

# SÍLABO

## Pruebas y Calidad de Software

<b>Código</b>	ASUC01006	<b>Carácter</b>	Obligatorio	
<b>Prerrequisito</b>	Construcción de Software			
<b>Créditos</b>	4			
<b>Horas</b>	<b>Teóricas</b>	2	<b>Prácticas</b>	4
<b>Año académico</b>	2025-00			

### I. Introducción

---

Pruebas y Calidad de Software es una asignatura obligatoria de especialidad que se ubica en el octavo periodo de la Escuela Académico Profesional de Ingeniería de Sistemas e Informática. Tiene como requisito haber aprobado la asignatura de Construcción de Software. Desarrolla en un nivel logrado las competencias transversales Experimentación y Medioambiente y Sostenibilidad, y en un nivel intermedio la competencia específica Análisis de Problemas. La relevancia de la asignatura reside en preparar al estudiante para describir y aplicar los procesos de pruebas y calidad del software.

**Los contenidos generales que la asignatura desarrolla son los siguientes:** fundamentos de las pruebas de software; pruebas a través del ciclo de vida del software; niveles de pruebas; técnicas de diseño de pruebas de software; pruebas funcionales; pruebas no funcionales, proceso de pruebas, gestión de pruebas, herramientas de pruebas de software, calidad de software, procesos de gestión de calidad del software. Consideraciones prácticas de la calidad del software

---

### II. Resultado de aprendizaje de la asignatura

---

Al finalizar la asignatura, el estudiante será capaz de realizar pruebas funcionales y no funcionales de un software para identificar y corregir la inyección de defectos que se cometieron en las actividades previas del periodo de vida de desarrollo de software.

---

**III. Organización de los aprendizajes**

<b>Unidad 1 Calidad de software</b>		<b>Duración en horas</b>	24
<b>Resultado de aprendizaje de la unidad</b>	Al finalizar la unidad, el estudiante será capaz de utilizar los atributos de calidad de un software, basándose en los modelos de calidad y estándares.		
<b>Ejes temáticos</b>	<ol style="list-style-type: none"> <li>1. Introducción a la calidad y pruebas de software</li> <li>2. Calidad del producto y modelos de calidad.</li> <li>3. Aseguramiento de la calidad</li> <li>4. Atributos de calidad: Modelo ISO 9126</li> </ol>		

<b>Unidad 2 Plan de pruebas de software</b>		<b>Duración en horas</b>	24
<b>Resultado de aprendizaje de la unidad</b>	Al finalizar la unidad, el estudiante será capaz de establecer la estrategia y planificación de las pruebas de software.		
<b>Ejes temáticos</b>	<ol style="list-style-type: none"> <li>1. Estrategias de pruebas de software</li> <li>2. Estimación de las pruebas de software</li> <li>3. Planificación de las pruebas de software</li> <li>4. Métricas de la calidad del software</li> </ol>		

<b>Unidad 3 Tipos de pruebas de software</b>		<b>Duración en horas</b>	24
<b>Resultado de aprendizaje de la unidad</b>	Al finalizar la unidad, el estudiante será capaz de aplicar técnicas de pruebas de software de acuerdo a cada proyecto.		
<b>Ejes temáticos</b>	<ol style="list-style-type: none"> <li>1. Pruebas unitarias e integración</li> <li>2. Pruebas funcionales</li> <li>3. Pruebas de regresión</li> <li>4. Pruebas de requerimientos no funcionales</li> <li>5. Pruebas de aceptación</li> </ol>		

<b>Unidad 4 Automatización de pruebas</b>		<b>Duración en horas</b>	24
<b>Resultado de aprendizaje de la unidad</b>	Al finalizar la unidad, el estudiante será capaz de realizar pruebas funcionales y no funcionales de un software, aplicando técnicas de pruebas con herramientas de automatización, asegurando la calidad del software.		
<b>Ejes temáticos</b>	<ol style="list-style-type: none"> <li>1. Herramientas de automatización</li> <li>2. Herramientas de pruebas unitarias</li> <li>3. Herramientas para pruebas funcionales, de integración</li> <li>4. Herramientas para pruebas de requerimientos no funcionales</li> </ol>		

#### IV. Metodología

##### Modalidad Presencial, Semipresencial, A Distancia

El desarrollo de la asignatura en ambas modalidades está basado en las metodologías:

- Aprendizaje experiencial
- Aprendizaje colaborativo
- Aprendizaje orientado a proyectos

#### V. Evaluación

##### Modalidad Presencial

Rubros	Unidad por evaluar	Fecha	Entregable/Instrumento	Peso parcial	Peso total
Evaluación de entrada	Prerrequisito	Primera sesión	Evaluación individual teórica / <b>Prueba objetiva</b>	<b>0 %</b>	
Consolidado 1 <b>C1</b>	1	Semana 1 - 4	Evaluación individual teórico-práctica / <b>Prueba de desarrollo</b>	40 %	<b>20 %</b>
	2	Semana 5 - 7	Ejercicios grupales de análisis de casos desarrollados en clase / <b>Rúbrica de evaluación</b>	60 %	
Evaluación parcial <b>EP</b>	1 y 2	Semana 8	Informe sobre estrategias y planificación de las pruebas de software / <b>Rúbrica de evaluación</b>	<b>20 %</b>	
Consolidado 2 <b>C2</b>	3	Semana 9 - 12	Evaluación individual teórico-práctica / <b>Prueba de desarrollo</b>	40 %	<b>20 %</b>
	4	Semana 13 - 15	Ejercicios grupales de análisis de casos desarrollados en clase / <b>Rúbrica de evaluación</b>	60 %	
Evaluación final <b>EF</b>	Todas las unidades	Semana 16	Informe final sobre pruebas funcionales y no funcionales de un software / <b>Rúbrica de evaluación</b>	<b>40%</b>	
Evaluación sustitutoria*	Todas las unidades	Fecha posterior a la evaluación final	<b>Aplica</b>		

\* Reemplaza la nota más baja obtenida en los rubros anteriores.

