba.

1. Un caso de prueba es un escenario de prueba que permite medir la funcionalidad en un conjunto de funciones o condiciones del software para verificar el resultado.
2. Los casos de prueba se aplican en el desarrollo de software o sistemas, se pueden utilizar pruebas manuales o se puede hacer uso de herramientas de gestión de casos de prueba.
3. Los casos de prueba y los scripts de prueba son diferentes.

Un script de prueba es un programa corto destinado a probar ciertas funciones.

Un caso de prueba se concentra en un documento, que contiene los pasos que deben completarse según lo planeado con anticipación.

Se debe tomar en cuenta que los casos de prueba deben ser meticulosamente planeados y contar con un instrumento de control.



# Buenas prácticas de software



Los casos de prueba se pueden diseñar en 8 apartado o pasos básicos.

## **Paso 1:** ID de caso de prueba.

Todos los casos de prueba deben llevar ID únicos para representarlos y/o identificarlos.

En la mayoría de los casos, seguir un **estándar** para el **ID** lo cual ayuda con la organización, la claridad y la comprensión.

## **Paso 2:** Descripción de la prueba.

Esta descripción debe detallar qué módulo, característica o función se está probando o qué se está verificando.

## **Paso 3:** Supuestos y condiciones previas.

Se describe el resultado que se espera de la función y permite verificar que la función cumple con el propósito.

## **Paso 4:** Datos de prueba.

Se deben indicar los datos que utiliza la función y la información que genera.



# Buenas prácticas de software



## **Paso 5:** Pasos a ejecutar.

Son las acciones que el usuario ejecuta y que espera el sistema responda.

## **Paso 6:** Resultado Esperado.

Son los resultados esperados después de la ejecución de la función.

## **Paso 7:** Resultado real y condiciones posteriores.

Se realiza la comparación del resultado obtenido con el esperado, por lo que se puede determinar el estado de funcionamiento del software.

## **Paso 8:** Determinación de resultados.

La comparación de los resultados da el indicador de falla o satisfactorio.

Mismo resultado = Aprobado Diferentes resultados = Falla



## Test de prueba.

Se debe considerar elaborar un test de prueba para identificar los resultados reales y poder hacer los comentarios, **ajustes, correcciones** o lo que proceda para que el software **cumpla con los requerimientos**.



# Buenas prácticas de software



## Ejemplo de test de prueba.

ID del caso de prueba	CP-FR-01
Nombre del caso de prueba	Autorización de crédito.
Descripción	Se autoriza al cliente el crédito solicitado en función de su límite de crédito.
Precondiciones	El cliente debe tener como límite de crédito \$10,000
Pasos a seguir	<ol style="list-style-type: none"><li>1. Ingresar nombre de cliente.</li><li>2. Revisar el límite de crédito.</li><li>3. Revisar la cantidad solicitada de crédito.</li><li>4. Validar la autorización.</li><li>5. Autorizar crédito.</li></ol>
Resultado esperado	El cliente obtiene la autorización del crédito.
Resultado real	<b>No se encuentra al cliente.</b> Comprobante de autorización de crédito.
Comentarios	Compra autorizada.





# Buenas prácticas de software



## Mejores prácticas para redactar casos de prueba.

1. Mantenga las cosas simples y transparentes.
2. Haga que los casos de prueba sean reutilizables.
3. Mantenga únicos los ID de los casos de prueba.
4. La revisión por pares es importante.
5. Los casos de prueba deben tener en cuenta al usuario final o los requisitos definidos.
6. Especifique los resultados esperados y los supuestos.



## 4.1.2 Matriz de trazabilidad.

La matriz de trazabilidad de requisitos (RTM) describe y sigue la vida de un requerimiento.

La RTM sigue y vincula los requisitos del proyecto, desde su origen hasta los entregables que lo satisfacen.

La matriz de trazabilidad ayuda a asegurar que cada requerimiento se cubrió y es el que se entrega.

La matriz de trazabilidad es de doble sentido va hacia adelante como hacia atrás o bien puede ser bidireccional.

Permite identificar si los resultados del desarrollo alcanzo y cumplió cada requisito solicitado.

Además, se visualizan que requisitos son los que ya se consideran entregables.

Permiten hacer el seguimiento de la ingeniería de requerimientos durante el ciclo de vida y se asegura que estos sean entregados según especificaciones.

Mediante el análisis gráfico es fácil verificar que se cumplen todos los requisitos mediante la validación de sus correspondientes requerimientos funcionales, casos de uso y casos de prueba.



# Buenas prácticas de software



## Tipos de RTM.

**Trazabilidad hacia delante RTMs:** Se asegura que el proyecto avance en la dirección deseada, mediante el mapeo de los requisitos para probar los casos de prueba.

**RTMs de trazabilidad retrospectiva:** Mediante esta matriz se revisa que el proyecto no ha tenido desvíos considerando los requerimientos iniciales y que el proyecto sigue en el camino correcto así como permite visualizar que el alcance original siga siendo el mismo.

**Trazabilidad bidireccional RTMs:** Esta matriz abarca tanto la trazabilidad hacia delante como hacia atrás, asegurando que todos los requisitos especificados tengan los casos de prueba correspondientes y viceversa.



