

# GUÍA DE LABORATORIO N° 19

## Implementación de casos de prueba

---

### GUÍA DE LABORATORIO N° 19

**Actividad de Proyecto: Ejecutar y documentar pruebas del software que cumplan con los estándares de calidad**

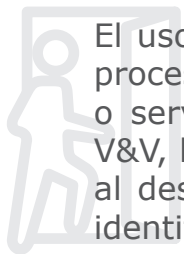
Estructura de contenidos

1. INTRODUCCIÓN .....	2
2. OBJETIVOS .....	2
3. CONSIDERACIONES .....	3
4. PROCEDIMIENTO .....	3
5. EVIDENCIAS A ENTREGAR .....	6
GLOSARIO .....	7
BIBLIOGRAFÍA .....	8





## 1. INTRODUCCIÓN

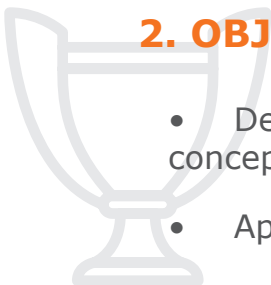


El uso de buenas prácticas en el desarrollo de software, requiere de un proceso de verificación y validación (V&V) adecuado para que el producto o servicio final alcance el nivel de calidad esperado por el cliente. En V&V, las pruebas de software o testing son fundamentales, pues proveen al desarrollador de un conjunto de técnicas dinámicas que permiten la identificación y depuración de fallos en el sistema construido.

Es necesario que como desarrollador de aplicaciones de software, en su proceso de formación académica adquiera las competencias para el diseño adecuado de un plan de pruebas, definiendo los casos de prueba y sus requerimientos.

En el presente laboratorio se presenta una situación problemática de aprendizaje, con actividades prácticas, mediante las cuales se busca el afianzamiento de los fundamentos teóricos relacionados con las pruebas de software.

## 2. OBJETIVOS



- Desarrollar actividades prácticas, que permitan el afianzamiento de conceptos relacionados con las pruebas de software.
- Aplicar los fundamentos teóricos presentados en el material de

apoyo, para el desarrollo de actividades que ayuden al entrenamiento en el diseño y aplicación de casos de prueba.

- Implementar casos de prueba, que permitan identificar elementos fundamentales en el diseño del plan de pruebas de software.
- Utilizar técnicas tradicionales en pruebas de software, basadas en la ejecución del código con el enfoque de caja blanca, fundamentales en la identificación de errores lógicos.

### 3. CONSIDERACIONES

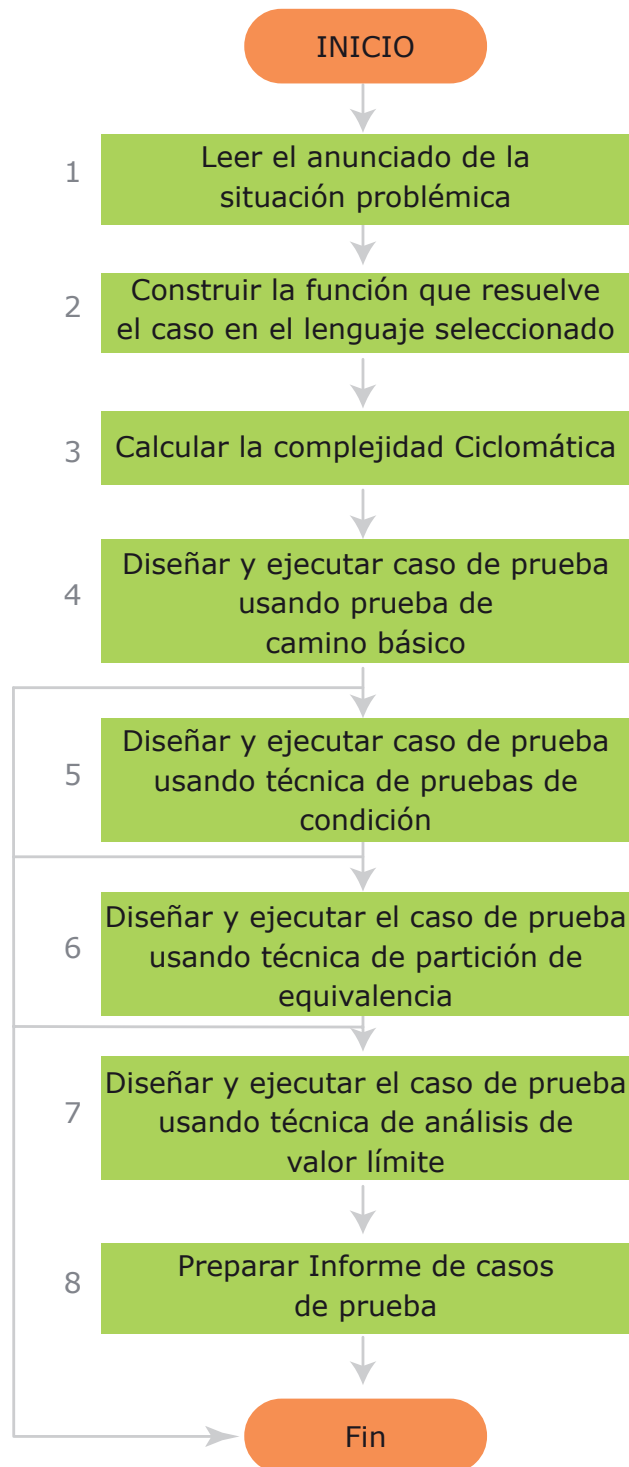
Item	Descripción
Soporte Teórico	<ul style="list-style-type: none"><li>• Objeto de aprendizaje pruebas de software.</li><li>• Objetos de aprendizaje de la fase de diseño.</li><li>• Objetos de aprendizaje de la fase de desarrollo</li></ul>
Productos requeridos	<ul style="list-style-type: none"><li>• Laboratorios de programación.</li><li>• Requerimientos del ejercicio a desarrollar.</li><li>• Código del sistema de información del proyecto de grado</li><li>• Plantilla de Caso de prueba.</li></ul>
Herramientas SW	Ambiente de Desarrollo del Lenguaje de programación de su preferencia.

### 4. PROCEDIMIENTO

Las siguientes actividades propuestas en el laboratorio, son de tipo práctico que buscan el entrenamiento del aprendiz en el diseño de casos de prueba para la evaluación de la lógica interna de un programa.

Se aplicarán técnicas de caja blanca en forma tradicional, que permitirán la construcción de la base de conocimiento necesaria para la implementación de buenas prácticas en las pruebas de software.

Los aprendizajes darán al estudiante lineamientos para incorporar en su proceso de pruebas si lo prefiere, software especializado que facilitará y ampliará las opciones en el diseño de pruebas.



### Descripción de actividades:

1. Analizar la siguiente situación problemática:

Una empresa que fabrica piezas torneadas para vehículos, paga a sus empleados de acuerdo a las horas trabajadas y a una tarifa de pago por hora, de acuerdo a la siguiente tabla:

Categoría	tarifa
1	12000
2	17000
3	22000

Si la cantidad de horas trabajadas es mayor a 40 horas, la tarifa se incrementa en un 25% para las horas extras. Calcular el salario del trabajador dadas las horas trabajadas y la categoría.

2. Construir una función que resuelva el caso presentado en la situación problemática e implementarla en el lenguaje de programación deseado.
3. Calcular la complejidad ciclomática, determinar los caminos resultantes, y casos de prueba según la técnica de prueba del camino básico.
4. Diseñar casos de prueba usando la técnica de pruebas de condición.
5. Diseñar casos de prueba usando la técnica de partición de equivalencia.
6. Diseñar casos de prueba usando la técnica de análisis de valores límite.
7. Presentar informe de los casos de prueba, usando el instrumento Plantilla\_Caso\_de\_prueba.xls.





## 5. EVIDENCIAS A ENTREGAR

- a. Código del Programa que implemente la función solicitada.
- b. Documento con:
  - Cálculo de la complejidad ciclomática.
  - Caminos resultantes según la técnica del camino básico. Construir grafo de flujo.
  - Casos de prueba según la técnica de pruebas de condición.
  - Casos de prueba según la técnica de partición de equivalencia.
  - Casos de prueba según la técnica de análisis de valores límite.
- c. Plantilla con el registro de los casos de prueba.

## GLOSARIO

---

**Función:** microprograma o módulo dentro de un programa, que contiene bajo un solo nombre pequeñas rutinas que pueden ser utilizadas una o varias veces.

**Grafo:** representación gráfica que representa relaciones binarias entre elementos de un conjunto, por medio de nodos y enlaces.

**Complejidad ciclomática:** medida que permite determinar el número de caminos a probar usando la técnica de camino básico.



## BIBLIOGRAFÍA

---

BOLAÑOS, D., SIERRA, A., & ALARCÓN, M. (2008). Pruebas de Software y JUnit. Madrid: Pearson Prentice Hall.

PRESSMAN, R. (2006). Ingeniería del Software: Un enfoque práctico. Sexta edición. McGrawHill.

IEEE Standard Glossary of Software Engineering Terminolgy [IEEE, 1990].





<b>GUÍA DE LABORATORIO N° 19</b> <b>Implementación de casos de prueba</b>	
Desarrollador de contenido Experto temático	José Ricardo Arismendi Santos
Asesor pedagógico	Rafael Neftalí Lizcano Reyes
Producción Multimedia	Luis Fernando Botero Mendoza
Líder expertos temáticos	Ana Yaqueline Chavarro Parra
Líder línea de producción	Santiago Lozada Garcés

