Ministerio de Capital Humano
Universidad Tecnológica Nacional
Facultad Regional Rosario

Rosario, 02 de julio de 2025.-

VISTO la presentación del Programa Analítico de la asignatura electiva "Técnicas y Tecnologías Avanzadas de Desarrollo de Software", correspondiente a la carrera Ingeniería en Sistemas de Información – Plan 2023, y

CONSIDERANDO

Que los objetivos y contenidos del mismo se ajustan a la reglamentación vigente.

Que dicho Programa Analítico cuenta con el aval del respectivo Consejo Departamental.

Que la Comisión de Enseñanza analizó el Expediente y aconsejó su aprobación.

Por ello y atento a las atribuciones otorgadas por el artículo 85° del Estatuto Universitario.

EL CONSEJO DIRECTIVO DE LA FACULTAD REGIONAL ROSARIO
DE LA UNIVERSIDAD TECNOLÓGICA NACIONAL
R E S U E L V E:

ARTÍCULO 1°.- Aprobar el Programa Analítico de la asignatura electiva "Técnicas y Tecnologías Avanzadas de Desarrollo de Software" para el tercer nivel de la carrera Ingeniería en Sistemas de Información – Plan 2023, que se agrega como Anexo I de la presente resolución. A partir del Ciclo Lectivo 2024.

ARTÍCULO 2°.- Establecer que la misma tendrá validez durante cuatro ciclos lectivos consecutivos, según la Ordenanza N° 1383 – Lineamientos para la implementación de asignaturas electivas para las carreras de grado en el ámbito de la Universidad.

ARTÍCULO 3°.- Regístrese. Comuníquese. Cumplido, archívese.

RESOLUCIÓN N° **569**

UTN FRRo
C.D.
S.R.

Ing. Rubén Fernando CICCARELLI
Decano

Ing. Antonio Luis MUIÑOS
Secretario Académico

Carrera: Ingeniería en Sistemas de Información
Asignatura: Técnicas y Tecnologías Avanzadas de Desarrollo de Software
PROGRAMA ANALÍTICO

1. Datos administrativos de la asignatura

Nivel en la carrera:	3	Dictado:	Anual
Plan de Estudio:	2023	Área:	Desarrollo de Software
Bloque curricular:	Tecnologías Aplicadas	Electiva:	SI
Carga horaria presencial semanal (hs. cátedra):	4	Carga Horaria total anual (hs. reloj):	96
Carga horaria no presencial semanal (hs. reloj) (si correspondiese)		% horas no presenciales (hs. reloj) (si correspondiese)	

2. Presentación, Fundamentación

La cátedra de Técnicas y Tecnologías Avanzadas de Desarrollo de Software busca desarrollar en los alumnos las capacidades necesarias para un egresado en relación al desarrollo de software:

- Desarrollar aplicaciones medianas y grandes que cumplan con parámetros de calidad mínimos tanto a nivel de producto como del proceso de desarrollo.
- Desarrollo de aplicaciones multicapas y orientadas a servicios aplicando patrones de diseño, buenas prácticas y patrones de arquitectura estándar de la industria.
- Construir e implementar aplicaciones con niveles aceptables de seguridad.
- Diseñar pruebas que aseguren el cumplimiento de los requerimientos funcionales de las aplicaciones
- Desarrollar interfaces de usuario aplicando buenas prácticas de usabilidad y user experience.
- Insertarse en cualquier equipo de trabajo profesional, conociendo las prácticas y técnicas más utilizadas.
- Conocer y dominar las herramientas de trabajo colaborativas y de productividad utilizadas normalmente en la industria.

- Contar con herramientas tanto técnicas como metodológicas y aplicarlas experimentalmente al desarrollo de aplicaciones medianas y grandes.
- Construir e implementar una solución de software de acuerdo con el análisis y diseño realizados.
- Desarrollar software en equipo, colaborando y cooperando como un grupo de desarrolladores, con las complejidades que esto implica.
- Identificar y utilizar aquellas tecnologías que pueden facilitar y mejorar los procesos de desarrollo de software.

