

INGENIERIA DE SOFTWARE I

Año 2021

Carrera/ Plan: (Dejar lo que corresponda)

Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07
Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07
Analista Programador Universitario Plan 2015/Plan 2007
Analista en Tecnologías de la Información y la Comunicación
Plan 2017

Año: 2do**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:** Taller de Programación**Profesor/es:** : Patricia Pesado

Marcos Boracchia

Silvia Esponda

Ariel Pasini

Alejandro Gonzalez

Hs. semanales: 6hs**OBJETIVOS GENERALES**

Introducir al alumno en los conceptos fundamentales de la Ingeniería de Software. En particular profundizar las primeras etapas del ciclo de vida (requerimientos, análisis y diseño de sistemas). Presentar los conceptos de calidad de software y de privacidad, integridad y seguridad de datos.

El alumno desarrollará trabajos experimentales de análisis y diseño de sistemas.

RESULTADOS DE APRENDIZAJE

- 1.5. Demostrar sensibilización ante la necesidad de contar con amplios conocimientos a la hora de crear aplicaciones informáticas en otras áreas temáticas (Básico).
- 2.1. Utilizar una serie de técnicas con las que identificar las necesidades de problemas reales, analizar su complejidad y evaluar la viabilidad de las posibles soluciones mediante técnicas informáticas (Adecuado).
- 2.2. Describir un determinado problema y su solución a varios niveles de abstracción (Adecuado).
- 2.3. Seleccionar y utilizar los correspondientes métodos analíticos, de simulación y de modelización (Adecuado).
- 2.4. Escoger los patrones de solución, algoritmos y estructuras de datos apropiados (Adecuado).
- 3.1. Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos (Básico).
- 3.2. Describir las fases implicadas en distintos modelos de ciclo de vida con respecto a la definición, construcción, análisis y puesta en marcha de nuevos sistemas y el mantenimiento de sistemas existentes (Básico).
- 3.3. Elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales, así como aplicaciones emergentes (Básico).
- 3.4. Describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador (Básico).
- 4.4. Explicar la importancia de la confidencialidad de la información y cuestiones relativas a la seguridad con respecto al diseño, desarrollo, mantenimiento, supervisión y uso de sistemas informáticos (Básico).
- 5.1. Demostrar conocimientos sobre los códigos y estándares de cumplimiento del sector (Básico).
- 5.2. Describir y explicar las técnicas de gestión correspondientes al diseño, implementación, análisis, uso y mantenimiento de sistemas informáticos, incluyendo gestión de proyectos, de configuración y de cambios, así como las técnicas de automatización correspondientes (Adecuado).
- 5.3. Identificar los riesgos, incluyendo riesgos de seguridad, laborales, medioambientales y comerciales y llevar a cabo una evaluación de riesgos, reducción de riesgos y técnicas de gestión de riesgos (Adecuado).

- 6.2. Comunicar mensajes de forma efectiva tanto oralmente como por medio de otros medios de comunicación ante distintas audiencias (Básico).
- 6.4. Identificar las distintas maneras de organizar equipos y los distintos roles dentro de dichos equipos (Básico).
- 6.5. Participar de manera efectiva en grupos de trabajo informático (Básico).

COMPETENCIAS

- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT2- Concebir, diseñar y desarrollar proyectos de Informática.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- CGT6 – Capacidad para identificar y gestionar los riesgos en Informática (ambientales, laborales, de seguridad y económicos).
- CGT9 Capacidad de organización de equipos de trabajo en proyectos de Informática, definiendo los roles en los mismos.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LI- CE6 – Controlar las normas de calidad en el software o software integrado a otros componentes. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.
- LS - CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS - CE5 – Establecer métricas y normas de calidad y seguridad de software, contralando las mismas a fin de tener un producto industrial que respete las normas nacionales e internacionales. Control de la especificación formal del producto, del proceso de diseño, desarrollo, implementación y mantenimiento. Establecimiento de métricas de validación y certificación de calidad. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

- El proceso de software. Ciclos de vida de software.
- Herramientas para el proceso de software.
- Ingeniería de requerimientos. Introducción a los métodos formales.
- Metodologías de Análisis y diseño. Lenguajes de modelado.
- Sistemas de Tiempo Real.
- Conceptos de calidad de software.
- Conceptos de Teoría General de Sistemas. Definición de Sistemas de Información
- Conceptos de Privacidad, Integridad y Seguridad en Sistemas de Información

PROGRAMA ANALÍTICO

1- Conceptos de software e ingeniería de software.

