



Ministerio de Educación  
Universidad Tecnológica Nacional  
Facultad Regional Rosario

"2022 - Las Malvinas son argentinas"

Rosario, 21 de diciembre de 2022.-

VISTO el Expediente ID N°: 8142001, relacionado con la presentación del Programa Analítico de la asignatura "Desarrollo de Software", correspondiente a la carrera Ingeniería en Sistemas de Información, y

CONSIDERANDO

Que la presentación realizada obedece a la implementación del nuevo Diseño Curricular aprobado por el Consejo Superior de la Universidad Tecnológica Nacional – Ordenanza CSU N° 1877.

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

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

RESUELVE:

ARTÍCULO 1°.- Aprobar el Programa Analítico de la asignatura "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.

ARTÍCULO 2°.- Regístrese. Comuníquese. Elévese. Publíquese. Cumplido, archívese.

RESOLUCIÓN N° **544**

|      |
|------|
| UTN  |
| FRRo |
| C.D. |
| S.R. |
|      |

Ing. Rubén Fernando CICCARELLI  
Decano

Ing. Antonio Luis MUIÑOS  
Secretario Académico

**Carrera: Ingeniería en Sistemas de Información**  
**Asignatura: 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:  | NO                     |
| Carga horaria presencial semanal (hs. cátedra):                        | 4                     | Carga Horaria total anual (hs. reloj):                     | 96                     |
| Carga horaria no presencial semanal (hs. reloj)<br>(si correspondiese) |                       | % horas no presenciales (hs. reloj)<br>(si correspondiese) |                        |

**2. Presentación, Fundamentación**

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.

| 3. Contenidos Mínimos  |
|--|
| <ul style="list-style-type: none"> <li>- Arquitectura de aplicaciones multicapa.</li> <li>- Herramientas de soporte al proceso de desarrollo.</li> <li>- Programación de la interfaz de usuario de una aplicación.</li> <li>- Aplicaciones orientadas a servicios.</li> <li>- Desarrollo Seguro.</li> <li>- Planes de prueba.</li> </ul>         |
| 4. Objetivos establecidos en el DC   |
| <ul style="list-style-type: none"> <li>- Conocer las arquitecturas, herramientas y patrones para el desarrollo de software.</li> <li>- Desarrollar interfaces de usuario.</li> <li>- Crear soluciones de software que den respuestas a necesidades reales.</li> <li>- Aplicar buenas prácticas y tecnologías en el desarrollo seguro.</li> </ul> |
| 5. Asignaturas correlativas previas  |
| Para cursar y rendir debe tener cursada:   |
| <input type="checkbox"/> Asignatura/s: <ul style="list-style-type: none"> <li>Paradigmas de Programación</li> <li>Análisis de Sistemas de Información</li> </ul>   |
| Para cursar y rendir debe tener aprobada:  |
| <ul style="list-style-type: none"> <li>• Asignatura/s:             <ul style="list-style-type: none"> <li>Lógica y Estructuras Discretas</li> <li>Algoritmos y Estructuras de Datos</li> </ul> </li> </ul>   |

|  |
|--|
|  |
|  |
| <b>6. Asignaturas correlativas posteriores</b>   |
| Indicar las asignaturas correlativas posteriores:  |
| <input type="checkbox"/> Asignatura/s que la requieren cursada:<br>Ingeniería y Calidad de Software                          |
| <input type="checkbox"/> Asignatura/s que la requieren aprobada:<br>Seguridad en los Sistemas de Información, Proyecto Final |

|  |
|--|
| <b>7. Programa analítico</b>   |
| 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 N°: 1<br>Título: Conceptos básicos de desarrollo de software profesional<br>Contenidos: <ul style="list-style-type: none"><li>● Implementación y aplicación de los conceptos de orientación a objetos aplicados en un lenguaje.</li><li>● Programación prototipada: uso y aplicación.</li><li>● Lenguajes de programación multiparadigma</li><li>● Sistemas Web<ul style="list-style-type: none"><li>○ Protocolo HTTP</li><li>○ Request y Response</li><li>○ Implementación y aplicación de Timeouts y Callbacks</li><li>○ HTML<ul style="list-style-type: none"><li>■ Tags, elements, components</li><li>■ DOM, Virtual DOM</li><li>■ Dynamic manipulation</li></ul></li><li>○ CSS<ul style="list-style-type: none"><li>■ Frameworks CSS</li><li>■ CSS preprocessors</li></ul></li><li>○ Intercambio de datos asincrónico: JSON y AJAX</li></ul></li></ul> |

Unidad N°: 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
  - Técnicas 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 N°: 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 N°: 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, Sesiones y Tokens
  - Manejo de estado en sistemas distribuidos y de HA
- Alta disponibilidad y escalabilidad.
  - Limitaciones de performance y arquitecturas de escalabilidad
  - Consideraciones de arquitecturas de alta disponibilidad
  - BCP

**Unidad N°: 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.
- Dinámicas 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/utnfirodsw/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