

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

**Metodología RUP**

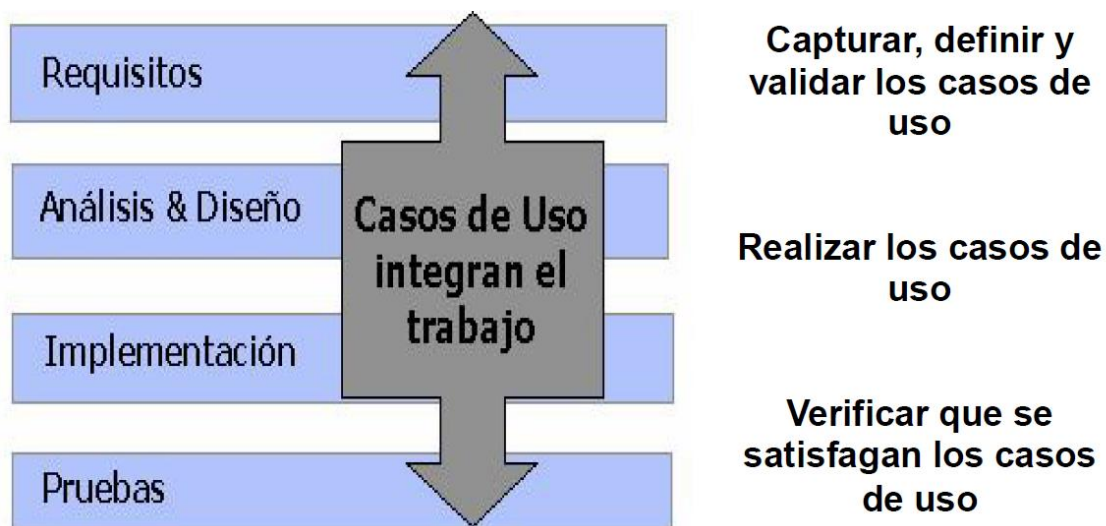
Es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos.

**Proceso dirigido por Casos de Uso**

Según Kruchten Philippe (2000), “los **Casos de Uso** son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar.

Se define un caso de uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema”.

Los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, sino que también guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo.



**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Los casos de uso Inician el proceso de desarrollo y proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Basándose en los casos de uso, se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada caso de uso.

**Proceso centrado en la arquitectura**

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

En el caso de RUP, *además de utilizar los casos de uso para guiar el proceso, se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.*

Existe una interacción entre los casos de uso y la arquitectura, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro.

La arquitectura dentro de RUP se representa en varias vistas. Todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura. Según Kruchten Philippe (1998) “el cual recibe este nombre porque lo forman las vistas lógica, de implementación, de proceso y de despliegue, más la de casos de uso que es la que da cohesión a todas”.

# Ingeniería de Software I

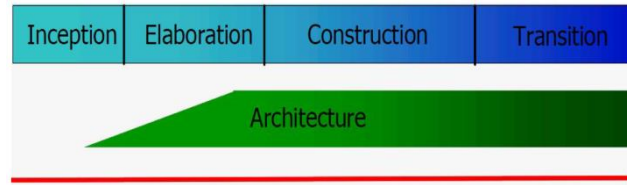
## Clase Número 5

### Metodología Ágil RUP

Dr. Sergio D. Conde

[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)

