

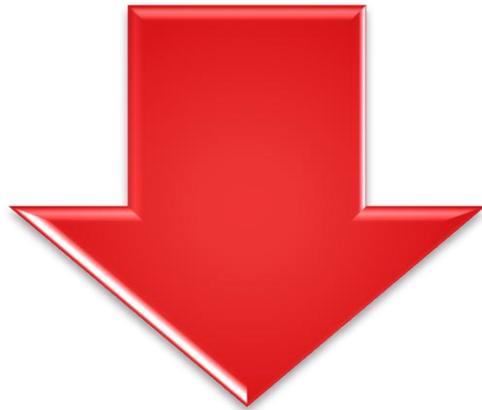
IN3501 - Tecnologías de Información y Comunicaciones
para la Gestión

CAPA DE NEGOCIOS O DE APLICACION

PROFESORES

Evelyn Andaur
Juan D. Velásquez
Gastón L'Huillier
Víctor Rebolledo Lorca

Costo / Beneficio



COSTOS

- No siempre la solución satisface requerimientos del negocio
- No incorpora mejores prácticas de negocio
- Puede volverse un sistema no compatible con la infraestructura tecnológica



BENEFICIOS

- Solución específica al negocio
- Más barato que un Paquete informático corporativo
- Bien diseñado y mantenido, puede ser la columna vertebral del negocio





Solución 2

 **USAR BPM SYSTEM**

Temario

- Gestión sobre los procesos de negocio
- Business Process Management BPM
- Business Process Management Systems BPMS
- Introducción a BPMN
- BPEL, BPEL4People & BAM
- Herramientas disponibles
- Costo / Beneficio

Gestión sobre procesos de negocio



改 = KAI = CHANGE

善 = ZEN = GOOD
(FOR THE BETTER)

改善 = KAIZEN
= CONTINUAL
IMPROVEMENT



Frederick
Taylor

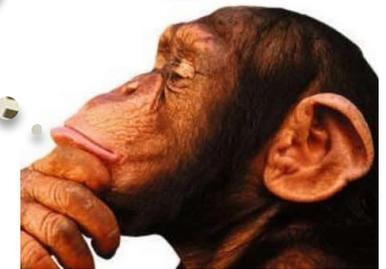


Total Quality
Management
TQM

Reingeniería
de Procesos

Six Sigma

Business
Process
Management
BPM



La Revolución Industrial (1920)

- Frederick Taylor
 - Ingeniero Mecánico y economista estadounidense
- Los procesos son **implícitos** a las *prácticas de trabajo*.
- La **Gestión de Procesos** se limitaba al *análisis de métodos y procedimientos*.
- Restringida a **procesos manuales** y *producción industrial*
- METODO TAYLOR:
 - Calcular tiempo promedio para producir determinado producto y obligar al obrero a acelerar ritmo de trabajo asimilándolo a una máquina.
 - El obrero debía ser un “**buen gorila amaestrado**”



Aislar al trabajador

Sueldo de acuerdo al rendimiento



Trabajador solo hace parte del trabajo

Total Quality Management (TQM)

- “A management approach of an organization, centered on quality, based on the participation of all of its members and aiming at long-term success through customer satisfaction, and benefits to all members of the organization and too society” (ISO)
- **“Control estadístico aplicado a la calidad de la producción”**
- Desarrollado en Japón durante la década de los 40 y liderada por gurús de calidad norteamericanos
 - Deming, Juran y Feigenbaum
- Kaizen → TOYOTA



TQM
It Really Works!



Total Quality Management (TQM)

Requiere de una aplicación rigurosa y exhaustiva

Mejora Continua

Foco en los procesos, más que en el resultado



Aproximación Científica

Uso de Herramientas estadísticas

Trabajo en Equipo

Incentivar cooperación
El conocimiento está en
“Cualquiera”

BPR – Business Process Reengineering

- Recesión de la década de los 90
- *Hammer and Champy* acuñan el concepto de “Business Process Reengineering” (BPR) como medida de salvación para **desburocratizar** las empresas y ser más eficientes en sus procesos de negocio.
- Muchos proyectos de BPR fueron una *racionalización del negocio* acompañada de una fuerte *reducción de personal*.
- **Grandes éxitos, pero muchos fracasos**
- **Reingeniería**
 - “Don’t Automate, Obliterate”
 - *El fin de rediseñar, repensar el negocio, borrar con las estructuras funcionales presentes*
 - *Gestión de procesos transversal por sobre la gestión por áreas funcionales.*
 - *Centrado en la satisfacción del cliente*
 - *El cambio debe ser apoyado por TIC*

Six Sigma

- *Metodología de mejora de procesos centrada en la reducción de la variabilidad de la salida (producto o servicio)*
- Creado por **Motorola** en la década de los 80
- Popularizado por **General Electric** en los 90
- Re encantamiento con el uso de control estadístico.
- Orientado en eficiencia.
- Suele ser usado con otras metodologías:
 - TPS (Toyota Production System)
 - Lean Production

| | DPMO | Eficiencia |
|------------|---------|------------|
| 1 σ | 690.000 | 68,27% |
| 2 σ | 308.000 | 95,45% |
| 3 σ | 66.800 | 99,73% |
| 4 σ | 6.210 | 99,994% |
| 5 σ | 230 | 99,99994% |
| 6 σ | 3,4 | 99,999966% |

DPMO - Defects per Million Opportunities

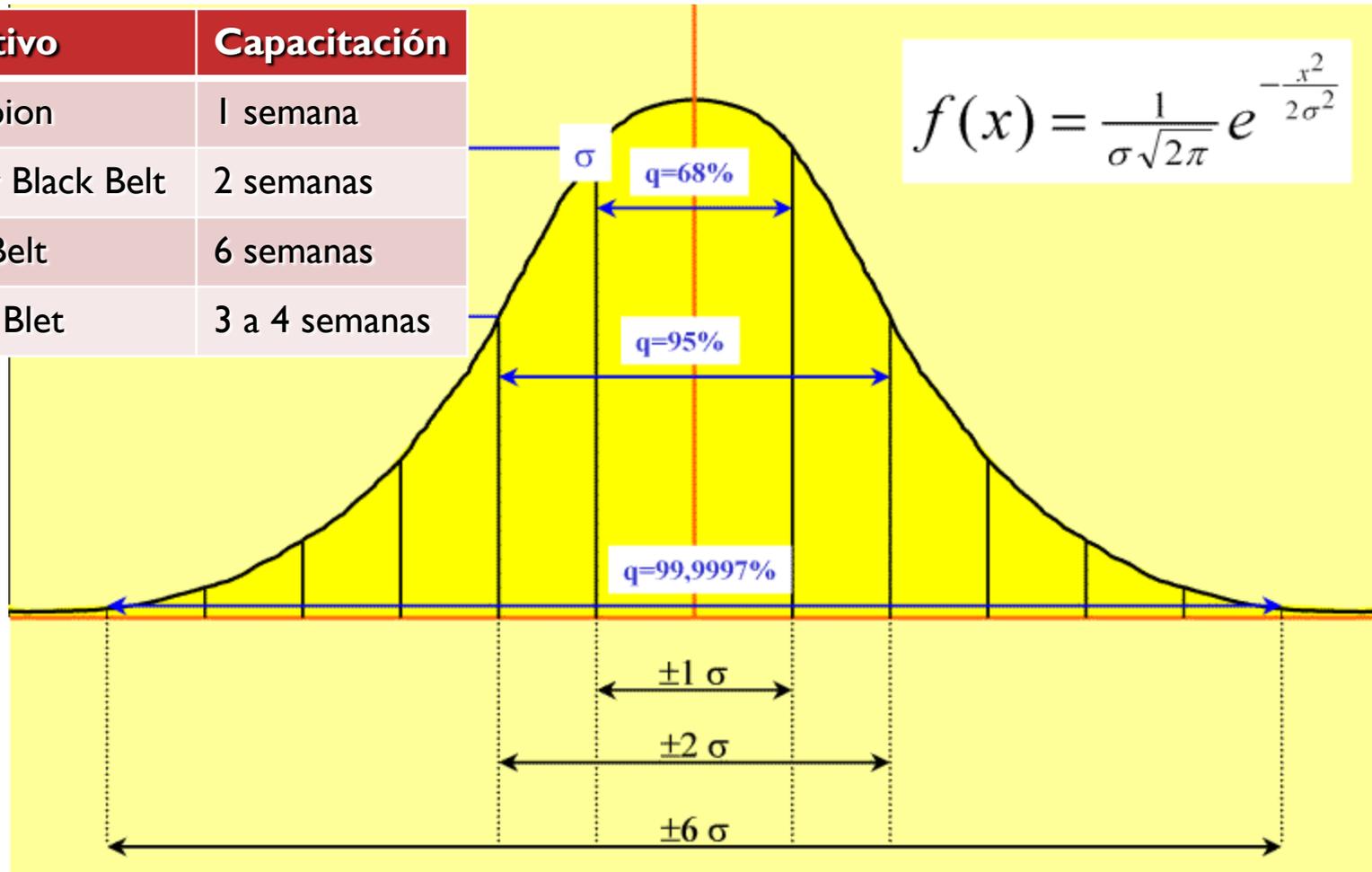
6 σ



“One output defect in **six Standard deviations** of a probability distribution for a particular process output”

