

INGENIERÍA DE SOFTWARE I

Tema 5: Introducción al Proceso Unificado

Grado en Ingeniería Informática
Fecha de última modificación: 9-2-2025

Dr. Francisco José García-Peñalvo / fgarcia@usal.es
Dra. Alicia García-Holgado / aliciagh@usal.es
Dra. Andrea Vázquez-Ingelmo / andreavazquez@usal.es



Departamento de Informática y Automática
Universidad de Salamanca



Resumen

Resumen	En este tema se hace una presentación del Proceso Unificado. Se hace especial hincapié en sus características, su ciclo de vida y sus artefactos. En el tema siguiente se da continuidad a estas características con la descripción de los flujos de trabajo de este proceso
Descriptoros	Proceso; Proceso Unificado; ciclo de vida; casos de uso; arquitectura <i>software</i> ; iteratividad; incremental
Bibliografía	[Jacobson et al., 2000] Capítulos 1, 2, 3, 4 y 5

Esquema

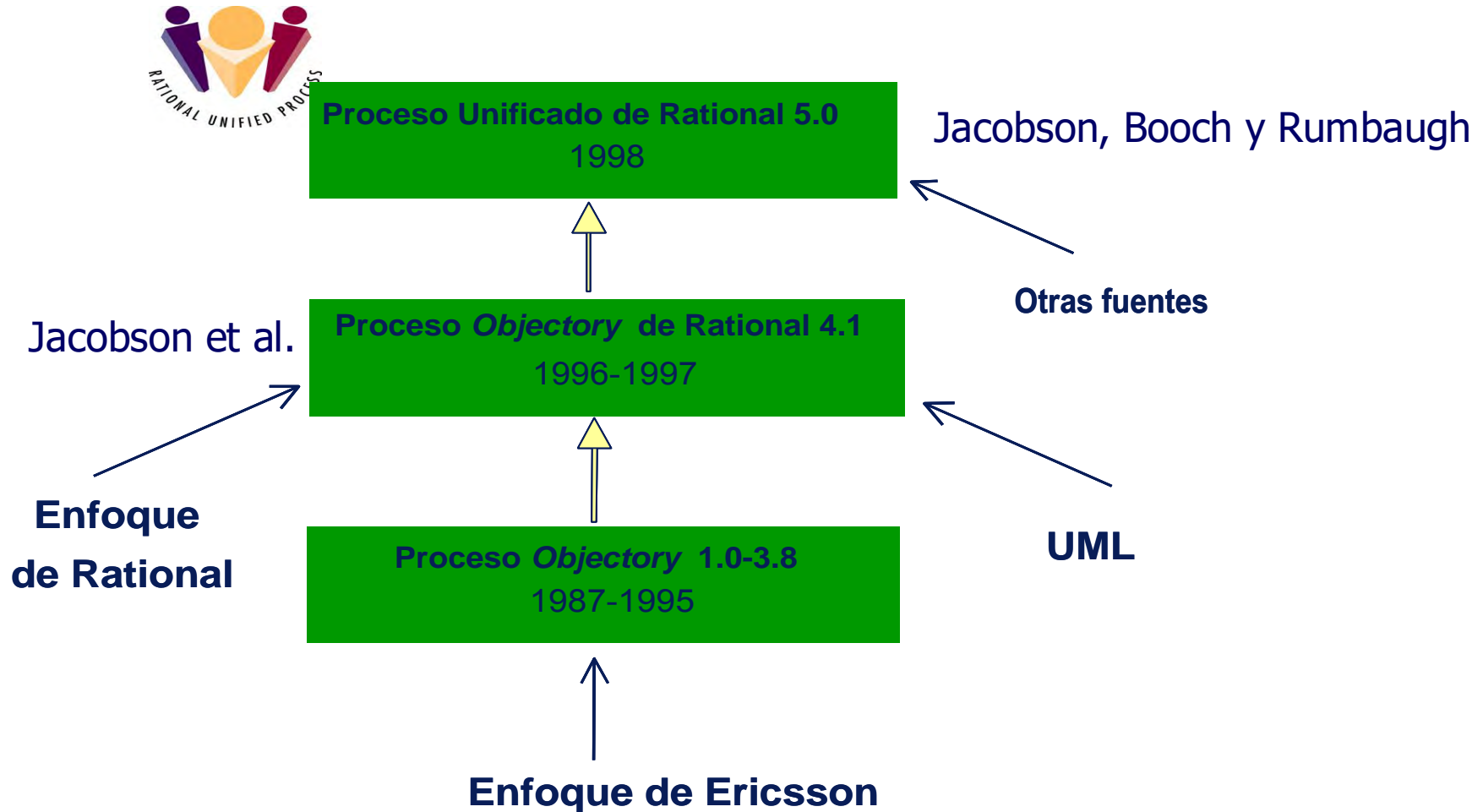
- Introducción
- La vida del Proceso Unificado
- El producto
- El proceso
- Aportaciones principales del tema
- Lecturas complementarias
- Referencias



1. Introducción

Introducción (i)

Orígenes del Proceso Unificado



Introducción (ii)

- En Febrero de 2003 IBM compró Rational y en 2006 creó una versión de RUP para procesos ágiles
- Plataforma ***IBM Rational Method Composer V7.5.2***
 - Herramienta de creación y publicación de métodos basada en Eclipse
 - Incluye una biblioteca de procesos
 - Ofrece una guía de mejores prácticas para el desarrollo de *software* (RUP y procesos ágiles)

Introducción (iii)

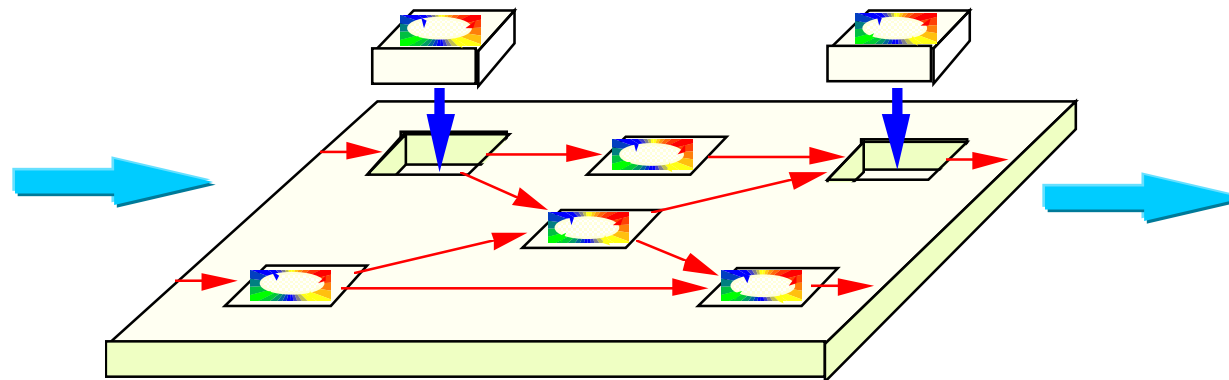
El Proceso Unificado es más que un simple proceso [Jacobson et al., 1999], es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas *software*, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos

- Características generales
 - Está basado en componentes
 - Utiliza UML [Booch et al., 1999; OMG, 2017]
- Características principales [Jacobson et al., 2000]
 - Es un proceso conducido por casos de uso
 - Está centrado en la arquitectura
 - Es iterativo e incremental

Introducción (iv)

