

UNTREF

UNIVERSIDAD NACIONAL
DE TRES DE FEBRERO

Diplomatura Superior en Ingeniería de Software Continua

Ciclo: posgrado

Modalidad: a distancia, un encuentro semanal

Inicio: 20 de septiembre del 2023

Duración: 2 trimestres



Ingeniería de Software Continua

La mayoría de las organizaciones se encuentran hoy atravesadas por la necesidad de desarrollar o contratar servicios de software para sobrevivir en un marco de profunda digitalización de sus servicios y operaciones. Lograr formar y liderar equipos de producto capaces de desarrollar y operar software es un gran desafío para los profesionales de hoy, tanto en lo tecnológico como en la gestión.

La diplomatura presenta una formación que balancea las tecnologías de desarrollo y operación de software con las prácticas continuas que los métodos ágiles y devops aportan para la generación de capacidades de entrega continua de software, necesarias para aumentar la capacidad de las organizaciones y permitirles desarrollar todo su potencial..

Los egresados de la carrera contarán con el marco teórico y la experiencia práctica en el diseño e implementación de procesos de entrega y operación continua, además del desarrollo de software, diseño de arquitectura y la gestión de proyectos. Realizarán trabajos que incluyan preferentemente las tecnologías y necesidades propias de su contexto laboral y finalizarán con una presentación pública de alguno de los trabajos realizados.

Objetivos

- Formar especialistas en el desarrollo, implementación y operación de soluciones basadas en software.
- Brindar conocimiento sobre el estado del arte en métodos y prácticas de desarrollo y entrega de software cubriendo tanto los fundamentos de la disciplina como también las últimas tendencias.
- Proveer a los estudiantes de una visión holística de desarrollo y entrega del software abarcando problemáticas de tecnología y de gestión.

Cursos

Entrega Continua (Software Delivery)

Docentes: Esp. Ing. Nicolás Paez y Lic. Diego Marcet

Procesos, prácticas y herramientas tecnológicas necesarios para la entrega continua de software, junto con su impacto en la capacidad organizacional. Incluye la mirada técnica y de gestión, desde implementación de pipelines hasta indicadores de gestión.

Objetivos:

- Entender el impacto de las capacidades de Software Delivery en la performance del negocio.
- Entender el conjunto de prácticas técnicas y de gestión que mejoran la performance de Software Delivery.
- Conocer posibles estrategias para la adopción e implementación de las mencionadas prácticas.

Contenidos mínimos: Ciclo de desarrollo y entrega de software. Medición de la performance de software delivery en una organización y su relación con la performance organizacional. El rol de IT y la capacidad de Software Delivery. Las distintas dimensiones del Software Delivery. Las prácticas técnicas y organizacionales; y sus relaciones. El aporte de Lean, DevOps y Ágiles. Posibles caminos de adopción. Estrategias de deployment: canaries, blue-green, etc. Pipeline de delivery. Métricas de entrega de software.

Bibliografía de referencia:

- Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. Nicole Forsgren PhD, Jez Humble, Gene Kim. IT Revolution, 2018
- Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Jez Humble, David Farley. Pearson Education, 2010
- The DevOps Handbook: How to Create World-class Agility, Reliability, and Security in Technology Organizations. Gene Kim, Patrick Debois, John Willis, Jez Humble. IT Revolution Press, 2016
- Team Topologies: Organizing Business and Technology Teams for Fast Flow. Matthew Skelton, Manuel Pais. IT Revolution Press, 2019

Ingeniería de Software Moderna

Docente: Mg. Ing. Carlos Fontela

La visión moderna de la ingeniería de software, como disciplina empírica que busca hallar soluciones eficientes y económicas a los problemas del desarrollo. Desarrollo como proyecto y como producto, sobre la base de un proceso incremental de aprendizaje a lo largo de las actividades, iterativas o continuas, de planificación, diseño, construcción y liberación de productos. Incluye las prácticas ágiles de descubrimiento y desarrollo de productos, desde BDD hasta temas de gestión de equipos y personas.

Objetivo:

- Comprender el ciclo de vida de desarrollo de software.
- Apropiarse de prácticas concretas para el desarrollo efectivo de software.
- Integrar una mirada moderna de producto y proceso del desarrollo de software.
- Incorporar prácticas de mejora continua a nivel equipo y organización.


Contenidos mínimos: Producto y Proceso. Ciclos de vida iterativos para el desarrollo de software. Planning Multi-dimensional, Complejidad. Entrega vs. Descubrimiento de producto, Scrum y XP, Gestión de riesgos, Estimación. Prácticas de mejora continua en equipos y organizaciones (retrospectivas). Estrategias de calidad y pruebas. BDD. Pruebas manuales y automatizadas. Desarrollo y construcción de un producto de software. Métricas. Perspectivas futuras: generación y validación de código con inteligencia artificial.

Bibliografía de referencia:

- The Art of Agile Development (2° edición). James Shore, Shane Warden. O'Reilly Media, 2021
- Software Engineering at Google: Lessons Learned from Programming Over Time. Titus Winters, Tom Manshreck, Hyrum Wright. O'Reilly Media, 2020
- Discovery: Explore Behavior Using Examples. Gaspar Nagy, Seb Rose. CreateSpace Independent Publishing Platform, 2018.
- Formulation: Explore Behaviour Using Examples. Gaspar Nagy, Seb Rose. CreateSpace Independent Publishing Platform, 2018.