Six Sigma

| Ejecutivo | Capacitación |
|-------------------|---------------|
| Champion | 1 semana |
| Master Black Belt | 2 semanas |
| Black Belt | 6 semanas |
| Green Blet | 3 a 4 semanas |



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

ERP – Enterprise Resource Management

- Aparecen a mediados de los 90
- Diseñados para mejorar la eficiencia administrativa
 - Incorporan “**mejores prácticas**” de gestión.
- No mejoraron la eficiencia y eficacia de los procesos de negocio.
- Poco “**flexibles**”
- **ERP**
 - **Planificación de recursos empresariales**
 - **Derivado de MRP**
 - *Sistemas de información integrados (módulos)*
 - *Producción*
 - *Lógica*
 - *Distribución*
 - *Inventario*
 - *Ventas*
 - *Contabilidad*
 - *Recursos Humanos*
 - *Etc.*

ERP – Enterprise Resource Planning

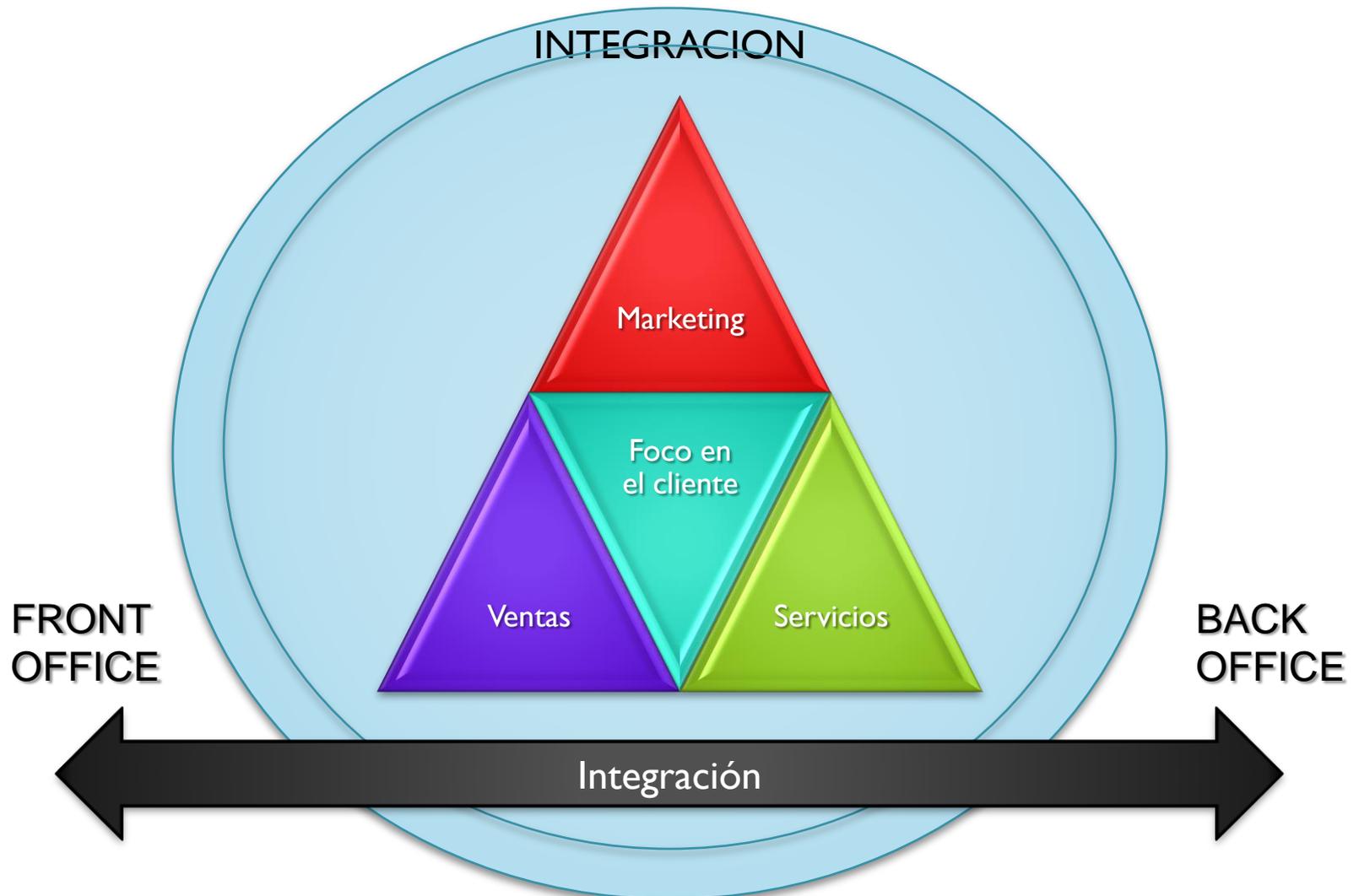


CRM – Customer Relationship Management

- Aparecen a fines de los 90 y principios de los 2000
- Metodología de gestión orientada a **estrechar relaciones** con los clientes
 - Marketing Relacional
 - Marketing I a I
- Sistema informático para la gestión **front-end** del negocio.
 - Ventas
 - Marketing
 - Clientes
- **Estudiar al cliente para agregarle valor y fidelizarlo.**
- Aplicación exitosa en **call centers**.



CRM – Customer Relationship Management

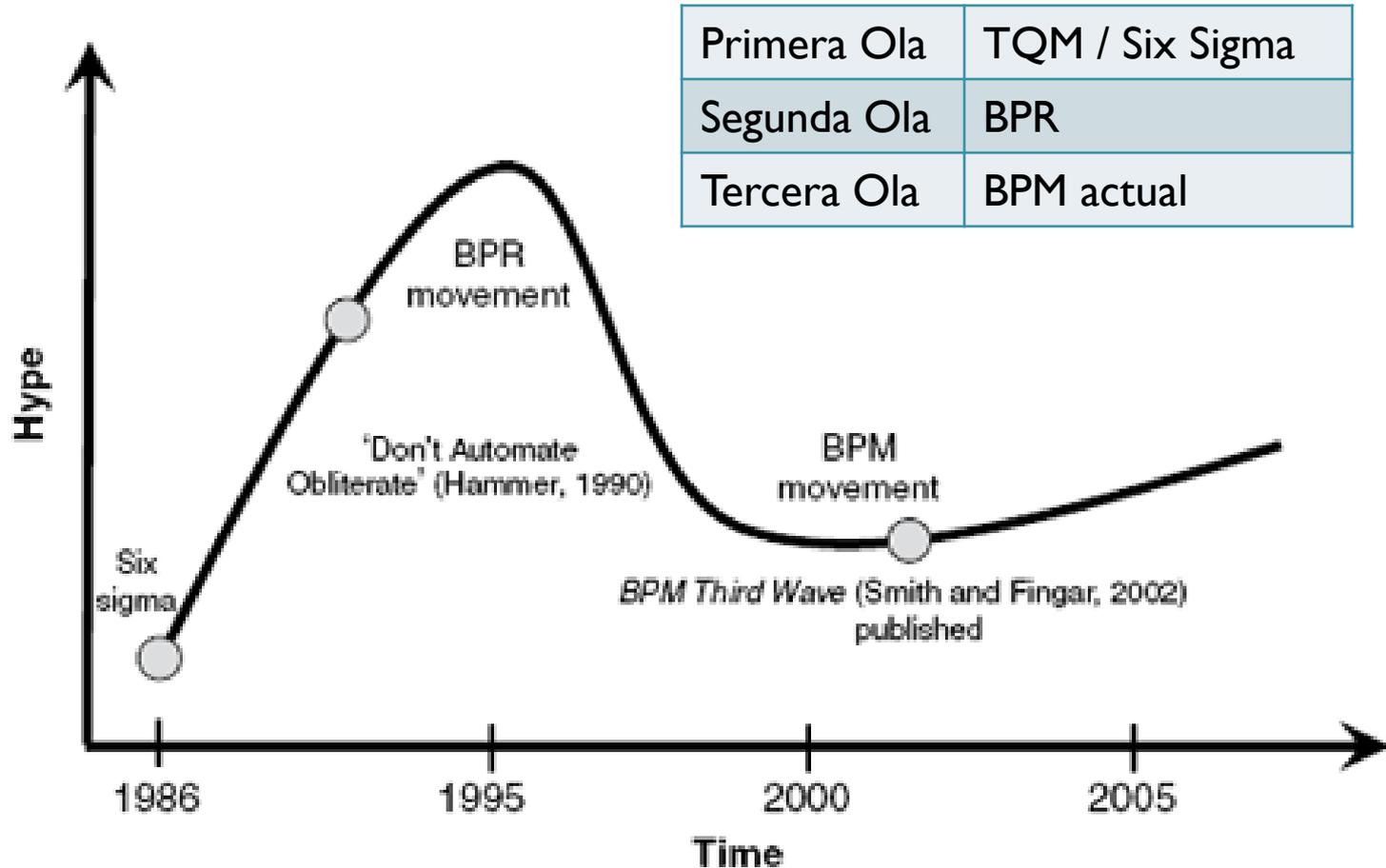


BPM – Business Process Management

- ¡¡¡Muchos conceptos, metodologías, tecnologías!!!
- Finalmente ...
- **¿Qué es BPM?**
- Mucha gente confundida con el término
- *¿Es una nueva moda?*
- *¿Un nuevo nombre para prácticas ya conocidas, pero integradas?*



BPM hype cycle



BPM – Business Process Management

- Según Jeston and Nelis (2007), BPM es:
 - Más que sólo software
 - Más que sólo la mejora y reingeniería de procesos
 - No es sólo una moda, es parte integral del management
 - Más que sólo levantamiento y modelado de procesos, también es la implementación y ejecución de los procesos, los cuales requieren ser analizados y mejorados.

