



# Conceptos de Algoritmos Datos y Programas



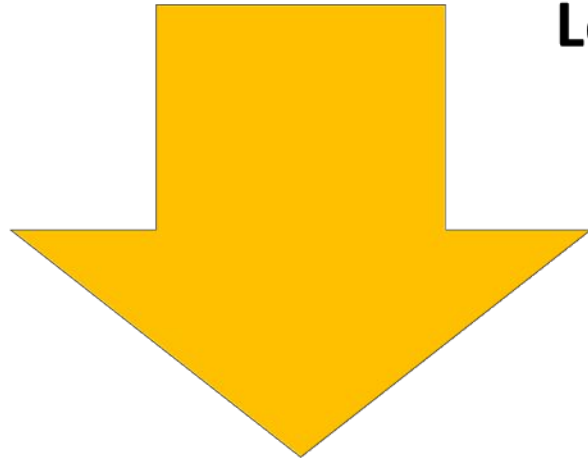
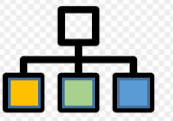
# CADP – TEMAS



● Modularización

● EJERCICIO – PREGUNTAS FINALES

# CADP – MODULARIZACION

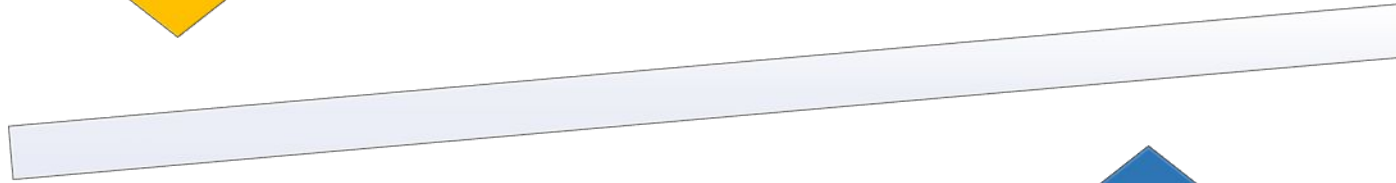


**Los problemas del mundo real implican:**

**Complejidad**

**Extensión**

**Modificaciones**

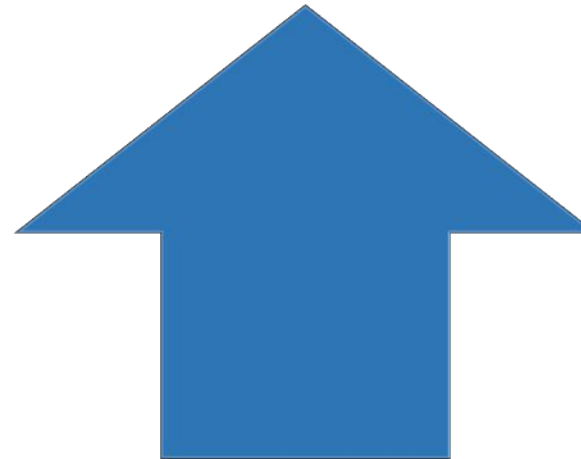


**Los tratamos de resolver con:**

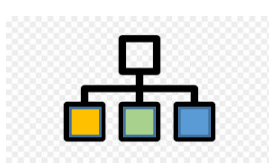
**Abstracción**

**Descomposición**

**Independencia Funcional**



# CADP – MODULARIZACION



## MODULARIZAR

Significa dividir un problema en partes funcionalmente independientes, que encapsulen operaciones y datos.

