



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## Especialización en Ingeniería de Software Modalidad a distancia

### Tópicos de Ingeniería de Software II

Año 2025  
Plan de Estudios 2021

Duración: 108 hs. Totales.

Cantidad de horas presenciales/VC: 40hs

Cantidad de horas de actividades en línea y de  
trabajo final: 68hs.

**Docente Responsable: Dr. Urbietta, Matías**

#### **Docentes Tutores:**

**Dr. Díaz Pace, Jorge A.**

(Académico/Tecnológico)

**Dr. Bibbó, Luis Mariano.**

(Académico/Tecnológico)

#### **Tutores:**

**Pizarro, María Alejandra** (Administrativo)

### **OBJETIVOS GENERALES:**

Brindar a los asistentes técnicas y conceptos avanzados del desarrollo de Software, que permiten construir software robusto, mantenible, extensible y reutilizable. Se abordará el diseño de software desde una perspectiva arquitectural y de atributos de calidad,. Se abordarán patrones de arquitectura, con un foco en el ámbito de aplicaciones Web. Por otro lado se introducirán conceptos de inteligencia artificial y cómo puede aplicarse al campo de la ingeniería de software.

### **COMPETENCIAS A DESARROLLAR EN RELACIÓN CON EL OBJETIVO DE LA CARRERA**

C.1- Manejar y aplicar tecnologías actuales para el desarrollo de sistemas de software, incluyendo métodos, lenguajes, arquitecturas, frameworks y herramientas.

C.3- Gestionar, planificar y controlar proyectos de software de distinta envergadura.

C.4- Definir parámetros de calidad tanto interna como externa de un producto software, y establecer procesos de evaluación y mejora que atiendan la satisfacción de todos los



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

involucrados (el cliente, los usuarios y su experiencia, y el equipo de desarrollo).

C.6- Tener capacidad de analizar el estado del arte en los distintos aspectos de la ingeniería de software, así como producir conocimiento científico en el área.

### **CONTENIDOS MÍNIMOS:**

- Nociones de arquitectura de software y atributos de calidad
- Arquitecturas web y diseño de aplicaciones con microservicios.
- Introducción a la Inteligencia Artificial, y su impacto sobre la Ingeniería de Software.
- Conceptos de Infraestructura como código, Contenerización y despliegues.

### **PROGRAMA**

#### **Arquitectura y Atributos de Calidad**

- Definición de arquitectura. Rol en el desarrollo de software.
- Atributos de calidad como conductores del diseño arquitectónico.
- Estilos arquitectónicos más comunes.
- Captura de decisiones arquitectónicas.
- Vistas de módulos, componentes y conectores, y despliegue.

#### **Arquitecturas Web y Microservicios**

- Frameworks tradicionales multicapa. Model-View-Controller. Principios.
- Frameworks de aplicaciones Single-Page.
- Arquitecturas basadas en Microservicios.
- Nociones de infraestructura para puesta en producción.

#### **Introducción a la Inteligencia Artificial.**

- Fundamentos, noción de modelo. Aprendizaje automático o machine learning. Aprendizaje supervisado vs. Aprendizaje no supervisado. Árboles de decisión. Deep Learning (Redes Neuronales Artificiales).
- IA generativa, Introducción a LLMs, prompting, RAGs, Workflows y agentes.
- Aplicaciones de la Inteligencia Artificial en el campo de la Ingeniería de Software.
- Ciclo de vida de los modelos de Inteligencia Artificial y MLOps

#### **Infraestructura como código**

- Fundamentos. Infraestructura as Code, Contenedores, infraestructura.
- Diseño de de infraestructura
- Esquema de despliegue



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## **Blockchain, smartcontracts, DAPPS**

- Fundamentos. Conceptos básicos de criptografía.
- Criptomonedas
- Smartcontracts
- DAPPS

## **MODALIDAD DE EVALUACIÓN Y ACREDITACIÓN**

La evaluación de la materia es a través de trabajos prácticos parciales y un trabajo práctico final integrador. La calificación final del alumno será en base a los trabajos parciales y la entrega final. Los trabajos prácticos parciales son de pequeña envergadura y tienen una función de poner rápidamente en práctica los conocimientos que se van abordando. Estos son realizados y aprobados durante la cursada de la materia.

El trabajo final integrador combina los conceptos presentados durante el curso para el diseño y prototipado de una solución al enunciado provisto por los profesores. Una vez finalizado el curso, el alumno cuenta con 2 meses para presentar el trabajo.

