

CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL



**PLANIFICACIÓN DE LA DOCENCIA
UNIVERSITARIA**

GUÍA DOCENTE

**INTRODUCCIÓN A LA CONSTRUCCIÓN DE
SOFTWARE**

1. DATOS DE IDENTIFICACIÓN DE LA ASIGNATURA.

Título:	GRADO DE INGENIERÍA EN DESARROLLO DE CONTENIDOS DIGITALES				
Facultad:	Centro Universitario de Tecnología y Arte Digital (U-tad)				
Departamento/Instituto:					
Materia:	Fundamentos de sistemas software y hardware				
Denominación de la asignatura:	INTRODUCCIÓN A LA CONSTRUCCIÓN DE SOFTWARE				
Código:	0048018				
Curso:	Segundo				
Semestre:	Segundo				
Tipo de asignatura (básica, obligatoria u optativa):	Básica				
Créditos ECTS:	6				
Modalidad/es de enseñanza:	PRESENCIAL				
Lengua vehicular:	ESPAÑOL				
Equipo docente:	José Jesús García Rueda				
Profesor/a:	José Jesús García Rueda				
Grupos:	IDCD2				
Despacho:	Sala de profesores				
Teléfono:	916402811	Ext.	133	E-mail:	jose.rueda@live.u-tad.com
Página web:	http://u-tad.blackboard.com				

2. REQUISITOS PREVIOS.

Esenciales:
Haber cursado la asignatura "Introducción a la programación"
Aconsejables:
Haber aprobado las asignaturas "Introducción a la programación" y "Algoritmos y estructuras de datos"

3. SENTIDO Y APORTACIONES DE LA ASIGNATURA AL PLAN DE ESTUDIOS.

Campo de conocimiento al pertenece la asignatura.
Esta asignatura pertenece al Módulo de Fundamentos y a la Materia de Fundamentos de la Ingeniería del Software.
Relación de interdisciplinariedad con otras asignaturas del curriculum.
Esta asignatura es previa a "Diseño de software y metodologías de desarrollo", conformando la docencia en la materia de Ingeniería del Software y metodologías de desarrollo software.
Aportaciones al plan de estudios e interés profesional de la asignatura. Planificar y gestionar proyectos de desarrollo software mediante metodologías iterativas. Formar equipos de trabajo para sacar adelante proyectos de desarrollo software. Describir arquitecturas y diseños de software mediante lenguajes gráficos.

4. RESULTADOS DE APRENDIZAJE EN RELACIÓN CON LAS COMPETENCIAS QUE DESARROLLA LA ASIGNATURA.

COMPETENCIAS ESPECÍFICAS	RESULTADOS DE APRENDIZAJE RELACIONADOS CON LAS COMPETENCIAS ESPECÍFICAS
<p>CE-4. Conocimiento de la estructura, arquitectura, organización, funcionamiento e interconexión de los sistemas informáticos y los fundamentos de su programación.</p> <p>CE-6. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad.</p> <p>CE-7. Capacidad para planificar, concebir y desplegar proyectos, servicios y sistemas informáticos, llevando a cabo su puesta en marcha y su mejora continua y valorando su impacto económico y social.</p> <p>CE-10. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.</p> <p>CE-15. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.</p>	<p>Conocer técnicas avanzadas de programación empleadas en proyectos software profesionales y de dimensión y complejidad considerable.</p> <p>Conocer y realizar las distintas fases de la construcción del software, desde los requisitos hasta su implementación.</p> <p>Modelar y diseñar soluciones atendiendo a los compromisos de eficiencia, modularidad, calidad y mantenibilidad.</p>

5. CONTENIDOS / TEMARIO / UNIDADES DIDÁCTICAS

- Tema 1. Análisis y diseño orientado a objetos.
- Tema 2. Ingeniería de requisitos de usuario.
- Tema 3. Modelado del dominio.
- Tema 4. GRASP.
- Tema 5. Modelado del diseño.
- Tema 6. Modelado de la implementación.
- Tema 7. Prueba de programas y gestión de la configuración.

