



Pla docent

Assignatura: Enginyeria de Software I

Estudis: Enginyeria Informàtica

Nombre de crèdits: 9

Curs acadèmic: 09-10

Trimestre: 1er

Professors:

Xavier Amatriain (xavier.amatriain@upf.edu)

Pau Arumí (pau.arumi@barcelonamedia.org)

David Garcia (david.garcia@barcelonamedia.org)

Nota: En les competències **i**:instrumental, **s**:sistèmica, **ip**:interpersonal

0. Competències de l'assignatura

Competències generals

1. Resolució de problemes

Instrumentals:

1. *Habilitats cognitives (comprensió, interrelació d'idees i pensaments, anàlisi, síntesi...)* (**i**): Totes aquestes competències generals són necessàries com a requisit previ per a les diferents competències que es presenten a continuació.
2. *Sentit comú* (**i**): De forma similar aquesta competència és l'eina més important a l'hora de poder assolir les altres competències que es descriuen en aquest apartat.
3. *Organització del temps* (**i**): Ser capaç de resoldre un problema complex de forma sub-òptima en un temps limitat, planificant abans la quantitat de temps que es destinarà a cada activitat.
4. *Cerca documental i de fonts de referència* (**i**): Per resoldre qualsevol problema de certa complexitat s'ha de ser capaç d'accedir a fonts de documentació externa, destriant la informació relevant per al problema que s'està adreçant.

Sistèmiques:

5. *Gestió de la informació* (**s**): Un problema complex se sòl presentar com un gran conjunt d'informació desestructurada però interrelacionada. S'ha de ser capaç d'identificar la informació substancial, organitzar-la, relacionar-la i prioritzar-la per poder plantejar-se una anàlisi coherent.
6. *Anàlisi de punts febles i forts* (**s**): Ser capaç d'analitzar els punts febles i forts de les diverses opcions que es presenten per a solucionar un problema concret.
7. *Presa de decisions* (**s**): A partir del resultat d'una anàlisi prèvia poder optar per una de les possibles opcions que es presenten, actuant de forma coherent amb la decisió presa.
8. *Introducció de millores* (**s**): Apart de resoldre un problema és molt important ser capaç d'analitzar de forma crítica la solució final i introduir millores concretes.
9. *Capacitat per generar noves idees (creativitat)* (**s**): Per resoldre un problema complex és important disposar de creativitat suficient com per a generar possibles alternatives.

2. Organització i planificació

Sistèmiques:

1. *Interès per la qualitat* (**s**): Es tracta d'una competència bàsica per poder assolir les fites plantejades en un projecte amb suficients garanties.
2. *Disseny i direcció de projectes* (**s**): S'ha de ser capaç de dissenyar i planificar les diverses tasques d'un projecte i després assumir la seva direcció per a que aquestes es duguin a terme en els plaços previstos.

3. *Planificació de canvi (s)*: Qualsevol projecte d'enginyeria ha d'estar preparat per a introduir canvis en qualsevol de les seves fases. L'alumne ha de ser capaç de planificar i avançar les possibles complicacions relacionades amb canvis en els requeriments o les solucions adoptades.
4. *Redisseny de processos (s)*: Quan un procés es mostra poc eficient o adient per a una situació concreta s'ha de poder redissenyar-lo, plantejant els canvis necessaris per a la seva millora.
5. *Capacitat d'adaptació a noves situacions (s)*: Un cop redissenyat un procés o introduït un canvi a la planificació s'ha de ser capaç d'adaptar-se a aquesta nova situació de forma eficient i en el menor temps possible.

3. Treball en equip

Interpersonals:

1. *Competència comunicativa (ip)*: En el context d'un treball en equip s'ha de ser capaç de comunicar els criteris propis i els resultats personals a la resta dels membres de forma convincent i clara utilitzant el llenguatge tècnic adient de forma que la comunicació resulti el menys ambigua possible. Aquesta competència comunicativa s'haurà d'assolir tant mitjançant comunicació verbal com escrita.
2. *Capacitat per comunicar-se amb experts d'altres àmbits (ip)*: De la mateixa manera és important ser capaç de comunicar-se amb experts d'altres àmbits o amb clients, abandonant en la mida del possible la jerga informàtica i adoptant un llenguatge planer i sintètic que sigui capaç de transmetre els punts importants del què volem comunicar.
3. *Lideratge (ip)*: Ser capaç de prendre decisions en un grup de treball, comunicar-les de forma convincent i aconseguir el suport dels altres membres del grup.

