Ministerio de Capital Humano
Universidad Tecnológica Nacional
Facultad Regional Rosario

Rosario, 02 de julio de 2025.-

VISTO el Expediente ID Nº 8175857, relacionado con 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 2008, 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 85º 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 1º.- 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 2008, que se agrega como Anexo I de la presente resolución. A partir del Ciclo Lectivo 2024.

ARTÍCULO 2º.- Establecer que la misma tendrá validez durante cuatro ciclos lectivos consecutivos, según la Ordenanza Nº 1383 – Lineamientos para la implementación de asignaturas electivas para las carreras de grado en el ámbito de la Universidad.

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

RESOLUCIÓN Nº **425**

UTN FRRo
C.D.
S.R.

Ing. Rubén Fernando CICCARELLI
Decano

Ing. Antonio Luis MUIÑOS
Secretario Académico

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL ROSARIO
DEPARTAMENTO DE INGENIERÍA EN SISTEMAS

RESOLUCION N° 425

ANEXO N° I

Programa analítico de asignatura electiva

Técnicas y Tecnologías Avanzadas de Desarrollo de Software

Carrera:	*Ingeniería en Sistemas de Información*						
Departamento:	*Ingeniería en Sistemas de Información*						
Titulación[1]:	X	*Ingeniería en Sistemas de Información*	X	*Analista universitario de Sistemas*			
Plan de Estudio:	*2008 – ordenanza 1150*		**Área[2]:**	*Programación*			
Dictado:	X	*Anual*		*Cuatrimestral*	**Nivel:**	*3*	**Electiva:** *Si*
Carga horaria Semanal:	*4*		Carga horaria total de la asignatura:				*128*
Fecha de Confección[3]:			Versión[4]				

Fundamentación de la asignatura:[5]	*La cátedra 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.*
Objetivos Generales[6]:	*- Conocer las arquitecturas, herramientas y patrones para el desarrollo de software.* *- Desarrollar interfaces de usuario.* *- Crear soluciones de software que den respuestas a necesidades reales.* *- Aplicar buenas prácticas y tecnologías en el desarrollo seguro.*

Programa de contenido analítico

[1] *Indique los títulos de la carrera para los que se propone el programa analítico. Márquelos con una cruz.*
[2] *Área a la que pertenece la asignatura*
[3] *refiere a la fecha en que se confecciona o desarrolla la versión*
[4] *Si el programa no es la primera vez que se entrega se produce un cambio en el número de versión cambio. Si el cambio es significativo cambia el entero sino los dígitos después del punto.*
[5] *Importancia para la formación profesional en función del perfil del egresado*
[6] Objetivos generales que justifican la inclusión de la asignatura.

Unidad temática Nº: 1

Eje Conceptual: Conceptos básicos de desarrollo de software profesional

Temas:

- 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 temática Nº: 2

Eje Conceptual: Tecnologías de soporte al desarrollo de software y técnicas de calidad asociadas

Temas:

- 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 herrmaintas de testing Unitario, Headless, User Interface, Integration y End User
  - Uso de herramientas con metodologías de TDD, BDD y DDD

Unidad temática Nº: 3

Eje Conceptual: Componentes de desarrollos de software profesional

Temas:

- 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 temática Nº: 4

Eje Conceptual: Características no funcionales de un sistema de información

Temas:

- 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 temática N°: 5

Eje Conceptual: Planificación y Gestión de proyectos de software

Temas:

- 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.

Bibliografía[7]

Obligatoria o básica:

Título	Autor/es	Editorial	Año de Edición
Patterns of Enterprise Application Architecture	Martin Fowler	Addison-Wesley	2003
Refactoring: Improving the Design of Existing Code(new edition)	Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts	Addison-Wesley Prentice Hall	2012
Continuous Delivery:Reliable Software Releases through Build, Test, and Deployment Automation	Jez Humble, David Farley	Addison-Wesley Professional	2011
Patterns, Principles, and Practices of Domain-Driven Design	Scott Millett, Nick Tune	Wrox	2015
JavaScript Design Patterns – Explained with Examples	Cocca, G	https://www.freecodecamp.org/news/javascript-design-patterns-explained/	2022
A Comprehensive Guide To JavaScript Design Patterns	Ravindranath H.	https://www.lambdatest.com/blog/comprehensive-guide-to-javascript-design-patterns/	2021
Learning JavaScript Design Patterns.	Osmani A.	https://www.patterns.dev/posts/classic-design-patterns/	2022
Concepts Every JavaScript Developer Should Know	Maldonado L.	https://github.com/utnfrrodsw/33-js-concepts/commit/b1b2e1cc95815df0d7b8cca28edc524fff73d136	2022
La Cocina del Código	Sacha Lifszyc.	https://youtube.com/playlist?list=PLfWyZ8S-XzecAttp3QU-gBBXvMqEZTQXB	2022

