

UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

PLANIFICACIÓN ANUAL 2023

Asignatura: Ingeniería de Software

LICENCIATURA EN SISTEMAS DE INFORMACIÓN
Plan de Estudios 2011 – Innovación Curricular 2022

EQUIPO CÁTEDRA

Profesor Asociado: Msc. Ing. Margarita María Álvarez

Profesor Asociado: Msc. Ing. Liliana Figueroa

Ayudante de 1era. Categoría: Lic. Adrian Bellumini

1. IDENTIFICACIÓN

1.1. NOMBRE DE LA ASIGNATURA: INGENIERÍA DE SOFTWARE

1.2. CARRERA: LICENCIATURA EN SISTEMAS DE INFORMACIÓN

1.3. PLAN DE ESTUDIOS: 2011

1.4. AÑO ACADÉMICO: 2023

1.5. CARÁCTER: Obligatorio

1.6. UBICACIÓN DE LA ASIGNATURA EN EL PLAN DE ESTUDIOS:

1.6.1. MÓDULO: 8° - AÑO: 4°

1.6.2. TRAYECTO AL QUE PERTENECE LA ASIGNATURA

Tabla 1: Carga horaria

| TRAYECTO | CARGA HORARIA PRESENCIAL |
|--|--------------------------|
| Ciencias Básicas Generales y Específicas | |
| Algoritmos y Lenguajes | |
| Arquitectura, Sistemas Operativos y Redes | |
| Ingeniería de Software, Bases de Datos y Sistemas de Información | 75 hs. |
| Aspectos Sociales y Profesionales | |
| Otros contenidos | |
| CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR | 75 hs. |

1.6.3-CORRELATIVAS

1.6.3.1 Anteriores:

- SISTEMAS DE INFORMACIÓN I (APROBADA)
- INGLÉS II (REGULAR)

1.6.3.2. Posteriores:

- METODOLOGÍA DE LA INVESTIGACIÓN I (REGULAR)
- INGENIERÍA WEB (REGULAR)
- PRÁCTICA PROFESIONAL SUPERVISADA (REGULAR)

1.7- CARGA HORARIA:

1.7.1. **Carga horaria semanal total:** 5 (cinco) hs.

1.7.2. **Carga horaria semanal destinada a la formación práctica:** 3(tres) hs.

1.7.3 **Carga horaria total dedicada a las distintas actividades de formación práctica:** 45 (cuarenta y cinco) hs.

1.8. ÁMBITOS DONDE SE DESARROLLAN LAS ACTIVIDADES DE FORMACIÓN PRÁCTICA:

Las actividades previstas se desarrollarán en las aulas de la FECyT y en el Laboratorio de Informática del Departamento Académico de Informática.

1.9. CANTIDAD DE COMISIONES EN LAS QUE SE DICTA LA ASIGNATURA: 1

2. PRESENTACIÓN

2.1. UBICACIÓN DE LA COMO TRAMO DE CONOCIMIENTO DE UNA DISCIPLINA

Esta obligación curricular es una de las asignaturas que pertenecen al Trayecto de Ingeniería de Software, Bases de Datos y Sistemas de Información. Forma parte del campo de Formación Orientada ya que hace a la formación específica de la disciplina Informática y de los Sistemas de Información. Está orientada

fundamentalmente a brindar a los estudiantes conocimientos, habilidades y destrezas acerca de la Ingeniería de Software.

2.2. CONOCIMIENTOS Y HABILIDADES PREVIAS QUE PERMITEN ENCARAR EL APRENDIZAJE DE LA ASIGNATURA

Teniendo en cuenta los contenidos mínimos de la asignatura se estiman adecuadas las correlatividades previas, que brindan a los alumnos los conocimientos y habilidades necesarios sobre diseño e implementación de sistemas de información, en todos sus aspectos. Es por esto que, se requiere como conocimientos previos, para cursar esta asignatura, conceptos fundamentales de resolución de problemas, proceso de desarrollo de software y sistemas de información.

2.3. ASPECTOS DEL PERFIL PROFESIONAL DEL EGRESADO A LOS QUE CONTRIBUYE LA ASIGNATURA

- Profundos conocimientos, entre otros, sobre Algoritmos y Lenguajes de Programación, Arquitectura de Computadores, Sistemas Operativos y Redes de Computación, Ingeniería de Software, Base de Datos, y Sistemas de Información.
- Una sólida formación en metodología de investigación científica y sobre técnicas y procedimientos que le permiten indagar, analizar e interpretar el campo de aplicación de la Informática, con énfasis en los Sistemas de Información y Sistemas Informáticos.

Está capacitado para:

- Identificar, formular y resolver problemas de Informática.
- Gestar, diseñar y desarrollar proyectos de Informática.
- Utilizar técnicas y herramientas de aplicación en la Informática.
- Generar desarrollos tecnológicos y/o innovaciones tecnológicas.
- Comunicarse de manera efectiva en el ámbito profesional.

