

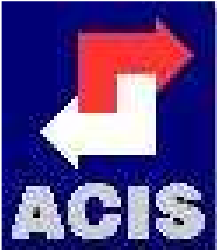
Planeación y gestión del cambio hacia una nueva tecnología de desarrollo

Victor Manuel Toro C.

VictorToro@cincosoft.com

XXVI Salón de **INFORMÁTICA**

La gobernabilidad de TI: Una responsabilidad y
reto para los directivos de TI

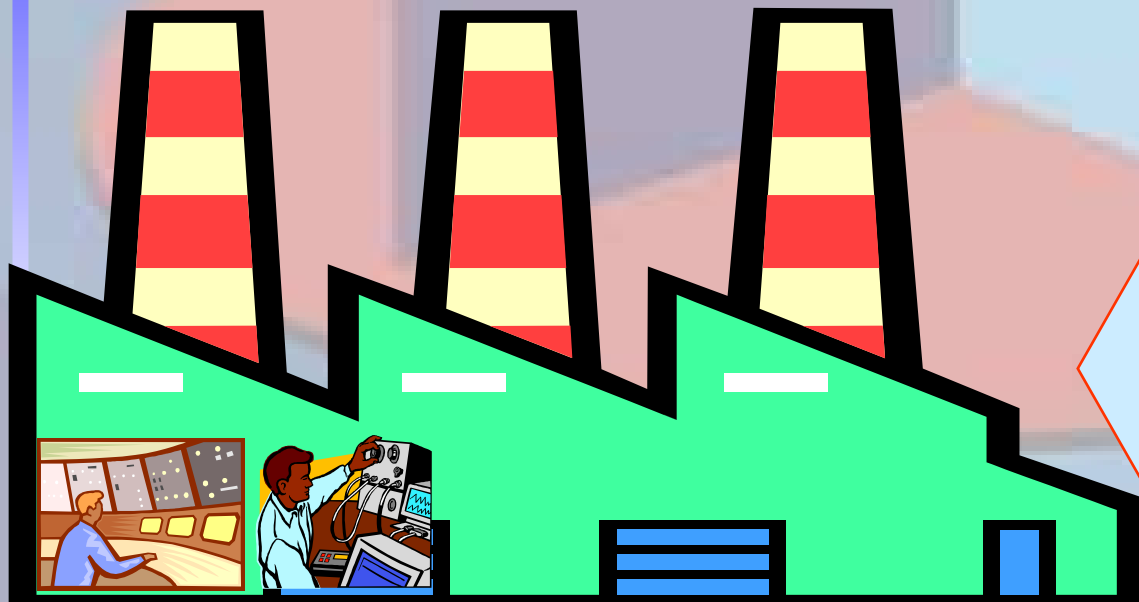


Hasta hace algunos años:

El papel de la informática en la empresa

Antes:

La informática era Respaldo y Soporte “a posteriori”



Vendedores
Agentes
Representantes
“Brokers”
Distribuidores
Asesores
Concesionarios
...



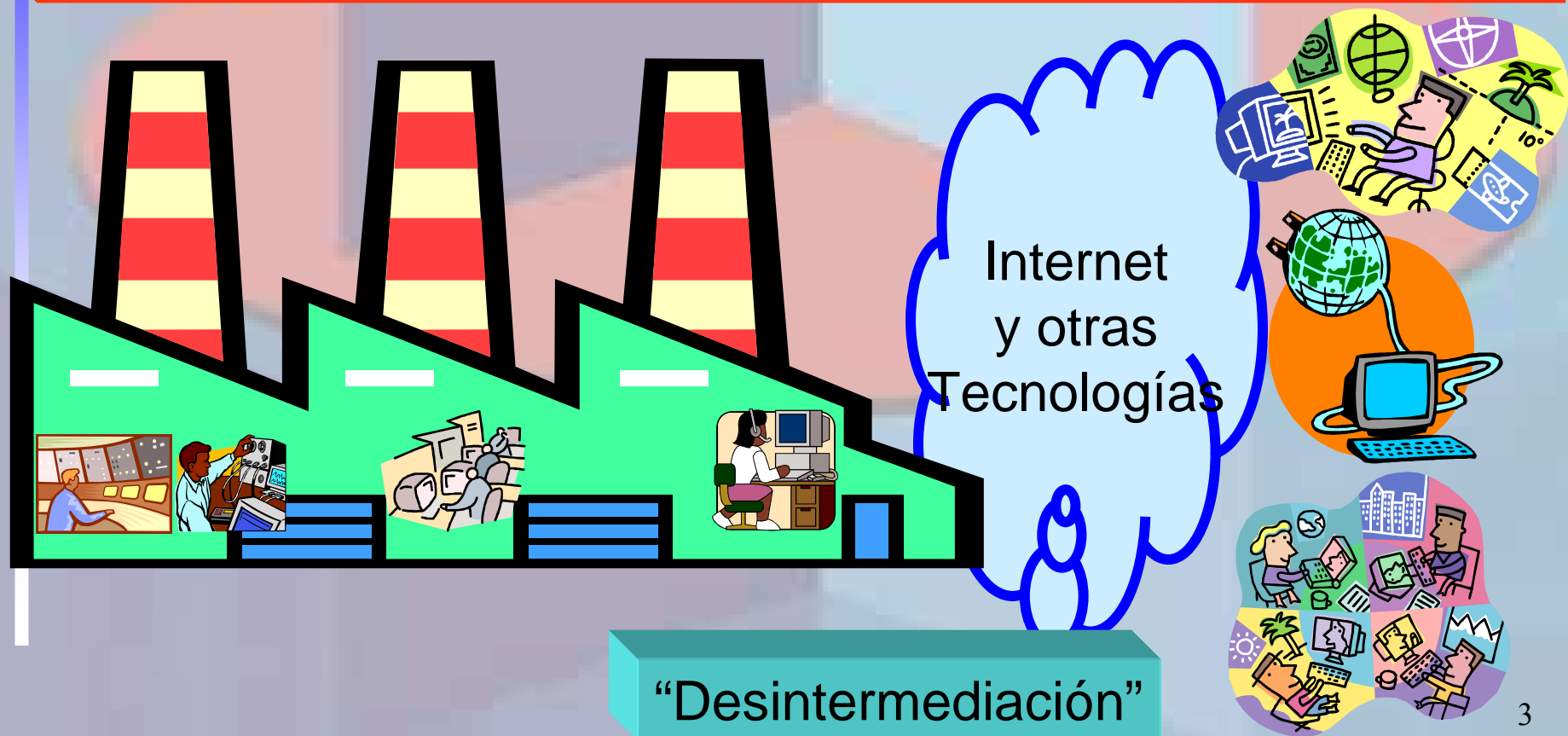


Ahora:

El papel de la informática en la empresa

Hoy:

La informática está en **toda** la empresa,
es la “fachada” de la empresa,
y muchas veces, es el principal/único contacto con el cliente





Si la informática es la “fachada” de la empresa ...

¿Cómo lograr una buena “fachada” ?

- Alinear los sistemas de información con los objetivos de la empresa ... dentro de un contexto globalizado!

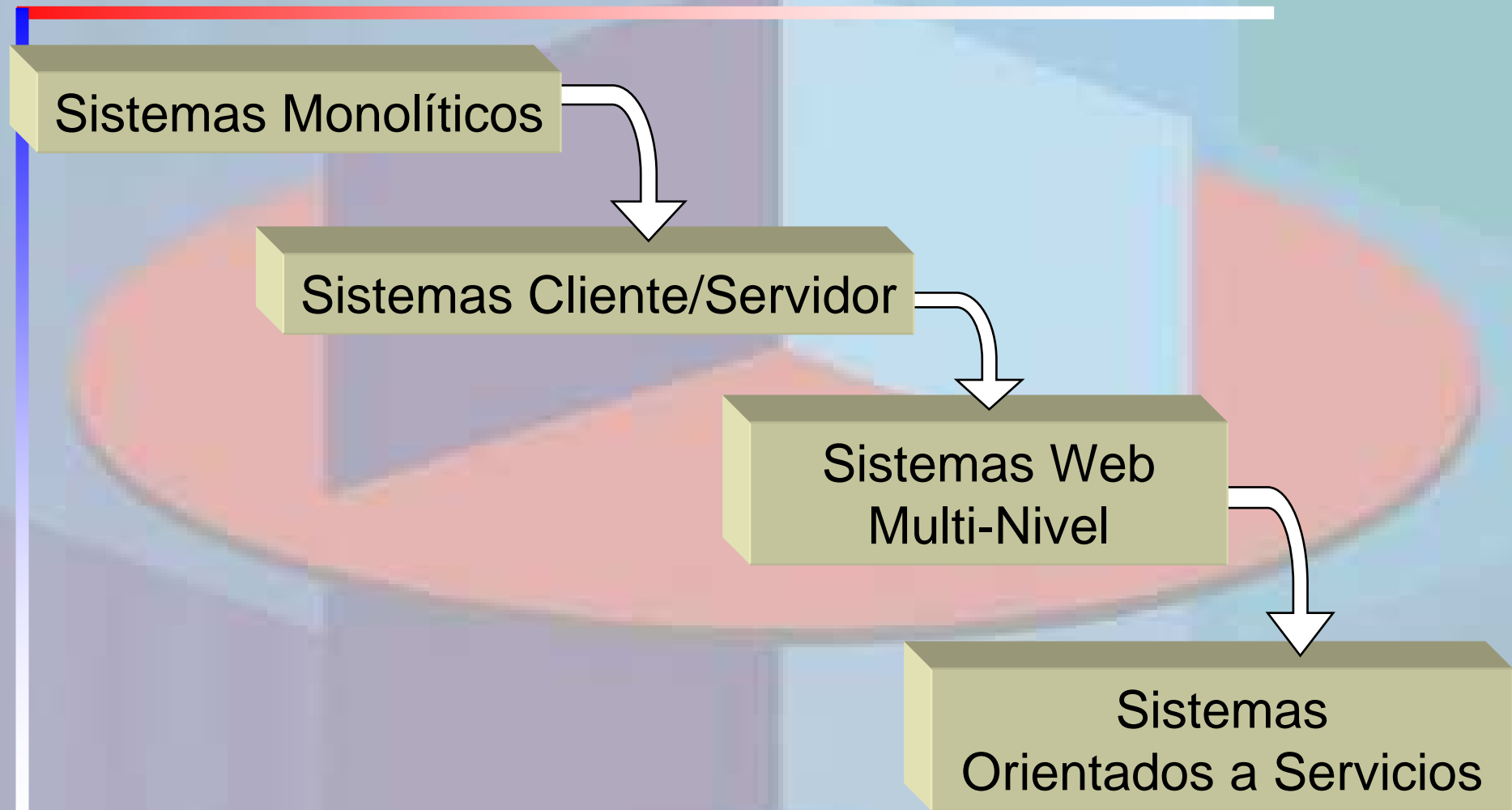
- Desarrollar los sistemas de información con una arquitectura moderna orientada a:
 - ↓ Internet
 - ↓ a Servicios (i.e. a la integración)
- Apoyarse en las nuevas tecnologías y metodologías
- Seguir procesos modernos de desarrollo de software !!



