

Principios de Análisis Informático

Tema 2: El proceso unificado de desarrollo de software

Eduardo Mosqueira Rey



LIDIA
Laboratorio de Investigación y
desarrollo en Inteligencia Artificial



Departamento de Computación
Universidade da Coruña, España



Índice



- 1. Características del proceso unificado**
- 2. Fases del proceso unificado**
- 3. Disciplinas del proceso unificado**
- 4. Artefactos del proceso unificado**
- 5. Herramientas**
- 6. Casos de estudio**



Índice



1. Características del proceso unificado

- El proceso unificado vs. el proceso unificado de Rational**
- Fases, iteraciones y disciplinas**
- Características básicas**



Características del proceso unificado



- **El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado (Unified Process o UP) es un marco de desarrollo software iterativo e incremental.**
- **El Proceso Unificado vs. el Proceso Unificado de Rational**
 - **El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational (RUP).**
 - **El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos.**
 - **De la misma forma, el Proceso Unificado de Rational también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP.**
 - **Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.**
 - **RUP es una marca registrada de IBM por lo que suele evitarse cuando se utiliza desde fuera de IBM o Rational Software Corporation (comprada por IBM en 2003) y se prefiere el término Proceso Unificado**



Características del proceso unificado



- **Refinamientos más conocidos del UP**
 - **Agile Unified Process (AUP)**, una variación ligera desarrollada por Scott W. Ambler
 - **Basic Unified Process (BUP)**, una variación ligera desarrollada por IBM y precursora del OpenUP
 - **Enterprise Unified Process (EUP)**, una extensión del Rational Unified Process
 - **Essential Unified Process (EssUP)**, una variación ligera desarrollada por Ivar Jacobson
 - **Open Unified Process (OpenUP)**, el proceso de desarrollo software seguido por el Eclipse Process Framework
 - **Rational Unified Process (RUP)**, el proceso de desarrollo de IBM / Rational Software
 - **Rational Unified Process-System Engineering (RUP-SE)**, una versión del RUP desarrollado por Rational Software para la ingeniería de sistemas



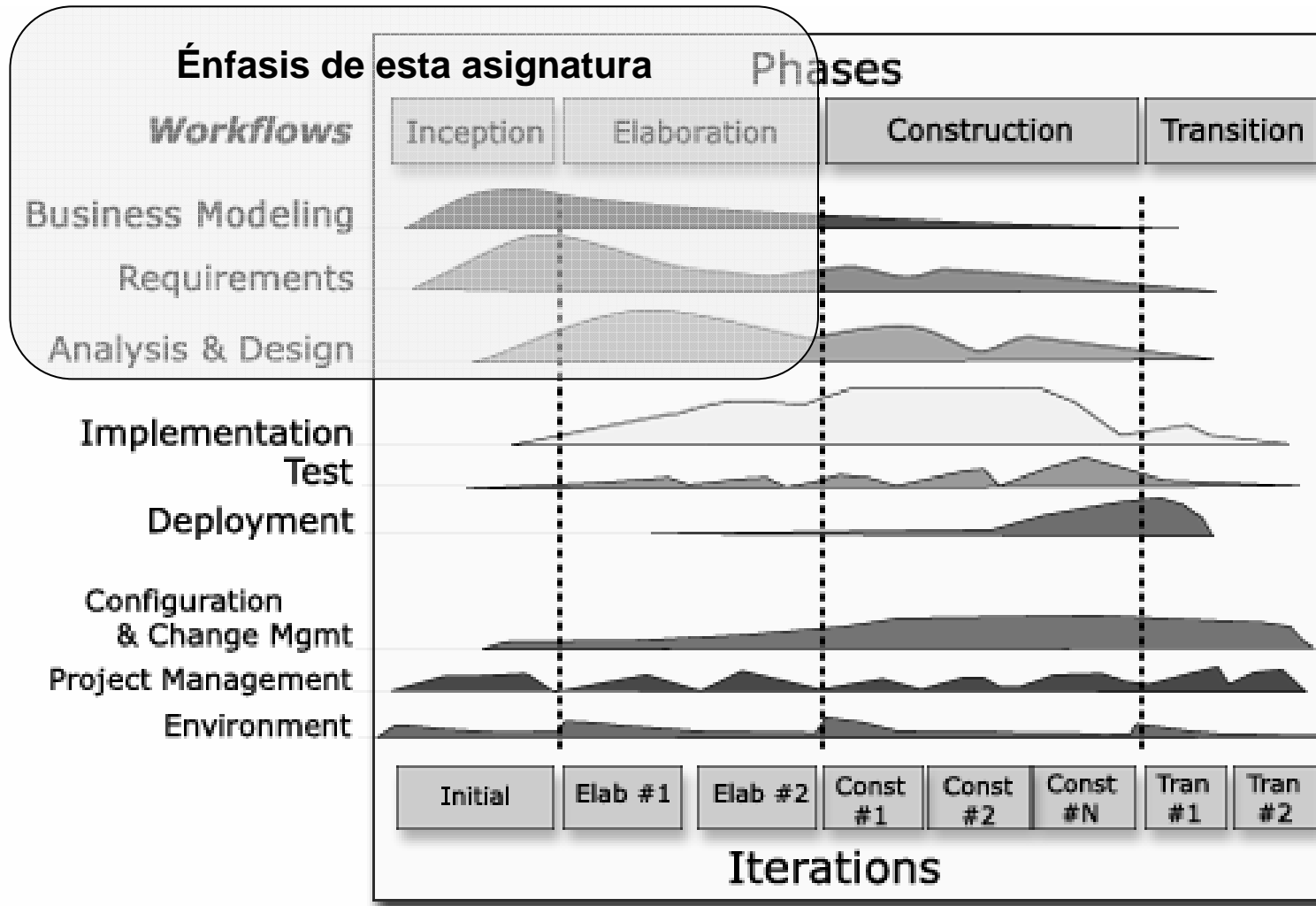
Características del proceso unificado



- **Características básicas:**
 - El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición.
 - Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes).
 - Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.
 - Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación, Prueba, etc.
 - Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