[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

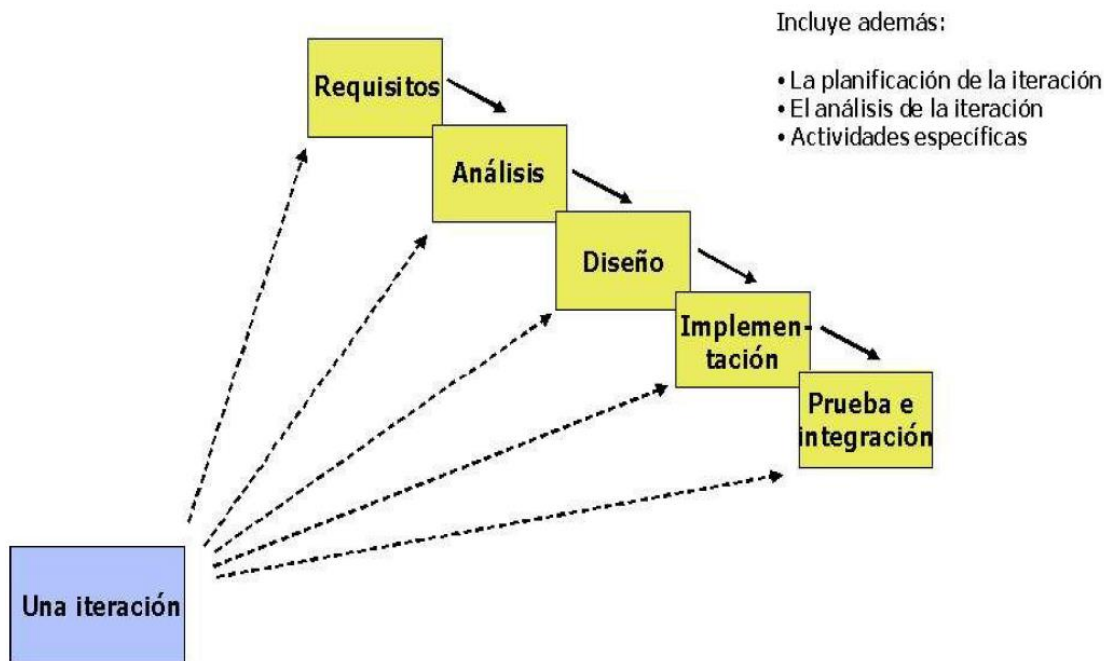


Al final de la fase de elaboración se obtiene una “baseline” (base de referencia) de la arquitectura donde fueron seleccionados una serie de casos de uso arquitectónicamente relevantes (aquellos que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para el usuario y aquellos que cubran las funcionalidades significativas).

### **Proceso iterativo e incremental**

El equilibrio correcto entre los casos de uso y la arquitectura es muy parecido al equilibrio de la forma y la función en el desarrollo de un producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)



Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando se termina. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones (que son una cantidad variable) según el proyecto, y en las que se hace un mayor o menor hincapié en los distintas actividades.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una “baseline” (base de referencia) de la arquitectura.

# Ingeniería de Software I

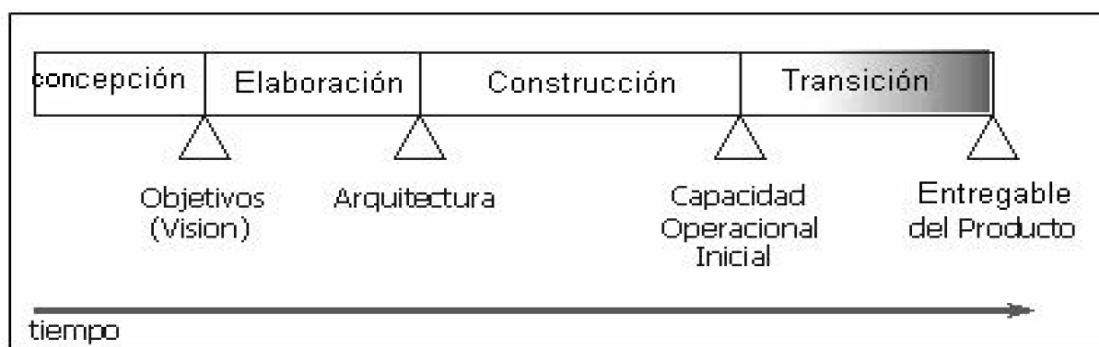
## Clase Número 5

### Metodología Ágil RUP

Dr. Sergio D. Conde  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Para cada iteración se escogen algunos casos de uso, se refina su análisis y diseño, y se procede a su implementación y pruebas. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la **fase de transición** se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios. Como se puede observar, en cada fase participan todas las disciplinas, pero, dependiendo de la fase, es el esfuerzo dedicado a una disciplina.



### Fases

El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales. En cada extremo de una fase se realiza una evaluación para determinar si los objetivos de la fase se han cumplido. Una vez que la evaluación obtiene los resultados deseados, se procede a la siguiente fase.

### Planeando las fases

El ciclo de vida consiste en una serie de ciclos, cada uno de los cuales produce una nueva versión del producto, cada ciclo está compuesto por fases y cada fase está compuesta por un número de iteraciones.