Evolución de la Arquitectura

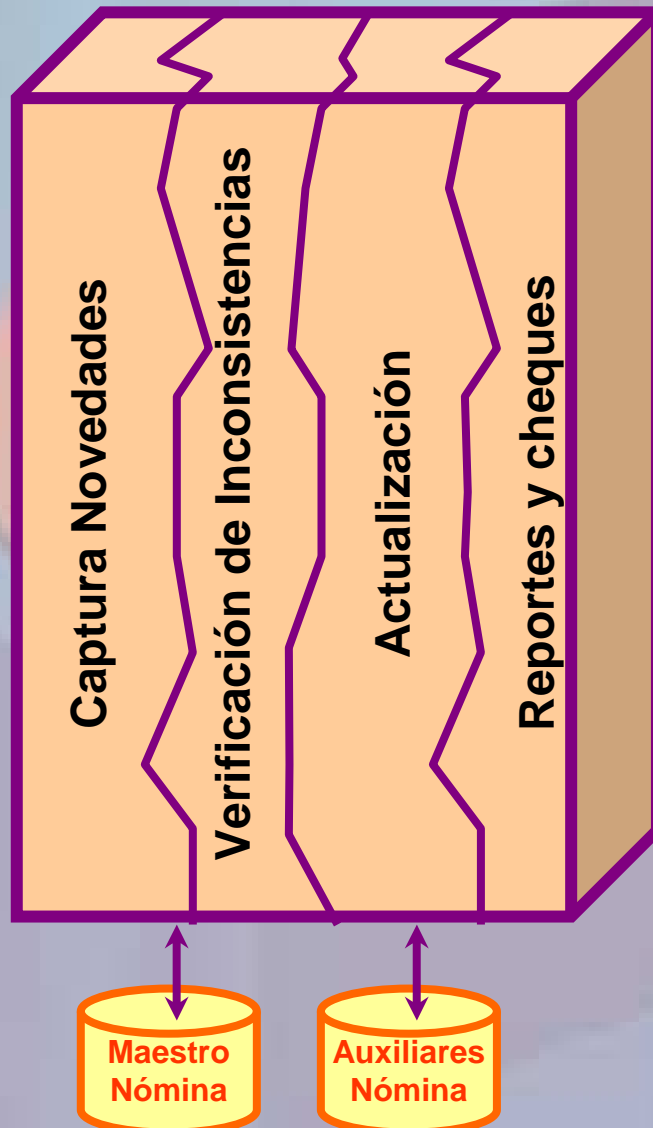


Evolución de la Arquitectura de los Sistemas de Información



Sistemas Monolíticos

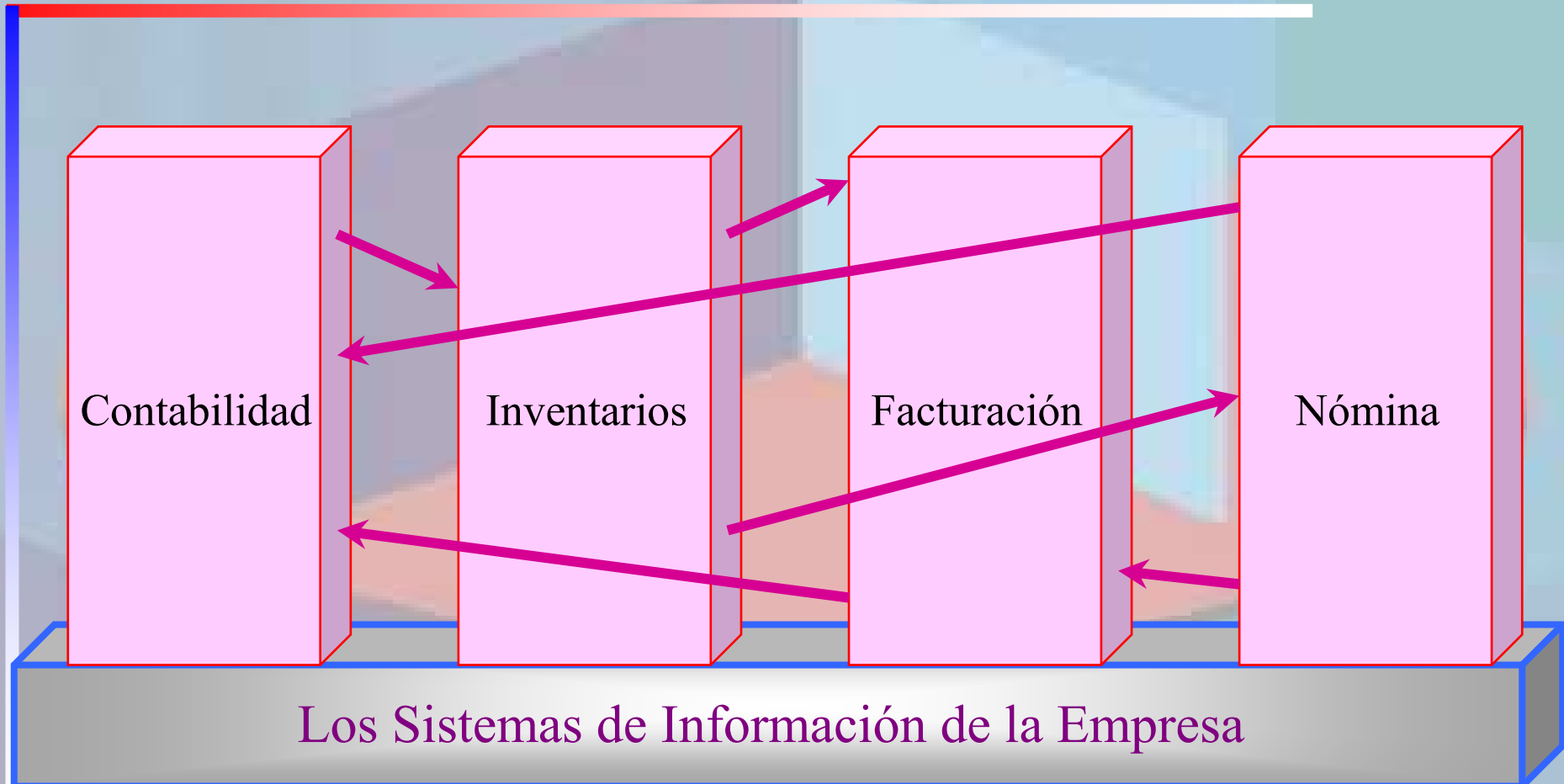
Sistema de Nómina



- Uno o varios programas
- Funcionan en un solo computador
- Comparten algunos archivos
- En el mismo computador están:
 - ☒ Programas
 - ☒ Datos
 - ☒ Usuarios



Sistemas de Información de la Empresa: Campo de Monolitos



- Totalmente Centralizado
- Costosamente Escalable
- Difícil intercambio de información

