

Projecte de Programació

Curs 2017/18 – Q2

Introducció

- Assignatura de projecte
- Introducció pràctica a la OOP
- Fomenta el treball en equip

- Classes de teoria:

Expliquen allò que cal per desenvolupar el projecte: conceptes OO, Java, Arquitectura 3 capes, etc.

- Utilitzarem part del que ja sabeu de IES però ens interessarà només allò que cal pel projecte.

- Classes de laboratori

(comencen la setmana del 26 de febrer):

Essencialment les utilitzareu per desenvolupar el projecte i resoldre dubtes amb el vostre tutor. També el tutor les pot fer servir per aclariments relacionats amb el projecte.

- Decisions predeterminades a PROP:

Enfoc del projecte: Orientació a Objectes

Notació disseny: UML

Metodologia de disseny:

Arquitectura de 3 capes

Llenguatge de programació: Java

- Exàmen teoria: **23/5** 12:30h – 14:30h
- Nota Final PROP:

80% Nota projecte

20% Nota exàmen

Cal tenir en compte que el projecte es fa en equip, però la nota del projecte és individual

- Projecte: DOS entregues.

Entrega 1 (2/5 17h):

Casos d'ús, disseny i implementació de les funcionalitats principals. Alguns tutors poden demanar exposicions presencials.

Entrega 2 (8/6 17h):

Programa complet. Entrega i demostració personal davant del tutor.

- Nota Projecte:

Sigui: N_k la nota de l'entrega k ($k=1,2$),
FT el factor de treball individual (entre 0 i 1)

La nota individual del projecte és:
$$FT \times (0.4 \times N_1 + 0.6 \times N_2)$$

Cada grup de laboratori s'anomena cluster.

Cada cluster té un tutor i es compon de un cert nombre de grups de 3 persones cada grup.

Hi ha un sol enunciat per a tots els grups i clusters. Surt publicat durant la 1^a setmana.

Cada grup de laboratori s'anomena cluster.

Cada cluster té un tutor i es compon de un cert nombre de grups de 3 persones cada grup.

Els canvis de grup només estan permesos si són intercanvis entre grups. És a dir, si algú vol canviar del grup A al grup B cal que trobi a algú del grup B que vulgui anar al grup A.

Programa classes de teoria:

- Introducció
- Repàs IES: Cicle de vida del programari
- Introducció a Java
- Repàs conceptes OO i implementació en Java
- Arquitectura 3 capes i implementació en Java
- Prova de programes: Drivers & Stubs
- Prova de programes: Testing amb Junit
- Patrons amb Java
- Disseny d'Interfícies amb Java

Darrera sessió teoria: **Setmana 14/5** (grups dilluns)
Setmana 21/5 (grups dimarts)

- Bibliografia:

An Introduction to Object Oriented Programming

Timothy Budd

Addison-Wesley 2002

The Object Oriented Thought Process (3rd. ed)

Matt Weisfeld

Pearson Ed. 2009

Smalltalk, Objects and Design

Chamond Liu

IUniverse 2000

JUnit in Action (2nd. ed.)

Tahcheiv, Petar; Leme, Felipe; Massol, Vincent; Gregory,
Gary

Manning , 2011

Cicle de Vida

Fases per les que passa una aplicació informàtica des de que es comença a pensar fins que deixa de ser útil

Guia la *metodologia* que s'utilitza. Hi ha tantes metodologies com cicles de vida:

Cowboy coding

Waterfall (cascada)

