

**PROGRAMA DE LA ASIGNATURA:**  
Laboratorio de Software (IF055)

**CÓDIGO:** IF055  
**AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:**  
3 año  
**FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:**  
2022-12-01  
**CARRERA/S:** Analista Universitario de Sistemas  
050/2017,

**CARÁCTER:** CUATRIMESTRAL (2do)  
**TIPO:** OBLIGATORIA  
**NIVEL:** PREGRADO - GRADO  
**MODALIDAD DEL DICTADO:** PRESENCIAL  
**MODALIDAD PROMOCION DIRECTA:** SI  
**CARGA HORARIA SEMANAL:** 7 HS  
**CARGA HORARIA TOTAL:** 105 HS

**EQUIPO DOCENTE**

Nombre y Apellido	Cargo	e-mail
FEDERICO EDUARDO GONZALEZ	Profesor Adjunto	fgonzalez@untdf.edu.ar
LUCILA LOURDES CHIARVETTO PERALTA	Asistente	lchiarvetto@untdf.edu.ar

## 1. FUNDAMENTACION

Esta asignatura permite que el alumno relacione todos los conceptos aprendidos en las cátedras anteriores y los utilice para resolver un problema concreto. Durante todo el cuatrimestre deberá cubrir las distintas etapas del desarrollo de software de una forma totalmente práctica llegando a diseñar, documentar e implementar las partes centrales de su propuesta.

## 2. OBJETIVOS

### a) OBJETIVOS GENERALES

El objetivo general del curso es capacitar al alumno para llevar adelante el desarrollo concreto de una aplicación, siguiendo los conceptos aprendidos en las asignaturas previas.

### b) OBJETIVOS ESPECIFICOS

Aprender a especificar correctamente los requerimientos de un sistema, elegir las herramientas adecuadas para resolver el problema y utilizarlas en el desarrollo de la aplicación, estimando correctamente los tiempos para cada etapa del trabajo.

## 3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA

La asignatura se desarrolla exclusivamente en grupos de dos o tres personas, bajo la modalidad de taller a través de clases prácticas, donde se discuten las decisiones tomadas por los alumnos para diseñar y desarrollar su propuesta. Se trata de un proceso adaptativo donde se trabaja con el ciclo de vida del desarrollo de software, ajustando el modelo a los requerimientos funcionales. Los alumnos presentan informes regulares que deben ser aprobados a medida que avanza en la

asignatura.

En cuanto a la forma de evaluación, los alumnos deberán presentar su trabajo a través de una exposición oral acompañada de un informe escrito. Si bien la presentación será grupal, la nota será individual en función del esfuerzo y capacidades desarrolladas por cada alumno. Se busca que sea capaz de justificar sus decisiones debiendo explicar no sólo lo que hizo, sino también cuáles fueron los motivos por los que seleccionaron las herramientas empleadas. Es importante analizar cómo se expresa cada alumno tanto en forma oral como escrita y la calidad del trabajo en cuanto a prolijidad, orden y facilidad de comprensión. Durante el curso, tendrán oportunidad para corregir los aspectos que sean necesarios según las indicaciones de los docentes.

Las distintas etapas de avance del documento deben ser terminadas y entregadas según el cronograma preestablecido al inicio del curso. Este documento junto con un 40% del desarrollo de la aplicación habilitan al alumno a obtener la regularidad.

Quienes hayan aprobado la regularidad deberán continuar avanzando en el desarrollo de la aplicación. La calificación final dependerá de la calidad del trabajo escrito, de la funcionalidad y completitud del código (con al menos el 80% de los requerimientos cubiertos) y de la defensa oral que el grupo y cada alumno realice.

Aunque la materia es cuatrimestral se responden consultas en ambos cuatrimestres para facilitar la tarea de los alumnos que, habiendo obtenido la regularidad, deban aprobar el final.

## REQUERIMIENTOS ESPECÍFICOS PARA OBTENER LA REGULARIDAD

Al finalizar el cuatrimestre cada grupo deberá entregar una copia del documento en formato digital y haber publicado su aplicación en un repositorio establecido por la cátedra.

1. Se deberá entregar un documento que describa claramente el trabajo realizado con especial énfasis en la descripción del proceso, incluyendo:
  - Documento de requerimientos aprobado.
  - Definición de cómo se aplicó la metodología.
  - Documentación técnica de las etapas desarrolladas. Incluirá los diagramas realizados durante el análisis y el diseño, y las principales pantallas de la aplicación.
2. Copia del software desarrollado a un 40% de su funcionalidad, según los requerimientos definidos para dicho trabajo.
3. Demostración grupal frente a la cátedra y sus compañeros del software realizado.

## REQUERIMIENTOS ESPECÍFICOS PARA APROBAR LA MATERIA

Una vez obtenida la regularidad, el grupo podrá acceder a la aprobación de la materia. Para esto cada grupo deberá entregar una copia del documento y de la aplicación tanto en formato digital como físico, quedando este último en resguardo de la Universidad y disponible para futuros alumnos.