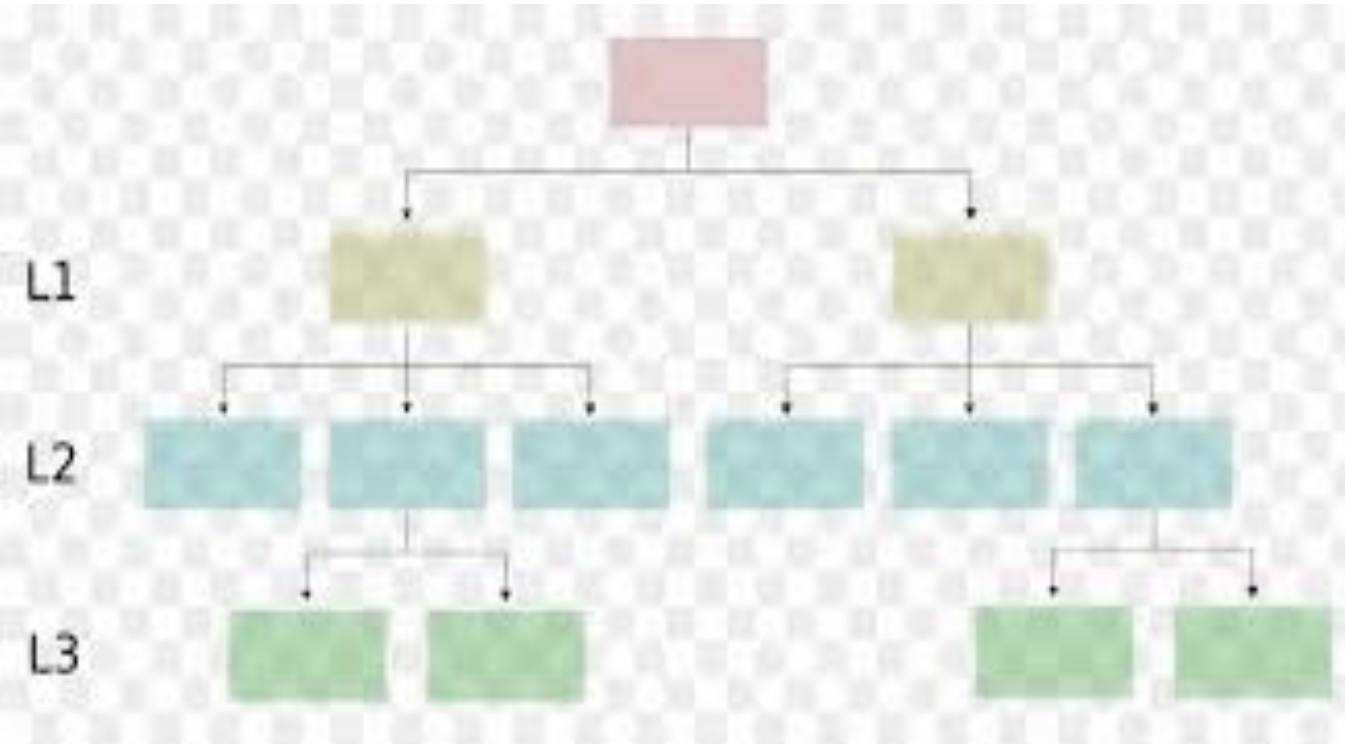
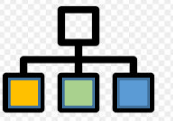


No se trata simplemente de subdividir el código de un sistema de software en bloques con un número de instrucciones dado.



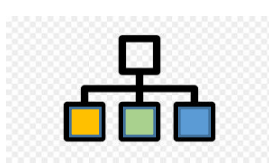
Separar en funciones lógicas con datos propios y datos de comunicación perfectamente especificados.

# CADP – MODULARIZACION



- Cada subproblema está en un mismo nivel de detalle.
- Cada subproblema puede resolverse independientemente.
- Las soluciones de los subproblemas puede combinarse para resolver el problema original.