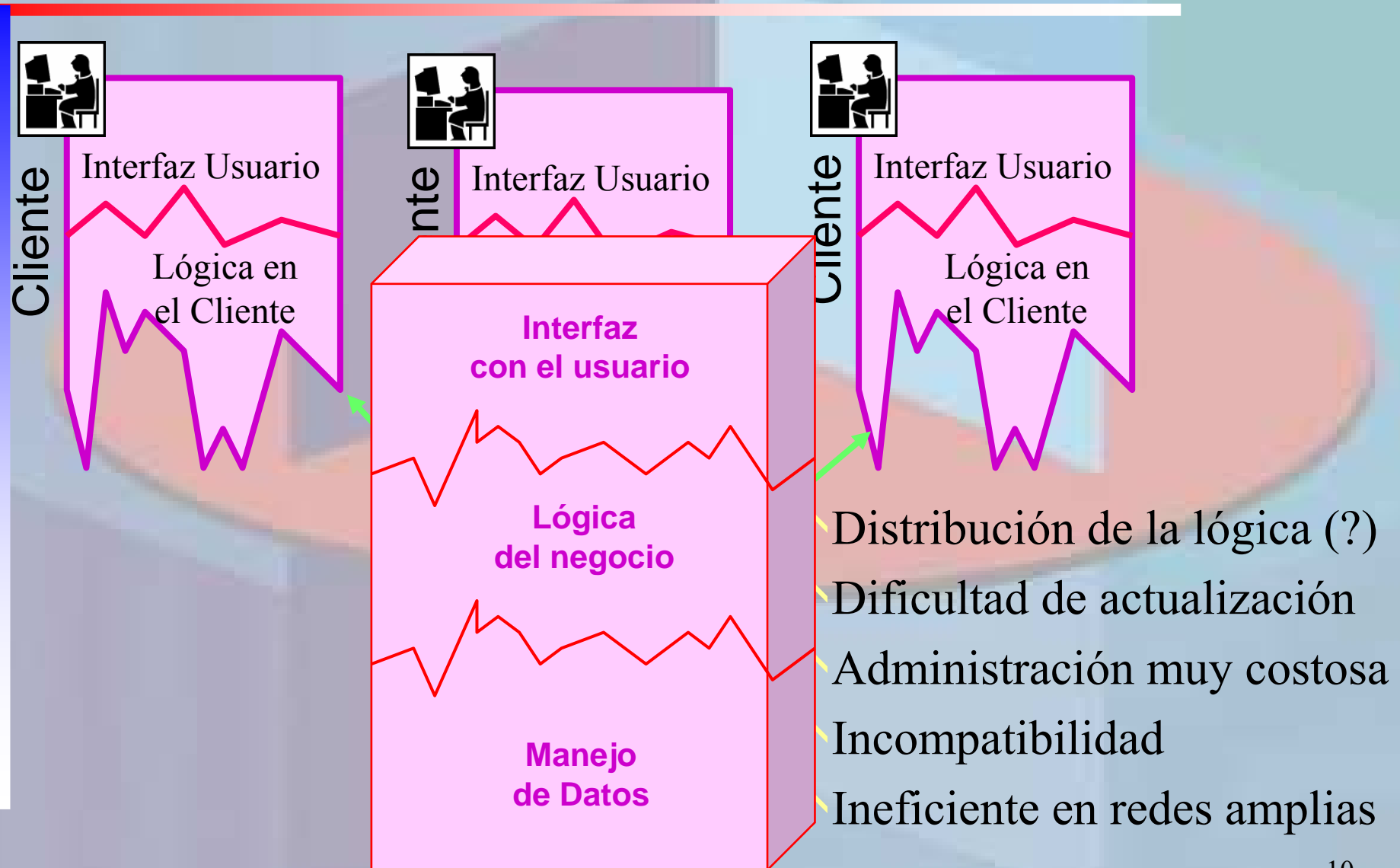
Mirando al interior de un Sistema de Monolítico

Un Sistema de Información



- Código con funcionalidad entremezclada.
- Difícil mantenimiento.
- Mínima reutilización.
- Portabilidad imposible
- ...

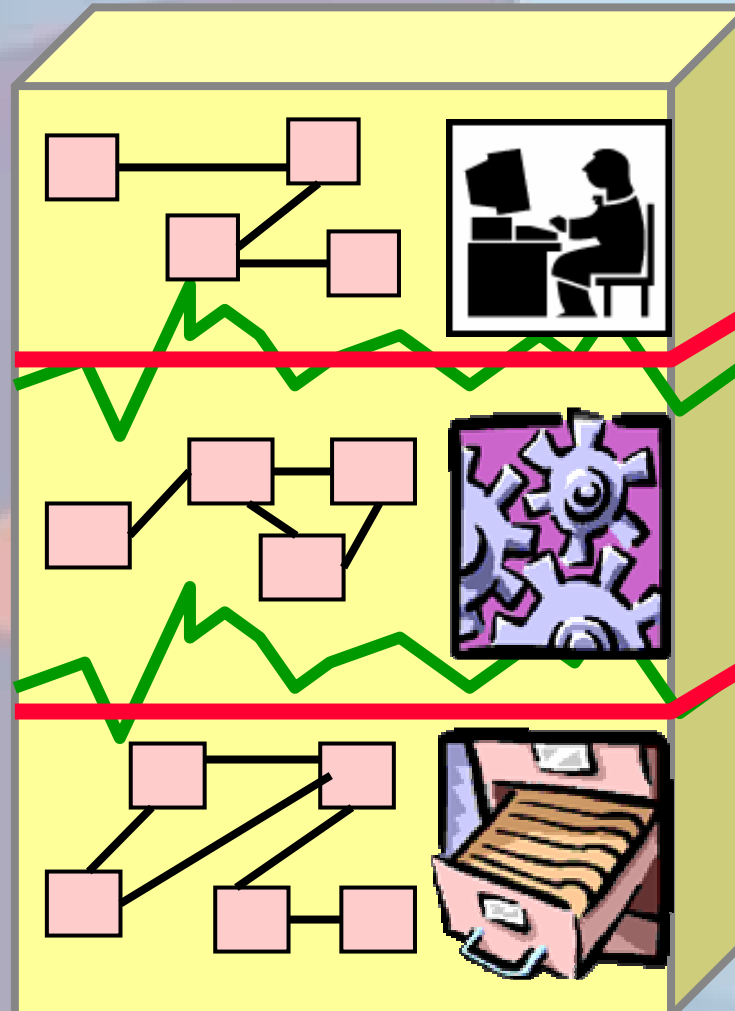
Arquitectura Cliente Servidor en 2 Niveles





