# Importancia de las métricas

Las métricas permiten evaluar la calidad del producto de software en diferentes aspectos.  
El obtener las métricas puede estimular la realización de actividades en futuras fases del desarrollo, como pueden ser:  
- Asociación con atributos de calidad  
- Refactoring  
- Aplicar patrones de diseño  
- Estimación de esfuerzo de testing

# Categorías

Se podrían dividir en tres categorías:

Orientadas al paquete (assembly)  
Ejemplo: Afferent Couplings (Ca); la cual indica el número de clases fuera del paquete que dependen de clases del paquete analizado.

Orientadas a la clase   
Ejemplo: Número de subclases de una clase.

Orientadas a los métodos  
Ejemplo: Complejidad Ciclomática de McCabe; indica la complejidad del método.

# Orientadas al paquete (assembly)

## Cohesión relacional

Mide la relación entre clases en un paquete  
R = número de relaciones entre clases internas al paquete (relaciones que no conectan con clases fuera del paquete)   
N = número de clases e interfaces dentro del paquete   
  
 **H = (R+1)/N**Buena 1.5 a 4.0.   
< 1.5 o > 4.0 no muy buena

## Medición de la estabilidad

La inestabilidad de un paquete impacta en el esfuerzo requerido para realizar un cambio.

### Principio de Dependencias Estables

La dirección de las dependencias entre paquetes debe ser acorde con la estabilidad de estos.

Un paquete debe depender solamente de paquetes que son más estables que él.

**Inestabilidad I = Ce /(Ce + Ca )**

I = Grado de inestabilidad asociado a un paquete.   
Ca = Número de clases fuera del paquete que dependen de clases dentro del paquete.  
Ce = Número de clases fuera del paquete de las cuales dependen clases dentro del paquete.

Valor cercano a 0 - máxima estabilidad porque no se depende de otros paquetes. Se debe intentar EXTENDER haciéndolo abstracto.

Valor cercano a 1 - máxima inestabilidad porque depende de otros paquetes. Este paquete es fácil de cambiar debido a que impacta en pocos paquetes.

## Medición de la abstracción

La abstracción de un paquete se mide calculando el ratio entre el número de clases abstractas e interfaces del paquete, y el número total de clases en el paquete.

### Principio Abstracciones Estables

Los paquetes más estables (dependen de él y él no depende) deben tender a ser abstractos (con interfaces o clases abstractas)

**Abstracción A=Na / Nc**

A = Abstracción del paquete.   
Na = Cantidad de clases abstractas e interfaces en el paquete.   
Nc = Cantidad de clases concretas, abstractas e interfaces del paquete.

Una abstracción cercana a cero indica un paquete concreto, mientras que un valor cercano a uno indica un paquete abstracto.

## Diagrama Descripción generada automáticamenteAbstracción vs Estabilidad

Distancia a la secuencia principal  
D = |A + I -1|/SQRT(2)   
D’ = |A + I -1|  
D’ esta entre [0, 1]   
Cuando el valor de D’ es más cercano a 0 más cerca de la secuencia principal se encuentra el paquete.

Valores cercanos a 0 (zona de dolor) indican que el paquete es concreto y estable (responsable). Estos paquetes no son buenos porque no son extensibles y si cambian impactan en otros.

Valores cercanos a 1 (zona de poca utilidad) indican que el paquete es abstracto e inestable. El paquete es extensible, pero tiene pocos paquetes que dependan de él.

# Orientadas a Clase

## Tamaño de clase

Evalúa la facilidad de comprensión del código.

Posibles métricas :   
LOC de un método   
Cantidad de variables de instancia.  
Contar la cantidad de métodos.

## Complejidad de la clase

### Weighted Methods per Class (WMC)

Cuenta los métodos implementados en una clase.  
Cuanto más grande es el número de métodos en una clase mayor es el impacto en sus subclases.

### Depth of Inheritance Tree (DIT)

Es la máxima longitud desde un nodo a la raíz de la jerarquía.   
Cuanto más lejos de la raíz se encuentra la clase hereda mayor número de métodos haciendo la clase más compleja.   
El valor es deseable que sea menor que 6.

### Response for a Class (RFC)

“Número de métodos locales” + “Número de métodos remotos”. Número de métodos locales incluye métodos heredados.

### Number Of Members (NOM)

Cuenta la cantidad de miembros de una clase, incluyendo atributos, operaciones, etc. Se puede o no incluir miembros heredados.