

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

PLANIFICACIÓN ANUAL 2024

Asignatura: Ingeniería de Software

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

EQUIPO CÁTEDRA

Profesor Titular: Msc. Ing. Margarita María Álvarez

Profesor Titular: Msc. Ing. Liliana Figueroa

Ayudante de 1era. Categoría: Lic. Adrián 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- Innovación Curricular 2022

1.4. AÑO ACADÉMICO: 2024

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)
- INGLES 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

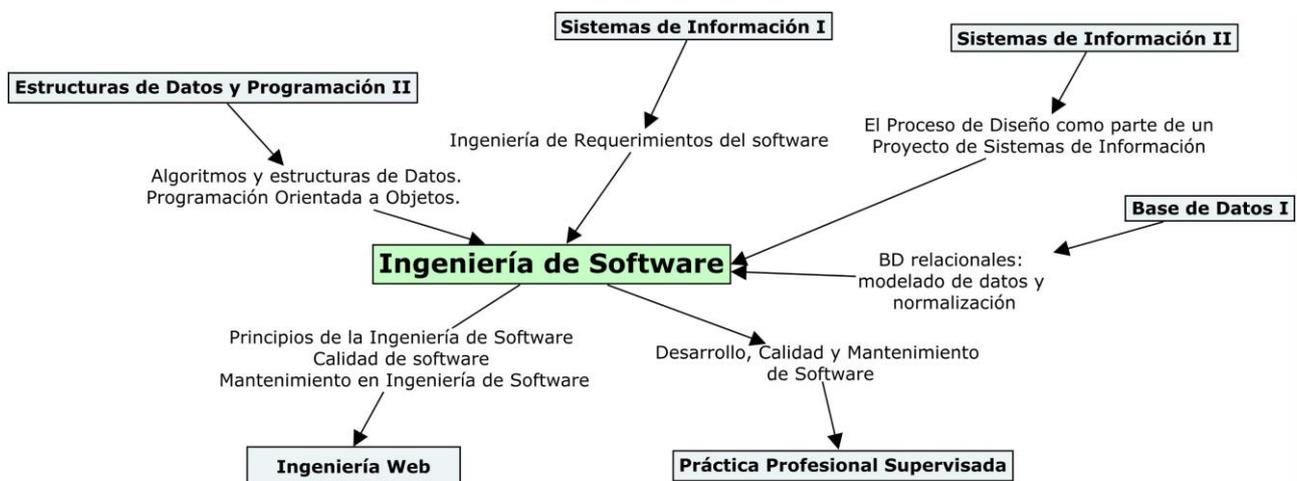


Figura 1. Integración horizontal y vertical con otras asignaturas

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

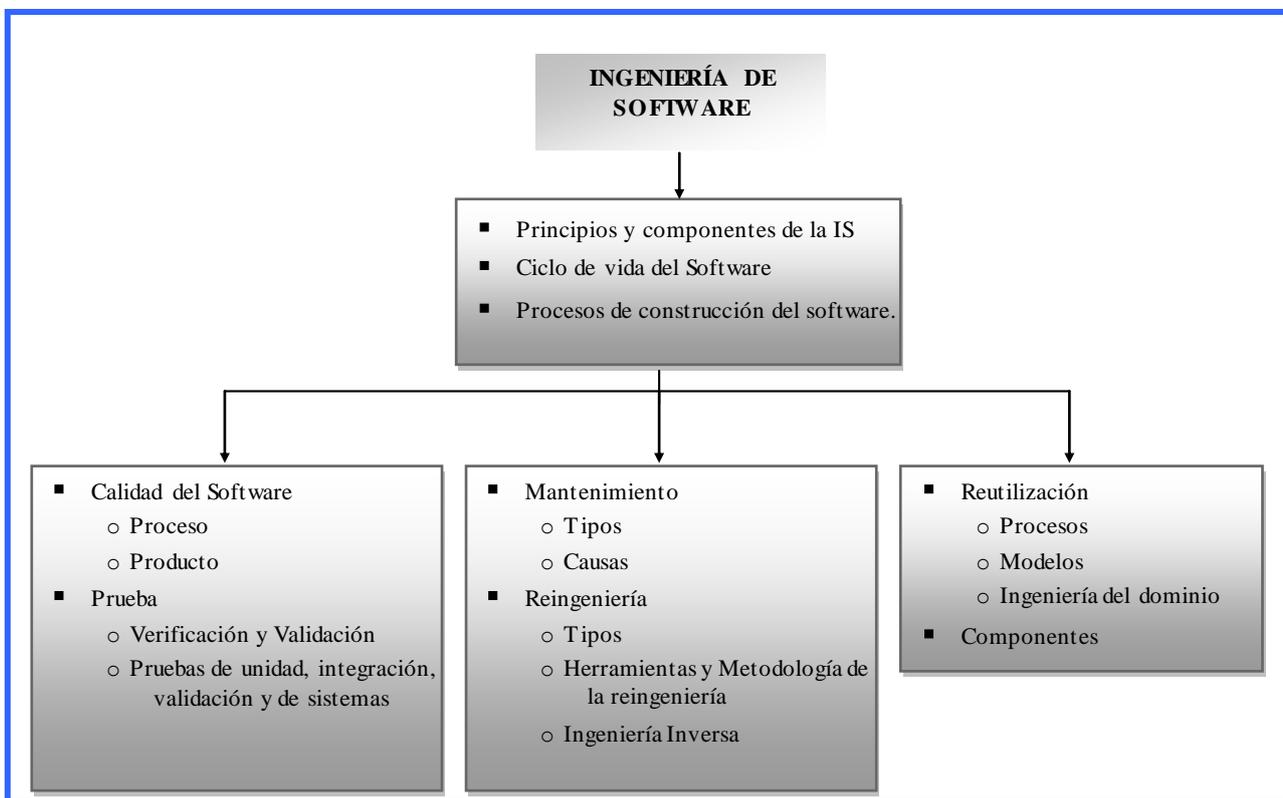


Figura 2: Articulación temática de la asignatura

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- Interpretar la realidad a través del diagnóstico y análisis de problemas y, promover 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.

A.1.1. CONTENIDOS: Software – Proceso Software – Ciclo De Vida De Proceso Software

A.1.2. OBJETIVOS

Con la realización de las actividades que se plantean, 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.

A.1.3. ACTIVIDADES

- 1- Analizar los casos hipotéticos planteados en los enunciados y realizar:
 - Describir el software y plantear en términos generales sus funciones.
 - Justificar el modelo de ciclo de vida que elegiría para su desarrollo.
- 2- Proponer ejemplos de casos /problemas, analizarlos y determinar el ciclo de vida más apropiado.

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 un proyecto de Software a largo de todo su ciclo de vida 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 para 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 equipos 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. ACTIVIDADES

Desarrollar el software que fue diseñado en las asignaturas de Sistemas de Información I y II, para lo cual se debe:

- Revisar la documentación técnica de análisis y diseño del sistema realizado en las asignaturas Sistemas de Información I y Sistemas de Información II.
- Seleccionar las funciones a desarrollar.
- Seleccionar las funciones que justifiquen su reusó para integrarlo en el software a desarrollar. Para ello: identificar componentes candidatos, localizarlos en los repositorios, extraerlos e integrarlos.
- Crear la Base de Datos previstas en el diseño de la solución software para el almacenamiento de los datos.
- Identificar y seleccionar recursos tecnológicos necesarios para el desarrollo de software.
- Codificar las funciones utilizando técnicas de programación orientada a objetos con aplicación de patrones arquitectónicos y de diseño de software.
