

# WUOLAH



Ram\_06

[www.wuolah.com/student/Ram\\_06](http://www.wuolah.com/student/Ram_06)



1372

## Resumen-tema-3.pdf

*Resumen tema 3.2*



2º Fundamentos de la Ingeniería del Software



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación  
Universidad de Granada

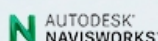
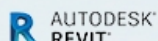
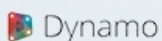


Escuela de **LÍDERES**

## Master BIM Management



60 Créditos ECTS



Jose María Girela  
Bim Manager.



## TEMA 3.2: DISEÑO DE LOS CASOS DE USO

### - MODELO DE INTERACCIÓN DE OBJETOS -

Es el resultado del **diagrama conceptual** y del **contrato de las operaciones**. Para representarlo se usan los diagramas de interacción UML de **secuencia** y de **comunicación**, que son semánticamente equivalentes.

### - PATRONES DE DISEÑO DE CRAIG LARMAN -

- Una **responsabilidad** es una obligación que tiene un objeto en su comportamiento, y debe incluirse en una operación o método de la clase a la que este pertenece.

Las responsabilidades de un objeto pueden ser:

- **Conocer**: los datos privados encapsulados por él, los objetos relacionados con él y la información que puede calcular o derivar.
- **Hacer**: algo él mismo, iniciar una acción en otros objetos, o controlar y coordinar actividades de otros objetos.

- **Patrón de diseño**: descripción de un problema con su solución en un determinado contexto.

Partes esenciales de un patrón: nombre del patrón, **problema**, **solución** y **consecuencias** (buenas y malas) que puede ocasionar su uso.

Patrones Craig Larman o patrones GRAPS: **Experto** en información, **creador**, **controlador**, **bajo acoplamiento**, **alta cohesión**, **polimorfismo**, **fabricación pura**, ...

### - EXPERTO EN INFORMACIÓN -

**Problema**: Complejidad en la búsqueda de información y acoplamientos fuertes entre clases en estas búsquedas.

**Solución**: Asignar responsabilidad a la clase que contiene la información necesaria para llevar a cabo la responsabilidad.

**Consecuencias**:

**Malas**: a veces va en contra de los principios de acoplamiento o cohesión

**Buenas**: mantiene el ocultamiento de la información y distribuye el comportamiento

### - CREADOR -

**Problema**: tener acoplamientos, mala encapsulación y reutilización, y poca claridad en la construcción de objetos.

**Solución**: asignar a la clase B la responsabilidad de crear una instancia de A en casos como B agrega objetos de A, B contiene objetos de A, o B utiliza objeto de A, entre otros.

**Consecuencias**:

**Malas**: no es conveniente usarlo cuando construyamos a partir de instancias que ya existen

**Buenas**: produce bajo acoplamiento

#### - BAJO ACOPLAMIENTO -

**Problema:** elementos que dependen de demasiados elementos. Una modificación conlleva demasiadas modificaciones colaterales, difíciles de entender aisladamente y difíciles de reutilizar.

**Solución:** Asignar responsabilidades de forma que tengamos elementos (clases, subsistemas, ...) que dependan justo de los elementos que necesite.

**Consecuencias:**

**Malas:** Llevado a un extremo puede ocasionar diseños pobres, en un conjunto de clases debe existir un nivel de acoplamiento adecuado.

**Buenas:** no afectan los cambios en otros objetos, fáciles de entender de manera aislada, aumento de la reutilización.

#### - ALTA COHESIÓN -

**Problema:** elementos con pocas tareas o con muchas pero no relacionadas, difíciles de entender, reutilizar y mantener, y se ven afectados por continuos cambios.

**Solución:** Asignar responsabilidades de forma que las tareas de un elemento estén para lograr un objetivo común.

**Consecuencias:**

**Malas:** ninguna, renunciar a la alta cohesión tan solo cuando esté muy justificado

**Buenas:** claridad y fácil entendimiento, simplificación del mantenimiento y mejoras, aumento de la reutilización.

#### - CONTROLADOR O FACHADA -

**Problema:** Comunicación entre los objetos de la capa del dominio de la solución y la capa de la interfaz.

**Solución:** Asignar responsabilidades de recibir o manejar un mensaje de evento del sistema a una clase que representa al sistema global o al caso de uso.

**Consecuencias:**

**Malas:** Controladores saturados

**Buenas:** se asegura que la lógica no se maneja en la interfaz, buena reutilización, bajo nivel de acoplamiento, y posibilidad de razonar sobre el estado de los casos de uso

#### - ELABORACIÓN DEL MODELO DE INTERACCIÓN -

- Las **bases** principales son los **contratos** y el **modelo conceptual**.
- El **modelo conceptual** nos dice qué **objetos** pueden interaccionar en una operación
- Lo especificado en el **contrato** tiene que ser satisfecho en el **diagrama de comunicación**
- Nos ayudamos de los **patrones de Craig Larman**

(Ver ejercicios de elaboración del modelo)