2.4- INTEGRACIÓN HORIZONTAL Y VERTICAL CON OTRAS ASIGNATURAS



Figura 1. Integración horizontal y vertical con otras asignaturas

3- OBJETIVOS

1. Analizar, diseñar, implementar y mantener Sistemas de Información desde la perspectiva de la Ingeniería de Software.
2. Controlar la calidad de sistemas de Software desde la perspectiva de la Ingeniería de Software.

3. Desarrollar proyectos de Software a lo largo de todo el ciclo de vida del software mediante la identificación de las fases y etapas que constituyen el desarrollo de un software.
4. Mantener software que satisfagan los requisitos del usuario y se comporten de forma fiable y eficiente, aplicando las teorías, principios, métodos y prácticas de la Ingeniería de Software.
5. Identificar problemas que hacen que un software sea candidato para aplicar la reingeniería de software.
6. Establecer dificultades de la construcción y descripción de componentes reutilizables.

4- SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

4.1 CONTENIDOS MÍNIMOS ESTABLECIDOS EN EL PLAN DE ESTUDIOS

Ingeniería de Software: Principios y componentes. El proceso de software. Modelos de proceso del software y Modelos de mejora del proceso del software. Ciclos de vida del software. Evaluación de Calidad de software: Calidad del proceso y del producto. Implementación. Mantenimiento en Ingeniería de Software: definición de mantenimiento, tipos de mantenimiento. Reingeniería de software. Tipos de reingeniería. Ingeniería inversa. Metodología de reingeniería. Reutilización: proceso de reutilización, modelo de procesos para la reutilización. Ingeniería del dominio. Ingeniería de software basada en componentes. Componentes reutilizables.

4.2. PROGRAMA SINTÉTICO SOBRE LA BASE DE LOS CONTENIDOS MÍNIMOS

Ingeniería del software. Reutilización del software. Calidad del software. Mantenimiento del software. Reingeniería del software.

