

Planificación de la Asignatura: Tópicos especiales en Tecnologías Básicas: Ingeniería de Software I

Fecha: 23/10/2024 13:02

Código: B0870-2

Carrera: Bioingeniería

Departamento Académico: Informática

Docente a cargo:

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

Régimen de Dictado: Cuatrimestral 1º Cuatrimestre

Carga Horaria Semanal: 5 horas semanales

Carga Horaria Total: 70 horas

Contenidos Mínimos:

Profundización de conocimientos avanzados de interés para la bioingeniería en el área de las Tecnologías Básicas.

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 alinea 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:

3° Año

Correlativas Aprobadas para cursar:

2° Año

Correlativas Aprobadas para promocionar o rendir el examen final:

2° Año

Insercion de la Asignatura en el plan de Estudios:

Partiendo desde una posición muy general, son muy pocas las personas que dudan de la necesidad del software en la vida diaria de todos nosotros. Lo podemos ver cotidianamente. Más aún, los negocios sin la ayuda del software, entre otras cosas, probablemente no puedan alcanzar los objetivos. A pesar de esta expansión e inserción en los aspectos más simples de nuestra vida, como por ejemplo buscar donde comprar un libro o juguete, hasta mucho más complejos como el control de los equipos de medicina, sigue habiendo una idea o mejor dicho una “manera de ver” al software como un producto desarrollado por el hombre con mucho de arte y de inspiración.

Por otro lado, hay muchas evidencias por los estudios realizados y por los innumerables fracasos, que el software es una actividad cara y que en general siempre tiene los problemas de tener muchos defectos, no tener las funciones que se pensaban y que tener un software terminado a tiempo es una esperanza que alguna vez podrá ocurrir. Construir un software, no es hacer un programa. Son cuestiones bastante diferentes. Un software es un elemento económicamente productivo, tiene asociados conceptos de productividad y uso social, y debe ser visto como cualquier otro bien que es producido por el hombre, a pesar de ser (o parecer) intangible.

Ahora bien, ¿Qué producto creado por el hombre puede ser considerado exitoso y útil para que el resto de la sociedad se beneficie de él, si no puede ser producido de manera extendida y mantener las mismas funcionalidades y funcionamiento originales y/o mejoradas? Esto se logra con cierto grado de invariabilidad en la manera de hacerlo, con aceptación de los compradores y de los usuarios y con el entendimiento de que debe existir una determinada secuenciación en la “fabricación de un producto de software”.

Por lo tanto con esta visión, estamos en presencia de un producto de ingeniería, donde existen las etapas más básicas de un proceso de ingeniería: concepción y entendimiento, diseño, construcción, pruebas del producto y puesta en funcionamiento. En definitiva todas aquellas profesiones, y principalmente aquellas que se valen del software como ayuda para la elaboración y uso de sus productos, que poseen algún grado de interés o involucramiento deben comprender los alcances conceptuales de esta disciplina para potenciar la mejora en su concreción y así proveer mejores soluciones a la sociedad. Precisamente a la Bionformática podemos conceptualizarla como la construcción de herramientas adaptables para procesar, administrar, analizar y visualizar la información biológica, la que no solo requieren de un conocimiento establecido de la ciencia subyacente, sino también la capacidad de escribir programas eficientes, o mejor dicho Productos de Software.

Esta materia en conjunto con “Ingeniería de Software II” deben apuntar desde el inicio en el desarrollo de capacidades sobre los alumnos para que los futuros profesionales puedan enfrentar las diferentes situaciones laborales, ya sea de índole científica o comercial, con un enfoque en la generación de productos

de calidad basado en un proceso de desarrollo con calidad. Es por ello que el contenido y los objetivos están basados en conceptos y buenas prácticas adoptadas por la industria del software, en el que se incluyen principios y técnicas que ayudarán a los alumnos a adquirir un conjunto de herramientas del conocimiento que puedan proveer los resultados que se esperan.

El software, por todo lo enunciado en los primeros párrafos, tiene una implicancia económica suprema en nuestra sociedad, y más aún dentro de una profesión donde constituye junto con las ciencias biológicas los basamentos de su concepción. Por ello esta materia tiene dentro de cada uno de los contenidos, la esencia para que la construcción de un producto de software tenga un abordaje donde se puede ahorrar tiempo y esfuerzo haciendo lo correcto, en el momento adecuado y de la manera correcta.

Ambas materias “Ingeniería de Software I” e “Ingeniería de Software II” van más allá de la codificación para examinar el ciclo de vida del proyecto. Es una guía través del proceso de desarrollo de aplicaciones de principio a fin, sin ocuparse de una metodología en particular, paradigma de programación o plataforma tecnológica.

Ambos cursos se centrará en introducir las diferentes áreas de conocimiento que comprende la ingeniería de software, basado en el cuerpo de conocimiento de la ingeniería de software (SWEBOK) propuesto por la IEEE.

En particular el alcance de los contenidos de “Ingeniería de Software I” abarcará principalmente la introducción el procesos de desarrollo de un producto de software, y a la modelización de la necesidades de quien finalmente usará el producto de software como herramienta.