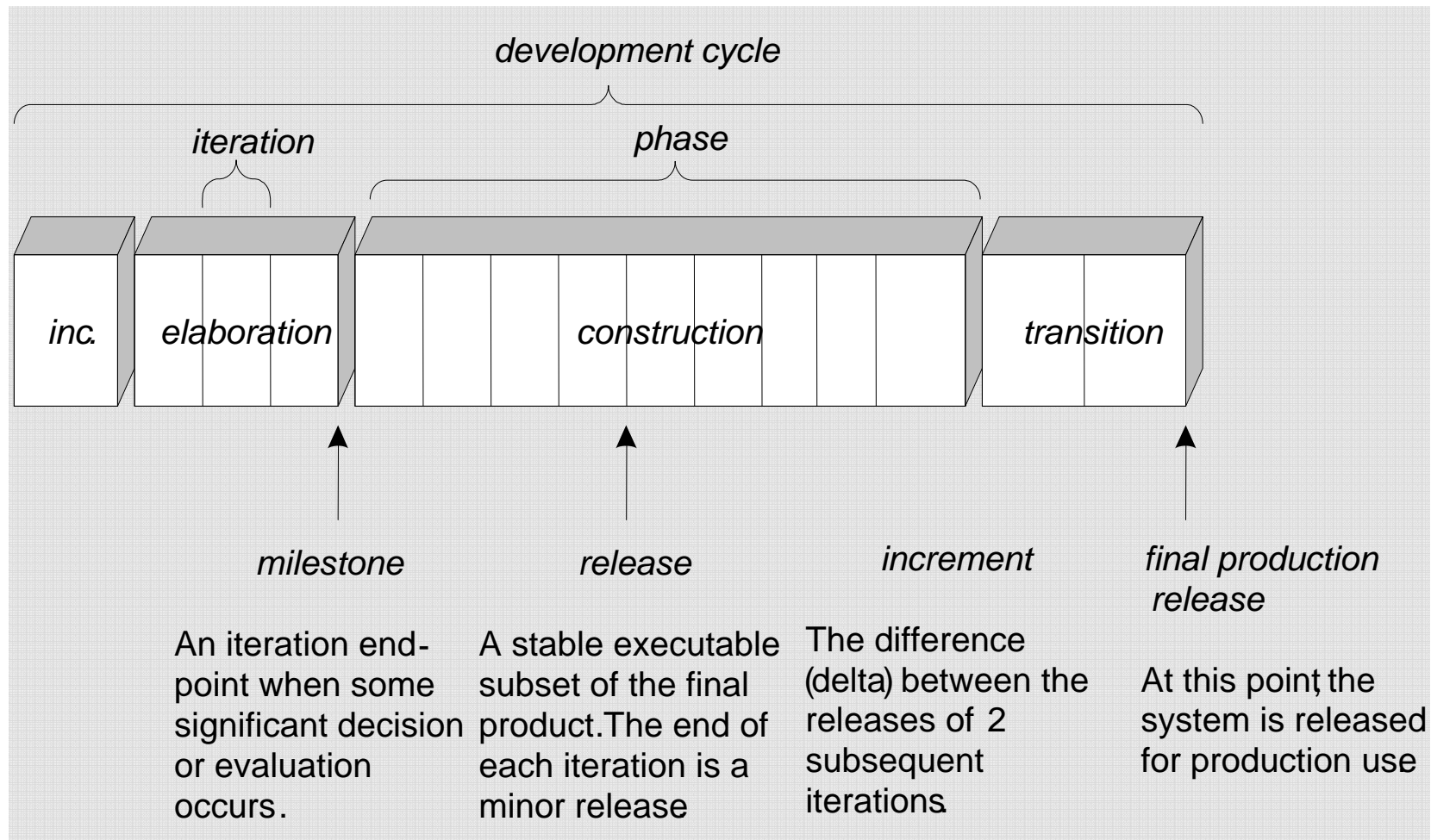


Características del proceso unificado





Características del proceso unificado

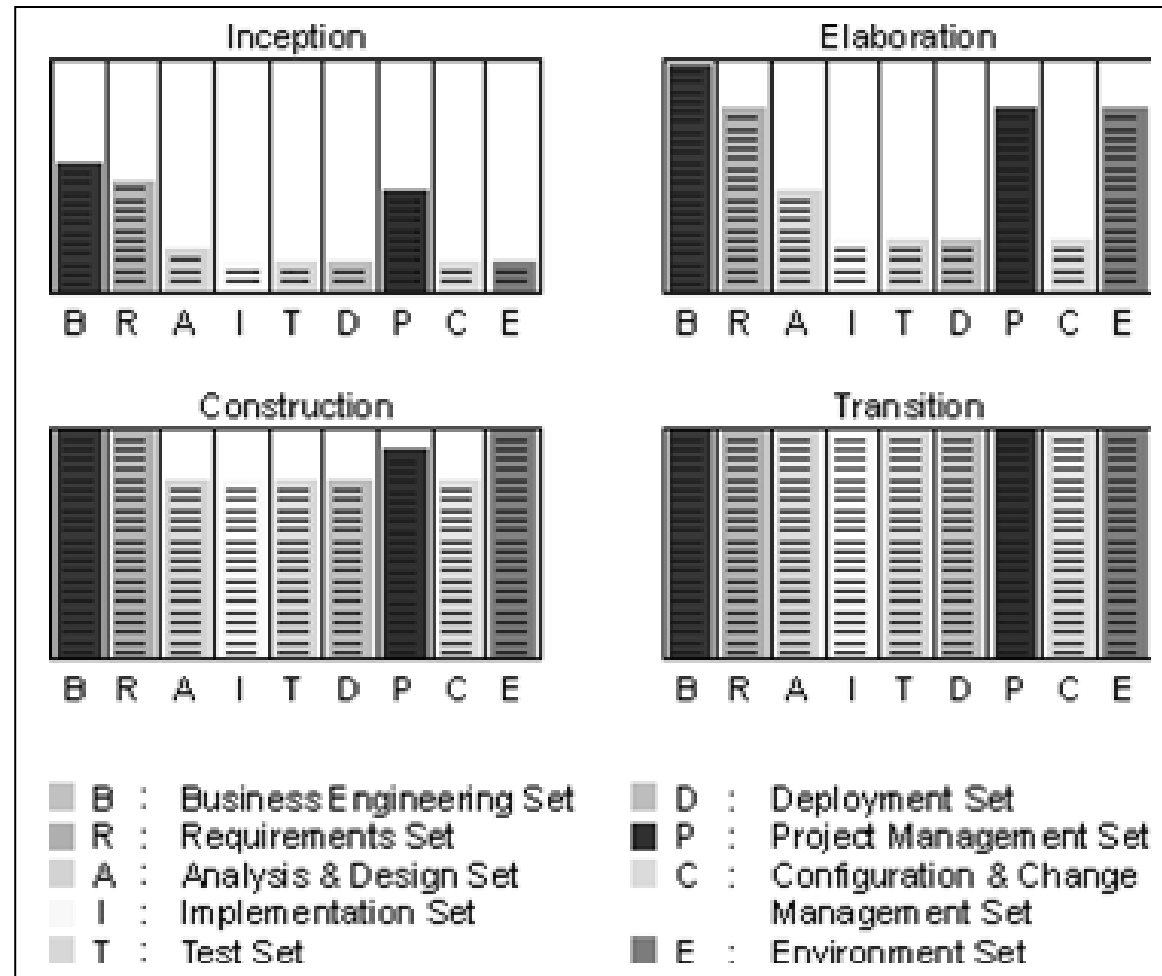




Características del proceso unificado



- Grado de finalización según la fase





Características del proceso unificado



- **Características básicas**
 - **Dirigido por los casos de uso**
 - Los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones.
 - La idea es que cada iteración coja un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.