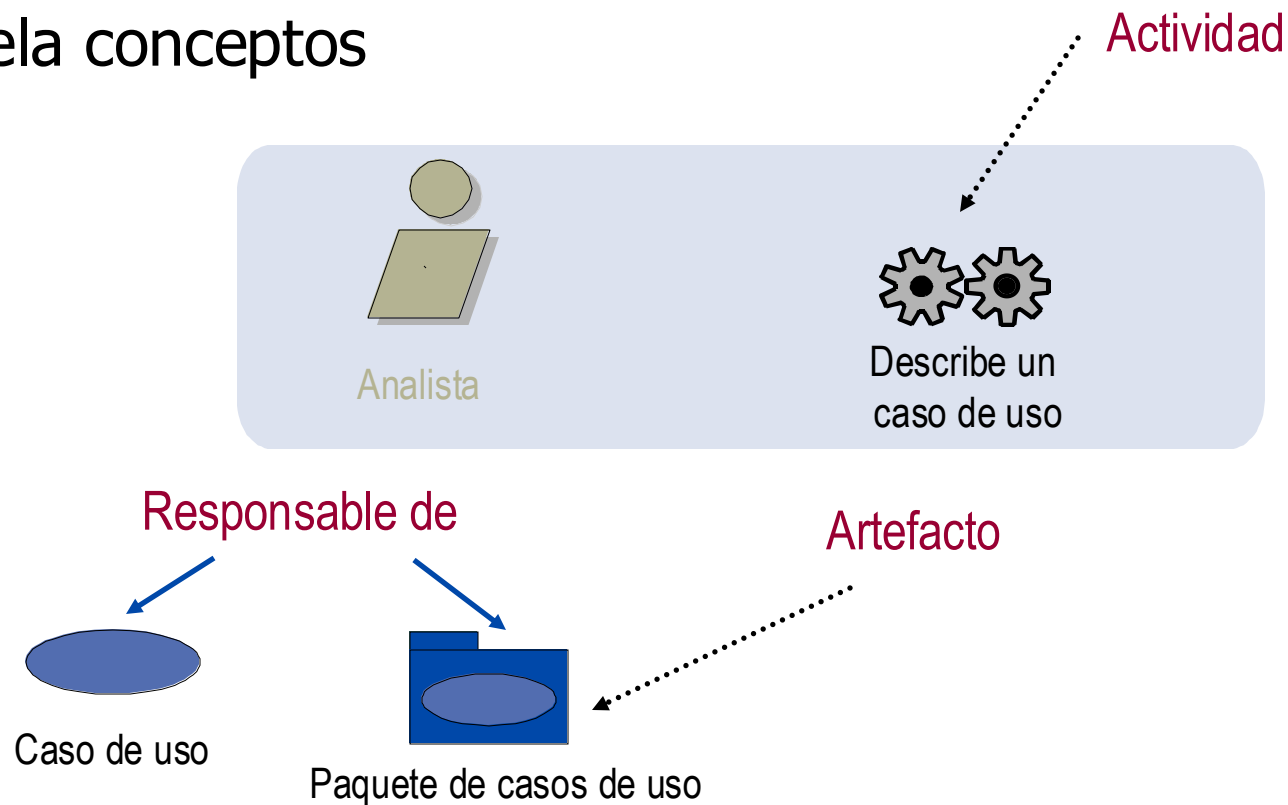
■ **Un marco de trabajo genérico**

- No existe un proceso universal
- Puede extenderse y especializarse para una gran variedad de sistemas de *software*
 - Flexibilidad
 - Está basado en componentes
- Permite gran variedad de estrategias de ciclo de vida
 - Se pueden definir diferentes conjuntos de productos
 - Se pueden definir actividades y encargados de las mismas



Introducción (v)

- Selecciona qué artefactos producir
- Define actividades y *stakeholders*
- Modela conceptos



2. La vida del Proceso Unificado



<https://unsplash.com/photos/ZhVKeFCb6NE>

La vida del Proceso Unificado (i)

- El Proceso Unificado se repite a lo largo de una serie de **ciclos de desarrollo** que constituyen la vida de un sistema
- Cada **ciclo de desarrollo** concluye con una **versión entregable** del producto
- Cada ciclo consta de cuatro **fases**
 - **Inicio**
 - Se define el alcance del proyecto y se desarrollan los casos de negocio
 - **Elaboración**
 - Se planifica el proyecto, se especifican en detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema
 - **Construcción**
 - Se construye el producto
 - **Transición**
 - El producto se convierte en versión beta
 - Se corrigen problemas y se incorporan mejoras sugeridas en la revisión

La vida del Proceso Unificado (ii)

- **Etapa de Ingeniería**
 - Equipos pequeños, actividades poco predecibles (análisis, viabilidad, planificación)
 - Comprende las fases
 - Inicio
 - Elaboración
- **Etapa de Producción**
 - Equipos grandes, actividades predecibles, menos riesgos (programación, pruebas)
 - Comprende las fases
 - Construcción
 - Transición



La vida del Proceso Unificado (iii)

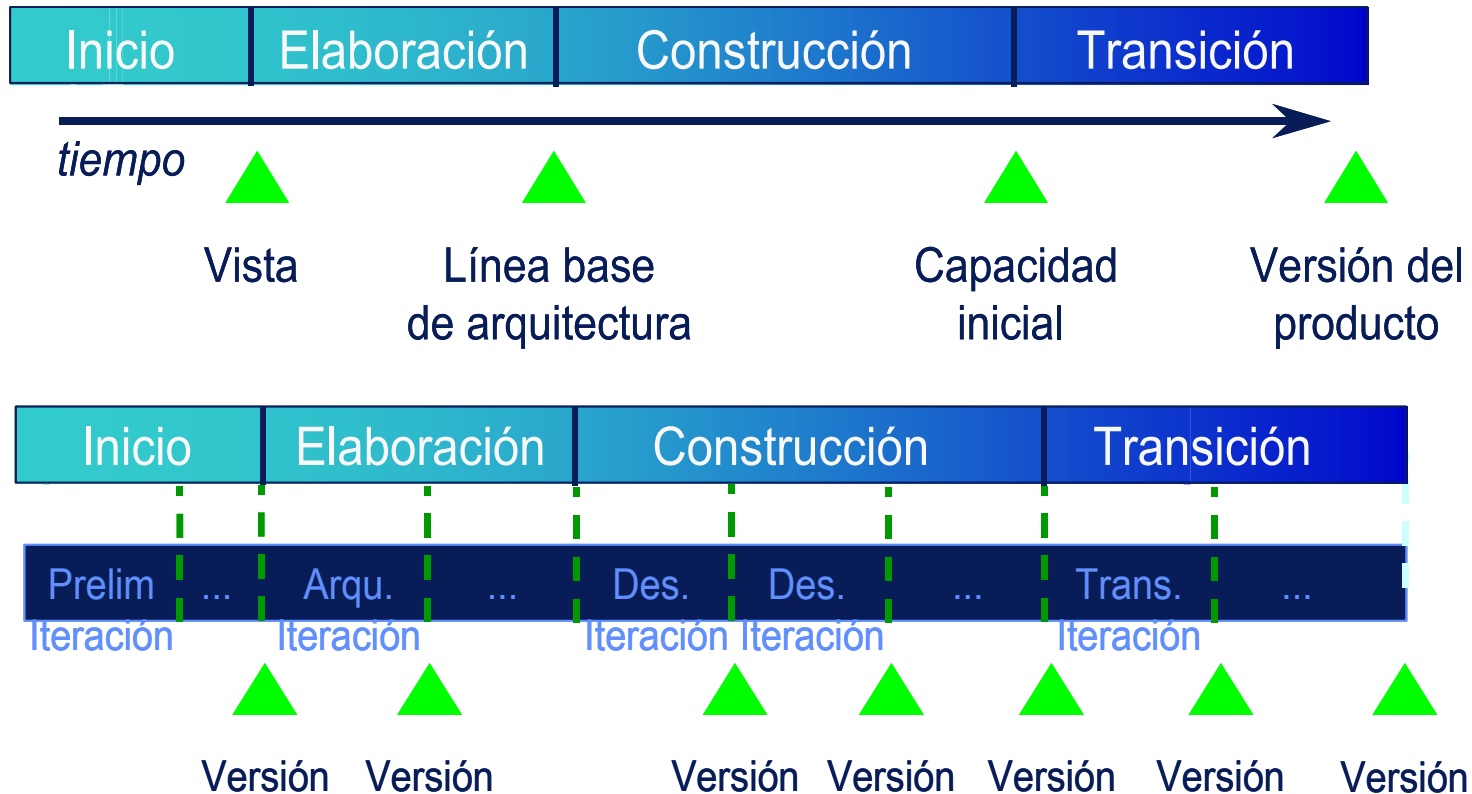
■ Hitos

- Los hitos son puntos de control en los cuales los participantes en el proyecto revisan el progreso del proyecto
- Se pretende
 - Sincronizar las expectativas y la realidad
 - Identificar los riesgos
 - Se evalúa la situación global del proyecto
- Se necesitan
 - Resultados tangibles para comparar con las expectativas
- Varios niveles
 - Hitos principales al final de cada fase
 - Hitos secundarios final de cada iteración

La vida del Proceso Unificado (iv)

- Una **iteración** es una secuencia de actividades con un plan establecido y unos criterios de evaluación, cuyo resultado es una **versión ejecutable no orientada a la entrega** (hito secundario)
- 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
- Las iteraciones discurren a lo largo de las disciplinas