- Desarrollar y catalogar un componente reutilizable.

- Establecer y aplicar normas para abordar aspectos de las responsabilidades éticas en el desarrollo de software. Tales como: privacidad de los usuarios, cumplimiento de la ley de protección de datos en Argentina, diseño para la inclusión y accesibilidad para personas con diferentes capacidades, etc.
- Planificar de desarrollo del software. Para ello: definir los objetivos y alcances del Plan, definir las actividades y recursos necesarios, definir el cronograma de actividades y asignar responsables para cada tarea.
- Gestionar el Plan de Desarrollo software, para lo cual: realizar reuniones periódicas del estado del proyecto, comparar la fecha de inicio real con la fecha de inicio planeada para cada tarea de proyecto, determinar los desfases y ajustar el plan para concluir en la fecha prevista.
- Reflexionar y documentar sobre el impacto social del software desarrollado. Para ello, realizar:
 - Revisar la documentación del ámbito de aplicación del software desarrollado.
 - Identificar los principales beneficiarios del software.
 - Definir el impacto social en el ámbito de la organización que se espera que el software tenga.
 - Definir indicadores concretos y cuantificables para determinar el impacto. tales como:
 - i. **Usuarios Activos:** cantidad de personas que se espera que usen el software.
 - ii. **Accesos:** Porcentaje de personas que, gracias al software, tienen acceso a servicios o información a la que antes no podían acceder.
 - iii. **Tasa de Adopción:** Velocidad a la que el software se espera que sea adoptado por la comunidad objetivo.
 - iv. **Costos-Beneficios:** relación costo beneficio gracias al uso del software.
 - Elaborar instrumentos para la recolección de las experiencias de usuario en el uso del software.
 - Realizar las mediciones y estadísticas.
 - Elaborar conclusiones y medidas para actuar en relación al impacto obtenido.
 - Documentar el impacto social y las medidas establecidas.
- Elaborar informe final técnico del desarrollo del software, de las pautas éticas aplicadas en el desarrollo de software y sobre el impacto social del software desarrollado. La información debe estar debidamente organizada de forma tal de facilitar la comprensión a los destinatarios, con la utilización de los recursos gráficos y los medios necesarios para comunicar de forma efectiva y, con un adecuado uso de las normas gramaticales y con bibliografía actualizada y debidamente referenciada.
- Exponer en grupo el trabajo realizado en el taller. Para la presentación: organice la información para facilitar la comprensión a los destinatarios, utilice los recursos gráficos y emplee adecuadamente técnicas de comunicación oral.

