

**TALLER 1**

**PARCIAL No. 2**

## **METODOLOGIAS DE DESARROLLO EN SOFTWARE**

METODOLOGIAS DE DESARROLLO EN SOFTWARE

ÁREA DE COMPUTACION

DEPARTAMENTO DE ITIN

## TALLER 1 SEGUNDO PARCIAL.

**Nombres del estudiante:**

Cristian Jesus Becerra Loaiza

Jhon Kevin Castillo Quishpe

Ismael Alejandro Silva Flores

**Nivel:** Tercer Nivel

**NRC:** 29022

**Asignatura:** Met. de Desarrollo en Software

**Nombre del profesor:** Ing. Jenny Alexandra Ruiz Robalino

METODOLOGIAS DE DESARROLLO EN SOFTWARE

ÁREA DE COMPUTACION

DEPARTAMENTO DE ITIN

## Contenido

<b>Taller #1_P2.....</b>	<b>2</b>
<span style="color: orange;">✍ Actividad 1: Exploración Guiada. ....</span>	
1. El Rational Unified Process (RUP) organiza el desarrollo en cuatro fases: .....	2
2. Disciplinas de desarrollo vs disciplinas de soporte.....	3
3. Ventajas del enfoque iterativo e incremental .....	4
<span style="color: lightblue;">💬 Actividad 2: Debate Crítico (15 min).....</span>	
<span style="color: green;">💡 Actividad 3: Aplicación Práctica (10 min) .....</span>	
	7

## Taller #1\_P2

### ✍ Actividad 1: Exploración Guiada.

**Modalidad:** Individual

**Objetivo:** Leer y comprender los fundamentos del RUP.

#### **Instrucciones:**

1. Lee el siguiente fragmento del Capítulo IX sobre RUP.

2. Extrae:

1. Las **4 fases** principales del RUP.

2. 2 diferencias entre **disciplinas de desarrollo** y **disciplinas de soporte**.

3. Ventajas del enfoque iterativo e incremental.

#### **1. El Rational Unified Process (RUP) organiza el desarrollo en cuatro fases:**

##### **Inicio:**

Se define la visión del sistema, los objetivos del negocio y el alcance. Aquí predominan las disciplinas como Modelado del negocio y Gestión del proyecto, porque se busca entender el contexto organizacional y planificar el trabajo.

##### **Elaboración:**

Se construye la arquitectura base y se detallan los requisitos críticos. Las disciplinas más activas son Requisitos, Análisis y diseño, y Gestión de configuración, ya que se establecen las bases técnicas y se controlan los cambios.

**Construcción:**

Se desarrolla el producto mediante Implementación, Pruebas y Despliegue parcial. Aquí el esfuerzo se concentra en codificar, integrar y validar componentes funcionales.

**Transición:**

Se entrega el sistema a los usuarios, se realizan pruebas beta y ajustes. La disciplina Despliegue cobra relevancia junto con Gestión del proyecto para asegurar una transición ordenada.

## **2. Disciplinas de desarrollo vs disciplinas de soporte**

**Desarrollo:**

Estas disciplinas están directamente relacionadas con la creación del producto de software. Se enfocan en transformar los requisitos en un sistema funcional.

**Modelado del negocio:** Define procesos y reglas del negocio para entender el contexto.

**Requisitos:** Captura y especifica lo que el sistema debe hacer.

**Análisis y diseño:** Define la estructura del sistema.

**Implementación:** Codificación y construcción de componentes.

**Pruebas:** Verificación y validación del software.

**Despliegue:** Instalación y entrega del sistema al usuario.

**Soporte:**

Estas disciplinas no crean el producto directamente, pero garantizan que el proceso sea ordenado, controlado y eficiente.

**Gestión de configuración y cambios:** Controla versiones y cambios.

**Gestión de proyectos:** Planificación, seguimiento y control del proyecto.

### **3. Ventajas del enfoque iterativo e incremental**

- Cada iteración permite validar requisitos antes de avanzar.
  - Cambios en requisitos se incorporan sin rehacer todo el proyecto.
  - Los usuarios prueban versiones parciales, con opción a una retroalimentación, o poder realizar alguna corrección.
  - Se liberan funcionalidades útiles en cada ciclo..
- 

 **Actividad 2: Debate Crítico (15 min)**

**Modalidad:** Grupos de 4 estudiantes

**Objetivo:** Evaluar la aplicabilidad del RUP en proyectos reales.

Instrucciones:

#### **1. Analiza el siguiente caso práctico:**

Una empresa mediana necesita desarrollar un sistema de gestión académica en 6 meses. Tiene equipos limitados y no cuenta con experiencia previa en RUP. 2. Responde en grupo: -

¿Aplicarías RUP en este caso? ¿Por qué? - ¿Qué fases o disciplinas serían prioritarias? - ¿Qué riesgos ?