 - **Centrado en la arquitectura**
 - Se asume que no existe un modelo único que cubra todos los aspectos del sistema.
 - Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura software de un sistema (analogía con la construcción)
 - **Enfocado en los riesgos**
 - Requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

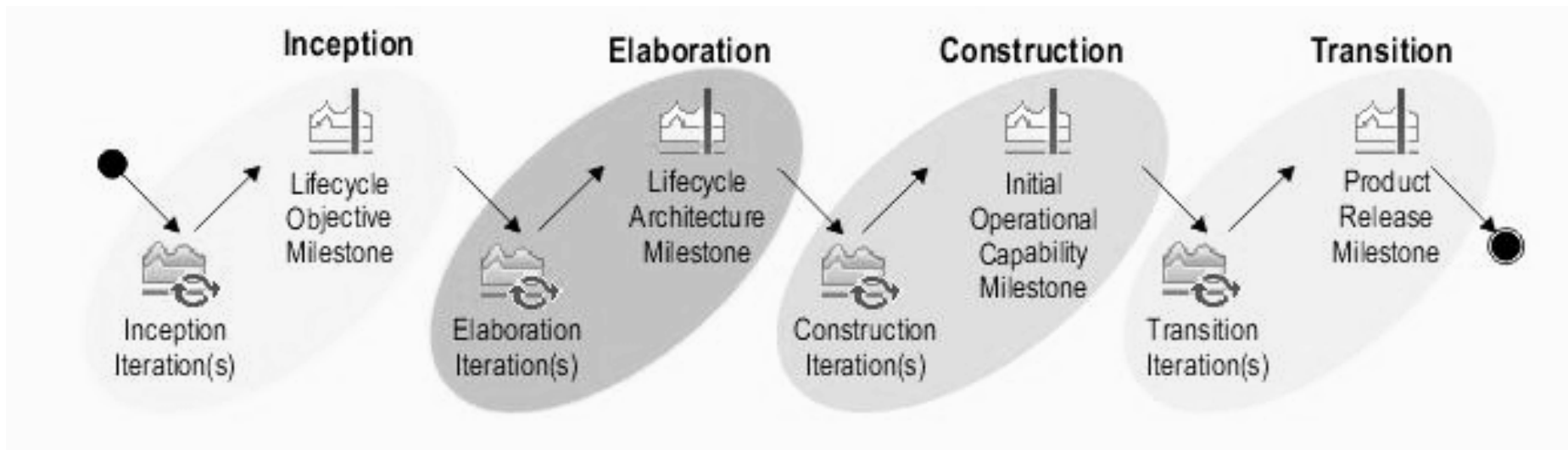


Índice



2. Fases del proceso unificado

- Fase de Inicio
- Fase de Elaboración
- Fase de Construcción
- Fase de Transición