A.2.4. RECURSOS TECNOLÓGICOS

- **MySQL:** Sistema de gestión de bases de datos relacional, multi-hilo y multiusuario.
- **Netbeans:** entorno de desarrollo.
- **JAVA:** Lenguaje de programación
- **Control de versiones Git:** es un software de control de versiones, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.
- **Jira Software:** es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos.
- **JavaDoc:** herramientas de apoyo a la documentación. Es una utilidad de Oracle para la generación de documentación de APIs en formato HTML a partir de código fuente Java. Javadoc

es el estándar de la industria para documentar clases de Java. La mayoría de los IDEs los generan automáticamente.

A.2.5. CRITERIOS DE EVALUACIÓN

Los criterios que se utilizan para la evaluación son:

- a) Identifica correctamente problemas que requieren soluciones software, elije adecuadamente el modelo de ciclo de vida más adecuado.
- b) Completitud de las funciones desarrolladas y reusadas.
- c) Correcto funcionamiento de las funciones desarrolladas y los componentes reusables desarrollados.
- d) Integra adecuadamente componentes de software.
- e) Correcto uso de los criterios de usabilidad.
- f) Describe adecuadamente los componentes reusables para su catalogación.
- g) Completitud en la documentación del desarrollo, manual de usuario y de instalación del software desarrollado.
- h) Correcta aplicación de las herramientas desarrollo.
- i) Correcta identificación y aplicación de las normas para abordar aspectos de las responsabilidades éticas en el desarrollo de software.
- j) Completitud en las actividades realizadas para determinar el impacto social del software en la organización.

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:

- Elaborar el 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 Taller A.
- Usar herramientas CASE durante la prueba. Con el uso de estas herramientas se pretende que los alumnos desarrollen las siguientes Competencias:
 - Optimizar las pruebas del software desarrollado 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 para al desarrollo de las actividades.

B.3. ACTIVIDADES

1. Realizar el estudio y análisis exploratorio sobre los distintos temas.
2. Elaborar el Plan de Garantía de Calidad. Para ello, definir los objetivos y alcances del Plan, definir las actividades y recursos necesarios, definir el cronograma de actividades, asignar responsables para cada tarea.

3. Gestionar el Plan Garantía de Calidad. Para lo cual: realizar reuniones periódicas del estado del proyecto, comparar la fecha de inicio real con la fecha de inicio planeada para cada tarea de proyecto, determinar los desfases y ajustar los planes para concluir en la fecha prevista.
4. Elabore el plan de pruebas con una visión estratégica teniendo en cuenta el proceso de depuración.
5. Probar el software mediante el uso de la herramienta CASE.
6. Elaborar informe final técnico de verificación y validación del software. La información debe estar debidamente organizado de forma tal de facilitar la comprensión a los destinatarios, con la utilización de los recursos gráficos y los medios necesarios para comunicar de forma efectiva y, con un adecuado uso de las normas gramaticales y con bibliografía actualizada y debidamente referenciada.
7. Exponer en grupo el trabajo realizado en el taller. Para la presentación: organice la información para facilitar la comprensión a los destinatarios, utilice los recursos gráficos y emplee adecuadamente técnicas de comunicación oral.

B.4. RECURSOS TECNOLÓGICOS

De Software Libre

- **Junit:** es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera.
- **Java Code Coverage:** permite realizar pruebas de cobertura de condición automáticas, evaluando de manera exacta la efectividad de las pruebas unitarias.

B.5. CRITERIOS DE EVALUACIÓN

Los criterios que se utilizan para la evaluación son:

- Completitud del Plan de Garantía de Calidad y su correcta descripción.
- Correcta aplicación de las herramientas CASE.
- Cantidad y pertinencia de las pruebas realizadas al software desarrollado.