- Evolución del software. Características. Componentes.

-
- Definición de Ingeniería de Software. Evolución.
 - Software de alta calidad.
 - Ingeniería de Sistemas.
- 2- Procesos del Software.**
- El significado de proceso.
 - Modelos de proceso. Modelo de cascada.
 - Iteración de procesos. Modelos incrementales. Modelos Evolutivos.
 - Prototipación. Metodologías ágiles. Desarrollo basado en componentes.
 - Métodos formales. Desarrollo orientado a aspectos. Proceso unificado.
 - Actividades del Proceso. Especificación. Diseño. Implementación. Validación. Evolución.
 - Herramientas y técnicas para modelado de procesos.
 - Ingeniería de Software Asistida por computadora.
- 3- Ingeniería de Requerimientos.**
- El proceso de requerimientos.
 - Tipos de requerimientos. Requerimientos funcionales, no funcionales, del usuario, del sistema.
 - Características de los requerimientos. Obtención y análisis de los requerimientos.
 - Técnicas de comunicación. Los problemas de la comunicación. Elicitación de requisitos. Entrevistas, cuestionarios, JAD, brainstorming.
 - Validación de requerimientos. Gestión de requerimientos. Medición de requerimientos.
 - Documentos de Especificación de requerimientos. STD 830.
- 4- Modelos del Sistema.**
- Modelo de contexto, de comportamiento, de datos, de objetos.
 - Técnicas de especificación de requerimientos: estáticas, dinámicas, relacionales, orientadas a estados, formales.
 - Tablas de Decisión, Diagramas de Transición de Estados, Redes de Petri.
 - Casos de Uso, Historias de Usuarios, Diagramas de Flujos de Datos, Diagramas de Flujos de Control, UML.
 - Prototipado de los requerimientos. Técnicas de construcción rápida.
- 5- Calidad.**
- Conceptos de Calidad y Calidad Total.
 - Calidad del proceso y del producto.
 - Aseguramiento y estándares de calidad.
 - Planeamiento de la calidad.
 - Control de la calidad.
 - Revisiones Técnicas formales. Inspecciones.
 - Modelos de madurez para las organizaciones de desarrollo de software (CMM-CMMI).
 - Normas ISO 9000.
- 6- Sistemas de Información.**
- Teoría General de Sistemas.
 - Clasificación de los sistemas organizacionales y de información administrativa.
 - Privacidad, Integridad y Seguridad.

BIBLIOGRAFÍA

- Ingeniería de Software. 9na Edición. Ian Sommerville. Pearson. 2011.
- Software Engineering: Theory and Practice. 4th Edition. Shari Pfleeger. Prentice Hall. 2009. (Edición en castellano: Ingeniería de Software. Teoría y Práctica. Shari Pfleeger. Pearson Education. 2002)
- Ingeniería de Software. Un enfoque práctico. 7ma Edición. Roger Pressman. McGraw-Hill. 2010.
- Sistemas de Información Administrativa. Murdick R. Prentice Hall. 1988

- Systems Analysis and Design, 9/E. Kendall & Kendall. Pearson. 2013. (Edición en castellano: Análisis y diseño de sistemas. 8va Edición. Kendall & Kendall. Pearson. 2011)

METODOLOGÍA DE ENSEÑANZA MODALIDAD PRESENCIAL

El curso consta de clases teóricas, explicaciones de práctica y clases prácticas. Las explicaciones de práctica apuntan a brindar las herramientas necesarias para la realización de los trabajos prácticos.