### **2.1.¿Aplicarías RUP en este caso? ¿Por qué?**

El uso del Proceso Unificado Racional (RUP) podría aplicarse, pero con adaptaciones importantes. Aunque RUP es un proceso robusto que organiza el desarrollo en fases y permite gestionar proyectos complejos mediante iteraciones, el contexto del caso presenta algunas limitaciones relevantes.

#### **Argumentos a favor**

- RUP estructura el desarrollo mediante un enfoque iterativo e incremental, útil para proyectos donde se requiere validar funcionalidades de forma progresiva.
- La orientación a casos de uso permite modelar claramente las interacciones académicas del sistema.
- Su fuerte enfoque en arquitectura puede garantizar una base técnica sólida desde etapas tempranas.

#### **Argumentos en contra**

- El plazo de 6 meses es reducido para aplicar RUP en su forma completa.
- El equipo no tiene experiencia previa, lo que genera una curva de aprendizaje significativa.
- RUP demanda roles, artefactos y documentación que podrían ser excesivos para un equipo pequeño.

- La adaptación incorrecta del proceso puede provocar retrasos y sobrecarga administrativa.

## **2.2;Qué fases o disciplinas serían prioritarias?**

RUP se compone de cuatro fases principales, todas presentes en cada iteración. Sin embargo, debido al tiempo limitado, se debe priorizar lo esencial.

### **Fases prioritarias**

- Inicio: Definir el alcance del sistema, el caso de negocio y los requisitos más importantes.
- Elaboración: Comprender el problema, analizar requisitos, establecer la arquitectura e identificar riesgos.
- Construcción: Implementar componentes funcionales, probarlos y entregar versiones incrementales.
- Transición: Realizar pruebas finales, capacitar usuarios y desplegar el sistema.

### **Disciplinas Prioritarias**

- Requirements (Requisitos): Esencial para definir funcionalidades y evitar cambios tardíos.
- Analysis & Design (Análisis y Diseño): Permite establecer la arquitectura base del sistema.
- Implementation (Implementación): Desarrollo iterativo y construcción del software.
- Test (Pruebas): Garantiza la calidad en cada incremento.
- Project Management (Gestión del Proyecto): Control de tiempos, recursos y riesgos.

## **2.3.Riesgos al usar RUP (Resumen)**

### **Riesgos técnicos:**

- Exceso de documentación.
- Arquitecturas innecesariamente complejas.
- Sobrecarga de actividades en cada iteración.

**Riesgos de equipo:**

- Falta de experiencia en RUP.
- Resistencia al cambio.
- Sobrecarga de roles por equipo pequeño.

**Riesgos de proyecto:**

- Retrasos por curva de aprendizaje.
- Alcance mal controlado.
- Disminución de calidad por presión de tiempo. odrían surgir al usar RUP?

---

☞ **Actividad 3: Aplicación Práctica (10 min)**

**Modalidad:** Individual

**Objetivo:** Aplicar los conceptos de RUP a un proyecto propio.

**Instrucciones:**

Realiza las actividades del capítulo 9.

**Junta los objetivos con la correspondiente fase del Proceso Unificado de Desarrollo.**

**Respuesta:**

1. Lograr versiones útiles (alfa, beta y otras versiones de prueba) tan rápido como sea práctico	A. Inicio/Elaboración/Construcción/Transición
2. Lograr que las partes interesadas estén de acuerdo en que las líneas de base de implementación están completas	B. Inicio/Elaboración/Construcción/Transición
3. Discriminar los casos de uso críticos del sistema, que son los escenarios principales de operación que impulsarán las principales compensaciones de diseño	C. Inicio/Elaboración/Construcción/Transición
4. Lograr la autosuficiencia del usuario.	D. Inicio/Elaboración/Construcción/Transición

Lograr versiones útiles (alfa, beta y otras versiones de prueba) tan rápido como sea práctico	<b>Construcción</b>
Lograr que las partes interesadas estén de acuerdo en que las líneas de base de implementación están completas	<b>Transición</b>
Discriminar los casos de uso críticos del sistema, que son los escenarios principales de operación que impulsarán las principales compensaciones de diseño	<b>Elaboración</b>
Lograr la autosuficiencia del usuario	<b>Transición</b>

### **Justificación:**

### **Construcción**

El texto establece que en la fase de Construcción se "desarrolla, diseña, programa y prueba el producto software", entregando al final "un producto software funcional". Esta descripción se alinea directamente con el objetivo de producir versiones útiles y ejecutables (alfa, beta) de forma iterativa y rápida (Jacobson, 2000).

