

Planificación de la Asignatura: Tópicos especiales en Tecnologías Aplicadas: Ingeniería del Software II

Fecha: 23/10/2024 13:02

Código: B0871-9

Carrera: Bioingeniería

Departamento Académico: Informática

Docente a cargo:

Correo del docente a cargo: victor.valotto@uner.edu.ar

Régimen de Dictado: Cuatrimestral 2º Cuatrimestre

Carga Horaria Semanal: 5 horas semanales

Carga Horaria Total: 70 horas

Contenidos Mínimos:

Nuevas tecnologías aplicables al área de la Bioingeniería.

Competencias Genéricas:

CT 1. Identificación, formulación y resolución de problemas de Bioingeniería. Nivel de Dominio 3

CT 2. Concepción, diseño y desarrollo de proyectos de Bioingeniería. Nivel de Dominio 3

CT 3. Gestión, planificación, ejecución y control de proyectos de Bioingeniería. Nivel de Dominio 3

CT 4. Utilización de técnicas y herramientas de aplicación en Bioingeniería. Nivel de Dominio 2

CT 5. Contribución a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas. Nivel de Dominio 2

CS 1. Fundamentos para el desempeño en equipos de trabajo. Nivel de Dominio 3

CS 2. Fundamentos para una comunicación efectiva. Nivel de Dominio 3

CS 3. Fundamentos para una actuación profesional ética y responsable. Nivel de Dominio 3

CS 4. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local. Nivel de Dominio 2

CS 5. Fundamentos para el aprendizaje continuo y autónomo. Nivel de Dominio 2

CS 6. Fundamentos para el desarrollo de una actitud profesional emprendedora, Nivel de Dominio 2

Competencias Específicas:

CE 1.1. Diseñar, calcular y proyectar instalaciones, equipamientos e instrumental de tecnología biomédica, procesamiento de señales biomédicas y sistemas derivados de biomateriales utilizados en el área de la salud. Nivel de Dominio 3

CE 1.2. Procesar señales e imágenes biológicas. Nivel de Dominio 3

CE 2.1. Proyectar, dirigir y controlar la construcción, operación y mantenimiento de lo anteriormente mencionado. Nivel de Dominio 3

CE 3.2. Asesorar en todos los procesos de elaboración de programas de compra, redactar normas y pliegos de adquisición, verificar los bienes y/o insumos adquiridos de equipos, sistemas y partes de sistemas de tecnología biomédica, sus complementos y accesorios, instalaciones y dispositivos afines necesarios a sus propósitos. Nivel de Dominio 2

Argumentación de aportes marcados en la matriz de competencias:

CT 1 - Identificación, formulación y resolución de problemas de Bioingeniería: La Ingeniería de Software enseña a descomponer problemas complejos en partes manejables y solucionarlos sistemáticamente. En la Bioingeniería, la identificación de problemas a menudo requiere una comprensión interdisciplinaria que se beneficia enormemente de un enfoque sistemático de la Ingeniería de Software para definir requisitos, formular soluciones y optimizar los resultados a través de iteraciones.

CT 2 - Concepción, diseño y desarrollo de proyectos de Bioingeniería: El proceso de diseño de software se

alineada estrechamente con los principios de ingeniería utilizados en el diseño y desarrollo de proyectos de Bioingeniería. La enseñanza de prácticas de diseño robustas y arquitecturas de software, además de modelado y análisis, proporciona una base sólida para la conceptualización y creación de sistemas complejos y sus integraciones de software.

CT 3 - Gestión, planificación, ejecución y control de proyectos de Bioingeniería: La gestión de proyectos de software aporta metodologías y herramientas que son fundamentales para la gestión de cualquier tipo de proyecto de ingeniería. Estas habilidades son transferibles y esenciales para la gestión eficaz de proyectos de Bioingeniería, donde la coordinación de tareas, la gestión de tiempos y la adaptación a cambios son críticos.