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. ACTIVIDADES

1. Identificar problemas a los que es necesario realizar un mantenimiento correctivo en el software desarrollado. Para ello, revise la documentación técnica de las pruebas de software realizadas.
2. Identificar problemas o nuevas necesidades de funcionalidad del software desarrollado que requieran realizar un mantenimiento perfectivo.
3. Planificar el proyecto de mantenimiento. Para ello, definir los objetivos y alcances del Plan, definir las actividades y recursos necesarios, definir el cronograma de actividades, asignar responsables para cada tarea.

4. Gestionar el Plan de Mantenimiento: realizar reuniones periódicas del estado del proyecto, comparar la fecha de inicio real con la fecha de inicio planeada para cada tarea de proyecto, determinar los desfases y ajustar los planes para concluir en la fecha prevista.
5. Realizar el mantenimiento de acuerdo al tipo de mantenimiento detectado. Para ello,
 - a. Revisar la documentación del software para realizar el mantenimiento.
 - b. Realizar el Análisis y Diseño de las funciones identificadas para el mantenimiento recurriendo a la ingeniería inversa y a la reingeniería.
 - c. Codificar las funciones siguiendo buenas prácticas de programación.
 - d. Realizar pruebas unitarias y de integración.
6. Elaborar informe final técnico de cada una de las actividades realizadas. La información debe estar debidamente organizado de forma tal de facilitar la comprensión a los destinatarios, con la utilización de los recursos gráficos y los medios necesarios para comunicar de forma efectiva y, con un adecuado uso de las normas gramaticales y con bibliografía actualizada y debidamente referenciada.
7. Exponer en grupo el trabajo realizado en el taller. Para la presentación: organice la información para facilitar la comprensión a los destinatarios, utilice los recursos gráficos y emplee adecuadamente técnicas de comunicación oral.

C.4. RECURSOS TECNOLÓGICOS

- **MySQL:** Sistema de gestión de bases de datos relacional, multi-hilo y multiusuario.
- **Netbeans :** entorno de desarrollo.
- **JAVA:** Lenguaje de programación
- **Control de versiones Git:** es un software de control de versiones, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.
- **Jira Software:** es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos.
- **JavaDoc:** herramientas de apoyo a la documentación. Es una utilidad de Oracle para la generación de documentación de APIs en formato HTML a partir de código fuente Java. Javadoc es el estándar de la industria para documentar clases de Java. La mayoría de los IDEs los generan automáticamente.

C.5. 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.

CONSIDERACIONES GENERALES PARA LOS TRES TALLERES

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 presentación 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 Cátedra

- 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.

CRITERIOS DE EVALUACIÓN PARA LOS TRES TALLERES

Los siguientes criterios de evaluación se aplican a los tres talleres en lo que respecta a la comunicación escrita y oral, al desempeño en equipos de trabajo y a la planificación y gestión de proyectos:

1. Entrega del informe en tiempo y forma: se utilizará los siguientes criterios de evaluación:
 - a. Organización coherente de la información y pertinencia de los temas incluidos.
 - b. Correcta redacción, utilización de lenguaje técnico y uso adecuado del estilo de escritura que facilite la comprensión.
 - c. Inclusión de fuentes de información o referencias acorde a las normas APA.
 - d. Presentación coherente de los resultados y conclusiones.
2. Exposición de trabajos - 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 y de vocabulario técnico.
 - c. Correcto uso de recursos gráficos y de técnicas de comunicación oral y, una adecuada organización de la información.
3. Trabajo en grupo y planificación y gestión del proyecto: se evaluarán de acuerdo a los siguientes criterios de evaluación:
 - a. Correctitud y completitud en la planificación de los proyectos identificando actividades y recursos de desarrollo, definición de la calendarización y especificación de roles de cada integrante.
 - b. Adecuado Gestión de los planes.
 - c. Adecuado nivel de involucramiento de cada miembro en las tareas asignadas.
 - d. Adecuada comunicación entre los miembros del equipo.
 - e. Realización de una autoevaluación del desempeño de cada miembro del equipo.

