

Fecha: Febrero de 2026

**PROGRAMA ACADÉMICO: INGENIERIA DE SISTEMAS Y COMPUTACIÓN**

**SEMESTRE: NOVENO**

**ASIGNATURA: ELECTIVA I - INGENIERÍA DE PRUEBAS DE SOFTWARE**

**CÓDIGO: 8108276**

**NÚMERO DE CRÉDITOS: 3**

### PRESENTACIÓN

El curso Ingeniería de Pruebas de Software está diseñado para que, una vez finalizadas las asignaturas del área de Ingeniería de Software, el estudiante profundice en las prácticas de aseguramiento de la calidad (QA) y en los métodos de prueba que contribuyen a mejorar la calidad de los productos de software. Su propósito es fortalecer las competencias del estudiante tanto en su rol como desarrollador, promoviendo buenas prácticas de verificación y validación, como en la exploración de nuevas alternativas profesionales vinculadas al campo del aseguramiento de la calidad del software.

La asignatura aborda los fundamentos, técnicas y herramientas propias del proceso de prueba de software, entendido como una etapa esencial antes del despliegue de un proyecto al cliente. A través de la correcta aplicación de estrategias de prueba, el estudiante comprenderá la importancia de anticipar fallas, validar los requerimientos funcionales y no funcionales, y garantizar que el producto final cumpla con altos estándares de calidad y satisfaga las expectativas del usuario.

Entre los temas a desarrollar se incluyen: pruebas unitarias, pruebas de comportamiento, pruebas de sistema y aceptación, automatización de pruebas, así como pruebas específicas para aplicaciones web y móviles.

### JUSTIFICACIÓN

El programa de Ingeniería de Sistemas y Computación de la UPTC busca formar profesionales capaces de comprender en profundidad los fundamentos, propósitos y alcances de las pruebas de software, reconociendo su papel esencial en el aseguramiento de la calidad y en el éxito de los proyectos de desarrollo. La asignatura de Ingeniería de Pruebas de Software tiene como propósito fortalecer esta comprensión, promoviendo en el estudiante una visión crítica sobre los desafíos, limitaciones y buenas prácticas asociadas al proceso de prueba.

Asimismo, busca desarrollar competencias técnicas y prácticas que permitan al futuro profesional aplicar adecuadamente las principales técnicas de prueba (Como las pruebas estáticas, de caja blanca y de caja negra), al tiempo que se familiariza con la terminología estándar y las metodologías internacionales del área. A través de estas experiencias, el estudiante obtendrá una visión integral de las actividades y procesos de prueba, desde la revisión de requisitos hasta la implementación final del sistema, consolidando una cualificación profesional valorada por empleadores, clientes y el sector del desarrollo de software.

### COMPETENCIAS

La asignatura está ligada a fortalecer competencias referentes al proceso de ingeniería de software, enfatizando en la calidad de software y, las mejores prácticas, al final del curso se espera que el estudiante:

- Conocer la terminología estándar, objetivos de la prueba, su rol en el ciclo de vida del software y sus relaciones con otros procesos de desarrollo.
- Interpretar especificaciones de requisitos funcionales y no funcionales para determinar enfoques, alcance y cobertura de pruebas apropiados.
- Elaborar casos de prueba estructurados y reproducibles utilizando técnicas de caja negra y caja blanca como análisis de valores límite, partición de equivalencia, análisis de flujo de control y decisión, y métodos basados en modelos.
- Diseñar estrategias de prueba que puedan ser implementadas en aplicaciones web, móviles, APIs, bases de datos, entre otras, considerando tecnologías específicas y plataformas.
- Crear casos de prueba que permitan automatizar las pruebas funcionales mediante el uso de herramientas estándar de la industria para pruebas funcionales, de regresión y de rendimiento.