4.3- ARTICULACIÓN TEMÁTICA DE LA ASIGNATURA

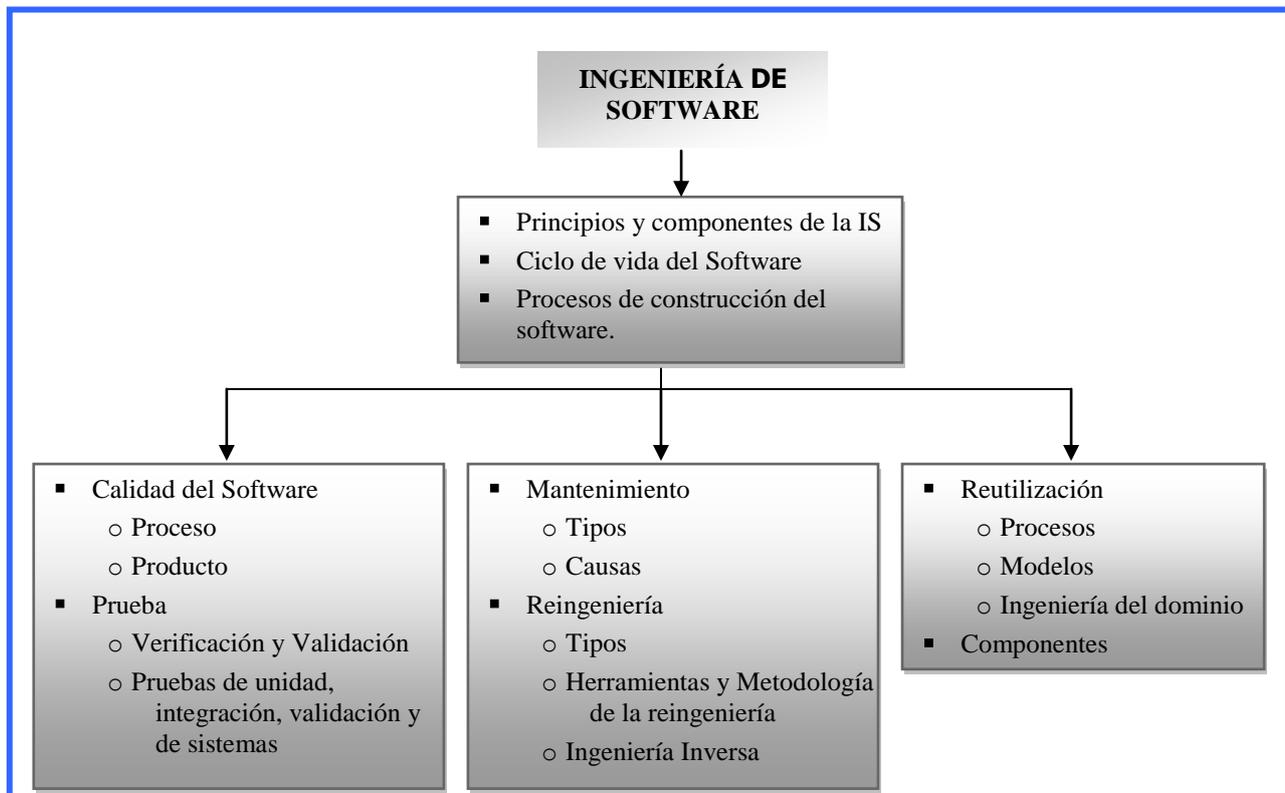


Figura 2: Articulación temática de la asignatura

4.4- PROGRAMA ANALÍTICO

Unidad 1: INGENIERÍA DEL SOFTWARE

Software. Principios y componentes de la Ingeniería de Software. Proceso de construcción de software. Modelos de proceso de software: tradicional y alternativos. Desarrollo e implementación de Software.

Modelo de madurez del software (CMM). Mejoras en el proceso del software. Niveles del proceso de madurez. Evolución del proceso del software.

Unidad 2: REUTILIZACIÓN

Concepto de reutilización. Dificultades de la reutilización. Proceso de reutilización. Modelo de procesos para la reutilización. Ingeniería del dominio. Ingeniería basada en componentes. Componentes reutilizables: definición de componentes reutilizables, descripción de componentes reutilizables: clasificación enumerada, por facetas, de atributos y valores.

Unidad 3: CALIDAD DEL SOFTWARE

Calidad. Concepto de calidad de software. Calidad del proceso y del producto. Garantía de calidad de software. Revisiones. Revisiones técnicas formales. Estándar de calidad: ISO 9001. Plan de garantía de calidad. Factores que afectan la calidad. Métricas para calidad del software.

Prueba. Enfoque estratégico de la prueba de software. Verificación. Validación. Aspectos estratégicos. Pruebas de unidad. Prueba de integración. Prueba de validación. Prueba de sistemas.

Unidad 4: MANTENIMIENTO DEL SOFTWARE

Crisis del software. Estado del arte en el mantenimiento del software. Definición de mantenimiento. Características de los sistemas que afectan el esfuerzo del mantenimiento. Naturaleza del mantenimiento. Tipos de mantenimiento. Problemas del mantenimiento. Mantenibilidad. Medidas de la mantenibilidad del Software.

Unidad 5: REINGENIERÍA DEL SOFTWARE.

Reingeniería de Procesos de negocio. Reingeniería de software. Opciones para reducir el mantenimiento del software. Objetivos de la Reingeniería del software. Proceso de reingeniería de software. Tipos de reingeniería. Ingeniería inversa. Beneficios de la ingeniería inversa. Ingeniería inversa en el desarrollo. Ingeniería inversa y la ingeniería hacia adelante. Ingeniería inversa de los datos. Ingeniería directa.

4.5- CRONOGRAMA PARA EL DESARROLLO DE LAS UNIDADES TEMÁTICAS

Tabla 2: Cronograma para el desarrollo teórico de las unidades temáticas

| UNIDAD | CARGA HORARIA | CRONOGRAMA DE DICTADO |
|---|---------------|-----------------------|
| Unidad 1: Modelos de proceso de software. Modelo de madurez del software. | 2 hs. | 1° semana |
| Unidad 1: Desarrollo e implementación de Software. | 2 hs. | 2° semana |
| Unidad 2: Reutilización. Conceptos. Modelo de procesos para la reutilización. Ingeniería del dominio. | 2 hs. | 3° semana |
| Unidad 2: Ingeniería basada en componentes. Componentes reutilizables | 2 hs. | 4° semana |

| UNIDAD | CARGA HORARIA | CRONOGRAMA DE DICTADO |
|---|---------------|-----------------------|
| Unidad 3: Calidad del Software. Garantía de calidad de software. Revisiones. Revisiones técnicas formales. | 2 hs. | 5° semana |
| Unidad 3: Calidad del Software. Estándar de calidad: ISO 9001. Plan de garantía de calidad. Factores que afectan la calidad. Métricas para calidad del software. | 2 hs. | 6° semana |
| Unidad 3: Calidad del Software. Prueba. Enfoque estratégico de la prueba de software. Verificación. Validación. Aspectos estratégicos | 2 hs. | 7° semana |
| Unidad 4: Mantenimiento del Software. Crisis del software. Estado del arte en el mantenimiento del software. Definición de mantenimiento. | 2 hs. | 8° semana |
| Unidad 4: Mantenimiento del Software. Naturaleza del mantenimiento. Tipos de mantenimiento. Problemas del mantenimiento. | 2 hs. | 9° semana |
| Unidad 4: Mantenimiento del Software. Mantenibilidad. Medidas de la mantenibilidad del Software. | 2 hs. | 10° semana |
| Unidad 5: Reingeniería del Software. Reingeniería de Procesos de negocio. Reingeniería de software. Opciones para reducir el mantenimiento del software. Objetivos de la Reingeniería del software. | 2 hs. | 11° semana |
| Unidad 5: Reingeniería del Software. Proceso de reingeniería de software. Tipos de reingeniería. Ingeniería inversa. Beneficios de la ingeniería inversa. | 2 hs. | 12° semana |
| Unidad 5: Reingeniería del Software. Ingeniería inversa en el desarrollo. Ingeniería inversa y la ingeniería hacia adelante. Ingeniería inversa de los datos. Ingeniería directa. | 2 hs. | 13° semana |
| TOTAL | 30 hs. | |