Fases del Proceso Unificado



- **Fase de Inicio**
 - Es la fase más pequeña del proyecto e, idealmente, debe realizarse también en un periodo de tiempo pequeño (una única iteración).
 - El hecho de llevar a cabo una fase de inicio muy larga indica que se está realizando una especificación previa excesiva, lo que responde más a un modelo en cascada
 - **Objetivos**
 - Establecer una justificación (caso de negocio o business case) para el proyecto
 - Establecer el ámbito del proyecto
 - Esbozar los casos de uso y los requisitos clave que dirigirán las decisiones de diseño
 - Esbozar las arquitecturas candidatas
 - Identificar riesgos
 - Preparar el plan del proyecto y la estimación de costes
 - El hito de final de fase se conoce como Hito Objetivo del Ciclo de Vida (Lifecycle Objective Milestone)



Fases del Proceso Unificado



- **Fase de elaboración**
 - Durante esta fase se capturan la mayoría de los requisitos del sistema
 - Los objetivos principales de esta fase serán la identificación de riesgos y establecer y validar la arquitectura del sistema
 - **Base de Arquitectura Ejecutable**
 - La arquitectura se valida a través de la implementación de una Base de Arquitectura Ejecutable (Executable Architecture Baseline)
 - Se trata de una implementación parcial del sistema que incluye los componentes principales del mismo.
 - Al final de la fase de elaboración la base de arquitectura ejecutable debe demostrar que soporta los aspectos clave de la funcionalidad del sistema y que muestra la conducta adecuada en términos de rendimiento, escalabilidad y coste
 - Al final de la fase se elabora un plan para la fase de construcción
 - El Hito Arquitectura del Ciclo de Vida (Lifecycle Architecture Milestone) marca el final de la fase



Fases del Proceso Unificado



- **Fase de construcción**
 - Es la fase más larga del proyecto
 - El sistema es construido en base a lo especificado en la fase de elaboración
 - Las características del sistema se implementan en una serie de iteraciones cortas y limitadas en el tiempo
 - El resultado de cada iteración es una versión ejecutable del software (executable release)
 - El Hito de Capacidad Operativa Inicial (Initial Operational Capability Milestone) marca el final de la fase



Fases del Proceso Unificado



- **Fase de transición**

- En esta fase el sistema es desplegado para los usuarios finales
- La retroalimentación recibida permite incorporar refinamientos al sistema en las sucesivas iteraciones
- Esta iteración también cubre el entrenamiento de los usuarios para la utilización del sistema
- El Hito de Lanzamiento del Producto (Product Release Milestone) marca el final de la fase