BPM – Business Process Management



Focalizado en el **cliente**

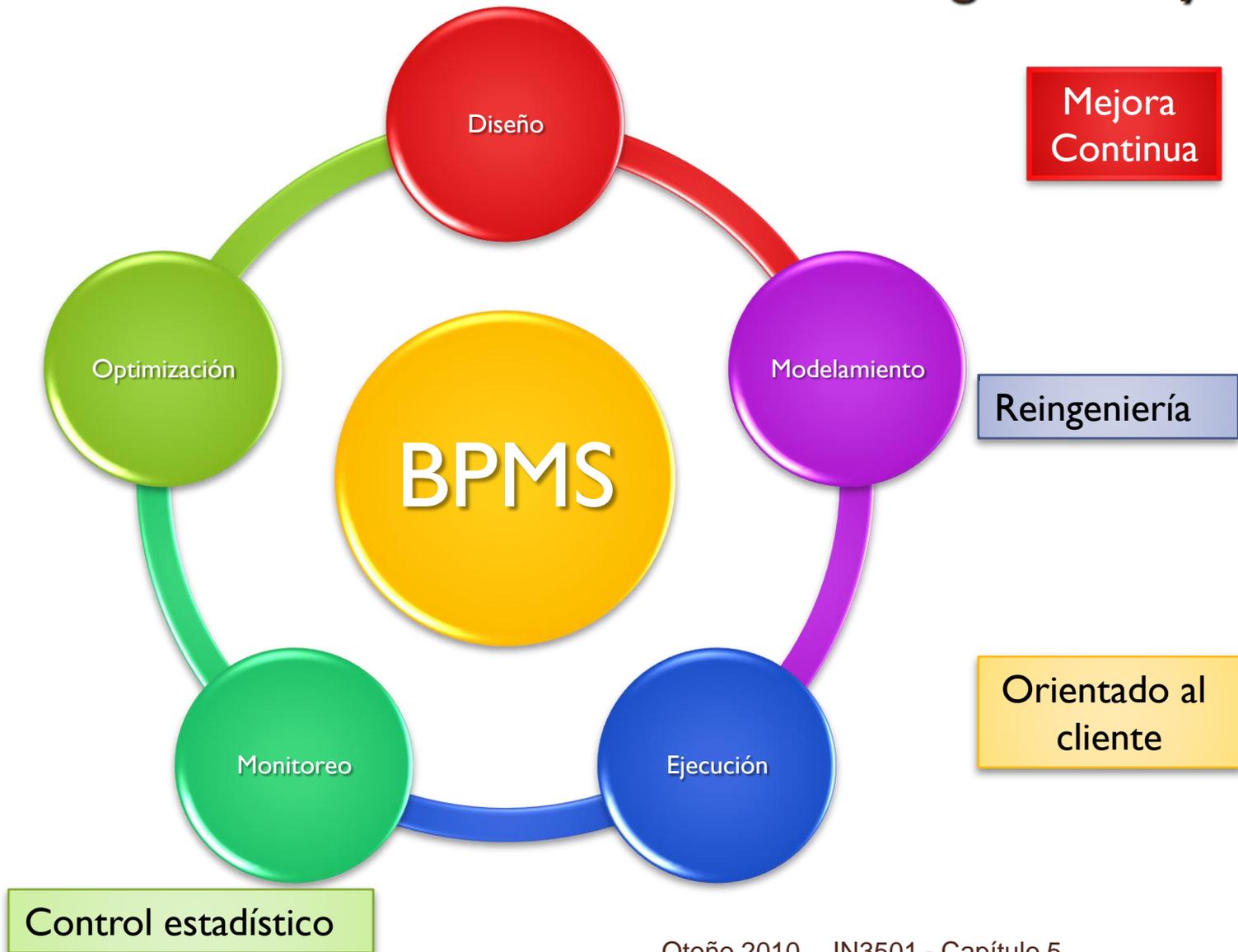
Finalmente ...

- *BPM no es un concepto simple y no es fácil de implementar, de hecho, es complejo y difícil, pero ha demostrado crear valor para los clientes y el negocio.*
- Según Gartner (2006) más del 90% de las empresas implementarán BPM en los próximos años
- Las ventas crecieron
 - De \$US 3,2 billones el año 2003
 - A \$US 6,3 billones el año 2006

BPMS — Business Process Management System

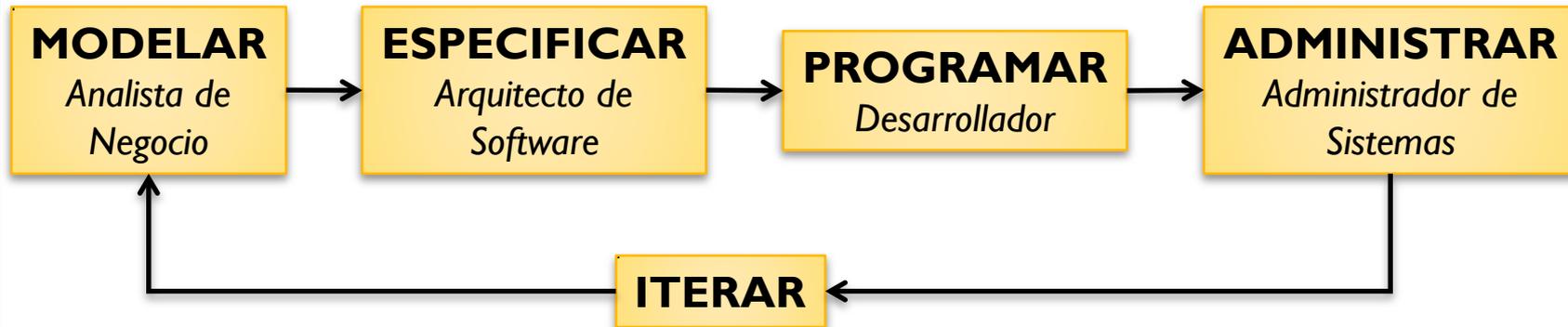
- **Sistema informático** que habilita a las compañías a *modelar, implementar y gestionar* procesos de negocio críticos en una compañía.
- Estos procesos pueden comprender *múltiples sistemas de gestión, departamentos corporativos, empleados de diversas especialidades, clientes y proveedores.*
- Puede ser considerado una **nueva categoría de software para la gestión de procesos**, *reduciendo las holguras* entre el modelamiento, la implementación y el monitoreo de procesos

