



escuela técnica superior
de ingeniería informática

Desarrollo de Frameworks

*Departamento de
Lenguajes y Sistemas Informáticos*

**Evolución y gestión de la configuración
4º Grado en Ingeniería Informática -
Ingeniería del Software**

UNIVERSIDAD DE SEVILLA

El principio "Hollywood"

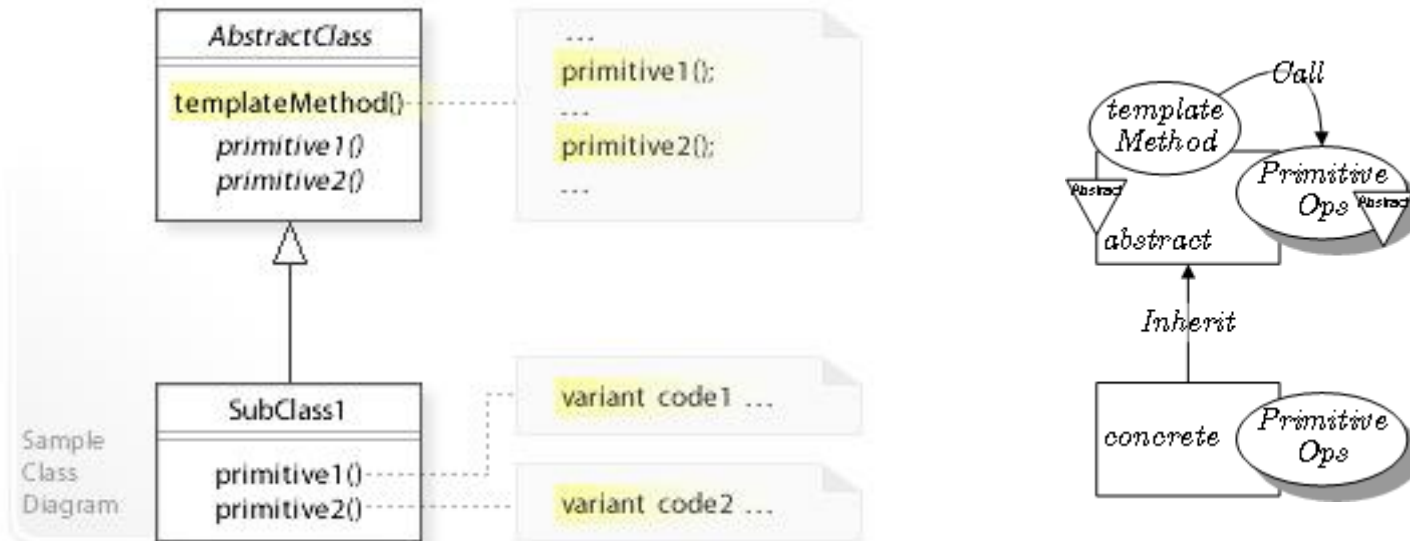
"Don't call us, we'll call you."

```
public class ServerFacade {  
    public <K, V> V respondToRequest(K request, DAO dao) {  
        return dao.getData(request);  
    }  
}
```

Larman, C (2001), *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process* (2nd ed.), [Prentice Hall](#), [ISBN 978-0-13-092569-5](#)

[Gamma, Erich](#); [Helm, Richard](#); [Johnson, Ralph](#); [Vlissides, John](#) (1994). [Design Patterns](#). Addison-Wesley. [ISBN 0-201-63361-2](#).

El patrón plantilla



¿Qué elementos tiene un framework?

- **RANURAS O HOT SPOTS:** Puntos en los que variarán las diferentes aplicaciones que creemos en el mismo dominio del problema.
- **METODOS DE ANCLAJE o HOOK METHODS:** constituyen las operaciones a las que el usuario tendrá acceso para la utilización de las clases concretas aportadas por el framework.
- **PUNTOS CONGELADOS o CONTRATOS DEL FRAMEWORK o FROZEN SPOTS:** Elementos inmutables que pertenecen a la estructura interna o esqueleto del marco.

