



Guia docent

230719 - ARQSOFT - Arquitectura de Software

Última modificació: 10/11/2022

Unitat responsable: Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Unitat que imparteix: 701 - DAC - Departament d'Arquitectura de Computadors.

Titulació: MÀSTER UNIVERSITARI EN ENGINYERIA DE TELECOMUNICACIÓ (Pla 2013). (Assignatura optativa).
MÀSTER UNIVERSITARI EN ENGINYERIA AERONÀUTICA (Pla 2014). (Assignatura optativa).
MÀSTER UNIVERSITARI EN TECNOLOGIES AVANÇADES DE TELECOMUNICACIÓ (Pla 2019). (Assignatura obligatòria).

Curs: 2022 **Crèdits ECTS:** 5.0 **Idiomes:** Anglès

PROFESSORAT

Professorat responsable: Consultar aquí / See here:
<https://telecos.upc.edu/ca/estudis/curs-actual/professorat-responsables-coordinadors/responsables-assignatura>

Altres: Consultar aquí / See here:
<https://telecos.upc.edu/ca/estudis/curs-actual/professorat-responsables-coordinadors/professorat-assignat-idioma>

CAPACITATS PRÈVIES

Bases conceptuals sòlides de programació orientada a objectes: objecte, atributs, mètodes, classe, excepció, encapsulació de dades, herència i polimorfisme. Bases sòlides de programació en Java: sintaxi de definició de classes, sintaxi de gestió d'herència, polimorfisme. Gestió d'excepcions en Java. Entrada / Sortida bàsica en Java. Gestió de contenidors més comuns: conjunts, llistes, mapes, grafs.

METODOLOGIES DOCENTS

L'assignatura s'impartirà en base a sessions de teoria en què es presentaran els conceptes teòrics més rellevants, acompanyats d'exemples d'aplicació dels mateixos. S'intercalaran al llarg del curs sessions de laboratori que serviran per consolidar l'assimilació dels continguts teòrics i posar aquests en pràctica.

S'haurà de desenvolupar un programa de grau de complexitat moderat / alt per grups, on es posaran en pràctica els elements teòrics presentats durant el curs.

OBJECTIUS D'APRENENTATGE DE L'ASSIGNATURA

Aconseguir que qui cursi aquesta assignatura sigui capaç de:

1. Afrontar amb èxit el treball en el si d'un equip que confronti la realització d'un programa de complexitat moderada / alta;
2. Dur a terme tasques d'anàlisi orientat a objectes, disseny orientat a objectes utilitzant correctament patrons de disseny de programari;
3. Desenvolupar entorns automatitzats de proves de programes.
4. Afegir als programes el nivell de persistència de dades utilitzant bases de dades relacionals. Ser capaç d'afrontar amb èxit el treball en el si d'un equip que confronti la realització d'un programa de complexitat moderada / alta. Ser capaç de dur a terme tasques de: anàlisi orientat a objectes, disseny orientat a objectes utilitzant correctament patrons de disseny de programari. Ser capaç de desenvolupar entorns automatitzats de proves de programes. Ser capaç d'implementar un nivell de persistència de dades utilitzant bases de dades relacionals.



HORES TOTS DE DEDICACIÓ DE L'ESTUDIANTAT

Tipus	Hores	Percentatge
Hores grup gran	26,0	20.80
Hores aprenentatge autònom	86,0	68.80
Hores grup petit	13,0	10.40

Dedicació total: 125 h

CONTINGUTS

Introducció a les proves del programari. JUnit

Descripció:

Disseny de programari preparat per ser provat.
Proves unitàries: JUnit
Simulació de components externs ("Fakes" i "Mocks"). Eina Mockito

Objectius específics:

Presentar les activitats de proves de codi en el context del procés de desenvolupament de programari. Presentar eines que faciliten aquestes activitats. Practicar amb JUnit com a eina que permet automatitzar proves de programari

Activitats vinculades:

Sessió de laboratori. Construcció de proves per comprovar correcció de codi en una classe.

Dedicació: 8h

Grup gran/Teoria: 3h
Grup petit/Laboratori: 3h
Aprenentatge autònom: 2h

Processos iteratius de desenvolupament de programari. Introducció a Scrum

Descripció:

Noció de procés de desenvolupament de programari. Característiques principals dels processos iteratius. Comparació amb els processos en cascada. Introducció a Scrum.

Objectius específics:

Presentar breument el concepte de procés de desenvolupament de programari com un conjunt d'activitats realitzades en equip, l'objectiu de les quals és el desenvolupament d'un programa. Mostrar les característiques principals dels processos iteratius com a alternativa als processos en cascada, posant de manifest la forta imbricació temporal d'anàlisi, disseny, codificació i proves. Mostrar els detalls d'un dels processos iteratius existents: Scrum.

Activitats vinculades:

Realització d'un projecte de desenvolupament d'un programa de complexitat moderada en equip.

Dedicació: 3h

Grup gran/Teoria: 2h
Aprenentatge autònom: 1h



Introducció a l'anàlisi orientat a objectes

Descripció:

Productes de l'anàlisi orientat a objectes: repertori de casos d'ús i model conceptual del problema.

Concepte de requisits. Tipus de requisits. Tècniques de captura de requisits. Casos d'ús com a mètode per capturar requisits.

Concepte i parts d'un cas d'ús.

Construcció del model conceptual a partir dels casos d'ús. Identificació de les classes. Identificació dels atributs de les classes.

Identificació i definició completa de les relacions entre classes.

Exemples.

Objectius específics:

Categoritzar els diferents tipus de requisits d'un programa. Ser capaç d'identificar casos d'ús. Ser capaç de desenvolupar al complet els casos d'ús més rellevants.

