# Acuerdos del curso

Objetivo: Conocer y aplicar las actividades fundamentales para el desarrollo de productos de software de calidad en equipos interdisciplinarios.

Temas:

* Introducción a la Ingeniería de Software
* SCM y versionado en Git
* Ingeniería de requerimientos
* Interfaz de usuario
* Codificación y test unitario
* Test funcional y reporte de issues

Habilidades transversales: trabajo en equipo, comunicación interdisciplinaria, foco en la calidad, autogestión y proactividad.

El curso requiere conocimientos previos de programación, sistemas operativos y tecnologías para el desarrollo de software. Consultar el documento de requisitos previos y recursos de repaso.

## Metodología

Buscamos un aprendizaje integrado teórico/práctico, de conocimientos y habilidades que sean útiles en el contexto profesional. Utilizamos como desafíos problemas abiertos y tecnologías del estado del arte. Intentamos crear un contexto de aprendizaje donde los estudiantes en conjunto con el equipo docente logren los objetivos planteados.

Al comienzo del curso publicaremos un cronograma y referencias para cada clase. Las evaluaciones (ejercicios individuales, proyecto e investigación de tecnología) están alineadas con los objetivos de aprendizaje.

Los docentes estarán disponibles para responder consultas a lo largo del curso, durante la clase o en el foro de discusión del grupo. No se responden consultas por chat.

## Compromiso de trabajo

Se espera por parte de todos: puntualidad, participación y dedicación exclusiva al tema de clase. Este código de conducta aplica a estudiantes y docentes con el objetivo de crear un ambiente de trabajo respetuoso y colaborativo.

Para lograr los objetivos de aprendizaje y aprobar el curso, se espera por parte de los estudiantes:

* Configuración de los entornos de desarrollo y de las herramientas de apoyo. Creación de una cuenta GitHub para las entregas de ejercicios y proyecto.
* Realización de tareas individuales dentro y fuera de clase. Las tareas consisten en la realización de guías, ejercicios y lecturas que se consideran necesarias para lograr los objetivos de aprendizaje.
* Entrega de ejercicios de práctica previa y realización de evaluaciones presenciales en clase (cuestionarios y ejercicios de desarrollo de software).
* Trabajo en equipo para el proyecto. Realizar entregas en checkpoints de avance y de informes académicos. Se debe justificar el proceso aplicado y dar evidencia del trabajo de los integrantes en todas las actividades.
* Realizar una defensa oral del proyecto en caso de que el equipo sea seleccionado.
* Investigación de tecnología aplicada en el proyecto. Elaboración de una ficha técnica y presentación en clase.

Al realizar estas actividades se espera que los estudiantes respeten el compromiso de honestidad académica del curso.

## Tecnología y resolución de problemas

Planteamos como objetivos del curso incorporar habilidades para el desarrollo de proyectos de software en equipos, manejo de tecnología y resolución de problemas emergentes. Se espera que los estudiantes sean proactivos en la instalación, uso y búsqueda de soluciones sobre tecnología. Durante el curso se brindará una ayudantía semanal de apoyo. Consultar en el documento de requisitos previos las tecnologías usadas en el curso.

## Evaluación

1. Ejercicios individuales
   * Cuestionarios - 15 puntos
   * Ejercicios de práctica – entrega previa obligatoria
   * Ejercicios de evaluación - 25 puntos
2. Proyecto (obligatorio)
   * Checkpoints de avance
   * Entrega 1 - 20 puntos
   * Entrega 2 - 35 puntos
3. Investigación de tecnología - 5 puntos

Los estudiantes que no obtengan 70 puntos a lo largo del curso pueden realizar una segunda instancia de entrega 2 del obligatorio. La segunda instancia utiliza la misma letra y rúbrica, se realiza una nueva entrega del mismo proyecto en forma individual.

El curso no tiene examen. Los estudiantes que abandonen luego del comienzo tendrán 0 en la escolaridad.

Las entregas deben realizarse por GitHub y por el sistema de gestión de la Universidad (informes académicos). No se pueden realizar cambios en el repositorio GitHub luego de la fecha y hora de cierre del sistema de gestión. Si no se cumple con estos requisitos, no se corregirá la entrega.