Índice

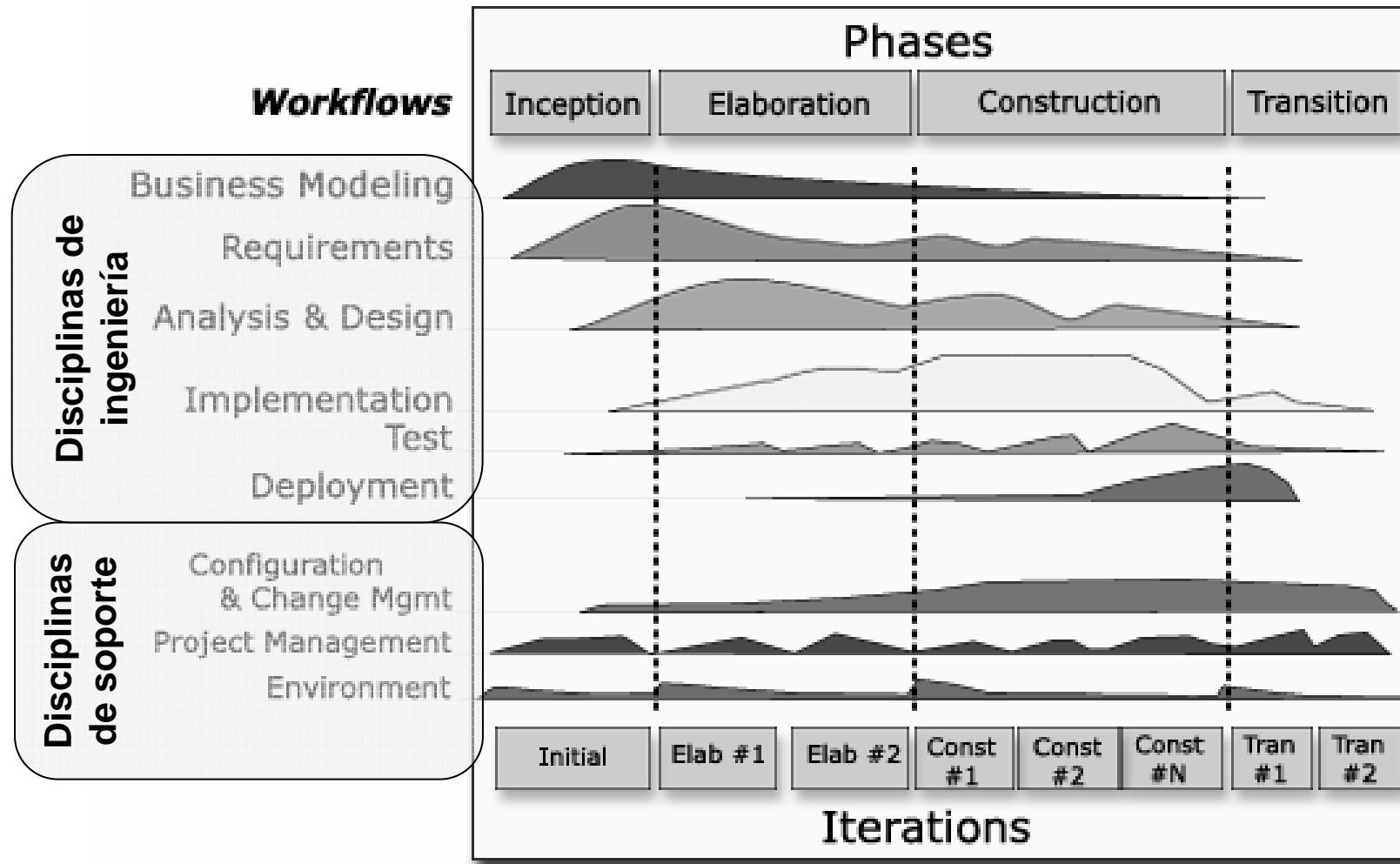


3. Disciplinas del proceso unificado

- Modelado de negocio**
- Requisitos**
- Análisis y Diseño**
- Implementación**
- Prueba**
- Despliegue**
- Gestión de configuraciones y cambios**
- Gestión del proyecto**
- Entorno**



Disciplinas del Proceso Unificado





Disciplinas del Proceso Unificado



- **Modelado del negocio**
 - El objetivo es establecer un canal de comunicación entre los ingenieros del negocio y los ingenieros del software
 - Los ingenieros del software deben conocer la estructura y dinámica de la organización objetivo (el cliente), los problemas actuales y sus posibles mejoras
 - Se plasma en la identificación del modelo del dominio en el que se visualizan los aspectos básicos del dominio de aplicación



Disciplinas del Proceso Unificado



- **Requisitos**
 - El objetivo es describir qué es lo que tiene que hacer el sistema y poner a los desarrolladores y al cliente de acuerdo en esta descripción
- **Análisis y diseño**
 - Describe como el software será realizado en la fase de implementación
 - Se plasma en un modelo de diseño que consiste en una serie de clases (agrupadas en paquetes y subsistemas) con interfaces bien definidos.
 - También contiene descripciones de cómo los objetos colaboran para realizar las acciones incluidas en los casos de uso



Disciplinas del Proceso Unificado



- **Implementación**
 - Se implementan las clases y objetos en términos de componentes (ficheros fuente, binarios, ejecutables, etc.)
- **Prueba**
 - Se comprueba que el funcionamiento es correcto analizando diversos aspectos: los objetos como unidades, la integración entre objetos, la implementación de todos los requisitos, etc.
- **Despliegue**
 - Se crea la versión externa del producto, se empaqueta, se distribuye y se instala en el lugar de trabajo. También se da asistencia y ayuda a los usuarios.



Disciplinas del Proceso Unificado



- **Gestión de configuraciones y cambios**
 - Gestiona aspectos como los sistemas de control de versiones
 - Controla las peticiones de cambios clasificándolas según su estado (nueva, registrada, aprobada, asignada, completa, etc.)
 - Los datos se almacenan en una BD y se pueden obtener informes periódicos
 - Herramientas como Rational ClearQuest o Bugzilla
- **Gestión del proyecto**
 - Encargada de definir los planes del proyecto global, los planes de fase y los planes de iteración
- **Entorno**
 - Se centra en las actividades necesarias para configurar el proceso de un proyecto
 - El objetivo es proveer a la organización de desarrollo software de un entorno de trabajo (que incluye procedimientos y herramientas) que soporten al equipo de desarrollo.



Índice



4. Artefactos

- Definición
- Organización



Artefactos del Proceso Unificado



- **Definición**
 - **Un artefacto es, definido de forma genérica, como el producto de la realización de un trabajo o una tarea**
 - **Puede ser un gráfico, un esquema de la base de datos, un documento de texto, un modelo, etc.**
 - **El proceso unificado define, dentro de cada una de las disciplinas, los artefactos que hay que llevar a cabo dentro de cada una de ellas**



Artefactos del Proceso Unificado



- **Organización**
 - I (inicio), R (refinamiento)
 - Nos centramos en aquellos artefactos que veremos en la asignatura

Disciplina	Artefacto	Inicio	Elab.	Cons.	Tran.
Modelado del negocio	Modelo del dominio		I		
Requisitos	Visión	I	R		
	Especificación suplementaria	I	R		
	Glosario	I	R		
	Modelo de casos de uso	I	R		
Diseño	Modelo de diseño		I	R	
	Modelo de datos		I	R	
	Documento de arquitectura SW		I		
Implementación	Modelo de implementación		I	R	R