**Modalidad Semipresencial - Blended**

Rubros	Unidad por evaluar	Fecha	Entregable/Instrumento	Peso parcial	Peso Total
Evaluación de entrada	Prerrequisito	Primera sesión	- Evaluación individual teórica / <b>Prueba objetiva</b>	<b>0 %</b>	
Consolidado 1 <b>C1</b>	1	Semana 1-3	- Actividades virtuales	15 %	<b>20 %</b>
			- Evaluación individual teórico-práctica / <b>Prueba de desarrollo</b>	85 %	
Evaluación parcial <b>EP</b>	1 y 2	Semana 4	- Informe sobre estrategias y planificación de las pruebas de software / <b>Rúbrica de evaluación</b>	<b>20 %</b>	
Consolidado 2 <b>C2</b>	3	Semana 5-7	- Actividades virtuales	15 %	<b>20 %</b>
			- Evaluación individual teórico-práctica / <b>Prueba de desarrollo</b>	85 %	
Evaluación final <b>EF</b>	Todas las unidades	Semana 8	- Informe final sobre pruebas funcionales y no funcionales de un software / <b>Rúbrica de evaluación</b>	<b>40 %</b>	
Evaluación sustitutoria *	Todas las unidades	Fecha posterior a la evaluación final	<b>Aplica</b>		

\* Reemplaza la nota más baja obtenida en los rubros anteriores.

**Modalidad A Distancia**

Rubros	Unidad por evaluar	Fecha	Entregable/Instrumento	Peso Total
Evaluación de entrada	Prerrequisito	Primera sesión	Evaluación individual teórica / <b>Prueba objetiva</b>	<b>0 %</b>
Consolidado 1 <b>C1</b>	1	Semana 2	Ejercicios grupales de análisis de casos desarrollados en clase / <b>Rúbrica de evaluación</b>	<b>20 %</b>
Evaluación parcial <b>EP</b>	1 y 2	Semana 4	Informe sobre estrategias y planificación de las pruebas de software / <b>Rúbrica de evaluación</b>	<b>20 %</b>
Consolidado 2 <b>C2</b>	3	Semana 6	Ejercicios grupales de análisis de casos desarrollados en clase / <b>Rúbrica de evaluación</b>	<b>20 %</b>
Evaluación final <b>EF</b>	Todas las unidades	Semana 8	Informe final sobre pruebas funcionales y no funcionales de un software / <b>Rúbrica de evaluación</b>	<b>40 %</b>
Evaluación sustitutoria*	Todas las unidades	Fecha posterior a la evaluación final	<b>Aplica</b>	

\* Reemplaza la nota más baja obtenida en los rubros anteriores.

Fórmula para obtener el promedio:

$$PF = C1 (20 \%) + EP (20 \%) + C2 (20 \%) + EF (40 \%)$$

## VI. Bibliografía

### Básica

Black, R. y Rueda, G. (2011). *Fundamentos de software: claves básicas para aprobar el examen de "Probador Certificado" (Certified Tester) conforme al estándar ISTQB*. Editorial RBCS. <https://at1z.short.gy/ieSF9o>

### Complementaria

Axelrod, A. (2018). *Complete guide to test automation: Techniques, practices, and patterns for building and maintaining effective software projects*. Apress.

Goerick, S. (2020). *The future of software quality assurance*. Springer.  
<https://bit.ly/3zSxRkB>

Graham, D., Black, R. y Veenendaal, E. (2019). *Foundations of Software Testing ISTQB Certification* (4.ª ed.). Cengage Learning.

Lancu, L. (2019). *QA Quality assurance & software testing fundamentals*. Independently published.

Pantaleo, G. (2016). *Calidad en el desarrollo del software* (2.ª ed.). Alfaomega.

Shen, J. (2019). *Software testing: Techniques, principles, and practices*. Independently published.

Walkinshaw, N. (2017). *Software Quality Assurance: Consistency in the Face of Complexity and Change* (Undergraduate Topics in Computer Science). Springer.

## VII. Recursos digitales

Cochiaro, C. (2018). *Selenium Framework Design in Data – Driven Testing: Build data – driven test Frameworks using Selenium Web Driver, AppiumDriver, Java y TestNG*. Packt Publishing

Intellipaat. (18 de noviembre de 2019). *SonarQube Tutorial | Installing and Configuring SonarQube | What is SonarQube | Intellipaat* [Video]. YouTube.  
<https://www.youtube.com/watch?v=31igoWxauEQ>

Matam, S. y Jain, J. (2017). *Pro Apache Jmeter: Web Application Performance Testing*. Apress.

SonarQube. (2021). *SonarQube 7.9.6 (former LTS)* [Software de computadora].  
<https://www.sonarqube.org/sonarqube-7-9-lts/>