FICHA IDENTIFICATIVA

Datos de la Asignatura

Código	44834
Nombre	Métodos de producción de software
Ciclo	Máster
Créditos ECTS	4.0
Curso académico	2024 - 2025

Titulación(es)

Titulación	Centro	Curso	Periodo
2234 - Máster Universitario en Tecnolog. Web, Computación Nube y Aplicac. Móviles	Escuela Técnica Superior de Ingeniería	1	Primer cuatrimestre

Materias

Titulación	Materia	Carácter
2234 - Máster Universitario en Tecnolog. Web, Computación Nube y Aplicac. Móviles	5 - Producción de software, seguridad y profesión	Obligatoria

Coordinación

Nombre	Departamento
PANACH NAVARRETE, JOSE IGNACIO	240 - Informática

RESUMEN

El desarrollo de software implica el uso de distintas metodologías a lo largo de todo el ciclo de desarrollo. En esta asignatura se abordarán los métodos más utilizados en cada una de las fases del proceso de desarrollo. En primer lugar se verán métodos de gestión de proyectos. La gestión de un proyecto está compuesta por diversos elementos: el cliente que desea un producto, los trabajadores que participan en el proyecto, los plazos de tiempo, los requisitos a satisfacer, los recursos, etc. Algunos de estos elementos implican limitaciones y restricciones al proyecto, por ejemplo, normativa legal, restricción de presupuesto, restricción de personal, etc. La asignatura está enfocada a que el alumno adquiera los conocimientos suficientes para poder combinar todos estos elementos de forma satisfactoria. Además, se le mostrará al alumno cómo adelantarse a posibles riesgos y evitarlos.

En segundo lugar se abordarán métodos de testeo del software para garantizar la calidad de los sistemas desarrollados a través de las pruebas. Se verán conceptos, principios y tareas básicas que componen la fase de pruebas de un ciclo de vida de un proyecto Web. El objetivo de las pruebas es el de definir un conjunto de casos con alta probabilidad de encontrar errores en el código. Se verán varias técnicas de pruebas, tanto las relacionadas a requisitos funcionales como a los no funcionales.

En tercer lugar se abordarán métodos para el análisis y diseño del software. La Ingeniería Web es la rama de la Ingeniería del Software que recoge y propone modelos de desarrollo específicos del área: implementación incremental, continuos y frecuentes cambios, plazos de desarrollo muy cortos. Se verán métodos ágiles como SCRUM y métodos dirigidos por pruebas como Test-Driven Development (TDD).

El trabajo a desarrollar el alumno pretende ser eminentemente práctico, cada uno de los conceptos teóricos explicados en clase se verán aplicados a un problema. Para la resolución de estos problemas se usarán herramientas utilizadas en empresas del mundo real.

CONOCIMIENTOS PREVIOS

Relación con otras asignaturas de la misma titulación

No se han especificado restricciones de matrícula con otras asignaturas del plan de estudios.

Otros tipos de requisitos

Conocimientos de las fases que componen el ciclo de desarrollo software
Conocimientos de técnicas de captura de requisitos
Conocimientos básicos de administración
Conocimientos básicos de elaboración de presupuestos
Conocimientos básicos de Java
Conocimientos de UML

COMPETENCIAS (RD 1393/2007) // RESULTADOS DEL APRENDIZAJE (RD 822/2021)

2234 - Máster Universitario en Tecnolog. Web, Computación Nube y Aplicac. Móviles

- Que los/las estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.
- Que los/las estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.
- Que los/las estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades.
- Que los/las estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo

- Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.
- Capacidad para la aplicación de los conocimientos adquiridos y de resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares, siendo capaces de integrar estos conocimientos.
- Capacidad para la elaboración, planificación, dirección, coordinación, gestión técnica y económica y la implantación de proyectos Web.
- Capacidad para comprender y aplicar la responsabilidad ética, la legislación y la deontología en el ejercicio profesional.
- Fomentar en contextos académicos y profesionales, el avance tecnológico, social o cultural dentro de una sociedad basada en el conocimiento y en el respeto a: a) los derechos fundamentales y de igualdad de oportunidades entre hombres y mujeres, b) los principios de igualdad de oportunidades y accesibilidad universal de las personas con discapacidad y c) los valores propios de una cultura de paz y de valores democráticos.
- Capacidad para aplicar metodologías de Ingeniería del Software en el desarrollo y gestión de un proyecto.

