

# ARQUITECTURAS

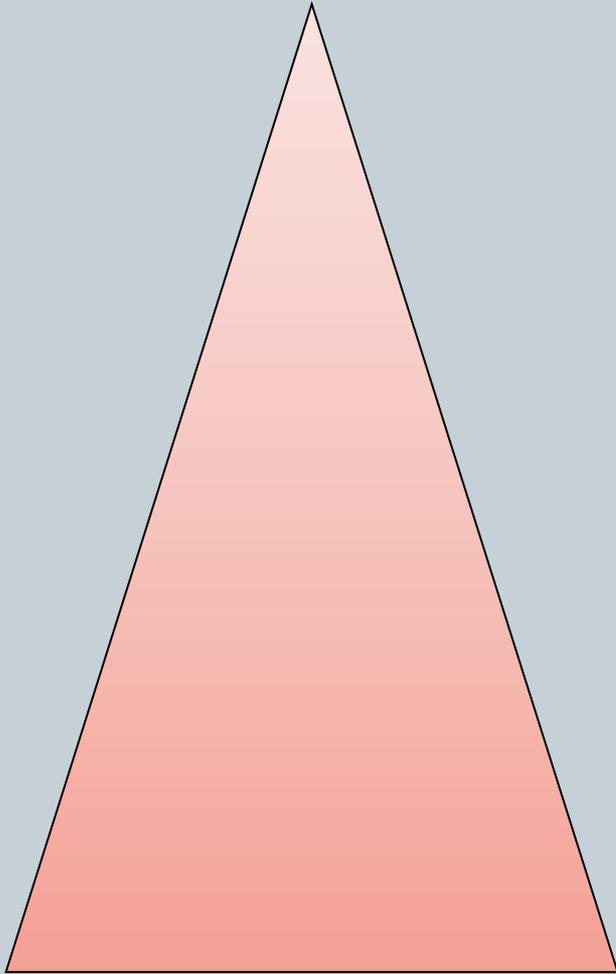
1

IN73J – Arquitectura, Diseño y  
Construcción de un Negocio con  
Apoyo TI

Carlos Reveco D.  
[creveco@dcc.uchile.cl](mailto:creveco@dcc.uchile.cl)

# Arquitectura de una aplicación

2



- **Arquitectura:** desarrolla un plan general del sistema, asegurando que las necesidades de los usuarios sean atendidas.
- **Ingeniería:** proyecta la estructura física interna, dando forma a los objetivos definidos por la arquitectura; considerando la eficiencia y la eficacia del proyecto.
- **Construcción:** elabora la estructura, con el uso de herramientas y datos

# Arquitectura de Software

3

- En los inicios de la informática, la programación se consideraba un arte y se desarrollaba como tal.
- Con el tiempo se han ido descubriendo y desarrollando formas y guías generales. A estas, se les ha denominado **Arquitectura de Software**, porque, semejanza de los planos de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software.
- Arquitectura es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema".

# Características Arquitectura de Software

- La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.
- Arquitectura Lógica
- establece los fundamentos para que analistas, diseñadores, programadores
- tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Sirve para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad
- La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea, sus interfaces y la comunicación entre ellos. Esta debe ser implementada en una arquitectura Física.

# Tipos de Arquitectura

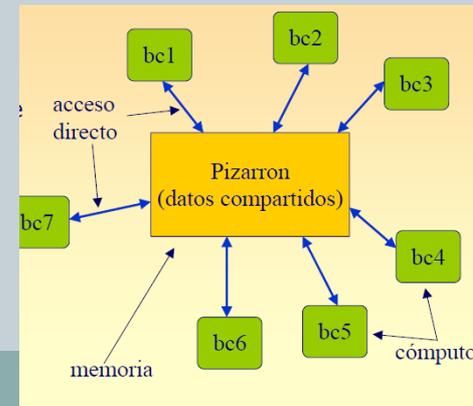
5

- **Monolítica:** Donde el software se estructura en grupos funcionales muy acoplados.
- **Cliente-servidor:** el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- **Arquitectura de tres niveles:** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

