VNIVERSITAT DE VALÈNCIA

Guia Docent
34840 Enginyeria del programari

FITXA IDENTIFICATIVA

Dades de l'Assignatura

Codi	34840
Nom	Enginyeria del programari
Cicle	Grau
Crèdits ECTS	6.0
Curs acadèmic	2022 - 2023

Titulació/titulacions

Titulació	Centre	Curs	Període
1407 - Grau d'Enginyeria Multimèdia	Escola Tècnica Superior d'Enginyeria	3	Primer quadrimestre

Matèries

Titulació	Matèria	Caràcter
1407 - Grau d'Enginyeria Multimèdia	5 - Desenvolupament del software multimèdia	Obligatòria

Coordinació

Nom	Departament
MARTINEZ PLUME, JAVIER	240 - Informàtica

RESUM

L'assignatura “Enginyeria del Programari” és una assignatura obligatòria que forma part del Grau en Enginyeria Multimedia. Té assignada una dedicació de 6 ECTS que s’imparteixen durant el 1er quadrimestre del 3er curs.

En esta assignatura, es tracta d'aprendre a desenrotllar projectes programari seguint un procés sistemàtic i recolzant-se en ferramentes que permeten millorar la qualitat del programari en entorns de producció.

S'introduirà a l'alumne en el coneixement i maneig de diferents metodologies de desenrotllament de sistemes d'informació.

Es tractarà d'aconseguir un coneixement suficient del procés de programari, de manera que l'alumne siga capaç de, usant com a mètode el Procés Unificat de Desenrotllament, capturar els requisits, analitzar, dissenyar, implementar, provar i implantar projectes programari de manera concreta i amb precisió.

Pel que fa a la part pràctica, en esta assignatura tractarem que l'alumne utilitzant el llenguatge de modelatge UML i,si escau, el llenguatge de programació Java, siga capaç de posar en pràctica els coneixements vistos en la part teòrica.

L'objectiu principal de l'assignatura és introduir l'alumne en el desenrotllament de projectes programari des de l'anàlisi de requisits fins a implantació i verificació del producte per part del client, de manera que siga capaç de:

- Conèixer l'origen i significat del terme “Enginyeria del Programari”, la seua evolució històrica i els desafiaments actuals (amb atenció al context sociocultural del seu desenrotllament), i ser conscient de la responsabilitat ètica i professional d'un Enginyer de Programari.
- Prendre consciència de la importància de realitzar sempre una anàlisi i disseny previs del problema, com a passos anteriors a la implementació en un llenguatge de programació.
- Ser conscient de la necessitat del modelatge i l'abstracció en el desenrotllament de programari.
- Conèixer el concepte de mètode de desenrotllament de programari i les seues principals classificacions.
- Distingir els conceptes de diagrama i model.
- Conèixer els principals diagrames UML: casos d'ús, classes, paquets, objectes, interacció (seqüència i comunicació), estats i activitats, i ser capaç d'aplicar-los al modelatge d'un projecte de grandària mitjana.
- Donada una aplicació de grandària mitjana, ser capaç d'abordar l'anàlisi de requisits centrat en casos d'ús, el modelatge del domini o conceptual, l'anàlisi de col·laboracions entre objectes amb una apropiada assignació de responsabilitats i tenint en compte detalls tecnològics.
    - Conèixer tècniques de disseny i aplicar-les en el marc d'un procés iteratiu.
    - Triar l'opció de disseny conceptual de dades més adequada entre diverses alternatives possibles, justificant i argumentant la decisió presa.
    - Conèixer i aplicar els patrons de disseny bàsics per a construcció de programari i valorar el seu paper com a forma de reutilització de l'experiència.

Utilitzar una ferramenta programari que permeta la creació dels diferents diagrames UML.

CONEIXEMENTS PREVIS

Relació amb altres assignatures de la mateixa titulació

No heu especificat les restriccions de matrícula amb altres assignatures del pla d'estudis.

Altres tipus de requisits

Sense haver-hi requisits previs de matrícula, es recomana haver cursat les següents matèries/assignatures:

Programació

COMPETÈNCIES

1405 - Grau d'Enginyeria Multimèdia

- B4 - Coneixements bàsics sobre l'ús i programació dels ordinadors, sistemes operatius, bases de dades i programes informàtics amb aplicació en enginyeria.
- B5- Coneixement de l'estructura, organització, funcionament i interconnexió dels sistemes informàtics, els fonaments de la seua programació, i la seua aplicació per a la resolució de problemes propis de l'enginyeria.
- I9- Coneixement i aplicació dels principis, metodologies i cicles de vida de l'enginyeria de programari.
- MM3 - Aplicar de forma adequada les metodologies, tecnologies, procediments i ferramentes en el desenrotllament professional dels productes multimèdia en un context d'ús real, aplicant les solucions adequades en cada entorn.
- MM5 - Saber aplicar els recursos teòrics i pràctics per a abordar en la seua globalitat una aplicació multimèdia.
- MM21 - Comunicar de forma efectiva, tant per escrit com oralment, coneixements, procediments, resultats i idees relacionades amb les TIC i, concretament de la Multimèdia, coneixent el seu impacte socioeconòmic.
- MM23 - Usar de forma apropiada teories, procediments i ferramentes en el desenrotllament professional de l'Enginyeria Multimèdia en un context real (especificació, disseny, implementació, desplegament i avaluació de solucions de sistemes multimèdia) .
- MM24 - Capacitat per a dissenyar, desenrotllar, avaluar i assegurar l'accessibilitat, ergonomia, usabilidad i seguretat dels sistemes, servicis i aplicacions multimèdia, així com de la informació que gestionen.
- MM26 - Capacitat per a concebre, desenrotllar i mantindre sistemes, servicis i aplicacions multimèdia emprant els mètodes de l'enginyeria del programari com a instrument per a l'assegurament de la seua qualitat, d'acord amb els coneixements adquirits segons les competències específiques establides.
- MM28 - Capacitat per a resoldre problemes amb iniciativa, presa de decisions, autonomia i creativitat. Capacitat per a saber comunicar i transmetre els coneixements, habilitats i destreses de la professió d'Enginyer Multimèdia.

RESULTATS DE L'APRENENTATGE

Esta assignatura permet obtindre els següents resultats d'aprenentatge:

- Aplicar metodologies per al desenrotllament, implantació i manteniment de sistemes d'informació.
- Planificar i executar correctament processos de desenrotllament de programari iteratius.
- Saber aplicar patrons de disseny programari en cada situació en funció de les necessitats del projecte de desenrotllament programari.
- Definir proves de validació i verificació de requisits.
- Obtindre requisits d'usuari i del programari.
- Conèixer els conceptes bàsics, els processos de desenvolupament, els mètodes i les eines de

l'enginyeria de programari
- Conèixer i comprendre els paradigmes actuals de l'enginyeria del programari dirigits al desenvolupament distribuït de programari, l'enginyeria del programari lliure i l'enginyeria Web.
- Conèixer els diferents models d'arquitectura, les tecnologies que poden integrar-se i les solucions comercials per formar una solució distribuïda particular.

Com a complement als resultats anteriors, esta assignatura també permet adquirir les següents destreses i habilitats socials:

- Analitzar un problema de desenrotllament de programari i deduir la seua naturalesa de manera concreta i amb precisió.

Dissenyar una estructura de mòduls, utilitzant patrons de disseny, per a solucionar un problema i avaluar altres alternatives.

DESCRIPCIÓ DE CONTINGUTS

1. Introducció al procés de Desenvolupament de programari UML

Destreses a adquirir:

Comprendre què és l'Enginyeria del Programari i la seua necessitat

Conéixer i comprendre els conceptes fonamentals que conformen la terminologia bàsica de l'enginyeria del programari

Comprendre les relacions entre els conceptes de procés programari, cicle de vida del programari i metodologia programari

Conéixer les característiques i explicar els avantatges i desavantatges de diferents models de procés del programari

Conéixer els principals tipus de metodologies programari

Conéixer les característiques bàsiques del Procés Unificat de desenrotllament del Programari

Entendre en què consistix el modelatge de programari i quins beneficis aporta

Reconéixer UML com a llenguatge estàndard en la construcció de programari

Continguts:

1.1 Visió general de l'Enginyeria del Programari
1.2 Conceptes bàsics de l'Enginyeria del Programari
1.3 Models i Processos
1.3.1 Introducció al Procés del Programari
1.3.2 Models de Procés del Programari
1.3.3 Metodologies de desenvolupament del Programari
1.4 El Llenguatge Unificat de Modelatge UML 2.0
1.4.1 Modelatge Estructural i Dinàmic
1.4.2 Vistes UML
1.5 Procés Unificat de Desenvolupament Software OO
1.5.1 Característiques
1.5.2 Fases
1.5.3 Activitats i Artefactes

2. Fase de Planificació i Especificació

Destreses a adquirir:

Comprendre la importància d'obtindre i gestionar els requisits i la seua influència en l'èxit d'un projecte

Entendre què són els requisits i la complexitat de l'extracció de requisits

Conèixer les activitats de requisits

Conèixer els distints tipus de requisits i ser capaç de diferenciar-los

Conèixer les diferents tècniques existents per a capturar els requisits d'un sistema

Conèixer en què consisteix el Document de Requisits

Conèixer l'estàndard IEEE/ANSI 830-199 per a SRS

Elaborar un document SRS per a sistemes de grandària mitjana

Conèixer els diversos elements i diagrames que UML proporcionar per a representar Casos d'Ús

Representar Requisits Funcionals per mitjà de Casos d'Ús

Realitzar especificacions detallades de Casos d'Ús

Continguts:

2.1. requisits

2.1.1 Definició i característiques dels Requisits

2.1.2 Tipus de Requisits

2.1.3 Activitats de Requisits

2.1.4 Tècniques d'obtenció de Requisits

2.1.5 Especificació de Requisits del Programari

2.1.6 Exercicis sobre requisits

2.2 Prototip

2.3 Casos d'Ús

2.3.1 Introducció.

2.3.2 Elements: Actors, Subjectes, Esdeveniments i Escenaris

2.3.3 Especificació de Casos d'Ús

2.3.4 Relacions: realització, generalització, extensió, inclusió

2.3.5 Diagrames de Casos d'Ús

2.3.6 Errors típics i Recomanacions

2.3.7 Exercicis sobre casos d'ús.

3. Fase d'anàlisi

Destreses a adquirir:

Conèixer els passos que s'ha de seguir per a completar l'etapa d'anàlisi en el primer cicle de desenrotllament i els artefactes a crear

Ser capaços d'elaborar el Diccionari de Dades

Ser capaços d'abstraure els Conceptes rellevants per a elaborar un Model Conceptual

Representar per mitjà de Diagrames de Classes el Model Conceptual d'un sistema

Identificar els esdeveniments del sistema en les descripcions dels Casos d'Ús per a determinar les Operacions del Sistema

Elaborar els Diagrames de Seqüència del Sistema dels diferents Casos d'Ús a partir de les Operacions del Sistema

Desenrotllar els Contractes de les Operacions del Sistema

Representar per mitjà de Diagrames d'Estat i Activitat comportaments complexos

Continguts:

3.1 Model Conceptual
3.2 Diagrama de classes
3.2.1 Classificadors: Tipus i propietats
3.2.2 Relacions: dependència, generalització, associació, realització
3.3 Diagrama d'objectes
3.3.1. instàncies
3.4 Exercicis de diagrames de classes i objectes
3.5 Diagrames de Seqüència del Sistema
3.5.1 Interaccions
3.5.2 Línies de vida
3.5.3 Missatges
3.6 Contractes
3.7 Exercicis de diagrames de seqüència i contractes
3.8 Comportament complex i Diagrames d'Estat / Activitat
3.8.1 Elements d'Activitat
3.8.2 Diagrames de Activitat
3.8.3 Elements de Màquina d'Estats
3.8.4 Diagrames d'Estat
3.9 Exercicis de diagrames d'Activitat i Estat

4. Fase de disseny

Destreses a adquirir:

Conèixer els passos que s'ha de seguir per a completar l'etapa de disseny en el primer cicle de desenrotllament i els artefactes a crear

Conèixer el concepte de responsabilitat

Conèixer i saber com aplicar una sèrie de patrons a l'hora de decidir l'assignació de responsabilitats a classes

Ser capaços d'elaborar els Diagrames d'Interacció de cada una de les operacions del sistema a partir dels seus Contractes

Elaborar el Diagrama de Classes de Disseny a partir del Model Conceptual

Continguts:

4.1 Model de Disseny
4.1.1 Responsabilitats
4.2 Diagrama de Classes de Disseny
4.3 Diagrames d'Interacció
4.3.1 Diagrama de seqüència d'operació
4.3.2 Diagrama de comunicació
4.4 Exercicis sobre diagrames d'interacció
4.5 Patrons per a l'assignació de responsabilitats.
4.5.1 Patrons GRASP
4.5.2 Patrons GOF

4.6 Exemple: Disseny del TPV

5. Arquitectura del sistema

Destreses a adquirir:

Conèixer els conceptes de capes, paquets i particions i com utilitzar-los en l'organització de l'arquitectura del sistema

Representar paquets i les seues relacions en Diagrames de Paquets

Decidir l'arquitectura a emprar i representar-la per mitjà de Diagrames de Paquets

Conéixer i saber aplicar altres patrons

Continguts:

5.1 Arquitectura multicapa i UML

5.1.1 Capes i particions

5.1.2 Paquets

5.1.3 Diagrama de paquets

5.2 Patrons de connexió entre paquets

5.2.1 Patrons GRASP

5.2.2 Patrons GOF

6. Fase d'Implementació

Destreses a adquirir:

Conéixer les decisions prèvies a prendre abans d'implementar

Conéixer els tipus de transformació espai del model-espai del codi

Ser capaços de transformar els artefactes del disseny en codi

Ser capaços de determinar la necessitat modificar els models per a introduir optimitzacions

Continguts:

6.1 Decisions prèvies

6.2 Tipus de transformació

6.2.1 Transformacions del model

6.2.2 Transformacions del codi

6.2.3 Transformacions del model al codi: Enginyeria directa

6.2.4 Transformacions del codi al model: Enginyeria inversa

6.3 Enginyeria directa

6.3.1 Mapeo de Classes

6.3.2 Mapeo de Relacions

6.3.3 Mapeo d'Herència

6.3.4 Creació de Mètodes

6.3.5 Mapeo de Contractes

7. Proves

Destreses a adquirir:

Conèixer i diferenciar els aspectes fonamentals relacionats amb les proves de programari

Entendre les proves com a part essencial del desenrotllament d'un sistema programari

Saber diferenciar els distints nivells de prova en funció de l'objectiu de la mateixa

Conèixer les diferents tècniques de prova del programari

Continguts:

7.1 Conceptes Bàsics: Errors, Defectes, Fallades, Casos de Prova

7.2 Verificació i Validació

7.2.1 Inspeccions del programari

7.2.2 Proves del programari

7.2.3 Depuració

7.3 Nivells de Prova del Programari

7.3.1 Proves Unitàries

7.3.2 Proves d'Integració

7.3.3 Proves d'Acceptació

7.3.4 Proves de Sistema

7.4 Tècniques de Prova del Programari

7.4.1 Proves de Caixa Negra

7.4.2 Proves de Caixa Blanca

7.5 Pla de Proves

VOLUM DE TREBALL

ACTIVITAT	Hores	% Presencial
Classes de teoria	30,00	100
Pràctiques en laboratori	20,00	100
Pràctiques en aula	10,00	100
Elaboració de treballs en grup	4,00	0
Estudi i treball autònom	4,00	0
Preparació de classes de teoria	23,00	0
Preparació de classes pràctiques i de problemes	50,00	0
Resolució de casos pràctics	9,00	0
TOTAL	**150,00**	

METODOLOGIA DOCENT

Les activitats formatives es desenrotllaran d'acord amb la distribució següent:

- **Activitats teòriques**.

En les classes teòriques es desenrotllaran els temes proporcionant una visió global i integradora, analitzant amb major detall els aspectes clau i de major complexitat, fomentant, en tot moment, la participació de l'estudiant.

- **Activitats pràctiques**.

Complementen les activitats teòriques amb l'objectiu d'aplicar els conceptes bàsics i ampliar-los amb el coneixement i l'experiència que vagen adquirint durant la realització dels treballs proposats. Comprenen els següents tipus d'activitats:

- Classes de problemes i qüestions en aula
- Sessions de discussió i resolució de problemes i exercicis prèviament treballats pels estudiants
- Pràctiques i seminaris en aula informàtica
- Treballs en grup per a planificació i desenrotllament de projectes programari i generació de dinàmiques de grup.
- Tutories programades (individualitzades)

Per a l'execució d'estes activitats, el grup teòric se subdividirà en subgrups més xicotet (20 alumnes com a màxim) d'acord amb la necessitat.

- **Treball personal de l'estudiant**.

Preparació de classes i exàmens (estudi). Esta tasca es realitzarà de manera individual i intenta potenciar el treball autònom.

- **Treball en xicotets grups**.

Realització, per part de xicotets grups d'estudiants (3-4) de treballs, qüestions, problemes fora de l'aula. Esta tasca complementa el treball individual i les activitats pràctiques i fomenta la capacitat d'integració en grups de treball. Comprén la realització de les activitats següents:

- Treball en grup d'investigació i recopilació d'informació sobre conceptes bàsics de l'Enginyeria del Programari, cicles de vida del programari, metodologies àgils, models de procés del programari, història d'UML.
- Presentació del treball en grup.
- Treballs en grup per a planificació i desenrotllament de projectes programari i generació de dinàmiques de grup.
- Desenrotllament de projectes de programari, la documentació del qual haurà de presentar-se per escrit.
- Presentació del projecte programari.
- Tutories programades (en grups).

S'utilitzarà la plataforma d'e-learning (Aula Virtual) de la Universitat de València com a suport de comunicació amb els estudiants. A través d'ella es tindrà accés al material didàctic utilitzat en classe, així com els problemes i exercicis a resoldre.

AVALUACIÓ

Els coneixements adquirits per l'estudiant es podran avaluar de les dos formes següents:

- Sistema d'avaluació contínua
- Sistema d'avaluació única.

Sistema d'Avaluació Contínua

Este és el mètode que es recomanarà als alumnes. Per mitjà d'este sistema s'avaluarà de forma regular la participació dels alumnes en l'aprofitament d'activitats formatives i la participació dels alumnes en el procés d'aprenentatge.

Es valoraran els següents aspectes:

- **Sessions Teòriques:** es valorarà la implicació, tenint en compte la assistència regular a las activitats presencials previstes, la entrega dels exercicis proposats i la participació en la seua resolució, incloent el treball sobre els temes 1 i 2 (N_Teoria).
- **Sessions Problemes:** es valorarà la implicació, tenint en compte la assistència regular a las activitats presencials previstes, l'entrega dels exercicis proposats, la participació en la seua resolució durant les classes i la participació activa als fors (N_Pràctiques).
- **Sessions de Laboratori**: es valorarà la implicació, tenint en compte l'assistència regular a les activitats presencials previstes i l'entrega dels exercicis proposats (N_Laboratori).
- **Prova objectiva individual:** consistent en un o diversos controls, o prova de coneixement, que constaran tant de qüestions teoricopràctiques com de problemes (N_Examen).