Índice



5. Herramientas

- Herramientas del Proceso Unificado
- Herramientas colaborativas



Herramientas



Herramientas del Proceso Unificado

- **Herramientas de pago**
 - Las herramientas nos ayudan a seguir el desarrollo del proceso unificado en un proyecto
 - La división de software Rational de IBM tiene un conjunto completo de herramientas que ayudan a seguir el RUP (<http://www-306.ibm.com/software/rational/>)
 - **Ventajas**
 - Ayudan a implementar todas las disciplinas y artefactos del proceso unificado generando documentación de forma eficiente
 - **Desventajas**
 - Son herramientas comerciales caras. Una licencia del Rational Software Architect cuesta 6.000 €
 - Requieren un nivel de aprendizaje de la herramienta, además del aprendizaje del Proceso Unificado



Herramientas



Herramientas del Proceso Unificado

- **Alternativas gratuitas**
 - **Rational Method Composer**
 - URL: <http://www-306.ibm.com/software/awdtools/rmc/>
 - Plataforma para la gestión de procesos que se usa para gestionar, por ejemplo, el RUP.
 - La herramienta no es gratuita pero la versión de prueba permite generar el sitio web con toda la información disponible del RUP en sus versiones *Classic, Medium y Small*.
 - Incluye guías, plantillas, ejemplos, etc. que ayudan a realizar un seguimiento del RUP



Herramientas



Herramientas del Proceso Unificado

- Ejemplo del RUP generado con el RMC

Rational Unified Process - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

eduardo.mosqueira

Rational Unified Process

Glossary Index Feedback About

Search Print

Where Am I Tree Sets

Team

Getting Started Role Sets

Delivery Processes

Welcome

- Getting Started
- RUP Lifecycle
- Delivery Processes
- Roles
- Disciplines
- Work Products
- Guidance
- What's New
- About

Main Description

Learning

- Getting Started
- How to Navigate
- Key Principles

Navigation Links

- Roles
- Work Products (by domain)
- Processes

Resources

- Overview
- RUP on developerWorks
- Training
- IBM Rational Method Composer
- The Rational Edge

Disciplines

- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment
- Configuration & Change Mgmt
- Project Management
- Environment

Phases

	Inception	Elaboration	Construction	Transition
Business Modeling	High	Medium	Low	None
Requirements	High	Medium	Low	None
Analysis & Design	Low	High	Medium	Low
Implementation	None	Low	High	Medium
Test	None	Low	High	Medium
Deployment	None	None	Low	High
Configuration & Change Mgmt	Low	High	Medium	Low
Project Management	Low	High	Medium	Low
Environment	Low	High	Medium	Low

Iterations

Initial E1 E2 C1 C2 CN T1 T2

Click on an area of the screen for more information.

The preceding figure illustrates the overall architecture of the RUP, which has two dimensions:



Herramientas



Herramientas del Proceso Unificado

- **Eclipse Process Framework**
 - Es el equivalente libre del Rational Method Composer, siendo muy similares incluso en el interfaz
 - Incluye como refinamiento del Proceso Unificado el OpenUP/Basic una versión ligera del mismo
 - URL:
http://www.eclipse.org/epf/downloads/openup/openup_downloads.php



Herramientas



Herramientas del Proceso Unificado

- Ejemplo del OpenUP/Basic generado con el EPF

OpenUP/Basic - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

eduardo.mosqueira

OpenUP/Basic

Glossary Feedback About

Print

Where am I | Tree Sets |

OpenUP/Basic

- Introduction to OpenUP/Basic
- Getting Started
- Core Principles
- Sub-processes
- Roles
- Work Products
- Disciplines
- OpenUP/Basic Lifecycle

Getting Started Core Principles Roles Work Products Disciplines Lifecycle

OpenUP/Basic is organized into four major areas of content, Communication and Collaboration, Intent, Solution, and Management, also known as sub-processes.

OpenUP/Basic is an iterative software development process with the following characteristics:



Herramientas



Herramientas del Proceso Unificado

- **ReadySET**
 - El proyecto Tigris.org pretende generar herramientas y documentación de código abierto sobre la ingeniería del software
 - Por ejemplo, la conocida herramienta de desarrollo de diagramas UML, ArgoUML se ha desarrollado dentro de este proyecto
 - ReadySET es un conjunto de plantillas y guías para el desarrollo de los artefactos de un producto software.
 - No siguen al pie de la letra el proceso unificado pero su estructura es similar.
 - URL: <http://readysset.tigris.org/>



Herramientas



Herramientas del Proceso Unificado

- ReadySET

ReadySET Templates - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

eduardo.mosqueira

ReadySET

- By Activity
 - Project Planning
 - Overview
 - Proposal
 - Target and benefits
 - Project plan
 - Resource needs
 - Legal issues
 - QA Plan
 - Requirements and Specification
 - User needs
 - Interview notes
 - Software Requirements Specification
 - Use case suite
 - Feature set
 - Architecture and Design
 - Design
 - Architecture worksheet
 - Source and build
 - User interface worksheet
 - Persistence worksheet
 - Security worksheet
 - Implementation and Testing
 - User guide
 - Test suite
 - Test case format
 - Test cases
 - Deployment and Installation
 - Release checklist
 - Installation / Quick start guide
 - Release notes
 - Demo script
 - Operations and Support
 - FAQ / Troubleshooting
 - Implementation notes
 - Continuous or Final
 - Glossary
 - Status report
 - Review meeting notes
 - Software development methodology
- By Suggested Sequence
- All Templates

How to download these templates:

SRS > Use Case Suite

Release Information

Project:	PROJECTNAME
-----------------	-------------

Internal Release Number:

X.Y.Z

Related Documents:

- [Project proposal > User needs](#)
- [SRS > Feature set](#)
- [Use case format](#)
- [Use case diagram](#)

LINKS TO RELEVANT STANDARDS
LINKS TO OTHER DOCUMENTS

Process impact: A use case suite is simply a table of contents for the individual use cases. Much like a test suite, organizing the suite of use cases by priority, functional area, actor, business object, or release can help identify parts of the system that need additional use cases.