5.2.-FORMACIÓN EN EJES TRANSVERSALES

En la tabla siguiente se relaciona para cada uno de los ejes transversales las actividades prácticas 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	Taller A. Desarrollo de Software 1- Analizar problemas hipotéticos que requieren soluciones software.	<ul style="list-style-type: none"> • Identifique problemas que requieran soluciones software y descríbalos funcionalmente. 	Alto

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
de informática	2-Describir el software y plantear en términos generales sus funciones. 3-Justificar el ciclo de vida más apropiado para los problemas hipotéticos planteados. 4-Ejemplificar casos/problemas para cada ciclo de vida de desarrollo de software.	<ul style="list-style-type: none"> • Establezca y justifique el modelo de ciclo de vida más apropiado para el desarrollo y mantenimiento del software descrito. • Ejemplifique y justifique la elección del modelo de ciclo de vida en los casos hipotéticos. 	
	Taller C. Mantenimiento del Software 1-Identificar problemas a los que es necesario realizar un mantenimiento correctivo en el software desarrollado. 2-Identificar problemas o nuevas necesidades de funcionalidad del software desarrollado que requieran realizar un mantenimiento perfecto.	<ul style="list-style-type: none"> • Determine situaciones/problemas que justifiquen la realización del mantenimiento del software desarrollado. 	
<p>Las actividades planteadas se corresponden con la formación del eje transversal descrito ya que los estudiantes afianzan en la identificación de problemas que requieran el desarrollo de software y determinan las características para la correcta elección del modelo de ciclo de vida apropiado. Además, realizan propuestas de problemas adecuados para cada ciclo de vida. También, en el Taller C los estudiantes deben identificar problemas que requieren mantenimiento, ya sea, por defectos o porque el usuario requiere incorporar nuevos requisitos.</p>			
Concepción, diseño y desarrollo de proyectos de informática	Taller A. Desarrollo de Software 1-Desarrollar funciones de la propuesta de solución software a partir de la interpretación de la documentación de análisis y diseño del sistema realizado en las asignaturas Sistemas de Información I y Sistemas de Información II. 2-Defina y aplique criterios de usabilidad en el desarrollo del software. 3-Seleccionar las funciones que justifiquen su reusó para integrarlo en el software a desarrollar. Para ello: identificar componentes candidatos, localizarlos en los repositorios, extraerlos e integrarlos. 4-Codificar las funciones siguiendo buenas prácticas de programación. 5-Crear la Base de Datos previstas en el diseño de la solución software para el almacenamiento de los datos. 6-Desarrollar y catalogar un componente reutilizable.	<ul style="list-style-type: none"> • Desarrolle el software respetando los criterios de usabilidad. • Reúse componentes de software. • Desarrolle componentes reusables. • Cree la Base de Datos a partir de la documentación de diseño. 	Alto
	Taller C. Mantenimiento del Software 1-Identificar funciones que requieren mantenimiento correctivo y perfecto. 2-Revisar la documentación del software para realizar el mantenimiento. 3-Realizar el Análisis y Diseño de las funciones identificadas para el mantenimiento. 4-Codificar las funciones siguiendo buenas prácticas de programación.	<ul style="list-style-type: none"> • Desarrolle/ adapte funciones para satisfacer los nuevos requerimientos para el mantenimiento correctivo y perfecto. 	
	Taller B. Garantía de la calidad de Software Probar el software desarrollado y mantenido.	<ul style="list-style-type: none"> • Pruebe las funciones desarrolladas. 	

