**Facultad de Ingeniería y Ciencias  
Escuela de Informática y Telecomunicaciones**

**PROGRAMA DE ASIGNATURA**Desarrollo Ágil y Aseguramiento de la Calidad (QA)

1. **Identificación de la asignatura:**

| Nombre de la Asignatura: Desarrollo Ágil y Aseguramiento de la Calidad (QA) | |
| --- | --- |
| Códigos: CIT-3610 | Créditos: 6 |
| Duración: Semestral | Ubicación en el plan de estudios: Semestre 9 |
| Requisitos: CIT-2312 Fund. de Ing. Y Arq. de software | |
| Sesiones cátedras semanales: 2 cátedras | |
| Sesiones de Ayudantía: 1 | |

1. **Descripción de la asignatura:**

El curso entrega conocimientos y habilidades fundamentales para aplicar metodologías ágiles en el desarrollo de software, integrando prácticas modernas de planificación, prototipado, testing y aseguramiento de la calidad. El enfoque está puesto en el trabajo colaborativo, el modelado funcional, el diseño centrado en el usuario y la integración continua. Los estudiantes desarrollarán un proyecto práctico iterativo, incorporando principios de mejora continua, automatización y mantenimiento sostenible del software.

1. **Resultados de aprendizaje:**
2. Implementa metodologías ágiles para resolver problemáticas de desarrollo de software, priorizando la entrega temprana de valor mediante la creación y validación iterativa de un producto mínimo viable.
3. Utiliza herramientas de modelado para representar procesos, requerimientos y diseño orientado a objetos.
4. Diseña interfaces centradas en el usuario, utilizando prototipos y pruebas de usabilidad.
5. Aplica estrategias de testing automatizado y QA, para asegurar la calidad del software.
6. Implementa prácticas de CI/CD para mantener ciclos de desarrollo eficientes y sostenibles.
7. Propone mejoras continuas en proyectos reales, mediante refactorización y análisis de deuda técnica.
8. Participa en equipos de trabajo, planificando, coordinando y ejecutando tareas con liderazgo y responsabilidad, comunicándose efectivamente y elaborando informes técnicos que reflejen procedimientos, resultados y análisis del trabajo realizado.
9. **Unidades Temáticas:**

* ***Unidad 1: Metodologías Ágiles de Desarrollo de Software***
  + *Manifiesto Ágil y valores clave*
  + *Scrum y Kanban: roles, artefactos y eventos*
  + *Diferencias con enfoques predictivos (cascada)*
* ***Unidad 2: Planificación y Seguimiento Ágil***
  + *User stories, estimación con puntos de historia*
  + *Burndown charts, tableros Kanban*
  + *Adaptación y gestión de cambios*
* ***Unidad 3: Modelamiento de Software Ágil***
  + *Casos de uso, diagramas de clases orientados a objetos*
  + *Modelado de procesos con BPMN*
* ***Unidad 4: Diseño de Interfaces y Experiencia de Usuario (UX)***
  + *Principios de diseño centrado en el usuario*
  + *Prototipos interactivos y pruebas de usabilidad*
* ***Unidad 5: Testing Automatizado y QA***
  + *Pruebas unitarias, de integración y aceptación*
  + *Estrategias de aseguramiento de calidad*
* ***Unidad 6: CI/CD y Evolución del Software***
  + *Integración continua y despliegue continuo*
  + *Automatización de pruebas y pipelines*
  + *Refactorización, gestión de deuda técnica, actualización tecnológica*

1. **Descripción general del método de enseñanza:**

Las clases se imparten en modalidad teórico-práctica. Se espera que el alumno adquiera los conocimientos técnicos y metodológicos necesarios mediante clases expositivas, y su aplicación a problemas realistas. Las clases presentan los principios, métodos y técnicas utilizadas en Ingeniería de Software. Para poner en práctica los conceptos aprendidos, se realizará un proyecto semestral en contextos reales de manera colaborativa con equipos formados por los alumnos.

A partir de las actividades antes mencionadas se desarrollará la capacidad para modelar formalmente sistemas informáticos. La disciplina de Ingeniería de Software tiene una relación estrecha con la habilidad de un ingeniero para reducir los riesgos del software. Planificar, analizar y diseñar son actividades orientadas a esos aspectos y serán reforzadas/evaluadas durante la ejecución del proyecto antes descrito y en las evaluaciones formales del curso.

1. **Descripción general de la modalidad de evaluación:**

Se contempla la realización de evaluaciones parciales (controles y trabajos), un proyecto semestral, dos pruebas solemnes de igual valor y un examen final. Si la nota asociada a la ejecución del proyecto semestral es inferior a 4.0, el alumno reprobará la asignatura con nota final igual a la nota obtenida en el proyecto.

Podrán eximirse aquellos alumnos que cumplan con los siguientes requisitos:

1. Nota de presentación mayor o igual a 5.0.
2. Solemne 1, Solemne 2 y Proyecto mayor o igual 4.0.
3. Todas las evaluaciones rendidas, incluyendo las dos solemnes, las 3 entregas del proyecto y todas las evaluaciones parciales.
4. **Bibliografía Básica Obligatoria:**
5. Beck, Kent. *Extreme Programming Explained: Embrace Change*, Addison-Wesley
6. Schwaber, Ken & Sutherland, Jeff. *Scrum Guide* (guía oficial)
7. Pressman, Roger. *Ingeniería del Software: Un enfoque práctico*, McGraw-Hill
8. Sommerville, Ian. *Ingeniería de Software*, 10ª ed.
9. Jez Humble & David Farley. *Continuous Delivery*, Addison-Wesley
10. Craig Larman. *UML y Patrones*, 3ª Ed.
11. Recursos del curso y documentación de herramientas (Figma, Jenkins, GitLab CI, etc.)

Elaborado por: Jonathan Frez

Fecha revisión: Mayo 2025

Fecha vigencia: Marzo 2026