RESULTADOS DE APRENDIZAJE (RD 1393/2007) // SIN CONTENIDO (RD 822/2021)

- Especificar y completar tareas informáticas que son complejas, definidas de forma incompleta o poco familiares
- Describir y explicar técnicas y métodos aplicables a su particular área de estudio e identificar sus limitaciones
- Organizar su propio trabajo de forma independiente, demostrando iniciativa y ejerciendo responsabilidad personal
- Realizar búsquedas bibliográficas y revisiones usando bases de datos y otras fuentes de información
- Aprender y mejorar el rendimiento personal como la base para el aprendizaje a lo largo de la vida y el desarrollo profesional
- Comunicar de forma efectiva tanto verbalmente como a través de otros medios de comunicación a una variedad de audiencias y preferiblemente en un segundo lenguaje
- Apreciar las habilidades requeridas para trabajar con y liderar un equipo que puede estar compuesto por diferentes disciplinas y diferentes niveles de cualificación
- Dominar las técnicas y herramientas para definir, gestionar y estimar recursos y costes de proyectos.
- Planificar y realizar el seguimiento de las actividades implicadas en un proyecto.
- Conocer los retos de gestión de recursos humanos que supone un proyecto y conocer los enfoques aplicables.
- Aplicar metodologías para el desarrollo, implantación y mantenimiento de sistemas.
- Planificar y ejecutar correctamente procesos de desarrollo de software iterativos.

- Aplicar patrones de diseño software en cada situación en función de las necesidades del proyecto.
- Diseñar la estructura modular de un sistema utilizando y evaluando los patrones de diseño orientados a la solución del problema, evaluando sus ventajas, inconvenientes y alternativas.
- Definir pruebas de validación y verificación de requisitos y probar aplicaciones de manera sistemática definiendo casos de prueba exhaustivos.

DESCRIPCIÓN DE CONTENIDOS

1. Planificación de proyectos y gestión del riesgo

Alcance de un proyecto
Estructura y Diagrama de Tareas
Estimación de duración de las tareas
Establecer dependencias entre tareas
Asignación de recursos a tareas
ISO 9000
Identificación de riesgos
Evaluación de riesgos
Habilidades directivas
Tipos de contratos

2. Gestión del tiempo y tareas

Detección del camino crítico y cadena crítica
Técnica PERT
Optimización de tiempo/coste
Limitaciones del intercambio tiempo/coste

3. Estimación de proyectos

Puntos de Función
Cosmic
Juicio de expertos

4. Gestión de costes

Medición del avance y curva S del proyecto
Medidas de actividad del proyecto

5. Herramientas de test, colaborativas y de integración continua

Pruebas unitarias: JUnit
Pruebas unitarias en aislamiento:JMock
Pruebas de base de datos: DBUnit
Test Driven Development
Integración continua
GIT
Jenkins

6. SCRUM

- Planificación con SCRUM
- Reporte del proyecto
- El equipo de trabajo
- Reglas

7. Calidad

- Sonarqube
- Selenium

VOLUMEN DE TRABAJO

ACTIVIDAD	Horas	% Presencial
Clases teórico-prácticas	40,00	100
Elaboración de trabajos en grupo	6,00	0
Estudio y trabajo autónomo	35,00	0
Preparación de clases prácticas y de problemas	16,00	0
Resolución de cuestionarios on-line	3,00	0
TOTAL	**100,00**	

METODOLOGÍA DOCENTE

- Clase de teoría
- Resolución de problemas
- Aprendizaje orientado a proyectos

EVALUACIÓN

Los resultados del aprendizaje se evalúan a través de los siguientes criterios:

SE1: Evaluación continua del estudiante, basada en la participación y grado de implicación en el proceso de enseñanza-aprendizaje, teniendo en cuenta la asistencia regulada a las actividades presenciales previstas y la resolución de cuestiones y problemas propuestos. Esta nota sólo se puede obtener en primera convocatoria, manteniéndose constante para la segunda.

