

2022

**PROCESOS**

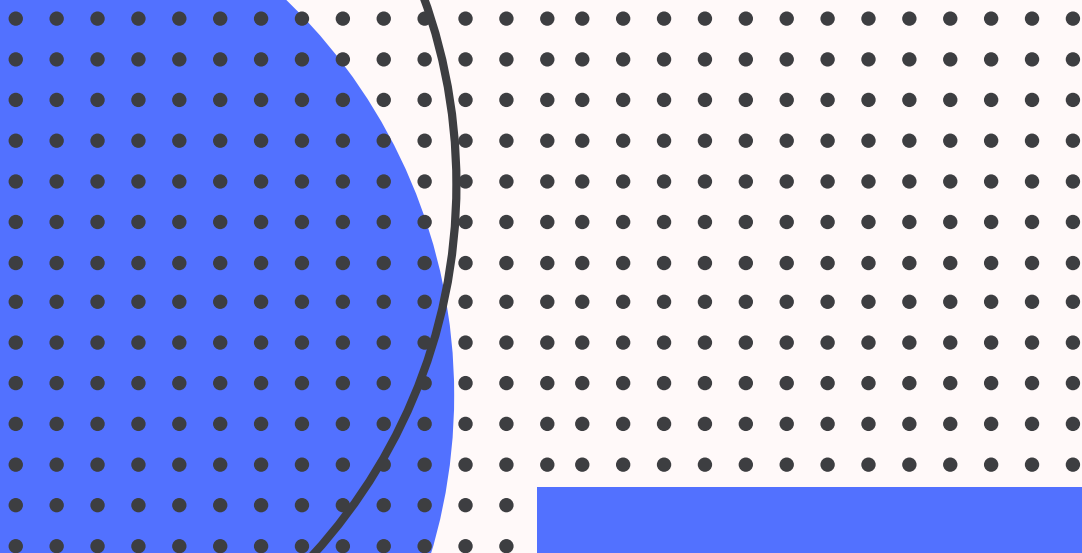


---

del

**SOFTWARE**

**NEW MINDS**



# ÍNDICE

- ITERACIÓN DE PROCESOS**
- MODELOS DE PROCESOS DEL SOFTWARE**

**(Marco Perez)**

**\*INGENIERIA DE REQUERIMIENTOS:**

- REQUERIMIENTOS FUNCIONALES**
- REQUERIMEINTOS NO FUNCIONALES**

**(Jair Coronel)**

- ACTIVIDADES DEL IR**
- REQUERIMIENTOS DE PROCESOS**

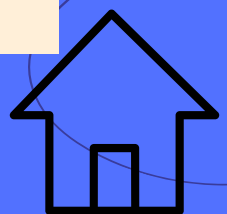
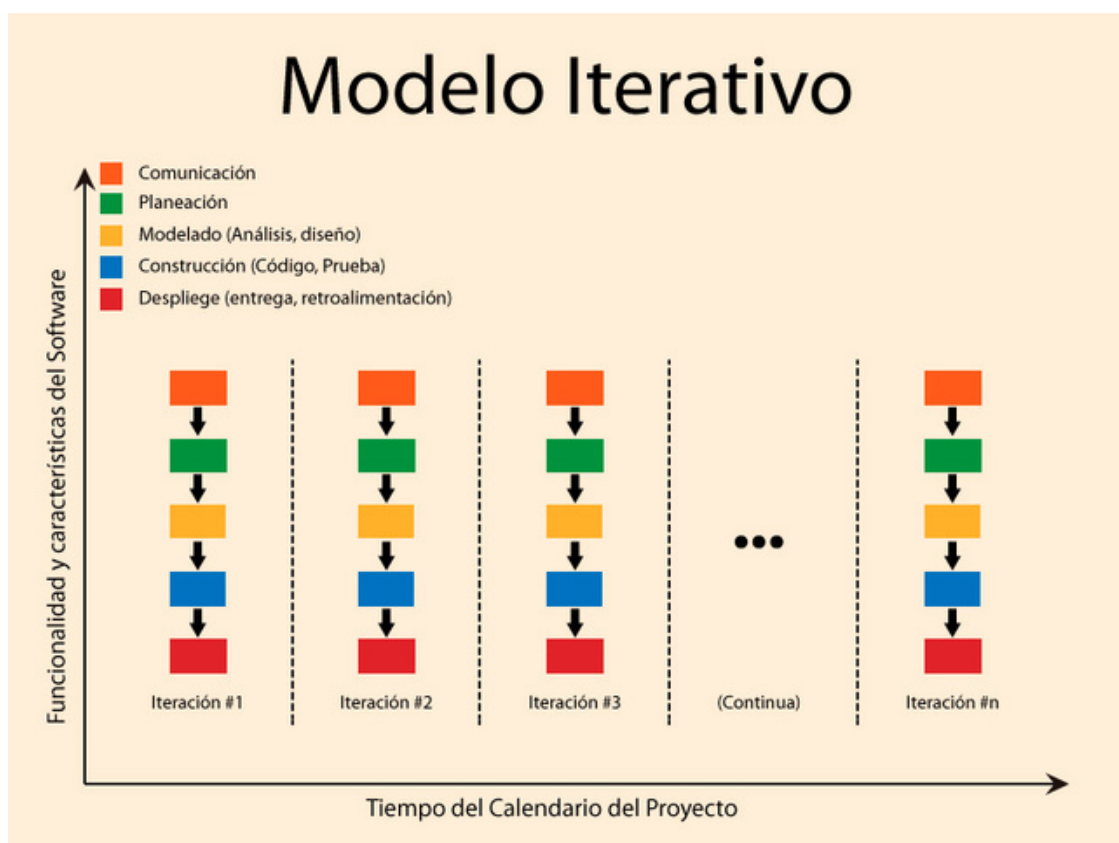
**(Fernando Paco)**

- REQUERIMIENTOS DE USUARIOS**
- REQUERIMIENTOS DEL SISTEMA**

**(Diego de la cruz)**

# ITERACIÓN DE PROCESOS

Cuando hablamos de procesos iterativos nos referimos a esos términos como metodologías ágiles, que nos hacen pensar automáticamente en equipos de ingeniería. Pero la mayoría de los equipos, de algún modo u otro, trabajan con un método iterativo para reducir riesgos, gestionar la eficiencia y afrontar los problemas de una manera flexible y dinámica.