Las actividades prácticas comprenden la realización de ejercicios de modelización de requerimientos aplicando diferentes herramientas explicadas en la teoría.

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real, especificación de los mismos como problemas resolubles desde la informática y en el desarrollo de soluciones verificables para los mismos utilizando las técnicas de especificación de requerimientos establecidas en el programa de la materia

Los ejercicios de la práctica son evaluados teniendo en cuenta el contenido técnico, pero también la estructura, organización, sintaxis, claridad conceptual.

Además los alumnos deben realizar una actividad grupal de elicitation de requerimientos que es evaluada de manera oral por el ayudante a cargo.

Todas las evaluaciones orales realizadas se reflejan en planillas escritas que conforman documentación de evaluación del trabajo.

EVALUACIÓN

Los alumnos obtienen la cursada aprobando dos exámenes prácticos en los cuales se evalúan, en cada uno, tres (3) temas. Los mismos cuentan con un recuperatorio donde se evalúan los temas no aprobados en el parcial. Además se cuenta con un parcial integrador, al final de la cursada, en caso de no lograr la aprobación de los seis (6) temas de los dos (2) parciales.

Para la aprobación final de la asignatura los alumnos tienen dos posibilidades:.

Alumnos por promoción:

Deben concurrir al 80% de las clases teóricas.

Deben rendir y aprobar dos evaluaciones teóricas con nota 6 o superior (cada una de ellas con una recuperación).

Alumnos regulares:

Deben rendir y aprobar un examen final.

METODOLOGIA DE ENSEÑANZA - MODALIDAD VIRTUAL

La asignatura utiliza la plataforma IDEAS para interactuar con los alumnos del curso.

CLASES TEÓRICAS

La teoría se realiza de manera asincrónica. Cada 15 (quince) días se dan consultas sincrónicas utilizando un servicio de Videoconferencias.

Una vez por semana según cronograma se realiza y sube un powerpoint a la plataforma con el desarrollo de la teoría de esa semana.

Además se sube un video generado a partir del powerpoint, con audio explicando dicho powerpoint.

Una vez por semana se propone en la plataforma una autoevaluación con preguntas cuyo objetivo es que el alumno pueda establecer si ha entendido el contenido teórico propuesto en la semana.

CLASES PRACTICAS:

Se graban las explicaciones prácticas y serán subidas a la plataforma IDEAS para que los estudiantes puedan acceder al material de consulta. De igual forma, las guías prácticas son subidas a IDEAS.

Las consultas se realizan semanalmente utilizando el servicio de Videoconferencias en 3 (tres) turnos que el alumno podrá elegir, teniendo en cuenta la capacidad del turno.

EVALUACIONES:

Los alumnos obtienen la cursada aprobando un examen práctico, con dos recuperatorios. El examen es tomado utilizando el servicio de Videoconferencias.

Para la aprobación final de la asignatura los alumnos tienen dos posibilidades:.

Alumnos por promoción:

Debe haber realizado el 100% de las autoevaluaciones propuestas.

Deben rendir y aprobar una evaluación teórica con nota 6 o superior (con una recuperación).

Alumnos regulares:

Deben rendir y aprobar un examen final.

CRONOGRAMA DE CLASES Y EVALUACIONES (Tentativo)

Semana	Teoría	Práctica
23/8	Conceptos de I.S.	
	Fases genéricas	
30/8	Técnicas de Comunicación	Técnicas de elicitación
06/9	Técnicas de Comunicación-Tablas de decisión	Técnicas de elicitación
13/9	Requerimientos-DTE	Tablas de decisión
20/9		DTE
27/9	Requerimientos-Redes de Petri	DTE + RP
04/10	Casos de Uso- Historias de Usuarios	
11/10		
18/10	DFD-DFC	RP + CU
25/10	Modelos de Procesos	Casos de Uso- HU
01/11	Metodologías Ágiles	Historias de Usuario
08/11	Calidad y Sistemas de Información	Consulta
15/11	Repaso	Consulta
22/11		
29/11	Repaso	Consulta
01/12	Repaso	

