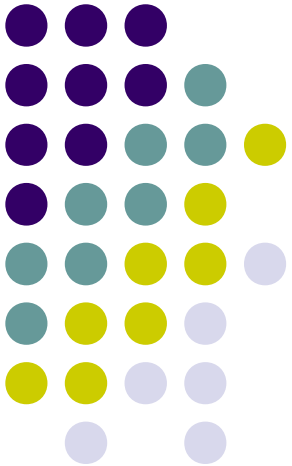
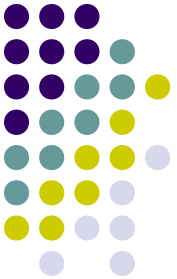
# Ingeniería de software I

## Tema III – Actividades del proceso de desarrollo

Universidad Nacional del Litoral
**FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS**

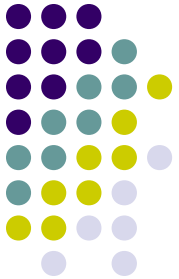


# Inicio del proyecto
## Razones para iniciarlo



- Mayor velocidad de procesamiento
- Mayor exactitud y consistencia
- Consulta más rápida a la información
- Integración de las áreas de la organización
- Reducción de costos
- Mayor seguridad

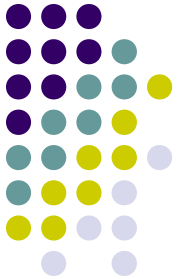
# Inicio del proyecto
## Origen de solicitudes de proyectos



- Directores, Gerentes, Jefes de alto rango
- Altos ejecutivos, Autoridades con cargos políticos (para las reparticiones públicas)
- Analistas de Sistemas, Sectoriales de Informática, Direcciones de Informática
- Entes externos (normalmente estatales)

Internos de la organización

# Inicio del proyecto
## Revisión y selección de proyectos

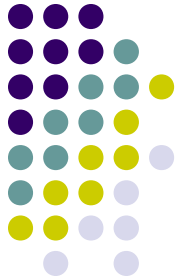


- **Grupo multidisciplinario:**
  - **Grupo directivo:** personal jerárquico de la organización (eventualmente autoridades políticas),
  - **Grupo de usuarios:** personal especialista en el tema tocado por la solicitud de sistemas y que se verían afectados directamente por la solución adoptada,
  - **Grupo de sistemas de información:** personal especialista en sistemas de información.

# Actividades del proceso de desarrollo

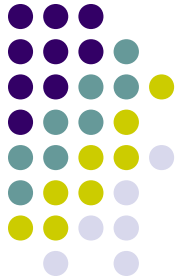
1. **Especificación del software**. En donde se define tanto la funcionalidad del software como las restricciones de su operación.
2. **Diseño e implementación**. Debe desarrollarse el software para cumplir con las especificaciones.
3. **Validación del software**. Verificación de que el software cumple con lo que el cliente quiere.
4. **Evolución del software**. El software tiene que evolucionar para satisfacer las necesidades cambiantes del cliente.

# 1. Requerimientos y Especificación del software



***La especificación del software o la ingeniería de requerimiento, consiste en el proceso de comprender y definir qué servicios se requieren del sistema así como la identificación de restricciones sobre la operación y el desarrollo. Es una etapa particularmente crítica del proceso de software ya que los errores aquí, conducen de manera inevitable a problemas posteriores tanto en el diseño como en la implementación del sistema y pueden concluir con el fracaso del proyecto.***

# 1. Requerimientos y Especificación del software



- Requiere mucha interacción con el usuario. Los tipos de usuarios son:
  - **Usuarios directos:** los que realmente interactúan con el sistema. Ingresan datos o reciben salidas (consultas).
  - **Usuarios indirectos:** se benefician de los resultados o informes producidos por el sistema, pero no interactúan directamente con el hardware o software.
  - **Usuarios administrativos:** administran los sistemas de aplicación. Tienen la autoridad para aprobar o desaprobar la inversión en el desarrollo de aplicaciones. Tienen además la responsabilidad de la organización para la efectividad de los sistemas. Estos usuarios participan activamente en el desarrollo del sistema.

# 1. Requerimientos y Especificación del software

## Actividades

- Investigación preliminar
- Estudio de factibilidad
  - Operativa
  - Técnica
  - Económico financiera
- Obtención y análisis de requerimientos
  - Entrevistas
  - Cuestionarios
  - Revisión de registros
  - Observación
- Especificación de requerimientos
- Validación de requerimientos

# 1. Requerimientos y Especificación del software

## Modelos

Aspectos:

- Perspectiva **externa** (modelo de contexto o entorno del sistema)
- Perspectiva de **interacción** donde se modela la interacción entre un sistema y su entorno o entre los componentes de un sistema
- Perspectiva **estructural** donde se modela la organización de un sistema o la estructura de datos que procese el sistema
- Perspectiva de **comportamiento** donde se modela el comportamiento dinámico del sistema y cómo responde ante ciertos eventos.