# CADP – MODULARIZACION



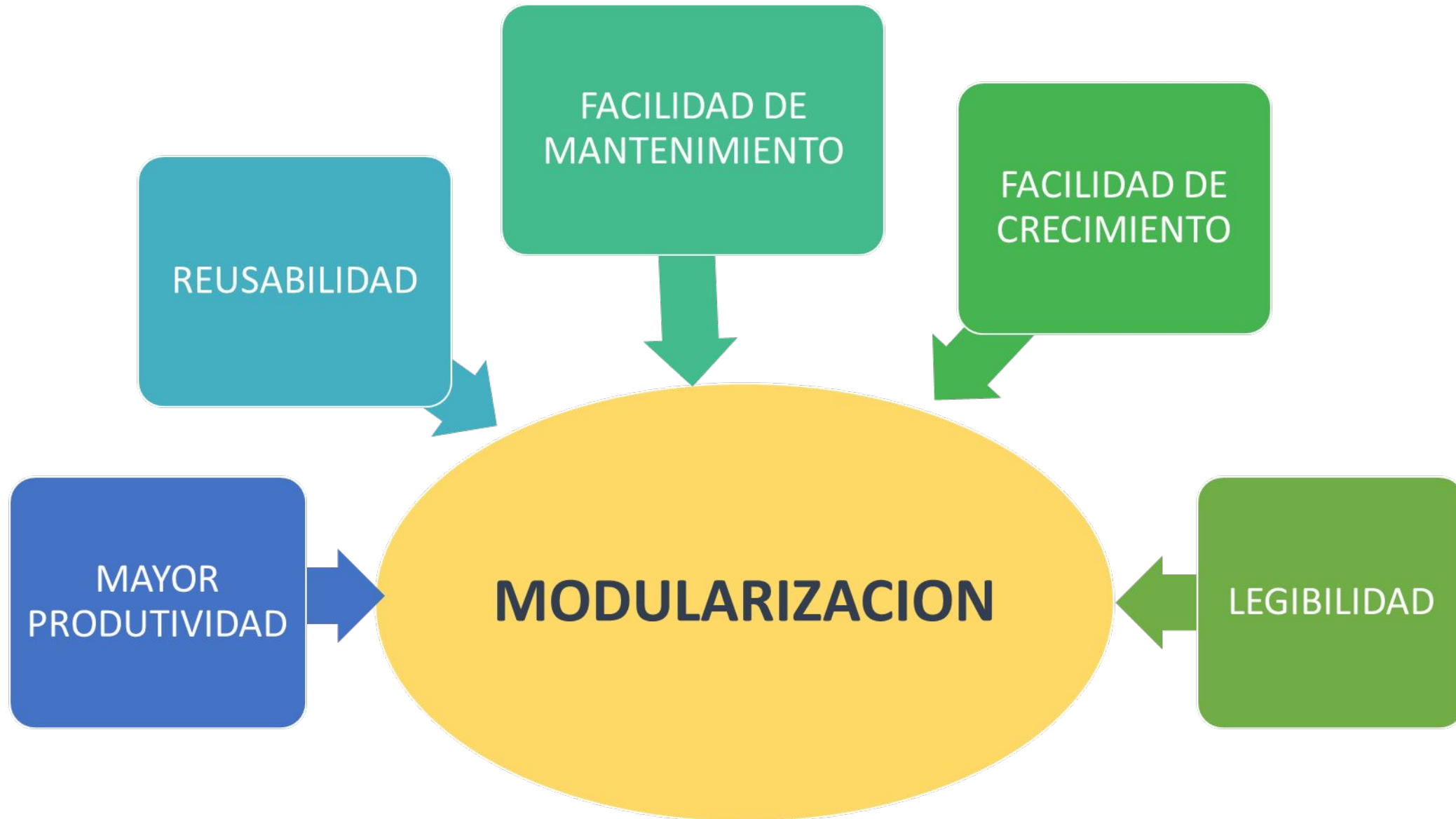
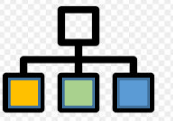
## MODULO

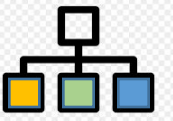
Tarea específica bien definida se comunican entre sí adecuadamente y cooperan para conseguir un objetivo común.

Encapsula acciones tareas o funciones.