6. CRONOGRAMA

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
Tema 1	Semana 1
Tema 1	Semana 2
Tema 2	Semana 3
Tema 2	Semana 4
Tema 3	Semana 5
Tema 3	Semana 6
Tema 4	Semana 7
Tema 4	Semana 8
Tema 5	Semana 9
Tema 5	Semana 10
Tema 6	Semana 11
Tema 6	Semana 12
Tema 7	Semana 13
Repaso	Semana 14

7. MODALIDADES ORGANIZATIVAS Y MÉTODOS DE ENSEÑANZA

MODALIDAD ORGANIZATIVA	MÉTODO DE ENSEÑANZA	COMPETENCIAS RELACIONADAS	HORAS PRESENCIALES	TRABAJO AUTÓNOMO	TOTAL DE HORAS
Clases teóricas	Lección magistral	CE4, CE6, CE7, CE10, CE15	21	1	22
Seminarios y talleres	Estudio de casos Resolución de ejercicios y problemas		0	0	0
Clases prácticas	Aprendizaje basado en problemas Aprendizaje orientado a proyectos	CE4, CE6, CE7, CE10, CE15	22	0	22
Prácticas externas			0	0	0
Tutorías	Aprendizaje orientado a proyectos Aprendizaje basado en problemas	CE4, CE6, CE7, CE10, CE15	7	0	7
Actividades de evaluación		CE4, CE6, CE7, CE10, CE15	7	0	7
Estudio y trabajo en grupo	Aprendizaje cooperativo	CE4, CE6, CE7, CE10, CE15	1	21	23
Estudio y trabajo autónomo, individual	Estudio de casos Resolución de ejercicios y problemas Aprendizaje basado en problemas Aprendizaje orientado a proyectos	CE4, CE6, CE7, CE10, CE15	0	68	68

Las clases teórico-prácticas seguirán el siguiente esquema:

1. Desarrollo teórico de contenidos y ejemplos. Presentación por diapositivas de los conceptos e ideas más significativos, junto con sus ejemplos adecuados. Los alumnos podrán y deberán interrumpir al profesor para resolver cualquier duda o comentario que tengan.
2. Desarrollo de uno o varios ejercicios de forma individualizada por parte de los alumnos. El profesor irá pasando por todas las mesas para resolver de forma personal las dudas que puedan surgir en la resolución de los mismos. Cada

alumno deberá resolver ese mismo día todos los ejercicios que se planteen, incluso fuera de la clase si no le da tiempo a terminarlos en la misma. Cuando se estime oportuno, el alumno deberá entregar la solución a los ejercicios a través del Campus Virtual en el plazo que se fije a tal efecto.

Las clases de exposición y debate seguirán el siguiente esquema:

1. Exposición por parte de los alumnos de trabajos de gestión de procesos de desarrollo software. Estos trabajos se realizarán y expondrán por grupos.
2. La asistencia a las exposiciones es obligatoria tanto para los alumnos que exponen como para el resto.
3. Cada exposición durará en torno a 20 minutos. Tanto el profesor como los alumnos oyentes pueden realizar preguntas durante la exposición, pudiéndose ampliar su duración hasta los 25-30 minutos.

Las clases serán totalmente participativas y amenas, intentando que la comunicación sea bidireccional (de profesor a alumnos y viceversa) el mayor tiempo posible. Además, en la medida de lo posible, las clases serán eminentemente prácticas. Se intentará que los ejemplos y los ejercicios tengan relación con los contenidos digitales para una mayor motivación del alumnado.

8. SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CRITERIOS DE EVALUACIÓN	VALORACIÓN RESPECTO A LA CALIFICACIÓN FINAL (%)
Problemas, ejercicios y trabajos de diseño	Cada problema se evaluará de 0 a 10.	20%
Proyectos de planificación, diseño e implementación	De 0 a 10 cada uno. Es obligatorio realizar todos los proyectos y obtener una nota mínima de 5 en cada uno para aprobar la asignatura.	40%
Examen final ordinario	De 0 a 10. Es obligatorio obtener al menos un 5 para aprobar la asignatura.	40%

Consideraciones generales acerca de la evaluación:

Según política del centro, es obligatorio que el alumno no falte injustificadamente a más del 80% de las clases para poder ser aplicable la evaluación continua de su calificación.