3. Contenidos Mínimos

No aplica

4. Objetivos establecidos en el DC

No aplica

5. Asignaturas correlativas previas

Para cursar y rendir debe tener cursada:

☐ Asignatura/s:

Paradigmas de Programación

Análisis de Sistemas de Información

Para cursar y rendir debe tener aprobada:

- Asignatura/s:

Lógica y Estructuras Discretas

Algoritmos y Estructuras de Datos

6. Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

☐ Asignatura/s que la requieren cursada:

Ninguna

☐ Asignatura/s que la requieren aprobada:

Ninguna

7. Programa analítico

Este programa analítico contempla los contenidos mínimos, previstos en el DC vigente, y aquellos que se consideran necesarios para desarrollar los resultados de aprendizaje propuestos.

Unidad Nº: 1

Título: Conceptos básicos de desarrollo de software profesional

Contenidos:

- Implementación y aplicación de los conceptos de orientación a objetos aplicados en un lenguaje.
- Programación prototipada: uso y aplicación.
- Lenguajes de programación multiparadigma
- Sistemas Web
  - Protocolo HTTP
  - Request y Response
  - Implementación y aplicación de Timeouts y Callbacks
  - HTML
    - Tags, elements, components
    - DOM, Virtual DOM
    - Dynamic manipulation
  - CSS
    - Frameworks CSS
    - CSS preprocesors
  - Intercambio de datos asyncrono: JSON y AJAX

Unidad Nº: 2

Título: Tecnologías de soporte al desarrollo de software y técnicas de calidad asociadas

Contenidos:

- IDE vs editores de código. Setup del entorno y proyecto
- Detectores de bugs durante el desarrollo y validadores de estándares de programación, uso y aplicación.
- Aplicación de herramientas y técnicas de gestión de código al desarrollo de código

- Source control, metodologías (branching, reverts, tagging, etc) y buenas prácticas.
- Código, artefactos y dependencias: diferencia y relaciones
- Técncas de Versioning
- Code review: concepto, técnica y herramientas

- Documentación de código e interfaces
- Herramientas para testing automatizado
  - Aplicación de herramientas de testing Unitario, Headless, User Interface, Integration y End User
  - Uso de herramientas con metodologías de TDD, BDD y DDD

Unidad Nº: 3

Título: Componentes de desarrollos de software profesional

Contenidos:

- Patrones de diseño y programación (de arquitectura, GoF, etc)
- Aplicación de buenas prácticas aplicadas al código fuente
  - Organización de código en unidades lógicas (packages, modules, etc)
  - Convención de nombres y formateo del código
  - SOLID
  - Once and only once
  - Tell, don't ask
  - Program to an interface not to an implementation
  - Favor object composition over class inheritance
  - Make it work, make it right, make it fast
  - Refactoring
- Introducción a la usabilidad y experiencia de usuario
- Desarrollo en capas.
  - Arquitecturas de 2-Capas, 3-Capas y N-Capas, componentes e implementación.
  - Aplicación de Técnicas y Herramientas para la Capa de datos
    - Mapeo de objetos. Object-Relational Mapping y Object-Document Mapping.
    - Connection Pool.
    - Object cache (memcache, redis. Etc)
    - Replicas y balanceo.
    - Seeding y Migrations
- Capa de presentación

- Aplicación e implementación de frameworks de frontend
  - Arquitecturas SPA, SSR, etc
  - Component based Frameworks
  - Routing
- Construcción Agnostic frontend.