### **Transición**

La fase de Transición define el "funcionamiento del producto software en un ambiente real". Para ello, es un requisito previo que las partes interesadas validen y acuerden que las líneas de base de implementación están completas y que el producto está listo para su despliegue (Jacobson, 2000).

### **Elaboración**

En la fase de Elaboración, uno de sus objetivos centrales es "establecer la arquitectura del producto software" y los entregables incluyen el "modelo de requisitos" y los "diagramas de casos de uso". Esto implica identificar y priorizar los casos de uso críticos que guiarán las decisiones de diseño y arquitectura (Jacobson, 2000).

### **Transición**

La fase de Transición considera aspectos como "la capacitación y el entrenamiento del personal", lo cual está directamente orientado a lograr que los usuarios finales sean autosuficientes en el uso del sistema antes de que concluya el proyecto (Jacobson, 2000).

**Seleccione la opción correcta. El Proceso Unificado de Desarrollo es. Seleccione una:**

**Metodología para el desarrollo de software que define claramente: quién, cómo, cuándo y qué debe hacerse en el proyecto**



Programa para desarrollar software con poca documentación, que permite el cambio ágil dentro del proyecto

Herramienta que permite el desarrollo de software avanzado, sin necesidad de datos específicos.

Metodología ágil para el desarrollo de software

#### **Justificación:**

El texto describe RUP como un proceso que tiene una "perspectiva estática que presenta las actividades del proceso que se establecen" y disciplinas como "administración de proyectos". Esto implica que define roles, actividades, tiempos y entregables de manera estructurada (Jacobson, 2000) (Somerville, 2011).

**Seleccione las opciones correctas. El Proceso Unificado de Desarrollo se basa en las siguientes características fundamentales: Seleccione una o más de una:**

Exige poca documentación

**Dirigido por casos de uso**

Proceso secuencial

**Iterativo e incremental**

**Centrado en la arquitectura**

Centrado en el diseño

#### **Justificación:**

**Dirigido por casos de uso:** El texto afirma de manera explícita que el RUP "está dirigido por los casos de uso" (Jacobson, 2000).

**Iterativo e incremental:** Se indica claramente que el proceso es "iterativo e incremental" y que su ciclo de vida consiste en "una serie de iteraciones o ciclos cortos" (Jacobson, 2000).

**Centrado en la arquitectura:** El texto señala que el proceso está "centrado en la



arquitectura" y que el objetivo de la fase de Elaboración es "establecer la arquitectura del producto software" (Jacobson, 2000).

**Una metodología de desarrollo de software es un conjunto de técnicas y \_\_\_\_\_ en fases para el desarrollo de \_\_\_\_\_, de manera eficaz, y abarca el \_\_\_\_\_ del mismo. Es una colección \_\_\_\_\_ para la resolución de una clase de problemas.**

**Las metodologías de desarrollo de software descomponen el proceso en actividades**

**Respuesta:**

- 1** procedimientos organizados
- 2** productos software
- 3** ciclo de vida
- 4** métodos

**Justificación:**

Estos términos completan la definición proporcionada en el texto, que forma parte de la explicación general de las metodologías de desarrollo (Somerville, 2011).

**En las siguientes oraciones complete con el término correspondiente:**

\_\_\_\_\_ es un conjunto de técnicas y procedimientos organizados en fases para el desarrollo de productos software, de manera eficaz, y abarca el ciclo de vida del mismo.

\_\_\_\_\_ es una técnica repetible para la resolución de un problema específico.

\_\_\_\_\_ es un conjunto de reglas gráficas o textuales para representar un modelo Proceso

**Metodología**

**Notación**

## Método

### Justificación:

Estas definiciones están basadas directamente en los conceptos presentados en el texto para diferenciar entre Metodología, Método y Notación (Jacobson, 2000) (Somerville, 2011).