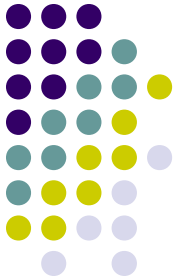
# 1. Requerimientos y Especificación del software

## Modelos

Objetivos:

- Facilitar la discusión sobre un sistema existente o propuesto.
- Documentar el sistema (planos del sistema)
- Descripción detallada del sistema que sirve para generar una implementación del sistema.

# 2. Diseño e implementación del software

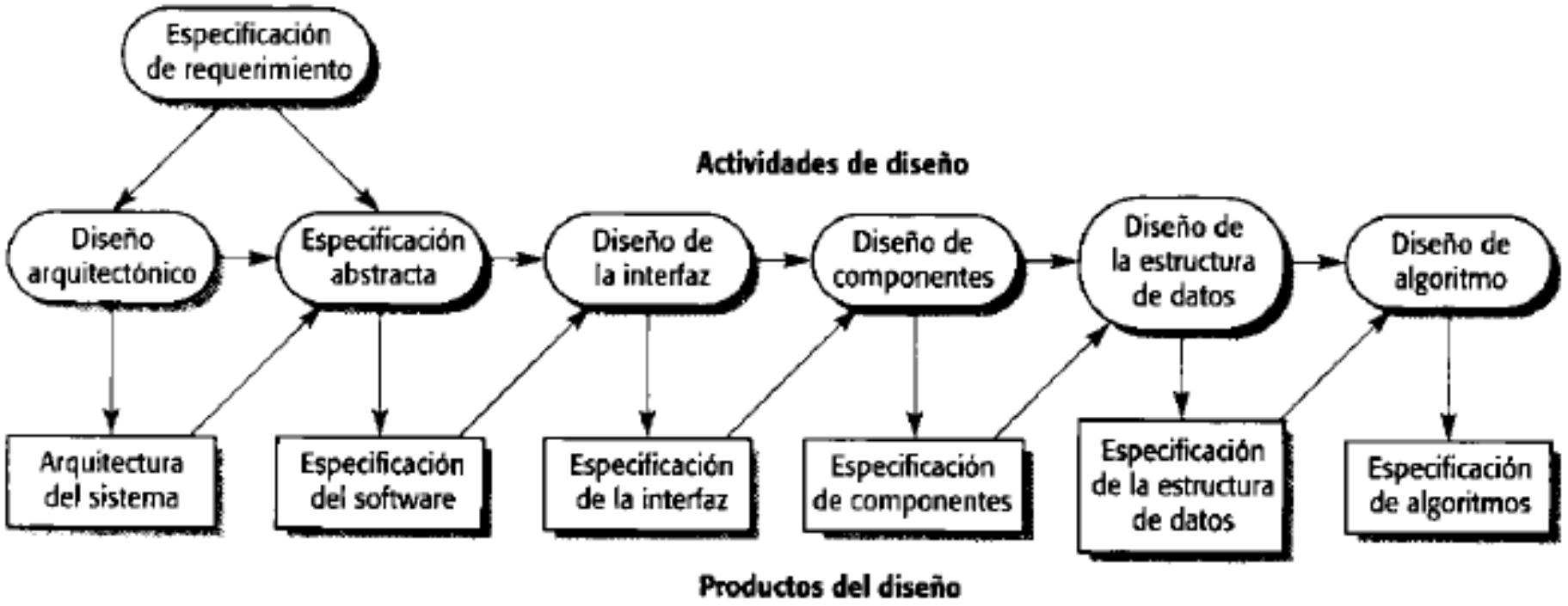


- Objetivo: Convertir una especificación del sistema en un artefacto ejecutable

- Un diseño de software es una descripción de la estructura del software que se va a implementar, los modelos y las estructuras de datos, las interfaces entre componentes y los algoritmos usados.