### RESULTADOS DE APRENDIZAJE

- RA1: Reconocer y aplicar terminología estándar en procesos de prueba: El estudiante será capaz de aplicar la terminología estandarizada de la ingeniería de pruebas en contextos reales de desarrollo, utilizando correctamente conceptos como verificación, validación, defecto, fallo, riesgo y demás terminología relacionada con Ingeniería de pruebas.
- RA2: Analizar especificaciones de requisitos para determinar estrategias de prueba: El estudiante será capaz de examinar y descomponer especificaciones de requisitos, diferenciando entre requisitos funcionales y no funcionales, identificando criterios de aceptación, riesgos implícitos y determinando la estrategia de prueba más efectiva en términos de cobertura, enfoque y recursos.
- RA3: Diseñar casos de prueba aplicando técnicas de caja negra y caja blanca: El estudiante será capaz de crear casos de prueba detallados y ejecutables aplicando técnicas de caja negra (partición de equivalencia, análisis de valores límite, métodos basados en modelos) y caja blanca (análisis de flujo de control, análisis de decisión, cobertura de caminos), asegurando que cada caso sea independiente, reproducible y verificable.
- RA4: Analizar contextos tecnológicos para seleccionar estrategias de prueba específicas: El estudiante será capaz de examinar características distintivas de diferentes plataformas y tecnologías (web, móvil, APIs, bases de datos, etc) e identificar las estrategias de prueba, herramientas y técnicas más apropiadas para cada contexto, considerando limitaciones, ventajas y riesgos específicos.
- RA5: Implementar casos de prueba para automatización de pruebas funcionales: El estudiante será capaz de diseñar e implementar casos de prueba automatizables que utilicen herramientas estándar para ejecutar pruebas funcionales, de regresión y de rendimiento de forma repetible, mantenible y eficiente, integrándose en pipelines de CI/CD.

### METODOLOGÍA

El Aprendizaje Basado en Problemas (ABP) se define como "un método de aprendizaje basado en el principio de usar problemas como punto de partida para la adquisición e integración de nuevos conocimientos". La estrategia central del curso es el ABP, diseñado para situar al estudiante en un contexto profesional simulado. A



diferencia de la enseñanza tradicional, donde primero se expone la teoría y luego se aplica, en este curso los estudiantes se enfrentarán a un desafío de ingeniería real (asegurar la calidad de un producto de software) que requerirá la adquisición y aplicación inmediata de los conceptos teóricos, técnicas de diseño y herramientas de automatización.

De igual manera en el curso se complementará con la metodología conocida como El Aprendizaje Basado en Casos (CBL - Case-Based Learning), la cual es una estrategia de enseñanza centrada en quien aprende en la que se analizan problemas auténticos, reales o ficticios, para abordar situaciones, resolver preguntas e identificar soluciones. Es una metodología didáctica en el cual los estudiantes analizan situaciones-problema contextualizadas en narrativas que representan desafíos reales. A diferencia de otros métodos, el CBL incluye una estructura narrativa explícita que permite a los estudiantes experimentar la incertidumbre y complejidad de situaciones auténticas, fomentando pensamiento crítico y toma de decisiones fundamentadas.

Por último se aplicará la metodología conocida como Aprendizaje Basado en Proyectos (ABP) en el desarrollo de proyectos donde el estudiante asume el rol de profesional de QA dentro de un contexto similar al de la industria. A partir de un sistema real o verosímil (aplicación web, móvil o basada en APIs), los equipos formulan un plan de pruebas, definen su estrategia (niveles, tipos y técnicas de prueba), diseñan casos de prueba estructurados y reproducibles, y gestionan el ciclo de vida de los defectos. De este modo, los resultados de aprendizaje del curso (analizar requisitos, diseñar casos de prueba, seleccionar estrategias según contexto tecnológico y preparar pruebas automatizables) se evidencian en artefactos concretos: planes de prueba, suites de casos de prueba, reportes de defectos y scripts de automatización.

### **INVESTIGACIÓN**

En la asignatura de Ingeniería de Pruebas de Software, la investigación se concibe como un eje transversal que acompaña el desarrollo del proyecto y las experiencias de laboratorio, y no como una actividad aislada de “búsqueda de información”. Desde las primeras semanas, los estudiantes se enfrentan a situaciones problemáticas reales o simuladas (por ejemplo, especificaciones incompletas, defectos que llegan a producción, estrategias de prueba poco efectivas), a partir de las cuales formulan preguntas de investigación relacionadas con las prácticas de testing, las técnicas más adecuadas en distintos contextos tecnológicos y el uso de herramientas de apoyo. Estas preguntas orientan la revisión sistemática de fuentes bibliográficas, estándares internacionales, documentación técnica y casos de la industria, que luego se sintetizan en informes breves, mapas conceptuales o presentaciones enfocadas en justificar decisiones de prueba.