4. Altres

Instrumentals:

1. *Ús d'instruments tecnològics (i)*: Poder aplicar diferents instruments tecnològics a la pràctica i ser capaç de decidir sobre la idoneïtat de la seva utilització.
2. *Capacitat per aplicar el coneixement teòric a la pràctica (s)*: S'ha de ser capaç d'aplicar el coneixement obtingut de forma teòrica (sobre diferents metodologies o forma d'organitzar processos i activitats) a la resolució d'un problema pràctic.

Sistèmiques:

3. *Capacitat d'aprendre a aprendre (s)*: Possiblement la competència general més important és la ser capaç d'aprendre noves tecnologies o assimilar experiències de forma acumulativa.

Competències Específiques

Les competències específiques d'aquesta assignatura es poden resumir en quatre grans competències-eix:

- 1. Coneixement de Metodologies de l'Enginyeria del Software:**
- 2. Capacitat d'analitzar i descriure els requeriments d'un problema complex d'enginyeria del software**
- 3. Capacitat de dissenyar la solució d'un problema complex d'enginyeria del software a partir d'uns requeriments**

A continuació desglossarem i detallarem cadascuna d'aquestes competències-eix. Més endavant, al següent apartat, comentarem amb més detall com s'avaluaran cada competència.

1. Coneixement de diferents metodologies de l'enginyeria del software.
 - 1.a. Coneixement dels diferents tipus de "cicles de vida" del software **(i)**
 - 1.b. Capacitat de triar una metodologia concreta entre les existents segons les necessitats del cas concret **(s)**
 - 1.c. Capacitat d'adaptar la metodologia usant mesures correctores **(i)**
 - 1.d. Capacitat d'entendre com les decisions d'anàlisi, disseny i procés afecten el producte final **(s)**
2. Capacitat d'analitzar i descriure els requeriments d'un problema complex d'enginyeria del software.
 - 2.a. Capacitat d'extreure'n els requeriments: identificar "stakeholders", actors, casos d'ús, classificar requeriments segons siguin funcionals, no funcionals i de domini **(s)**
 - 2.b. Capacitat d'expressar-los de forma organitzada i tècnica (concreta, unívoca) **(s)**
 - 2.c. Domini d'algun llenguatge de modelització estàndar (per exemple l'UML) amb el qual ajudar-se per expressar els requeriments **(i)**
 - 2.d. Capacitat de treballar els requeriments en equip **(ip)**
 - 2.e. Capacitat de comunicar els requeriments al client, de forma planera, per obtenir "feedback" i validar-los **(ip)**
3. Capacitat de dissenyar la solució d'un problema complex d'enginyeria del software a partir d'uns requeriments
 - 3.a. Capacitat d'entendre uns requeriments donats **(s)**
 - 3.b. Capacitat d'organització i planificació de les diverses fases i tasques involucrades **(s)**
 - 3.c. Presa de decisions de disseny **(s)**
 - a) Identificar les forces involucrades i trobar una solució de que les balanci.
 - 3.d. Capacitat per comunicar de forma convincent i clara les decisions preses en la resolució del cas pràctic **(ip)**
 - 3.e. Capacitat de treballar el disseny en equip **(ip)**
 - 3.f. Optimització del temps: Capacitat de trobar una aproximació prou bona a la solució d'un problema complex **(s)**
 - 3.g. Coneixement de tecnologies orientades a objecte **(i)**

- a) Domini un llenguatge orientat a objectes (per exemple el C++)
 - b) Coneixement d'un llenguatge de modelatge estàndar (per exemple l'UML)
 - c) Capacitat d'aplicar patrons de disseny.
- 3.h. Capacitat d'implementar el disseny **(s)**
- a) Capacitat d'obtenir feedback del codi per modificar el disseny.
- 3.i. Capacitat de realitzar testos automàtics de la solució implementada **(s)**