- Capas de servicios
  - Macro y micro servicios
  - Arquitecturas de servicios
  - APIs: definición, objetivo componentes esenciales, especificación y documentación
  - Web Services
  - Servicios internos y externos
  - Interfaces de servicios
  - Implementaciones: Rest, GraphQL gRPC, tRPC, etc

Unidad Nº: 4

Título: Características no funcionales de un sistema de información

Contenidos:

- Deployment
  - Aplicación de herramientas y técnicas
  - Tecnicas de deploy para lenguajes Compilados vs Interpretados
  - Aplicación de herramientas de deploy de artefactos
  - Release Management
  - Continuous Integration y Continuous Delivery: Herramientas y su implementación
    - Builds automatizados
    - Aplicar Web hooks para deploys
    - Integración con testing y mediciones
    - Aplicar AB Development y Canary releases
  - Containers
- Componentes mínimos de seguridad:
  - Usuarios, roles, perfiles y permisos
  - Auditoría y Autorizaciones
  - Single Sign-On y sistemas externos de validación de usuarios
- Manejo de errores y excepciones

- Jerarquía y niveles de excepciones
- Log de errores de aplicación
- Técnicas basadas en resultado

- Manejo de estado
  - Cookies, Sessiones y Tokens
  - Manejo de estado en sistemas distribuidos y de HA
- Alta disponiblidad y escalabilidad.
  - Limitaciones de performance y arquitecturas de escalabilidad
  - Consideraciones de arquitecturas de alta disponibilidad
  - BCP

Unidad Nº: 5

Título: Planificación y Gestión de proyectos de software

Contenidos:

- Aplicación de modelos de desarrollo de software a casos realistas:
  - Metodologías ágiles: Scrum, Lean y Kanban.
  - Gestión de pendientes: Backlog y Progress Boards.
- Aplicación e implementación de Prácticas de diseño y programación iterativas e incrementales.
- Dinamicas grupales.

Carga horaria por tipo de formación práctica de toda la asignatura

Tipo de formación práctica	Horas reloj
Formación experimental	0
Análisis y resolución de problemas de ingeniería y estudios de casos	32
Formulación, análisis y desarrollo de proyectos.	32

Bibliografía Obligatoria:

Martin Fowler (2003) *Patterns of Enterprise Application Architecture*. Addison-Wesley

Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts (2012) *Refactoring: Improving the Design of Existing Code(new edition).* Addison-Wesley Prentice Hall

Jez Humble, David Farley (2011) *Continuous Delivery:Reliable Software Releases through Build, Test, and Deployment Automation.* Addison-Wesley Professional

Scott Millett, Nick Tune (2015). *Patterns, Principles, and Practices of Domain-Driven Design* Wrox.

Cocca, G. (22 de junio de 2022). JavaScript Design Patterns – Explained with Examples https://www.freecodecamp.org/news/javascript-design-patterns-explained/

Ravindranath H. (9 de marzo de 2021). A Comprehensive Guide To JavaScript Design Patterns https://www.lambdatest.com/blog/comprehensive-guide-to-javascript-design-patterns/

Osmani A. Learning JavaScript Design Patterns. Recuperado el 12 de diciembre de 2022. https://www.patterns.dev/posts/classic-design-patterns/

Maldonado L. (5 de diciembre de 2022) 33 Concepts Every JavaScript Developer Should Know https://github.com/utnfrrodsw/33-js-concepts/commit/b1b2e1cc95815df0d7b8cca28edc524fff73d136

Sacha Lifszyc. La Cocina del Código (5 de octubre de 2022). 33 CONCEPTOS DE JAVASCRIPT QUE DEBERÍAS SABER [Lista de Reproducción]. Youtube. https://youtube.com/playlist?list=PLfWyZ8S-XzecAttp3QU-gBBXvMqEZTQXB

Bibliografía optativa y otros materiales a utilizar en la asignatura:

Stark E. (2014). *Agile Project Management QuickStart Guide: A Simplified Beginners Guide To Agile Project Management.* CreateSpace Independent Publishing Platform

Duran García M. A. (2022). *Aprendiendo Git.* LearnPub

Asignatura equivalente respecto al Plan Anterior

Técnicas y Tecnologías Avanzadas de Desarrollo de Software – Plan 2008