# Tipos de arquitectura

6

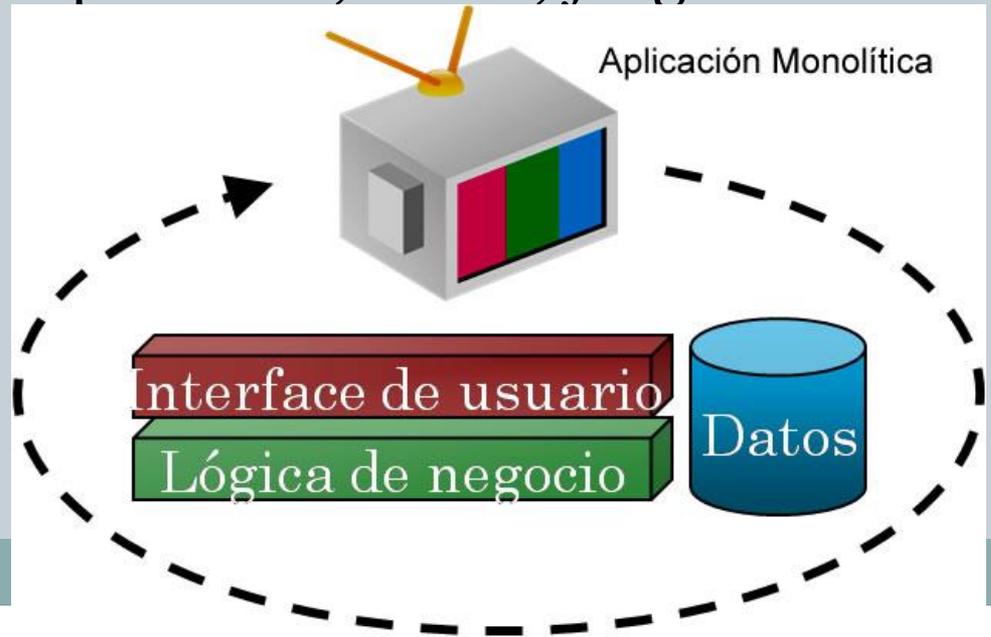
- La **Arquitectura Orientada a Servicios (SOA)** es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización
- Pizarrón
- Repositorio
- otros



# Arquitectura Monolítica

7

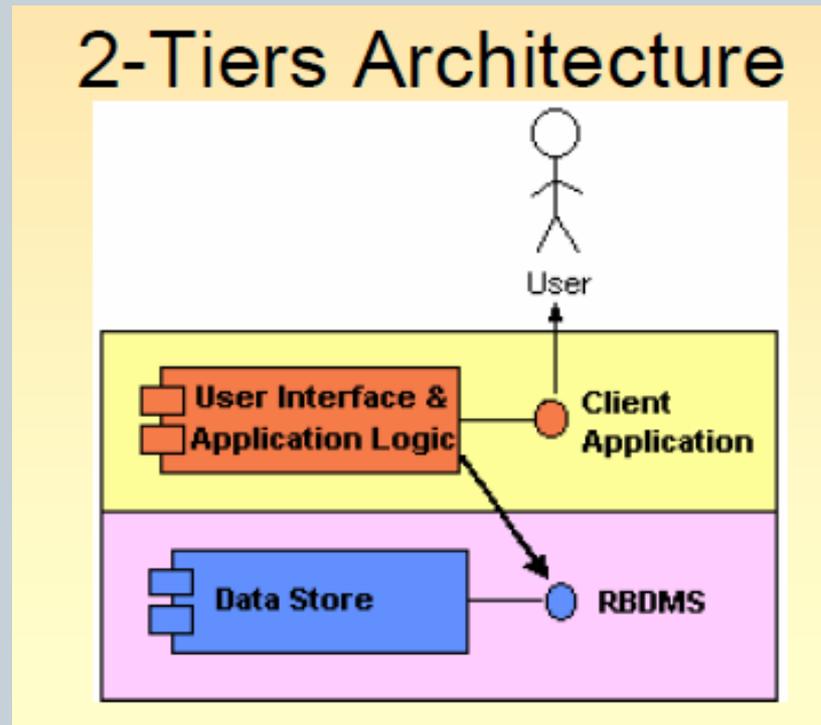
- Son aquellas en las que el software se estructura en grupos funcionales muy acoplados, involucrando los aspectos referidos a la presentación, procesamiento y almacenamiento de la información.
- En este rubro están considerados las distintas aplicaciones para escritorio: sistemas operativos, Office, juegos monousuario, etc.



