

Disseny de Software

Codi: 102759
Crèdits: 6

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	3	1
2502441 Enginyeria Informàtica	OT	4	1

La metodologia docent i l'avaluació proposades a la guia poden experimentar alguna modificació en funció de les restriccions a la presencialitat que imposin les autoritats sanitàries.

Professor/a de contacte

Nom: Joan Serrat Gual
Correu electrònic: Joan.Serrat@uab.cat

Utilització d'idiomes a l'assignatura

Llengua vehicular majoritària: català (cat)
Grup íntegre en anglès: No
Grup íntegre en català: Sí
Grup íntegre en espanyol: No

Altres indicacions sobre les llengües

Tots els materials de la assignatura (transparències, enunciats) són en anglès.

Equip docent

David Fernández Mota
Helena Bolta Torrell

Equip docent extern a la UAB

Rodolfo Guichón

Prerequisits

No hi ha prerequisits oficials, però cal que l'estudiant domini la notació de diagrames de classe UML, els conceptes d'orientació a objecte i la seva programació en algun llenguatge, preferiblement Java.

Objectius

Un cop el propòsit d'un software s'ha determinat mitjançant l'anàlisi de requeriments, els desenvolupadors de software dissenyen una solució. Aquesta va des de l'arquitectura dels seus components (classes, mòduls...) fins a la implementació en algun llenguatge de programació. Aquesta assignatura cobreix principalment dos aspectes claus del disseny: els patrons de disseny orientat a objecte i el disseny de la interfície d'usuari. A més d'aquests, tracta altres temes com són el disseny per contracte de classes i mètodes i l'estil de codificació.

El mètode d'aprenentatge és basat en un projecte (la pràctica) o dit d'una altra manera, 'aprendre per fer': tots els conceptes tractats a teoria són necessaris per tal de dissenyar i implementar una aplicació de software, a partir d'un enunciat que fa el paper de document de requeriments. Es tracta de fer el disseny i la implementació d'una aplicació (però no l'anàlisi de requeriments, la prova, el control de versions o el garantiment de la qualitat, que pertanyen a altres assignatures de la menció).

Competències

Enginyeria Informàtica

- Capacitat d'identificar i analitzar problemes i dissenyar, desenvolupar, implementar, verificar i documentar solucions de software sobre la base d'un coneixement adequat de les teories, models i tècniques actuals.
- Capacitat per concebre sistemes, aplicacions i serveis basats en tecnologies de xarxa, incloent Internet, Web, comerç electrònic, multimèdia, serveis interactius i computació mòbil.
- Capacitat per concebre, desenvolupar i mantenir sistemes, serveis i aplicacions informàtiques emprant els mètodes de l'enginyeria del software com a instrument per a assegurar-ne la qualitat.
- Capacitat per dissenyar, desenvolupar, avaluar i assegurar l'accessibilitat, l'ergonomia, la usabilitat i la seguretat dels sistemes, serveis i aplicacions informàtiques, així com de la informació que gestionen.
- Tenir una actitud personal adequada.
- Treballar en equip.

Resultats d'aprenentatge

1. Assumir i respectar el rol dels diversos membres de l'equip, així com els diferents nivells de dependència de l'equip.
2. Concebre sistemes de software basats en tecnologies de xarxa, incloent Internet, Web, comerç electrònic, multimèdia, serveis interactius i computació mòbil.
3. Conèixer i aplicar els components de software específics en problemes computacionals basats en comunicacions.
4. Desenvolupar la curiositat i la creativitat.
5. Dissenyar i documentar solucions de software.
6. Dissenyar una interface d'usuari partint de les especificacions del client.
7. Especificar les necessitats del client en un document d'especificació de software.
8. Identificar el paradigma que millor s'adapta al problema de disseny d'un sistema informàtic software específic tenint en compte els requeriments d'accessibilitat, ergonomia, usabilitat i seguretat.
9. Identificar, gestionar i resoldre conflictes.
10. Mantenir una actitud proactiva i dinàmica respecte al desenvolupament de la pròpia carrera professional, el creixement personal i la formació continuada. Tenir esperit de superació.
11. Treballar cooperativament.