7 Para textos: citar autor, título, ciudad, editorial, año. Para revistas: citar autor, título del artículo, nombre de la revista, n°, lugar, edición, año, páginas., Para sitios web: dirección de la página.

Complementaria:

Título	Autor/es	Editorial	Año de Edición
Agile Project Management QuickStart Guide: A Simplified Beginners Guide To Agile Project Management.	Stark E.	CreateSpace Independent Publishing Platform	2014
Aprendiendo Git.	Duran Garcia M. A.	LearnPub	2022

Propuesta Pedagógica

Se utilizarán las siguientes metodologías de enseñanza para desarrollar las competencias de egreso y objetivos marcados por el Plan de Estudios así como para cumplir con los contenidos mínimos en función del Programa Analítico aprobado.

En cuanto al enfoque de enseñanza la cátedra ha seleccionado los siguientes, que luego se indican asociados a los resultados de aprendizaje en la tabla correspondiente.

1. Visualización de videos, Resolución de Ejercicios, Formación Experimental en Laboratorios de Acceso Local y Aula Invertida:
La cátedra pone a disposición de los alumnos videos sobre los temas de la materia y material de soporte al video que incluye explicaciones, ejemplos, documentación, laboratorios y ejercicios.
El profesor indicará clase a clase el contenido que deben revisar.
Los alumnos deberán ver y analizar dichos videos, tomar notas y revisar el material de soporte provisto. Luego el alumno podrá consultar a los docentes sobre las dudas del contenido de los videos y el material provisto; también deberá realizar los ejercicios y laboratorios propuestos pudiendo consultar con el docente.

2. Aprendizaje Cooperativo en Grupos Pequeños, Aprendizaje Basado en Proyectos, Presentaciones Orales y Trabajo Autónomo:
Los alumnos deberán plantear un proyecto para desarrollar en grupo una aplicación de backend(BE) y frontend(FE) de acuerdo a las directrices de la cátedra.
Deberán especificar el alcance, desarrollar las aplicaciones de BE y FE correspondiente y realizar el seguimiento del proceso de desarrollo con la asistencia y seguimiento de los docentes
Finalmente realizarán una presentación/defensa del proyecto.

3. Aprendizaje Cooperativo en Grupos Pequeños, Presentaciones Orales, Presentaciones Escritas, Trabajo Autónomo y Seminario:
Se plantean una serie de temáticas para que los alumnos realicen un proyecto de Prueba de Concepto (PoC).
Entre varios grupos deberán elegir tecnologías alternativas o complementarias dentro de un mismo tema y llevar adelante un proceso de investigación de forma colaborativa e intergrupal.
Luego deberán construir una implementación utilizando las temáticas elegidas y producir un informe conjunto escrito.
Finalmente deberán realizar una presentación conjunta de las PoC intergrupal y discutir con la clase las conclusiones a las que llegaron.

Metodología de Evaluación

Instrumentos:

Ins 1. Resolución de Ejercicios, Corrección y Debate: La cátedra propone ejercicios que los alumnos deberán resolver primero y luego los profesores los resolverán en la clase con participación de los alumnos mediante debate y propuestas. El alumno realizará al finalizar una autoevaluación en base a la solución propuesta, el debate realizado y las consultas que realizó durante la instancia de resolución en clase.

Ins 2. Debate y realización de ejemplos: La cátedra brindará ejemplos de problemas del mundo real a través de los videos y material de soporte y planteará soluciones aplicando diversos contenidos de la cátedra. El alumno deberá analizar estos ejemplos y las soluciones propuestas, evaluar su solución y podrá debatir y consultar en clase sobre las mismas.