# Buenas prácticas de software



## Elementos de una matriz de trazabilidad.

**ID:** Número que permite identificar a cada uno de los requisitos.

**Tipo:** Responde a la clasificación que queramos establecer dentro de nuestro proyecto. Tanto este como cualquier otro atributo del requisito que queramos incluir en la matriz de seguimiento nos ayudarán a una mejor gestión y trazabilidad de su evolución, y a asegurar el resultado final del proyecto.

**Prioridad:** Permite indicar la prioridad asignada al requisito y permite hacer la valoración para tomar las decisiones cuando se está evaluado del desarrollo del proyecto.

**Estado:** Permite identificar el estado de desarrollo en que se encuentra el requisito, este puede ser activo, en proceso, terminado o cancelado.

**Objetivo:** es el objetivo que se cumple con el requisito.

**Entregable:** Se refiere al entregable que satisface el requisito.

**Estado del entregable:** Identifica cual es el estado del requisito entregable.



# Buenas prácticas de software



## Ejemplo de matriz de trazabilidad.

ID	Descripción	Tipo	Prioridad	Estado	Objetivo	Entregable	Estado
CP1-RF-01	El sistema recogerá los siguientes datos...	Venta	Alta	Activo	Obtener los datos necesarios de los...	Registrar, eliminar, modificar...	Pendiente
02	Los datos serán ingresados por...	Venta	Alta	Activo	Garantizar la seguridad de la información	Registrar, eliminar, modificar...	Pendiente
03							
04							
05							



# Buenas prácticas de software



## Herramienta para hacer una matriz de trazabilidad.

### Visure Requirements

Permite generar fácilmente una RTMs, detallando y mostrando elementos tanto en las cabeceras de las columnas como en las filas.

Así mismo permite en cada celda indicar si los elementos de la columna y la fila correspondientes se han trazado o no, así como la dirección de la traza, en caso de que proceda.

Puedes hacer una prueba en la siguiente liga.

<https://visuresolutions.com/es/>





# Buenas prácticas de software



## 4.1.3 Plan de pruebas.

En el desarrollo de software, la fase de pruebas es crítica para asegurar que el producto sea enviado a ambiente de producción con la calidad esperada por el cliente.

Es importante contar con un plan de pruebas de software para:

1. Especificar puntualmente las funciones a probar.
2. Identificar como serán ejecutadas cada una de las pruebas
3. Quién será el responsable de ejecutar las pruebas.
4. Hacer un cronograma para la ejecución de las pruebas.
5. El plan de pruebas puede elaborarse para que se realicen en cada interacción o a un conjunto determinado de casos o bien a todo el software.
6. Por otro lado, se pueden definir jerarquías de casos de prueba a realizar tomando en cuenta la importancia.
7. Hay que tener muy clara la manera en la que vamos a hacer las prueba, como vamos a probar, de los recursos de los que vamos a disponer.



[Esta foto](#) de Autor desconocido  
está bajo licencia [CC BY-SA-NC](#)



# Buenas prácticas de software



## Ejemplos de secciones de la plantilla del plan de pruebas.

**Historial de versiones:** Se indica quien elaboró la versión y la fecha.

**Ficha del proyecto:** Se indican los datos referentes al proyecto, así como el responsable de las pruebas.

**Pruebas realizadas:** En este apartado se deben indicar el tipo de pruebas que se realizarán, se realizaron o quedaron pendientes.

**Aprobaciones:** Aquí se indica quién aprueba el plan de pruebas, quien revisa y aprueba los resultados de las pruebas realizadas.

**Ejecutor de las pruebas:** Se debe especificar el profesional del servicio que hace las pruebas.

**Resumen ejecutivo:** Se debe elaborar un documento que contenga el plan de pruebas con cada uno de los apartados y el registro del resultado de las pruebas, dicho documento debe estar firmado por cada uno de los participantes en el plan y en las pruebas.



# Buenas prácticas de software



## Criterios de aceptación o rechazo:

En el plan de pruebas se deben especificar los criterios para dar por concluidas las pruebas.

Estos pueden ser:

**Pruebas completadas:** 100% de las estableciendo el 100% de efectividad para considerarlas como casos exitosos. Por ejemplo, considerar que se realizaron todos los tipos de pruebas, que se logró un 0% de errores o defectos corregidos.

**Suspensión de las pruebas:** Se debe definir o establecer con claridad bajo qué condiciones se detienen un conjunto de casos de pruebas. Por ejemplo, en caso de existir defectos que impidan la ejecución de más casos de pruebas.

**Reanudación de las pruebas:** Cuando se suspendieron las pruebas por algún motivo, es importante establecer las pautas de reanudación. Por ejemplo, después de ver dado solución a un mal funcionamiento de la función que ocasiono la suspensión de la prueba.

**Entregables:** Se debe entregar el documento del plan de las pruebas, los casos de pruebas, la especificación de diseño de los casos de prueba, los resultados de las pruebas, reportes de errores (bugs), evidencias de pruebas, reportes emitidos por herramientas de pruebas, la solución de los errores, entre otros.