Continguts

1. Principis d'orientació a objecte
2. Patrons de disseny : creacionals, estructurals, de comportament
3. Disseny per contracte
4. Estil de codificació i comentaris. Logging
5. Disseny de la interfície d'usuari
 - usabilitat
 - UX research
 - UX design : prototipat, user testing
 - internacionalització i localització
6. Programació d'interfícies en Flutter

Metodologia

Tal com hem dit als objectius, l'assignatura segueix una metodologia d'aprenentatge basat en el desenvolupament d'un projecte. La seva implementació concreta pel que fa a l'ús de les hores de classe és la següent:

Classes de teoria. En elles el professor introdueix els continguts teòrics i dona referències útils (llibres, articles, pàgines web) per que l'estudiant després en pugui continuar el seu aprenentatge, així com desenvolupar l'aplicació software proposada. Les transparències utilitzades en aquestes classes doncs no s'han de prendre com a font d'estudi, sinó que serveixen per que el professor pugui exposar de manera més àgil. En canvi, cal estudiar els llibres recomanats, i els articles, capítols de llibres d'accés obert i altre material que indicarem.

Classes de problemes. Aquestes classes s'empreran principalment per exposar exercicis, que són problemes de disseny més implementació, o be la aplicació de tècniques de UX. Depenent del calendari del curs, també es podran fer classes de problemes de la part de patrons.

Sessions de pràctiques. Els estudiants hauran format a principi de curs grups petits. A les classes de pràctiques els estudiants venen a 1) 'passar comptes' amb el professor de la feina feta cada setmana i resoldre dubtes, i 2) donat que la pràctica estarà dividida en fites, les setmanes que corresponen a una fita s'avalua la part corresponent in situ, segons un barem que figurarà a l'enunciat de la pràctica. Per tant, les sessions de pràctiques *no* són per fer la pràctica.

Tots els materials de l'assignatura juntament amb la planificació detallada es troben al campus virtual.

Competències transversals:

Competència transversal	Activitats on es treballaran	Com s'avaluarà
T03.01, T03.02, T03.03	preparació i classes de pràctiques	a les avaluacions de pràctiques (fites)
T06.01, T06.02	preparació de pràctiques i resolució de problemes	a les avaluacions de pràctiques (fites) i als exàmens de teoria

Nota: es reservaran 15 minuts d'una classe, dins del calendari establert pel centre/titulació, per a la complementació per part de l'alumnat de les enquestes d'avaluació de l'actuació del professorat i d'avaluació de l'assignatura/mòdul.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
classes de problemes	12	0,48	2, 3, 5, 6
classes de pràctiques	12	0,48	1, 2, 3, 5, 6, 9, 11
classes de teoria	26	1,04	2, 3, 5, 6, 8
Tipus: Autònomes			
estudi individual	29	1,16	3, 5, 6, 7, 8
preparacio practiques	52	2,08	1, 2, 3, 5, 6, 8, 9, 11
resolució de problemes	12	0,48	2, 3, 5, 6, 7, 8

Avaluació

La forma d'avaluació és continuada. Aquesta consta d'una part d'avaluació de grup i una altra d'individual. La nota de la primera (PR) s'obté fent promig de les notes de les fites de pràctiques (M1, M2, M3). La segona (EX) del promig de dues proves escrites durant el trimestre (EX1, EX2), o bé d'un examen final de recuperació (EXR1, EXR2). Els sufixes 1 i 2 es refereixen a les 2 parts que té l'assignatura, abans i després del primer examen EX1.

L'algoritme pel càlcul de la nota final (FM) és el següent (totes les notes són sobre 10) :

```
// EX1, EX2 examens parcials
// EXR1, EXR2 examen de recuperació
// M1, M2, M3 notes de les 3 fites de pràctiques
// FM nota final
```

$EX = (\max(EX1, EXR1) + \max(EX2, EXR2)) / 2.$

```
if (M1>0) and (M2>0) and (M3>0) {
  PR = (M1 + M2 + M3) / 3.
} else {
  PR = 0.
}
```

```
if (EX >=5.) and (PR >=5.) {
  FM =(EX + PR) / 2.
} else { // EX<5. o PR<5.
  FM = min(EX, PR)
}
```

La pràctica

La realització de la pràctica és essencial per assimilar els conceptes de teoria. La primera cosa doncs que volem assegurar és que la pràctica ha estat realment feta pel grup que la presenta, i que cada membre del grup hi ha contribuït en la mateixa mesura que els altres. La segona és que la pràctica es faci de manera progressiva durant el curs, en paral·lel a la teoria. Per això farem el següent:

- La nota de la pràctica s'obté combinant la nota de les parts en les que aquesta està dividida. Cada part serà avaluada in situ en una de les sessions de pràctiques que anunciarem a principi de curs (les fites). Els punts que es poden obtenir de cada part s'especificarà a l'enunciat. Aquestes avaluacions no tenen recuperació després.