Objetivo General:

Presentar los conceptos fundamentales, junto con un enfoque de ingeniería a la hora de enfrentarse a una solución basada en software como herramienta para las actividades relacionadas con la bioinformática. Esta asignatura está centrada la propuesta de ofrecer al futuro profesional la visión de que el software es un producto que tiene un proceso de construcción que determina la calidad y el éxito de su utilidad posterior.

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.
- Presentar situaciones reales y donde la solución a problemas no son solo técnicas sino son principalmente de uso social.

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.
- Presentar la importancia de los requisitos en el ciclo de vida del software.
- Introducir a las técnicas básicas de elicitación, documentación, especificación y prototipado de los requisitos de un sistema software.
- Demostrar la ventajas del uso de estándares en el modelado, y la necesidad de presentar diferentes visiones del mismo productos según las diferentes necesidades.
- Realizar un proyecto en grupo, aplicando los principios introducidos en la parte teórica de la asignatura.

Programa Analítico:

Programa:

- Módulo 1. El producto de Software: ¿Qué es el software? Características. ¿Qué es Ingeniería de Software? Diferencias con otras disciplinas. Proceso de Software. Costos y Desafíos. Código de Ética. Problemática actual. Evidencia de la problemática.
- Módulo 2. La Ingeniería del Software: Conceptos: Programa, Producto, Sistema. Tipos de Sistemas. ¿Qué es ingeniería? Ingeniería de Sistemas y de Software. Proceso de Ingeniería. Áreas de Aplicación Cuerpo de Conocimiento de la Ingeniería de Software.
- Módulo 3. El Proceso del Software: Modelos de procesos: Cascada, Code-and-Fix. Evolutivo, Desarrollo en Espiral, Desarrollo Incremental, Basado en componentes. Áreas de proceso del Desarrollo de Software. RUP. Desarrollo Ágil: motivación y características. Selección del ciclo de vida.
- Módulo 4 – Requerimientos Funcionales: Proceso y producto. Comprender los fracasos del software. El problema de los malos requerimientos. Concepto de requerimientos. Definiciones, niveles y tipos de requerimientos. La ingeniería de requerimientos. El ciclo de requerimientos. El analista de requerimientos. Proceso de elicitación de requerimientos. Los stakeholders, los usuarios. Actividades de elicitación. Técnicas para entender el problema. Análisis de requerimientos. Técnicas de modelado del problema. Clasificación y priorización. Modelado con Casos de Uso. Actores, Casos de Uso, escenarios. Historias de Usuario. Especificación de requerimientos.
- Módulo 5 – Requerimientos No Funcionales. Atributos de Calidad. Restricciones. Los atributos de calidad y la arquitectura del software, aspectos transversales. Escenarios de Atributos de Calidad. Clasificación de los Atributos de Calidad. Priorización y conflictos. Sistemas en Tiempo Real y sistemas Críticos.
- Módulo 6 – Diseño Centrado en el Usuario. Interacción del Usuario y presentación de la Información. Proceso de Diseño de la Interfaz de Usuario. Técnicas de Análisis de usuario. Prototipado de la Interfaz. Aspectos transversales en el diseño de interfaz de usuario. Usabilidad. Estándares y Guías. Diseño de Interfaz de Usuario Web.

Metodología Didáctica:

Las instancias para el desarrollo de la asignatura se dividen en dos tipos de enfoques de trabajo con los estudiantes: un bloque de actividades con una visión teórica y otro bloque dirigido a actividades prácticas. Cada módulo de la asignatura comenzará con elementos simples sobre los cuales se estructurará el conocimiento.

En cuanto al enfoque teórico de los módulos I y II, se utilizará una orientación de clase expositiva. A partir del módulo III, 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 y aprovechando las ventajas de la modalidad híbrida. 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, estas se compondrán de situaciones y escenarios que se presentan en proyectos reales. Estas actividades estimularán el análisis y la discusión para encontrar posibles soluciones 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:

Serán realizadas una cantidad de ejercicios por cada módulo, con una carga horaria para los alumnos estimada que comprende las horas de cátedra y horas individuales extra cátedra.

- Trabajo Practico 1: Procesos y Ciclos de Vida: 4 hrs.
- Trabajo Practico 2: Clasificación de Requerimientos: 2 hrs.
- Trabajo Práctico 3: Modelado con Casos de Uso: 6 hrs.
- Trabajo Práctico 4: Clasificación de los Atributos de Calidad: 2 hrs.
- Trabajo Práctico 5: Especificación de Requerimientos No Funcionales: 6 hrs.
- Trabajo Practico 6: Diseño de Interfaz de Usuario: 4 hrs.

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 y coloquios.

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:

Primer Examen Parcial: 04 de Junio de 2024

Recuperatorio 01: 11 de Junio de 2024

Cronograma de parciales durante el segundo Cuatrimestre:

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:

Segundo Cuatrimestre: Dictado de la materia Ingeniería de Software I .

Profesor Adjunto Dedicación Simple

Jefe de Trabajos Prácticos Dedicación Simple

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