CT 4 - Utilización de técnicas y herramientas de aplicación en Bioingeniería: Las técnicas y herramientas aprendidas en Ingeniería de Software son a menudo las mismas utilizadas en Bioingeniería, como la programación en diferentes lenguajes, el uso de software de simulación y el análisis estadístico de datos, lo que ayuda a la capacidad para manejar estas herramientas.

CT 5 - Contribución a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas: La Ingeniería de Software es intrínsecamente una disciplina de innovación y desarrollo tecnológico. Las habilidades adquiridas en esta materia son esenciales para contribuir al avance tecnológico en Bioingeniería, desde la creación de nuevos algoritmos hasta el desarrollo de software en las distintos ámbitos de aplicación.

CS 1 - Fundamentos para el desempeño en equipos de trabajo: La ingeniería de software, especialmente en un entorno de aprendizaje, promueve el trabajo en equipo a través de proyectos grupales. Estos proyectos requieren colaboración, compartición de conocimientos y la división de tareas, reflejando la naturaleza colaborativa de los proyectos de bioingeniería y preparando a los estudiantes para entornos de trabajo multidisciplinarios.

CS 2 - Fundamentos para una comunicación efectiva: Comunicar complejidades técnicas de manera efectiva es esencial en la ingeniería de software. Los estudiantes aprenderán a documentar código, crear informes de diseño de software y presentar sus soluciones a audiencias no técnicas, lo cual es vital para la interacción exitosa con otros profesionales y principales interesados en Bioingeniería

CS 3 - Fundamentos para una actuación profesional ética y responsable: La ética en la ingeniería de software se centra en la privacidad de los datos, la seguridad y la inclusión.

CS 4 - Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local: Al analizar situaciones y al desarrollar software con impacto social, como aplicaciones de salud o herramientas para la accesibilidad, los estudiantes podrán considerar las implicaciones sociales de su trabajo. Esto refuerza la conciencia de su impacto como bioingenieros en la sociedad y la importancia de considerar las repercusiones éticas y sociales de su labor.

CS 5 - Fundamentos para el aprendizaje continuo y autónomo: Dada la rápida evolución del campo del

software, se fomenta el aprendizaje autónomo al mantenerse actualizados con las últimas tecnologías y prácticas de programación.

CS 6 - Fundamentos para el desarrollo de una actitud profesional emprendedora: Los proyectos de ingeniería de software a menudo están vinculados a escenarios de startups y desarrollo de productos, fomentando una mentalidad emprendedora.

CE 1.1 - Diseño, cálculo y proyecto de tecnología biomédica: : En Ingeniería de Software, los estudiantes aprenden metodologías de diseño que pueden aplicarse al desarrollo de software para la tecnología biomédica. Esto incluye comprender cómo los requisitos del usuario y del sistema determinan las especificaciones de diseño de equipamientos e instrumentales, y cómo las técnicas de procesamiento de señales biomédicas son esenciales para el diseño de interfaces y software de diagnóstico.

CE 1.2 - Procesamiento de señales e imágenes biológicas: Los cursos de Ingeniería de Software pueden incluir módulos sobre el procesamiento de datos y algoritmos específicos para señales e imágenes biológicas. Los estudiantes aprenderán a desarrollar y aplicar software que pueda adquirir, analizar y procesar datos biomédicos, una habilidad esencial en el campo de la bioingeniería.

CE 2.1 - Proyección y control de la tecnología biomédica: La gestión de proyectos y el control de calidad son aspectos fundamentales en Ingeniería de Software. Los conocimientos adquiridos aquí son aplicables al manejo de proyectos que involucran tecnología biomédica, incluyendo su construcción, operación y mantenimiento.

CE 3.2 - Asesoramiento en adquisiciones tecnológicas biomédicas: El conocimiento de las normas y estándares de software, así como la capacidad para evaluar y verificar soluciones de software, es crucial al asesorar en la adquisición de equipos tecnológicos biomédicos y en la redacción de pliegos técnicos.