1. Avaluació

- *Fase 1. Criteris d'avaluació generals*

S'avaluaran les 4 competències eix esmentades en l'apartat anterior així com les competències generals. A tal efecte es realitzaran les següents activitats avaluadores:

Obligatòries:

1. Examen final amb la resolució d'un problema pràctic en un temps limitat:
40% de la nota final
2. Pràctiques realitzades en parella amb la realització d'un projecte durant tot el trimestre i avaluació continuada: **40% de la nota final**
3. Test teòric avaluant els coneixements adquirits a realitzar abans de l'acabament del trimestre: **15% de la nota final**
4. Discussió sobre la resolució d'uns exercicis pràctics i participació a classe **5 % de la nota final**

Per poder aprovar les quatre activitats obligatòries han de tenir una nota superior al 4. Per a la convocatòria extra-ordinària es conservarà la nota de les parts amb nota superior a 4.

A continuació detallarem i concretarem aquestes activitats

- *Fase 2. Concreció*

Activitat 1. Problema d'Examen

- Com s'avaluarà:

En un examen en que es plantejarà un problema concret amb una sèrie de preguntes d'anàlisi i disseny. El problema haurà de resoldre's en un temps limitat. Es podrà disposar d'apunts i material complementari.

- Què s'avaluarà:

Competències Generals:

- Resolució de problemes

- Organització i planificació

Competències Específiques:

- Capacitat d'analitzar i descriure els requeriments d'un problema complex d'enginyeria del software.
- Capacitat de dissenyar la solució d'un problema complex d'enginyeria del software a partir d'uns requeriments

Quan s'avaluarà:

S'avaluarà a finals de trimestre per assegurar que els alumnes han assolit la soltura i els coneixements necessaris.

Activitat 2. Pràctiques

Com s'avaluarà:

Els alumnes s'enfrontaran per equips a la resolució d'un cas pràctic que simularà una situació real i hauran de seguir tot el cicle de vida des de l'anàlisi de requeriments fins a la implementació. L'avaluació es realitzarà a partir de punts de control intermitjos en què els alumnes hauran de presentar els resultats parcials i defensar les seves decisions de forma coherent. També es tindrà en compte aspectes subjectius com ara el funcionament del grup de treball, la organització...

Què s'avaluarà:

Competències Generals:

- Resolució de problemes
- Organització i planificació
- Treball en equip
- Ús d'instruments tecnològics
- Capacitat per aplicar el coneixement teòric a la pràctica

Competències Específiques:

- Capacitat d'analitzar i descriure els requeriments d'un problema complex d'enginyeria del software.
- Capacitat de dissenyar la solució d'un problema complex d'enginyeria del software a partir d'uns requeriments

Quan s'avaluarà:

S'avaluarà durant tot el trimestre a partir de punts de control i lliuraments intermitjos.

Activitat 3. Examen Test

Com s'avaluarà:

Es realitzarà un test amb preguntes enfocades a demostrar els coneixements de l'alumne en els diversos conceptes presentats al llarg de l'assignatura

Què s'avaluarà:

Competències Específiques:

- Coneixement de Metodologies de l'Enginyeria del Software

Quan s'avaluarà:

Es realitzarà a mitjans de trimestre quan ja s'hagi introduït tot el contingut teòric. D'aquesta manera se separarà de l'examen final i es distribuïran esforços més homogèniament. Si l'alumne suspèn en aquesta primera convocatòria, podrà fer-lo també juntament amb l'examen a finals de trimestre.

Activitat 4. Discussió Exercicis

Com s'avaluarà:

Es realitzaran una sèrie de seminaris amb un màxim de 10 persones on els alumnes presentaran i discutiran les possibles solucions a exercicis realitzats prèviament.

Què s'avaluarà:

En realitat aquesta no és una activitat avaluativa i no es pretén avaluar cap competència. El fet que es consideri la participació dels alumnes com a element avaluator és una simple mesura incentivadora per a la seva participació.