# MODELOS DE PROCESOS DEL SOFTWARE

Cada modelo es una descripción de un proceso software que se presenta desde una perspectiva particular

Alternativamente, a veces se usan los términos ciclo de vida y Modelo de ciclo de vida.

## Clasificación de los modelos de proceso (ii)

### ■ Ejemplos de modelos de proceso de diferentes categorías

#### ■ Modelos tradicionales

- Clásico, lineal o en cascada
- Estructurado
- Basado en prototipos
- Desarrollo rápido de aplicaciones (RAD)

#### ■ Modelos evolutivos

- Incremental
- Iterativo
- En espiral

#### ■ Modelos para sistemas orientados a objetos

- De agrupamiento
- Proceso Unificado

#### ■ Modelos basados en reutilización

- Basado en componentes
- Proceso Unificado

#### ■ Procesos ágiles

- Programación extrema (XP)
- Scrum
- Desarrollo de *software* adaptativo
- Crystal

#### ■ Modelos para sistemas web

- Modelos de Pressman
- UML-based Web Engineering

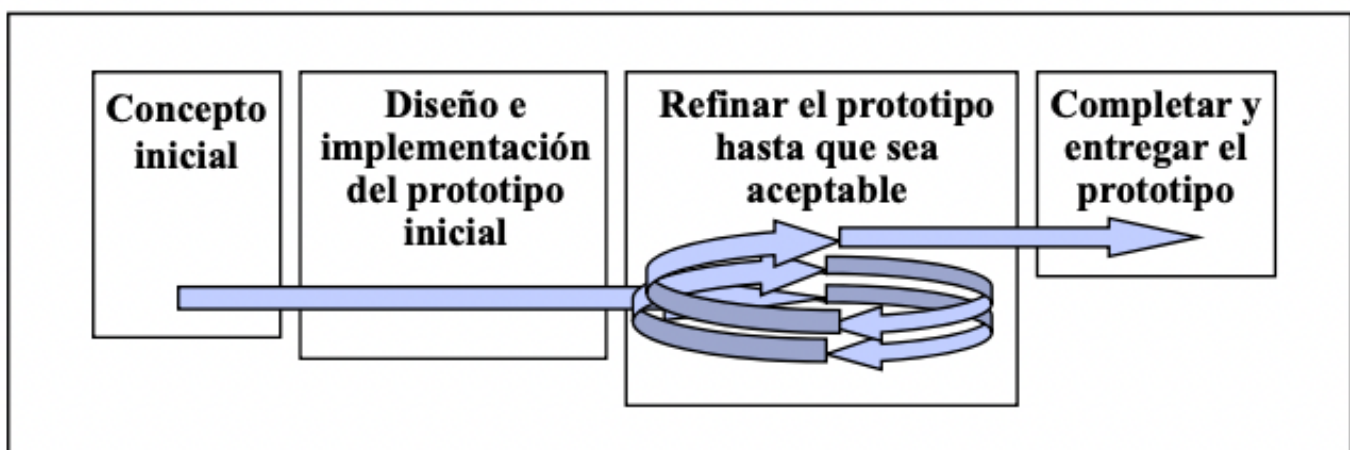




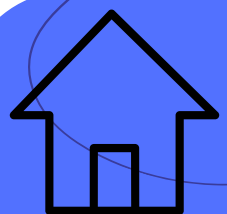
# Modelos basados en prototipos

## Prototipado evolutivo (ciclo de vida iterativo)

**Características** Enfoque de desarrollo que se utiliza cuando no se conoce con seguridad lo que se quiere construir Se comienza diseñando e implementando las partes más destacadas del sistema La evaluación del prototipo proporciona la realimentación necesaria para aumentar y refinar el prototipo El prototipo evoluciona y se transforma en el sistema final



Modelo de prototipado evolutivo

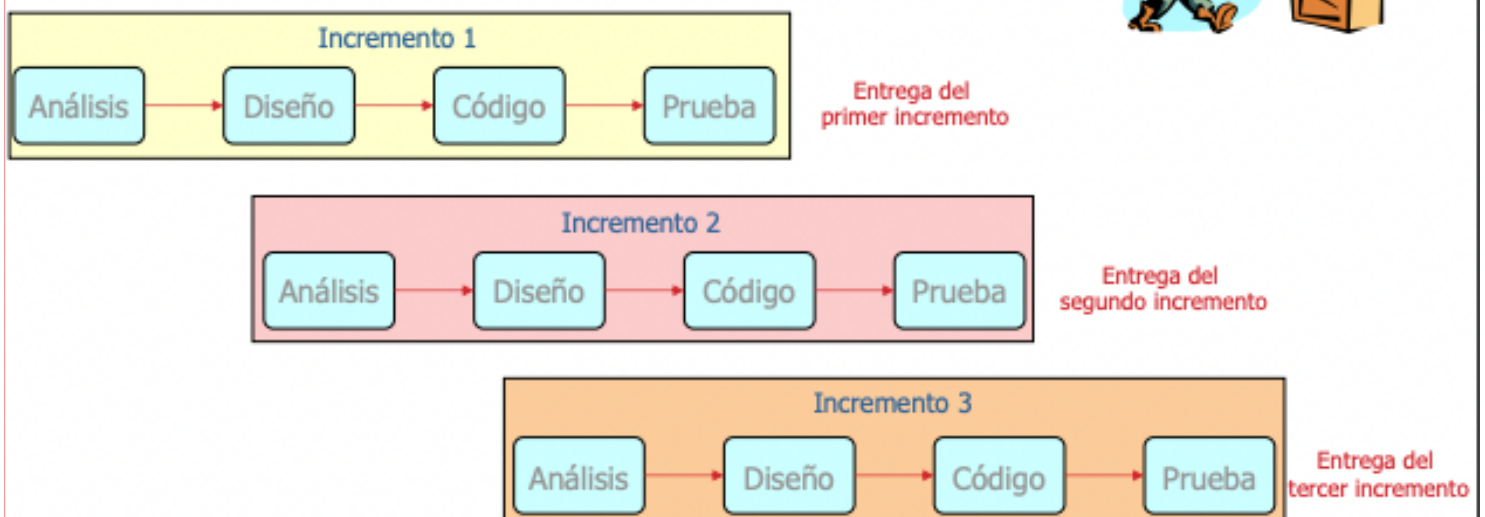


# Modelo incremental

En este modelo, el sistema, tal y como está especificado en la especificación de requisitos del software, se divide en subsistemas

de acuerdo a su funcionalidad. Las versiones se definen comenzando con un subsistema funcional pequeño y agregando funcionalidad con cada nueva versión.