Evaluaciones previstas	Semana
Primera Evaluación Teórica	20/9
Primer Parcial Práctico (TE + TD + DTE)	27/9
Segundo Parcial Práctico (RP + CU + HU)	25/10
Primer Recuperatorio del Primer Parcial Práctico (TE + TD + DTE)	01/11
Segunda Evaluación Teórica	08/11
Primer Recuperatorio del Segundo Parcial Práctico (RP + CU + HU)	22/11
Recuperatorio Primera Evaluación Teórica	30/11
Recuperatorio Global (TE + TD + DTE + RP + CU + HU)	1/12
Recuperatorio Segunda Evaluación Teórica)	8/12

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

www.ideas.info.unlp.edu.ar
sesponda@info.unlp.edu.ar
marcosb@lidi.info.unlp.edu.ar



Silvia Esponda
Firma del/los profesor/es

INGENIERIA DE SOFTWARE I
Redictado**Año 2021****Carrera/Plan:**

Licenciatura en Informática Planes 2003-2007-2012-2015

Licenciatura en Sistemas Planes 2003-2007-2012-2015

Analista Programador Universitario Plan 2007-2015

Analista en TIC Plan 2017

Año: 2°**Régimen de cursada:** Semestral**Carácter:** Obligatoria**Correlativas:**

Taller de Programación

Profesores: Patricia Pesado

Marcos Boracchia

Silvia Esponda

Ariel Pasini

Alejandro Gonzalez

Hs.semanales: 6 hs.**OBJETIVOS GENERALES**

Introducir al alumno en los conceptos fundamentales de la Ingeniería de Software. En particular profundizar las primeras etapas del ciclo de vida (requerimientos, análisis y diseño de sistemas). Presentar los conceptos de calidad de software y de privacidad, integridad y seguridad de datos. El alumno desarrollará trabajos experimentales de análisis y diseño de sistemas.

RESULTADOS DE APRENDIZAJE

- 1.5. Demostrar sensibilización ante la necesidad de contar con amplios conocimientos a la hora de crear aplicaciones informáticas en otras áreas temáticas (Básico).
- 2.1. Utilizar una serie de técnicas con las que identificar las necesidades de problemas reales, analizar su complejidad y evaluar la viabilidad de las posibles soluciones mediante técnicas informáticas (Adecuado)
- 2.2. Describir un determinado problema y su solución a varios niveles de abstracción (Adecuado).
- 2.3. Seleccionar y utilizar los correspondientes métodos analíticos, de simulación y de modelización (Adecuado).
- 2.4. Escoger los patrones de solución, algoritmos y estructuras de datos apropiados (Adecuado).
- 3.1. Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos (Básico).
- 3.2. Describir las fases implicadas en distintos modelos de ciclo de vida con respecto a la definición, construcción, análisis y puesta en marcha de nuevos sistemas y el mantenimiento de sistemas existentes (Básico).
- 3.3. Elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales así como aplicaciones emergentes (Básico).
- 3.4. Describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador (Básico).
- 4.4. Explicar la importancia de la confidencialidad de la información y cuestiones relativas a la seguridad con respecto al diseño, desarrollo, mantenimiento, supervisión y uso de sistemas informáticos (Básico).
- 5.1. Demostrar conocimientos sobre los códigos y estándares de cumplimiento del sector (Básico).
- 5.2. Describir y explicar las técnicas de gestión correspondientes al diseño, implementación, análisis, uso y mantenimiento de sistemas informáticos, incluyendo gestión de proyectos, de configuración y de cambios, así como las técnicas de automatización correspondientes (Adecuado).

- 5.3. Identificar los riesgos, incluyendo riesgos de seguridad, laborales, medioambientales y comerciales y llevar a cabo una evaluación de riesgos, reducción de riesgos y técnicas de gestión de riesgos (Adecuado)
- 6.2. Comunicar mensajes de forma efectiva tanto oralmente como por medio de otros medios de comunicación ante distintas audiencias (Básico).
- 6.4. Identificar las distintas maneras de organizar equipos y los distintos roles dentro de dichos equipos (Básico).
- 6.5. Participar de manera efectiva en grupos de trabajo informático (Básico).