Quan s'avaluarà:

A mitjans de trimestre

Activitat 5. Seminaris d'Empresa

Com s'avaluarà:

Es realitzaran una sèrie de seminaris amb la participació d'empreses externes del sector que exposaran les seves metodologies. Els alumnes seran convidats a intervenir i formular preguntes. Es contarà en la part de participació a classe.

Què s'avaluarà:

No és una activitat avaluativa sinó incentivadora.

Quan s'avaluarà:

A mitjans de trimestre

2. Continguts

PART 1. L'ENGINYERIA DE SOFTWARE I ELS SEUS PROCESSOS (cap. 1-4)

Bloc de contingut 1: Cicle de Vida i Metodologies (Nota: inclou els capítols 1, 2 i 3 dels apunts)

Conceptes:

- Noció d'Enginyeria de Software
- Relació de l'Enginyeria de Software amb altres àmbits de la professió informàtica i altres professions
- El **Cicle de Vida** d'un producte software i els seus models

Procediments:

- Identificació de les diverses parts o **processos del cicle de vida genèric** i les activitats que s'hi duen a terme
- Anàlisi dels **avantatges/inconvenients dels diversos models** de cicle de vida en compte el context d'un determinat projecte

Actituds:

- Interés per l'Enginyeria de Software i la seva importància en la majoria dels àmbits de la professió informàtica.
- Esperit crític a l'hora d'avaluar els avantatges/inconvenients dels diversos models de cicle de vida i metodologies.

Bloc de contingut 2: Enginyeria de Requeriments

Conceptes:

- Definició i classificació dels diversos **tipus de Requeriments**
- Elements d'un **Document de Requeriments** i formalismes associats
- El **cicle de vida** de l'Enginyeria de Requeriments

Procediments:

- Classificació dels **requeriments** en **funcionals, no funcionals i de domini**
- Identificació de **stakeholders** i **actors**.

- Utilització del **diagrama de casos d'us** UML per analitzar i descriure els requeriments funcionals
- Redacció d'un Document de Requeriments complet a partir de l'anàlisi d'un cas pràctic.
- Manteniment del Document de Requeriments: gestió del canvi

Actituds:

- Valoració de la importància d'uns requeriments ben definits en el marc d'un projecte informàtic
- Previsió i adequació al canvi

Bloc de contingut 3: Metodologies

Conceptes:

- Descripció de diversos **exemples** de metodologies
- Metodologies **àgils** vs. **tradicionals**
- Metodologies basades en el **procés** vs. metodologies basades en les **persones**

Procediments:

- Identificació dels beneficis i inconvenients de diverses metodologies
- Aplicació de les pràctiques i processos d'algunes de les metodologies

Actituds:

- Actitud crítica per avaluar una metodologia

PART 2. ANÀLISI I DISSENY OO (cap. 5-7)

Bloc de contingut 4: Anàlisi i Disseny OO

Conceptes:

- Definició i propietats dels **Objectes**
- Definició i propietats de les **Classes**
- Les relacions entre classes: **herència**, **composició**, **agregació**, **associació** i **dependència**
- El **Polimorfisme** i les **Jerarquies d'Herència**
- El llenguatge **C++** com a exemple de llenguatge orientat a objectes.

Procediments:

- **Particionament** d'un problema complex emprant els conceptes i els procediments del paradigma orientat a objectes.
- Utilització de l'UML per a modelar sistemes Orientats a Objecte
- Utilització del llenguatge C++ per a implementar un disseny orientat a objectes.
- Desenvolupament dirigit per testos

Actituds:

- Les bones pràctiques de programació (Encapsulació, no duplicació, mètodes curts, nomenclatura entenedora...)
- Adaptació a les convencions de codificació de l'equip on s'està treballant
- Adaptació al canvi: el codi es quelcom moldejable i en constant evolució
- Comprensió dels avantatges del desenvolupament dirigit per testos

Bloc de contingut 5:L'UML**Conceptes:**

- Avantatges del modelatge visual
- Definició i breu història de l'UML
- Els principals diagrames UML i els elements inclosos a tots ells

Procediments:

- Utilitat dels diversos diagrames UML a les diverses fases del cicle de vida
- Comprensió de qualsevol tipus de diagrama UML aplicat a un problema concret
- Utilització dels diagrames UML més importants (diagrama de casos d'us, estàtic, i de seqüència) en la resolució d'un problema i en menor mesura d'algun dels altres tipus de diagrames.