- Els dies d'avaluació de la pràctica (fites) farem una breu prova escrita de validació. Si de resultes de les respostes a aquesta prova o a les preguntes que faci el professor durant l'avaluació, tenim dubtes sobre l'autoria o la contribució dels membres del grup, llavors la nota de pràctica passarà de ser de grup a individual i en funció de quanta i quina part de la pràctica hagi fet cada membre.

Els exercicis

Proposarem alguns exercicis i petits problemes que són opcionals. Aquests exercicis es poden fer individualment o en petit grup, el mateix o diferent que el de pràctiques. Els exercicis poden sumar punts a la nota de teoria EX, amb un topall de 10. Diem 'podem' perquè la obtenció d'aquests punts extra estarà condicionada a una prova de validació per tal de comprobar que l'estudiant ha realitzat efectivament l'exercici que ha lliurat. Aquestes proves de validació les farem només un cop, als examens EX1 i EX2, i no tenen recuperació després.

Altres

No convalidem pràctiques, exercicis ni exàmens de cursos anteriors. Els estudiants repetidors no reben cap tractament diferenciat.

Si no s'entrega un examen o una part de la pràctica, la nota d'aquesta part és zero. La nota final serà 'no avaluable' si no s'ha entregat res avaluable, ja sigui examen o pràctica.

Les fites de pràctiques són avaluacions equivalents al examens escrits. Si un estudiant no s'hi presenta, la nota és zero.

Les notes MH les concedirem de manera discrecional als estudiants amb nota final superior 9.0 i tenint en compte tota mena de treballs fets (exàmens, pràctica i exercicis).

Les dates d'avaluació continuada i lliurament de treballs es publicaran al campus virtual de l'assignatura i poden estar subjectes a canvis en resposta a possibles incidències. Sempre s'informarà mitjançant el campus virtual, que serà el mecanisme de comunicació entre docents i estudiants.

Per a les activitats d'avaluació de tipus examen escrit, s'indicarà un lloc, data i hora de revisió en la que l'estudiant podrà revisar l'activitat amb el professor. Si l'estudiant no es presenta a aquesta revisió, no es revisarà posteriorment aquesta activitat.

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, es qualificaran amb un zero les irregularitats comeses per l'estudiant que puguin conduir a una variació de la qualificació d'un acte d'avaluació. Per tant, la còpia, el plagi, l'engany, deixar copiar, etc. en qualsevol de les activitats d'avaluació implicarà suspendre-la amb un zero. Les activitats d'avaluació qualificades d'aquesta forma i per aquest procediment no seran recuperables. Si és necessari superar qualsevol d'aquestes activitats d'avaluació per aprovar l'assignatura, aquesta assignatura quedarà suspesa directament, sense oportunitat de recuperar-la en el mateix curs, amb nota zero.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
avaluació de grup	50%	3	0,12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
avaluació individual	50%	4	0,16	2, 3, 4, 5, 6, 7, 8, 10

Bibliografia

Part 1

- Design patterns: elements of reusable object oriented software. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Addison Wesley, 1994. Versió en castellà de Pearson Educación, 2003.
- Design patterns explained simply. Alexander Shvets. <https://sourcemaking.com/design-patterns-ebook>.
- Head first design patterns. E. Freeman, E. Freeman, K. Sierra, B. Bates. O'Reilly, 2004. Ha aparegut una segona edició el 2020.
- Clean code: a handbook of agile software craftsmanship. R.C. Martin. Prentice Hall, 2008. O be l'edició en castellà Codigo limpio, d'Anaya Multimedia 2012.

Part 2

- The user experience team of one. A research and design survival guide. Leah Buley. Rosenfeld, 2013.
- Understanding your users. A practical guide to user research methods. Kathy Baxter, Catherine Courage, Kelly Caine. Morgan Kaufmann, segona edició 2015.
- Killer UX design. Jodie Moule. Sitepoint, 2012.
- Validating product ideas. Tomer Sharon. Rosenfeld, 2016.
- Rocket surgery made easy: the do-it-yourself guide to finding and fixing usability problems. S. Krug. New Riders, 2010. Versió en castellà Haz facil lo imposible, d'Anaya Multimedia 2010.

Programari

Llenguatges de programació: Java, Dart (Flutter)

Entorns de programació: IntelliJ IDEA (Java, Flutter), plugins Checkstyle, plantUML. Llibreria logback

Altres :

- eines de prototipat d'aplicacions per mòbil com marvelApp, inVision, Adobe XD etc.
- OptimalWorkshop per a realitzar sessions d'user-testing remot, així com altres tècniques de UX