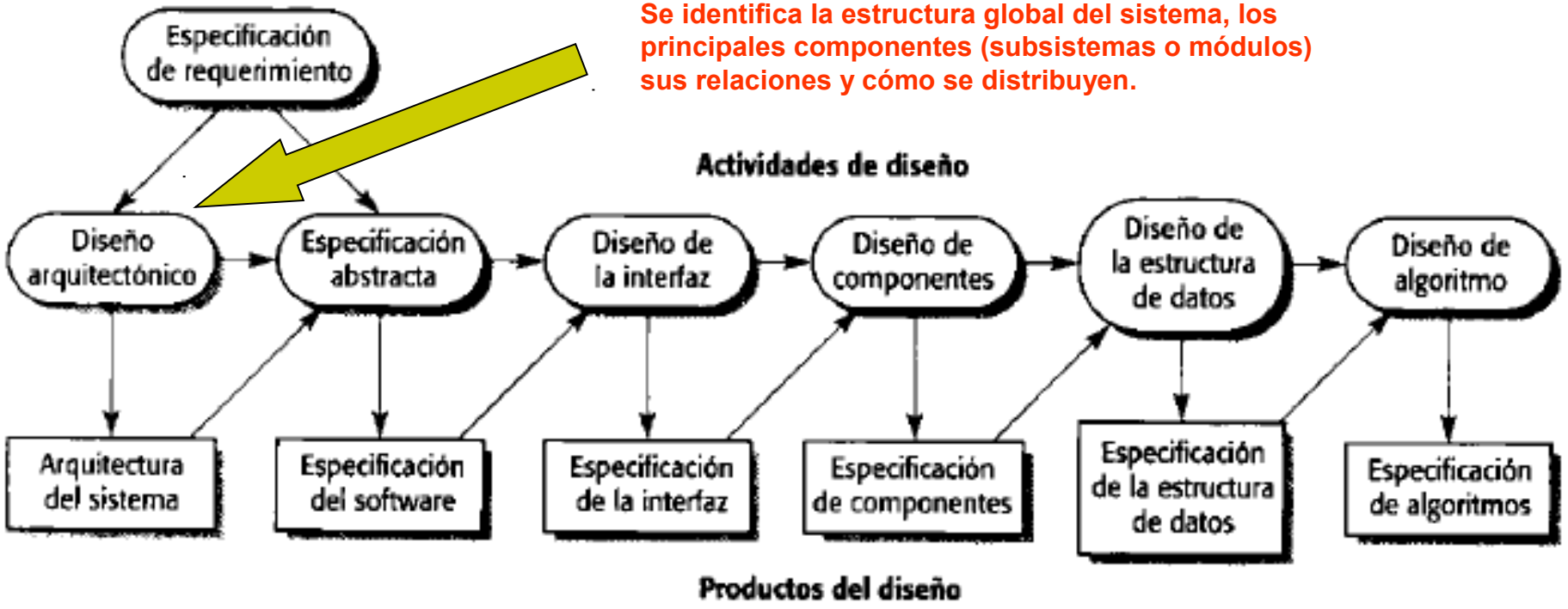
# 2. Diseño e implementación del software

## Actividades y productos de diseño



# 2. Diseño e implementación del software

## Actividades y productos de diseño

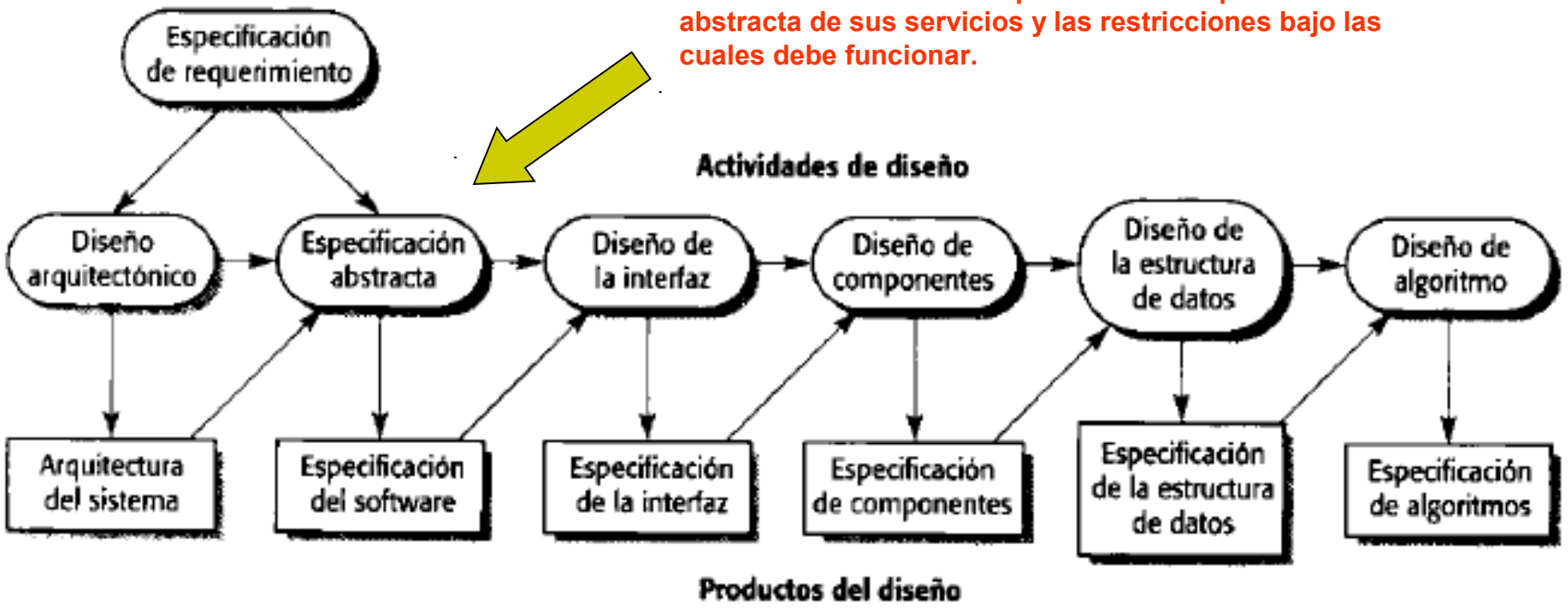



**Se identifica la estructura global del sistema, los principales componentes (subsistemas o módulos) sus relaciones y cómo se distribuyen.**