Actituds:

- Acceptació de la importància d'utilitzar un formalisme estàndar a l'hora d'utilitzar diagrames.

Bloc de contingut 6:Patrons Software**Conceptes:**

- Concepte de Patró Software
- **Catàlegs** de patrons i la seva utilitat

Procediments:

- Utilització d'UML per a entendre un patró de disseny concret
- Utilització dels catàlegs de patrons per a identificar un patró com a solució genèrica a un problema concret
- Utilització de patrons a l'hora de dissenyar la solució a un problema pràctic concret

Actituds:

- **No re-inventar la roda:** la majoria de problemes amb que un es troben ja han estat solucionats i que la solució disponible en forma de patró és segurament millor que la nosaltres li podríem donar de bon començament.

3. Coneixements previs/prerrequisits

- Coneixements de programació
- Coneixements bàsics del paradigma i les tecnologies Orientades a l'Objecte
- Competències generals a esperar en un alumne en 3er curs d'enginyeria en concret capacitats de comprensió i anàlisi d'un problema complex

4. Activitat ensenyament-aprenentatge

- *Fase 1. Disseny per competències*

Competència 1. Coneixement de diferents metodologies de l'enginyeria del software.

Activitats docents	Temporització	Activitats d'aprenentatge	Temporització
Classe Magistral: Explicacions de caire general fomentant la discussió.	5 hores	Estudi personal a partir d'apunts i bibliografia recomanada.	23 hores
Seminari empresa	2 hores	Realització d'exercicis tipus test de la col·lecció.	2 hores

Competència 2. Capacitat de descriure els requeriments d'un problema complex d'enginyeria del software.

Activitats docents	Temporització	Activitats d'aprenentatge	Temporització
Classe magistral/ discussió explicant tipus de requeriments, estructura de document...	6,5 hores	Pràctiques en parella realitzant un document de requeriments i utilitzant eines CASE.	15 hores
Revisió i discussió de requeriments preparats pels propis alumnes en clàsses pràctiques 10-15 alumnes	6 hores	Estudi i realització de problemes de la col·lecció	10 hores

Competència 3. Capacitat de dissenyar la solució d'un problema complex d'enginyeria del software

Activitats docents	Temporització	Activitats d'aprenentatge	Temporització
Classe magistral: Explicació dels elements més importants del paradigma Orientat a Objectes	3,5 hores	Resolució individual de problemes de la col·lecció	27 hores
Classe magistral: Explicació del llenguatge UML	4,5 hores	Resolució pràctica en parella d'un problema complex relacionat amb els requeriments prèviament analitzats	35 hores
Resolució de problemes prèviament proposats en seminaris	1,5 hores		
Projecte a realitzar a la pràctica	14 hores		

