



Principios de Desarrollo de Software

Asignatura Obligatoria
Curso: 2º
Créditos ECTS: 6
Clases por Semana: 2 (1h 40')




1



Profesores

- Coordinador y Profesor de Teoría
 - Javier García Guzmán
 - E-mail: jgarcia@inf.uc3m.es
 - Tutorías:
 - Lunes de 15:00 a 17:00
 - Viernes de 13:00 a 15:00
 - Se recomienda concertar una cita por e-mail con antelación
 - Despacho: 2.I.C.05
- Profesores de Prácticas
 - Javier García Guzmán (G81)
 - E-mail: jgarcia@inf.uc3m.es
 - Tutorías:
 - Lunes de 15:00 a 17:00
 - Viernes de 13:00 a 15:00
 - Se recomienda concertar una cita por e-mail con antelación
 - Despacho: 2.I.C.05
 - Mat Max Montalvo Martínez (G82)
 - E-mail: mamontal@pa.uc3m.es
 - Tutorías:
 - Pendientes de fijar
 - Se recomienda concertar una cita por e-mail con antelación
 - Despacho: 2.2.B10 Izda



2019 - 2020 Principios de Desarrollo de Software 2

2

Información General

- Calendario
 - Teoría
 - Lunes de 9 a 11
 - Ejercicios y Prácticas
 - Grupo 81: Viernes de 11 a 13
 - Grupo 82: Jueves de 11 a 13
- Clases
 - Teoría: 7.1.J.06
 - Práctica:
 - Grupo 81: 4.0.F18
 - Grupo 82: 4.0.F18



La Ingeniería del Software en el Grado en Ingeniería Informática

Asignatura	PDS	IS	DPDS	MITC	TADS	DSIC	MDV	EV
1 Software Requirements	X	u	u		u	u	u	
2 Software Design	I	X	u		u	u	X	
3 Software Construction	X				X	X	X	
4 Software Testing	X	u/f	u/f		X		X	
5 Software Maintenance			X		X			
6 Software Configuration Management			X		X			
7 Software Engineering Management			X		X	X		X
8 Software Engineering Process			X					
9 Software Engineering Models and Methods		X	u		u		X	
10 Software Quality			X			X	u	
11 Software Engineering Professional Practice	X		X					
12 Software Engineering Economics			X					
13 Computing Foundations								
14 Mathematical Foundations								
15 Engineering Foundations								



Competencias

- **Cognitivas (Saber)**
 - Conocimiento sobre modelos de ciclos de vida y proceso de desarrollo de software basados en el desarrollo dirigido por pruebas
 - Conceptos éticos en el proceso de desarrollo de software
 - Cultura de desarrollo ágil
 - Técnicas facilitadoras del desarrollo ágil
 - Desarrollo Dirigido por las pruebas
 - Diseño Simple y Refactoring
- **Procedimentales/Instrumentales (Saber hacer)**
 - Desarrollar un componente software aplicando normativas de código
 - Desarrollar un componente software aplicando propiedad colectiva de código
 - Desarrollar un componente software aplicando desarrollo dirigido por pruebas
 - Desarrollar un componente software aplicando técnicas de prueba estructurales y funcionales
 - Desarrollar un componente software aplicando mecanismos de prueba automatizada
 - Desarrollar un componente software aplicando mecanismos de integración continua y automatizada
 - Desarrollar un componente software aplicando refactoring
 - Desarrollar un componente software aplicando patrones de diseño
- **Genéricas**
 - Abstracción, Análisis, Síntesis, Habilidad para organizar y planificar, Habilidad para resolver problemas

2019 - 2020 Principios de Desarrollo de Software 5

5

Contenido

- **PARTE I. Aspectos éticos y legales en la profesión de Ingeniero de Software.**
 - Tema 1) La profesión de Ingeniero de Software.
 - Tema 2) Código ético de la profesión de Ingeniero de Software.
- **PARTE II. Prácticas genéricas del desarrollo ágil**
 - Tema 3) Programación en Parejas.
 - Tema 4) Estándares de codificación.
 - Tema 5) Propiedad Colectiva de Código.
- **PARTE III. Desarrollo Dirigido por Pruebas**
 - Tema 6) Principios del Desarrollo Dirigido por Pruebas
 - Tema 7) Automatización de Pruebas Unitarias
 - Tema 8) Técnicas de Pruebas Funcionales
 - Tema 9) Técnicas de Pruebas Estructurales
 - Tema 10) Integración Continua y Automatizada
- **PARTE IV. Refactoring y Diseño Simple**
 - Tema 11) Refactoring
 - Tema 12) Principios de Diseño Simple
 - Tema 13) Patrones de diseño básicos