# 2. Diseño e implementación del software

## Actividades y productos de diseño

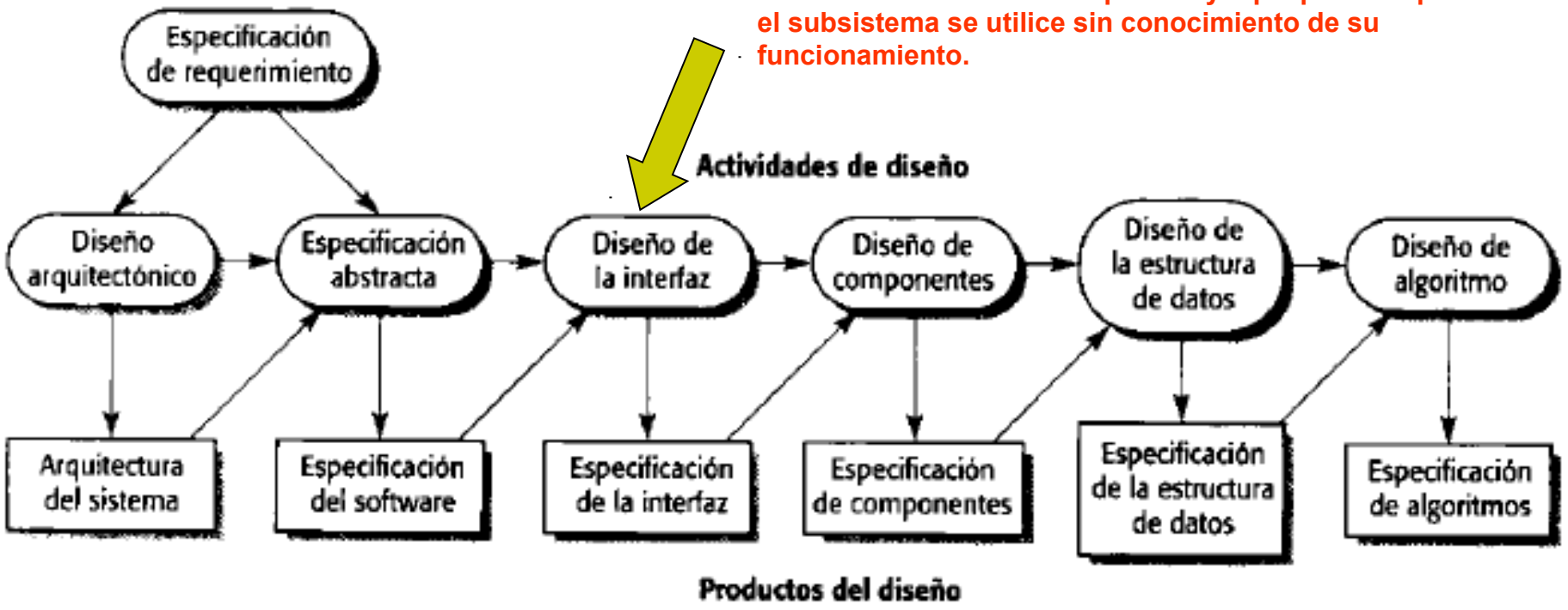
**Para cada subsistema se produce una especificación abstracta de sus servicios y las restricciones bajo las cuales debe funcionar.**



# 2. Diseño e implementación del software

## Actividades y productos de diseño

Para cada subsistema se diseña y documenta su interfaz con otros subsistemas. Esta especificación de la interfaz debe ser inequívoca ya que permite que el subsistema se utilice sin conocimiento de su funcionamiento.



