

**Enginyeria del Software I**

| <b>Codi</b>  | <b>Tipus</b>                                                                  | <b>Curs/semestre</b>              | <b>Crèdits</b> |
|--------------|-------------------------------------------------------------------------------|-----------------------------------|----------------|
| <b>21290</b> | <ul style="list-style-type: none"><li>• Troncal</li><li>• Semestral</li></ul> | <b>3er curs /<br/>5è semestre</b> | <b>6</b>       |

**Professors:**

| <i>Nom</i>   | <i>Dpt/Unitat</i>         | <i>Despatx</i> | <i>Adreça e-mail</i> | <i>Telèfon</i> |
|--------------|---------------------------|----------------|----------------------|----------------|
| Daniel Ponsa | Ciències de la Computació | S/246          | daniel@cvc.uab.es    | 93 581 2440    |

**Objectius:**

Un primer objectiu és donar una visió global i ordenada del procés de desenvolupament del software que no sigui simplement el de la programació --que constitueix, només, una fase dins de tot el procés de la enginyeria del software.

El temari es centra en les dues primeres etapes del cicle de vida del software: anàlisi i disseny. Per cada una de aquestes dues grans parts del curs s'estudiarà en detall la metodologia orientada a l'objecte. L'assignatura es centra bàsicament en l'estudi de UML com a llenguatge de modelat i RUP com a paradigma de desenvolupament.

L'objectiu és que l'alumne sigui capaç d'utilitzar aquestes metodologies per tal de realitzar l'anàlisi i el disseny de problemes reals, que li permetrà abordar la creació de software de manera rigorosa.

Coneixements

- Entendre què és el software, quins son els objectius que ens plantegem al desenvolupar un software i la problemàtica del associada al seu desenvolupament.
- Conèixer els principals paradigmes de desenvolupament del software.
- Aprendre què és l'anàlisi de requeriments i perquè és important. Abstracció i partició
- Aprendre què és el disseny. Com aconseguir un bon disseny. Modularitat efectiva.
- Introducció a l'UML: notació i construcció dels diagrames.
- Introducció a un procés de desenvolupament del software: RUP.

### Habilitats

- A partir d'un problema obert enunciat en llenguatge natural, saber analitzar que ens esta demanant el client: crear un document d'especificació.
- A partir del la informació obtinguda en el document d'especificació modelar-ho a partir dels diferents diagrames de UML.
- Ser capaç de veure que a l'hora de modelar hi ha punts en que no disposem de tota la informació i per tant s'ha de tornar a analitzar el problema: Iterar.
- Portar a terme aquestes activitats en un context orientat a objecte: els models (diagrames) que obtenim son diferents punts de vista del mateix problema on cada un modela algun característica de les metodologies orientades a objecte.
- Consciència de que un Enginyer en Informàtica no és només un programador i de que desenvolupar software és molt més que programar.
- Saber buscar i seleccionar la informació necessària per assolir els objectius plantejats.

### Competències

- Capacitat d'anàlisi i síntesi.
- Capacitat de resolució de problemes.
- Raonament crític.
- Motivació per la qualitat.
- Capacitat de resolució de problemes.
- Comunicació oral i escrita.
- Capacitat d'organització i planificació.
- Aprenentatge autònom.
- Lideratge

### **Capacitats prèvies:**

Es requereix que l'alumne tingui coneixements i capacitats de:

- Coneixement de Programació.
- Coneixement dels principis bàsics dels llenguatges orientats a objecte.
- Estructures de dades bàsiques.

Aquests conceptes corresponen a continguts de les assignatures:

- Algorismes i programació
- Llenguatges de programació.
- Estructures de dades.

## Continguts

### T0. Presentació de l'assignatura

- Presentació dels continguts, normativa i desenvolupament de l'assignatura

### T1. Principis de l'Enginyeria del Software.

- **Definició i objectius de l'ES.** Definició de software. Característiques del software. Aplicacions del software. Definició d'ES. Objectius de l'ES.
- **Evolució del software.** Etapes. Crisi del software: problemes i causes.
- **Procés, mètode i eina.** Definicions. Activitats en el procés de desenvolupament del software.
- **Paradigmes del desenvolupament del software.** Model lineal seqüencial vs iteratiu.

### T2. Anàlisi de Requeriments del Software.

- **Introducció.** Tipus de requeriments. Tasques a realitzar.
- **Comprensió del problema.** Tècniques de comunicació. Principis de l'anàlisi.
- **Especificació de requeriments.** Propietats desitjables Estàndards. Revisió i validació.

### T3. Disseny del Software

- **Introducció.** Procés de disseny. Disseny de dades, disseny arquitectònic, disseny de la interfície, disseny procedimental. Principis (objectius) del disseny.
- **Conceptes del disseny.** Abstracció. Modularitat. Refinament.
- **Disseny modular efectiu.** Independència funcional. Cohesió Acoblament. Heurístiques per a un disseny modular efectiu.

### T4. El Llenguatge Unificat de Modelat (UML)

- **Introducció.** Visió general d'UML. Notació.
- **Model estructural.** Classes. Relacions. Mecanismes comuns. Diagrames. Diagrames de classes.
- **Model de comportament.** Interaccions. Casos d'ús. Diagrames de casos d'ús. Diagrames d'interacció. Diagrames d'activitats. Diagrames d'estats.
- **Model arquitectònic.** Diagrames de components. Diagrames de desplegament.
- **Exemples.** Anàlisi de casos d'estudi.

### T5. Processos de Desenvolupament de Software

- **Introducció.** Objectius comuns.
- **Metodologies.** Propostes populars. Esquemes tradicionals vs àgils..

### T6. El Procés Unificat de Rational (RUP)

- **Introducció.** Gestió del desenvolupament del software. Les 6 "best practices".
- **El procés unificat.** Què és RUP?
- **Estructuració del RUP.** Estructura estàtica i dinàmica.
- **Exemples.** Anàlisi de casos d'estudi.

### T7. Tècniques d'Estimació en Projectes SW

- **Introducció.** Planificació de projectes. Dificultats en la planificació de projectes SW.
- **Estimació del treball.** Unitats. Mètodes T-shirt sizing i Planning Poker.

## **Metodologia docent:**

Aquesta assignatura s'imparteix en règim de "docència tutoritzada". Això implica que no hi ha classes tradicionals de teoria o problemes. En el seu lloc, es realitzen sessions de tutoria grupals. En aquestes sessions els alumnes poden aclarir els dubtes que els hagin sorgit sobre parts específiques del temari, que hauran d'haver treballat prèviament pel seu compte a partir de bibliografia recomanada.

No obstant, els alumnes que ho desitgin, poden optar al règim de docència normal, cursant l'assignatura 'Enginyeria del Software I' que s'imparteix a Enginyeria Informàtica del campus de Bellaterra. Només cal que ho comuniquin al professor a principi de curs, i ell s'encarregarà de coordinar aquest fet. Els alumnes que s'acullin a aquesta opció hauran de seguir el procés que s'especifiqui a la guia docent de l'assignatura 'Enginyeria del Software I' de Bellaterra.

## **Informació de les sessions de tutoria grupals:**

El calendari i contingut treballat a les sessions s'especificarà en el Campus Virtual.

## **Pràctiques:**

Els alumnes amb les pràctiques aprovades anteriorment les poden convalidar. Els alumnes amb les pràctiques pendents, poden realitzar-les en les sessions que es programin a l'assignatura 'Enginyeria del Software I' que s'imparteix a Enginyeria Informàtica del campus de Bellaterra. Cal que ho comuniquin al professor a principi de curs, i ell s'encarregarà de coordinar-ho.

## **Avaluació:**

### **Opció Règim docència tutoritzada**

L'avaluació de l'assignatura es durà a terme mitjançant les evidències d'aprenentatge recollides en els següents processos:

E1. Realització d'una prova escrita (examen).

E2. Realització d'una pràctica.

#### **E1: Avaluació de coneixements teòrics i pràctics**

Les evidències de l'apartat E1 es recolliran en una prova escrita, on s'avaluarà amb una nota entre 0 i 10 (ambdós inclosos) els conceptes que formen part del temari. És necessari tenir al menys un 5 en aquesta nota per poder aprovar l'assignatura.

Hi haurà dues convocatòries per realitzar aquesta prova escrita (al Gener i al Juliol). Per a cada convocatòria s'establirà un dia de revisió de la correcció.

#### **E2: Avaluació de la pràctica**

Els alumnes que ho desitgin, poden convalidar les pràctiques si les van aprovar anteriorment. En altre cas, poden realitzar les pràctiques en les sessions que es programin a l'assignatura 'Enginyeria del Software I' que s'imparteix a Enginyeria Informàtica del campus de Bellaterra.

#### **Nota final de l'assignatura**

Per superar l'assignatura, cal obtenir un mínim de 5 a la nota de E1 i a la nota final de l'assignatura, que es determina a partir de la següent fórmula.

$$\text{NotaFinal} = (0,7 * \text{NotaE1}) + (0,3 * \text{NotaE2})$$

### **Opció Règim docència normal a Bellaterra**

Els alumnes que s'acullin a aquesta opció hauran de seguir el procés d'avaluació que s'especifiqui a la guia docent de l'assignatura 'Enginyeria del Software I' que s'imparteix a Enginyeria Informàtica del campus de Bellaterra.

### **Bibliografia bàsica:**

- R. Pressman, **Ingeniería del software, un enfoque práctico**. Mc Grah-Hill, 5a ed., 2002.
- I. Sommerville, **Software engineering**. 6a Ed., Addison-Wesley , 2002.
- T. Pender, **UML Weekend Crash Course**, Wiley Publishing, Inc., 2002
- T. Quatrani. **Visual modeling with Rational Rose 2000 and UML**. Addison-Wesley, 2000.
- P. Krutchen. **The Rational Unified Process. An introduction**. Addison-Wesley, 2000.
- G. Booch, J. Rumbaugh, I. Jacobson. **El lenguaje unificado de modelado**. Addison-Wesley, 1999.
- I. Jacobson, G. Booch, J. Rumbaugh. **El proceso unificado de desarrollo de software**. Addison-Wesley, 1999.
- J. Rumbaugh , I. Jacobson, G. Booch. **El lenguaje unificado de modelado: manual de referencia**. Addison-Wesley, 1999.

### **Bibliografia complementària:**

- E. Yourdon, **Análisis estructurado moderno**. Prentice-Hall, 1993.
- J. Lladós, X. Roca. **Problemes d'enginyeria del software I**. Servei de Publicacions UAB, 1995.
- M.G. Piattini, J.A. Calvo-Manzano, J. Cervera, L. Fernández. **Análisis y diseño detallado de aplicaciones informáticas de gestión**. Ra-Ma, 1996.
- G. Booch, **Object oriented design with applications**. The Benjamin/Cummings Publishing Company, 1990.
- Yourdon Inc., **Yourdon systems method model-driven systems sevelopment**. Prentice Hall , 1993.
- P. Coad, E. Yourdon, **Object-oriented analysis**. Yourdon Press , 1991.
- P. Coad, E. Yourdon, **Object-oriented design**. Yourdon Press , 1991.