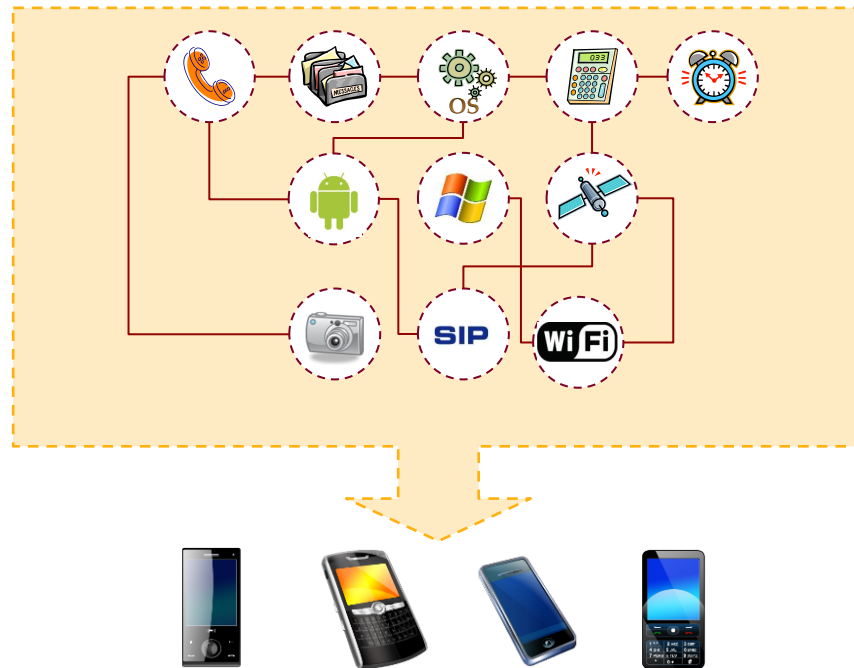
Frameworks de caja gris

- En esta práctica vamos a ver como desarrollar un framework de caja gris para el análisis de modelos de características
 - Caja negra “=” Sistema de plugins
 - Caja blanca “=” Herencia
 - Caja gris “=” Mezcla de los dos.