TODO: Before writing individual use cases, list the use cases that you think you will need. Organize them in a way that will purposely leave visible blanks on this page if you are missing use cases. E.g., see "Scalability and availability". Choose one or more of the organizations show below.

TIP: Refer back to the user stories in your [user needs](#) document. Use them for ideas and make sure that you cover all of them. Remember that use cases are more precise than user stories, and there may be several use cases for a given user story.

TIP: The use case suite can be organized into nested lists according to other coverage criteria, e.g., by actor. Or, it can be organized into tables that consider two aspects at a time, e.g., business objects vs. actor. If a certain section of the tree or table does not need use cases, explicitly mark it "N/A". Otherwise, mark it "TODO".

Use Cases by Functional Area

- User account management
 - [UC-00](#) Configure the site
 - [UC-01](#) Register as a new user
 - [UC-02](#) Request new password
 - [UC-03](#) Edit user profile
 - [UC-04](#) View user profile



Herramientas



Herramientas colaborativas

- Cuando el trabajo es llevado a cabo por muchas personas es fundamental contar con herramientas que faciliten el trabajo compartido
- Estas herramientas deben mantener un control de versiones y deben permitir visualizar las aportaciones de cada usuario
- Para el código fuente la más utilizada es CVS (Concurrent Versioning System). URL: <http://ximbiot.com/cvs/wiki/>
- Para la documentación es común usar un sistema basado en wikis, como por ejemplo MediaWiki, software utilizado en la Wikipedia. URL: www.mediawiki.org



Herramientas

Herramientas colaborativas



- Cliente CVS de NetBeans

The screenshot shows the NetBeans IDE 5.5 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Build, Run, CVS, Tools, Window, and Help. The toolbar contains various icons for file operations and development. The main window displays a CVS log for 'Tutorizacion1.java' with the following data:

File Revision	Time	User	Message
1.7	2007/01/17 11:44:32	jose	*** empty log message ***
1.6	2007/01/17 10:04:05	eduardo	Nueva pestaña de capacidad motriz
1.5	2007/01/16 17:39:39	jose	Prototipo con tres modelos de interface para los formularios.
1.4	2007/01/16 10:00:30	jose	*** empty log message ***
1.3	2007/01/15 16:22:23	jose	*** empty log message ***

Below the log, a diff view compares revisions 1.5 and 1.6. The left pane shows revision 1.5 with line numbers 2213 to 2225. The right pane shows revision 1.6 with line numbers 2251 to 2262. The diff highlights changes in the `.addGroup` method calls, specifically the addition of `.createParallelGroup` and `.addContainerGap` methods.

```
1.5
2213 .addGroup (jPanel13Layout.createSequentialGroup ())
*
2214 .addContainerGap ()
2215 .addGroup (jPanel13Layout.createParallelGroup (javax.swing.G
2216 .addGroup (jPanel13Layout.createSequentialGroup ())
2217 .addGap (428, 428, 428)
2218 .addComponent (jSeparator37, javax.swing.Group
2219 .addGap (88, 88, 88))
2220 .addGroup (jPanel13Layout.createSequentialGroup ())
2221 .addGroup (jPanel13Layout.createParallelGroup
2222 .addComponent (jPanel14, javax.swing.Group
2223 .addComponent (jPanel15, javax.swing.Group
2224 .addContainerGap ()))
2225 .addGroup (jPanel13Layout.createSequentialGroup ())

1.6
2251 .addGroup (jPanel13Layout.createSequentialGroup ())
2252* .addGroup (jPanel13Layout.createParallelGroup (javax.swing.G
2253 .addGroup (jPanel13Layout.createSequentialGroup ())
2254 .addContainerGap ()
2255 .addGroup (jPanel13Layout.createParallelGroup (javax.swi
2256 .addGroup (jPanel13Layout.createSequentialGroup ())
2257 .addGap (428, 428, 428)
2258 .addComponent (jSeparator37, javax.swing.GroupLayout
2259 .addGroup (jPanel13Layout.createParallelGroup (javax.s
2260 .addComponent (jPanel14, javax.swing.GroupLayout.Al
2261 .addComponent (jPanel15, javax.swing.GroupLayout.Al
2262 .addGroup (jPanel13Layout.createSequentialGroup ())
```

Save All finished.