#### 1.- Concepción, Inicio o Estudio de oportunidad

Define el ámbito y objetivos del proyecto, además de la funcionalidad y capacidades del producto.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)



**Analizando la Fase de Inicio tiene las siguientes Actividades:**

- Formular el alcance del Proyecto
- Planificar y preparar un caso de negocio
- Elaborar la arquitectura candidata
- Preparar el ambiente para el Proyecto.(equipos, personas, etc)

**Analizando la Fase de Inicio tiene los siguientes Artefactos:**

- Documento Visión
- Glosario
- Identificación Inicial de riesgos.
- Modelo Inicial de casos de uso

## 2.- Fase de Elaboración

Tanto la funcionalidad como el dominio del problema se estudian a profundidad. Se define una arquitectura básica y se planifica el proyecto considerando recursos disponibles.



**Analizando la Fase de Elaboración tiene las siguientes Actividades:**

- Validar y establecer una arquitectura.
- Refinar la visión
- Crear un plan detallado de iteración.
- Refinar la arquitectura y seleccionar los componentes.
- Refinar el caso de desarrollo.

# Ingeniería de Software I

## Clase Número 5

### Metodología Ágil RUP

Dr. Sergio D. Conde

[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)

[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

**Analizando la Fase de Elaboración tiene los siguientes Artefactos:**

- Prototipo ejecutable de la Arquitectura.
- Infraestructura de desarrollo.
- Modelo de diseño.
- Documento de la arquitectura de software

### 3.- Fase de Construcción

El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación. Las fases de concepción y elaboración sólo dieron una arquitectura básica que es refinada aquí de manera incremental conforme se construye (se permiten cambios en la estructura). Gran parte del trabajo es programación y pruebas, se documenta tanto el sistema construido como el manejo del mismo. En esta fase se hace una documentación junto con el producto.





**Analizando la Fase de Construcción tiene las siguientes Actividades:**

- Desarrollar componentes por completo.
- Optimizar y controlar procesos.
- Administrar los recursos.
- Integrar el sistema.
- Verificar versión del producto con la visión

**Analizando la Fase de Construcción tiene los siguientes Artefactos:**

- Desarrollar el Sistema.
- Suite de Pruebas
- Plan de Publicación.
- Material de Soporte al usuario final.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

**4.- Fase de Transición**

Se libera el producto y se entrega al usuario para un uso real. Se incluyen tareas de mercadotecnia, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.



**Analizando la Fase de Transición tiene las siguientes Actividades:**

- Ejecutar Planes para la publicación.
- Terminar el material de soporte del usuario final.
- Crear una versión del Producto.
- Probar el entregable del producto publicado.
- Obtener una retroalimentación de los usuarios finales.

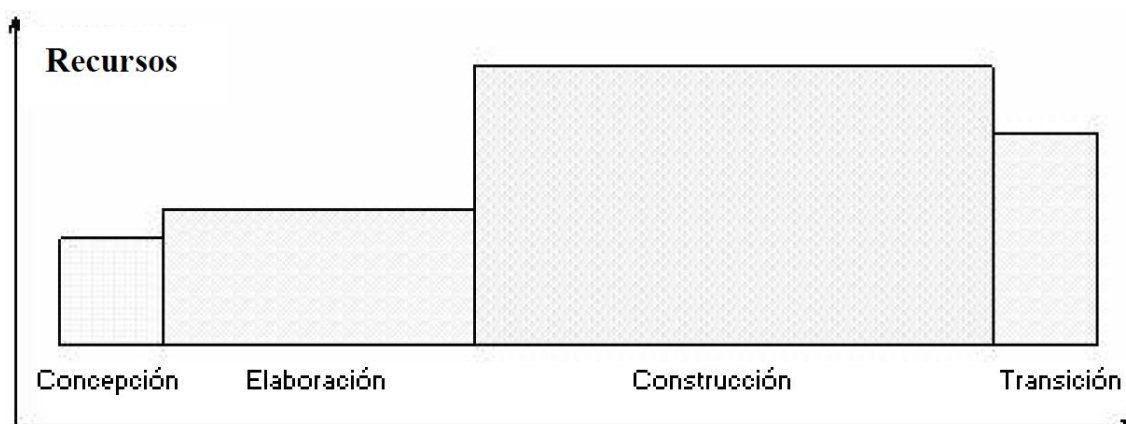
**Analizando la Fase de Transición tiene los siguientes Artefactos:**

- Tener en cuenta el Sistema.
- Elementos Implementados
- Suite de Pruebas
- Material de Soporte al usuario final.

