



Guía docente

340376 - INEP-I3O23 - Introducción a la Ingeniería del Programario

Última modificación: 02/07/2023

Unidad responsable: Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú
Unidad que imparte: 747 - ESSI - Departamento de Ingeniería de Servicios y Sistemas de Información.

Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2018). (Asignatura obligatoria).

Curso: 2023 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: Franch Gutiérrez, Xavier

Otros: López Cuesta, Lidia

CAPACIDADES PREVIAS

Conocimientos básicos de programación

REQUISITOS

Metodologías de la programación

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

3. CEFC1. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
6. CEFC2. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.
1. CEFC16. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
2. CEFC6. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
8. CEFC8. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
10. CETI5. Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados.
11. CEFC3. Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.

Transversales:

9. APRENDIZAJE AUTÓNOMO - Nivel 2: Llevar a cabo las tareas encomendadas a partir de las orientaciones básicas dadas por el profesorado, decidiendo el tiempo que se necesita emplear para cada tarea, incluyendo aportaciones personales y ampliando las fuentes de información indicadas.

12. APRENDIZAJE AUTÓNOMO: Detectar deficiencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento.

16. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicarse de forma oral y escrita con otras personas sobre los resultados del aprendizaje, de la elaboración del pensamiento y de la toma de decisiones; participar en debates sobre temas de la propia especialidad.

18. TRABAJO EN EQUIPO - Nivel 2: Contribuir a consolidar el equipo planificando objetivos, trabajando con eficacia y favoreciendo la comunicación, la distribución de tareas y la cohesión.

20. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo interdisciplinar ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

METODOLOGÍAS DOCENTES

Un caso de estudio que vehicula todo el curso.

Material en forma de transparencias, para el estudio y repaso de los contenidos.

Ejercicios y problemas que los estudiantes deben resolver por su cuenta como mecanismo de comprensión de los contenidos de la asignatura.

Una práctica en grupos de tres personas como mecanismo fundamental de aprendizaje y evaluación.

Diversas actividades a lo largo del curso, diseñadas como mecanismo de aprendizaje y evaluación.

Sesiones de laboratorio de tutoría del trabajo hecho.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Entender el concepto y la importancia de la ingeniería del software.

Conocer las distintas etapas del desarrollo de un sistema de software, y los diferentes objetivos de cada etapa.

Ser capaces de entender un problema complejo y de modelizarlo.

Saber documentar una especificación.

Ser capaces de diseñar la arquitectura de un sistema que responda a las necesidades del problema especificado.

Ser capaces de diseñar una base de datos relacional, crearla y realizar consultas básicas con el lenguaje SQL

Saber trabajar en equipo.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

Introducción a la ingeniería del software (IS)

Descripción:

Presenta los conceptos básicos de la IS que se desarrollarán a lo largo del curso

Objetivos específicos:

Conocer la necesidad y entender el concepto de IS.

Conocer el concepto de ciclo de vida del software como una combinación de distintas etapas.

Conocer las etapas del ciclo de vida.

Actividades vinculadas:

C1

Dedicación: 2h

Grupo grande/Teoría: 2h

Especificación de sistemas de software

Descripción:

Presenta la etapa de ingeniería de requisitos y los diversos modelos que permiten describir los requisitos de forma rigurosa

Objetivos específicos:

Conocer las actividades propias de la etapa de ingeniería de requisitos.

Conocer y aplicar los modelos proporcionados por el lenguaje UML para la especificación de requisitos funcionales.

Conocer y aplicar aproximaciones sistemáticas en la descripción de requisitos no funcionales.

Actividades vinculadas:

C1, Pr, T1

Dedicación: 34h

Grupo grande/Teoría: 12h

Grupo pequeño/Laboratorio: 10h

Aprendizaje autónomo: 12h

Diseño arquitectónico de sistemas de software

Descripción:

Presenta la etapa de diseño del software, basada en la aplicación de patrones

Objetivos específicos:

Conocer las actividades propias de la etapa de diseño del software.