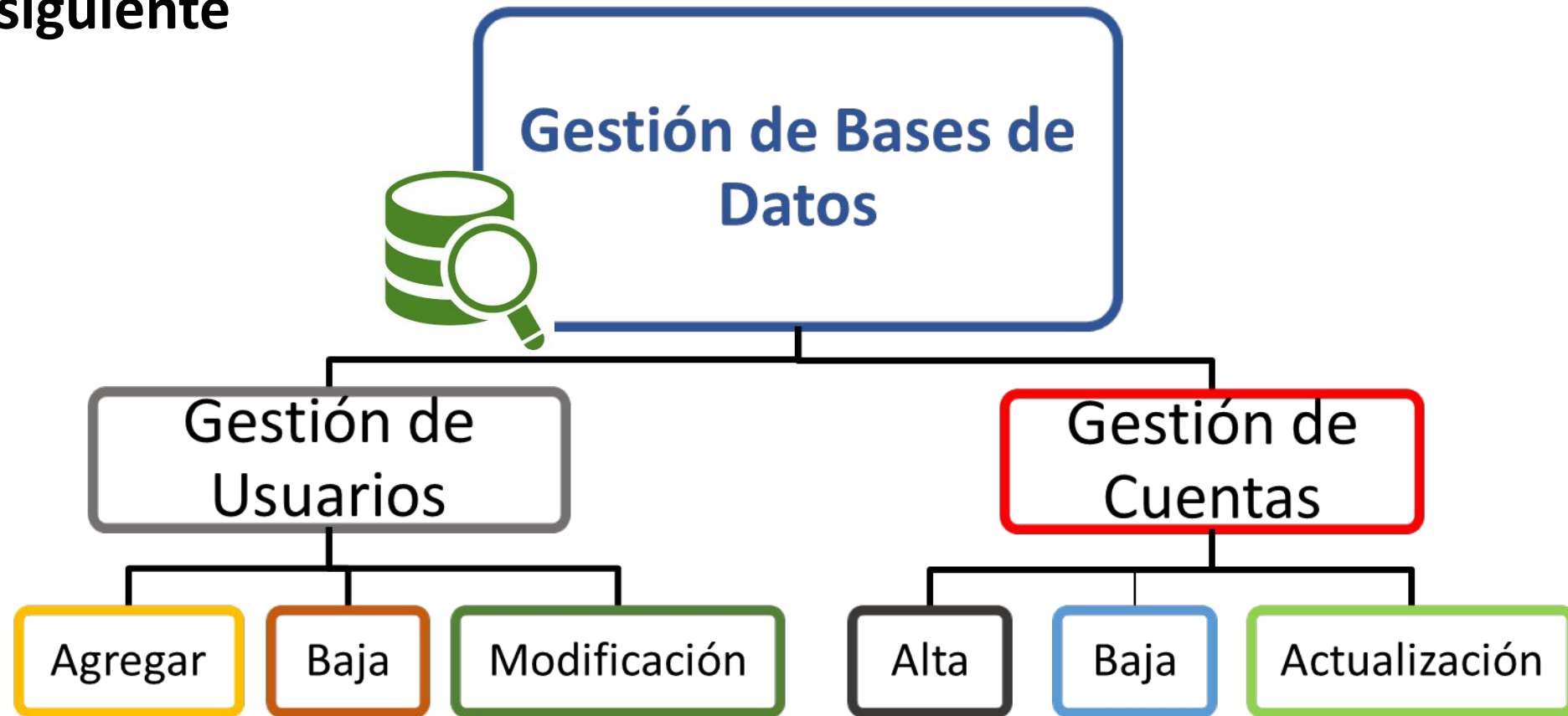
En ellos se pueden representar los objetivos relevantes del problema a resolver.