Los manuales de usuario se completan y refinan con la información anterior.

Ninguna fase es idéntica en términos de tiempo y esfuerzo. Aunque esto varía significativamente dependiendo del proyecto, un ciclo de desarrollo inicial típico para un proyecto de tamaño mediano debe anticipar la distribución siguiente del esfuerzo y horario:

	<u><b>Concepción</b></u>	<u><b>Elaboración</b></u>	<u><b>Construcción</b></u>	<u><b>Transición</b></u>
<b>Esfuerzo</b>	~5 %	20 %	65 %	10%
<b>Horario</b>	10 %	30 %	50 %	10%



Conforme un proyecto avanza y se intenta mejorar, llega un momento en que el ciclo (las cuatro fases) debe repetirse

# Ingeniería de Software I

## Clase Número 5

### Metodología Ágil RUP

Dr. Sergio D. Conde

[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)

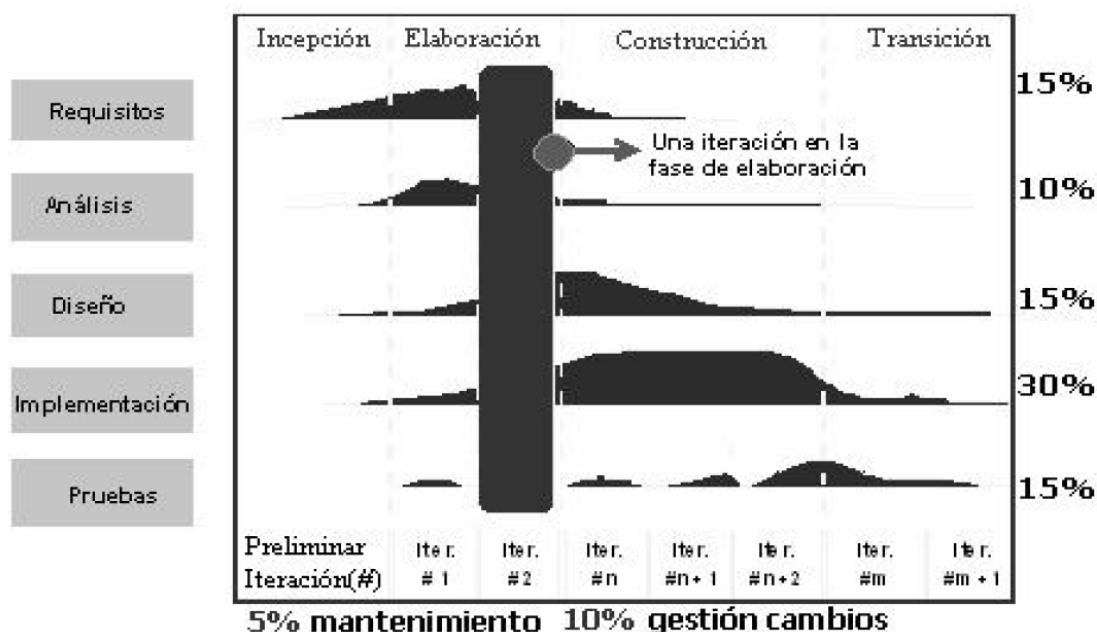
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

. Estos ciclos subsecuentes se llaman los *ciclos de la evolución*. Mientras que el producto pasa durante varios ciclos, se producen las nuevas generaciones.