EVALUACIÓN CONTÍNUA

1. Periódicamente habrá problemas, ejercicios o trabajos que se habrán de entregar a través del campus virtual, antes de la fecha señalada a tal efecto, o exponer en clase si así se requiere.
 - En el caso de los trabajos, los alumnos expondrán en clase el trabajo, debiendo responder a las cuestiones que el profesor u otros alumnos realicen durante la exposición.
 - La nota media de estos problemas/ejercicios/trabajos de investigación supondrá un 20% de la nota final.
2. Es necesario obtener un 5 en todos los proyectos por separado para aprobar la asignatura. De ser así, la nota media de los proyectos contará como un 40% de la nota final.
3. Es necesario obtener al menos un 5 en la nota del examen final para poder aprobar la asignatura. De ser así, la nota del examen final contará como un 40% de la nota final.
4. Es necesario obtener al menos un 5 en la nota final para poder aprobar la asignatura. Existen dos oportunidades para ello: la convocatoria ordinaria y la extraordinaria.
5. La nota de los problemas/ejercicios/trabajos sirve tanto para la convocatoria ordinaria como para la extraordinaria, en caso de necesitarse. En la convocatoria extraordinaria no se pueden recuperar/repetir los problemas/ejercicios/trabajos.
6. De ser necesario, para la convocatoria extraordinaria se pueden repetir los proyectos suspendidos en la ordinaria, con las modificaciones pertinentes impuestas por el profesor. No es necesario repetir los proyectos aprobados en convocatoria ordinaria.

Los exámenes y los ejercicios constarán de uno o varios de los siguientes tipos de preguntas:

- Cuestiones teóricas cortas.
- Preguntas de tipo test sobre teoría o elegir el resultado final de un ejercicio.
- Problemas.
- Preguntas sobre los proyectos.

La evaluación de los trabajos de diseño se hará en función de:

1. La adecuación del contenido presentado por los alumnos a lo requerido en el enunciado del trabajo. El profesor podrá hacer preguntas durante la exposición, las cuales deberán ser respondidas por los alumnos, para evaluar este aspecto.

2. La calidad de la exposición de cada alumno.
3. La calidad de la presentación realizada por los alumnos.
4. La participación de los alumnos en las exposiciones de los trabajos de los demás.

Cualquier escrito que el alumno presente (problemas, exámenes, comentarios de los programas, etc.) deberá estar bien presentado, correctamente redactado (con las comas, puntos y puntos y aparte en su lugar adecuado) y sin faltas ortográficas. La nota del escrito podrá bajar hasta un 20% en caso contrario, ya que a un universitario se le exige calidad máxima en su expresión escrita.

Las notas de los exámenes y de los problemas y ejercicios no se guardan entre cursos académicos sucesivos.

La asignatura COMPLETA estará suspensa si se descubre que un alumno ha copiado a otro (ambos estarán suspensos) o bien ha copiado de un libro o de Internet. Además, la universidad abrirá expedientes disciplinarios a ambos alumnos, pudiendo desembocar incluso en su expulsión.

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía general

Básica:

- Larman, C. (2004); APPLYING UML AND PATTERNS: AN INTRODUCTION TO OBJECT-ORIENTED ANALYSIS AND DESIGN AND ITERATIVE DEVELOPMENT; Editorial: Prentice Hall; ISBN: 0131489062.
- Vliet, H. (2007); SOFTWARE ENGINEERING: PRINCIPLES AND PRACTICES; Editorial: Wiley.

Recomendada:

- Fowler, M. y Scott, K. (1999); UML DISTILLED: A BRIEF GUIDE TO THE STANDARD OBJECT MODELING LANGUAGE; Editorial: Addison Wesley; ISBN: 0-201-65783-X. Segunda edición.

Bibliografía recomendada por temas

Tema 1, 2, 3, 4, 5, 6. Larman, C. (2004); APPLYING UML AND PATTERNS: AN INTRODUCTION TO OBJECT-ORIENTED ANALYSIS AND DESIGN AND ITERATIVE DEVELOPMENT; Editorial: Prentice Hall; ISBN: 0131489062.

Tema 7. Vliet, H. (2007); SOFTWARE ENGINEERING: PRINCIPLES AND PRACTICES; Editorial: Wiley.

10.- OBSERVACIONES

Recursos necesarios para la docencia:

- Ordenador del profesor conectado a proyector. Software instalado: "Microsoft PowerPoint", "Acrobat Reader" y "Eclipse".
- Pizarra digital.
- Pizarra blanca.
- Un ordenador para cada alumno en el aula. Software instalado: "Microsoft PowerPoint", "Acrobat Reader" y "Eclipse".

Recursos necesarios para el trabajo personal del alumno, fuera de clase:

- Ordenador de gama doméstica.
- Conexión a internet.
- Software "Eclipse", cuenta en "Lucidchart" y en "Tom'splanner".
- Software "Acrobat Reader" y "Microsoft PowerPoint".