

**Nombre de asignatura**  
**Planificación Ciclo lectivo 2022**

Datos administrativos de la asignatura			
Departamento:	<b>Sistemas de Información</b>	Carrera	INGENIERÍA EN SISTEMAS DE INFORMACIÓN
Asignatura:	Ingeniería de Software		
Nivel de la carrera	4	Duración	96
Bloque curricular:	Sistemas de Información		
Carga horaria presencial semanal:	6	Carga Horaria total:	96
Carga horaria no presencial semanal (si correspondiese)		% horas no presenciales (si correspondiese)	
Profesor/es Titular/Asociado/Adjunto:	Mg. Vanzetti Juan José	Dedicación:	Simple
Auxiliar/es de 1º/JTP:	Ing. Tosselli Laura	Dedicación:	Simple

**Presentación, Fundamentación**

El desarrollo de software es un proceso complicado que reviste varias etapas, comenzando con la definición del problema y siguiendo por el análisis del requerimiento, arquitectura de bajo y alto nivel, diseño, la construcción, el testeo, implementación y mantenimiento. En este curso se aplicarán los conceptos básicos de planificación, ingeniería de requerimientos, construcción de software y prueba

- **Relación de la asignatura con el perfil de egreso.** Dado que el ingeniero de sistemas de información debe ser un profesional de sólida formación analítica que le permite la resolución de problemas, esta asignatura le aportará diversos conocimientos para encarar dichas tareas.
- **Relación de la asignatura con los alcances del título.** Con los conocimientos obtenidos el alumno podrá aplicarlos en la especificación, gestión, desarrollo tanto de sistemas de información / software, como así también en lo concerniente al establecimiento de normas de calidad para el desarrollo y gestión del software

**Relación de la asignatura con las competencias de egreso de la carrera**

Detallar, en la tabla siguiente, la relación de la asignatura con las competencias de egreso específicas, genéricas tecnológicas, sociales, políticas y actitudinales de la carrera. Indicar a cuáles competencias de egreso tributa (aportes reales y significativos de la asignatura) y en qué nivel (0=no tributa, 1=bajo, 2=medio, 3=alto). Agregar un comentario general de justificación.

(Este detalle se integrará en una matriz de tributación de la carrera, dictada en la Facultad Regional, en la cual se explicita el desarrollo de las competencias específicas y genéricas de la carrera y el nivel en que tributa cada asignatura).

Competencias específicas de la carrera (CE)	Competencias genéricas tecnológicas (CT)	Competencias genéricas sociales, políticas y actitudinales (CS)
CE1: Especificar, proyectar y desarrollar de información (Alta)	CT1: Gestionar, planificar y desarrollar proyectos de ingeniería (Alta)	CS1: Desempeñarse de manera efectiva en equipos de trabajo(Alta)
CE2: Especificar, proyectar y desarrollar software(Alta)	CT2:Utilizar de manera efectiva las técnicas y herramienta de aplicación en la ingeniería	CS2:Comunicarse con efectividad(Alta)
CE3: Proyectar y dirigir lo referido a la seguridad informática(Media)		CS3:Aprender en forma continua y autónoma (Baja)
CE4: Establecer métricas y normas de calidad de software(Alta)		
CE5: Certificar el funcionamiento, condición de uso o estado de sistemas de información (Alta)		
CE6: Dirigir y controlar la implementación, operación y mantenimiento de sistemas de información (Media)		
CE7: Dirigir y controlar la implementación, operación y mantenimiento de sistemas de software (Media)		

### Propósito

El objetivo de la ingeniería de software es lograr productos de software de calidad, mediante un proceso apoyado por métodos y herramientas, para diseñar aplicaciones informáticas que se ajusten a las necesidades de las organizaciones. Así mismo, permita dirigir y coordinar el desarrollo de aplicaciones complejas, intervenir en todas las fases del ciclo de vida de un producto. Como así también, estimar los costos de un proyecto y determinar los tiempos de desarrollo.

### Objetivos establecidos en el Diseño Curricular

- Conocer los componentes de un proyecto de ingeniería de software
- Conocer los estándares asociados a la calidad del proceso de desarrollo de software y de productos
- Conocer los componentes de un plan de aseguramiento de la calidad

- Emplear las métricas que se aplican el desarrollo de software
- Aplicar los elementos de un proceso de prueba("Testing)
- Diseñar un plan de prueba unitario y de integración

### Resultados de aprendizaje

- RA1: Identificar los componentes de un proyecto de desarrollo de software.
- RA2: Distinguir las ventajas y desventajas de los diferentes ciclos de vida
- RA3: Describir las inactividades de la gestión de configuración de software
- RA4: Comparar las diferentes herramientas utilizadas en la gestión de configuración de software
- RA5: Analizar los principales modelos de calidad de software
- RA6: Utilizar técnicas relacionadas con el aseguramiento de la calidad del proceso y del producto con el objetivo de desarrollar un software de calidad
- RA7: Identificar y describir métricas del software