## Modelo incremental (ii)



# Modelo iterativo

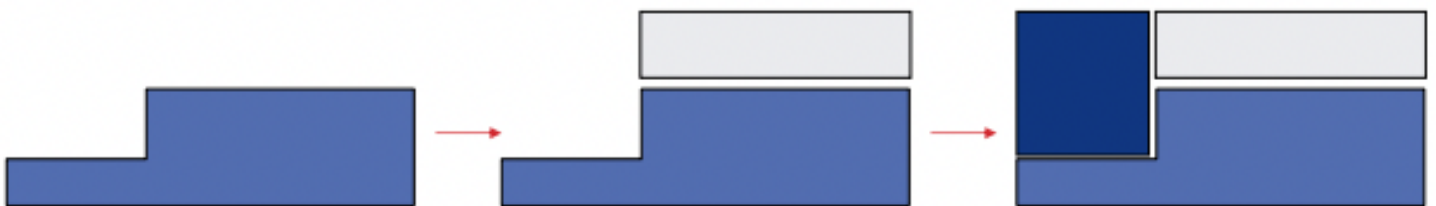
Entrega un sistema completo desde el principio, para posteriormente cambiar la funcionalidad de cada subsistema con cada versión

## Características del ciclo iterativo [Muller, 1997]

Se basa en la evolución de prototipos ejecutables, mensurables y evaluables Se van incorporando cambios en cada iteración Exige más atención e implicación de todos los actores del proyecto

### Incremental vs. iterativo

**Desarrollo incremental:** sistema parcial, funcionalidad completa



**Desarrollo iterativo:** sistema completo; funcionalidad parcial

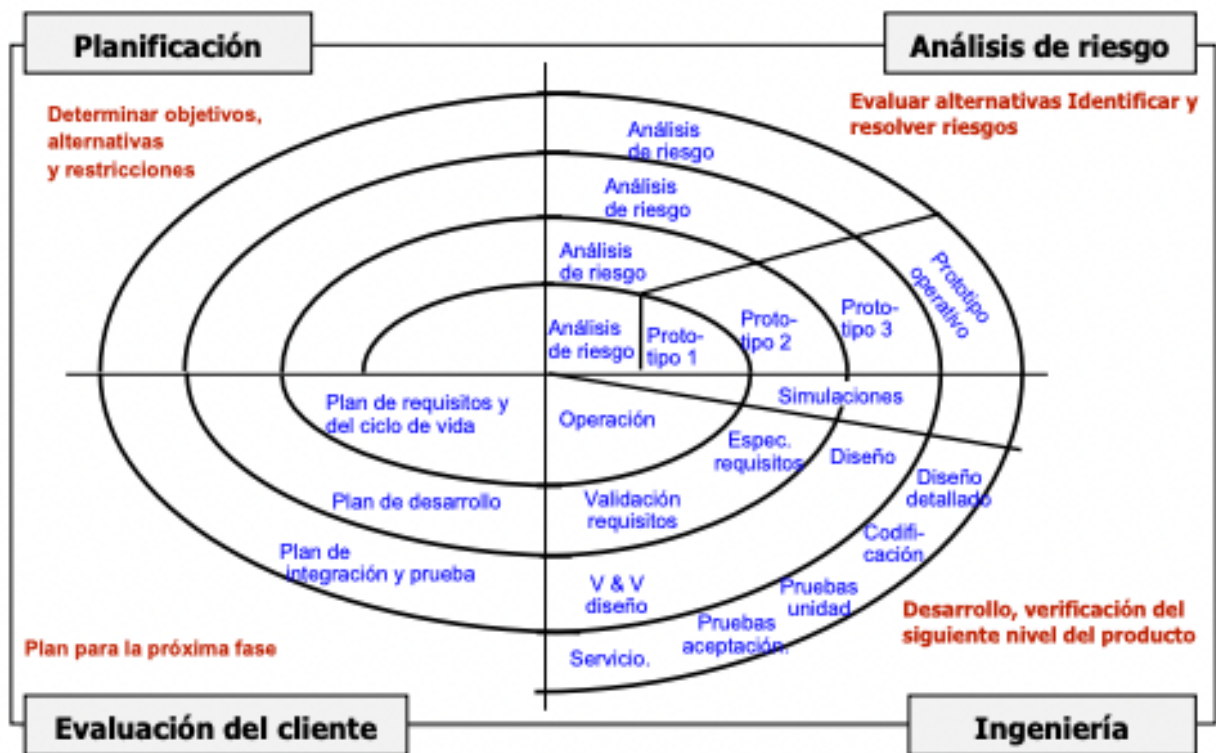




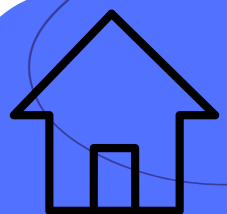
# Modelo en espiral

Fue propuesto inicialmente por B. Boehm [Boehm, 1986, 1988] Es un modelo de proceso de software evolutivo, que proporciona el potencial para el desarrollo rápido de versiones incrementales del software

