



Universitat de Lleida

GUIA DOCENT
**ENGINYERIA DEL
PROGRAMARI**

Coordinació: SENDÍN VELOSO, MONTSERRAT

Any acadèmic 2021-22

Informació general de l'assignatura

Denominació	ENGINYERIA DEL PROGRAMARI			
Codi	102018			
Semestre d'impartició	1R Q(SEMESTRE) AVALUACIÓ CONTINUADA			
Caràcter	Grau/Màster	Curs	Caràcter	Modalitat
	Doble titulació: Grau en Enginyeria Informàtica i Grau en Administració i Direcció d'Empreses	3	OBLIGATÒRIA	Presencial
	Grau en Enginyeria Informàtica	3	OBLIGATÒRIA	Presencial
	Màster Universitari en Enginyeria Informàtica		COMPLEMENTES DE FORMACIÓ	Presencial
Nombre de crèdits assignatura (ECTS)	6			
Tipus d'activitat, crèdits i grups	Tipus d'activitat	PRALAB		TEORIA
	Nombre de crèdits	3		3
	Nombre de grups	2		1
Coordinació	SENDÍN VELOSO, MONTSERRAT			
Departament/s	INFORMÀTICA I ENGINYERIA INDUSTRIAL			
Distribució càrrega docent entre la classe presencial i el treball autònom de l'estudiant	6 ECTS = (10 h de Classe presencial + 15 h de Treball autònom de l'estudiant) x 6 = 150 h de treball 40% Presencial (equivalent a 60 h) 60% Treball autònom (equivalent a 90 h)			
Informació important sobre tractament de dades	Consulteu aquest enllaç per a més informació.			
Idioma/es d'impartició	Preferentment en Català (Castellà si algun estudiant mostra dificultats amb el Català).			
Distribució de crèdits	Juan Manuel Gimeno Illa 4.5 Montserrat Sendin Veloso 4.5			

Professor/a (s/es)	Adreça electrònica professor/a (s/es)	Crèdits impartits pel professorat	Horari de tutoria/lloc
SENDÍN VELOSO, MONTSERRAT	montse.sendin@udl.cat	9	

Informació complementària de l'assignatura

Assignatura obligatòria de 3r curs (1r quadrimestre) pertanyent a la formació comú a la branca d'informàtica.
Matèria: Anàlisi i Disseny d'Aplicacions.

RECOMANACIONS: Es donen per coneguts els coneixements d'Orientació a Objectes de Programació II i Estructures de Dades.

Objectius acadèmics de l'assignatura

- Conèixer les bases conceptuals i els diferents aspectes de la disciplina, entre ells els models de procés del cicle de vida del software
- Aplicar la tècnica dels casos d'ús
- Especificar textualment les necessitats funcionals i no funcionals d'un determinat sistema software plantejat a través d'un enunciat (i/o altres inputs procedents de l'usuari)
- Desenvolupar el diagrama de classes d'un determinat sistema software seguint els principis del Modelat Orientat a Objectes
- Utilitzar una eina de modelat basada en UML
- Comprendre el concepte del codi com alguna cosa que evoluciona en el temps
- Ser capaç de programar proves unitàries bàsiques.
- Comprendre els principis fonamentals del disseny orientat a objectes.
- Reconèixer el concepte de responsabilitat com a fonamental a l'hora de plantejar un disseny orientat a objectes

Competències

Competències transversals

- **EPS11:** Capacitat de comprendre les necessitats de l'usuari expressades en un llenguatge no tècnic.

Competències específiques

- **GII-CRI2:** Capacitat per planificar, concebre, desplegar i dirigir projectes, serveis i sistemes informàtics en tots els àmbits, liderant la seva engegada i la seva millora contínua i valorant el seu impacte econòmic i social.
- **GII-CRI12:** Coneixement i aplicació de les característiques, funcionalitats i estructura de les bases de dades, que permetin el seu adequat ús, i el disseny i l'anàlisi i implementació d'aplicacions basades en elles.
- **GII-CRI13:** Coneixement i aplicació de les eines necessàries per a l'emmagatzematge, processament i accés als Sistemes d'informació, inclosos els basats en web.
- **GII-CRI16:** Coneixement i aplicació dels principis, metodologies i cicles de vida de l'enginyeria de programari.
- **GII-CRI17:** Capacitat per dissenyar i avaluar interfícies persona computador que garanteixin l'accessibilitat i usabilitat dels sistemes, serveis i aplicacions informàtiques.