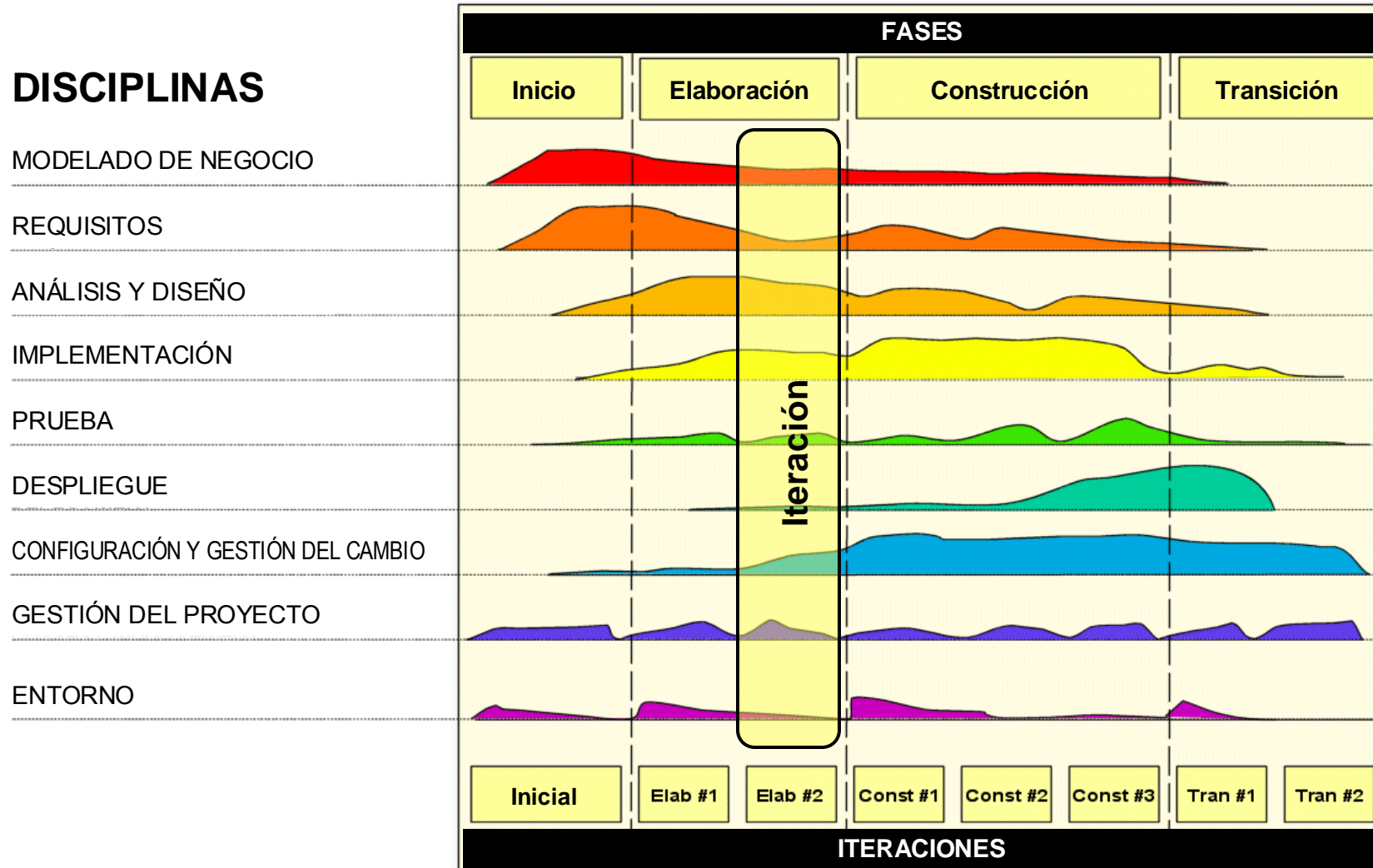
La vida del Proceso Unificado (v)



La vida del Proceso Unificado (vi)

- Las disciplinas o flujos de trabajo organizan las actividades fundamentales de gestión y desarrollo del proyecto
 - **Disciplinas de desarrollo**
 - Requisitos, análisis, diseño, implementación, pruebas...
 - **Disciplinas de gestión o soporte**
 - Gestión de proyecto, gestión de configuraciones, entorno, evaluación...
- Al contrario de lo que ocurre con las fases, las distintas actividades del equipo de desarrollo se pueden solapar en el tiempo

La vida del Proceso Unificado (vii)



https://unsplash.com/photos/m_HRfLhgABo



3. El producto

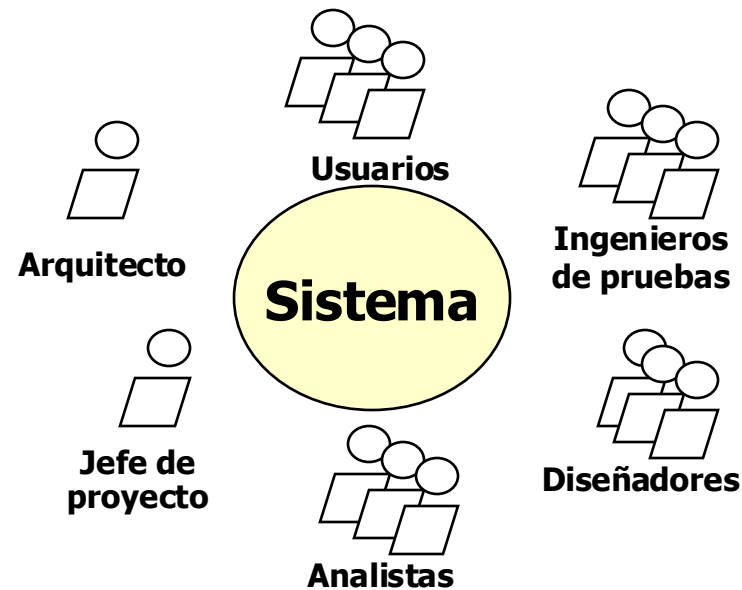
El producto (i)

- El producto que se obtiene es un **sistema de *software***
- El sistema lo componen todos los “artefactos” necesarios para representarlo de forma comprensible
- **Artefacto**
 - Término general para cualquier tipo de información creada, producida, cambiada o utilizada por los *stakeholders* en el desarrollo del sistema. Puede ser
 - De ingeniería
 - De gestión
- El artefacto más importante del Proceso Unificado es el **modelo**
- Un sistema posee una colección de modelos y las relaciones entre ellos

El producto (ii)

Un modelo es una abstracción semánticamente cerrada del sistema

- Los modelos recogen diferentes perspectivas del sistema (perspectivas de todos los *stakeholders*)



El producto (iii)

■ **Modelos**

■ **Modelo de casos de uso**

- Diagramas de casos de uso, secuencia, colaboración y actividad

■ **Modelos de análisis y diseño**

- Diagramas de clases, objetos, secuencia, colaboración y actividad

■ **Modelo de despliegue**

- Diagramas despliegue, secuencia y colaboración

■ **Modelo de implementación**

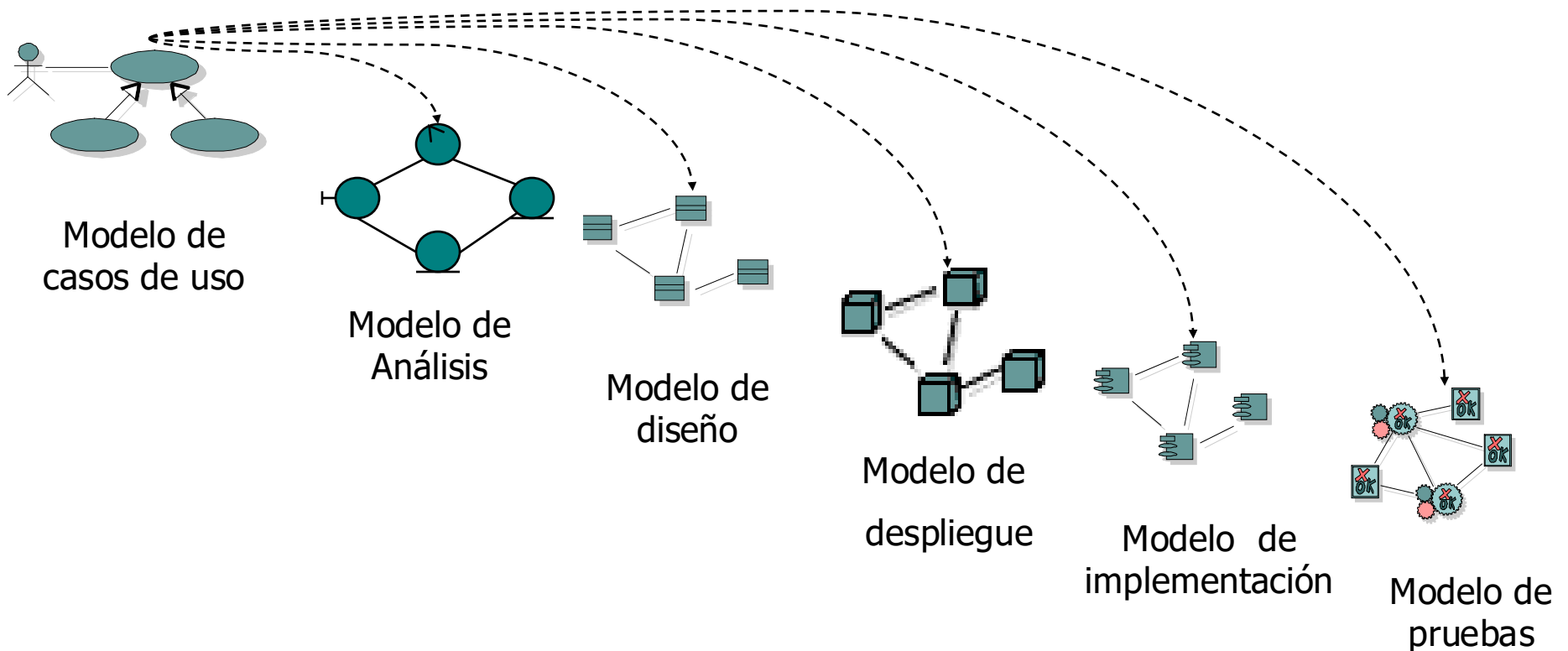
- Diagramas de componentes, secuencia y colaboración