## Modelos en espiral (ii)

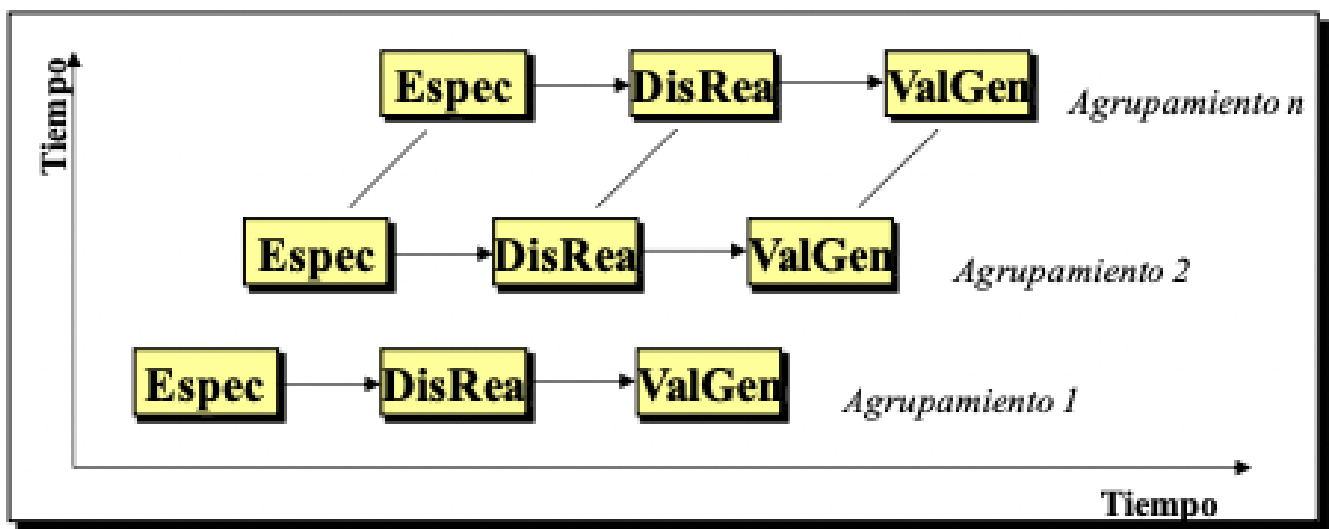


Ciclo de vida en espiral [Boehm, 1988]

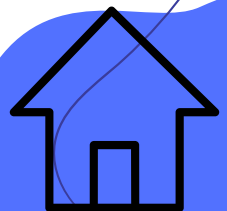


# Modelo de agrupamiento

**Enfoque ascendente** La ocultación de la información posibilita la forma del modelo de clusters de ingeniería concurrente



Distribución temporal de las fases de cada agrupamiento

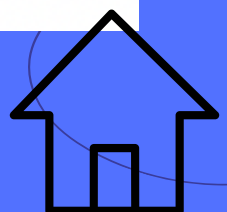


## El proceso unificado (i)

- Definido por **Rational Software Corporation** [Jacobson et al., 2000]
  - Evolución del proceso Objectory de Rational
  - Utilización de UML [Booch et al., 1999] como lenguaje de modelado
  - Basado en componentes
- Características
  - **Conducido por casos de uso**
    - Los casos de uso se implementan para asegurar que toda la funcionalidad se realiza en el sistema y verificar y probar el mismo. Como los casos de uso contienen las descripciones de las funciones, afectan a todas las fases y vistas
  - **Centrado en la arquitectura**
    - La arquitectura se describe mediante diferentes vistas del sistema. Es importante establecer una arquitectura básica pronto, realizar prototipos, evaluarla y finalmente refinarla durante el curso del proyecto
  - **Iterativo e incremental**
    - Resulta práctico dividir los grandes proyectos en mini proyectos, cada uno de los cuales es una iteración que resulta en un incremento

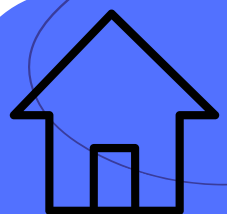
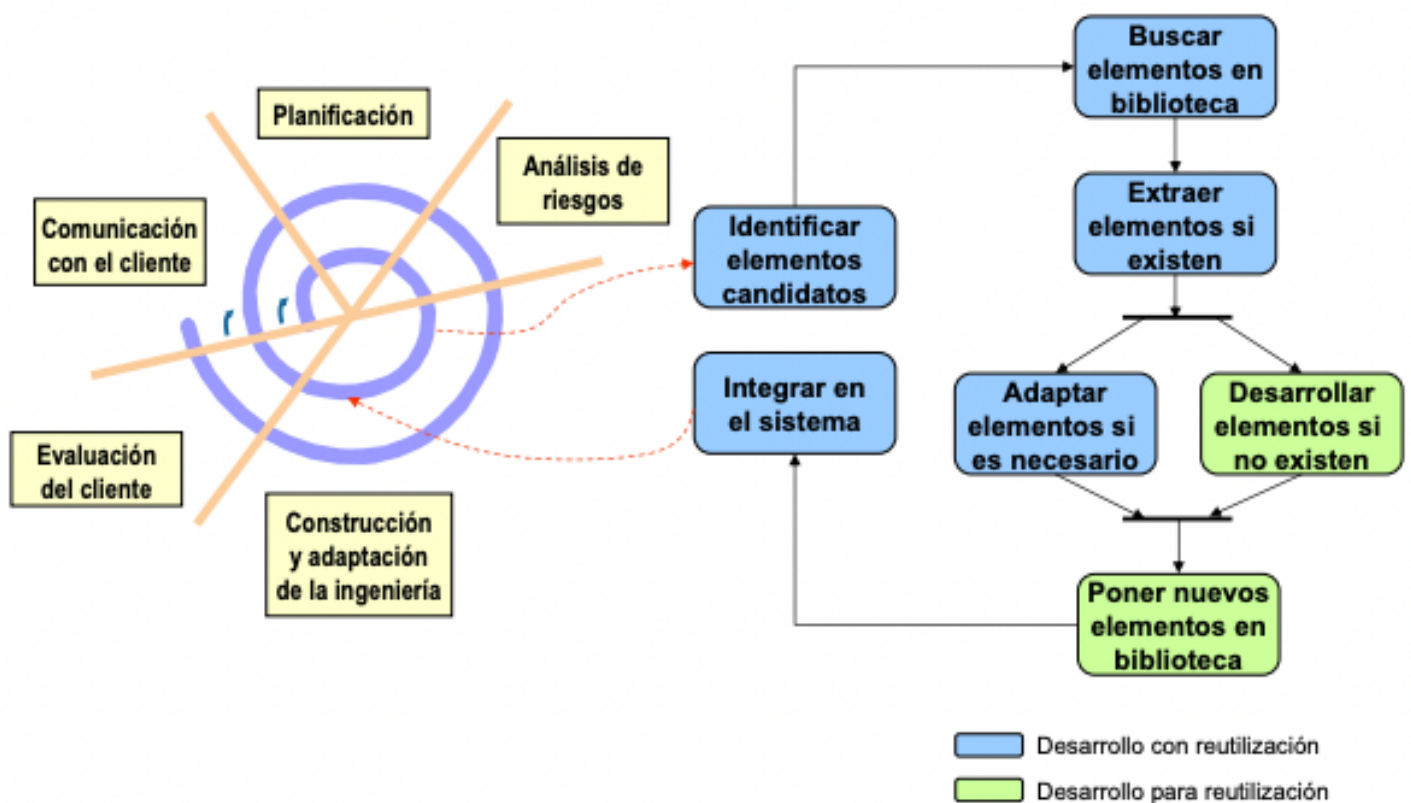
## El proceso unificado (iii)