Herramientas

Herramientas colaborativas



- **MediaWiki**

Modelo de casos de uso - Prosaicopedia - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.jcio.us

Eduardo mi discusión my preferencias lista de seguimiento mis contribuciones salir

artículo discusión editar historial trasladar vigilar

Modelo de casos de uso

Los casos de uso del sistema pueden verse en el siguiente gráfico

```
graph TD
    Actor1[Monitor] --- CU01((CU-01  
Entrevista inicial))
    Actor2[Fisioterapeuta] --- CU02((CU-02  
Introducción de datos personales))
    Actor3[Director técnico] --- CU07((CU-07  
Revisar datos presonales))
    Actor3 --- CU08((CU-08  
Revisar estado de salud))
    Actor3 --- CU09((CU-09  
Revisar cumplimiento))
    Actor3 --- CU10((CU-10  
Revisar cumplimiento))
    CU01 -.->|<<include>>| CU02
    CU01 -.->|<<include>>| CU03((CU-03  
Introducción del estado fisico))
    CU01 -.->|<<include>>| CU04((CU-04  
Introducción del volumen de trabajo))
    CU06((CU-06  
Realizar una revision)) -.->|<<include>>| CU07
    CU06 -.->|<<include>>| CU08
    CU06 -.->|<<include>>| CU09
    CU06 -.->|<<include>>| CU10
```

http://requemv1.dc.fi.udc.es/prosaicopedia/index.php/Imagen:Modelo_de_Casos_de_Uso.jpg



Herramientas

Herramientas colaborativas



- **MediaWiki**

The screenshot shows a Mozilla Firefox browser window displaying the 'Historial' (History) page for 'Modelo de casos de uso' on the 'Prosaicopedia' website. The page title is 'Modelo de casos de uso' and it shows a list of 20 revisions. The current revision is the first one, dated 10:16 9 ene 2007 by Jose. The page includes a navigation menu, a search box, and a list of tools. The status bar at the bottom indicates 'Terminado'.

Modelo de casos de uso - Historial - Prosaicopedia - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.jcio.us

Eduardo mi discusión my preferencias lista de seguimiento mis contribuciones salir

artículo discusión editar historial trasladar vigilar

Modelo de casos de uso

Historial de revisiones
Ver los registros de esta página

(Últimas | Primeras) Ver (50 previos) (50 siguientes) (20 | 50 | 100 | 250 | 500).

Leyenda: (act) = diferencia con la versión actual, (prev) = diferencia con la versión previa, M = edición menor

Comparar versiones seleccionadas

- (act) (prev) 10:16 9 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 21:51 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 21:47 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:51 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:50 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:49 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:49 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:48 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:31 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:30 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 18:21 8 ene 2007 Jose (Discusión | contribuciones)
- (act) (prev) 01:59 27 dic 2006 Eduardo (Discusión | contribuciones)
- (act) (prev) 23:21 26 dic 2006 Eduardo (Discusión | contribuciones)
- (act) (prev) 23:06 25 dic 2006 Eduardo (Discusión | contribuciones) *(Nuevo formato)*
- (act) (prev) 22:49 25 dic 2006 Eduardo (Discusión | contribuciones)
- (act) (prev) 10:13 23 dic 2006 Eduardo (Discusión | contribuciones) *(Precondiciones y postcondiciones)*

Terminado



Índice



6. Casos de Estudio

- NextGen POS
- Monopoly
- Punto de Inversión (PDI)



Casos de Estudio



- **NextGen POS System (PDV NuevaEra)**
 - **Un sistema POS (Point Of Sale) es una aplicación computerizada utilizada para registrar ventas y tratar pagos**
 - **Se usa típicamente en super e hipermercados**
 - **Deben ser sistemas relativamente tolerantes a fallos (caídas en los servicios remotos)**
 - **Debe poder soportar diferentes terminales de cliente (de escritorio, basados en web, PDAs, etc.)**
 - **Diferentes clientes puede requerir diferentes funcionalidades del sistema POS, por lo que el sistema debe ser flexible y configurable**



Casos de Estudio



- **Monopoly**
 - **Juego de tablero de estrategia empresarial inmobiliaria**
 - **Los jugadores compran, venden, construyen, hipotecan, etc. recursos inmobiliarios**
 - **Gana el jugador que consiga arruinar al resto**
 - **Para simplificar el diseño se creará una versión de simulación, es decir, una persona indica el número de jugadores y observa como el juego discurre hasta su conclusión**



Casos de Estudio



- **Punto de Inversión (PDI)**
 - **Un PDI no sería más que un dispositivo, similar a los cajeros actuales, pero que permita la realización de operaciones financieras avanzadas.**
 - **Para el acceso al PDI nos suministrarán una tarjeta.**
 - **Entre las distintas opciones disponibles podemos encontrar:**
 - **Sacar dinero de alguna de las cuentas asociadas a la tarjeta**
 - **Crear nuevas cuentas asociadas a la tarjeta**
 - **Dar de baja cuentas**
 - **Ingresar dinero en alguna de las cuentas asociadas a la tarjeta**
 - **Invertir en productos financieros: fondos de inversión, acciones de otras compañías**
 - **Variar condiciones de acceso (claves, PIN, etc.)**
 - **Etc.**



Referencias



- **Kroll, Per; Kruchten, Philippe (2003). The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP. ISBN 0-321-16609-4.**
- **Kruchten, Philippe (2004). The Rational Unified Process: An Introduction (3rd Ed.). ISBN 0-321-19770-4.**
- **Larman, Craig (2004). Agile and Iterative Development: A Manager's Guide. ISBN 0-13-111155-8.**
- **Larman, Craig (2005). Applying UML and Patterns. ISBN 0-13-148906-2**
- **Scott, Kendall (2002). The Unified Process Explained. ISBN 0-201-74204-7.**
- **Bergstrom, Stefan; Raberg, Lotta (2004). Adopting the Rational Unified Process: Success with the RUP. ISBN 0-321-20294-5.**