2019 - 2020 Principios de Desarrollo de Software 6

6

Bibliografía

- **Básica**
 - Beck, Ken Una explicación de la Programación Extrema, Addison-Wesley, 2000
 - Beck, Ken, et al. Test-Driven Development By Example, Three Rivers Institute, 2002
 - Craig S. Larman Applying UML and Patterns, Pearson Education , 3er Edition, 2012
 - Fowler, Martin et al. Refactoring: Improving the Design of Existing Code, Addison-Wesley, 1999
 - Lee Copeland A Practitioner's Guide to Software Test Design. , Artech House Publishers .
- **Complementaria**
 - Roger S. Pressman. Ingeniería de Software. Un enfoque práctico. McGraw-Hill, 7ª Edición, 2009
 - Software Testing: a craftsman's approach. Paul C. Jorgensen. CRC Press.
 - Test & Controle des Logiciels: Methodes, Techniques & Outils. Spyros Xanthakis, Michel Maurice, Antonio de Amescua, Olivier Houri, Luc Griffet. EC2.



Bibliografía por temas

TEMA	REFERENCIA BIBLIOGRÁFICA
PARTE I. Aspectos éticos y legales en la profesión de Ingeniero de Software.	
Tema 1) La profesión de Ingeniero de Software.	Código de Ética y Práctica Profesional de la Ingeniería del Software
Tema 2) Código ético de la profesión de Ingeniero de Software.	Código de Ética y Práctica Profesional de la Ingeniería del Software
PARTE II. Prácticas genéricas del desarrollo ágil	
Tema 3) Programación en Parejas.	Beck, Ken Una explicación de la Programación Extrema, Addison-Wesley, 2000, Capítulo 7
Tema 4) Estándares de codificación.	Beck, Ken Una explicación de la Programación Extrema, Addison-Wesley, 2000, Capítulo 7
Tema 5) Propiedad Colectiva de Código.	Beck, Ken Una explicación de la Programación Extrema, Addison-Wesley, 2000, Capítulo 7
PARTE III. Desarrollo Dirigido por Pruebas	
Tema 6) Principios del Desarrollo Dirigido por Pruebas	Beck, Ken, et al. Test-Driven Development By Example, Three Rivers Institute, 2002, Part I and Chapter 25 and 26
Tema 7) Pruebas Funcionales	Lee Copeland. A Practitioner's Guide to Software Test Design. Artech House Publishers. - Capítulo 3 y 4 Test & Controle des Logiciels: Methodes, Techniques & Outils. Spyros Xanthakis, Michel Maurice, Antonio de Amescua, Olivier Houri, Luc Griffet. EC2.
Tema 8) Pruebas Estructurales	Lee Copeland. A Practitioner's Guide to Software Test Design. Artech House Publishers. - Capítulo 10
Tema 9) Integración Continua Automatizada	Jean-Marcel Belmont. Hands-On Continuous Integration and Delivery. Packt Publishing. 2018
PARTE IV. Refactoring y Diseño Simple	
Tema 11) Refactoring	Fowler, Martin et al. Refactoring: Improving the Design of Existing Code, Addison-Wesley, 1999
Tema 12) Principios de Diseño Simple	UML y Patrones - Capítulo 9, 15 y 16
Tema 13) Patrones de Diseño Básicos	UML y Patrones - Capítulo 9, 15 y 16



Planificación - I

	LUNES	JUEVES/VIERNES
27/01 y 30/01-31/01	Presentación de la Asignatura	Ejercicio Guiado 1 – Aspectos Éticos y Legales
03/02 y 06-07/02	Aspectos Éticos y Legales de la Profesión de Ingeniería de Software	Ejercicio Guiado 1 – Aspectos Éticos y Legales
10/02 y 13-14/02	Prácticas genéricas del desarrollo ágil	Ejercicio Guiado 2 – Normativas y propiedad colectiva de código
17/02 y 20-21/02	Integración Continua y Automatizada	Ejercicio Guiado 2 – Propiedad colectiva de código, integración continua y Automatizada
24/02 y 27/02-28/02	Principios del Desarrollo Dirigido por Pruebas	Ejercicio Guiado 3 – Introducción a Junit - Automatización de Pruebas
02/03 y 05-06/03	Técnicas de Prueba Funcionales – Análisis de Clases de Equivalencia y Valores Límite	Ejercicio Guiado 3 – Desarrollo Dirigido por Pruebas
09/03 y 12-13/03	Técnicas de Prueba Funcionales – Análisis Sintáctico	Ejercicio Guiado 3 – Desarrollo Dirigido por Pruebas
16/03 y 19-20/03	Técnicas de Prueba Estructurales	Ejercicio Guiado 3 – Desarrollo Dirigido por Pruebas