BPMS – Business Process Management System



BPMS vs Solución a Medida

Solución a Medida

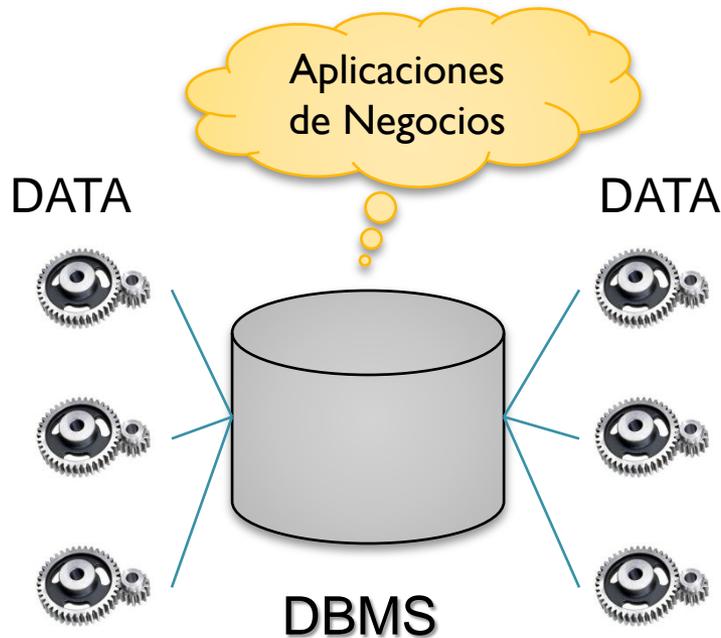


BPMS

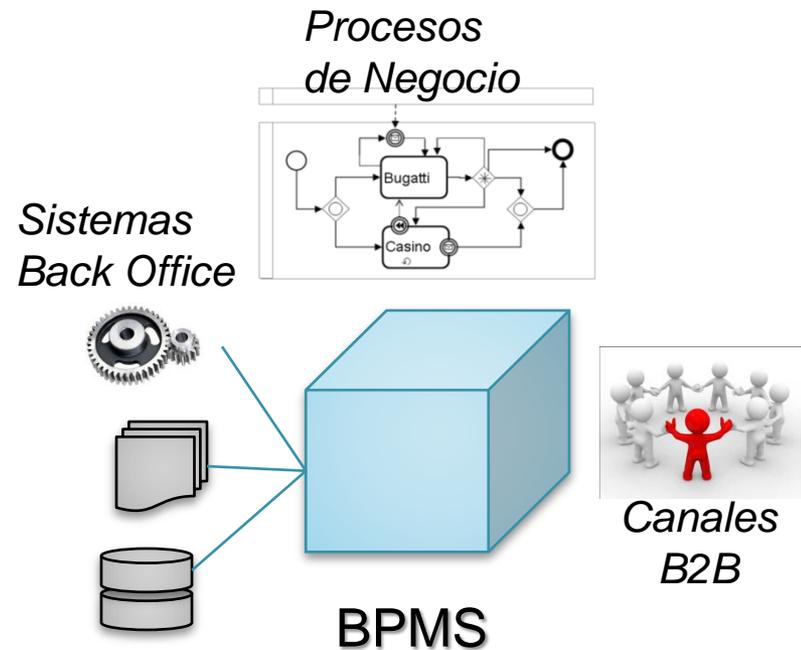


BPMS – Business Process Management System

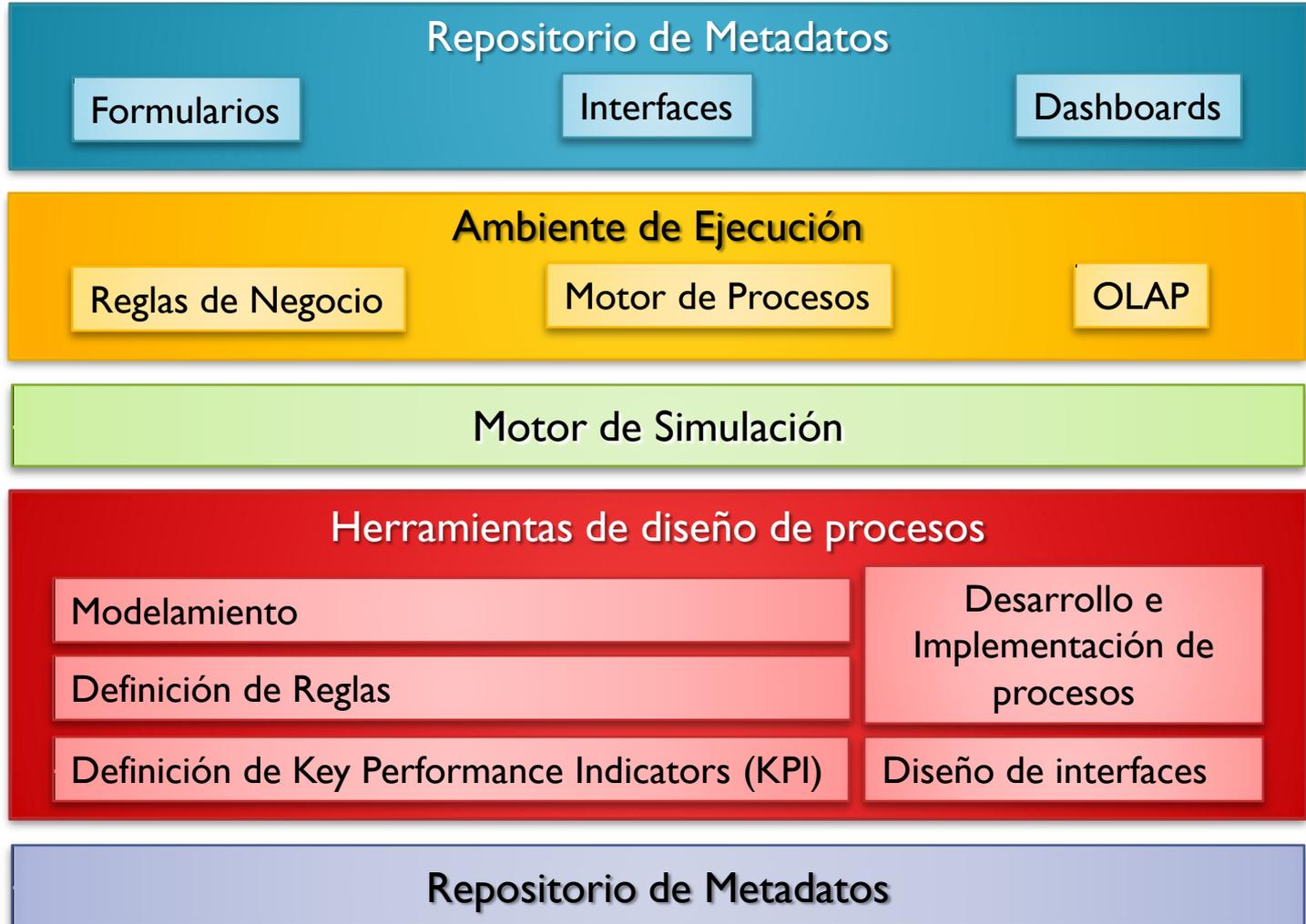
- Sin BPMS



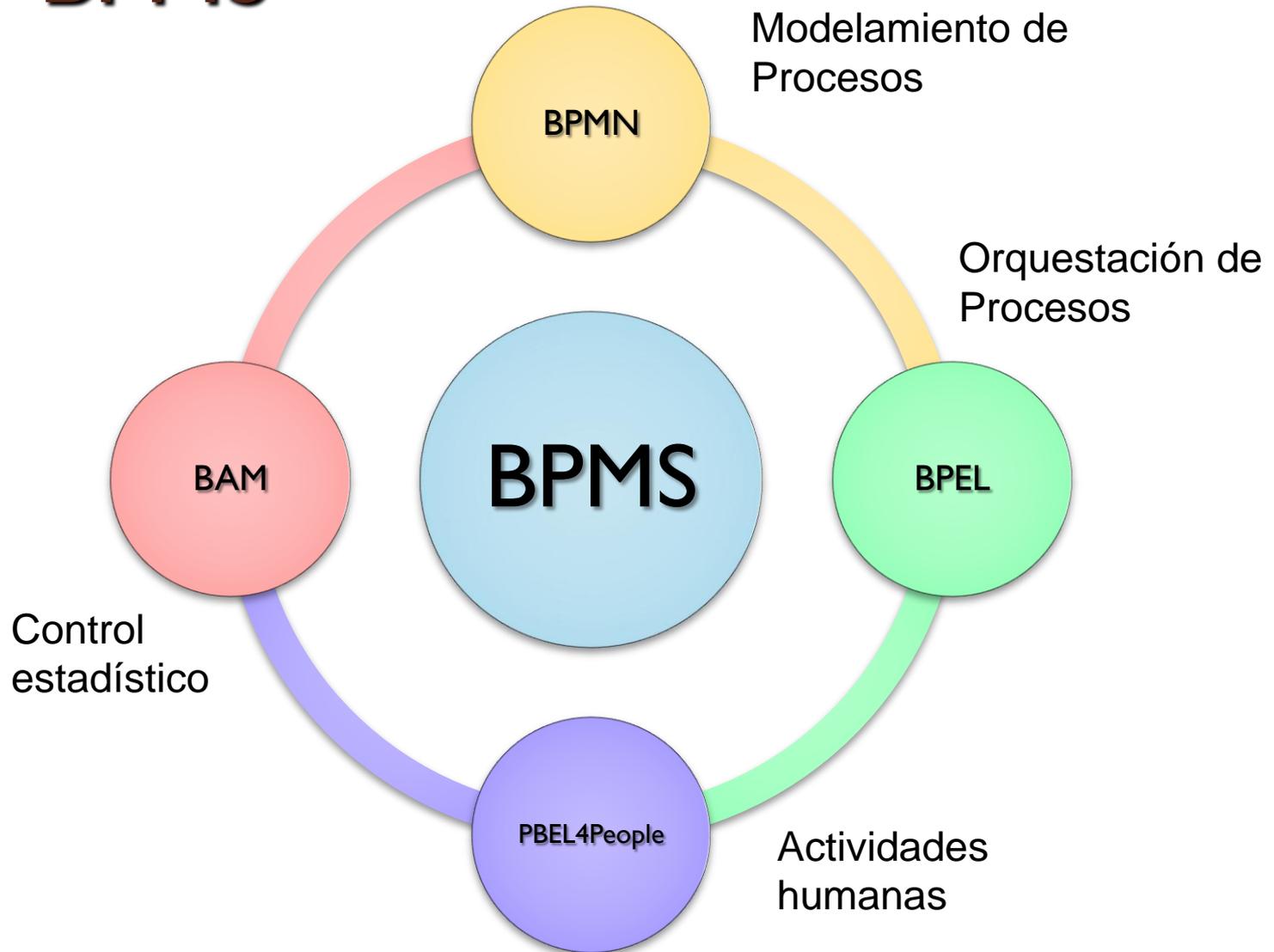
- Con BPMS



BPMS – Business Process Management System



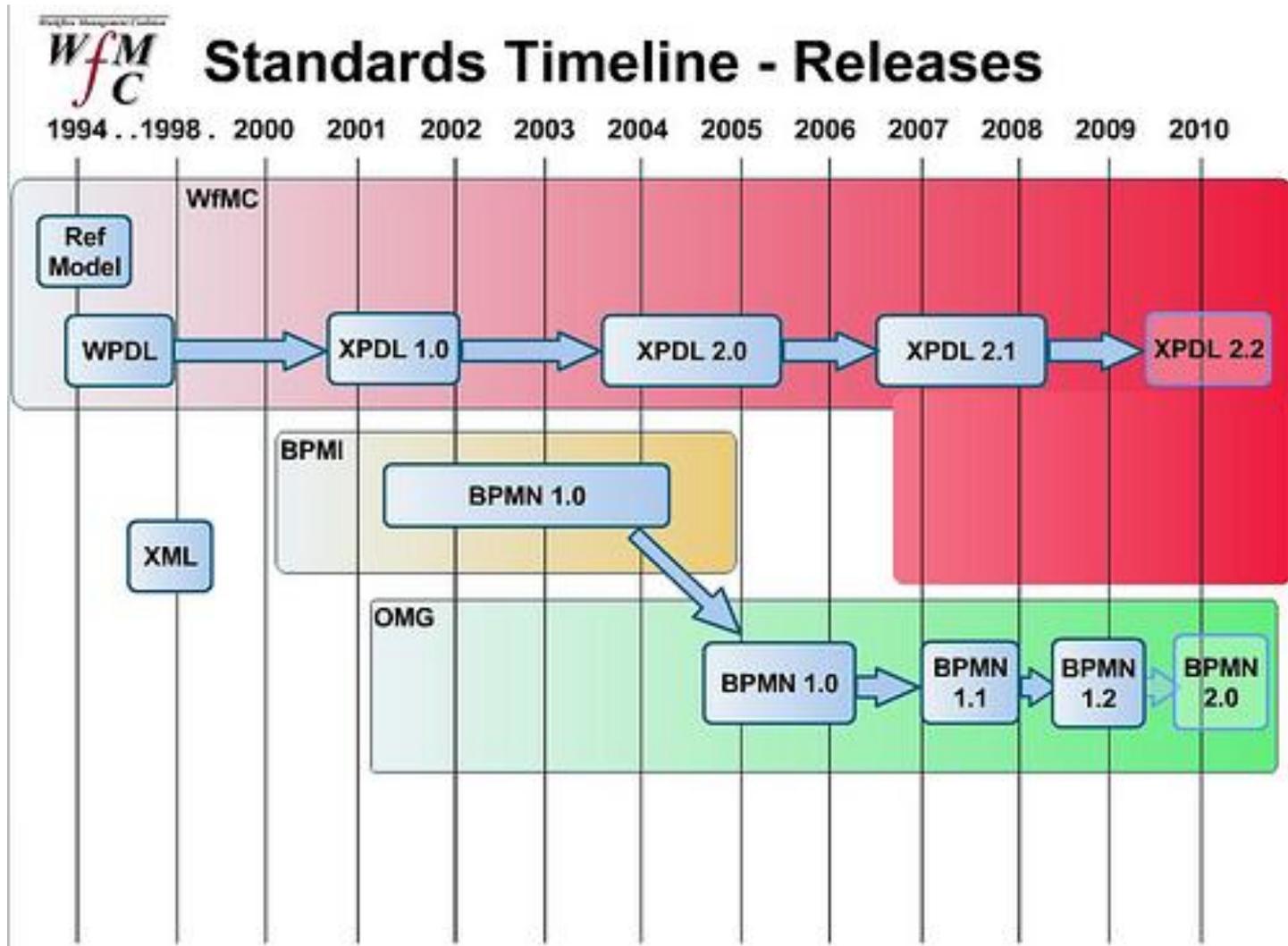
BPMS



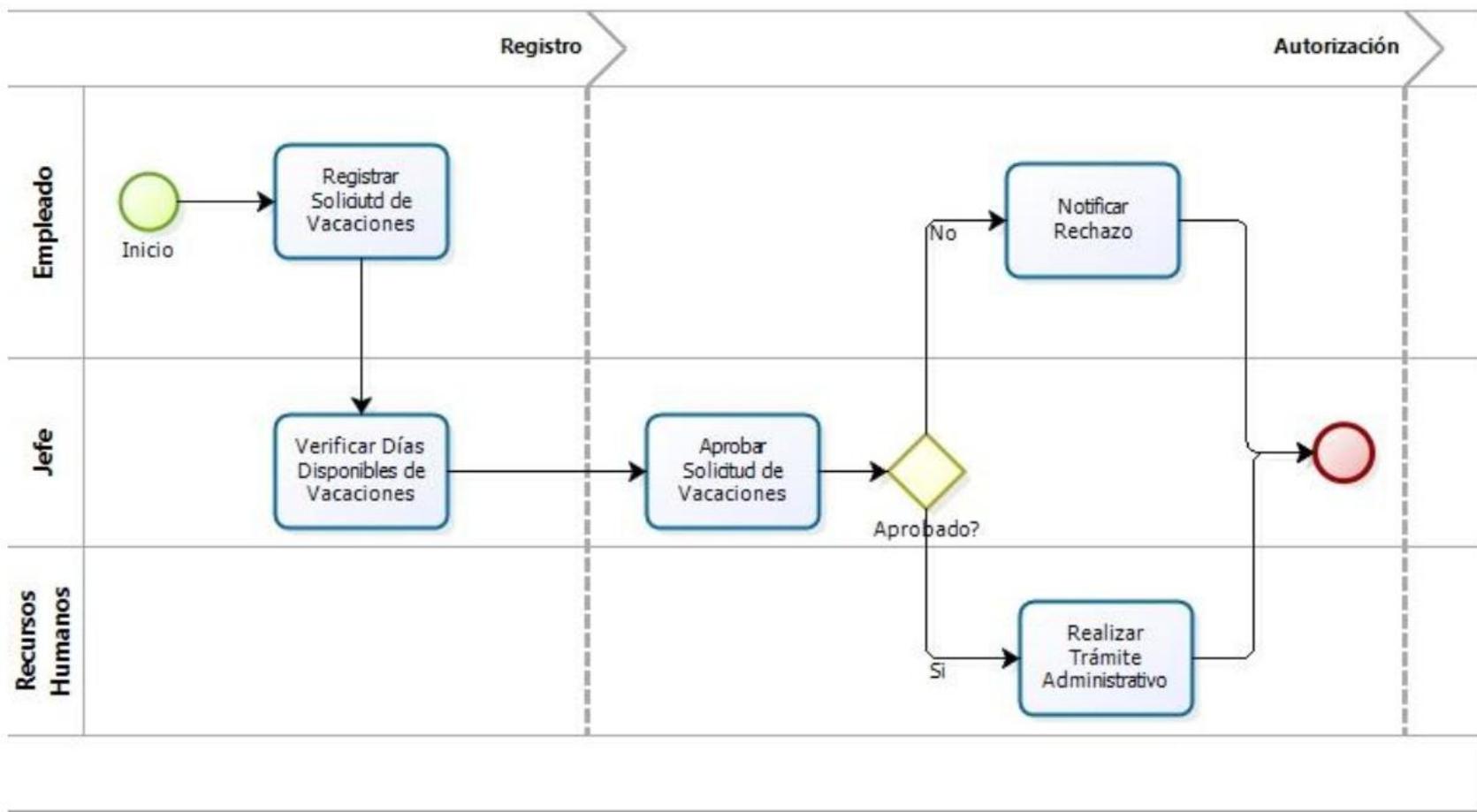
Introducción a BPMN

- **Business Process Management Notation**
- *Notación gráfica estandarizada para el modelado de procesos de negocio.*
- Desarrollada por la *Business Process Management Initiative (BPMI)* www.bpmi.org
- Mantenida por la *Object Management Group (OMG)* www.omg.org
- Existen otras alternativas:
 - Por ejemplo: XPD L impulsada por la *Workflow Management Coalition (WfMC)* www.wfmc.org
- Sin embargo, BPMN es usada por los principales BPMS
- Versión actual 1.1, se prevee una versión 2.0