### Asignaturas correlativas previas

Para cursar debe tener cursada:

- Probabilidades y Estadísticas
- Diseño de Sistemas
- Gestión de Datos

Para cursar debe tener aprobada:

- Análisis de sistemas
- Sintaxis y Semántica de los Lenguajes
- Paradigmas de Programación

Para rendir debe tener aprobada:

- Probabilidades y Estadísticas
- Diseño de Sistemas
- Gestión de Datos

### Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

- Proyecto Final

### Programa analítico, Unidades temáticas

#### 1- Ingeniería de Software.

- Concepto de Software .Naturaleza, Evolución, Importancia y Problemas del Software. Características. Mitos. Crisis del Software.
- Definición de Ingeniería de Software. Evolución.
- El proceso de desarrollo de software. Características. Ciclo de vida del software. Evaluación del proceso. Ventajas y desventajas .Modelos de proceso software. Objetivo de los modelos.Standard ISO/IEC 12207.Desarrollo ágil
- Componentes de un proyecto de Sistemas de información.
- Vinculo proceso-proyecto-producto en la gestión de un proyecto de desarrollo de software

#### 2- Gestión de configuración de software

- Gestión del Cambio. Gestión de Configuración. Concepto. Versiones, reléase. Patrones de rama. Línea Base.
- Planificación de la Gestión de Configuración de Software
- Elementos de con figuración del Software.
  - Identificación de Objetos en la Configuración de Software
  - Control de Versiones
  - Control de Cambios
- Continuous Integration
- Continuous Delivery

- Continuous Deployment. Estrategias.

### **3- Calidad del software**

- Calidad: concepto y visiones Garantía de calidad (SQA).
- Modelos de calidad: McCall, Boehm, ISO9126.SPICE Medición de la calidad. Mediciones de McCall. Concepto de defecto. Medidas de aspectos de la calidad: Medición de densidad de defectos, Mediciones de usabilidad, Mediciones de mantenimiento. Inspecciones de software.
- Introducción al CMMI. Nivel inicial. Niveles de madurez. Evolución del proceso. Estructura del CMMI. Áreas clave del proceso por nivel. Áreas clave del proceso: estructura. Usando el CMMI. Ejemplo de área clave de proceso

### **4- Métricas de proceso y proyecto**

- Métricas en dominicos del proceso y proyecto
- Métricas del software
- Métricas para calidad del software
- Integración de las métricas dentro del proceso de software
- Establecimiento de un programa de métricas del software

### **5- Estimación para proyecto de software**

- Estimación de proyecto de software
- Técnicas de descomposición
- Modelos empíricos de estimación
- Estimación para proyectos orientados a objetos

### **6- Verificación y Validación**

- Proceso, planeamiento y estrategias testing.
- Técnicas de Prueba
- Prueba de los programas.
- Prueba del sistema.

### **7- Auditoría y peritaje**

- Introducción. Definición de auditoría. Control interno. Tipos de auditoría. Objetivos de auditoría informática. Determinación de la información y estructura de la muestra. Organizaciones de auditorías ¿Qué es una auditoría Informática? Auditoría interna y auditoría externa. Alcance de la auditoría informática. Tipos y clases de auditorías. Técnicas y herramientas usadas por la auditoría informática

## **Metodología de enseñanza**

La cátedra contempla las siguientes estrategias metodológicas:

Exposición oral del profesor y los ayudantes de cátedra en las cuales se enfatizara el uso de la lectura anticipada de los temas a tratar, elaboración de mapas conceptuales e integración de contenidos.

El dictado de la materia se complementa con la presentación y análisis de publicaciones, presentación y análisis de videos, resolución de guías prácticas y la elaboración de un trabajo de investigación.

#### **Clases teóricas**

Durante el dictado de las clases teóricas se pretende que el alumno vaya adquiriendo de manera gradual conocimientos referidos a la Ingeniería de Software de acuerdo a diferentes aspectos tratados en las unidades temáticas

#### **Clases Prácticas**

Por medio de las actividades prácticas planteadas por el docente, se procura la comprensión y asimilación de los conceptos teóricos planteados por el mismo. Durante el desarrollo de todas las unidades temáticas se plantearán actividades prácticas que promuevan el desarrollo de aptitudes del alumno en la materia.

En la siguiente tabla se detallan los porcentajes de la carga horaria afectadas a las actividades de formación teórica y prácticas.

Actividades de formación	Carga Horaria
Clases teóricas	50%
Clases prácticas	50%
Total de horas	96

### Recomendaciones para el estudio

Describir las principales recomendaciones que se les pueden hacer a los/las estudiantes para abordar el aprendizaje de la asignatura, teniendo en cuenta la experiencia del cuerpo docente respecto de desarrollos anteriores.