Planificación - 2

	LUNES	JUEVES/VIERNES
23/03 y 26-27/03	Refactoring	Ejercicio Guiado 4 – Refactoring y Diseño Simple
30/03 y 02-03/04	Refactoring	Ejercicio Guiado 4 – Refactoring y Diseño Simple
13/04 y 16-17/04		Ejercicio Guiado 4 – Refactoring y Diseño Simple
20/04 y 23-24/04	Diseño Simple	Ejercicio Guiado 4 – Refactoring y Diseño Simple
27/04 y 30/04-01/05	Diseño Simple	Ejercicio Guiado 4 – Refactoring y Diseño Simple
04/05 y 07-08/05	Diseño Simple	Ejercicio Guiado 4 – Refactoring y Diseño Simple



Evaluación

- Convocatoria Ordinaria
 - Siguiendo el procedimiento de evaluación continua:
 - Evaluación Continua
 - Test de Teoría Periódicos: 30%
 - Ejercicios guiados (4 Ejercicios): 30%
 - Práctica Final (Examen): 40%
 - Siguiendo el procedimiento de examen final:
 - Examen Escrito
 - Examen de Teoría: 30%
 - Examen de Ejercicios: 30%
 - Práctica Final: 40%
 - Para superar la asignatura es imprescindible superar todas las partes
 - El estudiante solo deberá realizar examen escrito de las partes no superadas en la evaluación continua.
- Convocatoria extraordinaria
 - En caso de no superar algunas de las partes en la convocatoria ordinaria, en la convocatoria extraordinaria, el estudiante solo deberá examinarse de las partes no superadas en la ordinaria



Dinámica de Clase

- Teoría
 - Cuestiones acerca de las lecturas recomendadas para cada clase (30 minutos)
 - Ejercicios de Clase (45 minutos)
 - Resumen de la Teoría Relacionada (15 minutos)
 - Preguntas tipo test (10 minutos) – No en todas las clases, se avisará
- Práctica
 - Presentación del enunciado del ejercicio guiado
 - Presentación de un ejemplo acerca de la resolución del ejercicio guiado.
 - Revisión de Ejercicios Guiados Entregados anteriormente



Planificación de Ejercicios Guiados y Práctica Final

Ejercicio Guiado / Práctica Final	Publicación	Entrega
Ejercicio Guiado 1: Aspectos Éticos y Legales	28/01/2020	11/02/2020 ¹
Ejercicio Guiado 2: Normativas, propiedad colectiva de código e integración automatizada	11/02/2020	25/02/2020 ¹
Ejercicio Guiado 3: Desarrollo Dirigido por Pruebas	25/02/2020	24/03/2020 ¹
Ejercicio Guiado 4: Refactoring y Diseño Simple	24/03/2020	08/05/2020 ¹
Práctica Final	13/04/2020	29/05/2020 ¹

¹ Antes de 23:55



2019 - 2020 Principios de Desarrollo de Software

13

13

Evaluación de Ejercicios Guiados

Ejercicios Guiados	Valoración
Ejercicio Guiado 1: Aspectos Éticos y Legales	10
Ejercicio Guiado 2: Normativas y propiedad colectiva de código	10
Ejercicio Guiado 3: Desarrollo Dirigido por Pruebas	25
Ejercicio Guiado 4: Refactoring y Diseño Simple	25
Puntos Totales de los Ejercicios Guiados	70

Los criterios de evaluación se incluirán en el enunciado de cada ejercicio guiado
Para superar la parte de los ejercicios guiados es necesario obtener 30/60 puntos



2019 - 2020 Principios de Desarrollo de Software

14

14

Práctica Final - Alcance

- El desarrollo de un componente por parte de los alumnos incluyendo los siguientes aspectos
 - Prácticas genéricas (2/10)
 - Plan de Pruebas Automatizado (4/10)
 - Diseño y Desarrollo (Refactoring) (4/10)
- Puntos Totales: 10



Examen Final (Para los que no hagan evaluación continua)

- El examen final consiste en:
 - Examen de Teoría (Test)
 - 20 preguntas
 - 4 opciones (solo una respuesta válida)
 - Respuestas incorrectas descuentan 0,25 puntos cada una
 - Examen de Ejercicios
 - Técnicas de Prueba Funcionales
 - Técnicas de Prueba Estructurales
 - Refactoring y/o Diseño Simple