BPMN v/s otros estándares de modelamiento



BPMN

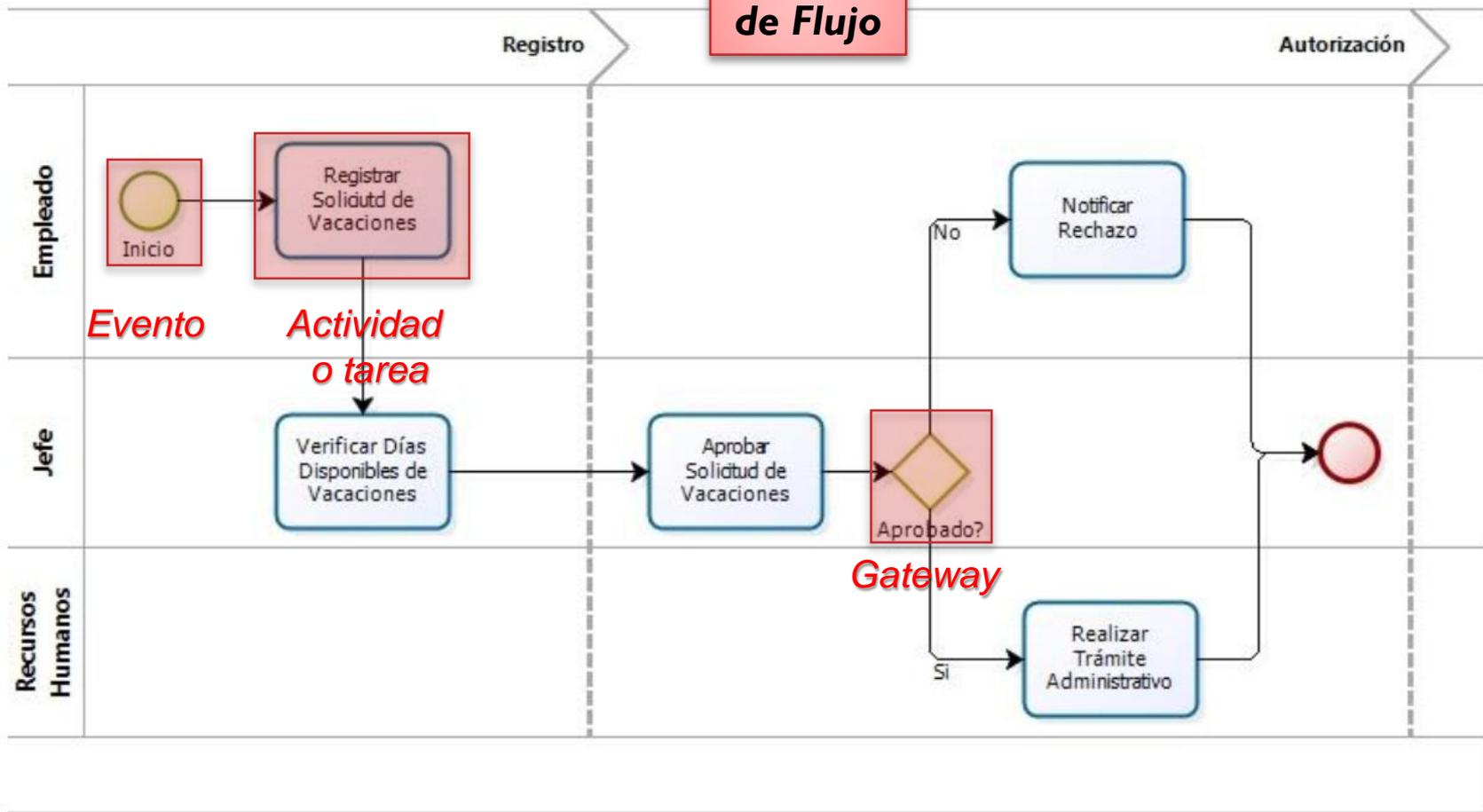


BPMN

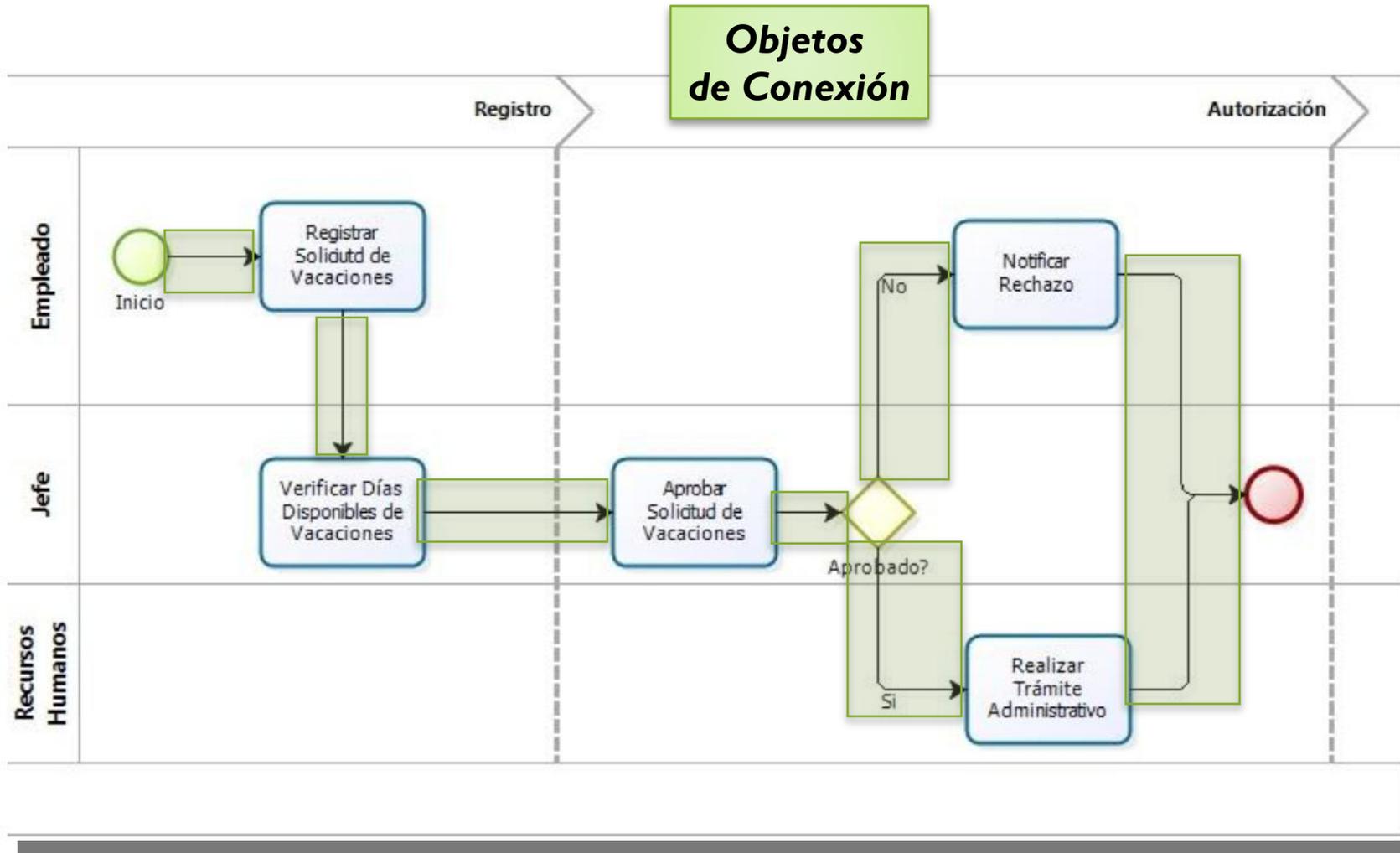
- Modela los procesos como **Flujos de trabajo**
 - *Fácilmente entendibles por los usuarios NO técnicos*
- **ELEMENTOS**
 - *Objetos de flujo:*
 - eventos, actividades, gateways (rombos de decisión)
 - *Objetos de conexión*
 - Flujos de secuencia, de mensaje y de asociación
 - *Swimlanes*
 - *Pool, Lane y Fase*
 - *Artefactos*
 - *Objetos de datos, grupos, anotación*