5. FORMACIÓN PRÁCTICA

5.1. DESCRIPCIÓN DE LAS ACTIVIDADES DE FORMACIÓN PRÁCTICA

Durante el cursado los alumnos realizan tres talleres que les permiten:

- 1- Interpretar problemas del mundo real relacionados con la aplicación de la Ingeniería de Software.
- 2- Intervenir de manera efectiva para resolver problemas del mundo real relacionados con la aplicación de la Ingeniería de Software.
- 3- Desarrollar habilidades prácticas en actividades de carácter tecnológico, experimentales y de resolución de problemas que los acerquen a la realidad de las organizaciones de Software.
- 4- Interpretan la realidad a través del diagnóstico y análisis de problemas y, promueven la intervención crítica en actividades de carácter tecnológico para resolver problemas de la Ingeniería de Software.

TALLER A: Desarrollo de Software

El taller se ha organizado en dos partes:

Parte 1. Ejercicios para analizar casos y determinar el ciclo de vida más apropiado y, propuestas por parte del estudiante de ejemplos de casos para cada ciclo de vida del software.

Con la realización de las actividades se persiguen los siguientes objetivos:

- Identificar la existencia de un problema y el modelo apropiado para su solución.
- Diferenciar entre proceso del software y ciclo de vida del software.
- Distinguir y aplicar los criterios en base a los cuales se debe elegir el ciclo de vida más apropiado a un proyecto de software.

PARTE 2: DESARROLLO DE SOFTWARE

A.2.1. CONTENIDOS: los conceptos teóricos sobre los cuales se sustentará la actividad están vinculados con los siguientes temas vistos en las asignaturas previas:

- Ingeniería de Requerimientos del software
- El proceso de diseño.
- Base de Datos relacionales: modelado de datos. Normalización
- Algoritmos y estructuras de datos. Programación orientada a objetos

A.2.2. OBJETIVOS

Desarrollar proyectos de Software a largo de todo el ciclo de vida del software mediante la identificación de las fases y etapas que constituyen el desarrollo de un software.

Que el alumno logre:

- Interpretar pautas metodológicas y documentación técnica de los proyectos de software.
- Desarrollar e implementar el software diseñado.
- Desarrollar componentes reutilizables.
- Implementar y administrar Bases de Datos u otra estructura de datos utilizada en el software.
- Habilidad identificar recursos tecnológicos necesarios para el desarrollo de software (lenguajes de programación, gestores de bases de datos, utilitarios y paquetes de software).
- Capacidad para desenvolverse en entornos de desarrollo de software libre o propietario.
- Capacidad para trabajar en grupo que le permita llevar a cabo el desarrollo del software.
- Ser capaz de comprender la responsabilidad ética de sus funciones en el desarrollo del software.
- Fundamentar el impacto social del software desarrollado en el contexto global y local.

A.2.3. METODOLOGÍA

Esta actividad consta del desarrollo del software que fue diseñado en las asignaturas de Sistemas de Información I y II, para lo cual se debe:

- 1- Realizar los ejercicios de Modelos de ciclo de vida del software.
- 2- Revisar la documentación de análisis y diseño del sistema.
- 3- Seleccionar las funciones a desarrollar.
- 4- Planificar y gestionar el proyecto de desarrollo de software.
- 5- Identificar y seleccionar recursos tecnológicos necesarios para el desarrollo de software (lenguaje de programación y el gestor de bases de datos).
- 6- Codificar las funciones y crear la Base de Datos.
- 7- Desarrollar un componente reutilizable.
- 8- Reflexionar y documentar sobre el impacto social del software desarrollado.
- 9- Elaborar informe final.

A.2.4. CRITERIOS DE EVALUACIÓN

Los criterios que se utilizan para la evaluación son:

- a) Completitud de las funciones seleccionadas para el desarrollo.
- b) Correcto funcionamiento de las funciones desarrolladas.
- c) Detección y depuración de errores.
- d) Facilidad de uso.