Ins 3. Desarrollo de una aplicación de backend: El alumno deberá proponer un desarrollo de software en grupos pequeños a realizar basado en requerimientos y alcance planteados por la cátedra. Deberá desarrollar la aplicación de backend en grupo, realizar el seguimiento del proyecto de desarrollo y presentar el código fuente del mismo. Luego, en el instrumento 6, presentará el software desarrollado. Este instrumento se evaluará 2 veces, una para la regularidad con un alcance parcial indicado por la cátedra y otra vez para evaluar la aprobación con el alcance completo.

Ins 4. Desarrollo de una aplicación de frontend: El alumno deberá proponer un desarrollo de software en grupos pequeños a realizar basado en requerimientos y alcance planteados por la cátedra. Deberá desarrollar la aplicación de frontend en grupo, realizar el seguimiento del proyecto de desarrollo y presentar el código fuente del mismo. Luego, en el instrumento 6, presentará el software desarrollado. Este instrumento se evaluará 2 veces, una para la regularidad con un alcance parcial indicado por la cátedra y otra vez para evaluar la aprobación con el alcance completo.

Ins 5. Prueba de Concepto (PoC): Desarrollo de una PoC. Cada grupo deberá elegir un tema a investigar de forma conjunta con otro(s) grupos. Deberán realizar la investigación en forma inter-grupal sobre temáticas alternativas/complementarias y desarrollar ejemplo(s) donde se apliquen las tecnologías y conceptos ejemplificados.

Los grupos deberán llegar a una conclusión sobre la prueba de concepto y confeccionar un informe en conjunto con el resultado. Finalmente deberán exponer de manera conjunta el resultado de la PoC en clase y realizar un debate con los demás alumnos.

Ins 6. Evaluación de Desarrollo. La evaluación inicia con una exposición oral ante los docentes, tanto sobre el producto final como sobre el proceso de desarrollo de software referido los instrumentos 3 y 4, las decisiones técnicas durante el mismo y la implementación realizada. El docente indagará sobre los aspectos del proceso o el software que no hayan sido expuestos y de ser necesario se presentará a los alumnos una evaluación que abarque los contenidos de la materia. La misma requerirá la aplicación de los conceptos teóricos y prácticos a un desarrollo de software. El formato podrá ser oral, escrito o multiple-choice.

Este instrumento se evaluará 2 veces, una para la regularidad con un alcance parcial indicado por la cátedra y otra vez para evaluar la aprobación con el alcance completo.

Asignaturas Correlativas del plan[8]

Asignaturas regulares para el cursado:	Paradigmas de Programación Análisis de Sistemas
Asignaturas aprobadas para el cursado:	Matemática Discreta Algoritmos y Estructuras de Datos
Asignaturas aprobadas para rendir:	Matemática Discreta Algoritmos y Estructuras de Datos

Justificación de correlatividades

Análisis de Sistemas: Dadas las condiciones prácticas de la materia se requiere en primer lugar la habilidad de detectar y especificar los requerimientos de las aplicaciones así como entender los artefactos que se utilizarán para especificar los ejercicios y las prácticas. Estos artefactos son enseñados en su mayoría en Análisis de Sistemas.
También dadas las características de evaluación se requiere que el alumno cuente con la capacidad de relevar y documentar los requerimientos de una aplicación así como también los componentes del sistema y el desarrollo del mismo.

Paradigmas de Programación: Dado el foco de la asignatura en el desarrollo de aplicaciones y ejemplos con lenguajes orientados a objetos a la vez que intenta profundizar en la orientación a objetos aplicando diversos patrones y arquitecturas para la POO se requiere que el alumno tenga una comprensión del paradigma de orientación a objetos. A su vez se espera instruir en lenguajes que combinan múltiples paradigmas lo que requiere que los alumnos comprendan al menos los principales paradigmas de programación es que se requiere a Paradigmas de Programación como correlativa.

Asignaturas Equivalentes respecto del plan anterior[9]

Asignatura/s equivalente respecto del plan anterior:	

Asignatura equivalente respecto del plan 2023

Desarrollo de Software

8 No está permitido indicar asignaturas electivas como correlativas. Además todos los cuadros deben estar completados.
9 Consignar asignaturas que se pueden otorgar como equivalentes para las posibles solicitudes de cambio de plan.