Cualquier evidencia de plagio en los trabajos entregados será considerado causal de desaprobación.

## **RECURSOS Y MATERIALES DE ESTUDIO**

El curso propone 15 encuentros sincrónicos en total, a desarrollarse en su mayoría a través de videoconferencia. Se requiere 80% de asistencia a los encuentros sincrónicos, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.

Los materiales de estudio son:

- Textos digitales: textos de lectura de referencia en en las temáticas tratadas durante el curso. Los textos de referencia son recuperados de la biblioteca del postgrado, revistas y/o de repositorios.
- Material preparado por los docentes del curso.
- Presentaciones digitales y materiales multimediales sobre el tema (de producción propia)

## **ACTIVIDADES EXPERIMENTALES Y DE INVESTIGACION PLANIFICADAS PARA LA APROPIACIÓN DE LOS SABERES Y LA EVALUACIÓN**

### Actividades prácticas:

Desarrollo de trabajos prácticos parciales luego de cada eje temático de la materia. Estos trabajos están pensados para ser iniciados en clases prácticas estilo taller, apuntando a una puesta en común, que luego continuará individualmente cada alumno.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

**Actividad 1:** luego del dictado de los primeros temas, se trabaja sobre un caso de estudio para que los alumnos puedan realizar un análisis de sus principales atributos de calidad y un (re-)diseño arquitectural del mismo, para luego implementarlo sobre alguna de las tecnologías a utilizar en el curso (por ej., Python). Además, se trabajará sobre la generación de diagramas de arquitectura basados en UML.

**Actividad 2:** se presentarán conceptos de microservicios que permitan exponer funcionalidades del sistema. Se introducen conceptos de contenerización e Infraestructure as code para el empaquetado y despliegue de los servicios en diferentes configuraciones de infraestructura. Se analizarán problemas de balanceo de carga, alta disponibilidad y diagnóstico de problemas.

**Actividad 3:** mediante esta actividad el alumno pondrá en práctica los conceptos teóricos acerca de cómo implementar y entrenar un modelo de Inteligencia Artificial a través de un framework. Se busca que los estudiantes comprendan los conceptos de "modelo", y las diferencias entre "entrenamiento" e "inferencia". Los estudiantes aprenderán a ajustar un modelo para realizar el entrenamiento y utilizarán diferentes métricas para medir su performance.

**Trabajo Final:** Consiste en la aplicación de los conceptos expuestos en las diferentes actividades para resolver un problema presentado por la cátedra.

## BIBLIOGRAFÍA BÁSICA

- Cervantes, H., Kazman, R. Designing Software Architectures: A Practical Approach. Addison-Wesley. 2016
- Keeling, M. (2017). Design it!: From Programmer to Software Architect. Estados Unidos. Pragmatic Bookshelf.
- Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Addison Wesley, 1994.
- S. Russell y P. Norvig. Artificial Intelligence. A Modern Approach. Prentice Hall, 3ra edición. 2010.
- Infrastructure as Code , MANAGING SERVERS IN THE CLOUD, Kief Morris, SBN: 978-1-491-92435-8
- Huyen, C. (2025). AI Engineering: Building Applications with Foundation Models. Estados Unidos, O'Reilly Media.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## BIBLIOGRAFÍA COMPLEMENTARIA

- Shai Shalev-Shwartz y Shai Ben-David. (Ed.). (2014). Understanding Machine Learning: From Theory to Algorithms. New York. United State: Cambridge University Press.
- Antonopoulos, A. M. (2014). Mastering Bitcoin: Unlocking Digital Cryptocurrencies. Estados Unidos: O'Reilly Media.
- Antonopoulos, A. M., D., G. W. P. (2018). Mastering Ethereum: Building Smart Contracts and DApps. Alemania: O'Reilly Media.
- Morris, K. (2025). Infrastructure as Code: Designing and Delivering Dynamic Systems for the Cloud Age. (n.p.): O'Reilly Media.
- Hafner, R. (2025). Terraform in Depth: Infrastructure as Code with Terraform and OpenTofu. Estados Unidos: Manning.
- Alammar, J., Grootendorst, M. (2024). Hands-on Large Language Models: Language Understanding and Generation. Estados Unidos, O'Reilly Media.
- Bass, L., Lu, Q., Weber, I., Zhu, L. (2025). Engineering AI systems: Architecture and DevOps Essentials. Estados Unidos. Addison-Wesley Professional.