Espiral

Iteratiu

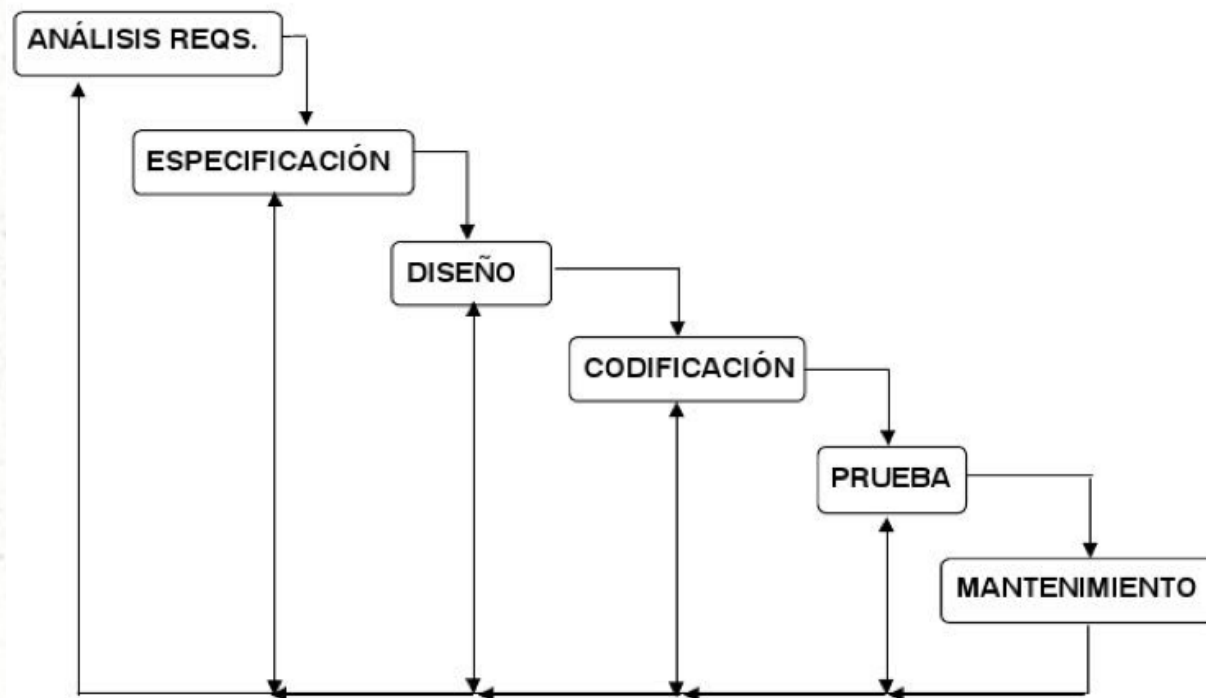
Àgil

etc...

(http://en.wikipedia.org/wiki/Software_development_process)

Cicle de Vida

A PROP farem servir el *model clàssic augmentat*, basat en el model de cascada



En aquest model podem replantejar-nos etapes prèvies del desenvolupament.

Cicle de Vida

Descripció de les etapes: **Anàlisi Requeriments**

En aquesta etapa cal saber **QUÈ** vol l'usuari. L'usuari és el “protagonista” d'aquesta etapa. A partir de la informació que ens proporciona cal esbrinar:

- Quin sistema cal construir
- És possible construir-lo?
- El que demana l'usuari és el que ell necessita?
etc...

Aquesta fase és *independent* de la tecnologia.

Cicle de Vida

Descripció de les etapes: **Anàlisi Requeriments**

- Requeriments del sistema
- Requeriments del *software*:
 - Funcionals
 - No Funcionals
- Document amb la descripció de l'aplicació
- Casos d'ús: Identificació

Cicle de Vida

Descripció de les etapes: **Especificació**

En aquesta etapa cal saber **QUÈ** ha de fer el sistema i descriure-ho detalladament.

- Especificació de les *dades*:

 - Model conceptual de les dades

- Especificació dels *processos*:

 - Model de comportament del sistema

Aquesta fase és *independent* de la tecnologia (no cal decidir encara el *com*).

Cicle de Vida

Primera entrega projecte (I):

- **Anàlisi requeriments:**
 - Casos d'ús (identificació)
- **Especificació:**
 - Model conceptual dades (diagrama de classes del model, versió especificació)
 - Casos d'ús (interacció amb l'usuari)

Cicle de Vida

Descripció de les etapes: **Disseny**

En aquesta etapa cal començar a pensar **COM** cal fer el que s'ha proposat a les etapes anteriors:

- Arquitectura de l'aplicació (a PROP no tindreu elecció, cal fer servir l'*arquitectura de tres capes*).
- Model conceptual de les dades (versió disseny)
- Diagrames de seqüència de les operacions*

Cicle de Vida

Descripció de les etapes: **Disseny**

En aquesta etapa cal començar a pensar **COM** cal fer el que s'ha proposat a les etapes anteriors:

- Contractes de les operacions de les classes*
- Llenguatge de programació
(a PROP serà Java)
- Estructures de dades i algorismes

*No caldrà fer-ho a PROP

Cicle de Vida

Descripció de les etapes: **Codificació i Tests**

En aquesta etapa cal implementar tot allò que s'ha decidit dins la fase de disseny.



Cal fer *tests* sobre tot allò implementat.

Cicle de Vida

Primera entrega projecte (II):

- **Disseny:**

- Arquitectura de l'aplicació (3 capes)
- Diagrama de classes del domini
- Estructures de dades i Algorismes
- Prototipus de l'interfície gràfica

- **Codificació i Tests:**

- Domini completament implementat i provat
- Codi de les funcionalitats principals
- Drivers, stubs, tests (JUnit) + Jocs de proves

Segona entrega projecte: **Projecte complet.**