1. Al documento entregado para la regularidad deberá agregarse:
  - Las principales herramientas utilizadas y una breve descripción de cada una.
  - Un capítulo donde se reflexione sobre los cambios que haya sufrido el proyecto, por ejemplo en sus iteraciones, pero también es de interés analizar el error en las estimaciones de tiempo y todo lo que haga a la evolución del proceso de desarrollo.
  - Una presentación visual de apoyo para la exposición oral.

- Descripción técnica de los requerimientos y pasos necesarios para la instalación del software desarrollado.

2. Copia del software desarrollado al menos en un 80% de su funcionalidad, según los requerimientos definidos para dicho trabajo.

3. Inscripción individual a una mesa de exámenes, previo acuerdo con la cátedra.

4. Defensa oral grupal frente al jurado conformado en la mesa de exámenes.

## **4. CONTENIDOS DE LA ASIGNATURA**

Contenidos mínimos:

Desarrollo específico integrando conocimientos teóricos y herramientas conocidas por el alumno.

A continuación se detallan los contenidos de la asignatura:

### Unidad 1. Definición del tema del trabajo

Los alumnos deberán definir grupos de dos o tres personas y proponer a la cátedra un trabajo de interés o problema que quieran resolver. La definición del desarrollo a realizar deberá ser determinada de común acuerdo entre la cátedra y los alumnos involucrados. Se pondrá especial interés en aquellas aplicaciones de utilidad para la comunidad o que cuenten con usuarios reales concretos. La cátedra definirá un conjunto de propuestas alternativas para que los grupos que no hayan definido la suya, puedan optar por desarrollar. En el caso de alumnos que no formen grupos, la cátedra los integrará en un grupo y definirá el trabajo a realizar.

### Unidad 2. Especificación de los Requerimientos

Elaboración de un documento donde se definan específicamente los requerimientos del sistema, acordes a la expectativa de la cátedra.

### Unidad 3. Metodología de Desarrollo

Cada grupo definirá una metodología para el desarrollo de software, a partir de las ya estudiadas en materias anteriores. Luego la utilizará durante todo el desarrollo del proyecto.

### Unidad 4. Análisis y diseño del sistema

En función de la definición del problema a resolver que cada grupo haya comprometido, se deberán establecer los mecanismos de análisis y diseño pertinentes. Según el tipo de trabajo esta etapa puede incluir diagrama de clases, entidad-relación, secuencia, actividad, casos de uso y todas las herramientas de análisis y diseño que sean necesarias para especificar técnicamente el trabajo a realizar. A la vez que deberán elegir y justificar adecuadamente las herramientas de trabajo empleadas. A partir de esta etapa, si la metodología definida en la Unidad 3 implica trabajar con iteraciones, será obligación reflejarlas tanto en el desarrollo de la aplicación como en el documento a entregar.

### Unidad 5. Desarrollo de la Aplicación

La programación del sistema se llevará a cabo en los lenguajes acordados entre el grupo y la cátedra, en función del trabajo a realizar y la experiencia previa de los alumnos. Se deberán utilizar herramientas colaborativas y de versionado, como GIT u otras.

### Unidad 6. Validación

Cada iteración del sistema deberá ser validada tanto a nivel de usuario como así también

respecto de los requerimientos funcionales a los que responda.

## 5. RECURSOS NECESARIOS

- Proyector

## 6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
1	Unidad 1	Definición del tema de trabajo.	
2	Unidad 1 y 2	Definición del tema de trabajo. Especificación de requerimientos.	
3	Unidad 3	Definición de metodología de desarrollo.	
4	Unidad 4	Análisis y diseño.	
5	Unidad 4	Análisis y diseño.	
6	Unidad 4	Análisis y diseño.	
7	Unidad 4 y 5	Análisis y diseño. Desarrollo y codificación.	
8	Unidad 5	Desarrollo y codificación.	
9	Unidad 5	Desarrollo y codificación.	
10	Unidad 4, 5 y 6. Entrega parcial	Entrega parcial para revisión del análisis y diseño. Validación. Correcciones en la codificación y/o modelo.	
11	Unidad 4 y 5	Análisis y diseño. Desarrollo y codificación.	
12	Unidad 5	Desarrollo y codificación.	
13	Unidad 5	Desarrollo y codificación.	
14	Unidad 5 y 6. Entrega parcial	Entrega parcial para revisión funcional. Validación. Correcciones en la codificación.	
15	Unidad 5 y 6	Desarrollo y codificación. Validación.	
16	Entrega final	Entrega final para obtener la regularidad de la materia. En caso de estar en condiciones de promocionar, deberá inscribirse en la próxima mesa de exámenes.	

## 7. BIBLIOGRAFIA DE LA ASIGNATURA

Autor	Año	Título	Capítulo/s	Lugar de la Edición	Editor / Sitio Web
Ian Sommerville	2012	Ingeniería del Software, 9ª edición	Libro completo		

Roger Pressman	2015	Ingeniería de Software, 8ª edición	Libro completo		
Kendall & Kendall	2011	Análisis y diseño de sistemas, 8ª edición	Libro completo		

-----  
Firma del docente-investigador responsable

VISADO		
COORDINADOR DE LA CARRERA	DIRECTOR DEL INSTITUTO	SECRETARIO ACADEMICO UNTDF
Fecha :	Fecha :	

**Este programa de estudio tiene una validez de hasta tres años o hasta que otro programa lo reemplace en ese periodo**