- Dentro de cada fase se puede, a su vez, descomponer el trabajo en **iteraciones** con sus incrementos resultantes
- Cada fase termina con un **hito**, cada uno de los cuales se caracteriza por la disponibilidad de un conjunto de componentes de *software*
  - Objetivos de los hitos
    - Toma de decisiones para continuar con la siguiente fase
    - Controlar el progreso del proyecto
    - Proporcionar información para la estimación de tiempo y recursos de proyectos sucesivos



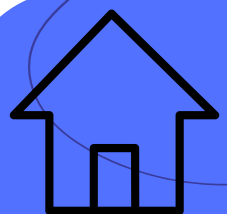
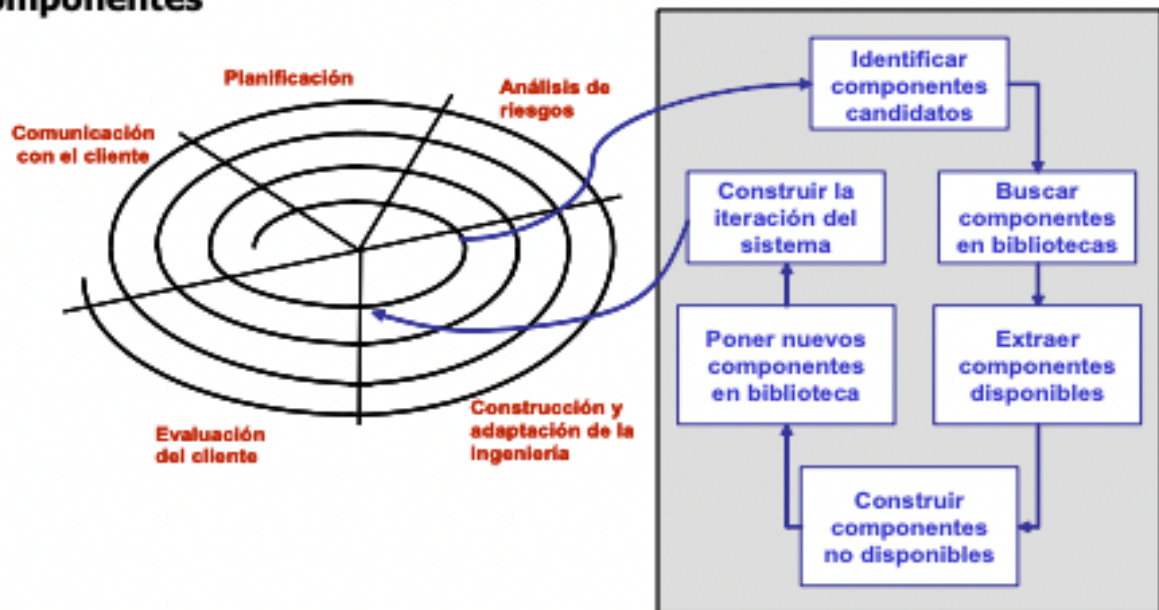
# Modelos basados en reutilización

Enfoque de desarrollo que trata de maximizar la reutilización de software existente [Sommerville, 2002] Se basa en la existencia de un número significativo de elementos reutilizables



## ■ Desarrollo basado en componentes (i)

- Configura aplicaciones a partir de componentes de *software* preparados [Pressman, 2002]
- Enfoque iterativo y evolutivo
- Se enmarca en un contexto más amplio: **Ingeniería del *Software* basada en componentes**



# Procesos ágiles

Constituyen un nuevo enfoque en el desarrollo de software cuyas principales características son Menor énfasis en el análisis, diseño y documentación Equipos pequeños Desarrollo incremental Programación (planificación temporal) en cajas de tiempo Supervivencia en un entorno caótico

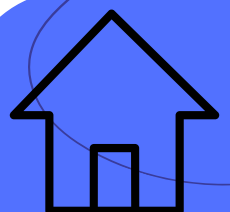
Diversos enfoques XP (eXtreme Programming) [Beck, 1999] Scrum [Schwaber, 1995] Crystal [Cockburn, 1999] Proceso Software Adaptativo [Highsmith, 2000]

Proceso

Unificado

Ágil

(<http://www.ambyssoft.com/unifiedprocess/agileUP.html>)



# Scrum

**Propuesto por Jeff Sutherland y desarrollado por Schwaber y Beedle**

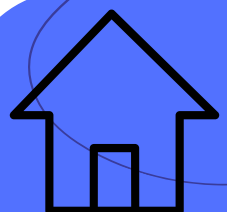
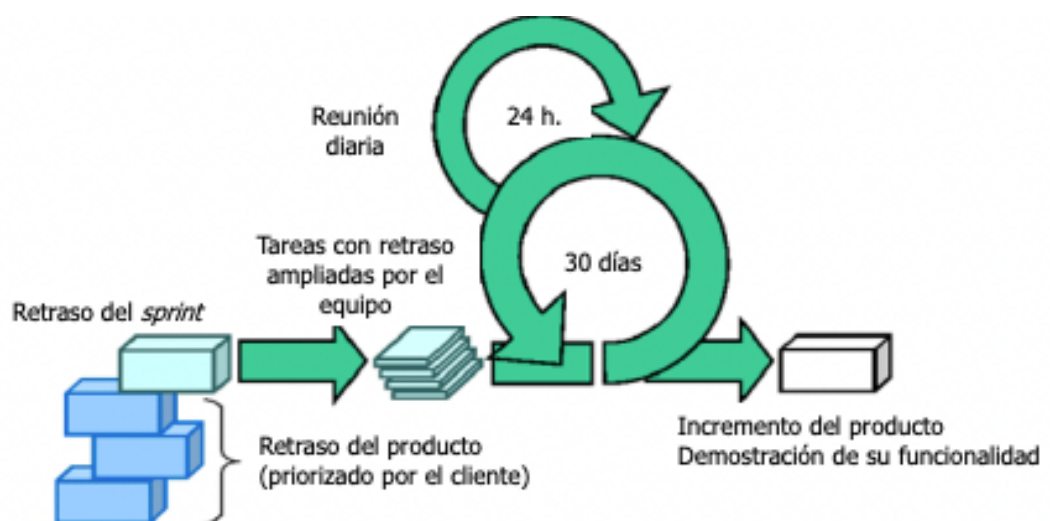
**Actividades estructurales Requisitos**

