



*Universidad Tecnológica Nacional*  
*Facultad Regional Buenos Aires*

## PROGRAMA ANALÍTICO DE ASIGNATURA

**DEPARTAMENTO:** Ingeniería en Sistemas de Información

**CARRERA:** Ingeniería en Sistemas de Información

**NOMBRE DE LA ACTIVIDAD CURRICULAR:** Tecnologías Avanzadas en la Construcción de Software

**Año Académico:** 2023

**Área:** Desarrollo de Software

**Bloque:** Electivas

**Tipo:** Electiva

**Modalidad:** Cuatrimestral

### Cargas horarias totales:

<i>Horas reloj</i>	<i>Horas cátedra</i>	<i>Horas cátedra semanales</i>
72	96	6

## FUNDAMENTACIÓN

En las últimas décadas las Tecnologías disponibles en el Desarrollo de Software crecieron de forma exponencial generando un alto impacto en la industria y uso de herramientas. Actualmente, resulta indispensable que los egresados de la carrera dispongan de los conocimientos necesarios y suficientes para acceder a un abanico de opciones que se presentan y utilizan en el mundo laboral. Es por ello que la presente electiva, se propone acercar a los estudiantes a transitar estas experiencias en clave a las tecnologías, herramientas y Desarrollo de Software para que, cuando integren un desafío laboral, dispongan del conocimiento para poder elegir las mejores opciones al momento de desarrollar software para la industria según el escenario en el que esté inserto.

## OBJETIVOS

- Identificar las tecnologías utilizadas en la actualidad para la construcción de software.
- Reconocer las tecnologías que podrían ser utilizadas en el futuro en clave a la construcción de software.
- Distinguir las tendencias actuales y futuras del mercado de desarrollo para la construcción de ideas propias que permitan aprovechar las herramientas y conceptos más útiles en cada caso.



*Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires*

- Distinguir tipos de tecnologías o frameworks en el proceso de desarrollo.
- Reconocer problemas que podrían ser resueltos por herramientas preexistentes.
- Distinguir tipos de herramientas comerciales y código abierto existentes en el mercado
- Seleccionar una herramienta a partir de la comprensión de sus ventajas y desventajas, y la relación de dichas características con los requerimientos funcionales y no funcionales del producto de software en desarrollo.
- Utilizar nuevas herramientas y frameworks seleccionados que impliquen la combinación de distintas formas de análisis de las mismas.
- Identificar las bases del funcionamiento técnico y conceptual de herramientas de construcción de software para optimizar su valor.
- Aplicar conceptos de diseño en la construcción de distintas herramientas.
- Identificar la complejidad, funcionamiento y arquitectura de las herramientas más utilizadas para abolir la distancia ficticia que se suele establecer entre el software que se puede adquirir en el mercado internacional y el que el propio profesional puede construir con un equipo.

## **CONTENIDOS**

### **Contenidos analíticos**

#### **Unidad Temática 1: Introducción a la construcción de software.**

Historia. Requerimientos no funcionales y cualidades de software. Visión del rol del Ingeniero en Sistemas y características a observar durante todos los temas que veamos en la asignatura.

#### **Unidad Temática 2: Comunicación entre aplicaciones.**

Tecnologías para comunicar aplicaciones. REST/SOAP ventajas y desventajas.

#### **Unidad Temática 3: Arquitectura web y frontend.**

Tecnologías para construir aplicaciones web e interfaces de usuario. Estado del arte. Diferencias entre frameworks más comúnmente usados. Aspectos a considerar a la hora de evaluar una tecnología de frontend.

#### **Unidad Temática 4: Infraestructura.**

Infraestructura donde alojar las aplicaciones. Conceptos de sesión y balanceo de carga. Infraestructura tradicional y en la nube. Diferentes esquemas de infraestructura en la nube. Service Mesh.

#### **Unidad Temática 5: Virtualización y contenedores.**

Concepto de máquina virtual y contenedor. Docker. Usos y buenas prácticas.

#### **Unidad Temática 6 -Persistencia.**



*Universidad Tecnológica Nacional*  
*Facultad Regional Buenos Aires*

NoSQL y diferencia con bases de datos relacionales. Tipos de bases NoSQL y usos. Teorema CAP. ACID vs BASE. Características del contexto a tener en cuenta a la hora de elegir una tecnología de persistencia.

#### **Unidad Temática 7: Microservicios.**

Concepto de microservicios y diferencia con monolitos. Características del contexto a evaluar a la hora de decidir una arquitectura. Metodología de refactor iterativo incremental.

#### **Unidad Temática 8: SRE.**

Site Reliability Engineering. Concepto y buenas prácticas. Objetivos del área y propuestas de organización del trabajo. Objetivos e indicadores.

### **BIBLIOGRAFÍA OBLIGATORIA**

- Bauer, C. & King, G. (2007). Java persistence with Hibernate. Greenwich, Conn. Ed. Manning.
- Eckel, B. (2006). Thinking in Java. Upper Saddle River, NJ. Ed. Prentice Hall.
- Fowler, M. (1997). Analysis patterns : reusable object models. Menlo Park, Calif. Ed. Addison Wesley.
- Fowler, M. (2003). Patterns of enterprise application architecture. Boston. Ed. Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). Design patterns: elements of reusable object-oriented software. Reading, Mass. Ed. Addison-Wesley.
- Horstmann, C. & Cornell, G. (2008). Core Java. Upper Saddle River, NJ. Ed. Prentice Hall/Sun Microsystems Press.
- Hunt, A. & Thomas, D. (2000). The pragmatic programmer : from journeyman to master. Reading, Mass. Ed. Addison-Wesley.
- Kan, S. (2003). Metrics and models in software quality engineering. Boston. Ed. Addison-Wesley.
- Lee, K. (2006). The buildmeister's guide : how to design and implement the right software build and release process for your environment. Ed. Buildmeister Books.
- Massol, V., Brien, T. & Loukides, M. (2005). Maven : a developer's notebook. Sebastopol, CA. Ed. O'Reilly.
- Beyer, B., Jones, C., Petoff, J. & Murphy, N. (2016). Site reliability engineering : How Google runs production systems. Sebastopol. Ed. O'Reilly Media.
- Jamsa, K. (2013). Cloud computing : SaaS, PaaS, IaaS, virtualization, business models, mobile, security, and more. Burlington, MA. Ed. Jones & Bartlett Learning.
- Khatri, A., Khatri, V., Nirmal, D., Pirahesh, H. & Herness, E. (2020). Mastering Service Mesh. City. Ed. Packt Publishing.
- Khatri, A., Khatri, V., Nirmal, D., Pirahesh, H. & Herness, E. (2020). Mastering Service Mesh. City. Ed. Packt Publishing.
- Newman, S. (2019). Monolith to microservices : evolutionary patterns to transform your monolith. Sebastopol, CA. Ed. O'Reilly Media.



*Universidad Tecnológica Nacional*  
*Facultad Regional Buenos Aires*

- Nickoloff, J. & Kuenzli, S. (2019). Docker in action. Shelter Island, NY. Ed. Manning Publications, Co.
- Sadalage, P. & Fowler, M. (2013). NoSQL distilled : a brief guide to the emerging world of polyglot persistence. Upper Saddle River, NJ. Ed. Addison-Wesley.

## **CORRELATIVAS**

### Para cursar y rendir

- Cursadas:
  - Análisis de Sistemas de Información
  - Sintaxis y Semántica de los Lenguajes
  - Paradigmas de Programación