Continguts fonamentals de l'assignatura

Tema 1. Aspectes introductoris

- 1.1. Qüestions inicials al voltant de la Enginyeria del Software
- 1.2. Una mica d'història
- 1.3. Procés de desenvolupament de software
- 1.4. Models de procés de software
- 1.5. Conclusions

Tema 2. Anàlisi de Requeriments

- 2.1. Especificació de requeriments
- 2.2. La tècnica dels Casos d'Ús
- 2.3. Un pas més en l'especificació: Els *Diagrames de Seqüència del Sistema*
- 2.4. Conclusions

Tema 3. Anàlisi del Domini

- 3.1. *Diagrames de Classes de l'Anàlisi*
- 3.2. Un pas més en l'anàlisi del domini: els *Contractes de les operacions*
- 3.3. Conclusions

Tema 4. Introducció al Disseny i Proves unitàries

- 4.1. La necessitat del disseny de codi
- 4.2. El framework JUnit per a proves unitàries

Tema 5. Els principis SOLID

- 5.1. Principi de responsabilitat única
- 5.2. Principi Obert-Tancat
- 5.3. Principi de substitució de Liskov
- 5.4. Principi de segregació d'interfícies
- 5.5. Principi d'inversió de dependències

Tema 6. Disseny basat en responsabilitats

- 6.1. El concepte de responsabilitat
- 6.2. Patrons GRASP d'assignació de responsabilitats
- 6.3. Realització d'un cas d'ús d'exemple

Eixos metodològics de l'assignatura

Grups Grans: Classes Teoria (3 crèdits)

- Part teòrica: classes suportades amb transparències i/o apunts
- Part d'aplicació pràctica: es treballa sempre amb exemples. Es disposa d'una **col.lecció de problemes**, i a classe es treballa la resolució de problemes concrets. Es van proporcionant les solucions al llarg del quadrimestre
- **Metodologia: Aula inversa** (donar la volta al mètode tradicional)
 - Participació activa estudiants**
 - Se tendirà cap a un aprenentatge actiu on l'estudiant és el protagonista
 - Sessions participatives i dinàmiques (incorporació de preguntes/sondejos, qüestions a discutir, revisió de punts explorats, aportació idees, etc.)
 - Enfocades a facilitar la interacció, l'anàlisi i les experiències pràctiques
 - => Requereix compromís per part de l'estudiant
 - Previ a la classe:
 - Revisió de material específic (material, enllaços, vídeos) per compte propi
 - Inici classe:
 - Sondeig i reforç continguts revisats

Grups Mitjans: Classes Laboratori (3 crèdits)

- Classes dirigides i seguiment personalitzat per grups de pràctiques
- Us d'eines de Modelat en UML: **ArgoUML** i/o **Visual Paradigm**
- Eines de control de versions amb **GIT** i framework de testing amb **JUnit**
- Treball continuat al voltant d'un **enunciat de pràctica**, que simularà el desenvolupament d'un projecte de software, com a aplicació pràctica dels continguts de l'assignatura

Treball Autònom (no presencial):

- La pràctica es completarà en hores **No Presencials**
- **Es recomana** que l'alumne resolgui per compte propi els problemes de la **col.lecció de problemes**, a fi de practicar i obtenir feedback per part del professor

El **sistema d'avaluació** (detallat en l'apartat corresponent) consta de: 1) proves escrites (els 2 exàmens parcials); i 2) pràctiques (a desenvolupar en equip).

Pla de desenvolupament de l'assignatura

Setmana	Teoria (GG)	Laboratori (GM)	Treball autònom
1	Presentació Assignatura T1: Aspectes introductoris	T1: Aspectes introductoris	Estudi
2	T1: Aspectes introductoris	T1: Aspectes introductoris	Estudi
3	T2: Anàlisi de requeriments Especificació de requeriments	T2: Anàlisi de requeriments Especificació de requeriments	Estudi i resolució problemes (Col.lecció de problemes Anàlisi)

