



## Desarrollo de Software Ordenanza 1877

### Datos administrativos de la asignatura

Departamento:	Ingeniería en Sistemas de Información	Carrera	Ingeniería en Sistemas de Información
Asignatura:	Desarrollo de Software		
Nivel de la carrera	3er año	Duración	Cuatrimestral
Bloque curricular:	Tecnologías Aplicadas	Área	Desarrollo de Software
Carga horaria presencial semanal:	6 horas reloj (8 horas catedra)	Carga Horaria total:	96 horas reloj (128 horas catedra)
Carga horaria no presencial semanal (si correspondiese)		% horas no presenciales (si correspondiese)	
Profesor/es Titular/Asociado/Adjunto:	Emilio Watemberg (Adjunto)	Dedicación:	1DS interino
Auxiliar/es de 1º/JTP:	Ignacio Ibañez (Ay. 1º) Jose Ramirez (Ay. 2º) Milagros Soberon (Ay. 2º) Paula Giudici (Ay. 2º)	Dedicación:	1DS Interino 1DS Interino 1DS Ad Honorem 1DS Ad Honorem

### Propósito

Brindar herramientas y técnicas para que los y las estudiantes sean capaces de:

- Diagramar un diseño básico de Arquitectura de Software, a partir de requerimientos Funcionales y No Funcionales.
- Especificar las funcionalidades, empleando diagramas UML de Actividad y Secuencia.
- Organizar las tareas especificadas en hitos, que permitan distribuir el trabajo en el equipo de desarrollo.
- Participar de la selección de un lenguaje de programación, a partir de las ventajas técnicas del mismo, conforme la necesidad planteada.
- Ejecutar el desarrollo de manera colaborativa aplicando los patrones de programación sugeridos para una arquitectura orientada a servicios.
- Conocer y utilizar el protocolo HTTP, sus métodos, encabezados y características de seguridad.
- Conocer el funcionamiento de un Servidor Web.
- Programar pruebas Unitarias.
- Resolver errores de programación utilizando el depurador.



ing. Guerrieri Ruben Alberto  
 Director de Departamento  
 DISI - UTN - FRLP



- Construir, empaquetar y probar el software construido.

### Objetivos establecidos en el DC

- 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.

### Resultados de aprendizaje

- RA01- Emplea técnicas de estimación de esfuerzo, según los recursos disponibles y los requerimientos solicitados.
- RA02- Organiza tareas de equipo, según las características técnicas requeridas y las habilidades individuales disponibles de los integrantes del equipo de trabajo.
- RA03- Utiliza software de programación como compiladores, depuradores y otras herramientas para crear código.
- RA04- Administra herramientas de seguimiento de trabajo y colaboración.
- RA05- Desarrolla y mantiene pruebas unitarias
- RA06- Realiza el despliegue del software construido.
- RA07- Posee la habilidad de complementar y asistir a sus pares ante inconvenientes e imprevistos.

### Asignaturas correlativas previas

Para cursar y rendir, el estudiante debe tener:

Cursada:

- Paradigmas de Programación
- Análisis de Sistemas de Información (integradora)

Aprobada:

- Lógica y Estructuras Discretas
- Algoritmos y Estructuras de Datos

### Asignaturas correlativas posteriores

Desarrollo de Software Cursada para:

- Ingeniería y Calidad del Software

Desarrollo de Software Aprobada para:

- Seguridad en los Sistemas de Información
- Proyecto Final (integradora)



MARIA EUGENIA LAVORATTO  
 DIRECTORA  
 DIRECCIÓN ACADÉMICA  
 U.T.N. F.R.L.P.

Ing. Guerrieri Ruben Alberto  
 Director de Departamento  
 DISI - UTN - FRLP



**Programa analítico. Unidades temáticas**

**Contenidos Mínimos**

- Arquitectura de aplicaciones multicapa.
- Herramientas de soporte al proceso de desarrollo.
- Programación de la interfaz de usuario de una aplicación.
- Aplicaciones orientadas a servicios.
- Desarrollo Seguro.
- Pruebas unitarias.

**CONTENIDOS ANALÍTICOS**

**UNIDAD N° 1: Arquitectura de Aplicaciones Multicapa**

- 1.1. Introducción a la arquitectura de software multicapa.
  - 1.1.1. Persistencia, Lógica de Negocio, Servicios, Seguridad, Bitácoras, Pruebas, Presentación.
- 1.2. Clasificación de requerimientos Funcionales y No Funcionales
  - 1.2.1. Performance disponibilidad, responsabilidad (funcionalidad), mantenimiento (desacoplamiento + modificabilidad), robustez, seguridad, recuperación, testeable, entregable.

Total horas: 9

**UNIDAD N° 2: Herramientas de Soporte al Proceso de Desarrollo.**

- 2.1. Organización del equipo, planificación de tareas
- 2.2. Gestión del Código Fuente

Total horas: 12h

**UNIDAD N°3: Programación de la interfaz de usuario de una aplicación.**

- 3.1. Herramientas de construcción de interfaz gráfica web.
- 3.2. Herramientas de construcción de interfaz gráfica de aplicaciones de escritorio.
- 3.3. SPA. Single Page Application.
- 3.4. Programación Reactiva.