# Arquitectura Cliente/Servidor

8

- Arquitectura Cliente/Servidor en **dos capas**:
  - ✦ **Front/end**
  - ✦ **Back/end**



# Modelo de Arquitectura Cliente/Servidor

9

- **Front/end**
  - Es la parte de la aplicación que **interactúa** con el usuario.
  - Basados en una interfaz gráfica con el usuario (GUI). El Cliente corre la aplicación que ofrece la interfaz con el usuario.
- **Back/end**
  - Es la parte no-interactiva de la aplicación. La mayor parte reside en las Bases de Datos (relacionales o no).

# Modelo de Arquitectura Cliente/Servidor

10

- **Aplicaciones Simples:** no requieren una gran **Base de Datos** compartida, pueden ser elaboradas solamente en el **C**liente.
- **Aplicaciones Complejas:** exigen dos capas, una para la aplicación del usuario (**C**liente) y otra para la base de datos (**S**ervidor).  
Eventualmente, el **C**liente y el **S**ervidor podrán estar en el mismo equipamiento.

# Procesos de Negocios

11

- Las organizaciones, independientemente de su tamaño, utilizan sus recursos y su tiempo en Procesos de Negocios complejos.
- Un Proceso de Negocio consiste en reglas y en la ejecución de un conjunto de pasos.
- Pensar los procesos en términos de reglas permite que la computadora pueda automatizar las parte mecánicas.

# Conductor de los Procesos de Negocios

12

- La actividad principal de los **Sistemas de Información Computarizados**, es la de:
  - **Conducir** a los **Procesos de Negocios**.
  - **Mantener** a la **BD**, Poner a **disposición** de toda la empresa los datos que precisan los usuarios y Coordinar el **acceso** a los datos.
- Lo ideal es que los **Procesos de Negocio** estén **automatizados**

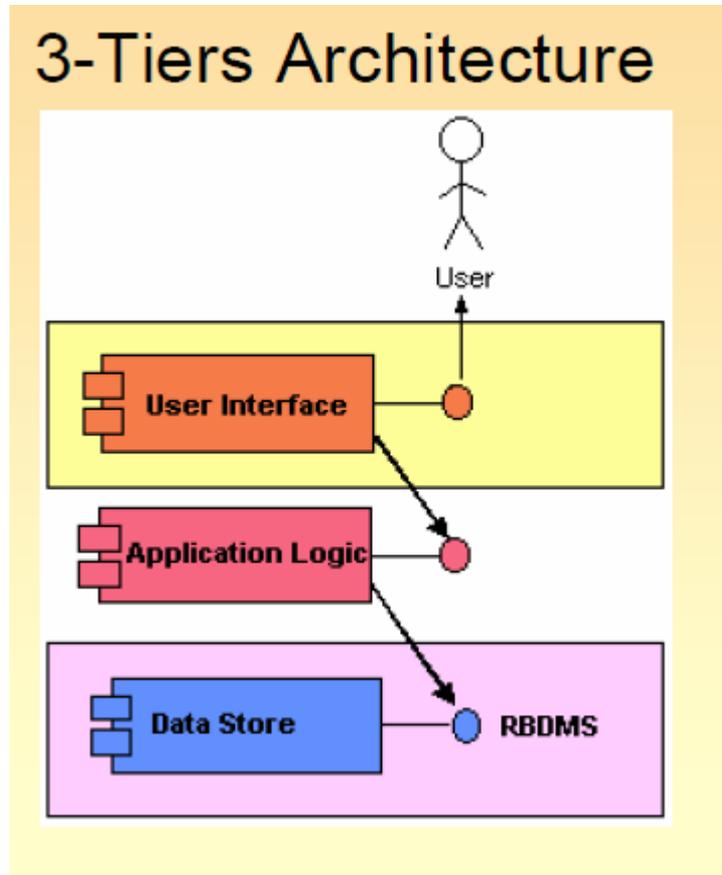
# Arquitectura 3 capas Cliente / Servidor

13

<b>Nivel</b>	<b>Contenido</b>
Aplicativo del Usuario	Aplicaciones de PC e interfaces gráficas
Reglas del negocio	Reglas del negocio y procesos de cálculo
Administración de datos	Base de Datos (relacionales y SQL)

