

	ENSENYAMENT D'ENGINYERIA EN INFORMÀTICA		
	ASSIGNATURA: Enginyeria de Software - I		
	PROFESSOR/A RESPONSABLE: Montserrat Sendín Veloso		
	CURS: Primer	CRÈDITS: 6	TIPUS: Troncal

1. OBJECTIUS

L'enginyeria del software és la part de la informàtica que defineix conceptes i metodologies per a desenvolupar aplicacions de qualitat. Com pot intuir-se, és una disciplina essencial sobre la qual es vertebrava el desenvolupament de programes que permeten que els sistemes realitzin les seves funcionalitats. És, per tant, una disciplina amb molts i variats components que precisen de moltes hores d'assimilació de conceptes teòrics i altres tantes de practicar aquests conceptes perquè puguin ser assimilats eficientment.

Per aquest motiu, l'estudi de la disciplina es divideix en tres assignatures troncal, Enginyeria del Software I, II i III, perquè l'alumne pugui fer de manera progressiva un recorregut complet pels coneixements principals que li conduiran a una visió global de la disciplina.

En aquesta primera assignatura es presenta una visió de conjunt de la disciplina amb l'objectiu d'assentar les bases conceptuals i introduir a l'alumne als diferents aspectes de la mateixa. Els continguts es centraran en les metodologies iteratives i orientades a objectes, que són les més utilitzades actualment.

En l'aspecte pràctic s'introdueixen tècniques de modelatge orientat a objectes, centrant el focus en els aspectes d'anàlisi (especificació) d'aplicacions. L'assignatura posterior (Enginyeria del Software-II) es centrarà en aspectes de disseny orientat a objectes.

Per tal d'expressar les nostres anàlisis, utilitzarem el llenguatge de modelització UML i l'eina CASE per a UML integrada en l'entorn Netbeans, un IDE multiplataforma, gratuït i redistribuïble.

2. ESTRUCTURA

L'assignatura s'imparteix totalment en format semipresencial durant el primer quadrimestre del curs.

L'organització es basa en tres tipus de sessions:

Teoria: Classes a realitzar durant les sessions presencials. Al tractar-se d'una formació semipresencial l'alumne disposarà sempre del contingut de cada sessió en format electrònic al campus virtual.

Es presenten els conceptes i metodologies, els quals són aplicats, sempre que és possible, sobre exemples concrets. De fet, al llarg del curs, treballarem sobre un mateix exemple o parell d'exemples (exemple que reprendrem a l'assignatura següent al parlar de temes de disseny).

Problemes: Les sessions de resolució de problemes es presenten de forma intercalada entre les classes teòriques amb l'objectiu de reforçar l'aprenentatge dels continguts vistos en aquelles.

Els **alumnes** treballareu sobre els problemes de la llista i es comentaran possibles solucions, solucions alternatives, errors, etc. Normalment, en hores de classe s'aniran solucionant els problemes i, quan sorgeixin dubtes es plantejaran i s'aniran resolent. D'aquesta manera es podran detectar els aspectes que no queden clars per tal d'anar veient els "típics" errors que es comenten.

Podrà fer-se entrega dels problemes desenvolupats, contribuint d'aquesta forma a configurar la nota final de l'assignatura.

Pràctiques: Al tractar-se d'un ensenyament semipresencial, en la part de laboratori l'alumne ha d'exercitar-se per compte propi amb les tecnologies i metodologies explicades. Els dubtes que puguin sorgir durant la resolució de la pràctica de curs proposada podran ser consultats amb el professorat corresponent (ja sigui mitjançant correu electrònic, mitjançant el campus virtual o en les hores de consulta assignades).

La pràctica es desenvoluparà en grups, a mode de projecte de software.

L'eina CASE triada serà la que s'utilitzarà per a desenvolupar una pràctica que s'anirà entregant al llarg del curs en diferents entregues programades.

3. PROGRAMA

1. Tema introductor

- Importància i definició del Software
- Qüestions inicials al voltant de la Enginyeria del Software
- Problemàtica i cost del software
- Procés de desenvolupament del software
- Models de procés del software
- Procés Unificat de desenvolupament
- Conclusions

2. Tècniques de Modelat

- Per què modelem? Principis de modelat
- Modelat Orientat a Objectes
- Llenguatge Unificat de Modelat (UML)

3. Especificacions i Requeriments

- Conceptes inicials
- Tipus de requeriments
- Procés d'Enginyeria de Requeriments
- Documents de requeriments
- Casos d'ús i documentació associada

- Conclusions
4. Metodologia Orientada a Objectes
- Diagrama de classes de l'anàlisi
 - Concepció
 - Classes, objectes i atributs
 - Relacions entre classes
 - Exemples
 - Glossari de termes
 - Diagrames de robustesa
 - Concepció
 - Exemples
 - Conclusions
5. Introducció al disseny
- Introducció al disseny software
 - Introducció a l'arquitectura software
 - Patrons d'assignació i responsabilitat (GRASP)

4. MATERIALS DE L'ASSIGNATURA I PROGRAMARI

- Transparències de tots els temes
- Col·lecció de problemes
- Tutorials d'ús de l'entorn
- Plantilles de lliurament de la documentació
- Exemple orientatiu per dur a terme la documentació

Pluggin de UML para Eclipse

5. BIBLIOGRAFIA

- Temes introductoris
 - *Ian Sommerville, Ingeniería de Software (6ª edición), Addison-Wesley, 2002*
 - *Roger S. Pressman, Ingeniería de Software: Un enfoque práctico (5ª edición), McGraw-Hill, 2002*
- Metodologia orientada a objectes
 - *Craig Larman, UML y Patrones, Prentice-Hall, 2003*
 - *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*
- Requeriments

- *D. Kulak, E. Guiney, Use Cases, Requirements in Context.* Addison Wesley, 2000
- *G. Kotonya, I. Sommerville, Requirements Engineering.* Wiley, 1998
- *I. Jacobson, Object-Oriented Software Engineering. A Use Case Driven Approach.* Addison-Wesley, 1992
- *Kotonya, G.; Sommerville, I. Requirements Engineering. Processes and Techniques.* JohnWiley, 1997.

Enllaços Web de referència:

- Manual de referencia UML (en anglès): <http://www.uhu.es/17124/refman.pdf>.
- Especificación UML (en anglès): <http://www.uhu.es/17124/omguml 1.3.pdf>.
- Carnegie Mellon Software Engineering Institute (SEI): <http://www.sei.cmu.edu>.

6. AVALUACIÓ

L'alumne per a superar l'assignatura podrà optar per dues modalitats d'avaluació:

A) **Avaluació continuada:** Es demana un treball continuat al voltant d'un cert enunciat de pràctica, mitjançant una sèrie d'entregues programades al llarg de tot el quadrimestre (amb dates concretes no prorrogables, de manera que es penalitzarà l'entrega de documentació al voltant del projecte fora del seu termini).

A més a més, a final de curs cal afrontar un examen validatiu que inclourà també una sèrie de qüestions de concepte i de procediment relacionades amb les tècniques aplicades.

Se suggereix a l'alumne que opti per aquesta modalitat d'avaluació.

B) **Avaluació per examen:** L'alumne no haurà de presentar aquesta sèrie de treballs durant el quadrimestre però, per contra, haurà de presentar la pràctica al final de curs, així com també superar un examen de l'assignatura que consta en aquest cas d'una part de fixació de conceptes i una altra de resolució de problemes relacionats amb les tècniques estudiades al llarg de l'assignatura.

Tot i que la modalitat B està sempre disponible, es recomana a l'alumne optar per la modalitat A.

La **nota final** s'obté d'aplicar una ponderació de les notes de cada part. Depenent de la modalitat aplicada, aquesta varia.

A) Avaluació continuada

- **Actitud, interès i realització i entrega de problemes:** 15%
- **Pràctiques:** 85%

B) Avaluació per examen

- **Examen:** 50%
- **Actitud, interès i realització i entrega de problemes:** 15%
- **Pràctiques:** 35%

Caldrà una nota mínima de **4** de les parts de teoria i pràctiques per poder fer aquesta ponderació.

No es convaliden notes de pràctiques de cursos anteriors.