■ **Modelo de pruebas**

- Todos los diagramas

El producto (iv)

- Existen dependencias entre el modelo de casos de uso y los demás modelos



<https://bit.ly/391VQ49>



4. El proceso

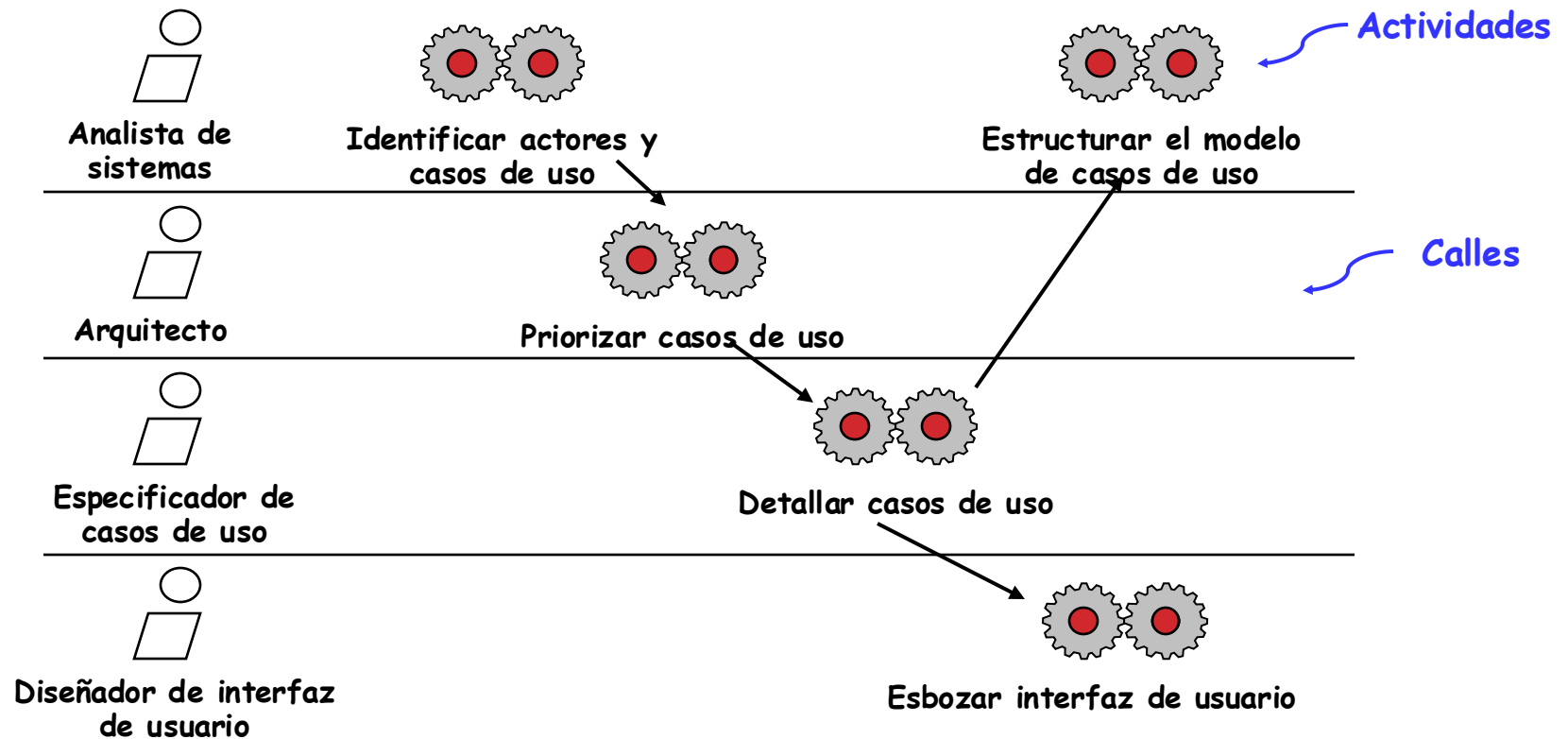
El proceso (i)

El **proceso de desarrollo de software** es una definición de un conjunto completo de actividades necesarias para convertir los requisitos de usuario en un conjunto consistente de artefactos que conforman un producto software, y para convertir los cambios sobre esos requisitos en un nuevo conjunto consistente de artefactos

- El proceso hace referencia a un contexto que sirve como plantilla que pueda reutilizarse para crear instancias de ella (proyectos)
- Las actividades relacionadas conforman **disciplinas** o **flujos de trabajo**
 - Su identificación parte de la identificación de los *stakeholders* y de los artefactos para cada tipo de *stakeholder*
 - Describen como fluye el proceso a través de los *stakeholders*

El proceso (ii)

- Representación de las disciplinas mediante flujos de trabajo



Disciplina del modelado de casos de uso

Características principales del proceso

- **Conducido por casos de uso**
 - Los casos de usos guían el desarrollo del sistema
 - 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 representa mediante vistas del modelo
 - Se puede tomar como arquitectura de referencia el denominado modelo de arquitectura de 4+1 vistas propuesto por Philippe Kruchten (1995)
- **Iterativo e Incremental**
 - En cada iteración se identifican y especifican los casos de uso relevantes, se crea un diseño basado en la arquitectura seleccionada, se implementa el diseño mediante componentes y se verifica que los componentes satisfacen los casos de uso
 - Si una iteración cumple con sus objetivos se pasa a la siguiente
 - En cada iteración se va desarrollando el sistema de forma incremental

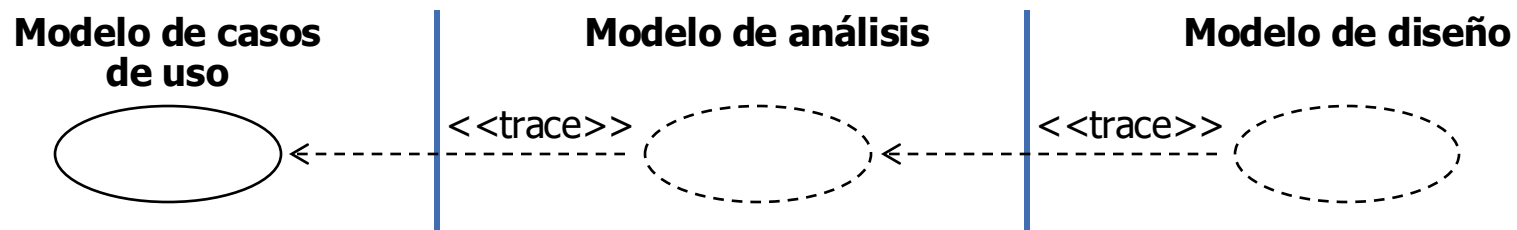
Proceso dirigido por casos de uso (i)

- Dirigen las actividades de desarrollo
 - Creación y validación de la arquitectura del sistema
 - Definición de casos de prueba y procedimientos
 - Planificación de iteraciones
 - Creación de documentación de usuario
 - Despliegue del sistema
- Sincronizan el contenido de los diferentes modelos