COMPETENCIAS

- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT2- Concebir, diseñar y desarrollar proyectos de Informática.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- CGT6 – Capacidad para identificar y gestionar los riesgos en Informática (ambientales, laborales, de seguridad y económicos).
- CGT9 Capacidad de organización de equipos de trabajo en proyectos de Informática, definiendo los roles en los mismos.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LI- CE6 – Controlar las normas de calidad en el software o software integrado a otros componentes. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.
- LS - CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS - CE5 – Establecer métricas y normas de calidad y seguridad de software, contralando las mismas a fin de tener un producto industrial que respete las normas nacionales e internacionales. Control de la especificación formal del producto, del proceso de diseño, desarrollo, implementación y mantenimiento. Establecimiento de métricas de validación y certificación de calidad. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

- El proceso de software. Ciclos de vida de software.
- Herramientas para el proceso de software.
- Ingeniería de requerimientos. Introducción a los métodos formales.
- Metodologías de Análisis y diseño. Lenguajes de modelado.
- Sistemas de Tiempo Real.
- Conceptos de calidad de software.
- Conceptos de Teoría General de Sistemas. Definición de Sistemas de Información
- Conceptos de Privacidad, Integridad y Seguridad en Sistemas de Información

PROGRAMA ANALÍTICO

1- Conceptos de software e ingeniería de software.

- Evolución del software. Características. Componentes.
- Definición de Ingeniería de Software. Evolución.
- Software de alta calidad.
- Ingeniería de Sistemas.

2- Procesos del Software.

- El significado de proceso.
- Modelos de proceso. Modelo de cascada.
- Iteración de procesos. Modelos incrementales. Modelos Evolutivos.
- Prototipación. Metodologías ágiles. Desarrollo basado en componentes.
- Métodos formales. Desarrollo orientado a aspectos. Proceso unificado.
- Actividades del Proceso. Especificación. Diseño. Implementación. Validación. Evolución.
- Herramientas y técnicas para modelado de procesos.
- Ingeniería de Software Asistida por computadora.

3- Ingeniería de Requerimientos.

- El proceso de requerimientos.
- Tipos de requerimientos. Requerimientos funcionales, no funcionales, del usuario, del sistema.
- Características de los requerimientos. Obtención y análisis de los requerimientos.
- Técnicas de comunicación. Los problemas de la comunicación. Elicitación de requisitos. Entrevistas, cuestionarios, JAD, brainstorming.
- Validación de requerimientos. Gestión de requerimientos. Medición de requerimientos.
- Documentos de Especificación de requerimientos. STD 830.

4- Modelos del Sistema.

- Modelo de contexto, de comportamiento, de datos, de objetos.
- Técnicas de especificación de requerimientos: estáticas, dinámicas, relacionales, orientadas a estados, formales.
- Tablas de Decisión, Diagramas de Transición de Estados, Redes de Petri.
- Casos de Uso, Historias de Usuarios, Diagramas de Flujos de Datos, Diagramas de Flujos de Control, UML.
- Prototipado de los requerimientos. Técnicas de construcción rápida.

5- Calidad.

- Conceptos de Calidad y Calidad Total.
- Calidad del proceso y del producto.
- Aseguramiento y estándares de calidad.
- Planeamiento de la calidad.
- Control de la calidad.
- Revisiones Técnicas formales. Inspecciones.
- Modelos de madurez para las organizaciones de desarrollo de software (CMM-CMMI).
- Normas ISO 9000.

6- Sistemas de Información.

- Teoría General de Sistemas.
- Clasificación de los sistemas organizacionales y de información administrativa.
- Privacidad, Integridad y Seguridad.



