

## **PROGRAMA de Desarrollo de Aplicaciones**

**Carrera/s:** Tecnicatura Universitaria en Programación Informática

**Asignatura:** Desarrollo de Aplicaciones

**Núcleo al que pertenece:** Avanzado Obligatorio

**Profesor/les:** Cristian López, Juan Pablo DelPino, Pablo Nieloud

**Asignaturas previas necesarias para favorecer el aprendizaje:**

Construcción de Interfaces de Usuario – Elementos de Ingeniería de Software – Estrategias de Persistencia.

### **Objetivos:**

Que el estudiantado:

- Lleve a cabo la construcción completa de una aplicación mediana, partiendo desde una versión preliminar de los requerimientos y llegando a una aplicación que pueda ponerse en producción.
- Aplique en una experiencia concreta de desarrollo los conocimientos sobre ciclo de desarrollo y metodologías adquiridos en Elementos de Ingeniería de Software.
- Comprenda los beneficios de respetar buenos principios y prácticas adquiridos en materias anteriores en un proyecto de desarrollo de software, mediante la experiencia concreta en la utilización e integración de los mismos.
- Tenga una noción de que es una arquitectura de software, y que consecuencias trae trabajar con una determinada arquitectura en el desarrollo.
- Tenga un conocimiento razonable acerca de herramientas tecnológicas, metodológicas y de diseño de software adecuadas para el desarrollo grupal de proyectos de software.
- Tenga nociones acerca del análisis de eventuales problemas de performance, la pertinencia de abordarlos en los momentos y en la medida adecuados, y la relevancia de una visión global al respecto incluyendo tanto cuestiones de configuración como cuestiones de código.
- Tenga nociones acerca de los desafíos y tareas inherentes al mantenimiento de una aplicación posterior a su primera puesta en producción.

### **Contenidos mínimos:**

- Herramientas metodológicas y conceptuales para trabajo en grupo.
- División de tareas planeando reunión de los resultados.

- Aprovechamiento de conceptos de objetos, citamos como posibles ejemplos: interfaces como forma de coordinar la tarea de distintas personas o grupos, mock objetos para simular los objetos de otros grupos, etc.
- Herramientas tecnológicas para trabajo en grupo: repositorios de código y de biblioteca, automatización de procesos involucrados en el desarrollo y despliegue.
- Aplicación concreta de las ideas de desarrollo iterativo.
- Nociones de arquitectura de software: que es una decisión de arquitectura, impacto en el desarrollo de trabajar dentro de cierta arquitectura. Diversidad de variantes en arquitecturas de software.
- Arquitectura de sistema. Variantes: stand-alone, servidor Web dinámico,
- servidor de aplicaciones, cliente-servidor. Soporte de ejecución concurrente. Implicancias en el diseño de software y en las tareas habituales de desarrollo, pertinencia de prácticas que agilicen el desarrollo bajo distintas arquitecturas de sistemas.
- Problemas de performance: instancias adecuadas para su análisis,
- posibilidad de problemas asociados a configuraciones defectuosas.
- Despliegue de una aplicación en distintos entornos: tareas que conlleva.

**Carga horaria semanal:** 6 hs

**Carga horaria total:** 108 horas (Presenciales: 68 horas / Semipresenciales: 40 horas)

**Bibliografía obligatoria:**

- Growing object-oriented software guided by tests, Addison-Wesley
- Clean code: A Handbook of Agile Software Craftsmanship, PRENTICE HALL
- Patterns of Enterprise Application Architecture, Addison-Wesley Professional
- Continuous Integration: Improving Software Quality and Reducing Risk, Addison-Wesley Professional; 1 edition

**Bibliografía de consulta:**

- Continuous Integration: Improving Software Quality and Reducing Risk, Addison-Wesley Professional;
- Enterprise Integration Patterns: Designing, Building, and Deploy in Messaging Solutions, Addison-Wesley Professional;
- Continuous Delivery: Reliable Software Releases through Build, Test and Deployment Automation, Addison-Wesley Professional
- The Clean Coder: A Code of Conduct for Professional Programmers, Prentice Hall

- Pragmatic programmer: from journeyman to master, Addison Wesley

### **Organización de las clases:**

La materia se compone de clases teóricas y prácticas con mucho foco en trabajo en clase. Al mismo tiempo, por la modalidad de la materia, se trabaja en laboratorios donde el trabajo en equipo y el compartir soluciones serán Relevantes.

### **Modalidad de evaluación:**

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18)

Existen 3 entregas de un trabajo práctico semestral (una entrega acumulativa cada 3 o 4 semanas aproximadamente) y una exposición sobre un paper a elección. La materia se aprueba con las 4 entregas del trabajo y alcanzando como mínimo el 70% de correctitud de los mismos. El aporte en clase, colaboración y participación, también serán tenidos en cuenta.

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial