

**FITXA IDENTIFICATIVA****Dades de l'Assignatura**

<b>Codi</b>	34886
<b>Nom</b>	Enginyeria del programari
<b>Cicle</b>	Grau
<b>Crèdits ECTS</b>	6.0
<b>Curs acadèmic</b>	2023 - 2024

**Titulació/titulacions**

<b>Titulació</b>	<b>Centre</b>	<b>Curs</b>	<b>Període</b>
1403 - Grau d'Enginyeria Telemàtica	Escola Tècnica Superior d'Enginyeria	3	Primer quadrimestre

**Matèries**

<b>Titulació</b>	<b>Matèria</b>	<b>Caràcter</b>
1403 - Grau d'Enginyeria Telemàtica	11 - Ingeniería del Software	Obligatòria

**Coordinació**

<b>Nom</b>	<b>Departament</b>
PANACH NAVARRETE, JOSE IGNACIO	240 - Informàtica

**RESUM**

L'assignatura “Enginyeria del Programari” és una assignatura obligatòria que forma part de la matèria Enginyeria del Programari i Gestió de Projectes del Grau en Enginyeria Telemàtica. 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 desenvolupar 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'alumnat en el coneixement i maneig de diferents metodologies de desenvolupament de sistemes d'informació.

Es tractarà d'aconseguir un coneixement suficient del procés de programari, de manera que l'alumnat siga capaç de, usant com a mètode el Procés Unificat de Desenvolupament, 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'alumnat utilitzant el llenguatge de modelatge UML i 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'alumnat 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.

## **CONEXEMENTS 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**

Es requisit imprescindible:

Haver aprovat les assignatures de primer i segon curs Informàtica (34877) i Ampliació d'Informàtica (34878)

Estar matriculat/ada o haver aprovat la assignatura Programació (34888)



## COMPETÈNCIES (RD 1393/2007) // RESULTATS DE L'APRENTATGE (RD 822/2021)

### 1403 - Grau d'Enginyeria Telemàtica

- R1 - Capacitat per aprendre de manera autònoma nous coneixements i tècniques adequats per a la concepció, el desenvolupament o l'explotació de sistemes i serveis de telecomunicació.
- G4 - Capacitat per resoldre problemes amb iniciativa, presa de decisions, creativitat, i de comunicar i transmetre coneixements, habilitats i destreses, comprnent la responsabilitat ètica i professional de l'activitat de l'enginyer tècnic de telecomunicació.
- G9 - Capacitat per treballar en un grup multidisciplinari i en un entorn multilingüe i de comunicar, tant per escrit com de forma oral, coneixements, procediments, resultats i idees relacionades amb les telecomunicacions i l'electrònica.
- R7 - Coneixement i utilització dels fonaments de la programació en xarxes, sistemes i serveis de telecomunicació.
- E3 - Capacitat per construir, explotar i gestionar serveis telemàtics utilitzant eines analítiques de planificació, de dimensionat i d'anàlisi.
- E4 - Capacitat per descriure, programar, validar i optimitzar protocols i interfícies de comunicació en els diferents nivells d'una arquitectura de xarxes.

## RESULTATS D'APRENTATGE (RD 1393/2007) // SENSE CONTINGUT (RD 822/2021)

Esta assignatura permet obtindre els següents resultats d'aprenentatge:

- Aplicar metodologies per al desenrotllament, implantació i manteniment de sistemes d'informació. **Competències G4, G9, R1, R7, E3 i E4**
- Planificar i executar correctament processos de desenrotllament de programari iteratius. **Competències G4, G9, R1, R7, E3 i E4**
- Saber aplicar patrons de disseny programari en cada situació en funció de les necessitats del projecte de desenrotllament programari. **Competències G4, R1, E3 i E4**
- Definir proves de validació i verificació de requisits. **Competències G4, R1, E3 i E4**
- Obtindre requisits d'usuari i del programari. **Competències G4, G9, R1, E3 i E4**
- Conèixer els conceptes bàsics, els processos de desenvolupament, els mètodes i les eines de l'enginyeria de programari. **Competència E3**
- 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. **Competències G4, R1 i E4**
- Conèixer els diferents models d'arquitectura, les tecnologies que poden integrar-se i les solucions comercials per formar una solució distribuïda particular. **Competències E3 i E4**



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. **Competències G4, G9, R1 i E3**
- Dissenyar una estructura de mòduls, utilitzant patrons de disseny, per a solucionar un problema i avaluar altres alternatives. **Competències G4, G9, R1 i E3**
- Implementar un mòdul que s'execute correctament i de manera eficient. **Competències R7, E3 i E4**
- Provar aplicacions de manera sistemàtica definint casos de prova exhaustius. **Competència E4**
- Treballar en un equipo reduït, col·laborant en tots els aspectes del desenvolupament de software intercanviant idees de manera constructiva i organitzada. **Competències G4, G9, R1, R7, E3 i E4**

## 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
- Conèixer 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
- Conèixer les característiques i explicar els avantatges i desavantatges de diferents models de procés del programari
- Conèixer els principals tipus de metodologies programari
- Conèixer les característiques bàsiques dun procés de desenrotllament del Programari generalista
- Entendre en què consistix el modelatge de programari i quins beneficis aporta
- Reconèixer 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 de Procés del Programari
- 1.4 Modelatge de Programari
- 1.5 El Llenguatge Unificat de Modelatge UML 2.0
  - 1.5.1 Marc Conceptual UML
  - 1.5.2 Vistes UML
- 1.6 Un Procés de Desenvolupament Software OO
  - 1.6.1 Fases
  - 1.6.2 Activitats i Artefactes

Laboratori:

Totes les sessions



## 2. Fase de Planificació

Destreses a adquirir:

- Comprendre la importància d'obtenir 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 Requisits Funcionals vs. No Funcionals

#### 2.1.3 Document de Requisits del Programari

#### 2.1.4 Exercicis sobre requisits

### 2.2 Prototip

### 2.3 Casos d'Ús

#### 2.3.1 Introducció.

#### 2.3.2 Actores

#### 2.3.3 Especificació de Casos d'Ús

#### 2.3.4 Relacions: generalització, extensió, inclusió

#### 2.3.5 Diagrames de Casos d'Ús

#### 2.3.6 Modelatge de Casos d'Ús

#### 2.3.7 Exercicis sobre Casos d'Ús.

Laboratori:

Sessió 1: Desenvolupament del Prototip de la UGI

Sessió 2: Treballant amb Diagrames de Casos d'Ús

## 3. Anàlisi

Destreses a adquirir:

- Conèixer els passos que s'ha de seguir per a completar l'etapa d'anàlisi en el primer cicle de desenvolupament i els artefactes a crear
- Ser capaç d'elaborar el Diccionari de Dades
- Ser capaç d'abstracte 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 la seua especificació expandida
- Desenvolupar els Contractes de les Operacions del Sistema

Continguts:

Part I:

- 3.1 Introducció
- 3.2 Diagrama de classes
  - 3.2.1 Classificadors
  - 3.2.2 Classes
  - 3.2.3 Interfases
  - 3.2.4 Relacions: dependència, generalització, associació, realització
- 3.3 Model Conceptual
- 3.4 Exercicis de diagrames de classes i objectes

Part II:

- 3.5 Interaccions
- 3.6 Diagrames de Seqüència
  - 3.6.1 Elements
  - 3.6.2 Modelatge de diagrames de seqüència
  - 3.6.3 Aplicacions al cicle de vida
- 3.7 Diagrames de Seqüència Generals del Sistema
- 3.8 Contractes
- 3.9 Exercicis de diagrames de seqüència i contractes

Laboratori:

- Sessió 3: Treballant amb Diagrames de Classes
- Sessió 4: Treballant amb Diagrames de Seqüència
- Sessió 5: Treballant el Disseny del Cicle 1
- Sessió 7: Treballant l'Anàlisi i el Disseny del Cicle 2

## 4. Disseny

Destreses a adquirir:

- Conèixer els passos que s'ha de seguir per a completar l'etapa de disseny en el primer cicle de desenvolupament 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 Disseny del Sistema



- 4.1.1 Responsabilitats
- 4.1.2 Diagrames de Seqüència de Disseny
- 4.1.3 Diagrames de Classes de Disseny
- 4.1.4 Patrons para la assignació de responsabilitats.
- 4.2 Exercicis

Laboratori:

Sessió 5: Treballant el Disseny del Cicle 1

Sessió 7: Treballant l'Anàlisi i el Disseny del Cicle 2

## 5. Implementació

Destreses a adquirir:

- Conèixer les decisions prèvies a prendre abans d'implementar
- Conèixer 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:

- 5.1 Decisions prèvies
- 5.2 Tipus de transformació
  - 5.2.1 Transformacions del model
  - 5.2.2 Transformacions del codi
  - 5.2.3 Transformacions del model al codi: Enginyeria directa
  - 5.2.4 Transformacions del codi al model: Enginyeria inversa
- 5.3 Enginyeria directa
  - 5.3.1 Mapeu de Classes
  - 5.3.2 Mapeu de Relacions
  - 5.3.3 Mapeu d'Herència
  - 5.3.4 Creació de Mètodes
  - 5.3.5 Mapeu de Contractes
- 5.4 Exercicis d'Implementació

Laboratori:

Sessió 6: Treballant la Implementació del Cicle 1

Sessió 8: Treballant la Implementació del Cicle 2

## 6. 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
- Conèixer i saber aplicar altres patrons



Continguts:

- 6.1 Arquitectura multicapa y UML
- 6.2 Patrones de connexió entre paquets
- 6.3 Exercicis

Laboratori:

Sessió 7: Treballant l'Anàlisi i el Disseny del Cicle 2

## 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 de Sistema
  - 7.3.4 Proves de Validació
- 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
Assistència a esdeveniments i activitats externes	3,00	0
Elaboració de treballs en grup	14,00	0
Elaboració de treballs individuals	6,00	0
Estudi i treball autònom	7,00	0
Preparació de classes de teoria	20,00	0
Preparació de classes pràctiques i de problemes	30,00	0
Resolució de casos pràctics	7,00	0
Resolució de qüestionaris on-line	3,00	0
<b>TOTAL</b>	<b>150,00</b>	

**METODOLOGIA DOCENT****Classes presencials.**

Les classes presencials es basaran en classes expositives actives on s'introduiran cada 20/25 minuts alguna activitat que exigeixi la intervenció dels alumnes, de manera que: 1) puguin posar en pràctica de forma immediata els continguts que acaben de veure; 2) recuperar el nivell d'atenció al següent bloc expositiu.

**Preparació de classes teòriques.**

Els alumnes haurien de preparar el contingut de la classe teòrica, seguint la planificació de l'assignatura. Per a això faran ús de la bibliografia suggerida pel professor així com dels materials proporcionats per aquest de manera eventual o altres orientacions donades.

Als alumnes se'ls proposaran activitats que haurien de realitzar a casa individualment o en grup i que en ocasions seran necessàries per a la realització de la següent sessió teòrica. Aquestes activitats podran ser avaluades abans del començament la classe o durant la classe així com en hores de tutories.

**Preparació de treballs pràctics.**

Per a assimilar millor els continguts de les classes teòriques, es realitzaran sessions pràctiques presencials. L'assistència a les sessions pràctiques és obligatòria i es verificarà per part del professor. Aquells alumnes que per motius laborals no puguin assistir han de posar-se en contacte abans del començament de les



pràctiques amb el seu professor de pràctiques. Els resultats d'aquestes activitats s'haurien de presentar al professor de forma escalonada al llarg del curs i en els termes que estableixi el professor. Els alumnes realitzaran/prepararan part d'aquestes activitats a casa. L'assistència a pràctiques és obligatòria.

### **Realització de treballs en equip.**

Al llarg del curs es plantegessin un conjunt de problemes de mitjana envergadura que deuran ser resolts en equips de 5 a 6 persones.

En el procés d'avaluació dels treballs en equip es qualificarà tant la nota conjunta del grup com la nota individual de cada membre.

S'utilitzarà la plataforma de e-learning (Aula Virtual) de la Universitat de València com suport de comunicació amb l'alumnat. 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'estudiantat s'avaluaran de la següent manera:

S'avaluarà de forma regular la seua participació en les diferents activitats, el grau d'aprofitament obtingut en les diferents activitats formatives i la implicació mostrada cap al seu propi procés d'aprenentatge. Per a això es valoraran els següents aspectes:

- C) Avaluació contínua, basada en la participació i grau d'implicació en el procés d'ensenyament-aprenentatge, tenint en compte l'assistència regular a les activitats presencials previstes i la realització dels treballs. Com a activitats dins de l'avaluació contínua, els alumnes realitzaran de manera individual un conjunt de butlletins d'exercicis pràctics o de desenvolupament teòric que seran entregats a través d'aula virtual dins del termini establert per a això. A més, es realitzaran controls tipus test o de qüestions breus d'una part de la matèria. Finalment, al llarg del curs s'elaborarà un projecte programari que es presentarà en finalitzar el curs en classe. Totes aquestes activitats donaran lloc a la nota d'avaluació contínua de la següent forma:

$$C \text{ (Nota Avaluació Contínua)} = 0,4 * \text{Controls} + 0,4 * \text{Butlletins} + 0,2 * \text{Presentació Projecte}$$

No es tindran en compte les activitats entregades fora de termini, ni es podran recuperar les activitats no realitzades. La còpia en qualsevol de les activitats serà penalitzada de manera estricta anul·lant-se totes les notes d'avaluació contínua de l'alumne.



### Competències G4, G9, R1, R7, E3 i E4

- (P) Avaluació de pràctiques. Es faran en grups de 5 o 6 persones i es valorarà tant la qualitat de la solució com la documentació en cadascuna de les pràctiques. **Competències G4, G9, R1, R7, E3 i E4**
- (E) Proves objectives individuals, consistent en un o diversos exàmens, o proves de coneixement, que constaran tant de qüestions teoricopràctiques com de problemes. Serà necessari aprovar cadascuna d'aquestes proves o exàmens per a poder superar l'assignatura. **Competències G4, R1 i E3**

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

$$\text{Nota} = 0,35 * C + 0,35 * E + 0,3 * P$$

Les notes de les activitats en grup no seran les mateixes necessàriament per a tots els membres del grup, podent variar en funció de la implicació de cada alumne/a.

Només es consideraran els treballs entregats en la data estipulada pel professorat. Això inclou totes i cadascuna de les activitats, qüestionaris i exercicis proposats, el projecte programari i, en general qualsevol tasca que se li encarregue a l'alumnat.

La part contínua no és recuperable en segona convocatòria. Les parts P i E només fan una mitjana de si s'aconsegueix un mínim de 5 en cadascuna d'elles.

En cas de no haver superat E o P amb nota superior a 5 i haver-se presentat a la prova E, la nota en actes es computarà com:

$$\text{Nota Final} = \text{mínim}(E, P, 4)$$

En cas de no presentar-se a E, la nota final és No Presentat



Atés que la qualificació de la part d'avaluació contínua no és recuperable, per a poder sol·licitar avançament de convocatòria, l'alumnat haurà d'haver cursat prèviament l'assignatura i haver superat l'avaluació contínua.

En qualsevol cas, l'avaluació de l'assignatura es farà d'acord amb el Reglament d'avaluació i qualificació de la Universitat de València per als títols de grau i màster aprovat per Consell de Govern de 30 de maig de 2017 (ACGUV 108/2017).

D'acord amb el reglament de la Universitat de València, la realització d'actuacions fraudulentament en una prova o part d'ella donarà lloc a la qualificació d'un zero en aquesta, amb independència del procediment disciplinari que es pugui obrir i de la sanció que siga procedent d'acord amb la normativa vigent.

## REFERÈNCIES

### Bàsiques

- Apuntes de la assignatura
- [Frank Tsui, Orlando Karam, and Barbara Bernal(2016)] Essentials of Software Engineering. Jones & Bartlett Learning, LLC  
[Recurs Electrònic:  
<https://ebookcentral.proquest.com/lib/univalencia/detail.action?docID=4826428>]
- [Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel] UML @ Classroom. An Introduction to Object-Oriented Modeling  
[Recurs Electrònic: <https://link.springer.com/book/10.1007%2F978-3-319-12742-2>]
- [C. Larman (2004)] Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd (Edition Prentice Hall)[Recurs electronic:  
<http://proquest.safaribooksonline.com/0131489062?uicode=valencia>]
- [Grady Booch, James Rumbaugh, Ivar Jacobson (2005)] The Unified Modeling Language User Guide (2nd Rev. Edition) (Addison-Wesley) [Recurs electronic:  
<http://proquest.safaribooksonline.com/0321267974>]

### 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)[Recurs electronic: <http://proquest.safaribooksonline.com/0131428489?uicode=valencia>]
- [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)
- [Bernd Bruegge, Allen H. Dutoit] Object-Oriented Software Engineering Using UML, Patterns, and Java, 3rd Edition (Edition Prentice Hall)