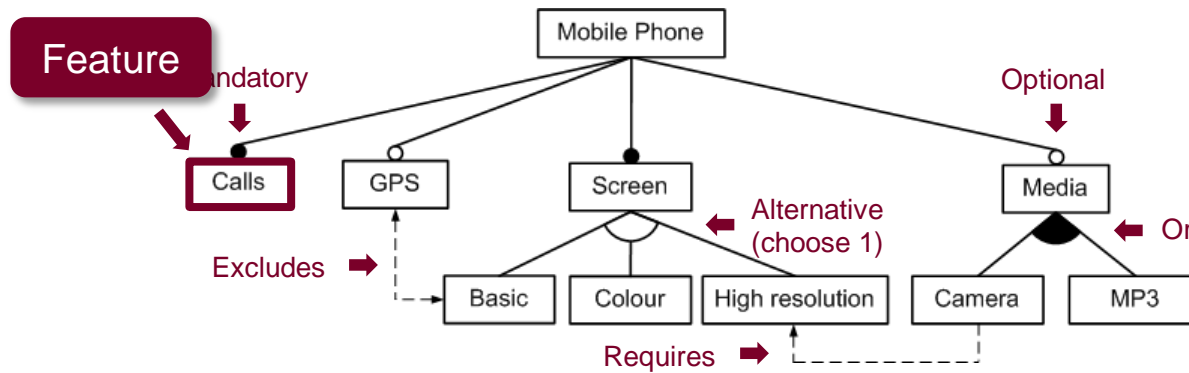
Iteración 1: Análisis

- Identificar frozen spots
- Abstracciones del dominio

Variability Models

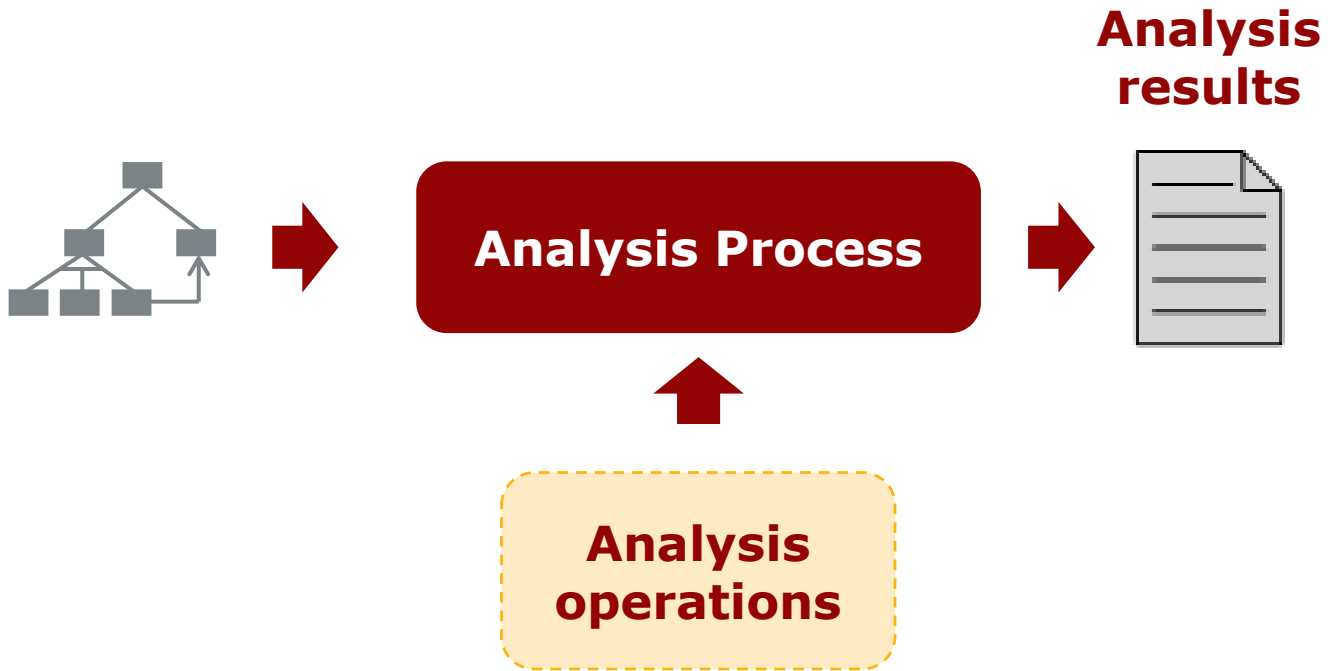


Feature Models

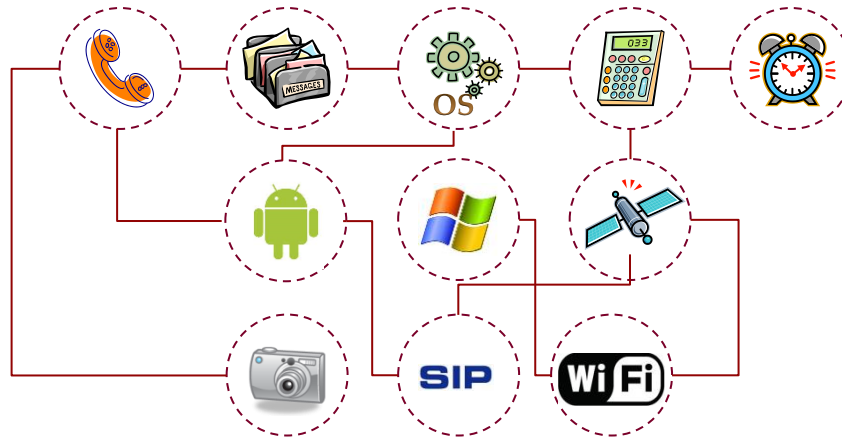


* Nosotros no implementaremos las requires ni las excludes

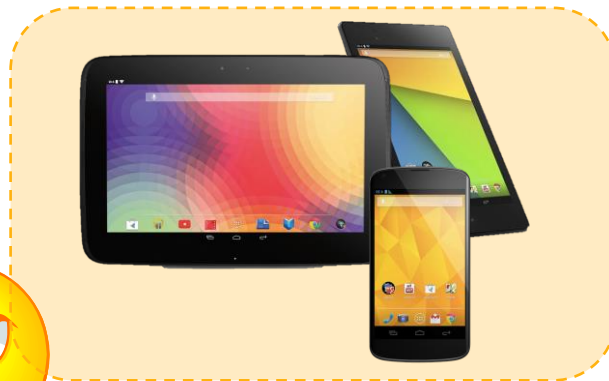
Automated Analysis



Analysis Operations



**My app requires
wifi and bluetooth 4**



**In how many different
devices will it work?**

Iteración 1: Diseño

- Core
 - Clases para representar el dominio
 - Clases para cargar plugins
- Plugins
 - Operaciones sobre el modelo
 - Contar cuantas features no tienen hijos en el modelo

Clases para implementar el dominio

Feature
Relation

- Mandatory Relation
- Optional Relation
- Set Relation
- Or Relation

Feature

name

relations = list[Relation]