Evolución hacia los Sistemas de 3 (ó mas) Niveles

Interfaz
con el usuario



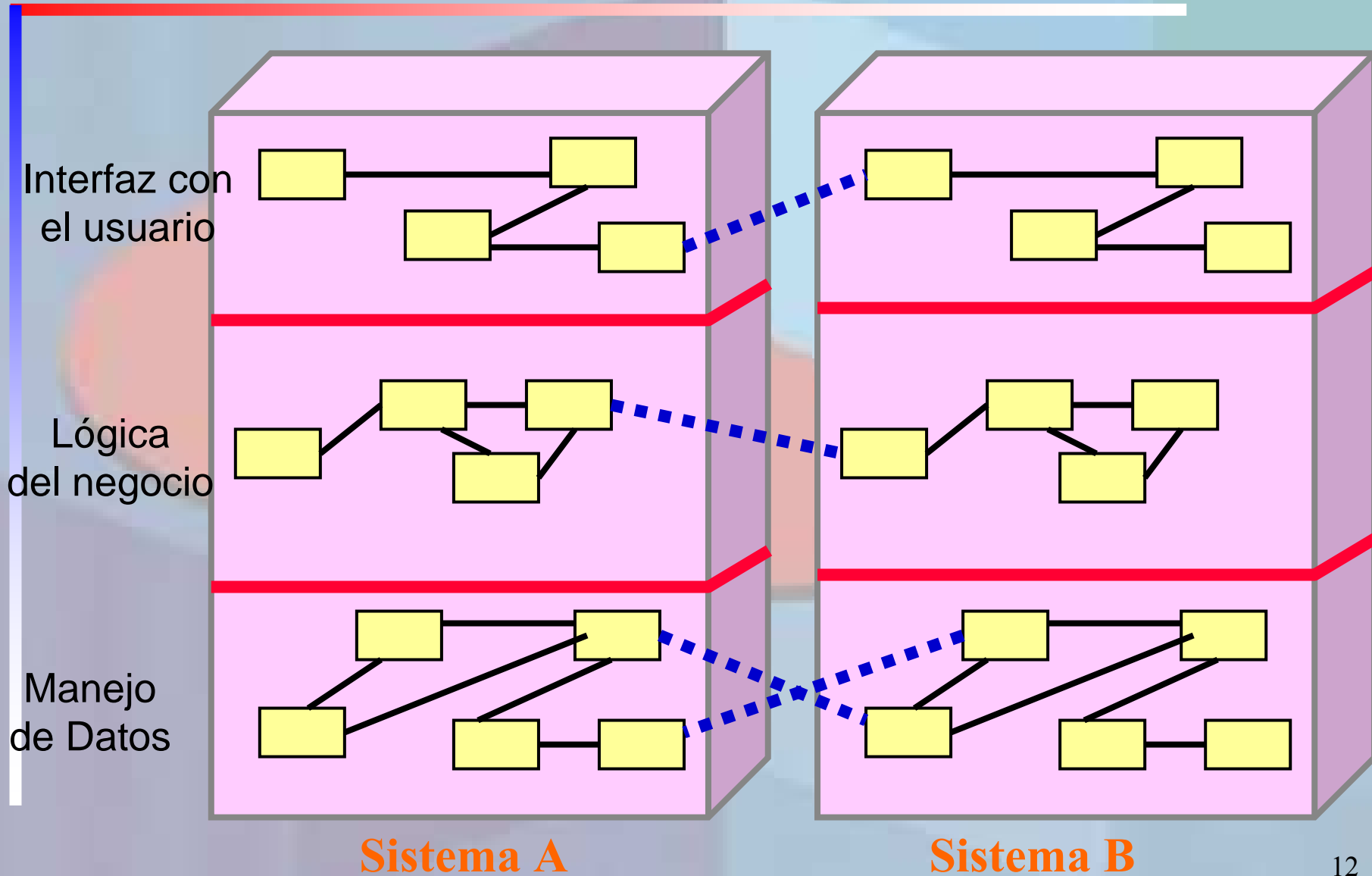
Lógica
del negocio

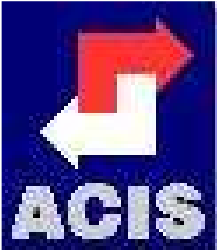
Manejo
de Datos

Sistema de Información



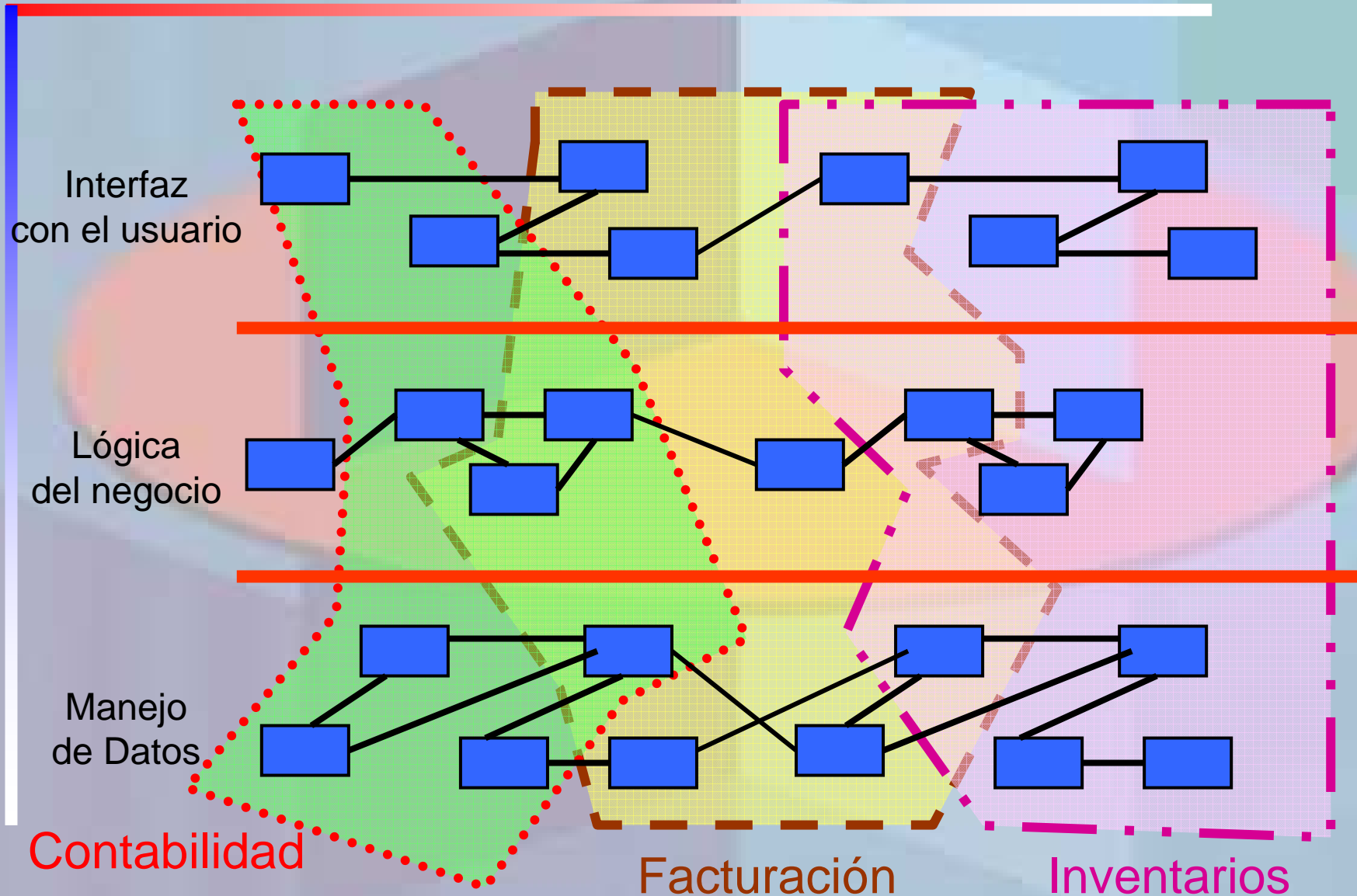
Evolución hacia los Sistemas de 3 (ó mas) Niveles



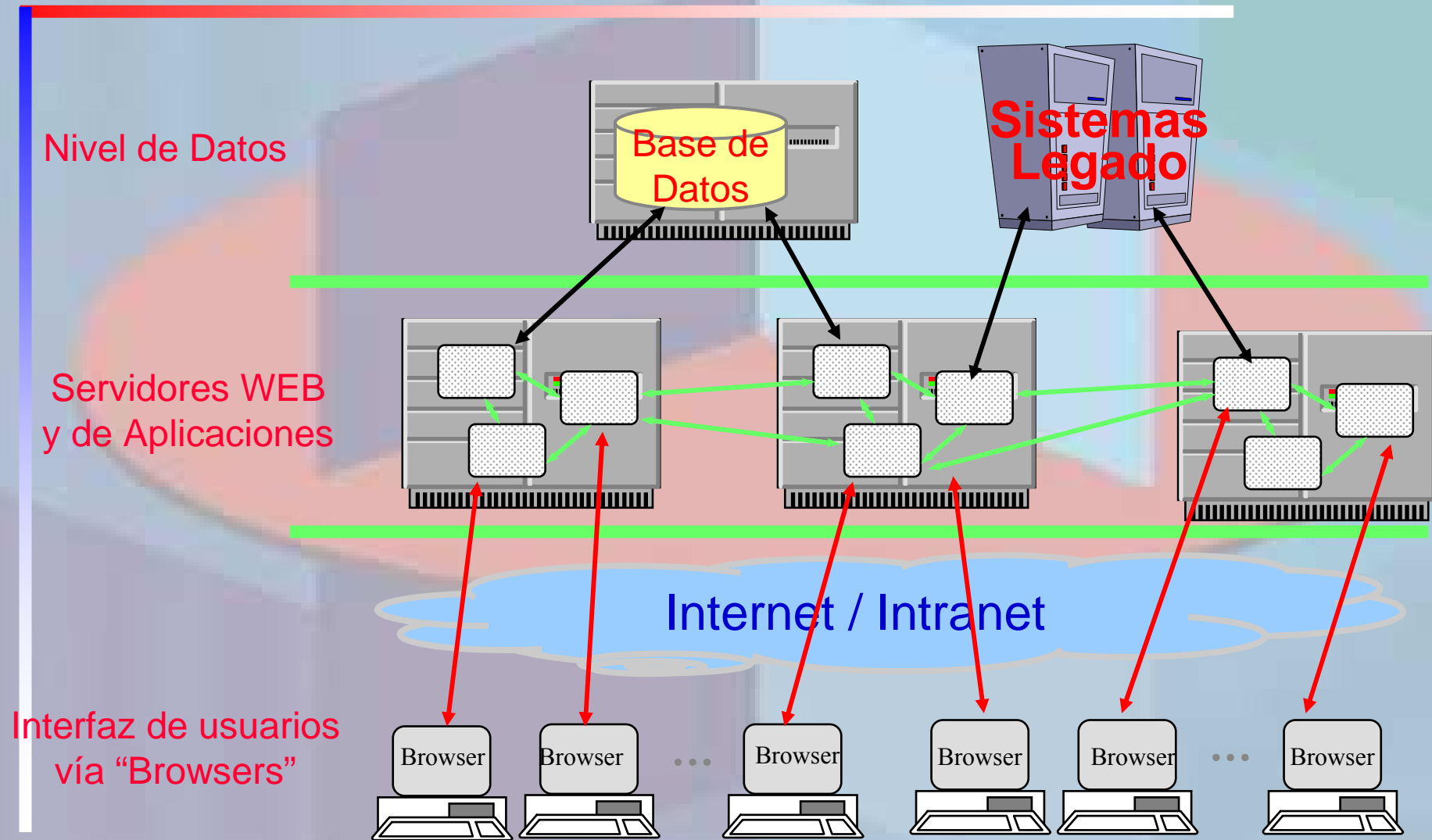


Sistemas de Información de la Empresa

Redes de Componentes de Software



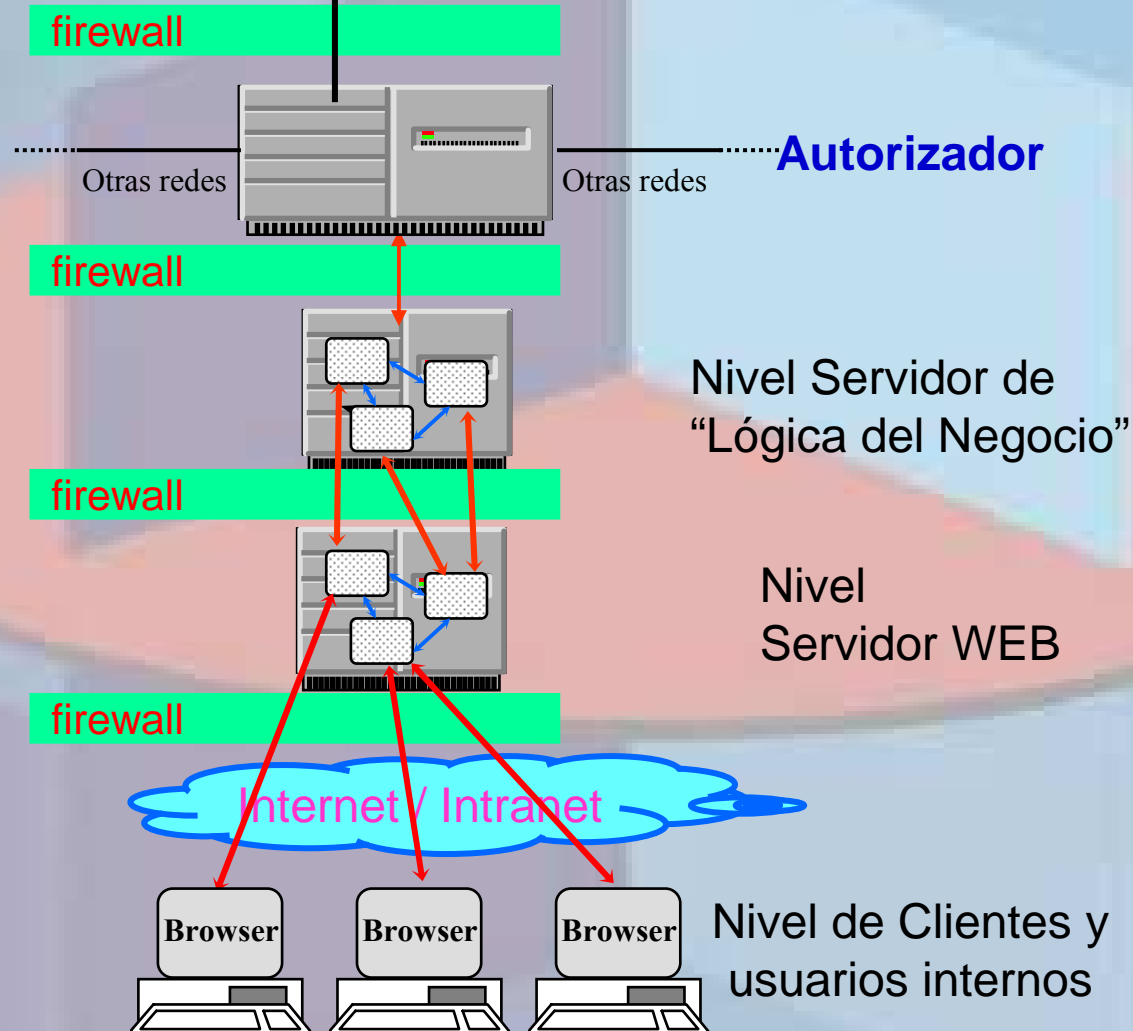
Sistemas de Información en 3 (ó mas) Niveles

