Universitat de Lleida

GUIA DOCENT

ENGINYERIA DEL PROGRAMARI

Any acadèmic 2013-14

ENGINYERIA DEL PROGRAMARI 2013-14

Informació general de l'assignatura

Denominació	ENGINYERIA DEL PROGRAMARI
Codi	102018
Semestre d'impartició	1r Q Avaluació Continuada
Caràcter	Obligatòria
Nombre de crèdits ECTS	6
Crèdits teòrics	0
Crèdits pràctics	0
Departament/s	Informàtica i Enginyeria Industrial
Informació important sobre tractament de dades	Consulteu aquest enllaç per a més informació.
Idioma/es d'impartició	Català 60% Castellà 30% Anglès 10%
Distribució de crèdits	Juan Manuel Gimeno Illa 4.2 Montserrat Sendin Veloso 4.2

ENGINYERIA DEL PROGRAMARI 2013-14

Juan Manuel Gimeno Illa
Montserrat Sendin Veloso

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
- Conèixer els models de procés del cicle de vida del software utilitzats al llarg dels anys
- Comprendre la filosofia de desenvolupament utilitzada en el Procés Unificat
- Aplicar la tècnica dels casos d'ús
- Especificar textualment les necessitats funcionals d'un determinat sistema software plantejat a través d'un enunciat i/o altres inputs procedents de l'usuari
- Especificar textualment els requeriments no funcionals d'un determinat sistema software plantejat a través d'un enunciat i/o altres inputs procedents de l'usuari
- Expressar gràficament el flux d'esdeveniments, com a conjunt d'entrades i sortides que descriu el comportament del sistema
- Desenvolupar el diagrama de classes de l'anàlisi seguint els principis del Modelat Orientat a Objectes
- Elaborar els contractes de les operacions
- 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 i saber aplicar els principis fonamentals del disseny orientat a objectes.
- Reconéixer el concepte de responsabilitat com a fonamental a l'hora de plantejar un disseny orientat a objectes.

Competències

Competències específiques de la titulació

- Coneixement i aplicació dels principis, metodologies i cicles de vida de l'enginyeria de programari.
- Capacitat per planificar, concebre, desplegar i dirigir projectes, serveis i sistemes informàtics en tots els àmbits, liderant la seva posada en marxa i la seva millora contínua i valorant el seu impacte econòmic i social.

Competències transversals de la titulació

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

Continguts fonamentals de l'assignatura

Tema I - *Aspectes introductoris*

- Qüestions inicials al voltant de la Enginyeria del Software

- Una mica d'història

- Procés de desenvolupament de software

- Models de procés de software

- Conclusions

Tema II - *Especificacions i Requeriments*

- Anàlisi de requeriments

  - Conceptes inicials
  - Tipus de requeriments
  - Exemples

- La tècnica dels Casos d'Ús

  - Conceptes i components
  - Identificació i especificació dels casos d'ús
  - Exemples

- Un pas més en l'especificació: Els Diagrames de Seqüència del Sistema

- Conclusions

Tema III - *Anàlisi del Domini*

- Diagrames de Classes de l'Anàlisi

  - Concepció
  - Fonaments del Modelat Orientat a Objectes
  - Construcció del Model del Domini
  - Exemples

- Un pas més en l'anàlisi: els Contractes de les operacions

- Conclusions

Tema IV - *Introducció al disseny i Proves unitàries*

- El codi com alguna cosa que varia en el temps.

  - El concepte de la degradació de codi.
  - Simptomes d'un codi degradat.

- El framework JUnit per a proves unitàries

  - Les proves com a especificacions executables
  - Les proves com a facilitadors de canvi.
  - Codi testejable i codi flexible.

Tema V - *Els principis SOLID*

  - Principi de responsabilitat única
  - Principi Obert-Tancat
  - Principi de substitució de Liskov
  - Principi de segregació d'interfícies
  - Principi d'inversió de dependències

Tema VI - *Disseny basat en responsabilitats*

  - El concepte de responsabilitat
  - Patrons GRASP d'assignació de responsabilitats

Eixos metodològics de l'assignatura

PART PRESENCIAL

Grups Grans

• **Classes Teoria (3 crèdits)**

- Classes suportades amb transparències i/o apunts
- Es treballa sempre amb exemples

Grups Mitjans

• **Classes Laboratori (3 crèdits)**

- Us de l'eina de Modelat en UML: Visual Paradigm
- Treball continuat al voltant d'un cert **enunciat de pràctica**, que simularà el desenvolupament d'un projecte de software

PART 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 d'obtenir feedback per part del professor

Sistema d'avaluació

Avaluació continuada

• **50% Teoria**

- **Parcial 1**: 25%
- **Parcial 2**: 25%
- Si Parcial 1 ó Parcial 2 < 4 = Recuperació de la/es part/s corresponent/s
- Tipologia d'examen: fixació de conceptes i resolució de problemes

• **50% Pràctica**

- Treball en parelles (Anàlisi) / individualment (Disseny)
- Entregues programades, dates no prorrogables
  - Anàlisi:
    - Anàlisi de Requeriments: 15%
      - Pre-entrega: Diagrama de Casos d'Ús
    - Model del Domini: 15%
  - Disseny:
    - Proves unitàries: 20%

• **Requisits:**

- **Mínim de 4** en <u>cada</u> part teòrica per ponderar amb la nota de pràctiques
- **Aprovat** = Nota Final >= 5

Bibliografia i recursos d'informació

BIBLIOGRAFIA BÀSICA

- Temes introductoris:

- I. Sommerville: “Ingeniería de Software”. Prentice-Hall, 2005 (7ª ed.)

- Requeriments:

- G. Kotonya, I. Sommerville: “Requirements Engineering: Processes and Techniques”. Wiley, 1998
- A. Sutcliffe: “User-Centred Requirements Engineering. Theory and Practice”. Springer-Verlag, 2002

- Proves unitàries:

- P.Tahchiev et al.: Junit in Action (2nd edition). Manning, 2011.

- Metodologia orientada a objectes:

- C. Larman: “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development” Prentice-Hall, 2005 (3ª ed.)
  Versió espanyola: “UML y Patrones” Prentice-Hall, 2002 (2ª ed.)
- Robert C. Martin: "Agile Software Development: Principles, Patterns, and Practices", Prentice-Hall, 2002.
- G. Booch, J. Rumbaugh, I. Jacobson: “El Lenguaje Unificado de Modelado”. Addison-Wesley, 2006 (2ª ed.)
- J. Rumbaugh, I. Jacobson, G. Booch: “El Lenguaje Unificado de Modelado. Manual de referencia”. Addison-Wesley, 2000
- I. Jacobson, G. Booch, J. Rumbaugh: “El Proceso Unificado de Desarrollo de Software”. Addison-Wesley, 2000

BIBLIOGRAFIA COMPLEMENTARIA

- Temes introductoris:

- R. S. Pressman: “Ingeniería de Software: Un enfoque práctico”. McGraw-Hill, 2005 (6ª ed.)

- Requeriments:

- D. Kulak, E. Guiney: “Use Cases, Requirements in Context”. Addison Wesley, 2000
- I. Jacobson: “Object-Oriented Software Engineering. A Use Case Driven Approach”. Addison-Wesley, 1992

- Metodologia orientada a objectes:

- M. Fowler: "Refactoring: Improving the Design of Existing Code", Addison-Wesley, 1999.
- M. Fowler, K. Scout: “UML Gota a Gota”. Addison-Wesley, 1999
- J. Conallen: “Building Web Applications with UML”. Addison Wesley, 1999