Correlativas Regulares para cursar:

No posee

Correlativas Aprobadas para cursar:

Ciclo Básico aprobado + Específica indicada en Planificación

Correlativas Aprobadas para promocionar o rendir el examen final:

Ciclo Básico aprobado

Insercion de la Asignatura en el plan de Estudios:

La presente materia constituye una segunda parte y complementa los contenidos relacionados con el Área de Conocimiento de la Ingeniería de Software, iniciada ya con la materia Ingeniería de Software I. Por lo tanto los conceptos fundamentales involucrados en la planificación de Ingeniería de Software I son aplicables a Ingeniería de Software II.

Objetivo General:

Transmitir al alumno la importancia de la especificación en el proceso de desarrollo de software. Estudio de la problemática que plantea el desarrollo de grandes sistemas de software, profundizar en los aspectos de especificación. Introducir al alumno en los formalismos y herramientas de especificación y especialmente en los de especificación del paradigma orientación a objetos.

- Dotar al alumno de conocimientos para que sea capaz de realizar especificaciones en modelos.
- Dotar al alumno de conocimientos sobre procesos de ingeniería del software haciendo hincapié en los métodos iterativos e incrementales. Mostrar el papel de las especificaciones en dichos procesos.
- Que el alumno logre comprender la importancia de las buenas prácticas y que toda construcción y mantenimiento de un producto de software con calidad debe ser soportado por procesos de ingeniería.

Objetivos Particulares:

- Describir las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software.
- Describir los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software.
- Introducir a la visión de arquitectura de software como base del proceso de desarrollo de los grandes sistemas.
- Comprender los fundamentos del diseño de sistemas software.
- Aplicar de forma práctica los conceptos teóricos sobre el desarrollo estructurado y orientado a objetos.
- Comprender las técnicas y métodos de planificación de proyectos de software.
- Comprender la importancia de las técnicas de estimación de software.
- Comprender el alcance de las actividades de aseguramiento de la calidad y administración de la configuración.
- Realizar un proyecto en grupo, aplicando los principios introducidos en la parte teórica de la asignatura.

Programa Analítico:

- **Módulo 1 - Diseño Arquitectural:** ¿Qué es la arquitectura de software? Decisiones de diseño. Estilos y modelos arquitectónicos. Modelo de Referencia y patrones de arquitectura. Sistemas distribuidos desde el punto de vista de arquitectura. Arquitectura de capas. Concepto de Servicios. Atributos de calidad. Especificación de los escenarios de calidad. Vista de arquitectura.
- **Módulo 2 - Diseño Detallado:** ¿Qué es el modelado? Fundamentos de UML. Elementos y diagramas. Paquetes, diagramas de actividad, diagramas de interacción (secuencia y colaboración), diagramas de clases, diagramas de transición de estados, diagramas de componentes y diagrama de despliegue.
- **Módulo 3 - Mantenimiento del Software:** Concepto de Cambio, impacto del cambio en la organización. Características y problemas. Políticas y administración del cambio. Proceso de Administración de cambios, roles y responsabilidades. Análisis de Impacto. Trazabilidad. Administración de requerimientos, seguimiento y control de cambios.
- **Módulo 4 - Testing de Software:** Conceptos. ¿Qué es testear software? Axioma del Testing. Taxonomías de Testing (Tipos de Testing). El proceso de testing. Planificación de Testing. Diseño de Caso de Testing. Testing de caja blanca y testing de caja negra. Estrategias de Testing: Particiones de Equivalencias y Valores Límites. Diseño de casos de pruebas funcionales. Axiomas del proceso de Testing.
- **Módulo 5 - Gestión de Proyectos:** Definiciones de Proyecto. Procesos de Administración de Proyectos. Organización de proyectos. Roles y Responsabilidades. Alcance. Ciclo de Vida. WBS. Programación de actividades. Diagrama de Gantt: tareas, asignaciones, hitos, holguras, camino crítico. Administración de Riesgos: Identificación del riesgo, análisis del riesgo, exposición del riesgo, mitigación y contingencia. Planificación de la comunicación. Plan de comunicación. Tipos y herramientas de comunicación. Control y seguimiento: Analizar datos del proyecto. Analizar variaciones. Cierre. Indicadores de gestión. Acciones adaptativas. Cierre del proyecto. Lecciones aprendidas.
- **Módulo 6 - Estimación de Proyectos:** ¿Porque estimar?. Métricas de Tamaño: Líneas de Código, Costeo del Proyecto, Puntos de Función, Puntos de Caso de Uso, StoryPoints. Métricas de esfuerzo y costo: COCOMO.
- **Módulo 7 - Gestión de la calidad :** Concepto de Calidad. Los hacedores de la Calidad: Juran, Deming y