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
<p>Las actividades planteadas se corresponden con la formación del eje transversal descrito ya que los estudiantes realizan el desarrollo de un software y de las funciones que deben ser mantenidas. Para ello, los estudiantes deben revisar la documentación del sistema analizado y diseñado en asignaturas previas. Por lo que, el estudiante realiza actividades que contribuyen al acervo de conocimientos y habilidades para la concepción, diseño y desarrollo.</p>			
<p>Gestión, planificación, ejecución y control de proyectos de informática</p>	<p>Taller A. Desarrollo de Software Taller B. Plan de garantía de calidad Taller C. Mantenimiento del Software 1-Planificar: el desarrollo del software, de mantenimiento y de garantía de la calidad del software. Para ello, se debe:</p> <ol style="list-style-type: none"> a. Definir Objetivos y Alcances de los Planes. b. Definir las actividades y recursos necesarios. c. Definir el cronograma de actividades. d. Asignar responsables para cada tarea. <p>2-Gestionar los planes realizados:</p> <ol style="list-style-type: none"> a. Realizar reuniones periódicas del estado del proyecto b. Comparar la fecha de inicio real con la fecha de inicio planeada para cada tarea de proyecto c. Determinar los desfases y ajustar los planes para concluir en la fecha prevista. 	<ul style="list-style-type: none"> • Defina el Plan de Desarrollo de Software, el Plan de Mantenimiento y el Plan de Garantía de la Calidad. • Gestione el Plan de Desarrollo de Software, de Mantenimiento y de Garantía de Calidad. • Ajuste los planes realizados de acuerdo a los desvíos. 	<p>Medio</p>
<p>Las actividades planteadas se corresponden con la formación del eje transversal: <i>Gestión, planificación, ejecución y control de proyectos de informática</i> ya que, los estudiantes definen y gestionan tres planes: Plan de desarrollo de software, Plan de Mantenimiento y Plan de Garantía de la Calidad del software. Para ello, los estudiantes deben conocer las tareas involucradas, deben aplicar técnicas de planificación de proyectos informáticos y gestionarlos adecuadamente.</p>			
<p>Utilización de técnicas y herramientas de aplicación en la informática</p>	<p>Taller A. Parte 2. Desarrollo de Software Taller C. Mantenimiento del Software</p> <ul style="list-style-type: none"> • Identificar, instalar y utilizar herramientas para la construcción del software: MySQL, Netbeans, JAVA, Git, Jira Software y JavaDoc. • Identificar, instalar y utilizar plugins de la IDE para el mantenimiento. • Utilizar técnicas de programación orientada a objetos con aplicación de patrones arquitectónicos y de diseño de software. <p>Taller B. Plan de garantía de calidad</p> <ul style="list-style-type: none"> • Identificar, instalar y utilizar herramientas para que den soporte a la calidad del software tales como: Junit y Java Code Coverage. 	<p>Utilice adecuadamente las herramientas y técnicas para el desarrollo y mantenimiento del software.</p> <p>Instale y utilice adecuadamente las herramientas para las pruebas del software.</p>	<p>Alto</p>
<p>Las actividades planteadas se corresponden con la formación del eje transversal: <i>Utilización de técnicas y herramientas de aplicación en la informática</i>, ya que permite que los estudiantes logren: habilidad para identificar y utilizar los recursos tecnológicos necesarios que den soporte al desarrollo, mantenimiento y calidad del software y, capacidad para desenvolverse en entornos de desarrollo de software libre o propietario.</p>			
<p>Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.</p>	<p>Taller A. Desarrollo de Software Taller B. Plan de garantía de calidad Taller C. Mantenimiento del Software</p> <ul style="list-style-type: none"> • Desarrollar y mantener el software con los estándares de garantía de la calidad. 	<ul style="list-style-type: none"> • Desarrolle y mantenga una solución tecnológica de calidad para atender requerimientos de una organización. 	<p>Alto</p>

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
<p>El producto de las actividades realizadas en el Taller A: desarrollo de software, taller B: Garantía de la calidad de software y Taller C: mantenimiento de software, es un software de calidad que se considera un desarrollo tecnológico ya que representa aplicaciones prácticas de la tecnología para resolver problemas de las organizaciones, mejora la eficiencia operativa, y facilita la toma de decisiones en las organizaciones.</p>			
<p>Fundamentos para el desempeño en equipos de trabajo</p>	<p>Taller A. Desarrollo de Software Taller B. Plan de garantía de calidad Taller C. Mantenimiento del Software</p> <ul style="list-style-type: none"> • Trabajar en equipos. Para ello: <ul style="list-style-type: none"> • Crear el equipo de trabajo. • Realizar los acuerdos iniciales de trabajo (formas y frecuencias de las reuniones). • Revisar los planes realizados para el desarrollo de software, mantenimiento y para la garantía de la calidad del software. • Asignar roles, responsabilidades y tareas a cada integrante. • Evaluar el desempeño de los integrantes del equipo de trabajo. 	<ul style="list-style-type: none"> • Organice el grupo de trabajo para realizar las tareas de creación del grupo. • Distribuya responsabilidades y actividades a cada integrante del grupo de trabajo. • Evalúe el desempeño de los integrantes y del trabajo en equipo. 	<p>Alto</p>
<p>Para la realización de todos los talleres propuestos como actividades de formación práctica se requiere la formación y trabajo en equipo, por lo que los estudiantes afianzan la competencia de desempeño en equipos de trabajo. Se proponen tareas para la creación, asignación de roles de los integrantes y la autoevaluación del desempeño del grupo.</p>			
<p>Fundamentos para la comunicación efectiva</p>	<p>Taller A. Desarrollo de Software Taller B. Plan de garantía de calidad Taller C. Mantenimiento del Software</p> <p>Elaborar informes técnicos del desarrollo, mantenimiento y de garantía de calidad del software.</p> <p>Taller A. Desarrollo de Software Taller B. Plan de garantía de calidad Taller C. Mantenimiento del Software</p> <p>Exponer los trabajos realizados en el desarrollo, mantenimiento y garantía de calidad del software.</p>	<p>Elabore informes técnicos del desarrollo, mantenimiento y 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.</p> <p>Exponga los trabajos organizando la información para facilitar la comprensión a los destinatarios, utilizando recursos gráficos y con técnicas de comunicación oral.</p>	<p>Alto</p>
<p>Las actividades de redacción y exposición de los trabajos realizados en los talleres A, B y C se corresponden con la formación del eje transversal descripto ya que los estudiantes deben organizar adecuadamente la información, redactar correctamente, utilizar un adecuado estilo de escritura y emplear fuentes de información o referencias bibliográficas acorde a las normas APA. Además, los estudiantes deben exponer los trabajos afianzando la competencia de comunicación oral mediante la utilización correcta de técnicas de comunicación oral y de vocabulario técnico.</p>			
<p>Fundamentos para la acción ética y responsable.</p>	<p>Taller A. Desarrollo de Software</p> <ul style="list-style-type: none"> • Establecer normas para abordar aspectos de las responsabilidades éticas en el desarrollo de software. Tales como: privacidad de los usuarios, cumplimiento de la ley de protección de datos en Argentina, diseño para la inclusión y accesibilidad para 	<ul style="list-style-type: none"> • Defina las normas sobre las responsabilidades éticas para el desarrollo del software y determina su abordaje. • Aplique normas éticas en el desarrollo del software. 	<p>Medio</p>