4. Fase 2. Disseny temporal

TREBALL A L'AULA

	Dilluns 10:30-12:30	Dimecres 8:30-10:30	Dijous 8:30-10:30	Divendres 12:30-14:30	Sessió pràctiques	Sessions Teòriques
SETMANA 1: 21/09-25/09		TOTS (X)				
SETMANA 2: 28/09-02/10	TOTS (X)	G11 (D) + G12 (P)	TOTS (X)	G11 (D) + G12 (P)	A1. Sistemes de control de versions	Intro + Cicle de Vida + Metodologies (Temes 1, 2, 3)
SETMANA 3: 05/10-09/10	TOTS (X)	G11 (D) + G12 (P)	TOTS (X)	G11 (D) + G12 (P)	A2. Redacció de casos d'ús i requeriments associats	Metodologies i Requeriments (Temes 3,4)
SETMANA 4: 12/10-16/10		G11 (D) + G12 (P)	TOTS (X)	G11 (D) + G12 (P)	A3. Anàlisi: diagrama estàtic i altres	Requeriments (Tema 4)
SETMANA 5: 19/10-23/10	G11 (D) + G12 (P)	G11 (D) + G12 (P)		G11 (D) + G12 (P)	B1. C++, accessors, TDD (Es començarà la B2 el divendres)	
SETMANA 6: 26/10-30/10	G11 (D) + G12 (P)	TOTS (X)	TOTS (X)	TOTS (X) Seminari Empresa	B2. Patró Facade, test funcionals, associacions, test fallada	Dubtes + Preguntes
SETMANA 7: 02/11-06/11	TOTS (X) Test	G11 (D) + G12 (P)	TOTS (X)	G11 (D) + G12 (P)	B3. Patró Adapter. Associacions polimorfiques.	OO + UML (Temes 5 i 6)
SETMANA 8: 09/11-13/11	TOTS (X)	G11 (D) + G12 (P)	TOTS (X)	G11 (D) + G12 (P)	C1. Subscripcions i notificacions.	OO + UML (Temes 5 i 6)
SETMANA 9: 16/11-20/11	TOTS (X)	G11 (D) + G12 (P)	TOTS (X)	G11 (D) + G12 (P)	C2. Observer i Strategy. Refactorings de llarga durada.	Patrons (Tema 7)
SETMANA 10: 23/11-27/11	TOTS (X)	G11 (D) + G12 (P)	TOTS (X)	G11 (D) + G12 (P)	C3. Neteja de les falles de la correcció presencial.	Dubtes + Preguntes
SETMANA 11: 30/11-04/12	Sem 1 G11 (D) + G12 (P)	Sem 2 G11 (D) + G12 (P)	Sem 3 G11 (D) + G12 (P)	Sem 4 G11 (D) + G12 (P)		

Festa

Presentacions on Empreses externes expliquen les seves metodologies

Seminaris de 90 minuts amb grups de 10 persones en els que els alumnes han de mostrar problemes resolts i discutir la seva solució

Sessió pràctiques a les aules d'ordinadors amb grups de 10-15 estudiants – Sessió on els dos grups tenen pràctica

Sessió d'avaluació

Sessió magistral on es fomenta la participació dels alumnes

Nota: Segons Bolonya l'Estudiant ha de dedicar 25 hores per crèdit. Enginyeria de Software I té 9 crèdits i això dona un total de 225 hores o 22,5 hores per setmana. D'aquestes 7 hores es realitzaran presencialment a classe i la resta corren a càrrec de l'alumne.

Sessió pràctiques	Sessions Teòriques	Teoria	Exercicis			Pràctica				
		Lectura i Estudi Tema 1. Introducció a l'Enginyeria de Software, lectura Tema 2. El Cicle de Vida	8hores				Lectura de la Pràctica	2hores	SETMANA 1: 21/09-25/09	10
A1. Sistemes de control de versions	Intro + Cicle de Vida + Metodologies (Temes 1, 2, 3)	Estudi Tema 2. El cicle de Vida, lectura Tema 3. Metodologies	4hores	Realitzar preguntes tipus test del tema Cicle de Vida	1hores		Traballar Sessió A1.	5hores	SETMANA 2: 28/09-30/10	10
A2. Redacció de casos d'ús	Metodologies i Requiriments (Temes 3,4)	Estudi Tema 3. Metodologies i Requiriments i UML Diagrames Casos d'Us	6hores				Ulurar Sessió A1. Traballar Sessió A2.	4hores	SETMANA 3: 05/10-09/10	10
A3. Anàlisi de diagrama estàtic i B1. C++ accessors, TDD iES	Requiriments (Tema 4)		3hores	Realitzar exercicis del tema de Requiriments	4hores		Ulurar Sessió A2. Traballar Sessió A3.	4hores	SETMANA 4: 12/10-16/10	11
B2. Patró Facada, test funcional,	Dubtes + Preguntes	Repàs Temes 1-4	3hores	Realitzar preguntes tipus test pendents	3hores		Ulurar Sessió A3. Traballar Sessió B1.	4hores	SETMANA 5: 19/10-23/10	10
B3. Patró Adapter	Estudi per a test	Estudi per a test	4hores	Repassar exercicis tipus test	4hores		Ulurar Sessió B1. Traballar Sessió B2.	3hores	SETMANA 6: 26/10-30/10	11
C1. Subscripció	OO + UML (Temes 5 i 6)	OO + UML (Temes 5 i 6)	3hores	Realitzar exercicis del tema de Disseny	3hores		Ulurar Sessió B2. Traballar Sessió B3.	4hores	SETMANA 7: 02/11-05/11	10
C2. Observer i Strategy. Refactorings de llarga durada.	OO + UML (Temes 5 i 6)	OO + UML (Temes 5 i 6)	4hores	Realització dels exercicis pendents	3hores		Ulurar Sessió B3. Traballar Sessió C1.	4hores	SETMANA 8: 09/11-13/11	11
C3. Netjea de les dades de la connexió presencial.	Patrons (Tema 7)	Estudi Tema 7. Patrons Software	3hores	Realització dels exercicis pendents	3hores		Ulurar Sessió C1. Traballar Sessió C2.	5hores	SETMANA 9: 16/11-20/11	11
	Dubtes + Preguntes						Ulurar Sessió C2. Entrega presencial. Traballar Sessió C3.	4hores	SETMANA 10: 23/11-27/11	11
		Repàs dels 5, 6 i 7	3hores	Realització dels exercicis pendents	4hores			4hores	SETMANA 11: 30/11-04/12	12
		Repàs	3hores		3hores		Ulurar Sessió C3.	4hores		