Existen diferentes metodologías para usarlos en los programas en particular nosotros usaremos la **METODOLOGIA TOP-DOWN**

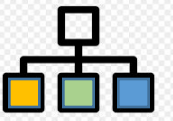




Supongamos que tenemos que resolver el siguiente proyecto

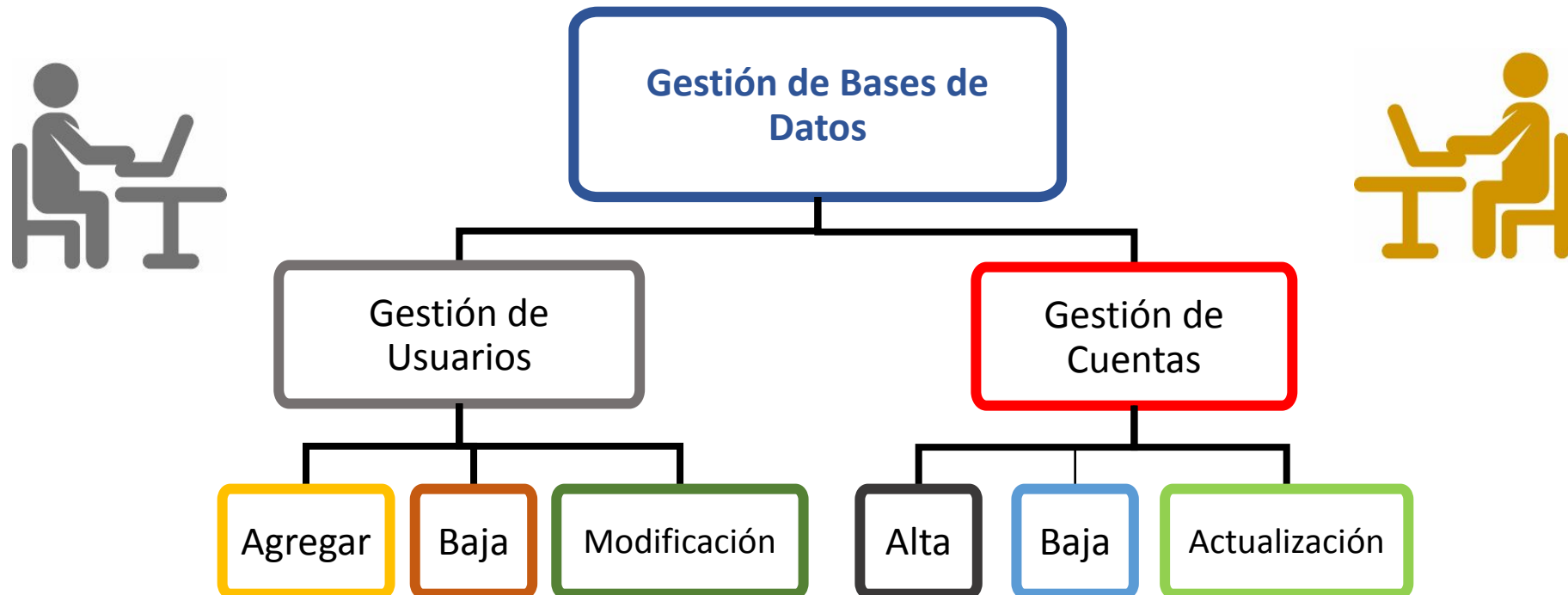


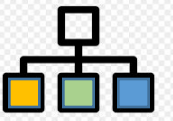




Al dividir un sistema de software en módulos funcionalmente independientes, un equipo de desarrollo puede trabajar simultáneamente en varios módulos, incrementando la productividad (es decir reduciendo el tiempo de desarrollo global del sistema).