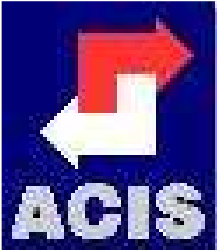




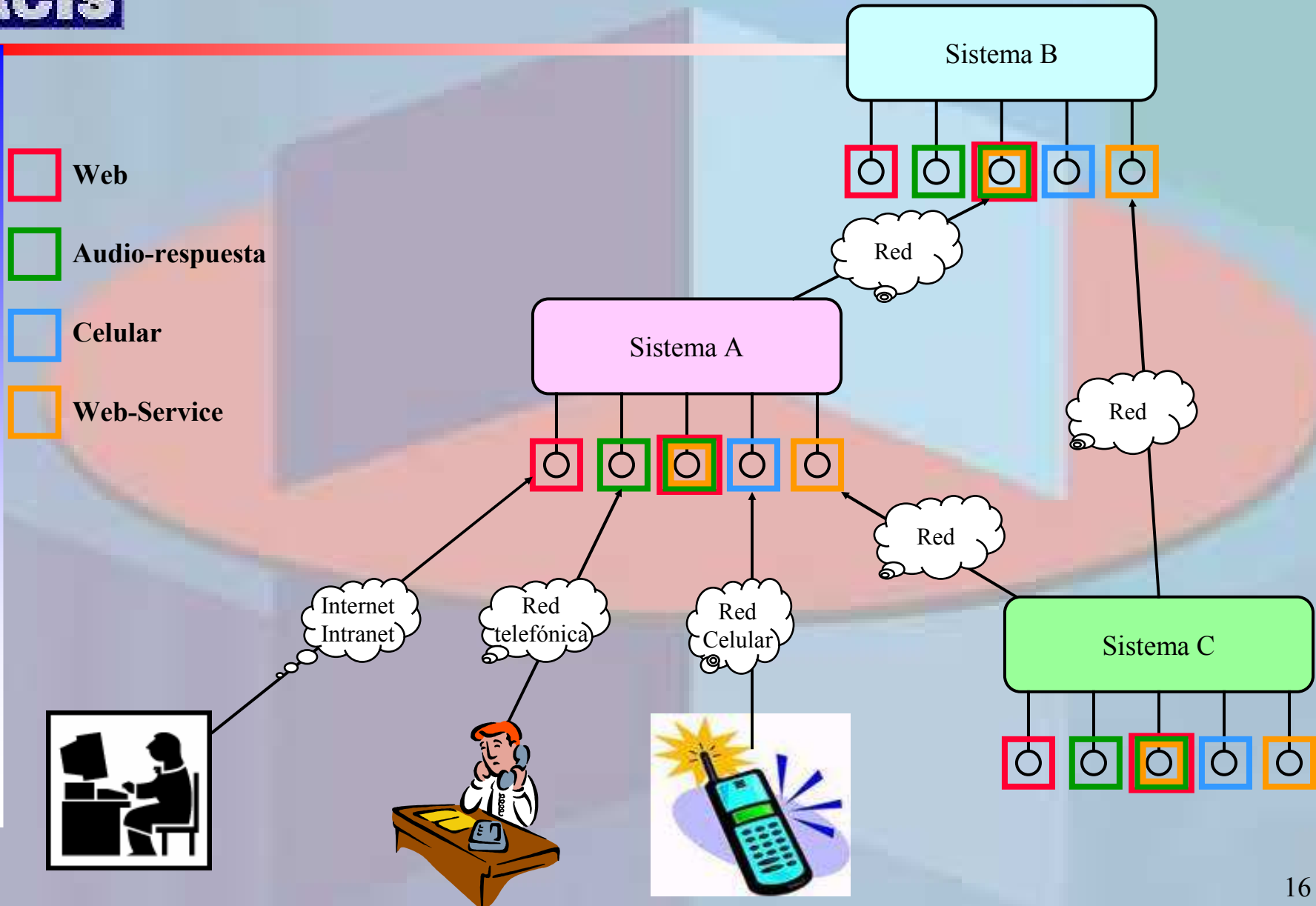
Sistema Bancario

Sistemas Especializados **en 5 niveles**





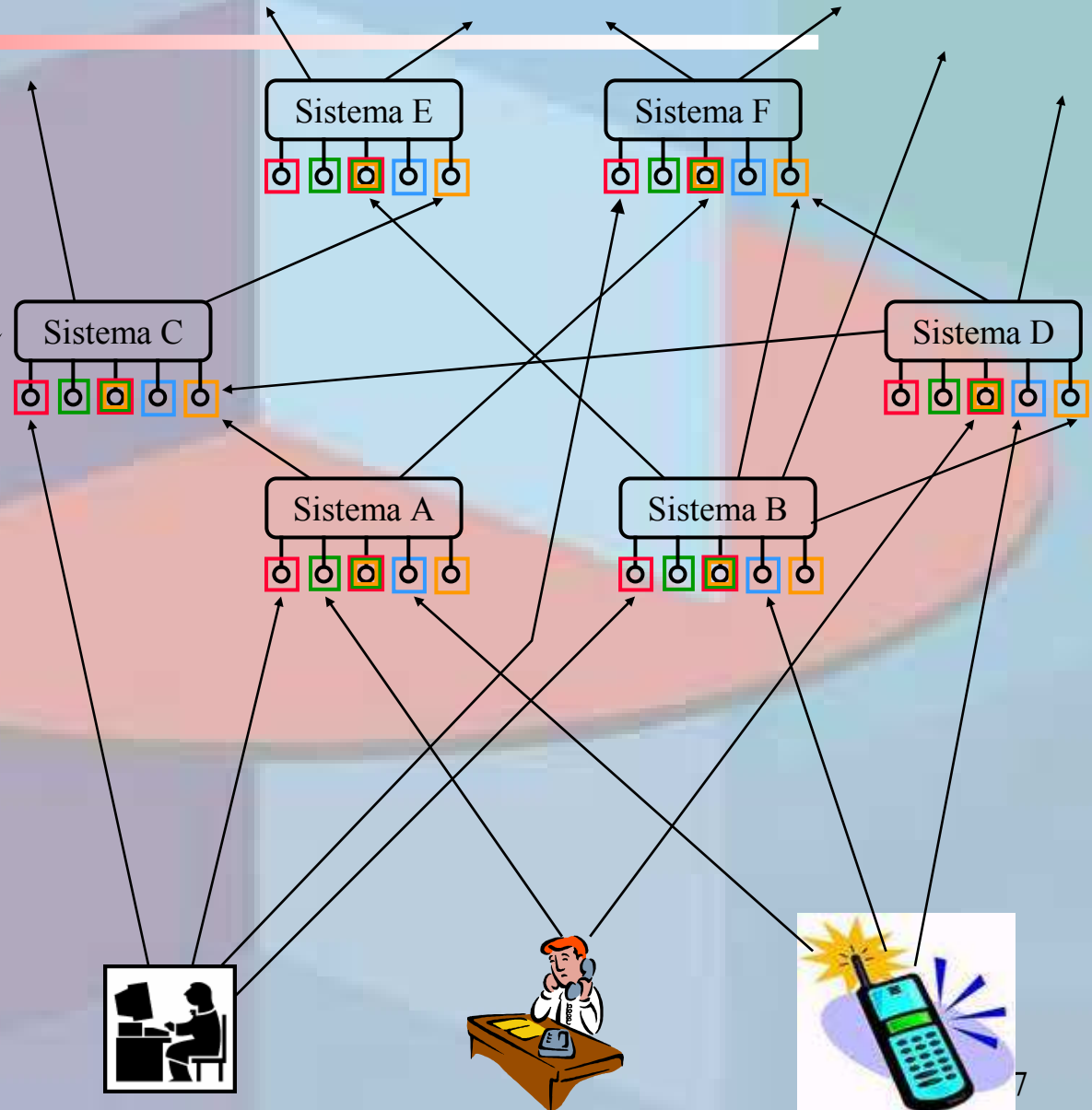
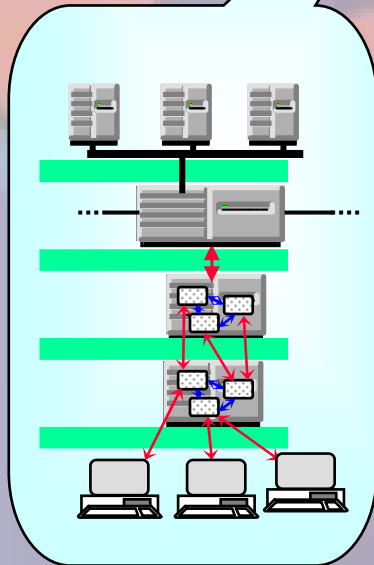
Sistemas Orientados a Servicios





Sistemas de Información de la Empresa Arquitectura Empresarial Orientada a Servicios

- Web
- Audio-respuesta
- Celular
- Web-Service





¿ Cómo se construyen estos
nuevos sistemas ?