Relation

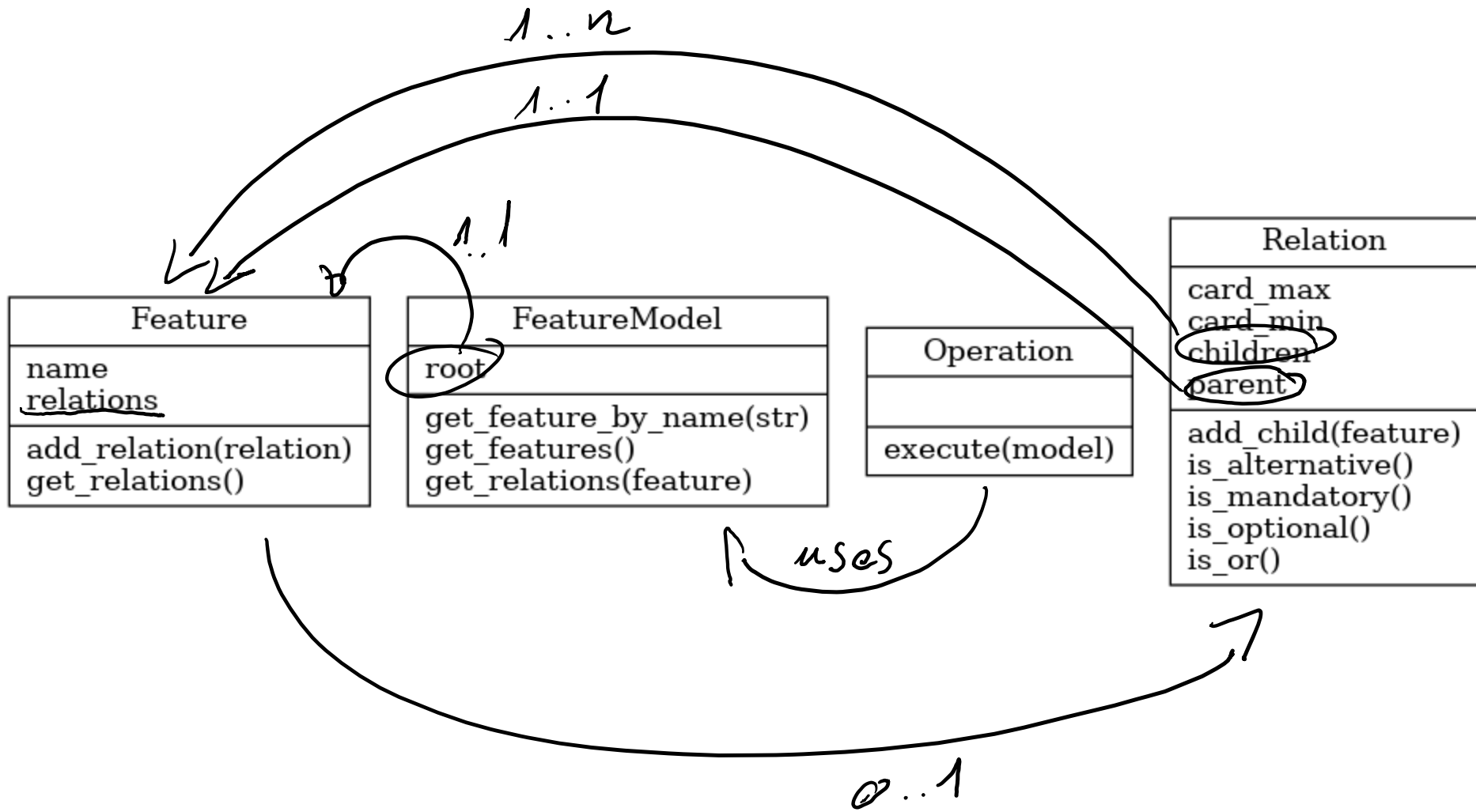
parent<Feature>

childs = list[Feature]

int min_childs

int max_child

UML



Iteración 1: Instanciación

- Contador de hojas de un modelo
 - Crear un main.py que consuma el plugin y muestre un modelo previamente cargado en memoria



Iteración 2: Analisis

Nos surge un nuevo cliente que nos pide desarrollar un nuevo producto pero en este caso está interesado en conocer cuantas características tiene un modelo.



Iteración 2: Diseño

- Opción 1: Creamos la clase en el core (será común a todos los productos del portfolio de posibilidades)
- Opción 2: Extendemos el sistema e plugins añadiendo uno nuevo para contar las características

Opción 1: Creamos un nuevo plugin

Promises of Frameworks

Más
abstracción

Evita la
redundancia

Delimita mejor
los problemas

Usa elementos
del dominio
para programar

Podríamos preguntarnos lo siguiente

- ¿Qué clases podríamos considerar parte de un framework de caja blanca?
- ¿Qué clases podríamos considerar parte de un framework de caja negra?
- Este framework, a partir de cuantos puntos de variabilidad (aquellos puntos de extensión) consideráis que sería rentable.
- ¿Si usamos venv... en que nos ayuda?