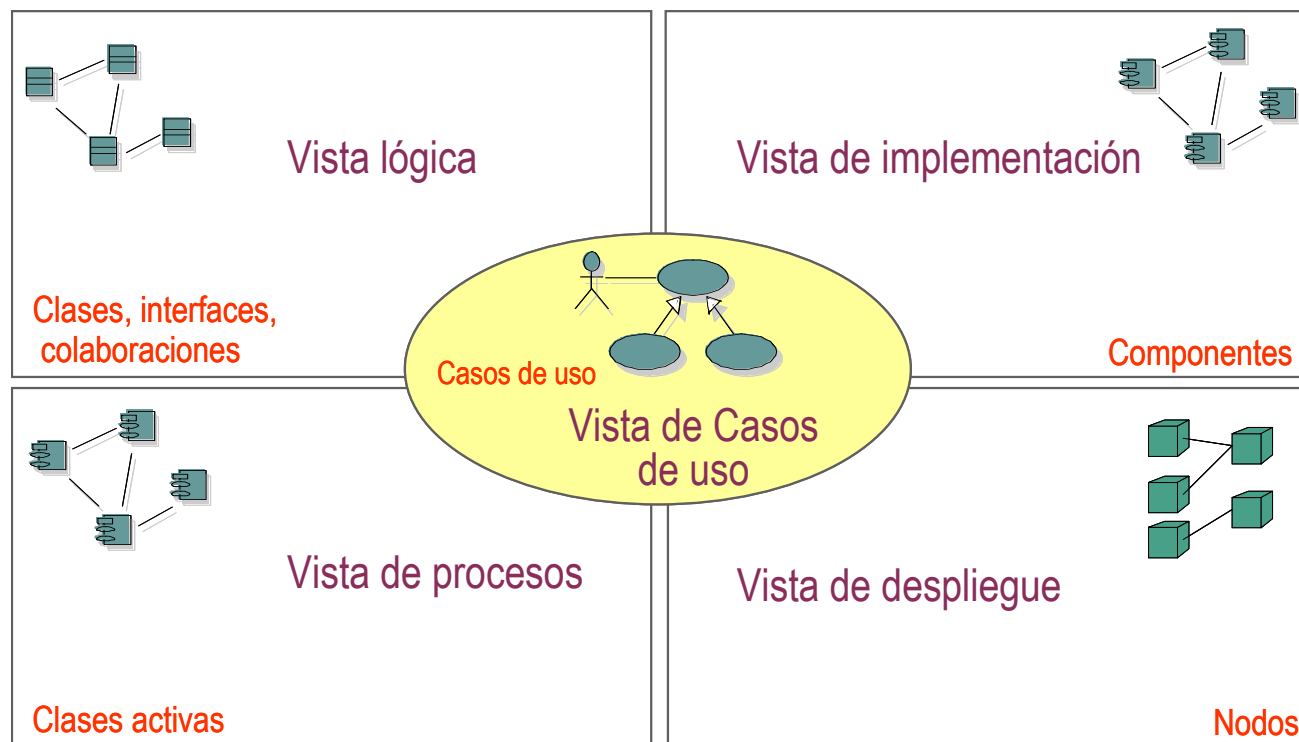
Proceso dirigido por casos de uso (ii)

- Inicialmente los casos de uso se utilizan para la captura de requisitos funcionales
- Durante el análisis y el diseño se transforma el modelo de casos de uso mediante un modelo de análisis en una estructura de clasificadores y **realizaciones de casos de uso**
- En cada iteración, los casos de uso sirven de guía a través del conjunto completo de disciplinas



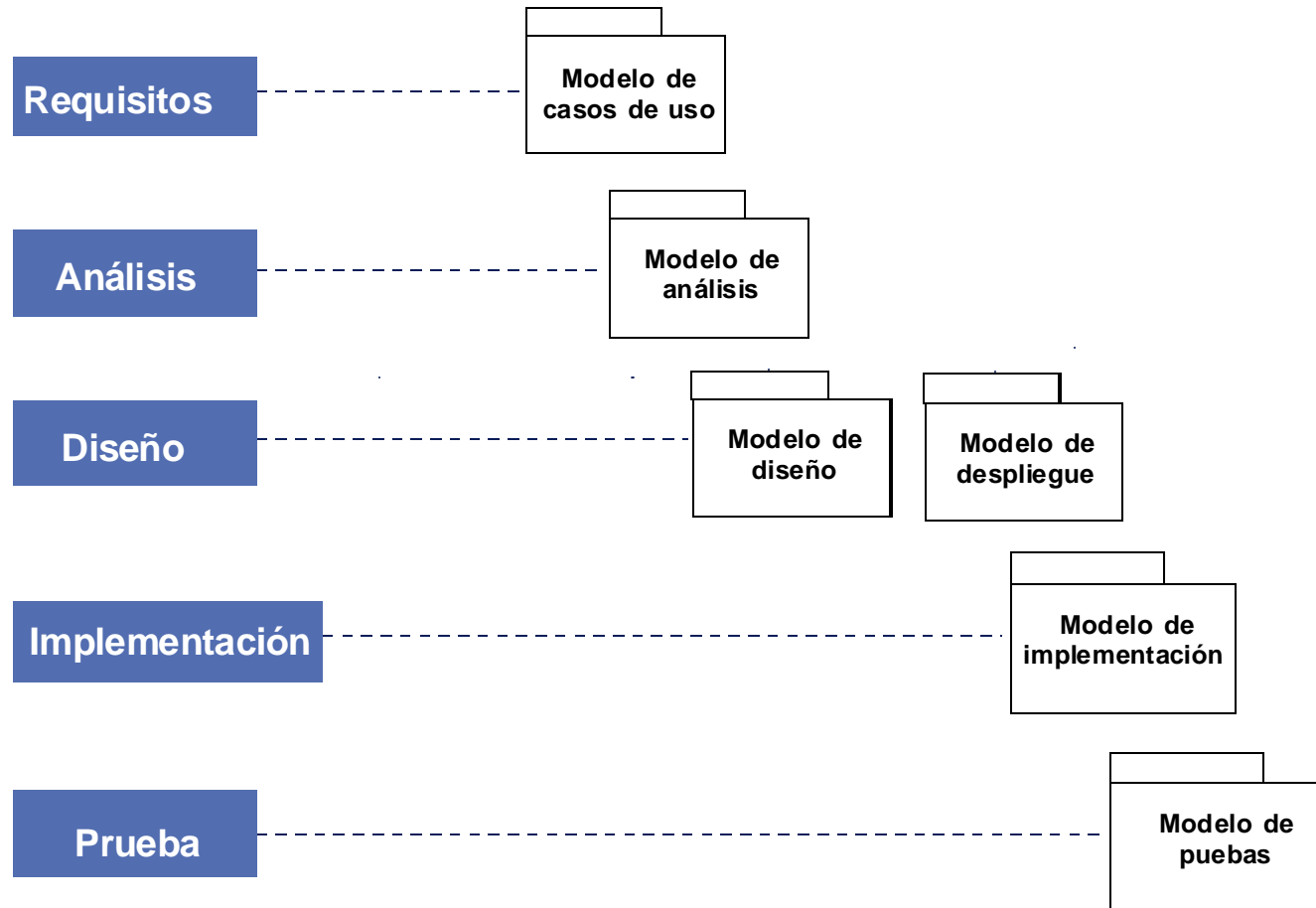
Proceso centrado en la arquitectura (i)

- Se puede tomar como arquitectura de referencia el denominado modelo de arquitectura de 4+1 vistas, propuesto por Philippe Kruchten (1995)
 - Cada vista es una parte de un modelo



Proceso centrado en la arquitectura (ii)

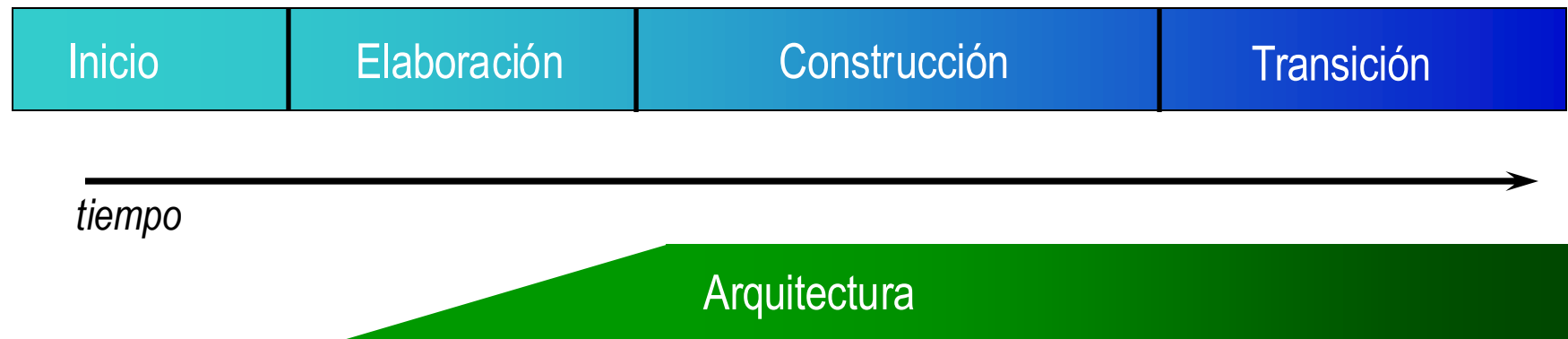
- **Centrado en la arquitectura:** diferentes vistas del sistema



Relación entre los flujos de trabajo y los modelos que forman la arquitectura

Proceso centrado en la arquitectura (iii)

- Centrado en la arquitectura: diferentes vistas del sistema
 - Los **modelos** son los vehículos para visualizar, especificar, construir y documentar la arquitectura
 - El Proceso Unificado prescribe los sucesivos refinamientos de una arquitectura ejecutable



Proceso centrado en la arquitectura (iv)

■ Diseño de la arquitectura

- Seleccionar escenarios: aspectos críticos y riesgos
- Identificar las clases principales y sus responsabilidades
- Distribuir el comportamiento en clases
- Estructurar en subsistemas, capas y definir interfaces
- Definir distribución y concurrencia
- Implementar prototipos de arquitectura
- Derivar casos de prueba a partir de los casos de uso
- Evaluar la arquitectura