- e) Diseños de recursos gráficos, pantallas y mensajes apropiados para una correcta interacción.
- f) Completitud en la documentación, manual de usuario y de instalación del software desarrollado
- g) Presentación del Programa fuente y ejecutables e instalador.
- h) Correcta aplicación de las herramientas CASE.

TALLER B: Plan de garantía de calidad

B.1. CONTENIDOS: los contenidos temáticos necesarios para abordar la actividad son los descritos en la unidad 3. A continuación se detalla sintéticamente, los más importantes:

- Plan de garantía de calidad.
- Métricas para calidad del software.
- Revisiones: técnicas formales.
- Prueba de unidad, de integración, de validación y de sistemas.

B.2. OBJETIVOS

Que el alumno logre:

- Realizar un plan de garantía de la calidad.
- Aplicar los distintos tipos de prueba en el software realizado en la actividad *Desarrollo de Software* descrita en el apartado 5.1-Taller A.
- Usar herramientas Case durante la prueba. Con el uso de estas herramientas se pretende que los alumnos desarrollen las siguientes Competencias:
 - Optimizar el desarrollo de proyectos software con el uso de herramientas de software libre.
 - Habilidad para identificar recursos tecnológicos necesarios que den soporte a la calidad del software.
 - Capacidad para desenvolverse en entornos de desarrollo de software libre o propietario.
 - Fomentar en grupo de desarrollo una comunidad colaborativa entorno al desarrollo de una aplicación.
 - Reflexionar y documentar sobre el impacto social del software desarrollado de mala calidad.

B.3. METODOLOGÍA

1. Realizar el estudio y análisis exploratorio sobre los distintos temas.
2. Realizar el plan de garantía.
3. Ejecutar el plan de garantía.
4. Probar el software mediante el uso de la herramienta CASE.
5. Elaborar el informe escrito con la documentación de cada una de las etapas realizadas.

B.4. CRITERIOS DE EVALUACIÓN

Los criterios que se utilizan para la evaluación son:

- a. Completitud del plan de garantía.
- b. Correcta aplicación de las herramientas CASE.
- c. Cantidad y pertinencia de las pruebas realizadas al software desarrollado.
- d. Completitud en el manual de usuario y de instalación.

TALLER C: Mantenimiento del Software

C.1. CONTENIDOS: los contenidos temáticos necesarios para abordar la actividad son los descritos en la unidad 4. A continuación se detalla sintéticamente, los más importantes:

- Mantenimiento.
- Tipos de mantenimiento.

- Medidas de la mantenibilidad del Software.

C.2. OBJETIVOS

Que el alumno logre:

- Identificar tipos de mantenimiento.
- Mantener el software realizado en la actividad *Desarrollo de Software* descrita en el apartado 5.1 Taller A, de manera tal que satisfagan los requisitos del usuario y se comporten de forma fiable y eficiente.

C.3. METODOLOGÍA

1. Realizar el estudio y análisis exploratorio sobre los distintos temas.
2. Realizar el mantenimiento de acuerdo al tipo de mantenimiento detectado.

C.4. CRITERIOS DE EVALUACIÓN

Los criterios que se utilizan para la evaluación son:

- a) Completitud de las operaciones a las que se realizó el mantenimiento.
- b) Funcionamiento correcto de las operaciones realizadas.
- c) Diseños de pantalla adecuados y recursos gráficos convenientes.

CONSIDERACIONES GENERALES

1. Sobre el Desarrollo de las Actividades

Para el desarrollo de estas actividades previstas en los talleres se debe tener en cuenta los siguientes considerandos:

- Conformar un grupo de desarrollo integrado por estudiantes (no más de cuatro alumnos).
- Durante la realización, el grupo de desarrollo deberá realizar informes de avance sobre el desarrollo a la cátedra, en cada uno de los encuentros que se han programado.
- La presentación de los trabajos se realizará en varias entregas.
- Los trabajos serán valorados con Aprobado – Desaprobado.
- Si no logra satisfacer los requisitos solicitados, tendrá la oportunidad de una nueva presentación teniendo en cuenta las condiciones ya establecidas. La nueva fecha de presentación será fijada por la cátedra en acuerdo con los estudiantes.

2. Sobre la Instancia de Presentación de los Talleres

Se efectuará una defensa oral en el que el grupo responsable exponga y aclararen dudas sobre los trabajos, así como los conocimientos generales que se adquirieron durante su desarrollo. Se evaluará, principalmente, que los alumnos conozcan el problema y la solución propuesta y que exposición sea clara y ordenada.

3. Sobre el Cronograma de los Talleres

La presentación de los trabajos se realizará por etapas en el mes de noviembre.

4. Sobre la Responsabilidad del Equipo Catedra

- Las actividades teóricas vinculadas con los talleres estarán a cargo de los docentes de la asignatura. En estas actividades se desarrollarán discusión de conceptos esenciales para respaldar su aplicación práctica.
- Las actividades que comprenden el estudio y manejo de los recursos tecnológicos estarán a cargo del Auxiliar Docente quien desarrollará actividades relacionados con el manejo de herramientas tecnológicas y además el seguimiento del desarrollo del software.
- Se prevé que los docentes atiendan las consultas e inquietudes que se planten relacionadas con las temáticas de los talleres.