A lo largo del curso, la investigación se articula en tres niveles: conceptual, metodológica y aplicada. En el nivel conceptual, el estudiante contrasta definiciones, enfoques y terminología estándar de ingeniería de pruebas, consolidando un marco teórico que fundamenta su trabajo. En el nivel metodológico, analiza y compara diferentes técnicas (caja negra, caja blanca, pruebas estáticas, automatización, pruebas para web y móviles), identificando criterios de selección y limitaciones. Finalmente, en el nivel aplicado, integra estos hallazgos en su proyecto: diseña y ajusta estrategias de prueba, selecciona herramientas, justifica la priorización de casos, mide cobertura y calidad, y documenta lecciones aprendidas. El resultado esperado es que el estudiante desarrolle una actitud investigativa permanente, capaz de fundamentar sus decisiones como ingeniero de pruebas con evidencia técnica y académica, y no solo con intuiciones o prácticas rutinarias.

### **MEDIOS AUDIOVISUALES**

Para el mejor desarrollo de las actividades propuestas, se hará uso de: herramientas de comunicación síncrona como videoconferencias y asíncronas como foros; se acudirá al uso de la plataforma institucional Moodle con el fin de socializar material de estudio y lectura, efectuar la entrega de trabajos y tareas, además para la realización de algunas evaluaciones.

### EVALUACIÓN

#### EVALUACIÓN COLECTIVA

Se incentiva el trabajo colaborativo, para que los equipos de trabajo en sinergia puedan contar con los recursos tecnológicos necesarios para el desarrollo de la asignatura.

Para la evaluación se propone actividades que pueden ser realizadas de manera colectiva, se encuentran las siguientes:

- Talleres en clase, exposiciones, trabajos escritos: 30%
- Desarrollo y sustentación de los retos de la asignatura: 40%

#### EVALUACIÓN INDIVIDUAL

Se realizará al menos un examen parcial individual durante el semestre, el cual vale el 30% de la asignatura

### CONTENIDOS TEMÁTICOS CENTRALES

- **CAPÍTULO I: INTRODUCCIÓN**
- **CAPÍTULO II: FUNDAMENTOS DEL TESTING**
- **CAPÍTULO III: TIPOS DE PRUEBAS**
- **CAPÍTULO IV: METODOLOGÍAS DE PRUEBAS**
- **CAPÍTULO V: AGILE TESTING**
- **CAPÍTULO VI: TESTING WEB**
- **CAPÍTULO VII: AUTOMATIZACIÓN DE PRUEBAS**
- **CAPÍTULO VIII: TESTING MÓVIL**
- **CAPÍTULO IX: TESTING A BASES DE DATOS**
- **CAPÍTULO X: PRUEBAS DE API**

### LECTURAS MÍNIMAS

- Documentos y artículos relacionados con la asignatura, proporcionados por el docente.
- Capítulos sobre la bibliografía entregada.

### BIBLIOGRAFÍA

- Aniche, M. (2022). Effective Software Testing: A Developer's Guide. Manning Publications.
- Axelrod, A. (2018). Complete guide to test automation: Techniques, practices, and patterns for building and maintaining effective software projects. Apress.
- Graham, D., Van Veenendaal, E., Evans, I., & Black, R. (2019). Foundations of Software Testing: ISTQB Certification (4th ed.). Cengage Learning EMEA.
- Myers, G. J., Sandler, C., & Badgett, T. (2011). The Art of Software Testing (3rd ed.). John Wiley & Sons.
- Kelhini, F. K., & Mayhew, B. (2026). Hands-On Automated Testing with Playwright. Packt Publishing.
- Khorikov, V. (2020). Unit testing: Principles, practices, and patterns. Manning Publications.
- Mohan, G. (2022). Full Stack Testing: A Practical Guide for Delivering High Quality Software. O'Reilly Media.

- Serna M., E. (2021). Métodos Formales, Ingeniería de Requisitos y Pruebas del Software. Editorial Instituto Antioqueño de Investigación.
- Toledo, F. (2014). Introducción a las Pruebas de Sistemas de Información. (Disponible gratuitamente en versión digital).
- Winter, A., & Meaney, R. (2021). Team guide to software testability: Better software through greater testability. Self-published.
- International Software Testing Qualifications Board. (2023). Certified tester foundation level (CTFL), syllabus version 4.0. ISTQB

Nombre del docente responsable: **Docentes del área de software**