Facultad de Ingeniería y Ciencias
Escuela de Informática y Telecomunicaciones

DESCRIPTOR 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: CIT3610	Créditos: 6
Duración: Semestral	Ubicación en el plan de estudios: Semestre 9
Requisitos: CIT2312 Fund. de Ing. Y Arq. de software	
Sesiones cátedras semanales: 2 cátedras	
Sesiones de ayudantía: 1	

2. 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.

3. Resultados de Aprendizaje:

1. 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.
2. Utiliza herramientas de modelado para representar procesos, requerimientos y diseño orientado a objetos.
3. Diseña interfaces centradas en el usuario, utilizando prototipos y pruebas de usabilidad.
4. Aplica estrategias de testing automatizado y QA, para asegurar la calidad del software.
5. Implementa prácticas de CI/CD para mantener ciclos de desarrollo eficientes y sostenibles.
6. Propone mejoras continuas en proyectos reales, mediante refactorización y análisis de deuda técnica.
7. 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.

4. 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

5. Descripción general del método de enseñanza:

Las clases se imparten en modalidad teórico-práctica. Se espera que el/la estudiante 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 Desarrollo Ágil y Aseguramiento de la Calidad (QA). Para poner en práctica los conceptos aprendidos, se realizará un proyecto semestral en contextos reales de manera colaborativa con equipos formados por el estudiantado.

A partir de las actividades antes mencionadas se desarrollará la capacidad para modelar formalmente sistemas informáticos. El Desarrollo Ágil y Aseguramiento de la Calidad 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.

6. Descripción general de la modalidad de evaluación:

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

Podrán eximirse los/las estudiantes 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.

Fórmula de evaluación:

- Nota de Presentación = (20% Solemne 1 + 20% Solemne 2 + 20% Proyecto + 10% Parciales)/0.7
- Si promedio solemnes es menor a 4.0, entonces:
  - Nota de Presentación = (30% Solemne 1 + 30% Solemne 2 + 10% Parciales)/0.7
  - Nota Final = 70% Nota de Presentación + 30% Examen.

7. Planificación de Sesiones:

La planificación de las sesiones dependerá del calendario académico vigente al momento de dictar la asignatura. En cualquier caso, cada docente que la imparta deberá informar a la Escuela (antes del comienzo del semestre) la planificación de sus actividades, evaluaciones, y mecanismos de diversificación de las mismas.

Esta información será informada al estudiantado durante la primera semana de clases.

8. Bibliografía Básica Obligatoria:

1. Beck, Kent. Extreme Programming Explained: Embrace Change, Addison-Wesley Professional, 2004.
2. Schwaber, Ken & Sutherland, Jeff. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org, 2020. [Versión más reciente de la guía oficial]
3. Pressman, Roger. Ingeniería del Software: Un enfoque práctico, 8ª ed. McGraw-Hill Education, 2014.
4. Sommerville, Ian. Ingeniería de Software, 10ª ed. Pearson Educación, 2015.
5. Jez Humble & David Farley. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional, 2010.
6. Craig Larman. UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al desarrollo iterativo, 3ª ed. Pearson Prentice Hall, 2008.
7. Recursos del curso y documentación de herramientas (Figma, Jenkins, GitLab CI, etc.)

PAUTAS ÉTICAS BÁSICAS

El aula es un espacio donde los intercambios buscan generar un clima que potencie el aprendizaje, basado en el respeto y el buen trato. Las diferencias, tanto entre estudiantes, como entre estudiante y docentes, deben abordarse desde este marco de respeto.
La universidad cuenta con dos reglamentos importantes de conocer:

- Reglamento de Convivencia Estudiantil
- Normativa de Prevención y Sanción de Acciones de Discriminación, Violencia Sexual y/o de Género.

Puedes consultar los reglamentos aquí: https://www.udp.cl/universidad/reglamentos-y-politicas/

Así también, 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
Revisado por: Jonathan Frez
Fecha revisión: Octubre 2025
Fecha vigencia: Marzo 2026