Ser capaç de construir el model conceptual a partir dels casos d'ús: conceptes / classes, atributs i relacions.

Activitats vinculades:

Exemples de classe i projecte desenvolupat en equip

Dedicació: 16h 30m

Grup gran/Teoria: 4h 30m

Aprenentatge autònom: 12h

Introducció al disseny orientat a objectes

Descripció:

Productes del disseny orientat a objectes: diagrames de classes, diagrames d'interacció, documentació de classes dissenyades.

Introducció als diagrames d'interacció en UML: diagrames de seqüència i diagrames de col·laboració.

Diagrama de seqüència del sistema

Tècniques per identificar i especificar els mètodes de les classes del programa. Patrons GRASP per identificar mètodes i assignar-los a la classe pertinent.

Ús de patrons GRASP per construir diagrames d'interacció a partir dels casos d'ús.

Objectius específics:

Ser capaç de construir diagrames de seqüència UML. Ser capaç de construir el diagrama de seqüència d'un sistema. Ser capaç d'utilitzar correctament els patrons GRASP d'assignació de responsabilitats a les classes a partir dels casos d'ús per a identificar els mètodes de les classes que es dissenyen.

Activitats vinculades:

Exemples i projecte realitzat en equip

Dedicació: 16h 30m

Grup gran/Teoria: 4h 30m

Aprenentatge autònom: 12h



Introducció a les bases de dades

Descripció:

Concepte de bases de dades. Breu repàs històric. Tipus de bases de dades. Bases de dades relacionals. Breu introducció al SQL. Ús d'una base de dades relacional en Java.

Objectius específics:

Saber què és una base de dades relacional. Ser capaç de dissenyar taules senzilles de bases de dades relacionals. Entendre els elements bàsics d'un diagrama Entitat-Relacion. Conèixer elements introductoris del llenguatge SQL. Connectar un programa Java amb una base de dades SQL

Activitats vinculades:

Sessió de laboratori dedicada i projecte realitzat en equip

Dedicació: 6h

Grup gran/Teoria: 2h

Grup petit/Laboratori: 2h

Aprenentatge autònom: 2h

Completant el disseny orientat a objectes amb patrons de disseny

Descripció:

Introducció del concepte de patró de disseny de programari. Patrons GoF. Classificació.

Objectius específics:

Entendre el concepte de patró de disseny de programari, els seus tipus i els beneficis que el seu ús comporten en un disseny.

Activitats vinculades:

Presentació en sessió de teoria

Dedicació: 2h

Grup gran/Teoria: 1h

Aprenentatge autònom: 1h

Estudi d'alguns patrons de disseny rellevants

Descripció:

Presentació i estudi d'alguns dels patrons de disseny més rellevants: Estratègia, Decorador, Mètode Fàbrica, Fàbrica Abstracta, Compost, Observador

Objectius específics:

Conèixer els detalls dels patrons estudiats. Ser capaç de postular correctament l'ús de patrons de disseny de programari en la fase de disseny d'un programa. Ser capaç d'implementar correctament els patrons estudiats.

Activitats vinculades:

Sessions de teoria

Sessions de laboratori per a alguns patrons

Projecte realitzat en grup

Dedicació: 29h

Grup gran/Teoria: 8h

Grup petit/Laboratori: 8h

Aprenentatge autònom: 13h



Agregació de patrons: el patró model-vista-controlador

Descripció:

Estudi del patró model-vista-controlador com agregació de diversos dels patrons de disseny abans estudiats.

Objectius específics:

Aprendre els detalls del model-vista-controlador com el resultat d'afegir diversos dels patrons de disseny estudiats en el tema anterior. Implementar correctament aquesta agregació de patrons.

Activitats vinculades:

Sessió de teoria i projecte desenvolupat en equip

Dedicació: 3h

Grup gran/Teoria: 2h

Aprenentatge autònom: 1h

Projecte

Descripció:

contingut català

Dedicació: 41h

Aprenentatge autònom: 41h

SISTEMA DE QUALIFICACIÓ

Lliuraments parcials d'activitats relacionades amb el projecte: 10%

Examen: 20%

Projecte: 60%

Avaluació subjectiva: 10%

NORMES PER A LA REALITZACIÓ DE LES PROVES.

L'examen es realitzarà sobre l'ordinador. No es permetrà l'ús de sistemes de missatgeria electrònica de cap tipus.

BIBLIOGRAFIA

Bàsica:

- Freeman, E.; Freeman, E.; Sierra, K.; Bates, B. Head First design patterns. 1st ed. Sebastopol, CA: O'Reilly, 2004. ISBN 0-596-00712-4.

- Larman, C. Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development. 3rd ed. Upper Saddle River, N.J.: Prentice Hall PTR, 2005. ISBN 0-13-148906-2.

- Gamma, E.; Helm, R.; Johnson, R.; Vissides, J. Patrones de diseño: elementos de software orientado a objetos reutilizable. 1st ed. Madrid: Addison Wesley, 2003. ISBN 84-7829-059-1.

RECURSOS

Enllaç web:

- Writing effective use cases / Alistair Cockburn. Boston : Addison-Wesley, cop. 2001. ISBN: 0201702258.
<https://www.infor.uva.es/~mlaguna/is1/materiales/BookDraft1.pdf>.

https://discovery.upc.edu/discovery/fulldisplay?docid=alma991002517209706711&context=L&vid=34CSUC_UPC:VU1&lang=ca- Java design patterns - example tutorial. <https://www.journaldev.com/1827/java-design-patterns-example-tutorial/>- Design patterns.

<https://www.oodesign.com/>