

Principios de la Ingeniería de Software



Cada estrato se basa en los inferiores y es más susceptible a cambios con el paso del tiempo

Principios y cualidades

- En general la elección de principios y técnicas está determinada por las cualidades que se desean para el software.
- Los principios que se enunciarán están orientados a obtener sistemas confiables y evolucionables.
- El IS no sólo necesita buenos principios para obtener productos de calidad sino también técnicas, metodologías y herramientas que se apoyen sobre ellos.

Rigor y formalidad

Sólo una aproximación rigurosa puede producir productos más confiables, controlar sus costos e incrementar su confiabilidad.

La ventaja de la formalidad sobre el rigor es que la formalidad puede ser la base para la mecanización del proceso.

No hay necesidad de ser formal durante todo el desarrollo pero el IS debe saber cuándo, cómo y dónde utilizar la formalidad.

Separación de intereses

Permite lidiar con aspectos individuales del problema.

Primero se debe intentar separar los temas que no están íntimamente relacionados entre sí.

La separación se puede hacer según varios criterios:

- Tiempo (ciclo de vida del software)

- Cualidades

- Vistas (flujo de datos, de control, estático, dinámico, etc.)

- Partes (estructura, diseño)

Modularidad

Un sistema complejo debe ser dividido en partes.

Permite aplicar el principio anterior en dos fases:

 detalles de cada parte sin tomar en cuenta las restantes

 relaciones entre partes sin tomar en cuenta los detalles

No sólo aplica a los aspectos estructurales, sino a todo el proceso de desarrollo.

Se basa en: descomposición, composición y comprensión.

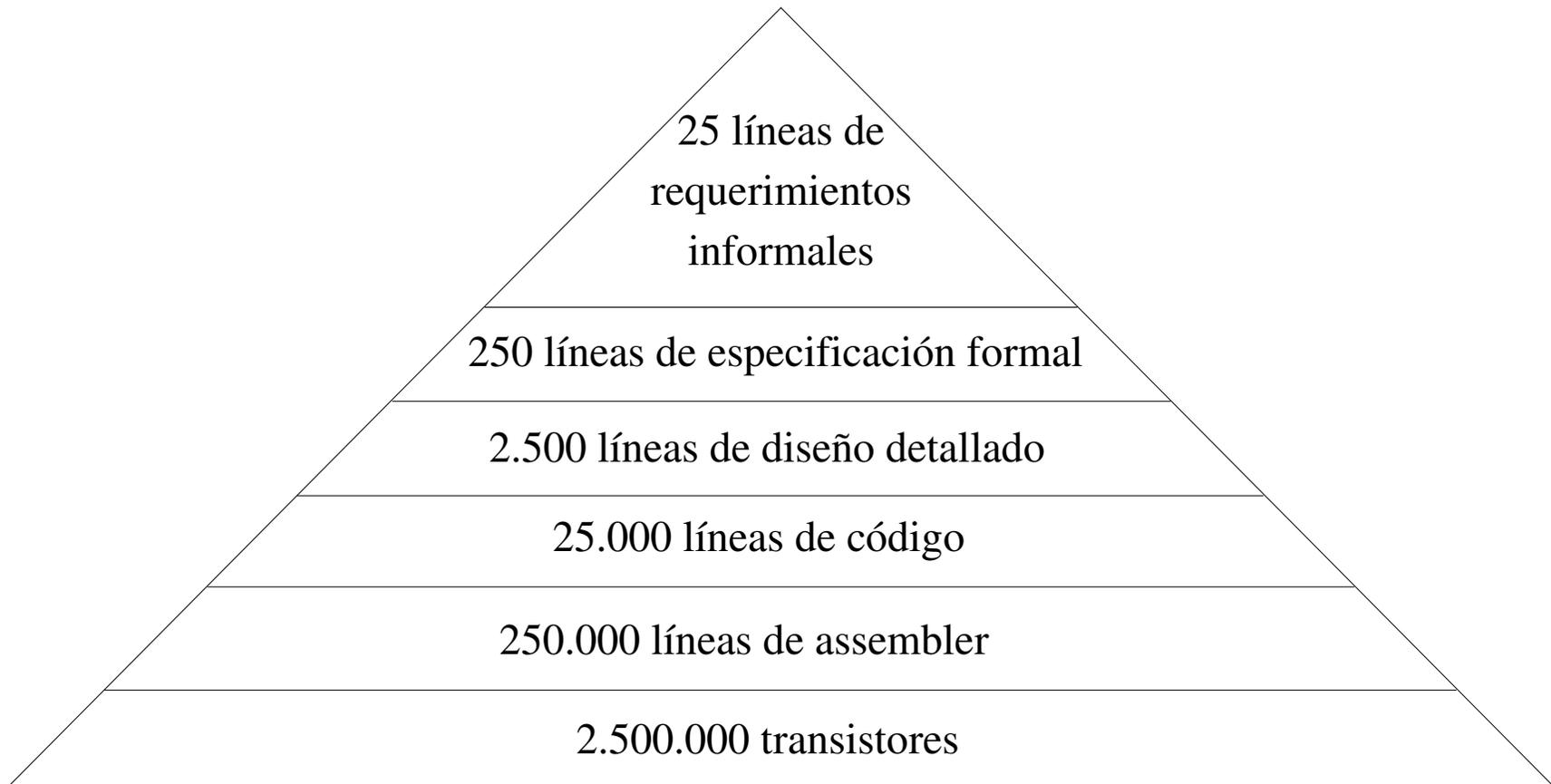
Cohesión y acoplamiento.

Abstracción

Los modelos que construimos para entender los fenómenos son abstracciones de la realidad.

El uso de modelos formales abstractos permite aproximarse mediante refinamientos a la solución final pudiendo demostrar que cada descripción verifica la anterior.

Abstracción



Anticipación del cambio

- ❑ El software sufre cambios permanentemente
- ❑ Se requiere un esfuerzo especial en las fases iniciales para anticipar cómo y dónde será probable que se den los cambios.
- ❑ Los cambios probables deben ser aislados en porciones específicas del software.

Generalidad

- Toda vez que se le pida que resuelva un problema, trate de focalizar su atención en el descubrimiento de un problema más general que puede estar oculto detrás del problema en cuestión. Puede suceder que el problema generalizado no sea más complejo. Siendo más general, la solución será más reusable.

Incrementalidad

Este principio puede aplicarse al identificar tempranamente subconjuntos útiles de una aplicación para así obtener rápido *feedback*.

Este principio surge luego que la experiencia ha demostrado concluyentemente que los requerimientos del usuario cambian a medida que se desarrolla el producto.