**Análisis Diseño Evolución Entrega**

**Dentro de cada actividad las tareas se organizan con un patrón de proceso denominado sprint**

**El trabajo del sprint se adapta al problema y se define y modifica en tiempo real por el equipo Scrum**

**Uso de patrones de proceso de demostrada eficacia en proyectos críticos, con plazos cortos y requisitos cambiantes**



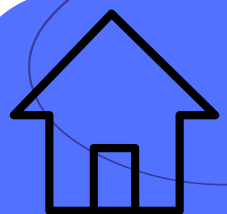


## Proceso Unificado Ágil (iii)

- Lanzamiento de versiones incrementales a lo largo del tiempo
  - Versiones de desarrollo
    - Se lanzan al final de cada iteración
  - Versiones de producción
    - Son versiones de desarrollo que han pasado procesos de aseguramiento de la calidad, prueba y despliegue
    - Se lanzan con menos frecuencia que las de desarrollo



Copyright 2005 Scott W. Ambler

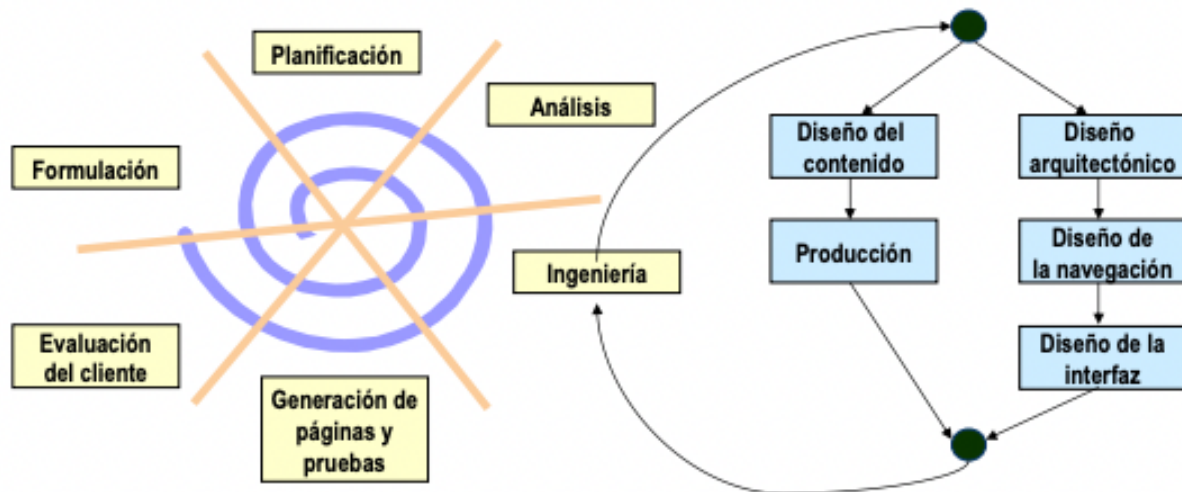




# Modelos para la ingeniería web

## Modelos de Pressman (i)

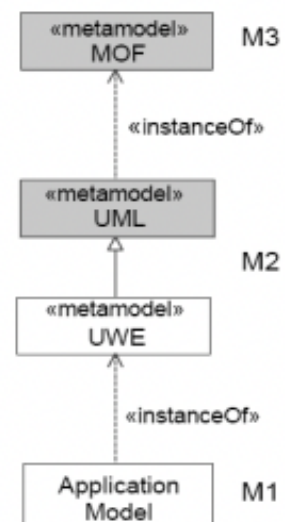
- Pressman propuso inicialmente el modelo IWEB [Pressman, 2002] tomando como base su modelo de ciclo de vida en espiral



Modelo de proceso de IWEB [Pressman, 2002]

## UWE (UML-based Web Engineering) (i)

- Características [Koch, 2001], [Hennicker y Koch, 2000]
  - **Desarrollo iterativo e incremental:** basado en el Proceso unificado
  - **Uso de UML:** perfil UML propio
  - **Centrado en la sistematización y automatización**
    - Proceso sistemático de diseño
    - Generación semiautomática de aplicaciones web a través de un *framework* de publicación XML (UWEXML)
  - **UWE comprende**
    - Una **notación**
    - Un **método**
    - Un **metamodelo**
    - Un **proceso de desarrollo**
    - Una **herramienta CASE**



# Ingeniería de requerimientos

## *Según Zave:*

- Rama de la ingeniería del software que trata con el establecimiento de los objetivos, funciones y restricciones de los sistemas software.
- Asimismo, se ocupa de la relación entre estos factores con el objeto de establecer especificaciones precisas.

## *Según Boehm:*

- Ingeniería de Requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema.

## *Según Loucopoulos:*

- Trabajo sistemático de desarrollo de requisitos, a través de un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido.