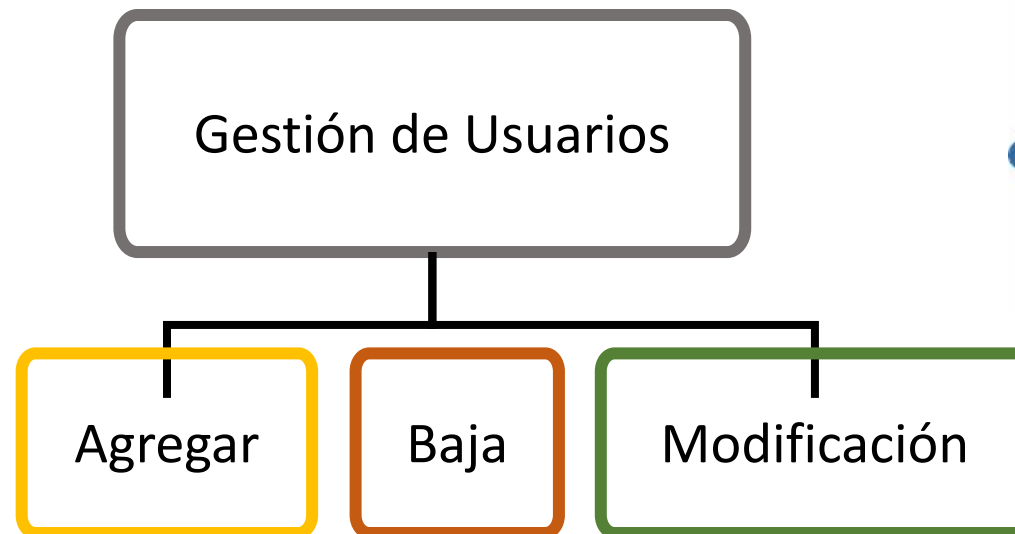
**MAYOR  
PRODUCTIVIDAD**

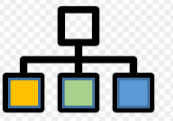




Un objetivo fundamental de la Ingeniería de Software es la reusabilidad, es decir la posibilidad de utilizar repetidamente el producto de software desarrollado. Naturalmente la descomposición funcional que ofrece la modularización favorece el reuso.

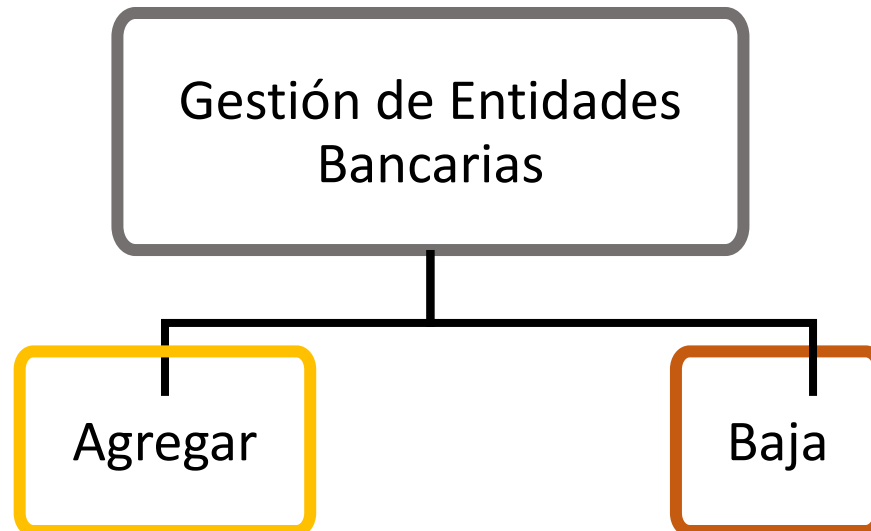
## REUSABILIDAD

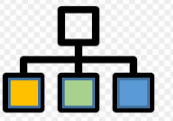




Los sistemas de software reales crecen (es decir aparecen con el tiempo nuevos requerimientos del usuario). La modularización permite disminuir los riesgos y costos de incorporar nuevas prestaciones a un sistema en funcionamiento.

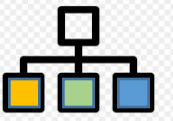
## FACILIDAD DE CRECIMIENTO





**Un efecto de la modularización es una mayor claridad para leer y comprender el código fuente. El ser humano maneja y comprende con mayor facilidad un número limitado de instrucciones directamente relacionadas.**

## LEGIBILIDAD



Fortran

- Subroutine

Modula

- Module

Ada, Pascal, C

- Procedure/Function

Orientado a objetos

- Class