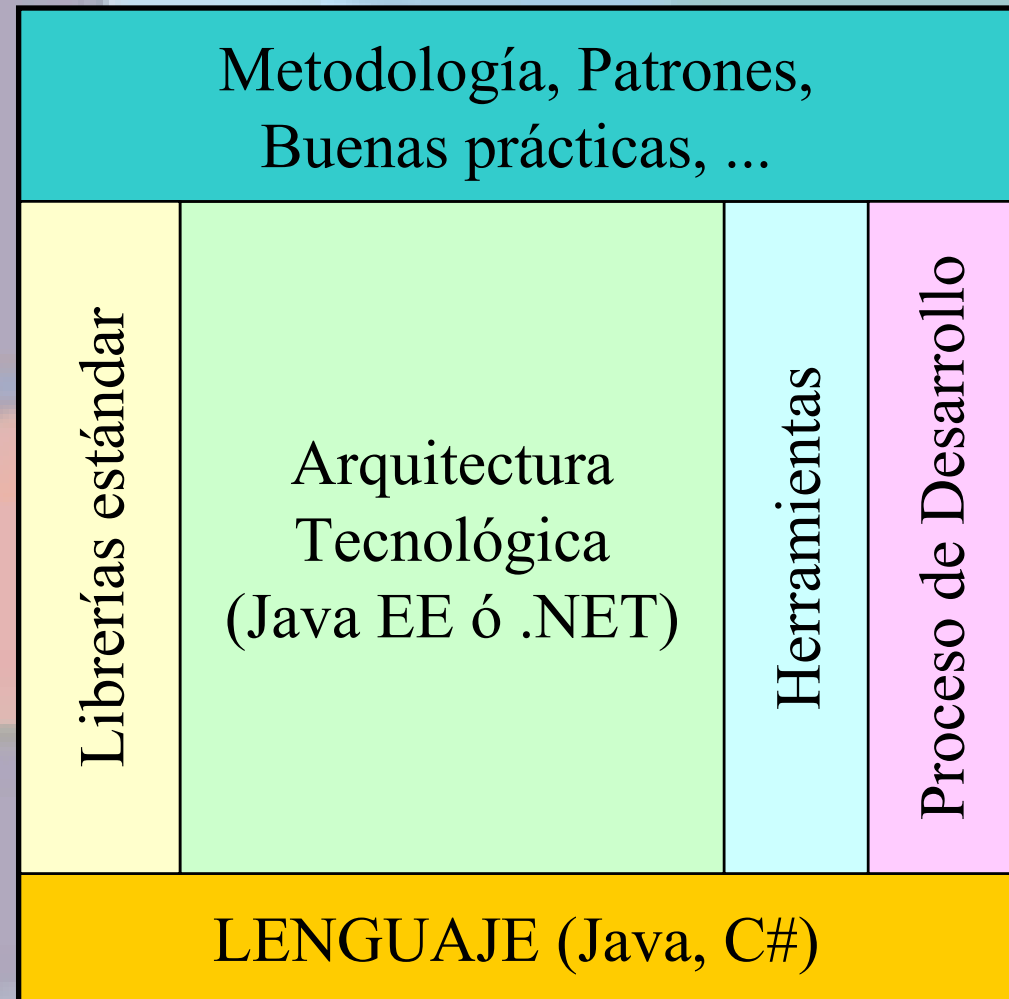


¿Cómo se construyen los nuevos sistemas ?

- Apoyándose en las nuevas plataformas tecnológicas:
 - ↓ Java EE
 - ↓ Microsoft .Net
 - ↓ LAMP (Linux, Apache, MySQL, Php/Perl/Phyton ... y sus variantes)
- Usando herramientas modernas:
 - ↓ Eclipse, ant, xml, JUnit, ...
 - ↓ VisualStudio (Microsoft)
 - ↓ numerosas herramientas libres
- Siguiendo procesos modernos de desarrollo de software
 - ↓ RUP (Rational Unified Process)
 - ↓ Extreme Programming
 - ↓ AUP (Agile Unified Process), ...

¿ Entonces ... cuál es el problema?

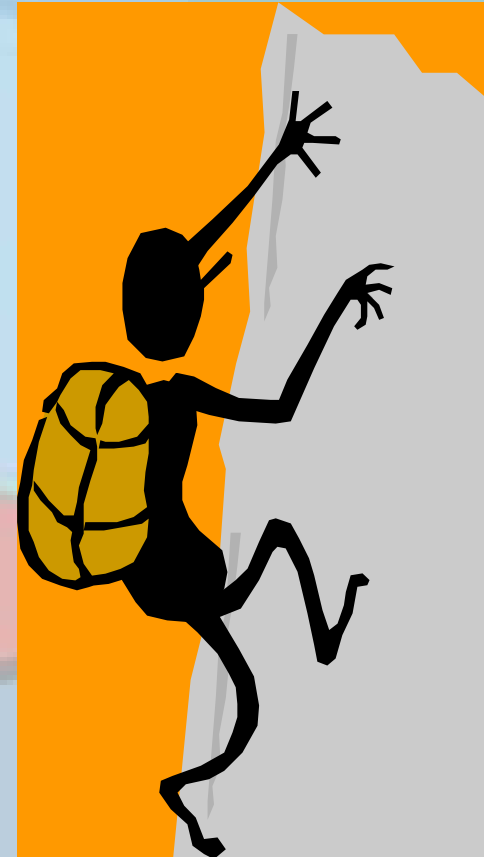
¡ Hay demasiado por aprender !



¿ Entonces ... cuál es el problema?

La curva de aprendizaje es larga!

- Documentación abundantísima
(decenas de miles de páginas) !!
- Leer ↻ Ensayar ↻ Leer ↻ Ensayar ↻
...
- Estudiar buen código fuente
- Ir creando su propio:
 - ↓ estándar
 - ↓ librería
 - ↓ manual
 - ↓ buenas prácticas



Pero en concreto ...

¿ Qué es lo que
de verdad se necesita ?



¿Qué es lo que de verdad se necesita para que un grupo de desarrollo esté listo para un proyecto?

- Proceso de desarrollo que defina secuencia de actividades.
En particular:
 - ↓ Definición y Estándarización de Entregables
 - ↓ Dinámica ágil para el manejo del ciclo central:
 - ↙ diseño (ajustes al) detallado
 - ↙ creación (modificación) del código fuente
 - ↙ detección/registro de problemas (errores, cambios, malentendidos, mal uso)
- Esqueleto global que defina la ubicación y compromisos mínimos de todos los componentes del software
- Estándar de organización y documentación del código
- Selección e integración de herramientas de desarrollo
- Plan de capacitación para los nuevos miembros

Veamos uno por uno:

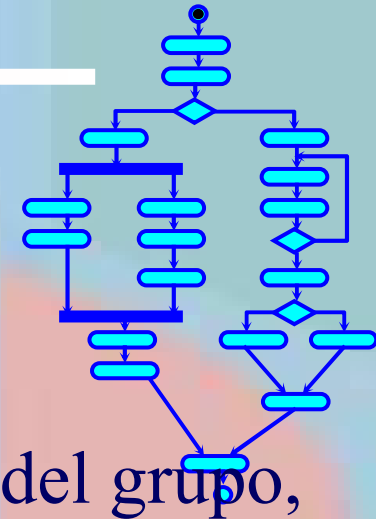
- Proceso
- Esqueleto global
- Organización del código
- Selección de herramientas
- Plan de capacitación

Lo que dicen los libros ... nuestra experiencia



Proceso de Desarrollo de Software:

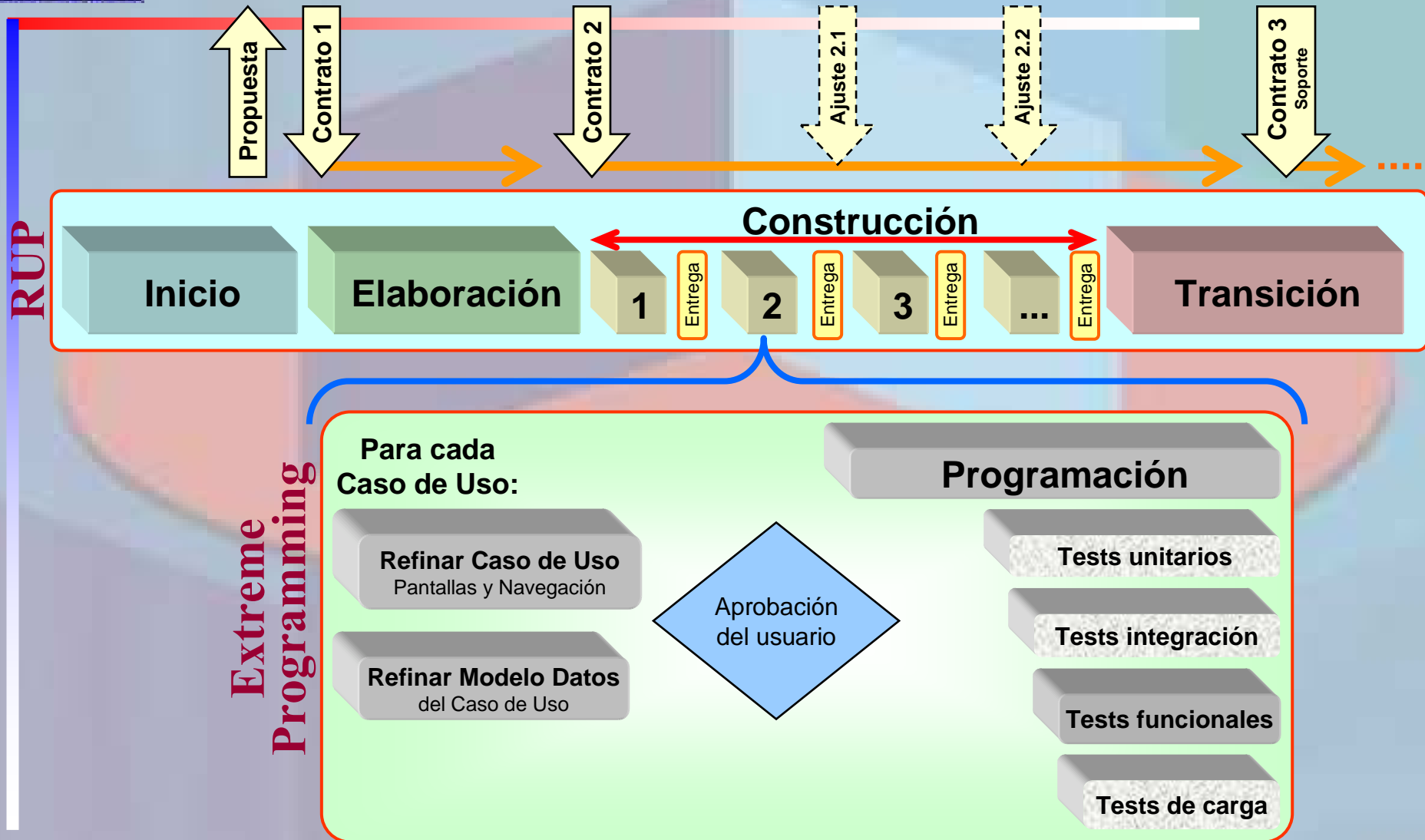
- secuencia de actividades,
- asignación de responsabilidades,
- dinámica de interacción entre los miembros del grupo,
- conjunto de *entregables* (documentos, código, pantallas, manuales, ...),
- conjunto de herramientas, adecuadamente sintonizadas e integradas.





Nuestra experiencia:

Proceso de Desarrollo de Software





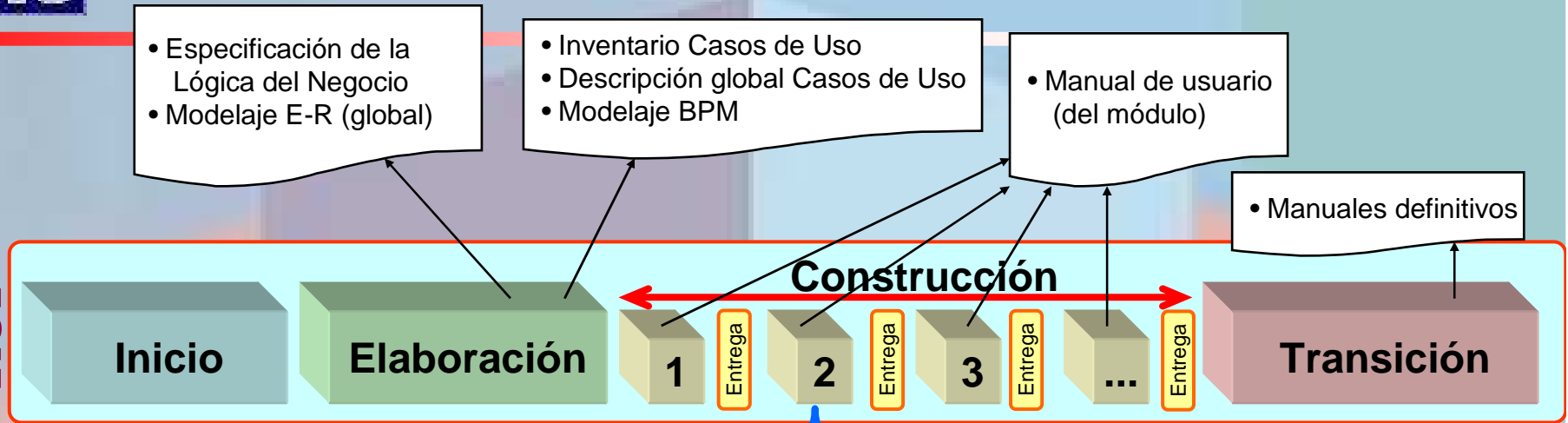
Estándarización de Entregables

- Documentos que se elaboran y entregan a lo largo del proyecto (especificación de lógica del negocio, diseño de casos de uso, diseño de B.D., registros de prueba, manuales, ...)
- Cada entregable:
 - ↓ tiene una **Tabla de Contenido** definida de antemano
 - ↓ se produce siguiendo cierta **metodología**
 - ↓ se desarrolla usando ciertas **herramientas**
 - ↓ se expresa en determinado **lenguaje**
- Con excepción de los “Manuales de Usuario”, la cantidad y nivel de detalle de los entregables debe mantenerse cercana al “mínimo necesario”.

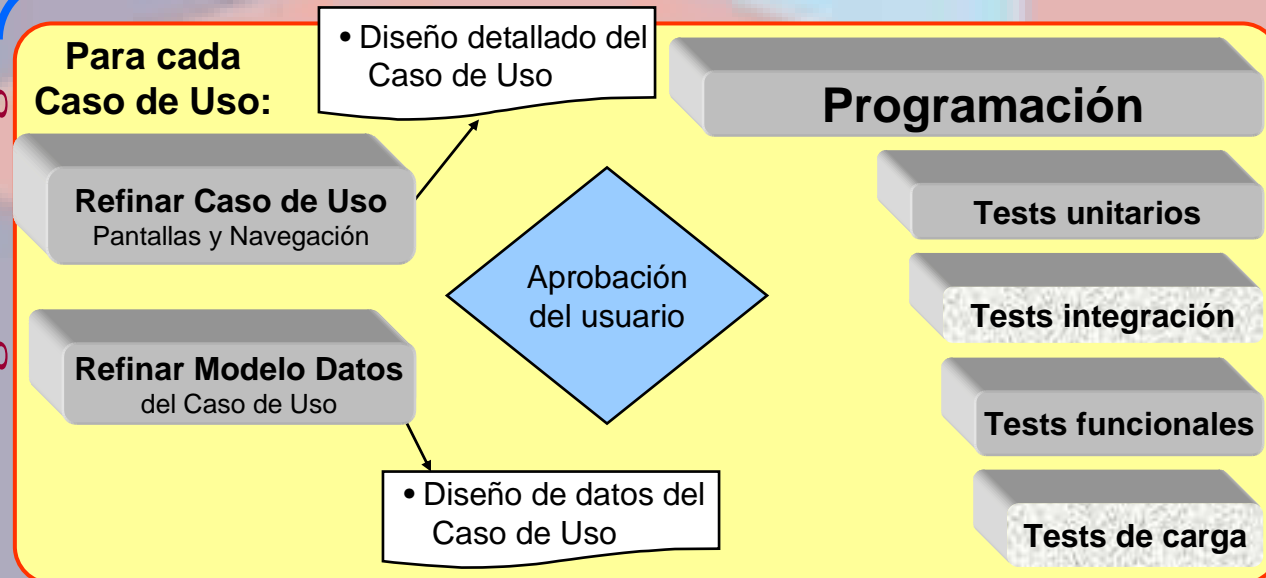


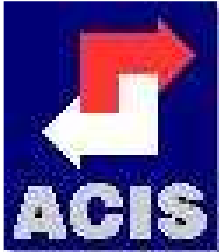
Estándarización de Entregables: Nuestra experiencia

RUP



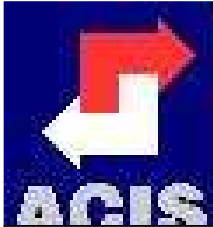
Extreme Programming





Estándar de organización y documentación del código

- Un buen sistema de software vive y evoluciona durante muchos años.
- La documentación inicial se desactualiza rápidamente.
- la única información estructural y semántica que se mantiene vigente ... es el código fuente.
- Hay que cuidar y mantener el código fuente como un activo muy valioso ... y volátil !