# **REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES**

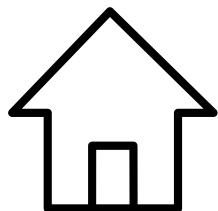
## **Requerimientos Funcionales**

Los requerimientos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer

## **requerimientos No Funcionales**

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

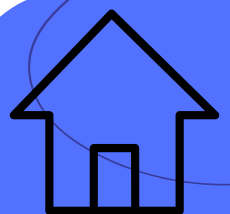


# **¿Qué es la ingeniería de requisitos/requerimientos (IR)?**

**La IR es la rama de la ingeniería de software que se encargar de recopilar analizar y verificar las necesidades exactas del cliente para un sistema de software.**

**Entonces siguiendo este concepto a que nos referimos con “requerimientos”; según la IEEE (Institute of Electrical and Electronics Engineers) el termino requerimientos es definido como:**

**-Condición o capacidad requerida por el usuario o cliente para resolver un problema o alcanzar un objetivo.**





## Extracción

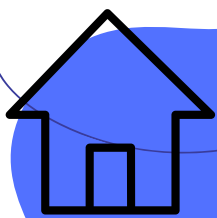
Aquí, los analistas de requerimientos deben trabajar junto al cliente para descubrir el problema que el sistema debe resolver, los diferentes servicios que el sistema debe prestar, las restricciones que se pueden presentar, etc. Es importante, que la extracción sea efectiva, ya que la aceptación del sistema dependerá de cuan bien éste satisfaga las necesidades del cliente. ¿Como recolectar dicha información?, bueno hay muchas maneras como por ejemplo: entrevistas frente a frente, entrevistas grupales , cuestionarios, prototipos,etc.

## Análisis

Sobre la base de la extracción realizada previamente, comienza esta fase en la cual se enfoca en descubrir problemas con los requerimientos del sistema identificados hasta el momento. Usualmente se hace un análisis luego de haber producido un bosquejo inicial del documento de requerimientos; en esta etapa se leen los requerimientos, se conceptúan, se investigan, se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones, y luego se van fijando reuniones con el cliente para discutir los requerimientos.

## Especificación

esta etapa se va realizando conjuntamente con el análisis, se puede decir que la especificación es el "pasar en limpio" el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML (Lenguaje de Modelado Unificado), que es un estándar para el modelado orientado a objetos, por lo que los casos de uso y la obtención de requerimientos basada en casos de uso se utiliza cada vez más para la obtención de requerimientos.



## **Validación**

La validación es la etapa final de la IR. Su objetivo es, ratificar los requerimientos, es decir, verificar todos los requerimientos que aparecen en el documento especificado para asegurarse que representan una descripción, por lo menos, aceptable del sistema que se debe implementar. Esto implica verificar que los requerimientos sean consistentes y que estén completos.

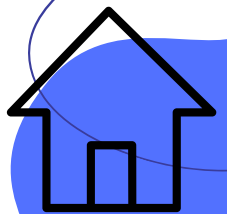
# **Requerimientos de procesos**

El proceso para convertir los requerimientos indicados en requerimientos reales consiste en un proceso de filtrado según el significado y otros aspectos según considere.

-Un proceso involucra por lo general un conjunto de herramientas y técnicas.

-Un proceso es un conjunto ordenado de tareas; una serie de pasos que involucran actividades, restricciones y recursos que producen una determinada salida esperada.

Entonces podemos decir que un proceso es una serie de pasos ordenados, que combinan herramientas y técnicas para generar un producto.



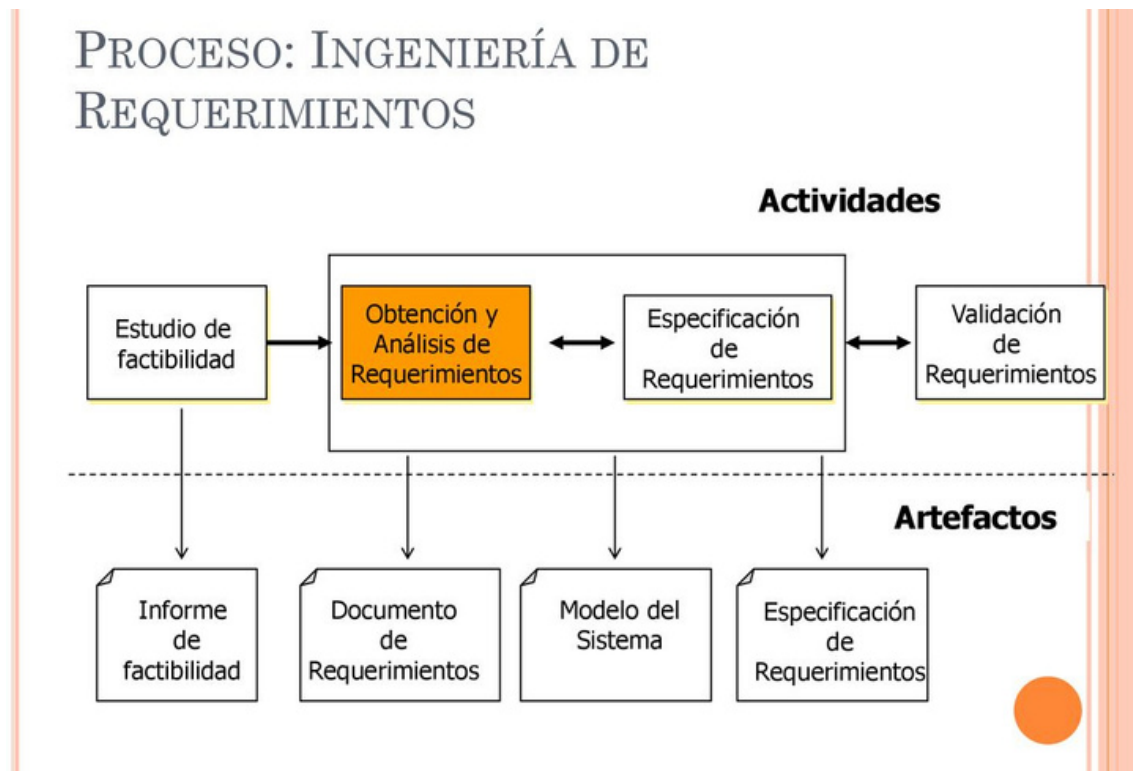
# Características de los procesos

**Un proceso debe utilizar recursos.**

**Está sujeto a una serie de restricciones.**

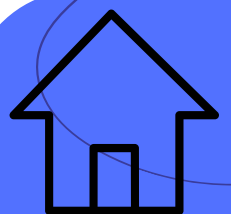
**Genera productos intermedios y finales.**

**Las actividades se realizan secuencialmente.**



**Cuando el proceso implica la construcción de algún producto, solemos referirnos al proceso como un ciclo de vida.**

**El proceso de desarrollo de software se denomina ciclo de vida del software. Los procesos son importantes porque imponen consistencia y estructura sobre un conjunto de actividades.**



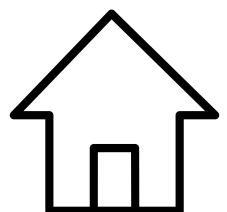
# ¿Por qué debemos modelar un proceso?