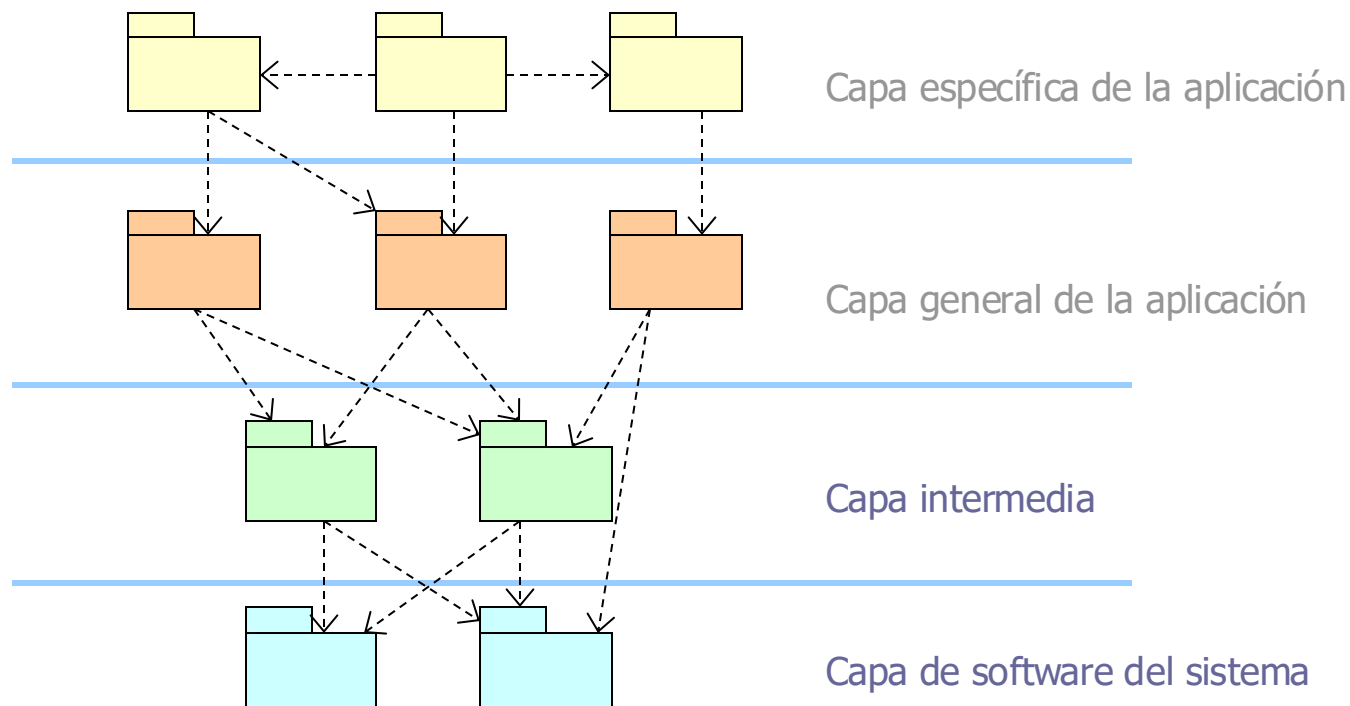
Iterar

■ La arquitectura se desarrolla mediante iteraciones (**en capas**)

- Comienza con una línea base de arquitectura (primera versión de los modelos)
- La línea base evoluciona hasta convertirse en un sistema estable

Proceso centrado en la arquitectura (v)

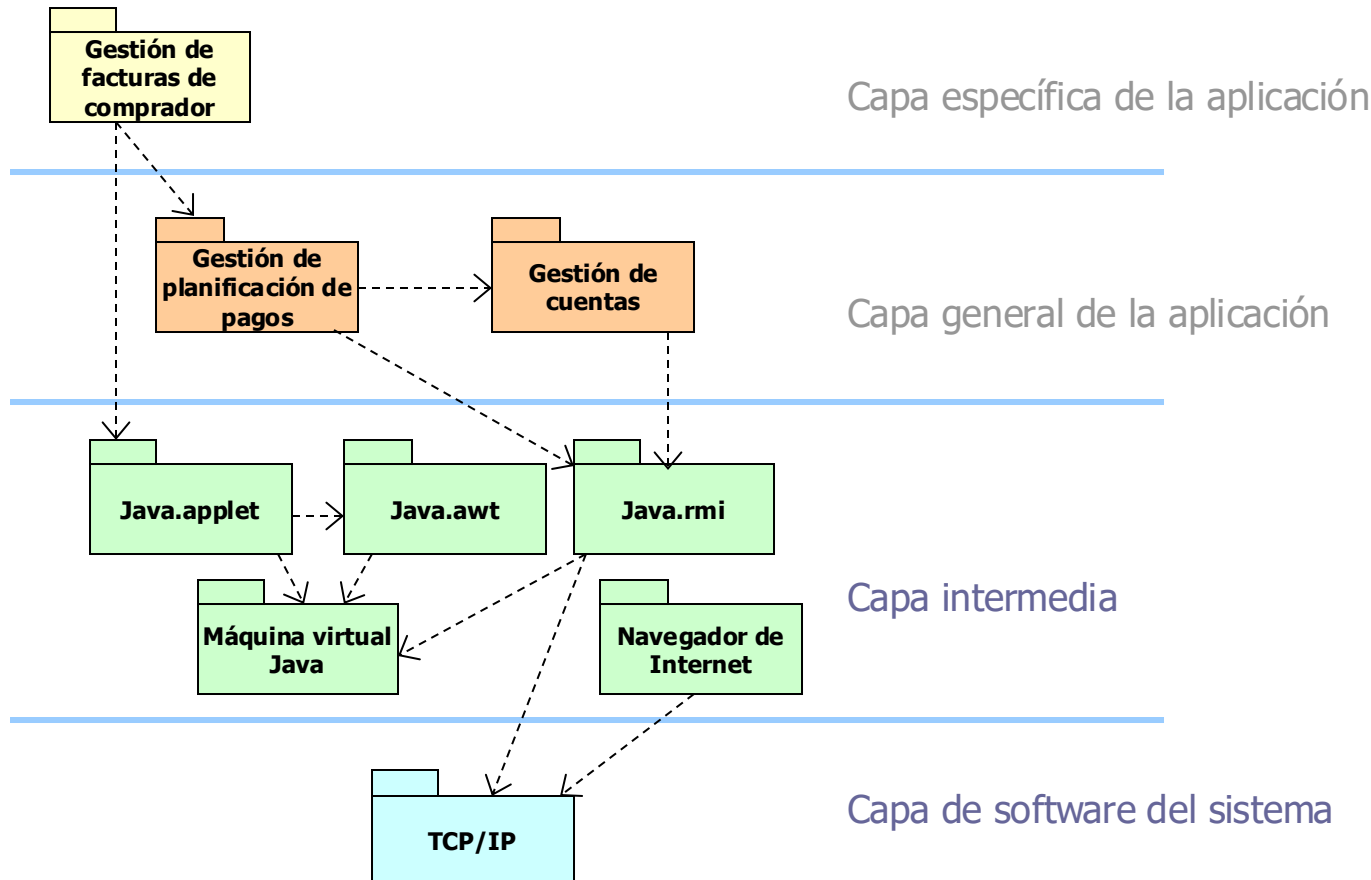
- Diseño de la arquitectura



Patrón de capas de la arquitectura del sistema

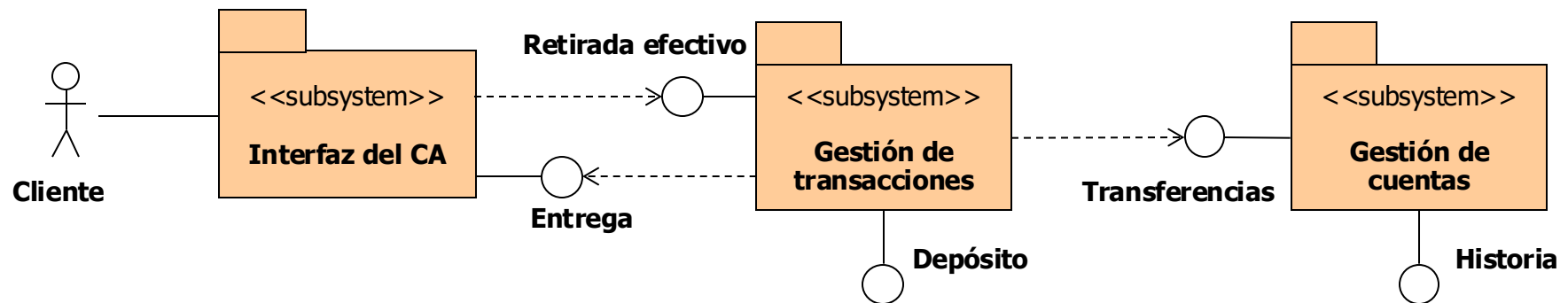
Proceso centrado en la arquitectura (vi)