# 2. Diseño e implementación del software

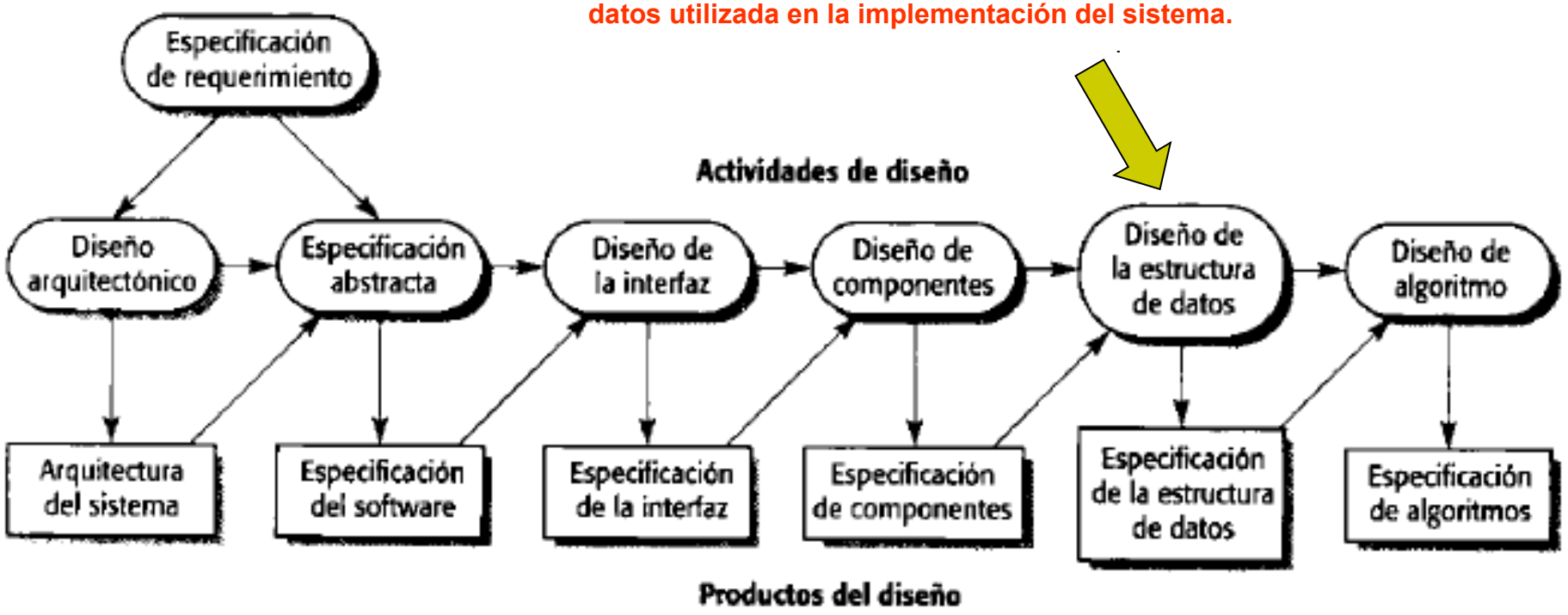
## Actividades y productos de diseño

Se toma cada componente del sistema y se diseña cómo funcionará. Esto puede ser un simple dato de la funcionalidad que se espera implementar, y al programador se le deja el diseño específico.

# 2. Diseño e implementación del software

## Actividades y productos de diseño

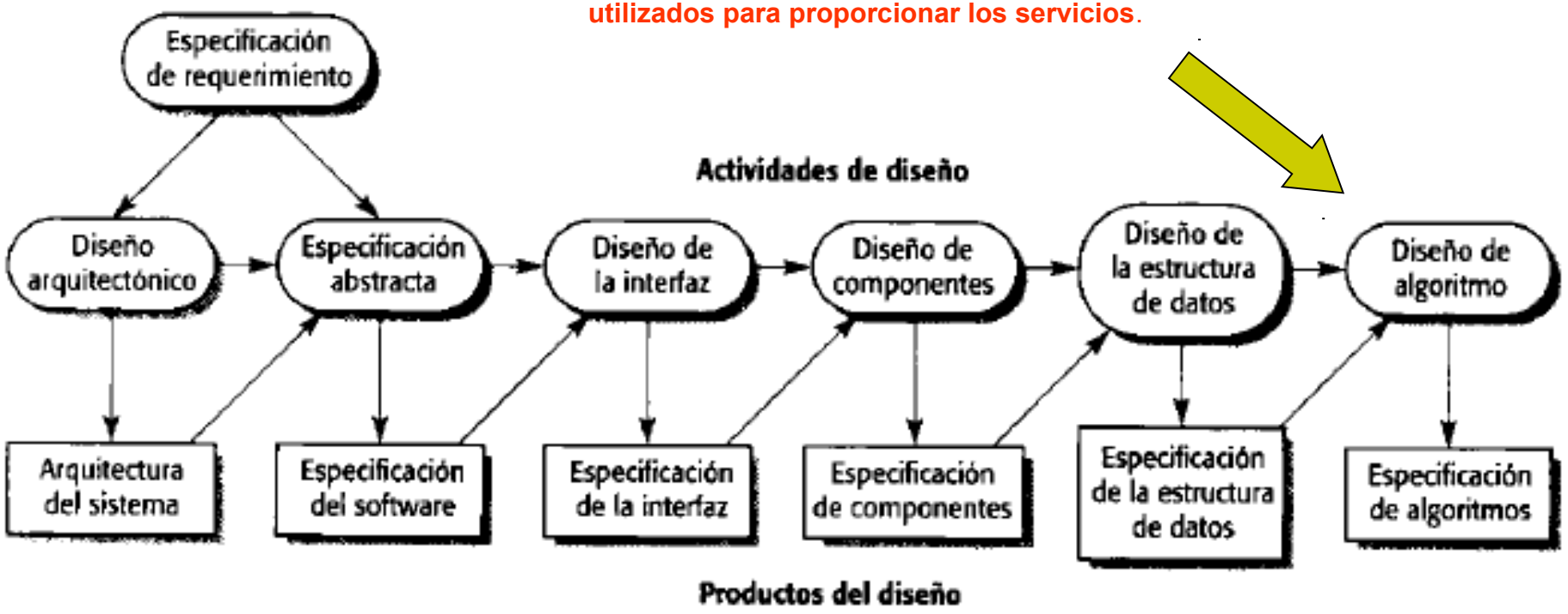
**Se diseña en detalle y especifica la estructura de datos utilizada en la implementación del sistema.**