### Seleccione los términos correctos. RUP tiene dos dimensiones:

1. El eje _____ representa _____ y muestra los aspectos del _____ del proceso a medida que se desarrolla en iteraciones.	A. Horizontal/el tiempo/ ciclo de vida/
2. 1. El eje _____ representa _____ como requisitos, análisis y diseño, implementación, que lógicamente agrupan _____ por afinidad	B. Vertical/las disciplinas/las actividades/

### Respuesta:

El eje *horizontal* representa *el tiempo* y muestra los aspectos del *ciclo de vida* del proceso a medida que se desarrolla en iteraciones. → A. Horizontal/el tiempo/ciclo de vida/

El eje *vertical* representa *las disciplinas* como requisitos, análisis y diseño, implementación, que lógicamente agrupan *las actividades* por afinidad.

→ B. Vertical/las disciplinas/las actividades/

### Justificación:

En el texto se menciona que RUP tiene una "perspectiva dinámica" (eje horizontal-tiempo-fases) y una "perspectiva estática" (eje vertical-disciplinas-actividades). Las fases (Inicio, Elaboración, Construcción, Transición) transcurren a lo largo del tiempo (eje horizontal), mientras que las disciplinas o flujos de trabajo (requisitos, análisis, diseño, etc.) agrupan actividades (eje vertical) (Jacobson, 2000).

**Empareje las disciplinas con su definición.**

1. Pone el sistema a disposición de los usuarios finales	A. Entrega/Implementación/Modelamiento/Gestión de Proyecto/Prueba/Entorno/Entrega/Gestión de la configuración.
2. Administrar el acceso a los artefactos del proyecto y controla y gestiona los cambios	B. Entrega/Implementación/Modelamiento/Gestión de Proyecto/Prueba/Entorno/Entrega/Gestión de la configuración
3. Comprende el negocio y el dominio del problema y presenta una solución viable	C. Entrega/Implementación/Modelamiento/Gestión de Proyecto/Prueba/Entorno/Entrega/Gestión de la configuración.
4. Gestionar riesgos y dirige y coordinar personas	D. Entrega/Implementación/Modelamiento/Gestión de Proyecto/Prueba/Entorno/Entrega/Gestión de la configuración
5. Asegura la calidad verifica que los requisitos se cumplan	E. Entrega/Implementación/Modelamiento/Gestión de Proyecto/Prueba/Entorno/Entrega/Gestión de la configuración
6. Asegurar que el equipo cuente con lo necesario, orientación y herramientas adecuados	F. Entrega/Implementación/Modelamiento/Gestión de Proyecto/Prueba/Entorno/Entrega/Gestión de la configuración
7. Transforma los modelos en código fuente	G. Implementación/Modelamiento/Gestión de Proyecto/Prueba/Entorno/Entrega/Gestión de la configuración.

**Respuesta:**

1. Pone el sistema a disposición de los usuarios finales→ Entrega
2. Administrar el acceso a los artefactos del proyecto y controla y gestiona los cambios→ Gestión de la configuración

3. Comprende el negocio y el dominio del problema y presenta una solución viable→ Modelamiento (se refiere a Modelado de Negocio o de Requisitos)
4. Gestionar riesgos y dirige y coordina personas→ Gestión de Proyecto
5. Asegura la calidad verifica que los requisitos se cumplan→ Prueba
6. Asegurar que el equipo cuente con lo necesario, orientación y herramientas adecuados→ Entorno
7. Transforma los modelos en código fuente→ Implementación

### **Justificación:**

Estas disciplinas se mencionan en el texto como parte de los flujos de trabajo de RUP:

- *Modelado del negocio, requisitos, análisis y diseño, implementación, pruebas, instalación* (disciplinas centrales) (Jacobson, 2000).
- *Administración de la configuración y cambios, administración de proyectos y ambiente* (disciplinas de soporte) (Jacobson, 2000).

### **Referencias**

- Jacobson, I. &. (2000). *El proceso unificado de desarrollo de software* (primera edición). . Madrid, España: S.A., Pearson Educación. :  
[https://www.academia.edu/11946867/El\\_Proceso\\_Unificado\\_de\\_desarrollo\\_de\\_software?sm=b](https://www.academia.edu/11946867/El_Proceso_Unificado_de_desarrollo_de_software?sm=b)
- Somerville, I. (2011). *Software Engineering* 9. . Software Engineering 9. : <https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/index.html>