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
	personas con diferentes capacidades, etc. <ul style="list-style-type: none"> • Desarrollar el software incluyendo las normas éticas. 		
Las actividades planteadas permitirán a los estudiantes identificar y aplicar normas éticas que fundamenten y garanticen que el software se alinea con principios morales y sociales. Esto permitirá a los estudiantes comprender sus responsabilidades éticas en el desarrollo del software.			
Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local	Taller A. Parte 2. Desarrollo de Software <ul style="list-style-type: none"> • Revisar la documentación del ámbito de aplicación del software desarrollado. • Identificar los principales beneficiarios del software. • Definir indicadores concretos y cuantificables para determinar el impacto social que se espera obtener del software en el ámbito de la organización. • Elaborar instrumentos para la recolección de las experiencias de usuario en el uso del software. • Realizar las mediciones y estadísticas. • Elaborar conclusiones y medidas para actuar en relación al impacto obtenido. • Documentar el impacto social y las medidas establecidas. 	<ul style="list-style-type: none"> • Determine una evaluación estratégica del impacto del software desarrollado en el ámbito de la organización. • Realice las mediciones y proponga medidas para actuar en relación al impacto obtenido. 	Medio
Las actividades prácticas descriptas aportan al eje transversal ya que, los estudiantes realizan tareas que permiten establecer medidas y evaluar el impacto del software desarrollado en el contexto de la organización.			

5.3 CRONOGRAMA DE FORMACIÓN PRÁCTICA

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
Taller A – Parte 1	3 hs.	1º semana
Taller A – Parte 2	18 hs.	2º, 3º, 4º, 5º, 6º y 7º semana
Taller B	12 hs.	8º, 9º, 10º y 11º semana
Taller C	12 hs.	12º, 13º, 14º y 15º 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: conceptos, métodos y herramientas	García Molina Jesús, García Rubio Felix, Pelechano Vicente, Vallecillo Antonio,	Alfaomega	1	2014

TÍTULO	AUTOR(ES)	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
	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. ESTRATEGIAS 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. En el marco de esta formación continua, los resultados de aprendizaje se evaluarán de acuerdo con los criterios de evaluación que se consignan en la siguiente Tabla.

Resultados de Aprendizaje	Criterios de evaluación
Identifique problemas que requieran soluciones software y describalo funcionalmente.	Identifica correctamente problemas que requieren soluciones software y describe detalladamente sus subsistemas principales.
Establezca y justifique el modelo de ciclo de vida más apropiado para el desarrollo y mantenimiento de los softwares descriptos.	Elige adecuadamente el modelo de ciclo de vida para los problemas hipotéticos planteado de acuerdo con el tipo de proyecto, las necesidades y los contextos específicos.
Ejemplifique y justifique la elección del modelo de ciclos de vida en los casos hipotéticos.	Ejemplifica adecuadamente problemas que requieren soluciones software y justifica correctamente el ciclo de vida.
Determine situaciones/problemas que justifiquen la realización del mantenimiento del software desarrollado.	Identifica correctamente situaciones/problemas que justifiquen la realización del mantenimiento del software desarrollado.
Desarrolle el software respetando los criterios de usabilidad.	Desarrolla en forma completa las funciones del software e incorpora los criterios de usabilidad en las interfaces de usuario. Funciona correctamente el software desarrollado.
Reúse componentes de software.	Busca exhaustivamente los componentes que reusará. Integra adecuadamente los componentes en el software.
Desarrolle componentes reusables	Desarrolla en forma completa componentes de software para el reuso y los cataloga adecuadamente. Funciona correctamente los componentes desarrollados.
Cree la Base de Datos a partir de la documentación de diseño.	Crea en forma correcta y completa la base de datos.
Desarrolle/ adapte funciones para satisfacer los nuevos requerimientos para el mantenimiento correctivo y	Desarrolla y adapta en forma completa funciones que requieren mantenimiento.