Crosby. Total Quality Management. Problemas asociados a la calidad. Planificación de la Calidad, Aseguramiento de la Calidad, Control de la Calidad. Atributos de calidad. Cuantificando los atributos de Calidad. Costos de la Calidad. Roles y Responsabilidades del área de QA. Revisiones. Tipos de Revisiones. Revisiones Formales. Proceso. Administración de la Configuración. Problemas: en el desarrollo de software. Identificación de los elementos de configuración. Conceptos. Elementos de Configuración, Línea Base, Hitos y Entregables. Almacenado de los elementos de configuración. Gestión de Versiones, conceptos. Desarrollo concurrente. Recomendaciones. Gestión de entregas. Procedimientos. Tipos y clasificaciones de las entregas. Gestión de Entornos. Tipos de entornos.

Metodología Didáctica:

Las instancias para el desarrollo de la asignatura serán clases teóricas, prácticas. Para el desarrollo de cada una de las partes en que se divide la asignatura, se partirá de elementos simples sobre los que se estructurará el conocimiento.

Se continuará con el enfoque ya desarrollado en Ingeniería de Software I, donde se propone que los estudiantes trabajen en un esquema de autoestudio durante las horas asignadas a la cátedra.

Posteriormente, habrá una instancia final en cada módulo, correspondiente a una actividad de coloquio dirigida por el docente a cargo con el objetivo de mejorar conceptos, estimular el trabajo grupal, enfatizar el análisis crítico y participar en actividades basadas en juegos para la autoevaluación. Esta visión tiene una perspectiva de aula invertida.

Se intentará apuntar a un método de enseñanza que es tanto interactivo como reflexivo, adaptándose a la diversidad de estilos de aprendizaje de nuestros estudiantes. Cada semana, los estudiantes incursionarán en nuevos aspectos de la Ingeniería de Software a través de videos educativos breves y concisos, diseñados para introducir conceptos clave de manera efectiva. Estos videos serán el punto de partida para una exploración más profunda, complementados con una variedad de materiales adicionales como artículos relevantes, estudios de caso y recursos en línea.

Para garantizar una comprensión sólida y activa de estos conceptos, cada video estará acompañado de un cuestionario corto de múltiples opciones. Estos cuestionarios, diseñados para ser tanto desafiantes como informativos, servirán para reforzar el aprendizaje y proporcionarán una retroalimentación inmediata sobre la comprensión de los temas. Esta estrategia formativa permitirá a los estudiantes autoevaluar su progreso y a los docentes identificar y abordar áreas que puedan necesitar atención adicional.

Respecto a las actividades prácticas, éstas se compondrán de situaciones y/o escenarios que se pueden presentar en proyectos reales, donde en general lo que dispara son elementos de análisis y discusiones para las posibles resoluciones a los problemas.

Formación Práctica:

La realización de las actividades prácticas se centrarán en los módulos relacionados con el Producto de Software principalmente.

Con excepción del primer ejercicio, el resto está basado en un Caso de Estudio propuesto por la cátedra.