BPMN

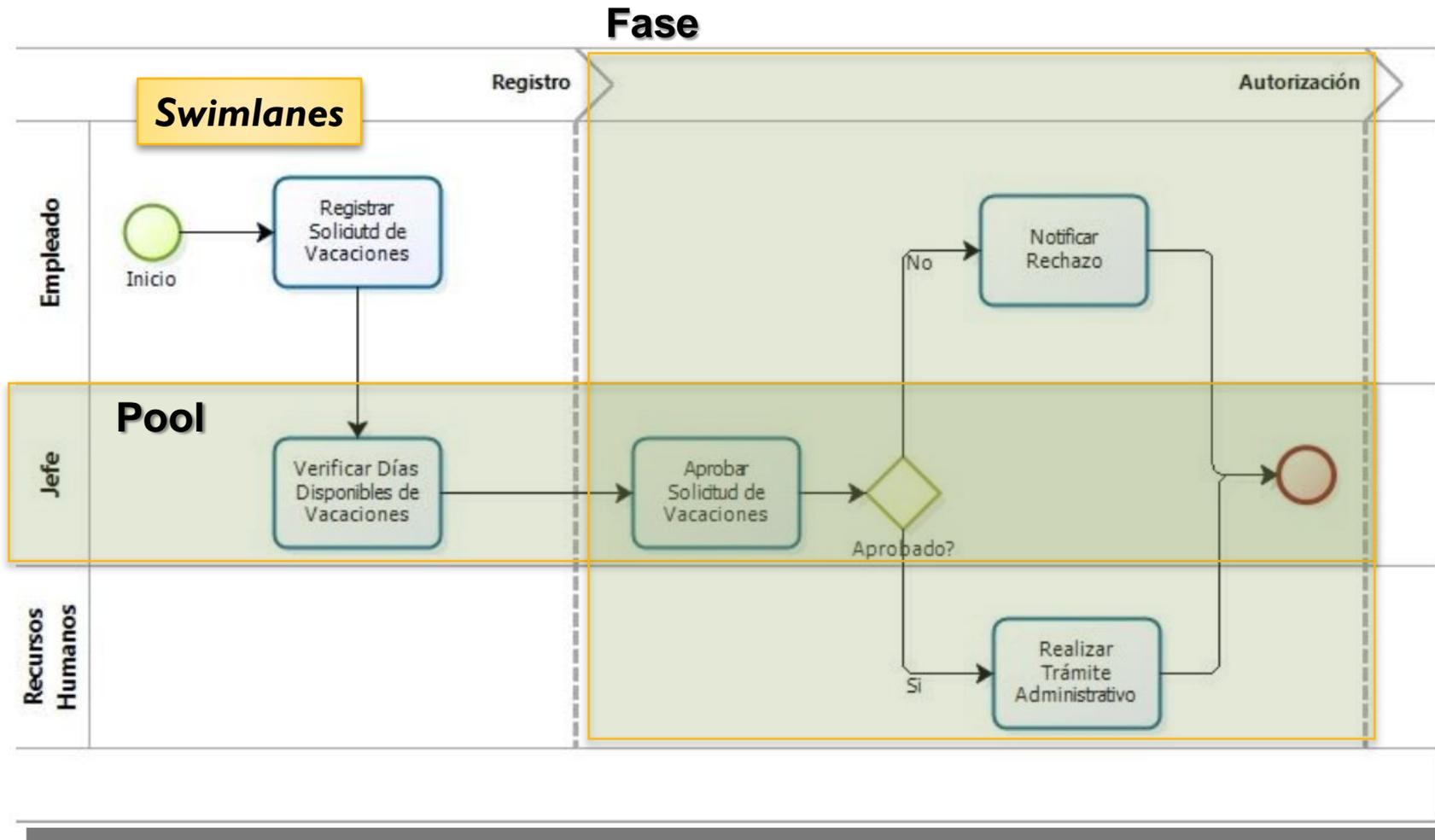
Objetos de Flujo



BPMN



BPMN



Business Process Execution Language - BPEL

- Lenguaje estandarizado, **basado en XML**, para la composición de procesos a través del intercambio de información mediante servicios web
- Creado por *Organization for the Advancement of Structured Information Standards (OASIS)*
- Basado en WSDL
- **Lenguaje de Orquestación de Procesos**
- **NOTA:** La versión original no comprendía tareas humanas, sólo interacción entre sistemas artificiales.



BPEL4People

- **BPEL no soporta tareas humanas.**
 - BPMS comerciales hicieron desarrollos propios para soportar dichas interacciones
- Especificación creada por las **grandes compañías de BPMS** para incorporar **tareas humanas** en BPEL.
 - Bea, IBM, Oracle, SAP, etc.
- Modela gente dentro de los procesos VVS-BPEL



Business Activity Monitoring - BAM

- Monitoreo en tiempo real de los *KPI (Key Performance Indicators)*
- **Control estadístico** de procesos
- Aplicación de **Process Intelligence**
- **Uso de herramientas analíticas OLAP**
- Incorporación de alarmas ante ciertos umbrales
- Ayuda a la automatización de las reglas de negocio
- Facilitan aplicación de metodologías como **Six Sigma** sobre procesos



Herramientas disponibles

- **Comerciales**

- BeaSystems
- IBM
- Oracle
- SAP
- Microsoft
- Auraportal
- Bizagi

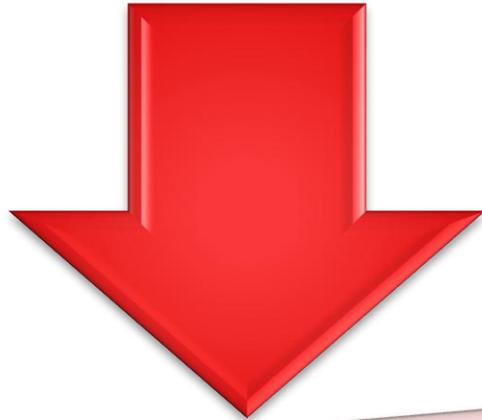


- **Open Source**

- **BPM**
 - jBPM
 - uEngine
 - Bonitasoft
- **Workflows**
 - ProcessMaker



Costo / Beneficio



COSTOS

- Herramientas muy costosas.
- Requieren alta especialización
- Integración e implementación compleja
- Cambio en cultura organizacional



BENEFICIOS

- Gestión documental
- Simulación de rediseños
- Procesos son modelados y difundidos en toda la organización.
- Monitoreo, automatización de reglas de negocio y alarmas.
- Integración de sistemas informaticos.
- Inteligencia de Negocio





Solución 3

° **USAR FRAMEWORK
MVC**

Temario

- Patrón de diseño de software MVC
- MVC y Arquitectura de 3 capas
- Concepto de Framework
- Alternativas disponibles
- Costo / Beneficio

Patrón MVC

- **MVC: Modelo Vista Controlador**
 - Es un *patrón de diseño* en ingeniería de software.
 - Separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
 - Permite la especialización de los desarrollos
 - Trabajo en paralelo
 - Estandarización de las interfaces de comunicación
- **Modelo**
 - Es la interfaz entre la capa media y la capa de datos
- **Vista**
 - Es la interfaz entre la capa media y la capa de presentación.
- **Controlador**
 - Programas que se ejecutarán en la Capa Media

Historia MVC

- Descrito en 1979 por **Trygve Reenskaug**, que trabajaba sobre el lenguaje **Smalltalk** para la compañía Xerox PARC
 - Paper “*Applications Programming in Smalltalk-80: How to use Model–View–Controller*”
 - **NOTA:** Smalltalk es un lenguaje de programación **estrictamente orientado a objetos**.
- Ha sido **interpretado de varias maneras**, pero el concepto básico sigue siendo el mismo.
- En el 2002, la W3C estandarizó la arquitectura XForms bajo conceptos MVC para todas las futuras aplicaciones web.