**Para tener en una comprensión común de las actividades, recursos y restricciones del proyecto.**

**Para construir un producto de calidad y que sea efectivo.**

**Para cumplir con cada actividad, con las restricciones y el presupuesto establecido.**

**Para encontrar las inconsistencias, redundancias y omisiones en el proceso y en las partes que lo contribuyen**





# Requerimientos del usuario

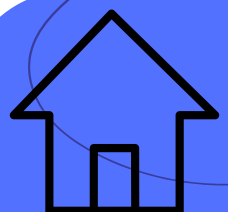
**Son declaraciones en lenguaje natural y en diagramas informales, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.**

## **Usuario final:**

**Son las personas que usarán el sistema desarrollado. Ellos están relacionados con la usabilidad, la disponibilidad y la fiabilidad del sistema; están familiarizados con los procesos específicos que debe realizar el software, dentro de los parámetros de su ambiente laboral. Serán quienes utilicen las interfaces y los manuales de usuario.**

## **Usuario Líder:**

**Son los individuos que comprenden el ambiente del sistema o el dominio del problema en donde será empleado el software desarrollado. Ellos proporcionan al equipo técnico los detalles y requerimientos de las interfaces del sistema.**



### **Personal de Mantenimiento:**

**Para proyectos que requieran un mantenimiento eventual, éstas personas son las responsables de la administración de cambios, de la implementación y resolución de anomalías.**

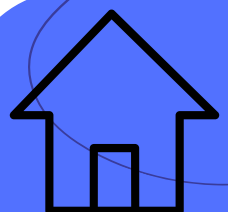
**Su trabajo consiste en revisar y mejorar los procesos del producto ya finalizado.**

### **Analistas y programadores:**

**Son los responsables del desarrollo del producto en sí; ellos interactúan directamente con el cliente.**

### **Personal de pruebas:**

**Se encargan de elaborar y ejecutar el plan de pruebas para asegurar que las condiciones presentadas por el sistema son las adecuadas. Son quienes van a validar si los requerimientos satisfacen las necesidades del cliente.**

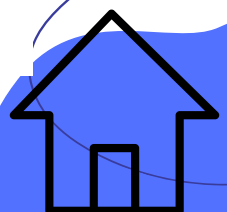
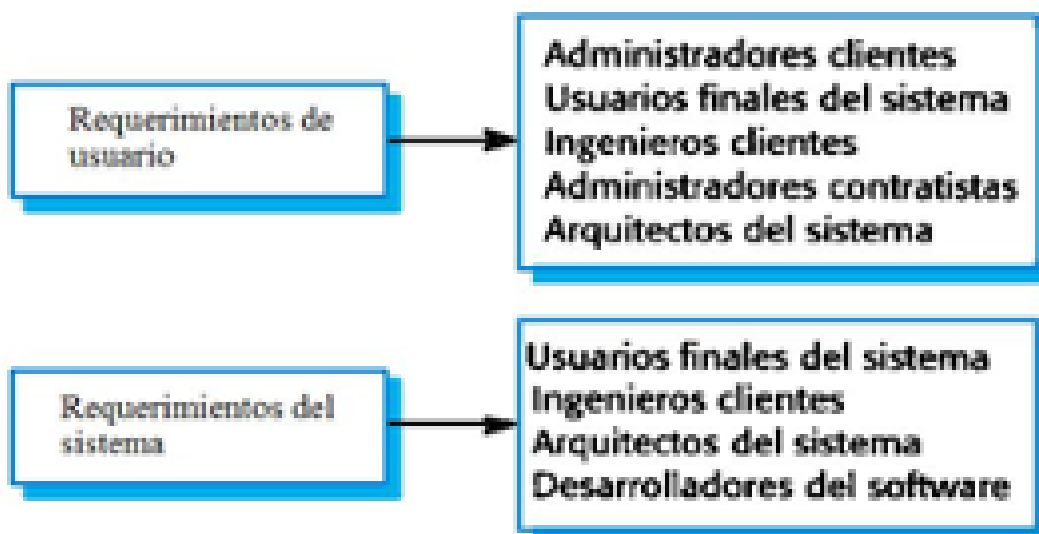


# Requerimientos del sistema

Son descripciones más detalladas de los requerimientos del usuario. Definen el contrato de la especificación del sistema y debe ser una especificación completa y consistente del sistema. Son el punto de partida de los ingenieros del software hacia el diseño del sistema.

Establecen con detalle las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema debe ser preciso. (Se incluye en el contrato)

## Requisitos de los lectores



## **Objetivos:**

- Fomentar la realización de una ingeniería de requisitos de acuerdo a los principios de calidad y eficiencia en el desarrollo establecida por MADEJA
- Trasmitir la importancia de la ingeniería de requisitos y la trazabilidad de los requisitos como aspecto de un impacto directo en la calidad del sistema de información.

## **Responsabilidades:**

- Definir y establecer pautas que ayuden a estandarizar el desarrollo de procesos y actividades relacionadas con la ingeniería de requisitos de acuerdo a las buenas prácticas propuestas por MADEJA. Establecer recursos que faciliten la integración de estas buenas prácticas dentro del desarrollo común de aplicaciones
- Facilitar herramientas que ayuden en la automatización, adopción y mantenimiento de las buenas prácticas establecidas por MADEJA para el conjunto de actividades y procesos relacionadas
- Facilitar la plantilla del documento de Especificación de Requisitos del Sistema (ERS).

## **Actividades:**

- Identificar las necesidades de negocio de clientes y usuarios
- Desarrollar los requisitos de un sistema software que satisfaga las necesidades de negocio
- Gestionar los requisitos del sistema software a desarrollar



# Conclusión

**Concluimos con el conocimiento necesario y comprendiendo que en retrospectiva es muy importante el modelado de los procesos, comprender los requerimientos para desarrollar un buen sistema que sea escalable con el tiempo**