### Metodología de evaluación

#### EVALUACIÓN

Para llevar a cabo la evaluación desde el enfoque formativo es necesario que el docente incorpore en el aula estrategias de evaluación congruentes con las características y necesidades individuales de cada alumno y las colectivas del grupo. El cumplimiento en la entrega en tiempo y forma de las actividades prácticas. En caso que las actividades contengan errores, los alumnos tendrán la posibilidad de ir corrigiéndolos a media que transcurre el cuatrimestre en entregas sucesivas. Al finalizar el cuatrimestre, los alumnos deberán haber concluido correctamente todas las actividades prácticas. Se evaluarán además a los alumnos en instancias parciales

**Momentos:** Evaluación Diagnóstica, Evaluación formativa o continua

#### Instrumentos:

- 2 parciales teóricos y 1 parcial práctico
- Resolución de guías de trabajos prácticos.

#### Criterios de:

##### A) Regularidad:

Para obtener la regularidad de la materia el alumno deberá:

- Tener el 75% de asistencias
- Tener completas y entregadas las guías de trabajos prácticos.
- Tener aprobados los parciales prácticos y teóricos con nota promedio 6 o superior.
- Podrá recuperar una instancia de parcial práctico y uno teórico.

##### B) Promoción:

Para obtener la promoción directa de la materia el alumno deberá:

- Tener el 75% de asistencias
- Tener completas y entregadas las guías de trabajos prácticos.
- Tener aprobados los parciales prácticos y teóricos con nota 8 o superior.
- Podrá recuperar una instancia de parcial práctico y uno teórico.

### Cronograma de clases/trabajos prácticos/exámenes (tentativo)

Cronograma:

Clase Nº	Día	Fecha	Tema
1	Lunes	15/08/2022	Feriado
2	Miércoles	17/08/2022	Presentación materia. Encuesta Presentación materia. Unidad 1 Software. Naturaleza .Evolución.
3	Lunes	22/08/2022	Unidad 1: Crisis del software. Definición Ingeniería de software. Características internas/externas Proceso. Proceso de desarrollo de software. Características
4	Miércoles	24/08/2022	Práctica Gestión de la configuración. Git
5	Lunes	29/08/2022	Unidad 1: Modelos de proceso de desarrollo de software. Características. Comparación. Procesos Agiles
6	Miércoles	31/08/2022	Gestión de proyecto. Áreas de la gestión. Visión PMI. Generación de un plan
7	Lunes	5/09/2022	Unidad 1: Ejercicio errores clásicos de la gestión
8	Miércoles	7/09/2022	Practico
9	Lunes	12/09/2022	Unidad 2 : Cambio. Concepto. Características. Gestión de cambios. Gestión de la configuración. Proceso
10	Miércoles	14/09/2022	Practico
11	Lunes	19/09/2022	Gestión de configuración estrategias. Integracion Continua Estrategias
12	Miércoles	21/09/2022	Practico
13	Lunes	26/09/2022	Unidad 3: Calidad. Concepto. Objetivos
14	Miércoles	28/09/2022	Practico
15	Lunes	03/10/2022	Unidad 3: Modelos de Calidad
16	Miércoles	05/10/2022	Unidad 3: CMMI
17	Lunes	10/10/2022	Feriado
18	Miércoles	12/10/2022	Practico
19	Lunes	17/10/2022	Práctica CMMI.
20	Miércoles	19/10/2022	Practica
21	Lunes	24/10/2022	Unidad 5: Estimación. Concepto. Objetivos Técnicas
22	Miércoles	26/10/2022	Realización de trabajos prácticos en clase
23	Lunes	31/10/2022	Unidad 4: Métrica. Concepto. Objetivos. Tipos
24	Miércoles	2/11/2022	Unidad 4: Programa de métricas
25	Lunes	7/11/2022	Unidad 4: Métrica. Ejercitación
26	Miércoles	9/11/2022	Unidad 6: Testing. Concepto. Objetivos
27	Lunes	14/11/2022	Unidad 6: Testing. Tipos de pruebas
28	Miércoles	16/11/2022	Unidad 6: Testing. Ejercitación
29	Lunes	21/11/2022	Feriado.
30	Miércoles	23/11/2022	Unidad 6: Testing. Ejercitación
31	Lunes	28/11/2022	Unidad 7 Peritaje
32	Miércoles	30/11/2022	Practico

**Recursos necesarios**
**Materiales curriculares (recursos)**

 Recopilación de apuntes de la Cátedra  
 Retroproyector, proyección de videos

**Referencias bibliográficas (citadas según Normas APA)**

**a) Obligatoria o básica:**

- ✓ Ingeniería de Software. 9ma Edición. Ian Sommerville. Pearson - Addison Wesley. 2011
- ✓ Ingeniería de Software. Un enfoque práctico. Roger Pressman. McGraw Hill. 2010.
- ✓ Auditoria en Informática 2da Edición. Jose Antonio Echenique Garcia. Mc Graw Hill

**Función Docencia**

**Reuniones de asignatura y área**

**Atención y orientación a las y los estudiantes**

Se establecerán canales y horarios para la atención virtual, dependiendo de las necesidades de los alumnos.