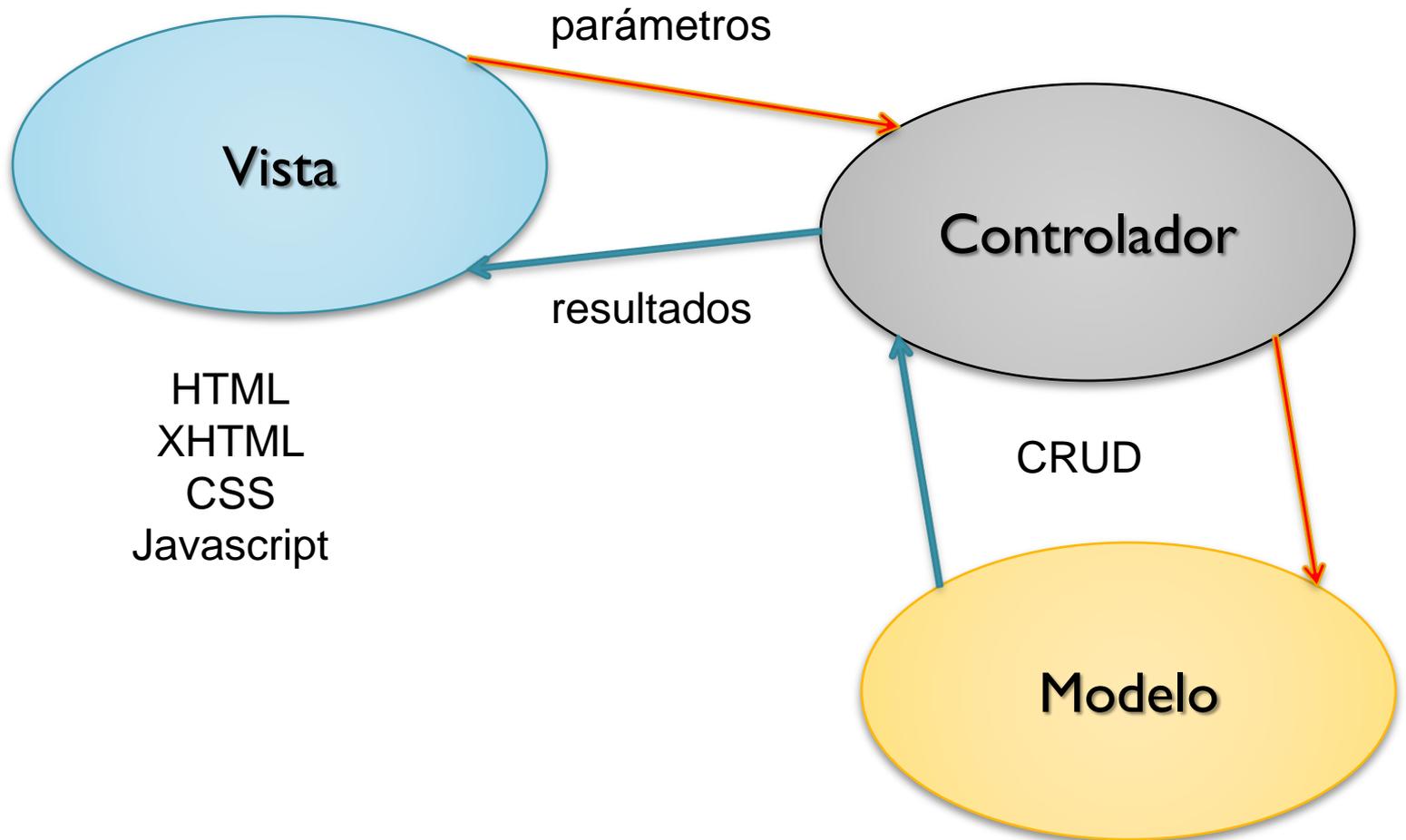
MVC como patrón de arquitectura

- MVC calza perfectamente con una **arquitectura de 3 capas** o **arquitecturas de n capas para sitios web muy transaccionales**.
 - Los **modelos** comprenden la interfaz a través de la cual la Capa de Datos (Servidor de datos) se comunicará con la Capa Media o de Negocios (Servidor de aplicaciones).
 - Facilitarán el uso de las operaciones CRUD
 - Las **vistas** representan la interfaz de comunicación de la Capa Media (Servidor Web) con la Capa de Presentación (Computador Cliente)
 - Templates para generación de HTML, XHTML, CSS y/o Javascript.
 - Los **controladores** serían los programas que se ejecutarían en los **Servidores de Aplicaciones**

MVC como patrón de diseño

- Desde el punto de vista de **Ingeniería de Software**, el modelo MVC puede ser visto como un patrón de diseño para aplicaciones que sirven redes de computadores.
 - **Modelo** como capa de persistencia de datos
 - **Vista** como el generador de interfaces para la entrada y salida de datos de la aplicación
 - **Controlador** que atiende peticiones a través de objetos, clases y/o funciones parametrizadas:
 - Los parámetros llegan a través de un formulario generado por la vista.
 - El controlador opera con los datos a través del modelo
 - Los resultados son enviados a través de la interfaz descrita por la vista

MVC como patrón de diseño



Patrón MVC: Modelo

- Es la **representación de los datos** que utilizará la aplicación.
 - Puede comprender más de un *Sistema Administrador de Base de datos*
 - Por lo general se construye en base a **objetos**
 - Uso de Object Relational Mapping ORM
 - Cada entidad es una clase
 - Los atributos son definidos como atributos de la clase
 - Ejemplo:

```
from django.db import models

class Person(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```

Patrón MVC - Vista

- Generador de **interfaces de usuario**
 - Uso de **templates**
 - Programas que generan *HTML, XHTML, CSS & Javascript*
 - **Formularios**
 - Para el ingreso de parámetros en la aplicación
 - Para consultar BBDD
 - Para administrar BBDD
 - **Renderiza resultados de las peticiones**
 - Contenido dinámico
 - Tablas con datos
 - Gráficos
- La vista es la componente que genera el código que será **interpretado en el Browser del Cliente**

Patrón MVC: Controlador

- Programas que responden a las solicitudes de los clientes haciendo uso de los datos de la aplicación.
- Corresponden al **CORE** de la aplicación
- *Funciones parametrizadas*
 - El funcionamiento del controlador **depende de los parámetros** enviados a través de la vista
 - El controlador **puede modificar el modelo**, por lo tanto, la estructura y contenido de la base de datos
- Requiere de una **conexión permanente** con la base de datos
 - *Encapsulación*

Framework

- **Estructura de soporte** mediante el cual un proyecto de software puede ser *organizado y desarrollado*
- Incorporan una **metodología o patrón de diseño de software**
 - Mejores prácticas de desarrollo
- Generalmente comprende:
 - **Librerías** para el desarrollo de tareas de uso común
 - Programas para **la coordinación de las componentes**
 - **Lenguaje de programación** para el desarrollo

Framework para el desarrollo web

● Características

- Seguridad
 - ◆ Autenticación y autorización
- Acceso y Representación de datos
 - ◆ ORM: Object Relational Mapping
- Mapeo de URL
 - ◆ Optimización de búsquedas
- Sistema para creación de plantillas o Templates
- Manejo de memoria Cache
- Integración con AJAX
- Configuraciones automáticas
 - ◆ Principio Convención sobre Configuración
- Proveen de API para implementación de Web services
 - ◆ **SOA**: Service Oriented Architecture
 - ◆ **RESTful**: Representational State Transfer

¿Por qué usar un Framework?

- El desarrollo web implica desarrollar **funcionalidades recurrentes**
 - El framework provee una librería con aquellas funcionalidades implementadas
- El **equipo desarrollador es multidisciplinario**
 - El framework fija una **pauta para separar las tareas**
 - **Máxima cohesión, mínimo acoplamiento**
- Es deseable la reutilización del código
 - El framework lo facilita
- Algunas aplicaciones web son complejas
 - El framework permite **dividir para conquistar**, incorpora metodología o patrón de diseño con mejores prácticas de desarrollo.

Frameworks MVC

- Corresponden a frameworks orientados a la construcción de sitios web dinámicos, aplicaciones web y servicios web usando el **patrón MVC**.
- Comenzaron a popularizarse con la irrupción de *Ruby on Rails*
- Suelen usar un **lenguaje interpretado** para el desarrollo.
 - PHP, Perl, Python, Ruby, ASP.
 - Sin embargo, hay frameworks que hacen uso de lenguajes compilados como Java



• Alternativas Disponibles

Frameworks en PHP

Librerías de desarrollo

- Las funciones siguen MVC
- Zend Framework



- Ez Components



Frameworks

- Implementan MVC



- PHP 4/PHP5
 - CodeIgniter
 - CakePHP
 - Seagull
 - Akelos



- PHP 5
 - Symphony
 - Prado
 - Yii Framework
 - Kumbia PHP



Frameworks en Perl



- **Catalyst Framework**

- Pensado para ser sencillo y simple
- No busca “reinventar la rueda”
- Hace **uso extensivo de las librerías CPAN de Perl**
- Funcionamiento
 - Usando **FastCGI** en cualquier servidor web: Apache, IIS
 - **Mod_perl** en Apache
- Filosofía:
 - **There Is More Than One Way To Do It**
 - **Don't Repeat Yourself (DRY)**

Frameworks en Python

Full Stack Framework

- Django

The Django logo consists of the word "django" in a white, lowercase, sans-serif font, centered within a dark green rectangular background.

- Web2py

The Web2py logo features the text "web2py" in a white, lowercase, sans-serif font, with the "2" in orange. It is set against a dark teal rectangular background with a thin red border.

Glue Framework

- Pylons

The Pylons logo features a yellow sun icon on the left and the word "Pylons" in a white, sans-serif font on the right, all contained within a black rectangular background with a red border.

- TurboGears



Frameworks en Ruby

- El **más extendido framework** para Ruby es Ruby on Rails
- Implementa el patrón MVC
- Está pensado para ser usado bajo una **metodología ágil** de desarrollo
- Filosofía:
 - **Convention over Configuration (CoC)**
 - **Don't Repeat Yourself (DRY).**
- **Paquetes**
 - ActiveRecord
 - ActiveResource
 - ActionPack
 - ActiveSupport
 - ActionMailer



Frameworks en Java

Struts

- Extiende la API de Servlets para facilitar el desarrollo de aplicaciones web bajo MVC
- Hace uso de Servlets y JSP



Light weight Frameworks

- Spring MVC
- Tapestry
- Grails



JSF

- Impulsado por SUN para simplificar el desarrollo de aplicaciones en Java
- Especificación JSR - 314
- Alta integración con AJAX

Costo – Beneficio del Framework



Costos

- Se necesita personal técnico especializado en el uso de la herramienta
- No incorpora las mejores prácticas de negocio si presentes en ERPs y BPMS
- Permanece el GAP entre los requerimientos de negocio y los requerimientos técnicos como en el caso de la solución a medida.



Beneficios

- Facilita el desarrollo ágil de aplicaciones web
- Incorpora mejores prácticas de desarrollo
- Ordena código y facilita mantención
- Es más barato que un BPMS