■ Diseño de la arquitectura

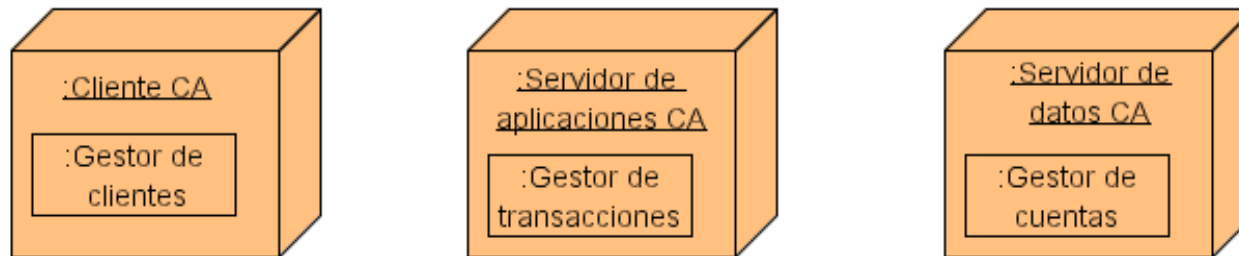


Proceso centrado en la arquitectura (vii)

- Diseño de la arquitectura



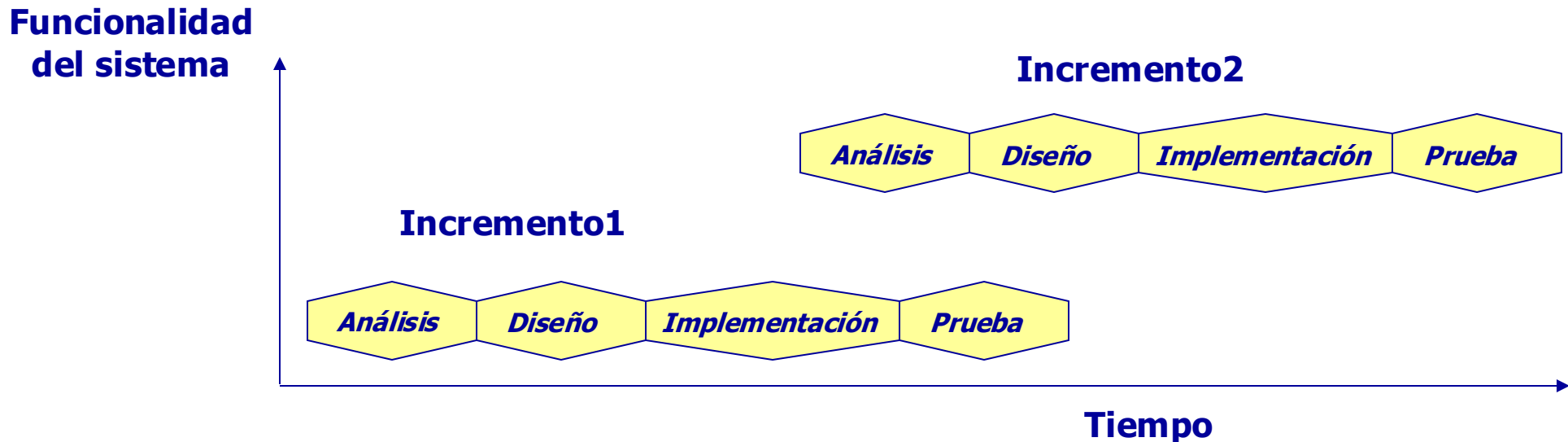
Estructura estática de la arquitectura en el modelo de diseño



Vista arquitectónica del modelo de despliegue

Proceso iterativo e incremental (i)

- La característica fundamental del Proceso Unificado es ser un **proceso iterativo**
 - Se basa en la ampliación y el refinamiento del sistema
 - Una serie de desarrollos cortos (mini proyectos de 2 a 6 semanas, cada iteración reproduce el ciclo de vida a menor escala)
 - No solo se mejora sino que el sistema también crece: proceso iterativo e incremental



Proceso iterativo e incremental (ii)

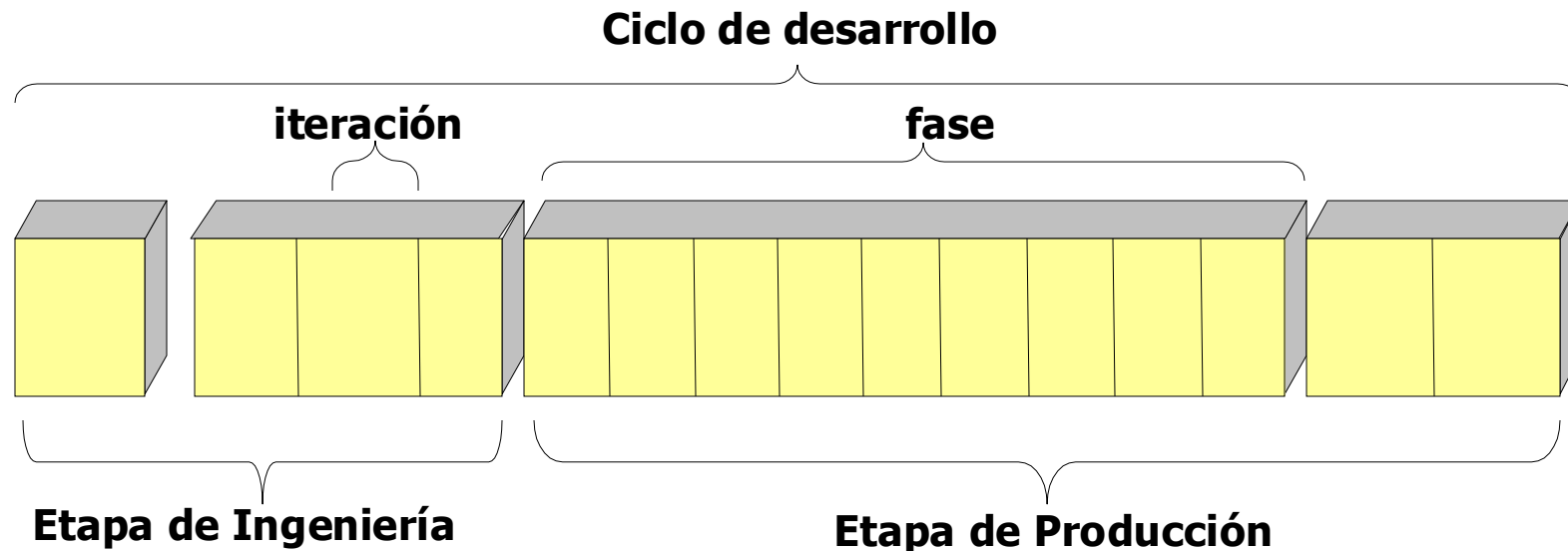
- El resultado de cada iteración es un sistema ejecutable (aunque sea incompleto y no esté listo para su instalación)
- Un sistema instalable requiere varias iteraciones
- Evolución de prototipos ejecutables
- Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes
- Concepto de *time-boxing*
 - Cada iteración debe tener una duración fija (el máximo, 6 meses)
 - En lugar de retrasar el final de una iteración se recomienda eliminar algunos de los requisitos (se dejan para la siguiente iteración)
- La realimentación del usuario es fundamental en este proceso
- El progreso es visible

Proceso iterativo e incremental (iii)

- **Fases**
 - Es preciso diferenciar temporalmente las fases del ciclo de vida
 - La división temporal necesita puntos de control
- **Puntos de control o hitos**
 - Separan las etapas, las fases, las iteraciones
- **Disciplinas o Flujos de trabajo**
 - Organizan las actividades fundamentales de gestión y desarrollo
 - Se pueden solapar en el tiempo
 - El resultado de las actividades de los flujos de trabajo son los artefactos
- **Artefactos**
 - Cualquier tipo de información producida por los desarrolladores de un sistema (diagramas UML, código, ejecutables, casos de prueba...)
 - Se construyen de forma incremental