5. Criterios de evaluación: a continuación se enuncian los criterios de evaluación que se considerarán en los tres talleres:

1. Entrega del informe en tiempo y forma: se utilizará la siguiente criterios de evaluación:

- a. Organización de la información y pertinencia de los temas incluidos.
 - b. Correcta redacción y uso adecuado del estilo de escritura que facilite la comprensión.
 - c. Utilización de ejemplos acordes al tema en cuestión
 - d. Inclusión de fuentes de información o referencias acorde a las normas APA.
2. Coloquio individual: Se evaluará en este caso la correcta expresión oral y la solvencia para responder las preguntas, utilizando los siguientes criterios:
 - a) Respuesta de forma clara y precisa a las preguntas o comentarios.
 - b) Utilización correcta de técnicas de comunicación oral.
 - c) Utilización de vocabulario técnico.
 3. Trabajo en grupo y planificación y gestión del proyecto: se evaluarán de acuerdo a los siguientes criterios de evaluación:
 - a) Realización de los acuerdos iniciales de trabajo (formas y frecuencias de las reuniones).
 - b) Realización de la planificación del proyecto identificando actividades y recursos de desarrollo, definición de la calendarización y especificación de roles de cada integrante.

5.2.-FORMACIÓN EN EJES TRANSVERSALES

En la tabla siguiente se relaciona para cada uno de los ejes transversales las actividades prácticas seleccionadas con los resultados de aprendizaje.

Tabla 3: Formación en Ejes Transversales

| EJE | ACTIVIDADES | RESULTADOS DE APRENDIZAJE | GRADO DE PROFUNDIDAD EN EL TRATAMIENTO |
|--|--|--|--|
| Identificación, formulación y resolución de problemas de informática | <ul style="list-style-type: none"> • Taller A • Taller B | <ul style="list-style-type: none"> • Identificar problemas que requieran soluciones software y establecer los modelos de ciclo de vida más apropiados. • Justificar la elección de modelos en casos hipotéticos. • Identificar las funciones que requieren de mantenimiento. | Alto |
| Concepción, diseño y desarrollo de proyectos de informática | <ul style="list-style-type: none"> • Taller A • Taller B | <ul style="list-style-type: none"> • Desarrollar el software • Reusar componentes de software • Desarrollar componentes reusables • Desarrollar nuevas funciones o adaptar funciones existentes a partir de requerimientos de mantenimiento. | Alto |
| Gestión, planificación, ejecución y control de proyectos de informática | <ul style="list-style-type: none"> • Taller A • Taller B • Taller C | <ul style="list-style-type: none"> • Definir un plan de construcción de software, de mantenimiento y de garantía de la calidad. • Gestionar los proyectos de construcción de software, de mantenimiento y de garantía de la calidad. • Ajustar los planes realizados de acuerdo a los desvíos. | Medio |
| Utilización de técnicas y herramientas de aplicación en la informática | <ul style="list-style-type: none"> • Taller A • Taller B • Taller C | Instalar y utilizar adecuadamente: <ul style="list-style-type: none"> • Lenguajes de programación. • Herramienta para generación de código. • Gestores de BD relacionales • Herramientas CASE para pruebas. • Software de Soporte para documentación. • Software de Seguimiento y control de proyectos software. • Softwares de control de seguimiento y versiones. | Alto |

| EJE | ACTIVIDADES | RESULTADOS DE APRENDIZAJE | GRADO DE PROFUNDIDAD EN EL TRATAMIENTO |
|--|--|---|--|
| Generación de desarrollos tecnológicos y/o innovaciones tecnológicas. | <ul style="list-style-type: none"> • Taller A • Taller B • Taller C | <ul style="list-style-type: none"> • Determinar recursos tecnológicos necesarios para el desarrollo de software | Alto |
| Fundamentos para el desempeño en equipos de trabajo | <ul style="list-style-type: none"> • Taller A • Taller B • Taller C | <ul style="list-style-type: none"> • Crear el equipo de trabajo. • Planificar las actividades. • Asignar roles, responsabilidades y tareas a cada integrante. • Evaluar el desempeño de los integrantes del equipo de trabajo. | Alto |
| Fundamentos para la comunicación efectiva | <ul style="list-style-type: none"> • Taller A • Taller B • Taller C | <ul style="list-style-type: none"> • Elaborar informes técnicos del desarrollo, mantenimiento y el plan de garantía de calidad del software, organizando la información adecuadamente de forma tal de facilitar la comprensión a los destinatarios, utilizando los recursos gráficos y los medios necesarios para comunicar de forma efectiva la información, con adecuado uso de las normas gramaticales y con bibliografía actualizada y debidamente referenciada. • Exponer los trabajos organizando la información para facilitar la comprensión a los destinatarios, utilizando recursos gráficos y con técnicas de comunicación oral. | Alto |
| Fundamentos para la acción ética y responsable. | <ul style="list-style-type: none"> • Taller A • Taller C | Comprender sus responsabilidades éticas para el desarrollo del software. | Medio |
| Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local | <ul style="list-style-type: none"> • Taller A • Taller C | Reflexionar y documentar sobre el impacto social del software desarrollado en el contexto local. | Medio |