BIBLIOGRAFÍA

- Ingeniería de Software. 9na Edición. Ian Sommerville. Pearson. 2011.
- Software Engineering: Theory and Practice. 4th Edition. Shari Pfleeger. Prentice Hall. 2009. (Edición en castellano: Ingeniería de Software. Teoría y Práctica. Shari Pfleeger. Pearson Education. 2002)
- Ingeniería de Software. Un enfoque práctico. 7ma Edición. Roger Pressman. McGraw-Hill. 2010.
- Sistemas de Información Administrativa. Murdick R. Prentice Hall. 1988
- Systems Analysis and Design, 9/E. Kendall & Kendall. Pearson. 2013. (Edición en castellano: Análisis y diseño de sistemas. 8va Edición. Kendall & Kendall. Pearson. 2011)

CONDICIONES DE ACCESO AL RE-DICTADO

Para acceder al re-dictado de la materia, los alumnos deben haberse presentado a rendir en al menos dos fechas del parcial de Ingeniería de software I del curso dictado en el semestre anterior.

METODOLOGÍA DE ENSEÑANZA

El curso consta de clases teóricas, explicaciones de práctica y clases prácticas.

Las explicaciones de práctica apuntan a brindar las herramientas necesarias para la realización de los trabajos prácticos.

Las actividades prácticas comprenden la realización de ejercicios de modelización de requerimientos aplicando diferentes herramientas explicadas en la teoría.

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real, especificación de los mismos como problemas resolubles desde la informática y en el desarrollo de soluciones verificables para los mismos utilizando las técnicas de especificación de requerimientos establecidas en el programa de la materia, incluyendo la evaluación de riesgos y las técnicas de gestión y reducción del riesgo. Asimismo se considera en particular la confidencialidad de la información y cuestiones relacionadas con la seguridad. Los ejercicios de la práctica son evaluados teniendo en cuenta el contenido técnico, pero también la estructura, organización, sintaxis, claridad conceptual.

Además los alumnos deben realizar una actividad grupal de elicitation de requerimientos que es evaluada de manera oral por el ayudante a cargo.

Todas las evaluaciones orales realizadas se reflejan en planillas escritas que conforman documentación de evaluación del trabajo.

Las actividades podrán ser realizadas de forma presencial o mediante videoconferencia, en función de las disposiciones de la Facultad.

La asignatura utiliza la plataforma MOODLE de la Facultad de Informática y herramientas de video conferencias

EVALUACIÓN

Los alumnos obtienen la cursada aprobando un examen práctico con dos recuperatorios

Para la aprobación final de la asignatura los alumnos tienen dos posibilidades:

Alumnos por promoción:

Deben concurrir al 80% de las clases teóricas (presenciales o virtuales).

Deben rendir y aprobar una evaluación teórica con nota 6 o superior con un recuperatorio

Alumnos regulares:

Deben rendir y aprobar un examen final.

CRONOGRAMA DE CLASES Y EVALUACIONES

Semana	Teoría	Práctica
30/3	Introducción - Conceptos de I.S. Ingeniería de requerimientos	
6/4	Técnicas de Comunicación	Practica TE
13/4	Técnicas de especificación de requerimientos Historias de Usuarios	Practica HU
20/4	Explicación de HU	
20/4	Casos de Uso – TD	Práctica CU
27/4	Explicación CU	
4/5	DTE – RPs /	Practica DTE
11/5	Explicación DTE – RPs	Practica RP
18/5	MP- MA	Practica TD
25/5	Feriado	
1/6	Calidad	Primer parcial
8/6	Consulta	
15/6	Evaluación Teórica	Consultas
22/6	---	Primer recuperatorio
29/6	Consultas	
6/7	Recuperatorio Evaluación Teórica	Consultas
14/7		Segundo recuperatorio

Evaluaciones previstas	Semana
Primer parcial práctico	9/6
Evaluación Teórica	15/6
Primer recuperatorio práctico	23/6
Recuperatorio Evaluación Teórica	6/7
Segundo recuperatorio práctico	14/7

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

apasini@lidi.info.unlp.edu.ar / sesponda@lidi.info.unlp.edu.ar

Firma del/los profesor/es



Ariel Pasini