# 2. Diseño e implementación del software

## Actividades y productos de diseño

**Se diseñan en detalle y especifican los algoritmos utilizados para proporcionar los servicios.**



# 2. Diseño e implementación del software

## Actividades y productos de diseño

→→ PROGRAMACIÓN →→

# 2. Diseño e implementación del software

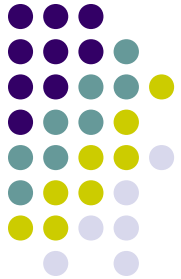
## Actividades y productos de diseño

## PROGRAMACIÓN

- No hay proceso general, es una actividad personal
- Comenzar con los componentes que se entienden.
- Desarrollar a lo último los que se entienden
- Comenzar con la definición de los datos de prueba

- **Prueba**: establece la existencia de defectos
- **Depuración**: localiza defectos y los corrige

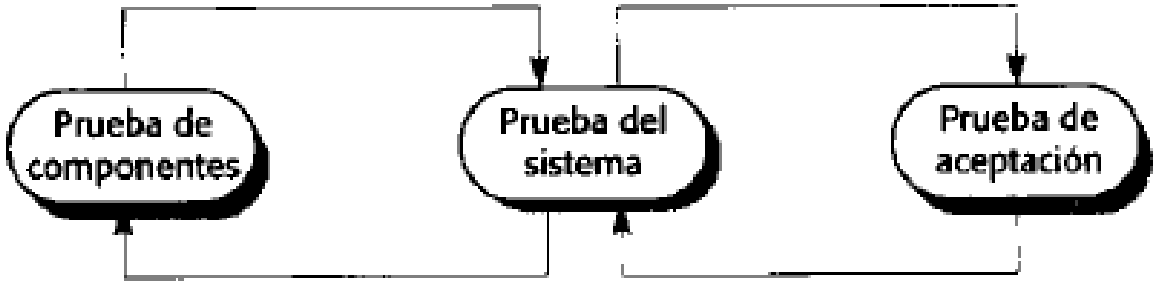
# 3. Validación del software



- La verificación y validación del software se crea para mostrar que un sistema cumple tanto con sus especificaciones como las expectativas del cliente.
- Las pruebas con datos simulados son la principal técnica de validación.
- No se prueban unidades monolíticas.

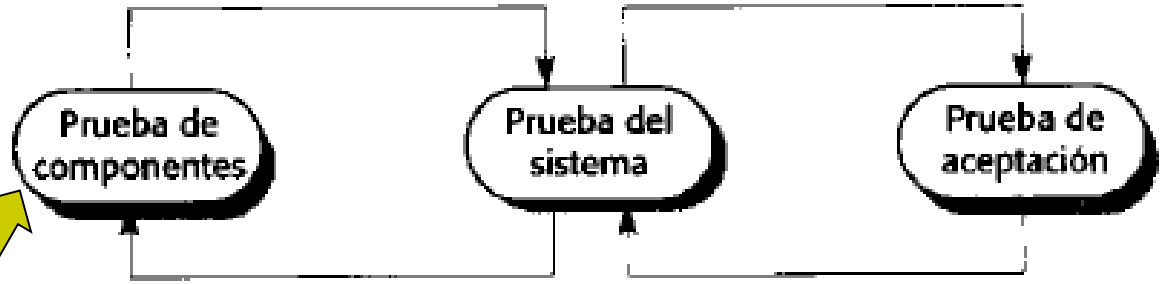
# 3. Validación del software

- La verificación y validación del software se crea para mostrar que un sistema cumple tanto con sus especificaciones como las expectativas del cliente.
- Las pruebas con datos simulados son la principal técnica de validación.
- No se prueban unidades monolíticas sino:



# 3. Validación del software

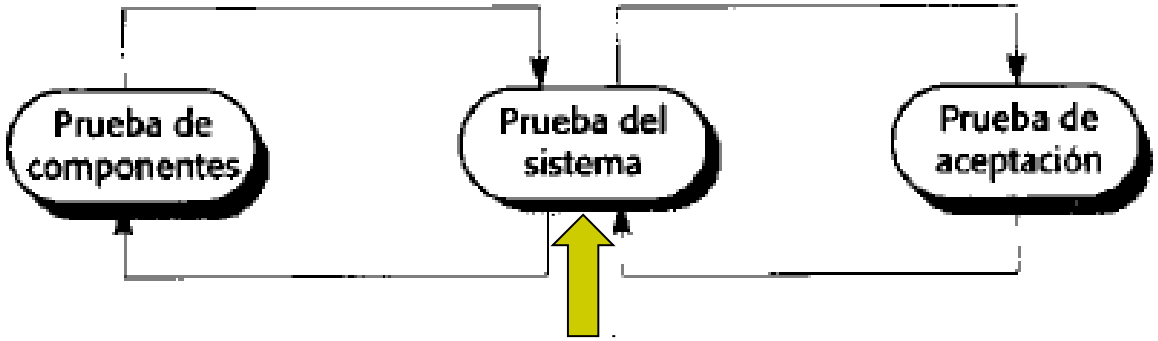
- La verificación y validación del software se crea para mostrar que un sistema cumple tanto con sus especificaciones como las expectativas del cliente.
- Las pruebas con datos simulados son la principal técnica de validación.
- No se prueban unidades monolíticas sino:




**Prueba de componentes aislados o de desarrollo**

# 3. Validación del software

- La verificación y validación del software se crea para mostrar que un sistema cumple tanto con sus especificaciones como las expectativas del cliente.
- Las pruebas con datos simulados son la principal técnica de validación.
- No se prueban unidades monolíticas sino:

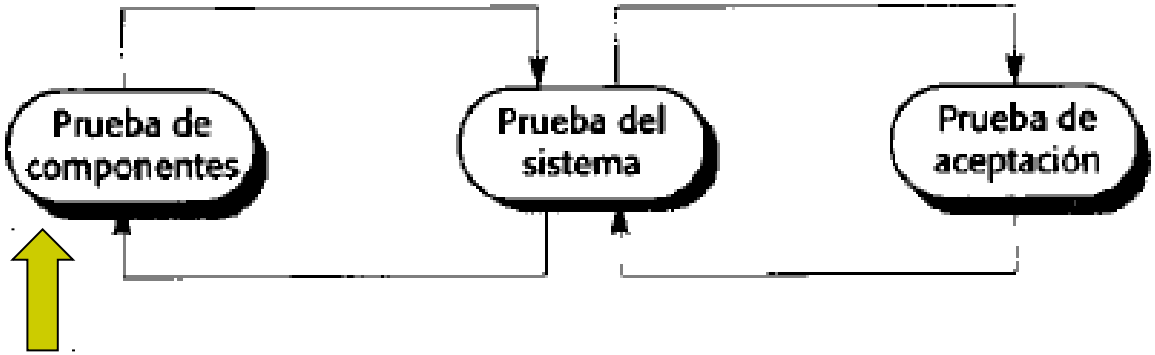



Prueba del sistema integrado – Interacción entre componentes

# 3. Validación del software

- La verificación y validación del software se crea para mostrar que un sistema cumple tanto con sus especificaciones como las expectativas del cliente.
- Las pruebas con datos simulados son la principal técnica de validación.
- No se prueban unidades monolíticas sino:

# 3. Validación del software
## Prueba de desarrollo

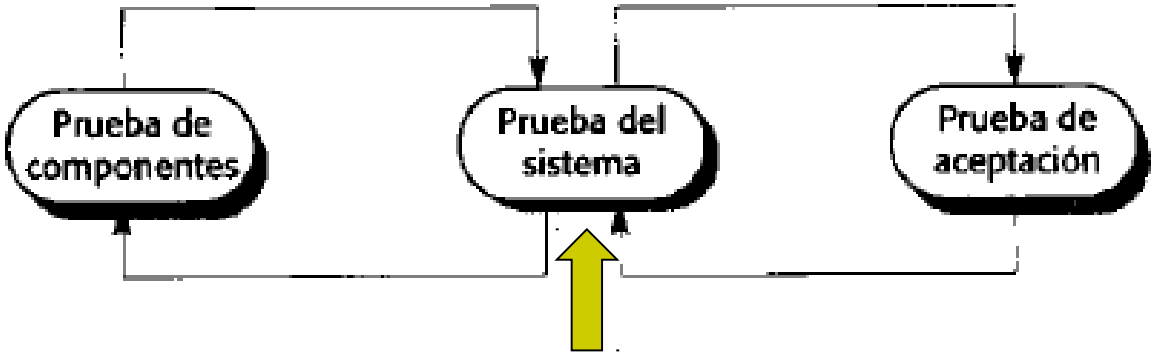



- Las personas que desarrollan el sistema ponen a prueba los componentes que lo constituyen.

- Cada componente se prueba de manera independiente. Éstos pueden ser simples entidades como funciones o clases de objeto o agrupamientos coherentes de dichas entidades.

- Se suelen utilizar herramientas de automatización de pruebas como JUnit, que pueden volver a correr pruebas de componentes cuando se crean nuevas versiones.

# 3. Validación del software
## Prueba del sistema

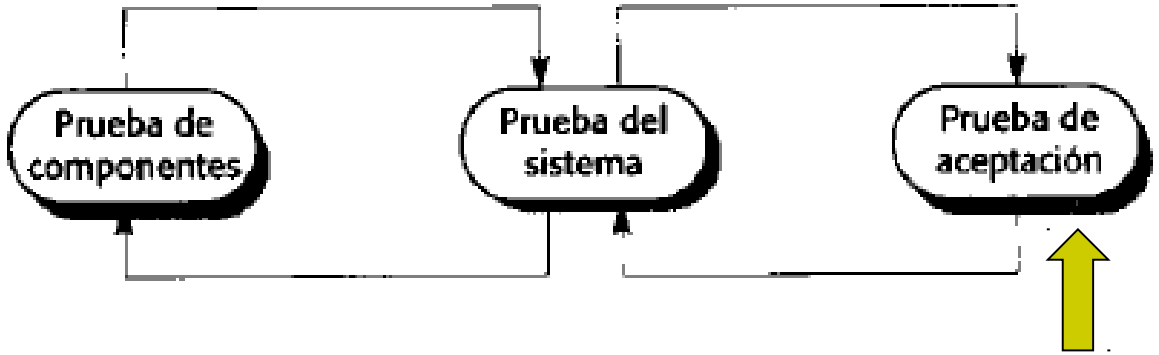



- Los componentes del sistema se integran para crear un sistema completo.