4	T2: Anàlisi de requeriments. La tècnica dels Casos d'Ús. Problemes	Ús eina modelat UML Aplicació pràctica tècnica de Casos d'Ús	Estudi, resolució problemes (Col.lecció de problemes Anàlisi) i desenvolupament pràctica anàlisi
5	T2: Anàlisi de requeriments. Especificació de Casos d'Ús. Problemes	Aplicació tècnica de Casos d'Ús en enunciat de pràctica T2: Anàlisi de requeriments Diagrames de Seqüència del Sistema	Estudi, resolució problemes (Col.lecció de problemes Anàlisi) i desenvolupament pràctica anàlisi <i>Entrega Anàlisi Requeriments (1ª part)</i>
6	T3: Anàlisi del Domini Tècnica de Modelat Orientat a Objectes	Aplicació DSS en enunciat de pràctica	Estudi, resolució problemes (Col.lecció de problemes Anàlisi) i desenvolupament pràctica anàlisi
7	T3: Anàlisi del Domini Tècnica de Modelat Orientat a Objectes Problemes	Ús eina modelat UML Aplicació pràctica Tècnica de Modelat Orientat a Objectes	Estudi, resolució problemes (Col.lecció de problemes Anàlisi) i desenvolupament pràctica anàlisi <i>Entrega Anàlisi Requeriments (2ª part)</i>
8	T3: Anàlisi del Domini Tècnica de Modelat Orientat a Objectes Problemes	Construcció Model del Domini enunciat de pràctica	Estudi, resolució problemes (Col.lecció de problemes Anàlisi) i desenvolupament pràctica anàlisi domini
9	Primer parcial		Desenvolupament pràctica anàlisi domini
10	T3: Anàlisi del Domini Contractes de les operacions T4: Introducció al disseny Concepte de proves	Problemes simples testing	Estudi, resolució de problemes (Col.lecció de problemes testing) i desenvolupament pràctica anàlisi domini
11	T4: JUnit 5 Objectes substituïts	Problemes testing amb substitucions	Estudi i resolució de problemes (Col.lecció de problemes testing) <i>Entrega Anàlisi Domini i Contractes</i>
12	T5: Principis SOLID Intro, OCP & LSP	Problemes de testing Aspectes avançats de JUnit	Estudi, resolució de problemes (Col.lecció de problemes testing) i desenvolupament pràctica testing
13	T5: Principis SOLID SRP, ISP & DIP	Ús de Git	Estudi, resolució de problemes (Col.lecció de problemes testing) i desenvolupament pràctica testing
14	T6: Patrons GRASP Concepte de responsabilitat	Ús de Git	Estudi, resolució de problemes (Col.lecció de problemes testing) i desenvolupament pràctica testing
15	T6: Patrons GRASP Expert, Creador, Baix acoblament	T6: Patrons GRASP Alta cohesió, Controlador	Estudi i desenvolupament pràctica testing
16	Setmana per al segon parcial		Desenvolupament pràctica testing
17	Setmana per al segon parcial		<i>Entrega Proves Unitàries</i>
18	Tutories		
19	Recuperació		

Sistema d'avaluació

Activd.	Descripció	Ponderació	Nota mínima	En grup	Presencial	Obligatòria	Recuperable
Parc1	Primer parcial	25%	3,0	No	Sí	Sí	Sí
Parc2	Segon parcial	25%	3,0	No	Sí	Sí	Sí
Actv1	Anàlisi de Requeriments	20%	No	Sí	No	Sí	No
Actv2	Model del Domini i Contractes	10%	No	Sí	No	Sí	No
Actv3	Proves unitàries	20%	No	Sí	No	Sí	No
Particip	Participació	10%	No	No	Sí	No	No

Nota final = $0,25 * \text{Parc1} + 0,25 * \text{Parc2} + 0,20 * \text{Actv1} + 0,10 * \text{Actv2} + 0,20 * \text{Actv3} + 0,10 * \text{Particip}$

- L'assignatura s'aprova si la **nota final** és superior a **5** i s'arriba a les notes mínimes als parcials.
- Totes les **activitats obligatòries** són un requisit per aprovar l'assignatura.
- La **part opcional** es valora amb **1 punt**. En cas de participar en les activitats proposades a classe, es pot arribar a obtenir un **11 (1 punt extra)**.

Altres consideracions i criteris:

- Tipologia dels exàmens parcials: fixació de conceptes i resolució de problemes.
- Per totes les activitats avaluables: Entregues programades, dates no prorrogables.
- **Cas de no arribar a la nota mínima a algun dels exàmens parcials**, cal anar al examen de RECuperació de la/es part/s corresponent/s.
 - Si tot i així no s'arriba a la nota mínima a algun dels parcials, o al 5 en la ponderació, la **nota final serà un 4,5 com a màxim**.
- L'examen de RECuperació és també una oportunitat per atansar l'aprovat (nota actual < 5), i inclús per a millorar nota (tenint en compte que prevaleix la nota obtinguda en ell).
 - En qualsevol cas cal examinar-se, com a mínim, del parcial amb nota més baixa.
- *Particip*: La participació de l'estudiant en activitats proposades addicionalment, a desenvolupar durant les sessions de classe teòrica, seran valorades **fins a 1 punt extra en la nota**. Poden consistir en: la sintetització del material facilitat, l'aportació d'idees i intervenció en debats, preparació de micro-temes acordats entre tots, etc. Serviran per dinamitzar les classes i a fomentar la participació activa, tal i com es menciona a la metodologia.

Bibliografia i recursos d'informació

Bibliografia bàsica

- Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice-Hall, 2005 (3^a ed.)
- Boni García: Mastering Software Testing with Junit 5. Packt, 2017

Bibliografia complementària

- Gerald Kotonya, Ian Sommerville: Requirements Engineering: Processes and Techniques. Wiley, 1998
- Robert Martin: Agile Software Development: Principles, Patterns, and Practices, Prentice-Hall, 2002.
- Lasse Koskela, Effective Unit Testing. A guide for Java developers. Manning, 2013