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

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

1. **Identificación**
   * Nombre : Desarrollo Ágil y Aseguramiento de la Calidad (QA)
   * Código :
   * Créditos : 6
   * Duración : Semestral
   * Ubicación en plan de estudio: Semestre 2
   * Requisitos :
   * Sesiones semanales : 2 cátedras, 1 ayudantía
2. **Descripción del curso**

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**

Al finalizar el curso el alumno será capaz de:

1. Aplicar metodologías ágiles para planificar y gestionar proyectos de software.
2. Utilizar herramientas de modelado para representar procesos, requerimientos y diseño orientado a objetos.
3. Diseñar interfaces centradas en el usuario, utilizando prototipos y pruebas de usabilidad.
4. Aplicar estrategias de testing automatizado y QA para asegurar la calidad del software.
5. Implementar prácticas de CI/CD para mantener ciclos de desarrollo eficientes y sostenibles.
6. Proponer mejoras continuas en proyectos reales mediante refactorización y análisis de deuda técnica.
7. **Unidades Temáticas**
8. ***Metodologías Ágiles de Desarrollo de Software***
   1. ***Manifiesto Ágil y valores clave***
   2. ***Scrum y Kanban: roles, artefactos y eventos***
   3. ***Diferencias con enfoques predictivos (cascada)***
9. ***Planificación y Seguimiento Ágil***
   1. ***User stories, estimación con puntos de historia***
   2. ***Burndown charts, tableros Kanban***
   3. ***Adaptación y gestión de cambios***
10. ***Modelamiento de Software Ágil***
    1. ***Casos de uso, diagramas de clases orientados a objetos***
    2. ***Modelado de procesos con BPMN***
11. ***Diseño de Interfaces y Experiencia de Usuario (UX)***
    1. ***Principios de diseño centrado en el usuario***
    2. ***Prototipos interactivos y pruebas de usabilidad***
12. ***Testing Automatizado y QA***
    1. ***Pruebas unitarias, de integración y aceptación***
    2. ***Estrategias de aseguramiento de calidad***
13. ***CI/CD y Evolución del Software***
    1. ***Integración continua y despliegue continuo***
    2. ***Automatización de pruebas y pipelines***
    3. ***Refactorización, gestión de deuda técnica, actualización tecnológica***
14. **Metodología**

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. **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. En caso contrario, la nota de presentación se calcula a través de la siguiente formula.

Si promedio solemnes es mayor o igual a 4.0:

Nota de Presentación = (20% Solemne 1 + 20% Solemne 2 + 20% Proyecto+ 10% Parciales)/0.7

Si promedio solemnes es menor a 4.0:

Nota de Presentación = (30% Solemne 1 + 30% Solemne 2 + 10% Parciales)/0.7

Nota Final= 70% Nota de Presentación + 30% Examen.

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**

 Beck, Kent. *Extreme Programming Explained: Embrace Change*, Addison-Wesley

 Schwaber, Ken & Sutherland, Jeff. *Scrum Guide* (guía oficial)

 Pressman, Roger. *Ingeniería del Software: Un enfoque práctico*, McGraw-Hill

 Sommerville, Ian. *Ingeniería de Software*, 10ª ed.

 Jez Humble & David Farley. *Continuous Delivery*, Addison-Wesley

 Craig Larman. *UML y Patrones*, 3ª Ed.

 Recursos del curso y documentación de herramientas (Figma, Jenkins, GitLab CI, etc.)

**PAUTAS ETICAS BASICAS**

El plagio es el uso de las ideas o trabajo de otra persona sin el adecuado consentimiento. El plagio puede ser intencional o no. El plagio intencional es el claro intento de hacer pasar el trabajo o ideas ajenas como el suyo propio para su beneficio. El plagio no intencional puede ocurrir si Ud. no conoce el mecanismo adecuado de referenciar la fuente de sus ideas e información. Si no está seguro de los métodos aceptados para referenciar, debería consultar con su profesor, tutor o personal de biblioteca.

El plagio comprobado es una actitud que puede resultar en severas sanciones disciplinarias y/o en la exclusión de la Universidad (Artículo 44, Reglamento del Estudiante de Pregrado).

Elaborado por: Jonathan Frez

Fecha revisión: Marzo de 2025

Fecha vigencia: Marzo de 2026