Per a poder aplicar este tipus d'avaluació serà necessari un índex d'assistència a les classes superior al 75%. Este percentatge s'aplicarà de forma separada a cada bloc. És a dir, serà necessari assistir a més del 75% de les sessions de teoria, a més del 75% de les sessions de pràctiques i a més del 75% de les sessions de laboratori.

Només es consideraran els treballs entregats en la data estipulada pel professor. Açò inclou els exercicis proposats en classe (teoria i pràctiques), els exercicis de laboratori, el treball dels dos primers temes i el projecte programari.

Per a poder amitjanar les notes de les diferents categories se'ls demanarà una nota mínima de 4 punts (sobre 10) en cada una d'elles (*N_Teoria, N_Problemes, N_Laboratorio, N_Examen)*.

La nota final s'obtindrà d'aplicar la següent fórmula

Nota=20%N_Continua+30%N_Laboratori+50%N_Examen

N_Continua=50%N_Teoria+50%N_Problemes

Sistema d'Avaluació Única

Este mètode s'aplicarà a qualsevol alumne que, por un motivo raonat i admès pel professor, no puga assistir amb regularitat a les classes o bé no haja superat l'avaluació continua en primera convocatòria.

Les notes de les sessions de teoria (N_Teoría), de problemes (N_Problemes) i de laboratori (*N_Laboratorio*) no es recuperaran amb cap altra activitat.

En ambdós mètodes, l'avaluació s'ajustarà a la Normativa de Qualificacions de la Universitat de València. Al moment de redacció de la present guia docent, la normativa vigent és la aprovada pel Consell de Govern de la UVEG de 27 de gener de 2004, que s'ajusta a lo establert a tal efecte pels Reials Decrets 1044/2003 i 1125/2003. En ella s'estableix bàsicament que las qualificacions seran numèriques de 0 a 10 amb expressió d'un decimal i a las que cal afegir la qualificació qualitativa adient a la escala següent:

De 0 a 4,9: "Suspens"

De 5 a 6,9: "Aprovat"

De 7 a 8,9: "Notable"

De 9 a 10: "Excel·lent" o "Excel·lent amb Matrícula de Honor"

REFERÈNCIES

Bàsiques

- Apuntes de la asignatura
- [Roger S. Pressman (2009)] Software Engineering: A Practitioner's Approach, 7th Edition (Mc Graw Hill)
- [I. Sommerville (2011)] Software Engineering, 9th Edition (Addison-Wesley)
- [S. Sánchez Alonso, M. A. Sicilia Urbán, D. Rodríguez García (2011)] Ingeniería de software: un enfoque desde la guía SWEBOK (Garceta)
- [Grady Booch, James Rumbaugh, Ivar Jacobson (2005)] The Unified Modeling Language User Guide (2nd Rev. Edition) (Addison-Wesley)
- [C. Larman (2004)] Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd (Edition Prentice Hall)
- [Bernd Bruegge, Allen H. Dutoit] Object-Oriented Software Engineering Using UML, Patterns, and Java, 3rd Edition (Edition Prentice Hall)

Complementàries

- [Kenneth E. Kendall, Julie E Kendall (2010)] Systems Analysis and Design, 8th Edition (Prentice Hall)
- [Michael R. Blaha, James R Rumbaugh (2005)] Object-Oriented Modeling and Design with UML (2nd Edition) (Prentice Hall)

- [A. Weitzenfeld (2004)] Ingeniería de software orientada a objetos con UML, Java e Internet (Thomson)
- [Robert C. Martin (2003)] UML for Java programmers (Prentice Hall)