SE2: Evaluación de los problemas planteados en las clases presenciales. Estos problemas se pueden entregar en las 2 convocatorias. La entrega en la primera convocatoria se hará de forma continua conforme se vaya avanzando en el temario. Los plazos de entrega vendrán marcados por el profesor. La entrega en segunda convocatoria será conjunta de todos los problemas en una única entrega, cuyo plazo será establecido por el profesor.

SE6: Evaluación del proyecto a desarrollar de forma incremental durante cada uno de los temas. El proyecto se puede entregar en las 2 convocatorias en una única entrega cuya fecha la marcará el profesor.

SE4: Exposición pública del proyecto. La exposición se puede hacer en las 2 convocatorias siempre y cuando se haya entregado el proyecto dentro del plazo marcado por el profesor.

La nota final se calcula mediante la siguiente fórmula:

Nota final=SE1*0,1+SE2*0,3+SE6*0,4+SE4*0,2

Si en SE2, SE6 o SE4 se obtiene la calificación de “No Presentado”, la nota final de la asignatura será la de “No Presentado”.

Si la nota de SE2 y SE6 es distinta a “No Presentado” y la nota de SE2 o SE6 es menor a 5, la nota final se calcula con la siguient fórmula:

Nota final= Mínimo(4,SE2, SE6)

El sistema de calificaciones está especificado en el siguiente enlace:

http://www.uv.es/uvweb/universidad/es/estudios-postgrado/informacion-administrativa-postgrado/permanencia-calificaciones/calificaciones-1285897761928.html

La normativa aplicable se encuentra en el siguiente enlace:

http://www.uv.es/uvweb/universidad/es/estudios-grado/informacion-academica-administrativa/normativas/normativas-universidad-valencia-1285850677111.html

REFERENCIAS

Básicas

- Cuadernos de Ingeniería de Proyectos III: Dirección, Gestión y Organización de Proyectos. Salvador Capuz, Eliseo Gómez, Álvaro Torrealba et al. Servicio de la publicaciones de la Universidad Politécnica de Valencia, 2000.
- El Arte de Dirigir Proyectos. Ángel Díaz, 3ª Edición. Editorial RA-MA, 2010
- Gestión Integral de Proyectos. Luis Guerra, Adriano Coronel, Luis Martínez de Irujo, et al. Editorial FC, 2002
- COSMIC Function Points: Theory and Advanced Practices, Reiner Dumke, Alain Abran, 2011.
- Java Unit Testing with JUnit 5 : Test Driven Development with JUnit 5. Shekhar Gulati, Rahul Sharma, CA Berkeley, Apress, 2017.
- Agile Project Management with Scrum. Ken Schwaber, Microsoft Press, 2009
- The Agile edge : managing projects effectively using Agile Scrum. Brian Vanderjack, Business Expert Press, 2015.
- Test-driven development by example. Kent Beck, Addison-Wesley, 2003

Complementarias

- Dirección y Gestión de Proyectos. Alberto Domingo. RA-MA. 2000
- Desarrollo y Gestión de proyectos Informáticos. Steve McConnell, Editorial McGraw-Hill, 1997.
- Microsoft Project 2000. Paso a Paso, Carl S. Chatfield, Timothy D. Johnson, Editorial McGraw-Hill, 2000
- Microsoft Project para Dummies, Nancy Stevenson, Editorial WILEY, 2004
- Professional Java tools for extreme programming: Ant, Xdoclet, JUnit, Cactus, and Maven. Richard Hightower et al. Indianapolis, IN : Wiley, 2004
- Sams teach yourself extreme programming in 24 hours. Stewart Baird, Indianapolis, IN : Sams, 2003
- Extreme programming explained: embrace change. Kent Beck with Cynthia Andres, Boston, MA :Addison-Wesley, 2005

- Pruebas de Software y JUnit. Daniel Bolaños, Almudena Sierra, Miren Alarcón, Editorial Pearson Prentice-Hall, 2007