Estándar de organización y documentación del código: Nuestra experiencia

The screenshot shows a file explorer window with a project structure on the left and a list of files on the right. The project structure is as follows:

- SampleProject
 - .settings
 - ant
 - build
 - doc
 - bin
 - JavaSource
 - com
 - cincosoft
 - common
 - control
 - entities
 - test
 - utils
 - hotels
 - bookingList
 - hotelApproval
 - hotelBooking
 - hotelInscription
 - hotelSearching
 - security
 - changePassword
 - login
 - logout
 - register
 - lib
 - sql
 - Oracle10g
 - postgresql
 - WebContent
 - Meta-inf
 - pages
 - Web-inf

The file list on the right is as follows:

Name	Size	Modified
Atraccion.java	2KB	07/13/2006 11:43 AM
Booking.java	4KB	07/13/2006 11:43 AM
Hotel.java	4KB	07/13/2006 11:43 AM
Module.java	2KB	07/13/2006 11:45 AM
Profile.java	2KB	07/13/2006 11:45 AM
ProfileService.java	2KB	07/13/2006 11:45 AM
Service.java	2KB	07/13/2006 11:45 AM
Usecase.java	3KB	07/13/2006 11:46 AM
User.java	2KB	06/29/2006 4:34 PM
UserProfile.java	2KB	07/13/2006 11:47 AM

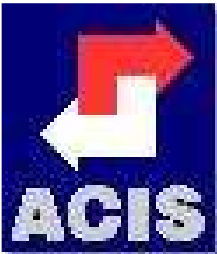
Estándar de organización:

- TODO el software bajo una misma jerarquía (código, scripts, descriptores, sql, librerías, ...)
- Organizado por Módulos
- Dentro de cada módulo, organizado por casos de Uso y Web-services
- Correspondencia de nombres (directorio = caso de uso = clase java, ...)
- Administrado por controlador de versiones (CVS ...).



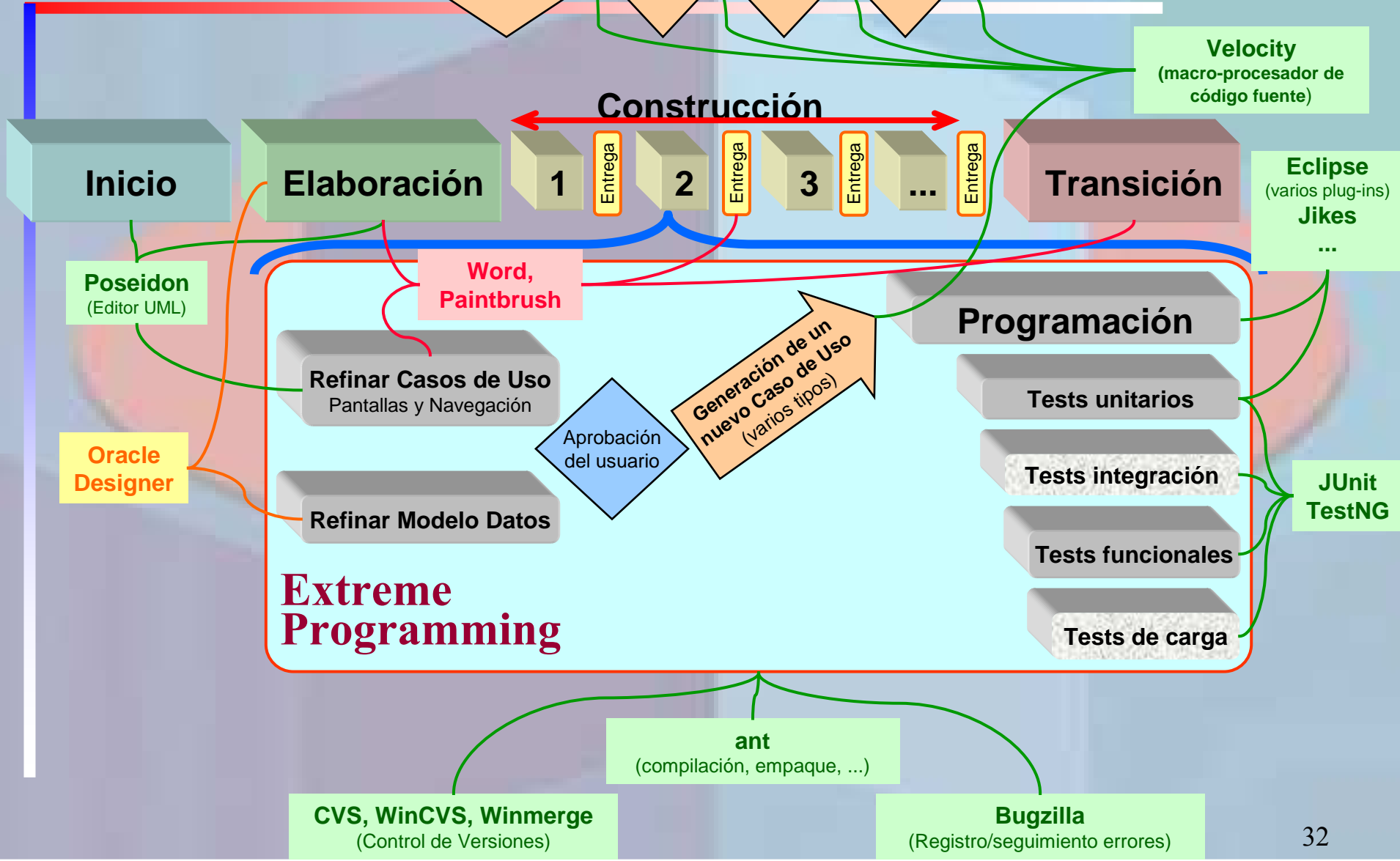
Selección e integración de herramientas

- Desarrollar en las nuevas tecnologías requiere usar muchas herramientas especializadas.
- Para plataformas abiertas (Java EE, LAMP) existe una cantidad ENORME de herramientas libres de alta calidad.
- En torno a cada herramienta libre hay una comunidad de soporte mutuo, que comparte información muy valiosa.
- Cada herramientas debe ser cuidadosa y uniformemente sintonizada, e integrada con las otras.



- Generación del Sistema inicial: Seguridad, Menus, Control, Librerías básicas
- Generación de un nuevo módulo
- Generación de una nueva Entidad
- Generación de un nuevo Webservice

Selección e integración de herramientas:
Nuestra experiencia





Nuestra experiencia:

Esqueleto global de un sistema (“Meta-Framework”)





Esqueleto global de un sistema: “Framework”

- “Esqueleto + librerías” para facilitar la construcción de un aspecto de un sistema de software.
- Solo cubren aspectos específicos del desarrollo:
 - ↓ Desarrollo de interfaz de usuario (JSF, Tapestry, Struts, ...)
 - ↓ Conversión Modelo de Objetos \boxtimes Modelo Base de Datos (EJB 3.0, Hibernate, ...)
 - ↓ Manejo de Componentes (EJB 3.0, Spring, ...)
 - ↓ Generación de reportes (Jasper, Datavision, ...), ...
- Es importante escoger cuáles se van a usar, y decidir:
 - ↓ Cómo integrarlos
 - ↓ Qué reglas deben seguir los componentes que se van a desarrollar para poder usar los frameworks.



Nuestra experiencia con un Meta-Framework [1]

- Se desarrolló un esqueleto completo de sistema Java EE 5.0, compilable y ejecutable desde el primer día.
- Organiza, integra, define la ubicación y los compromisos mínimos de todos los componentes de software que se vayan añadiendo al proyecto.
- Integra varios frameworks especializados (libres):
 - ↓ JavaServer Faces JSF 1.2: interfaces Web orientadas por eventos (Java EE 5.0).
 - ↓ Enterprise Javabeans EJB 3.0: componentes para implementar la lógica de negocio y la persistencia (parte de Java EE 5.0).
 - ↓ ... sigue ...



Nuestra experiencia con un Meta-Framework [2]

- ... Integra varios frameworks especializados (libres):
 - ↓ Business Process Management jBPM: manejo procesos de negocio y workflow.
 - ↓ Log4J: Manejo de logs.
 - ↓ Quartz: Scheduler.
 - ↓ Jasper: Facilidades básicas de generación de reportes.
 - ↓ Procesamiento de XML
 - ↓ Generación de PDF
 - ↓ Generación archivos Excel
 - ↓ ...



Esqueleto global de un sistema:

Nuestra experiencia con un Meta-Framework [3]

- Establece guías para la programación de los componentes propios de la aplicación, de manera que:
 - ↓ se ubiquen en el sitio correcto
 - ↓ se integren adecuadamente con los otros componentes
 - ↓ aprovechen las facilidades de los frameworks especializados.



Esqueleto global de un sistema:

Nuestra experiencia con un Meta-Framework [4]

- Se desarrollaron varios generadores de código que aceleran la construcción del nuevo sistema mediante:
 - ↓ Generación de todo el código de integración que requiere Java EE,
 - ↓ Generación de esqueletos -ya integrados- de los componentes que se vayan agregando al proyecto, con los *placeholders* a cargo del programador claramente señalados:
 - ↙ Un nuevo módulo
 - ↙ Un nuevo Caso de Uso
 - ↙ Un nuevo Web-Service
 - ↙ Una nueva Entidad
 - ↙ ...



Conclusiones



Conclusiones [1]

- Los elementos que “de verdad” hay que dominar/construir son:
 - ↓ Proceso de desarrollo
 - ↙ Entregables
 - ↙ Ciclo central
 - ↓ Estándar de organización del código
 - ↓ Selección e integración de herramientas
 - ↓ Plan de capacitación de los miembros del grupo
 - ↓ Esqueleto global (“Meta-framework”)
- Construir/dominar estos elementos requiere mucho tiempo y esfuerzo.
- En nuestro caso, para Java EE 5.0, hemos requerido 15 Meses*Ingeniero netos (entre 3 ingenieros con bastante experiencia)

- Disponer de un “Meta-framework” nos ha permitido simplificar y acelerar enormemente el desarrollo.
- Un nuevo programador es productivo en 10 días, pues solo deben aprender:
 - ↓ las herramientas comunes
 - ↓ La dinámica del ciclo central (diseño detallado, programación, pruebas, ajustes al diseño, ajustes a la programación, pruebas, ajustes ...)
 - ↓ “su parte” de la tecnología
 - ↙ Desarrollo de interfases (html, JSF)
 - ↙ Desarrollo de funcionalidades (Java)
 - ↙ Desarrollo de base de datos (SQL, jdbc), ...



Preguntas

Comentarios

Otros puntos de vista