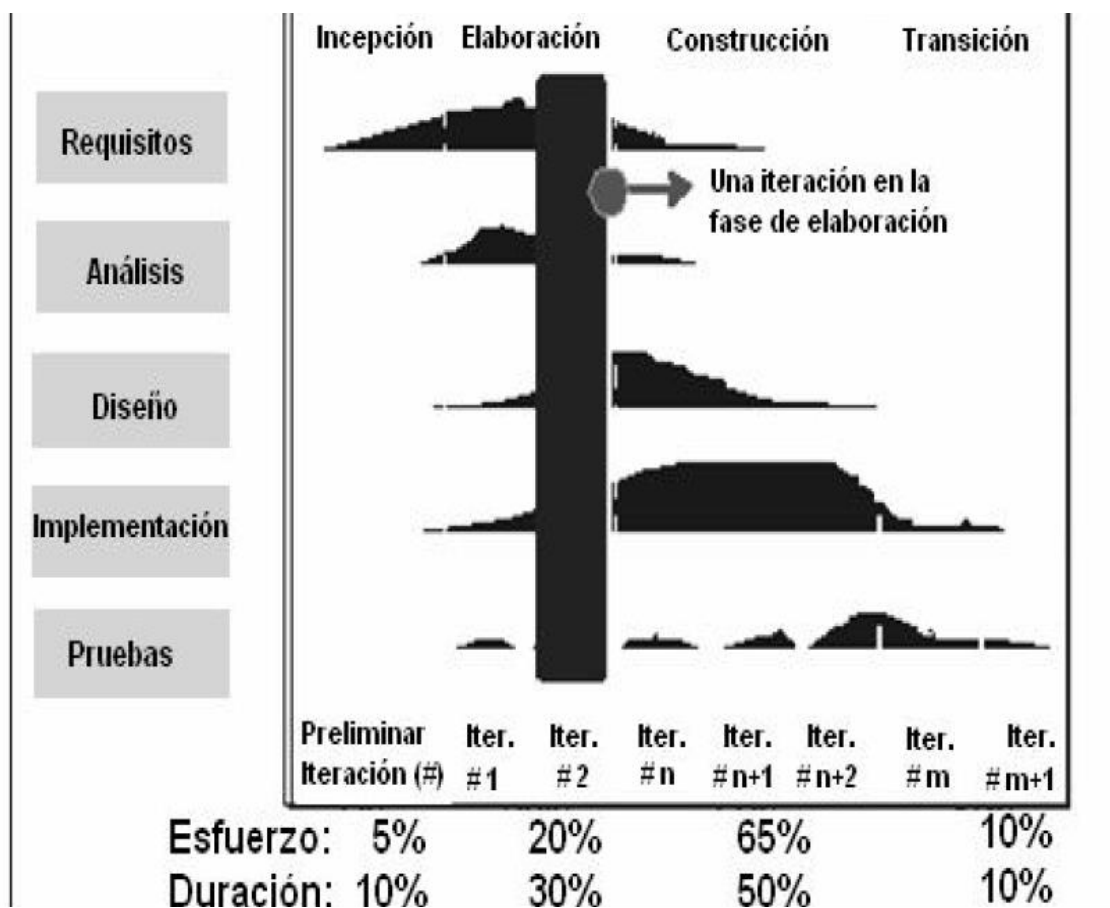
Esfuerzo respecto de los flujos de trabajo (o disciplinas)

En la imagen posterior del documento se muestra el porcentaje el esfuerzo que se tiene que realizar por cada una de las disciplinas o flujos de trabajo, y los dos porcentajes que se muestran de forma horizontal son para todo el proyecto.

Se puede observar que para la obtención de requisitos en la fase de concepción se empiezan a conseguir, en la fase de elaboración tiene su auge, y va declinando en la fase de construcción. Con las demás sucede algo similar en distintas iteraciones.



Esfuerzo respecto de las fases



**Las disciplinas** son los flujos del trabajo, los cuales son una secuencia de pasos para la culminación de cada disciplina, estas disciplinas se dividen en dos grupos: las primarias

# Ingeniería de Software I

## Clase Número 5

### Metodología Ágil RUP

Dr. Sergio D. Conde

[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)

[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

y las de apoyo. Las primarias son las necesarias para la realización de un proyecto de software, aunque para proyectos no muy grandes se pueden omitir algunas; entre ellas se tienen: modelado del negocio, requerimientos, análisis y diseño, implementación, pruebas y despliegue. Las de apoyo son las que complementan y brindan mejoras a las primarias y especifican otras características en la realización de un proyecto de software; entre estas se tienen: entorno, gestión del proyecto, gestión de configuración y cambios.

