
Programación II

Docentes

Ariel Rossanigo (arielrossanigo@gmail.com)

Sebastian Henzenn (sebyshenzenn@gmail.com)

Grupo de trabajo

Dirección web del grupo: https://groups.google.com/group/programacion-ii-ucse-dar

Condiciones de la materia

Para regularizar:

- Aprobar el examen con 60% o más
- Aprobar el TP (entrega y defensa)

Para promocionar

- Aprobar el examen con 80% o más
- Aprobar el TP (entrega y defensa)
- Aprobar un coloquio de unidades faltantes

Complejidad del software

Tipos de software:

- Software no complejo.
- Software complejo.

¿Por qué el software es complejo?

- **Complejidad del dominio del problema**
  - Muchos requerimientos
  - *Communication gap* entre programadores y usuarios
  - Los requerimientos cambian durante el proceso
- **Dificultad para gestionar el proceso de desarrollo**
  - El software crece hasta un punto donde se necesita un equipo de desarollo
  - El equipo demanda trabajo extra (comunicacion, integridad de diseño)

¿Por qué el software es complejo?

- **Flexibilidad alcanzable gracias al software**
- **Problemas de caracterizar el comportamiento de sistemas discretos**
  - Pequeñas variaciones en la entrada pueden producir grandes variaciones en las salidas
  - Necesidad de testing

Atributos de sistemas complejos

- **Estructura jerárquica**
  - Cada sistema está formado de subsistemas.
  - El valor agregado depende de las relaciones y no de los subsistemas en si
- **Primitivas relativas**
  - Lo que para una persona es primitivo, para otro puede no serlo

Atributos de sistemas complejos

- **Separación de preocupaciones**
  - El soft puede ser modularizado, aunque los módulos tengan dependencias entre ellos
  - Los vinculos internos de un módulo son más fuertes que los externos módulos
  - Permite el estudio de los módulos por separado
- **Patrones comunes**
  - Permite reutilizar pequeños componentes que forman partes de diversos sistemas

Atributos de sistemas complejos

- **Formas intermedias estables**
  - Los sistemas tienden a evolucionar
  - En la mayoría de los casos, los sistemas complejos son evoluciones de sistemas más simples que funcionaban
  - Es muy dificil producir un sistema complejo desde cero.

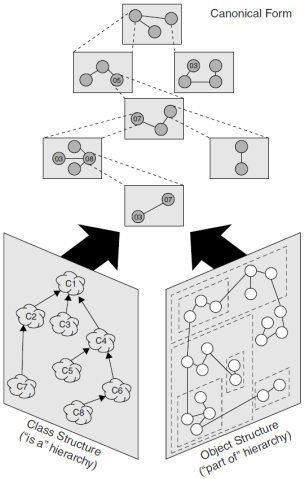
Forma canónica de un sistema complejo

- **Jerarquía *"parte de"* <=> estructura de objetos**
    - Representa los objetos que interactúan en el sistema
- **Jerarquía *"es un"* <=> estructura de clases**
    - Captura propiedades comunes de los objetos

En ambas jerarquías los niveles muestran el nivel de primitivas del sistema.

El conjunto de ambas jerarquías se denomina *arquitectura*

Forma canónica de un sistema complejo



Limitaciones de las personas para lidiar con la complejidad

Si sabemos como lidiar con sistemas complejos, ¿Por qué seguimos teniendo problemas en desarrollarlos?

Los sistemas aumentan en complejidad pero las personas solo pueden lidiar con 5 +/- 2 cosas a la vez.

La pregunta a responder es: ¿Como podemos resolver este dilema?

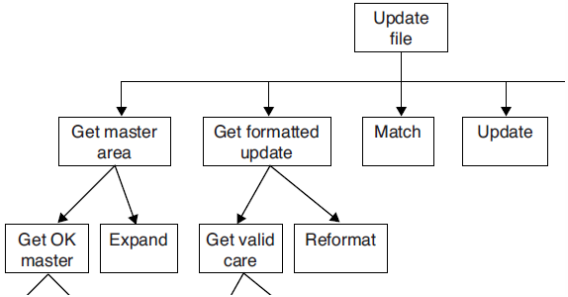
- Descomposición
- Abstracción
- Jerarquías

Descomposición (divide et impera)

Descomponer un sistema en subsistemas hasta que estos sean lo suficientemente pequeños como para entenderlos.

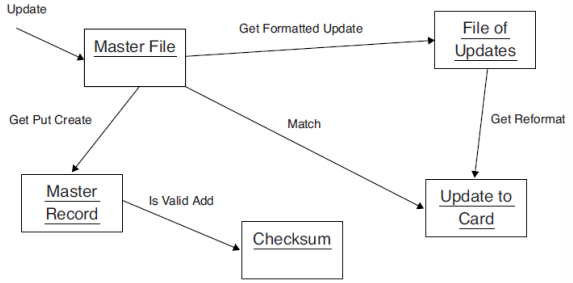
Descomposición algoritmica

Cada módulo consiste en un paso de un proceso que lo contiene



Descomposición orientada a objetos

Descomposición en objetos que se derivan directamente del vocabulario del problema. Se ve el mundo como un conjunto de objetos autónomas que colaboran entre si para lograr un objetivo.



Bibliografía y enlaces útiles.

- Booch, Grady et all: Object Oriented Analysis and Design with applications - Third Edition