Total horas: 27h

**UNIDAD N°4: Aplicaciones orientadas a Servicios.**

- 4.1. Desarrollo de servicios web.
- 4.2. Acceso a datos

Total horas: 27h

**UNIDAD N°5: Desarrollo Seguro.**

- 5.1. Conceptos básicos de seguridad en el servidor web y aplicaciones.



ing. Guerrieri Ruben Alberto  
 Director de Departamento  
 DISI - UTN - FRLP



## 5.2. Implementación en un servidor web

Total horas: 15h

### UNIDAD N°6: Pruebas Unitarias.

6.1. Incorporación de cambios de código

6.2. Pruebas unitarias

Total horas: 6h

### Metodología de enseñanza

Las estrategias metodológicas se basan en:

- Clases presenciales, expositivas, dialogadas con preguntas alternadas, para propiciar la participación reflexiva de las y los estudiantes.
- Análisis y desarrollo de casos de estudio para promover el intercambio de ideas y la construcción colectiva de conocimiento.
- Propuesta de trabajos ABP (Aprendizaje basado en problemas) para favorecer el desarrollo del pensamiento crítico, la comunicación, la colaboración y la resolución de problemas.
- Uso de herramientas de control de código fuente donde se irá entregando y construyendo la historia evolutiva del desarrollo de software.
- Uso de secuencia de actividades como plan de ejecución para la etapa de desarrollo del software. En cada etapa del plan, se incorporarán los conceptos básicos necesarios y material para el estudio.
- Clases virtuales sincrónicas para favorecer y asegurar el desarrollo de software desde la PC del estudiante, garantizando la experiencia completa del desarrollo de software en un ambiente conocido y no preparado.

### Recomendaciones para el estudio

Debido a la gran cantidad de contenidos de la materia y a la correlación entre los mismos, es recomendable que las y los estudiantes desarrollen la autogestión de sus aprendizajes a través de las guías de estudio propuestas por la cátedra.

Del mismo modo, se recomienda la asistencia a clases a efectos de aprovechar las explicaciones y discusiones reflexivas brindadas por los docentes, ya que muchos conceptos se brindan y/o analizan en clase y son difíciles de obtenerlos de la bibliografía.

Es de fundamental importancia la participación en clase, la realización de consultas, ya sean presenciales o virtuales (sincrónicas o asincrónicas). Es de destacar que los contenidos de la materia son acumulativos, es decir, que se van encadenando los temas nuevos con los previos, siendo indispensable aplicarlos en forma incremental en la resolución de los problemas propuestos.

### Metodología de evaluación ACADÉMICA

ES COPIA FIEL DEL ORIGINAL

Se propone realizar dos tipos de evaluaciones:

Formativa o de proceso, a aplicar a lo largo de la cursada de la materia, que permita observar y analizar los procesos educativos de las y los estudiantes, para identificar las



MARIA EUGENIA LAVORATTO  
DIRECTORA  
DIRECCIÓN ACADÉMICA  
U.T.N. F.R.L.P.

Ing. Guerrieri Ruben Alberto  
Director de Departamento  
DISI - UTN - FRLP



fortalezas y dificultades, y en función de ello, definir qué acciones tomar como docente. Esta modalidad se llevará a cabo a través de diferentes instrumentos: preguntas disparadoras o dialogadas; análisis y desarrollo de casos de estudio, (en ambos casos para fomentar la participación en clase del estudiante); cuestionarios con retroalimentación en el campus y guías de resolución de problemas.

- Sumativa, a través de la revisión de las entregas, que permitan determinar el grado de participación y apropiación de los saberes (saber saber, ser y hacer).

**Condiciones de aprobación de la cursada y la materia:**

**Modalidad 1°- Aprobación no directa (común o sin Promoción)**

La cursada se regulariza mediante:

- la **aprobación de las entregas formales de cada etapa del proyecto**. Nota de aprobación de cada instancia evaluatoria: **4 (cuatro) o 5 (cinco)**

La materia se aprueba mediante la exposición final del Proyecto realizado durante la cursada. El trabajo realizado deberá ser expuesto a través de una presentación, con la posibilidad de responder preguntas. Nota de aprobación: **6 (seis) o superior**

**Modalidad 2°- Aprobación directa (con Promoción)**

La materia se promociona mediante:

- Asistencia al 75% de las clases.
- la **entrega y aprobación de cada etapa del proyecto**.

La **calificación final** será obtenida a partir de la evaluación individual de las distintas etapas, complementándola con la exposición final y cuestionario oral final. Dicho **promedio final debe ser con nota 6 (seis) o superior**.

**Observación:**

Proyecto: Desarrollo de un sistema informático.

El trabajo debe ser entregado completo como condición necesaria para obtener la cursada, es decir, incluyendo la documentación funcional (Casos de Uso, comportamiento, arquitectura).

Si la construcción del diseño es considerada suficiente para su aprobación el estudiante recibirá el mérito por esta tarea para poder obtener la promoción directa.

**Modalidad 3°- Recursar la materia con cursada aprobada**

**ES COPIA FIEL DEL ORIGINAL**

De acuerdo con la Resolución 1648 se aceptará que un/a estudiante solicite ser admitido a recursar la materia, sin exigirle el 75% de asistencia a clases. Debiendo cumplir con los requisitos para alcanzar la aprobación directa de la materia en lo que respecta al desarrollo e implementación del sistema informático.



MARIA EUGENIA LAVORATTO  
DIRECTORA  
DIRECCIÓN ACADÉMICA  
U.T.N. F.R.L.P.

Ing. Guerrieri Ruben Alberto  
Director de Departamento  
DISI - UTN - FRLP



### Recursos necesarios

- Aula con proyector para el desarrollo de las clases
- Gabinete de computación con proyector y software instalado para realizar ejercitación sobre distintas herramientas para el desarrollo de sistemas
- Software libre para que los estudiantes puedan desarrollar los proyectos fuera de gabinete, utilizando equipos propios.
- Biblioteca o libros en digital según la conveniencia de cada estudiante
- Acceso a Internet para búsqueda de material, resolver situaciones problemáticas con soluciones ya probadas por otros desarrolladores, etc.



Ing. Guerrieri Ruben Alberto  
Director de Departamento  
DISI - UTN - FRLP