6. Recursos didàctics

- Apunts en format llibre entregats a principi de trimestre
- Col·lecció de Problemes
- Web de l'assignatura: <http://xavier.amatriain.net/es1>
- Bibliografia recomanada:

Bibliografia recomanada

Fent un clic damunt del títol de cada document, enllaçaràs amb el seu registre bibliogràfic en el catàleg de la Biblioteca. D'aquesta manera et serà fàcil localitzar el document i/o reservar-lo en cas que tots els exemplars disponibles estiguin en préstec.

Fowler, Martin

[UML distilled : a brief guide to the standard object modeling language / Martin Fowler with Kendall Scott](#)

Reading (Mass.) : Addison-Wesley, cop. 2000

Meyer, Bertrand

[Construcción de software orientado a objetos / Bertrand Meyer ; traducción: Miguel Katrib Mora, Rafael García Bermejo, Salvador Sánchez ; revisión técnica: Jesús García](#)

Madrid [etc.] : Prentice Hall, 1998

Sommerville, Ian

[Software engineering / Ian Sommerville](#)

Harlow : Addison-Wesley, 2001

Bibliografia complementària

Costal, Dolors

[Enginyeria del software especificació : especificació de sistemes orientats a objectes amb la notació UML / Dolors Costal, M. Ribera Sancho, Ernest Teniente](#)

Barcelona : Edicions UPC, 2000

Eliëns, Anton

[Principles of object-oriented software development / Anton Eliëns](#)

Workingham : Addison-Wesley, 1995

Joyanes Aguilar, Luis

[Programación orientada a objetos / Luis Joyanes Aguilar](#)

Madrid : Osborne/McGraw-Hill, cop. 1998

Larman, Craig

[Applying UML and patterns : an introduction to object-oriented analysis and design / Craig Larman](#)

Upper Saddle River, N.J. : Prentice Hall PTR, cop. 1998

Peralta Giménez, Allen J.

[Enginyeria del software : programació orientada a objectes / Allen J. Peralta Giménez, Horacio Rodríguez Hontoria](#)

Barcelona : Edicions UPC, 1994

Pressman, Roger S.

[Ingeniería del software un enfoque práctico / Roger S. Pressman adaptación: Darrel Ince traducción: Rafael Ojeda Martín ... \[et al.\] dirección, coordinación y revisión](#)

Madrid McGraw-Hill cop. 2002

Schach, Stephen R.

[Software engineering / Stephen R. Schach](#)

Boston : Irwin, cop. 1993

Shtern, Victor

[Core C++ : a software engineering approach / Victor Shtern](#)

Upper Saddle River : Prentice Hall PTR, cop. 2000

Stroustrup, Bjarne

[The C++ programming language / Bjarne Stroustrup](#)

Reading (Mass.) : Addison-Wesley, cop. 1997