5.3 CRONOGRAMA DE FORMACIÓN PRÁCTICA

| ACTIVIDAD | CARGA HORARIA | CRONOGRAMA DE DESARROLLO |
|---------------------------|---------------|--------------------------------|
| Taller A – Parte 1 | 3 hs. | 1° semana |
| Taller A – Parte 2 | 18 hs. | 2°, 3°, 4°, 5°, 6° y 7° semana |
| Taller B | 12 hs. | 8°, 9°, 10° y 11° semana |
| Taller C | 12 hs. | 12°, 13°, 14° y 15° semana |
| TOTAL | 45 hs. | |

Tabla 4: Cronograma para el desarrollo de las Actividades Prácticas

6. BIBLIOGRAFÍA

| TÍTULO | AUTOR(ES) | EDITORIAL | EJEMPLARES DISPONIBLES | AÑO DE EDICIÓN |
|--|-----------------------------------|-----------|------------------------|----------------|
| Ingeniería de software | Pantaleo Guillermo | Alfaomega | 1 | 2015 |
| Calidad en el desarrollo de software | Pantaleo Guillermo | Alfaomega | 1 | 2011 |
| Desarrollo de software dirigido por modelos: | García Molina Jesús, García Rubio | Alfaomega | 1 | 2014 |

| TÍTULO | AUTOR(ES) | EDITORIAL | EJEMPLARES DISPONIBLES | AÑO DE EDICIÓN |
|--|--|---|------------------------|----------------|
| conceptos, métodos y herramientas | Felix, Pelechano Vicente, Vallecillo Antonio, Vara Juan Manuel, Vicente Cristina Chicote | | | |
| Fábricas de software: experiencias, tecnologías y organización | Mario Piattini Velthuis y Javier Garzás Parra | Alfaomega Ra-Ma | 1 | 2007 |
| Ingeniería de software. Un enfoque Práctico | Pressman, Roger | 6° Edición. Edit. McGraw Hill. ISBN: 9789701054734 | 1 | 2005 |
| Ingeniería de software | Sommerville Ian | 7° Edición. Edit. Pearson. Addison Wesley ISBN: 84-7829-074-5 | 1 | 2005 |

7. ESTRATEGÍAS METODOLÓGICAS

7.1. ASPECTOS PEDAGÓGICOS Y DIDÁCTICOS

A fin de lograr los objetivos definidos, se considera de fundamental importancia la auto-gestión, por parte del estudiante, fundamentalmente por el tramo de la carrera en el que se encuentran. Para alcanzar la mencionada auto-gestión, la Cátedra proveerá el primer día de clase:

- el programa detallado de la asignatura con la bibliografía asociada,
- el cronograma detallado de actividades y
- las condiciones de regularidad de la asignatura y la metodología para la realización y aprobación de los Trabajos.

El docente, antes que un rol de expositor, se desempeñará como consultor o moderador de las discusiones que se puedan plantear, al tratar los distintos tópicos. Para ello el alumno deberá leer con anterioridad, los temas a tratar.

Las clases en todos los casos serán de carácter teórico-prácticas, con exposición de temas teóricos y planteo de problemas y/o ejemplos. Asimismo, dado que se trata de alumnos de cuarto año se les encomendará la tarea de presentar problemas y sus soluciones (en base a los métodos y técnicas estudiadas), enfocando la situación, desde el punto de vista, de cómo debe presentarse un profesional ante un eventual cliente, a fin de “defender y vender” su producto.

En las clases prácticas la técnica metodológica por excelencia será el trabajo grupal que permite promover la construcción compartida del conocimiento y lograr así no sólo la apropiación activa del mismo por parte de los miembros del grupo, sino también la indispensable socialización del estudiante, ya que toda su vida deberá transcurrir en contacto y en cooperación con sus semejantes. Además, para la formación práctica se emplea principalmente la estrategia didáctica de Aprendizaje basada en proyectos.

La articulación e integración teoría y práctica supone la alternancia, sucesión y predominio de momentos teóricos y prácticos. Se incluyen ejercitación, ejemplificación y resolución de problemas de modo de integrar los conceptos con la praxis.