Diseño y Evolución de Arquitecturas de Software

Docentes: Dr. Andrés Díaz Pace, Dr. Ing. Diego Fontdevila



Teoría y práctica de los procesos de diseño iterativo de arquitecturas de software, factor clave de éxito para la evolución de productos digitales. Partiendo del conocimiento clásico de arquitecturas, realiza un recorrido por tecnologías cloud y microservicios, hasta llegar a alternativas utilizando IA.

Objetivos:

- Desarrollar en forma teórica y práctica los conceptos de arquitectura de software
- Reconocer el impacto organizacional y estratégico de la arquitectura de software
- Incorporar tácticas y herramientas de diseño
- Ubicar a la tecnología en el espacio de la arquitectura de software.
- Establecer parámetros de referencia para la evaluación de arquitecturas.

Contenidos mínimos: Arquitectura y diseño. Diseño como actividad creativa, colaborativa y técnica. Calidad interna y externa. Atributos de calidad. Coherencia e integridad conceptual. Aspectos técnicos y de gestión. Tácticas y Patrones de diseño. Métodos de diseño. Complejidad y Diseño emergente. Patrones de arquitectura. Revisión, verificación y validación de diseño. Evaluaciones de arquitectura. Arquitecturas y organizaciones. Roles en arquitectura de software. Características del buen arquitecto/a. La arquitectura como servicio al ecosistema. Estándares y variabilidad. Arquitecturas de líneas de productos. Sistemas de sistemas. Integración de sistemas. Estructuras de asignación/ ubicación de la arquitectura. Arquitectura y tecnología. Cuestiones de reuso. Heterogeneidad y complejidad. Apropiación y selección de tecnologías. Comunicación y documentación de arquitectura. Arqueología de arquitecturas. Perspectivas. Sistemas auto-regenerables. Generación de código y arquitecturas generadas por Inteligencia Artificial. Arquitecturas basadas en la nube. Serverless. PaaS. Asincronismo.

Bibliografía de referencia:

- Fundamentals of Software Architecture. Mark Richards, Neal Ford, O'Reilly, 2020.
- Building Evolutionary Architectures: Support Constant Change. Neal Ford, Rebecca Parsons, Patrick Kua. O'Reilly Media, Inc., 2017.
- Design It!: From Programmer to Software Architect. Michael Keeling. Pragmatic Bookshelf, 2017
- Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Gregor Hohpe, Bobby Woolf. Addison-Wesley, 2012.
- Designing Software Architectures: A Practical Approach. Humberto Cervantes, Rick Kazman. Addison-Wesley Professional, 2016
- Documenting Software Architectures: Views and Beyond (2 ed.). Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, Judith Stafford. Pearson Education, 2010.



Operación y Gestión de Servicios de Software con DevOps

Docente: Mg. Ing. Federico Casuscelli

Desafíos e implementación de soluciones de operación y gestión de servicios para sostener los objetivos de escala y confiabilidad de los productos software. Incluye desde infraestructura como código hasta gestión y monitoreo de servicios.

Objetivos:

- Presentar las actividades y desafíos de la gestión y operación de servicios de software.
- Dotar de herramientas y técnicas de gestión específicas de la operación.
- Vincular la tecnología y el hardware con la gestión de servicios de software.

Contenidos mínimos: Atributos de calidad relacionados: Disponibilidad, confiabilidad, escalabilidad, etc. Evaluación y seguimiento. Modelos de gestión, SRE, ITIL, CoBit. Gestión Visual. Métricas e indicadores. Monitoreo y feedback desde producción, a nivel infraestructura, aplicación y negocio. Prácticas de operación: guardias, soporte multi-nivel, análisis de incidentes, acuerdos y objetivos de nivel de servicio. Backups. Tecnologías/plataformas de despliegue (Virtualización y Contenedores, Cloud, IaaS, Server-less, almacenamiento). Elementos de gestión de redes, firewalls, load-balancing, CDN. Seguridad de la información, aseguramiento y evaluación. Gestión de costos, proveedores, niveles de servicio. Estrategias de evolución y soporte de aplicaciones e infraestructura, Chaos Engineering.

Bibliografía de referencia:

- Infrastructure as Code (2 ed). Kief Morris. O'Reilly Media, Inc., 2020
- Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems. Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea, Adam Stubblefield. O'Reilly Media, Inc., 2020.
- Site Reliability Engineering: How Google Runs Production Systems. Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy. O'Reilly Media, Inc., 2016.
- Observability Engineering. Charity Majors, Liz Fong-Jones, George Miranda. O'Reilly Media, Inc., 2022.





Modalidad de cursada

La diplomatura superior se cursa online, con 1 único encuentro de 3 horas por semana. Cada materia consta de una carga total de 40 hs (horas de clases online sincrónicas y horas de estudio asincrónicas). Se cursan dos materias por cuatrimestre cuyas clases se realizan de manera online semana por medio, de forma que el alumno tiene una clase todas las semanas alternando la materia que cursa.

Requisitos de admisión

Contar con al menos 3 años de experiencia comprobable desarrollando proyectos de Ingeniería de software.

Articulación Institucional

Esta diplomatura superior se articulará con una carrera de posgrado de Especialización en Ingeniería de Software Continua que está en proceso de aprobación institucional.

Arancel

10 (diez) cuotas de \$35.000 (treinta y cinco mil pesos argentinos)

Coordinación

Director/Coordinador: Dr. Ing. Diego Fontdevila
Coordinador Académico: Esp. Ing. Nicolás Paez

Contacto

Los interesados en cursar esta diplomatura deben completar su preinscripción en el sistema guaraní (<https://bit.ly/ingenieriadesoftware-untref>) y a partir de ello recibirán más detalles para completar la inscripción. Por cualquier otra inquietud pueden escribir a ingenieriadesoftware@untref.edu.ar.