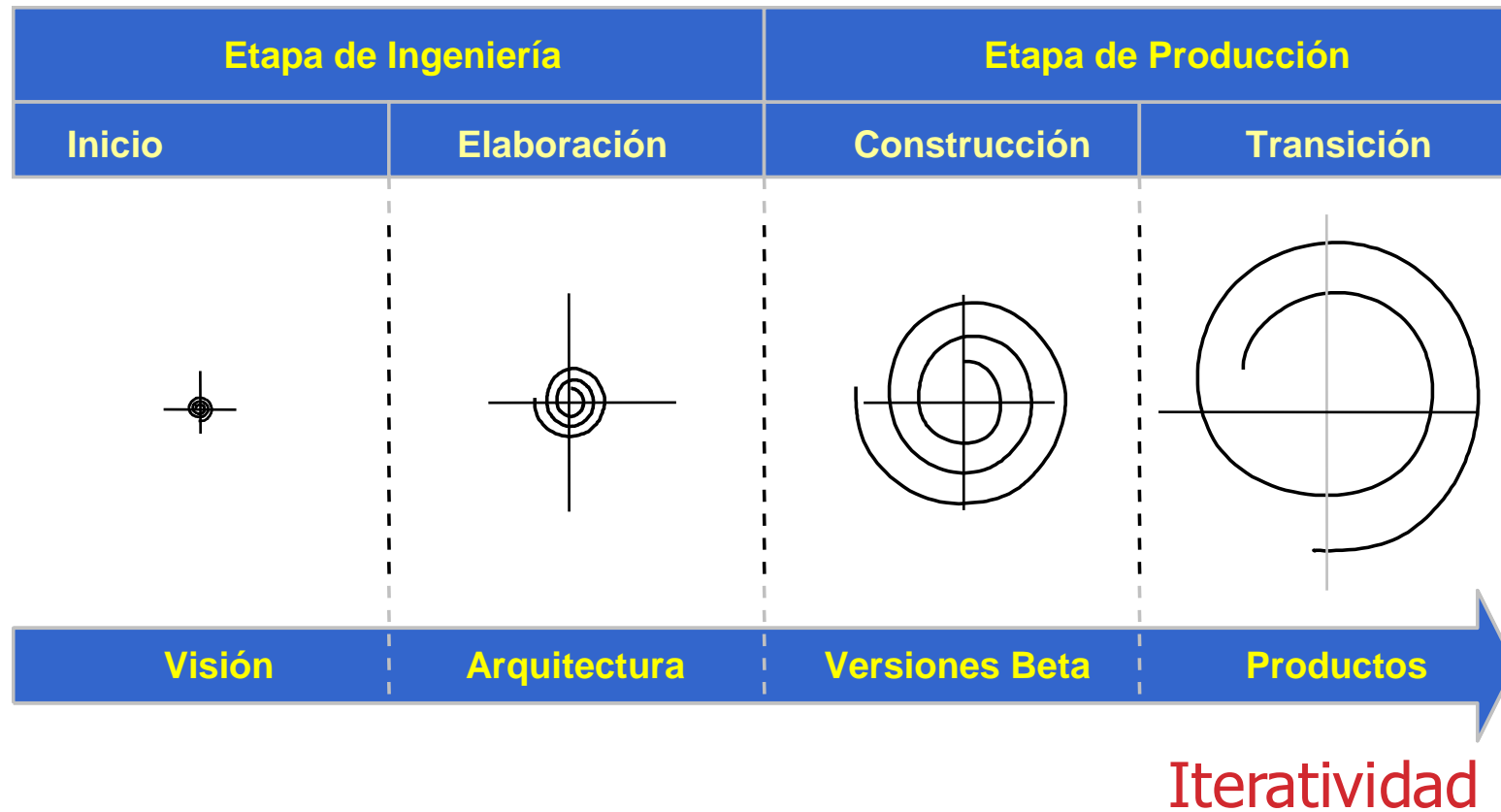
Proceso iterativo e incremental (iv)

- El Proceso Unificado propone una serie de ciclos de desarrollo
 - Hay que separar claramente la etapa de Ingeniería de la etapa de Producción
 - Cada una de las dos grandes etapas se dividen en fases
 - Las fases se dividen en iteraciones

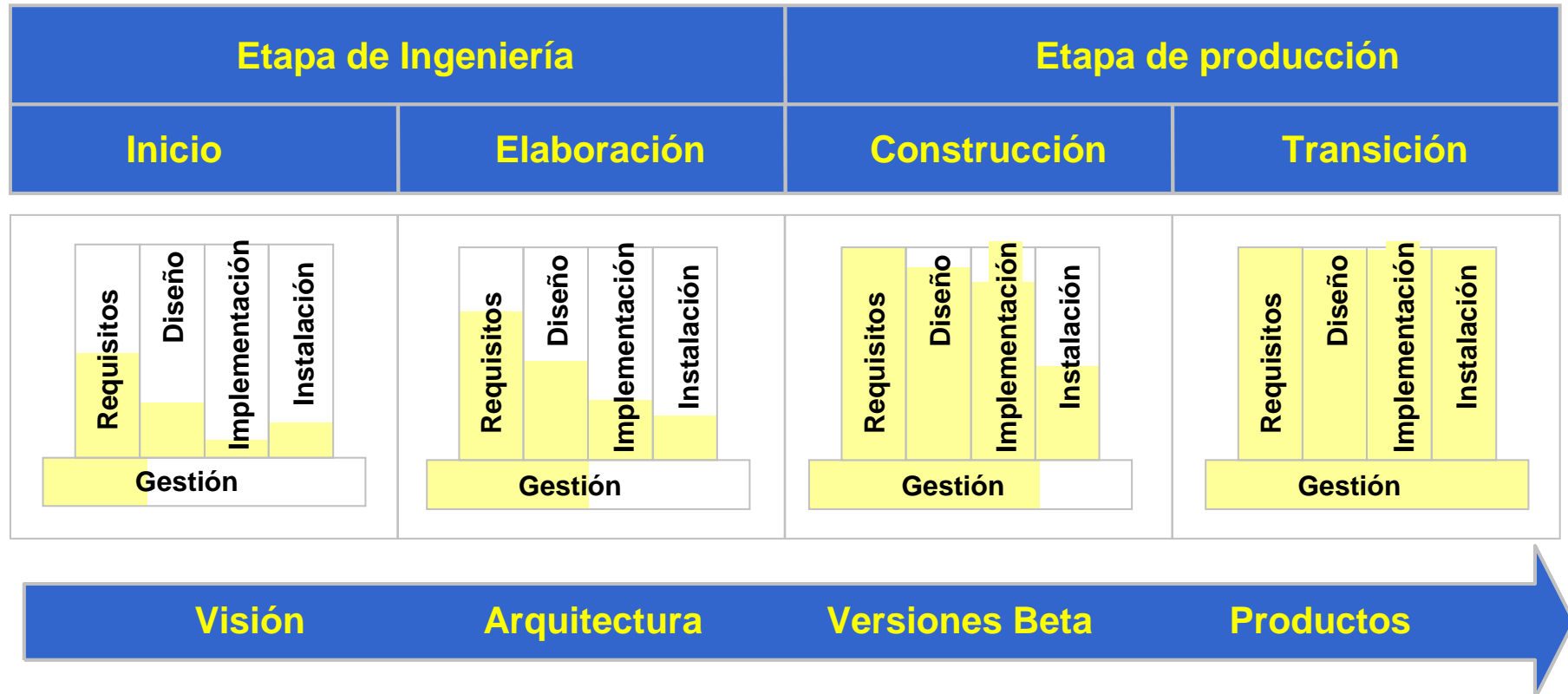


Proceso iterativo e incremental (v)

■ Iterativo e incremental



Proceso iterativo e incremental (vi)



Incremental



https://unsplash.com/photos/0nodNW_qR6s

5. Aportaciones principales del tema

Aportaciones principales

- Proceso Unificado propone y aporta un marco flexible y adaptable para el desarrollo de proyectos tecnológicos
- No existe un único proceso que aplicar como una guía inflexible, es un marco que se puede adaptar a un abanico muy amplio de proyectos
- Las características es un proceso conducido por casos de uso, centrado en la arquitectura y es iterativo e incremental

<https://unsplash.com/photos/o0Qgw2.1-0NI>



6. Lecturas complementarias

Lecturas complementarias

- P. B. Kruchten, "The 4+1 View Model of architecture," *IEEE Software*, vol. 12, no. 6, pp. 42-50, 1995. doi: 10.1109/52.469759. Disponible en: <https://goo.gl/nhzySN>
 - En este artículo se presenta el patrón arquitectónico 4+1 vistas
- Rational Software, "Rational Unified Process. Best practices for software development teams," Rational Software, Cupertino, CA, USA, Rational Software White Paper, TP026B, Rev 11/01, 1998.
 - Buenas prácticas con el Proceso Unificado de Rational

<https://unsplash.com/photos/yKc4YuGMPC4>



7. Referencias

Referencias

- [Booch et al., 1999]** Booch, G., Rumbaugh, J., Jacobson, I. *"El Lenguaje Unificado de Modelado"*. Addison Wesley, 1999
- [Jacobson et al., 2000]** Jacobson, I., Booch, G., Rumbaugh, J. *"El Proceso Unificado de Desarrollo"*, Addison Wesley, 2000
- [Kruchten, 1995]** Kruchten, P. *"The "4+1" View Model of Software Architecture"*. IEEE Software, 12(6):42-50, November 1995
- [OMG, 2017]** OMG. *"OMG Unified Modeling Language Specification. Version 2.5.1"* Object Management Group Inc. December 2017. <https://www.omg.org/spec/UML/2.5.1/> [Última vez visitado, 13-1-2025]

INGENIERÍA DE SOFTWARE I

Tema 5: Introducción al Proceso Unificado

Grado en Ingeniería Informática
Fecha de última modificación: 9-2-2025

Dr. Francisco José García-Peñalvo / fgarcia@usal.es
Dra. Alicia García-Holgado / aliciagh@usal.es
Dra. Andrea Vázquez-Ingelmo / andreavazquez@usal.es



Departamento de Informática y Automática
Universidad de Salamanca