- Se intenta descubrir errores que resulten de interacciones no anticipadas entre componentes y problemas de interfaz entre ellos, así como de mostrar que el sistema cubre sus requerimientos.

- Para sistemas grandes, se hace por etapas, donde los componentes se juntan para formar subsistemas que se ponen a prueba de manera individual antes de que se integren para establecer el sistema final.

# 3. Validación del software
## Prueba de aceptación



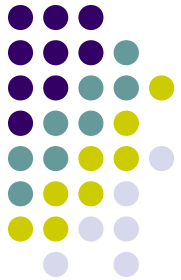

- Ésta es la etapa final en el proceso de pruebas, antes de que el sistema se acepte para uso operacional.

- El sistema se pone a prueba con datos suministrados por el cliente ya que los datos reales ejercitan el sistema de distinta manera que los de prueba.

- Revelan errores y omisiones en la definición de requerimientos.

- Revelan problemas de requerimientos cuando las instalaciones del sistema no cumplen las necesidades del usuario o cuando sea inaceptable el rendimiento.

# 3. Validación del software



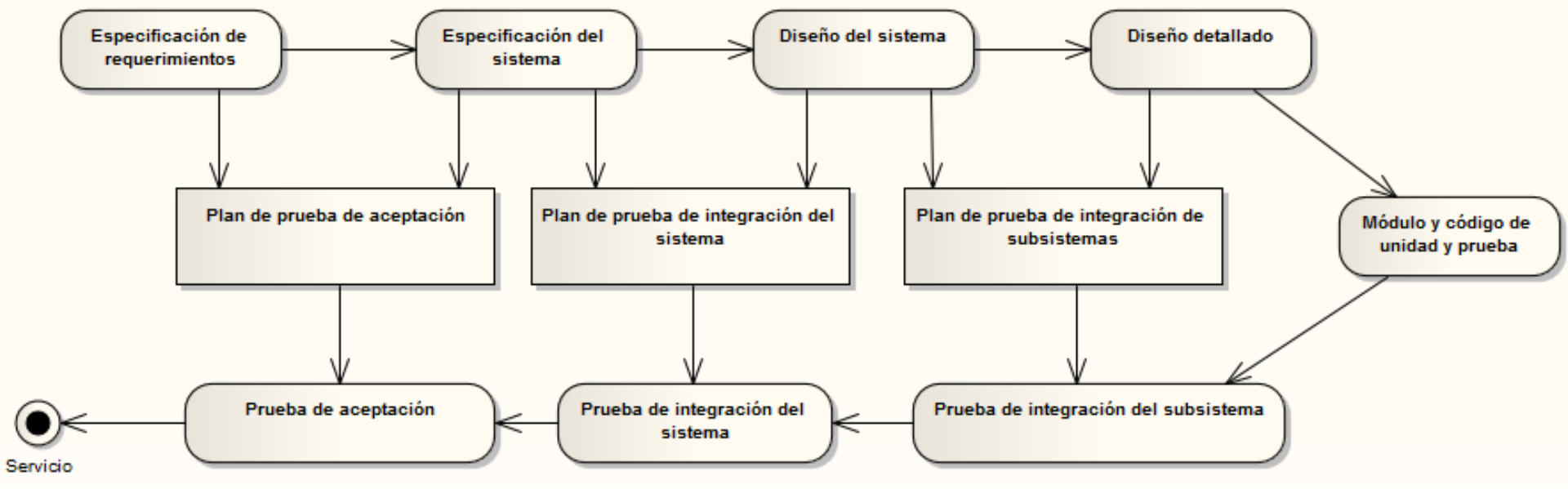
- Si es utiliza un **enfoque incremental**, cada incremento debe ponerse a prueba conforme se diseña y tales pruebas se basan en los requerimientos para dicho incremento.

- Si se sigue un plan, las pruebas se realizan mediante un conjunto de planes de prueba.

- Un equipo independiente de examinadores trabaja con los planes desarrollados a partir de la especificación y diseño de sistemas.

# 3. Validación del software

## Modelo de desarrollo “V”



# 4. Evolución del software

- Paradoja de “desarrollo de software” vs. “proceso de evolución”.
- Los sistemas no son totalmente nuevos → el mantenimiento es un **continuo**.
- En vez de procesos separados, el desarrollo y el mantenimiento conforman un solo proceso evolutivo donde el software cambia continuamente