Conocer el concepto de patrón y la distinción entre patrón arquitectónico y patrón de diseño.

Conocer algunos patrones básicos tanto de arquitectura como de diseño y entender sus efectos sobre los requisitos.

Aprender a transformar una especificación del problema en un diseño del sistema mediante la aplicación sistemática de patrones.

Actividades vinculadas:

C2, Pr, T2

Dedicación: 22h

Grupo grande/Teoría: 8h

Grupo pequeño/Laboratorio: 6h

Aprendizaje autónomo: 8h



Introducción a la persistencia y las bases de datos

Descripción:

Introduce los aspectos esenciales de la persistencia de los datos basada en la tecnología relacional

Objetivos específicos:

Comprender el concepto de persistencia de los datos.

Comprender la tecnología relacional de representación de los datos.

Aprender cómo transformar una especificación de los datos expresada en un modelo conceptual UML, en un modelo relacional.

Aprender cómo realizar consultas simples sobre modelos relacionales usando el lenguaje SQL.

Ser capaces de transformar un modelo estatico de una solución en un modelo de persistencia basado en bases de datos relacional.

Ser capaces de hacer consultas SQL simples en una base de datos

Actividades vinculadas:

C2, Pr, T3

Dedicación: 18h

Grupo grande/Teoría: 6h

Grupo pequeño/Laboratorio: 6h

Aprendizaje autónomo: 6h

Conclusión

Descripción:

Realiza un compendio de los conocimientos y prácticas impartidos en la asignatura y avanza los desafíos que se estudiarán en la asignatura AMEP

Objetivos específicos:

Resumir y relacionar los conocimientos y prácticas impartidas en la asignatura.

Enumerar las limitaciones de las técnicas vistas.

Avanzar algunas soluciones a estas limitaciones, y posibles técnicas alternativas

Actividades vinculadas:

T4

Dedicación: 2h

Grupo grande/Teoría: 2h

Desarrollo de una práctica

Descripción:

Aplica los conocimientos introducidos durante el curso en una práctica de laboratorio desarrollada en equipo

Objetivos específicos:

Aprender a utilizar eficazmente un entorno de desarrollo de software adecuado para los objetivos del curso.

Aprender a analizar el enunciado de un problema no trivial por su implementación.

Aplicar las nociones teóricas de especificación, diseño y persistencia en caso de uso definido en el enunciado.

Afrontar los retos típicos de programación de un sistema aplicando los principios y métodos de la ingeniería del software.

Actividades vinculadas:

Pr, C2

Dedicación: 72h

Grupo pequeño/Laboratorio: 8h

Aprendizaje autónomo: 64h



SISTEMA DE CALIFICACIÓN

Ver versión en catalán

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Ver versión en catalán

BIBLIOGRAFÍA

Básica:

- Garcia-Molina, Hector; Ullman, Jeffrey D; Widom, Jennifer. Database systems : the complete book [en línea]. 2nd ed. Harlow, Essex: Pearson Education Limited, 2013 [Consulta: 14/02/2024]. Disponible a : <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-origsite=primo&docID=5174436>. ISBN 9781292037301.
- Pressman, Roger S. Ingeniería del software : un enfoque práctico [en línea]. 9a ed. México [etc.]: McGraw-Hill, 2021 [Consulta: 16/02/2024]. Disponible a : https://www-ingebook-com.recursos.biblioteca.upc.edu/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272. ISBN 9781456287726.
- Sommerville, Ian. Software engineering [en línea]. 10th ed. Harlow: Pearson Education Limited, 2016 [Consulta: 14/02/2024]. Disponible a : <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-origsite=primo&docID=5185655>. ISBN 9781292096148.
- Larman, Craig. Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development. 3th ed. Upper Saddle River, N.J.: Prentice Hall PTR, 2005. ISBN 0131489062.