La formación en los ejes transversales se desarrollará en los talleres: Taller A: Desarrollo de software, Taller B: Mantenimiento del software y Taller C: Plan de garantía de calidad.

7.2- MECANISMOS PARA LA INTEGRACIÓN DE DOCENTES

Se listan las siguientes actividades a desarrollar, como equipo interdisciplinario de docentes:

- Reuniones periódicas del trayecto curricular, para detectar problemas académicos y pronta solución.

- Elaborar un plan de seguimiento y mejora para el año siguiente, en funciones de la problemática anterior detectada y el curso de acción tomado.
- Solicitar a las autoridades de la institución, actividades de capacitación para los docentes del trayecto, de modo de profundizar la investigación y el perfeccionamiento académico de los docentes.

7.3- RECURSOS DIDÁCTICOS

Se utilizarán como recursos didácticos:

- Bibliografía actualizada (libros y publicaciones científicas). Estos se utilizarán como una manera de acercar a los alumnos a los avances producidos dentro de la disciplina; como una forma de que el alumno adquiera habilidad para sintetizar e integrar informaciones e ideas; como un medio para que conozcan distintas perspectivas y valoraciones en el área de la Ingeniería de Software, y desarrollen una actitud de apertura hacia nuevas ideas, logrando así una comprensión informada de la ciencia y la tecnología.
- Tiza, pizarrón, computadora y proyector. Estos se utilizarán para las clases expositivas del docente.
- Software libre para el desarrollo de distintas actividades prácticas.
- Los recursos disponibles en los Laboratorios de Informática.

8. EVALUACIÓN

8.1. EVALUACIÓN DIAGNÓSTICA

No se prevé.

8.2- EVALUACIÓN FORMATIVA

La evaluación formativa es de carácter continuo y está más dirigida a evaluar el proceso de enseñanza-aprendizaje, por lo que se llevará a cabo durante todo el desarrollo de la asignatura.

8.3. EVALUACIONES PARCIALES

8.3.1- Programa de Evaluaciones Parciales

La cátedra no ha previsto evaluaciones parciales, planteando esquemas de evaluación en cada uno de los talleres que se han planificado.

Las presentaciones son en grupo, pero se evalúa individualmente a cada estudiante.

Si en la evaluación de los talleres no se logra satisfacer los requisitos de aprobación, tendrá la oportunidad de una nueva presentación que se considera como Recuperatorio. La nueva fecha de presentación será fijada por la cátedra en acuerdo con los estudiantes.

8.3.2- Criterios de Evaluación

Los criterios de evaluación a aplicar en la evaluación de los talleres y recuperatorios están expresados en forma genérica y son:

- Adecuado manejo de los conceptos vinculados con la temática de los talleres.
- Capacidad para sintetizar e integrar los conocimientos vinculados.
- Cumplimiento de las consignas solicitadas.
- Entrega de los trabajos en tiempo y forma.

8.3.3- Escala de Valoración

La escala de valoración a emplear: para las evaluaciones de los talleres y los recuperatorios será cuantitativa politómica: (Escala: 1 al 10. Se aprueba con 6).

8.4- EVALUACIÓN INTEGRADORA

- No se prevé.

8.5- EVALUACIÓN SUMATIVA

Del resultado de cada una de las **EVALUACIONES** previstas se obtiene:

EVALUACIÓN SUMATIVA referida a la puntuación-calificación que se le da a cada alumno sobre la base de los resultados de las evaluaciones de los talleres.

8.5.1- CONDICIONES PARA LOGRAR LA PROMOCIÓN SIN EXAMEN FINAL DE LA ASIGNATURA

- No se prevé.

8.5.2- CONDICIONES PARA LOGRAR LA REGULARIDAD DE LA ASIGNATURA

- Presentar y aprobar según las condiciones establecidas los Talleres evaluativos.
- Cumplir con las condiciones de alumno regular establecidos en el reglamento de alumnos.

8.6- EXAMEN FINAL

La evaluación final será escrita u oral sobre los temas incluidos en la programación analítica de la asignatura.

8.7- EXAMEN LIBRE

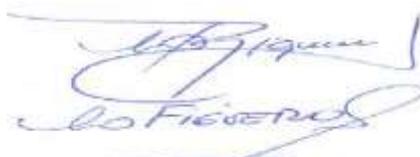
Los alumnos libres deberán cumplir las siguientes etapas, cada una de ellas eliminatória.

1ra. Etapa) Presentar los talleres equivalentes a los que realizan los alumnos regulares. Los trabajos del alumno libre deberán ser presentados a la cátedra al menos con siete días antes de la fecha del examen. Si la cátedra aprueba los trabajos presentado por el alumno libre, está en condiciones de pasar a la segunda instancia del examen final.

2a etapa) Aprobar una evaluación oral de tipo teórica.



Msc. Ing. Margarita María Álvarez



Msc. Ing. Liliana María Figueroa