A continuación veamos cada disciplina:

#### **Modelado del negocio**

Tiene como objetivos comprender la estructura y la dinámica de la organización, comprender problemas actuales e identificar posibles mejoras, comprender los procesos del negocio.

#### **Requerimientos**

Sus objetivos son: establecer lo que el sistema debe hacer, se definen los límites del sistema, y una interfaz de usuario. También realiza una estimación del costo y tiempo de desarrollo.

#### **Análisis y diseño**

Define la arquitectura del sistema y tiene como objetivos trasladar requisitos en especificaciones de implementación, al decir análisis se refiere a transformar CU (casos de uso) en clases, y al decir diseño se refiere a refinar el análisis para poder implementar los diagramas de clases de análisis de cada CU, los diagramas de colaboración de cada CU, el de clases de diseño de cada CU, el de secuencia de diseño de CU, el de estados de las clases, etc.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

### **Implementación**

Tiene como objetivos implementar las clases de diseño como componentes, asignar los componentes a los nodos, probar los componentes individualmente (pruebas unitarias) e integrar los componentes en un sistema ejecutable.

### **Pruebas**

Verificar la integración de los componentes (prueba de integración), verificar que todos los requisitos han sido implementados (pruebas del sistema), asegurar que los defectos detectados han sido resueltos antes de la distribución.

### **Despliegue**

Sus objetivos son asegurar que el producto está preparado para el cliente, para proceder a su entrega y recepción por el cliente. En esta disciplina se realizan las actividades de probar el software en su entorno final (Prueba Beta), empaquetarlo, distribuirlo e instalarlo, así como la tarea de enseñar al usuario.

### **Gestión y configuración de cambios**

Éste es esencial para controlar el número de artefactos producidos por la cantidad de personal que trabajan en un proyecto conjuntamente. Los controles sobre los cambios son de mucha ayuda ya que evitan confusiones costosas, como la compostura de algo que ya se había arreglado.

### **Entorno**

Esta disciplina se enfoca sobre las actividades necesarias para configurar el proceso que engloba el desarrollo de un proyecto y describe las actividades requeridas para el desarrollo de las pautas que apoyan un proyecto.

Su propósito es proveer a la organización que desarrollará el software, un ambiente en el cual basarse, el cual provee procesos y herramientas para poder desarrollar el software.



**Organización y elementos del RUP**

Ya conociendo varias partes del RUP nos concentraremos ahora en los elementos que lo componen, entre estos se tienen: flujos de trabajo, detalle de los flujos de trabajo, actores, actividades (o acciones) y artefactos.

**Actores o roles**

Son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP

- ✓ Analistas
- ✓ Analista del Proceso del Negocio.
- ✓ Diseñador del Negocio.
- ✓ Revisor del Modelo del Negocio.
- ✓ Revisor de Requerimientos.
- ✓ Analista del Sistema.
- ✓ Especificador de Casos de Uso.
- ✓ Diseñador de Interfaz del Usuario.

**Desarrolladores**

- Arquitecto.
- Revisor de la Arquitectura.
- Diseñador de Cápsulas.
- Revisor del Código y Revisor del Diseño.
- Diseñador de la Base de Datos.
- Diseñador.
- Implementador y un Integrador.



**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

**Probadores Profesionales**

- Diseñador de Pruebas.
- Probador.

**Encargados**

- Encargado de Control del Cambio.
- Encargado de la Configuración.
- Encargado del Despliegue.
- Ingeniero de Procesos.
- Encargado de Proyecto.
- Revisor de Proyecto.

**Otros**

- Cualquier trabajador.
- Artista Gráfico.
- Stakeholder.
- Administrador del Sistema.
- Escritor técnico.
- Especialista de Herramientas.

**Desventajas de la metodología:**

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágil RUP**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

No capturar un caso de uso a tiempo puede causar una reconstrucción parcial de la arquitectura.

La falta de especificaciones en los requisitos por parte del usuario puede causar ciertas discrepancias.

Se requiere mucho personal (dependiendo del proyecto).

**Ventajas de la metodología:**

Es el proceso de desarrollo más general de los existentes actualmente.

Es una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo).

Reutilización

El diseñador piensa en términos del comportamiento de objetos y no en detalles de bajo nivel

Confiabilidad, Integridad y Estabilidad.

Mantenimiento más sencillo. Modificaciones locales.