Resultados de Aprendizaje	Criterios de evaluación
perfectivo.	Funciona correctamente las funciones mantenidas..
Pruebe las funciones desarrolladas.	Realiza pruebas exhaustivas y detalladas de todas las funciones desarrolladas, identificando y corrigiendo errores. Utiliza casos de prueba variados y representativos para asegurar la robustez y la funcionalidad correcta de las funciones requeridas. Correcta depuración de errores. El software funciona correctamente luego del proceso de depuración.
Utilice adecuadamente las herramientas y técnicas para el desarrollo y mantenimiento del software. Instale y utilice adecuadamente las herramientas para las pruebas del software.	Instala las herramientas necesarias para el desarrollo, mantenimiento, documentación y prueba del software resolviendo las situaciones excepcionales que se le pudieran presentar durante la instalación. Demuestra dominio de las herramientas.
Defina las normas sobre las responsabilidades éticas para el desarrollo del software y determina su abordaje.	Define correctamente normas éticas para el desarrollo del software.
Aplique normas éticas en el desarrollo del software.	Aplica correctamente las normas en el desarrollo del software
Determine una evaluación estratégica del impacto del software desarrollado en el ámbito de la organización.	Identifica con precisión los indicadores que permiten determinar el impacto social del software desarrollado y elabora un adecuado instrumento para la recolección de datos.
Realice las mediciones y proponga medidas para actuar en relación al impacto obtenido.	Realiza correctamente las mediciones de las experiencias de usuario, emite conclusiones y medidas para actuar en relación al impacto obtenido.
Defina el Plan de Desarrollo de Software, el Plan de Mantenimiento y el Plan de Garantía de la Calidad.	Definición correcta y completa de los planes identificando actividades y recursos de desarrollo, definición de la calendarización y especificación de roles de cada integrante.
Gestione el Plan de Desarrollo de Software, de Mantenimiento y de Garantía de Calidad.	Revisa regularmente el avance y realiza ajustes según sea necesario.
Ajuste los planes realizados de acuerdo a los desvíos	Identifica y corrige desviaciones respecto al plan original.
Organice el grupo de trabajo para realizar las tareas de creación del grupo. Distribuya responsabilidades y actividades a cada integrante del grupo de trabajo. Evalué el desempeño de los integrantes y del trabajo en equipo.	Adecuada organización del grupo de acuerdo a las fortalezas de cada integrante. Distribución equitativa de tareas. Adecuado nivel de involucramiento de cada miembro en las tareas asignadas. Adecuada comunicación entre los miembros del equipo. Realización de una autoevaluación del desempeño de cada miembro del equipo.
Elabore informes técnicos del desarrollo, mantenimiento y 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.	Organización coherente de la información y pertinencia de los temas incluidos. Correcta redacción, utilización de lenguaje técnico y uso adecuado del estilo de escritura que facilite la comprensión. Inclusión de fuentes de información o referencias acorde a las normas APA. Presentación coherente de los resultados y conclusiones.
Exponga los trabajos organizando la información para	Respuesta de forma clara y precisa a las preguntas o

Resultados de Aprendizaje	Criterios de evaluación
facilitar la comprensión a los destinatarios, utilizando recursos gráficos y con técnicas de comunicación oral.	comentarios. Utilización correcta de técnicas de comunicación oral y de vocabulario técnico. Correcto uso de recursos gráficos y de técnicas de comunicación oral y, una adecuada organización de la información.

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 se definen en la descripción de los talleres y en el punto 8.2: Evaluación formativa.

8.3.3- Escala de Valoración

La escala de valoración a emplear: para las evaluaciones de los talleres y los recuperatorios será **APROBADA** y **DESAPROBADA**.

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.

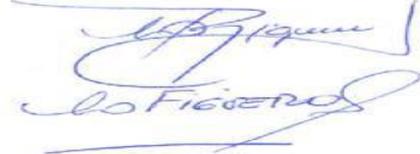
8.7- EXAMEN LIBRE

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