Los ejercicios apuntan a ir evolucionando en sus especificaciones de manera que los alumnos vayan proveyendo soluciones parciales en cada ejercicio. Estas soluciones son el producto de la realización de los ejercicios, los cuales están basados en la elaboración de modelos para cada una de las fases propuestas en los ejercicios, y serán presentadas por cada grupo o alumno.

El desarrollo de estas presentaciones serán realizadas con la asistencia de herramientas de software recomendadas por la cátedra, con recomendaciones del uso de herramientas gratuitas o de uso académico.

Estos trabajos tienen por objetivo generar discusiones y debates sobre el criterio de uso de las diferentes técnicas, métodos y herramientas ante diferentes situaciones.

Estos aspectos se desarrollarán durante los horarios de clases, para revisar los resultados y analizar los mismos, durante horas anexas que los alumnos utilizarán según la necesidad individual.

La principal fuente para la generación de los prácticos corresponden a los dos libros bases de la bibliografía.

Listado de Actividades de Formación Práctica:

- Ejercicio 1: Definición de Arquitectura:
- Ejercicio 2: Diseño detallado – Colaboración de Objetos.
- Ejercicio 3: Diseño detallado – Clases y Componentes del Sistema.
- Ejercicio 4: Diseño de Casos de Prueba.
- Ejercicio 5: Dimensionamiento del Sistema.

Intensidad de la formación práctica

Detalle de la carga horaria total prevista para cada una de las siguientes actividades:

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 1: horas

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 2: 30 horas

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 3: horas

Horas totales de actividades de formación práctica: 30 horas

Metodología de Evaluación Durante el cursado:

Se hará un seguimiento de los estudiantes a lo largo del dictado de la materia, a través los trabajos prácticos.

Para la evaluación de los conocimientos adquiridos los estudiantes deberán realizar una exposición sobre un contenido definido por la cátedra, seguido de preguntas relacionadas con dichos contenidos.

Metodología de Evaluación en Exámenes Finales:

Tanto alumnos libres como regulares serán evaluados de acuerdo a un examen escrito.

Condiciones de Regularidad :

Para la evaluación de los conocimientos adquiridos los estudiantes deberán realizar una exposición sobre un contenido definido por la cátedra, seguido de preguntas relacionadas con dichos contenidos.

Habrà una instancia de recuperación en caso que la evaluación de esta presentación sea inferior a 6(seis) aprobado. Si la evaluación es de 8(ocho) o superior la condición del estudiante es promocionado. En cualquier otro caso, es decir, entre 6 inclusive o menor a 8 la condición de regular.

Cronograma de parciales durante el primer Cuatrimestre:

Cronograma de parciales durante el segundo Cuatrimestre:

Primer Examen Parcial: 29 de Octubre de 2024

Recuperatorio 01: 05 de Noviembre de 2024

Bibliografía Principal:

Ingeniería del Software, un enfoque práctico. Roger Pressman. MCGRAW-HILL, 2005, Edición Número 6.

Ingeniería del Software, Ian Sommerville. PEARSON ADDISON WESLEY, 2005, Séptima edición.

Bibliografía Complementaria:

Equipo de Cátedra:

Profesor Adjunto Dedicación Simple

Jefe de Trabajos Prácticos Dedicación Simple

Primer Cuatrimestre: Dictado de la materia Ingeniería de Software I

Asignación:

Profesor Adjunto: 10 hrs por semana

Jefe de Trabajos Prácticos: 6 hrs por semana

Segundo Cuatrimestre: Dictado de la materia Ingeniería de Software II

Asignación:

Profesor Adjunto: 10 hrs por semana

Jefe de Trabajos Prácticos: 6 hrs por semana

Actividades de Investigación Gestión y Extensión:

Sin actividades

Requisitos de admisión para alumnos oyentes:

Sin requisitos

Infraestructura, equipamiento y recursos necesarios:

Laboratorio de Computación

Otros:

Sin información