# Arquitectura Cliente / Servidor

14



# Funciones de los Niveles en la Arquitectura



Nivel	Responsabilidad	Funciones	Herramientas
<b>Aplicativo del Usuario</b>	Interfaz comprensible y eficiente	Presentación, navegación, manejo y análisis	Herramientas gráficas y lenguaje de programación
<b>Regla de Negocio</b>	Política: reglas y heurística	Toma de decisiones, políticas, administración de recursos	Lenguaje de programación
<b>Base de Datos</b>	Datos consistentes y seguros	Mantenimiento, actualización, integridad, seguridad	Base de Datos, lenguaje de BD

# Arquitectura de tres Niveles

16

- La Arquitectura de tres niveles **es lógica** y no física. Se preocupa con las funciones y no la implantación.
- La Arquitectura puede ser utilizada para desarrollar **sistemas Centralizados o Distribuidos**.
- La Arquitectura **facilitará la distribución** de los componentes del sistema.

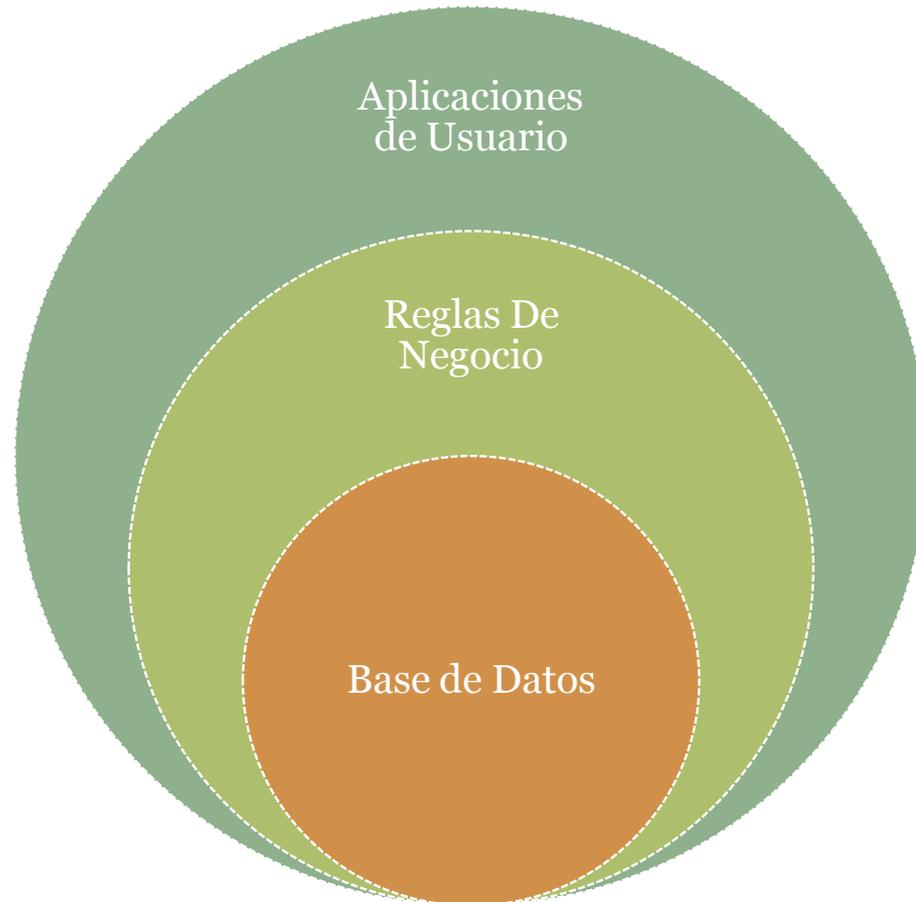
# Beneficios

17

- Estructura para la elaboración de aplicativos **flexibles y fáciles de modificar**, según las necesidades del negocio(cambio).
- Alto nivel de **reutilización del software y datos**.
- Fácil y rápido desarrollo de sistemas distribuidos que dan soporte a la administración central y a equipos autogestionados

# Niveles de abstracción

18



# Beneficios de la abstracción

19

- Esconde de cada nivel los detalles contenidos en los niveles inferiores:
  - Desarrollo simplificado del aplicativo:
    - ✦ al desarrollar un nivel no precisa preocuparse por el otro
  - Mayor seguridad y protección:
    - ✦ el desarrollador de un nivel no puede controlar físicamente otra tipo en ningún nivel de detalle

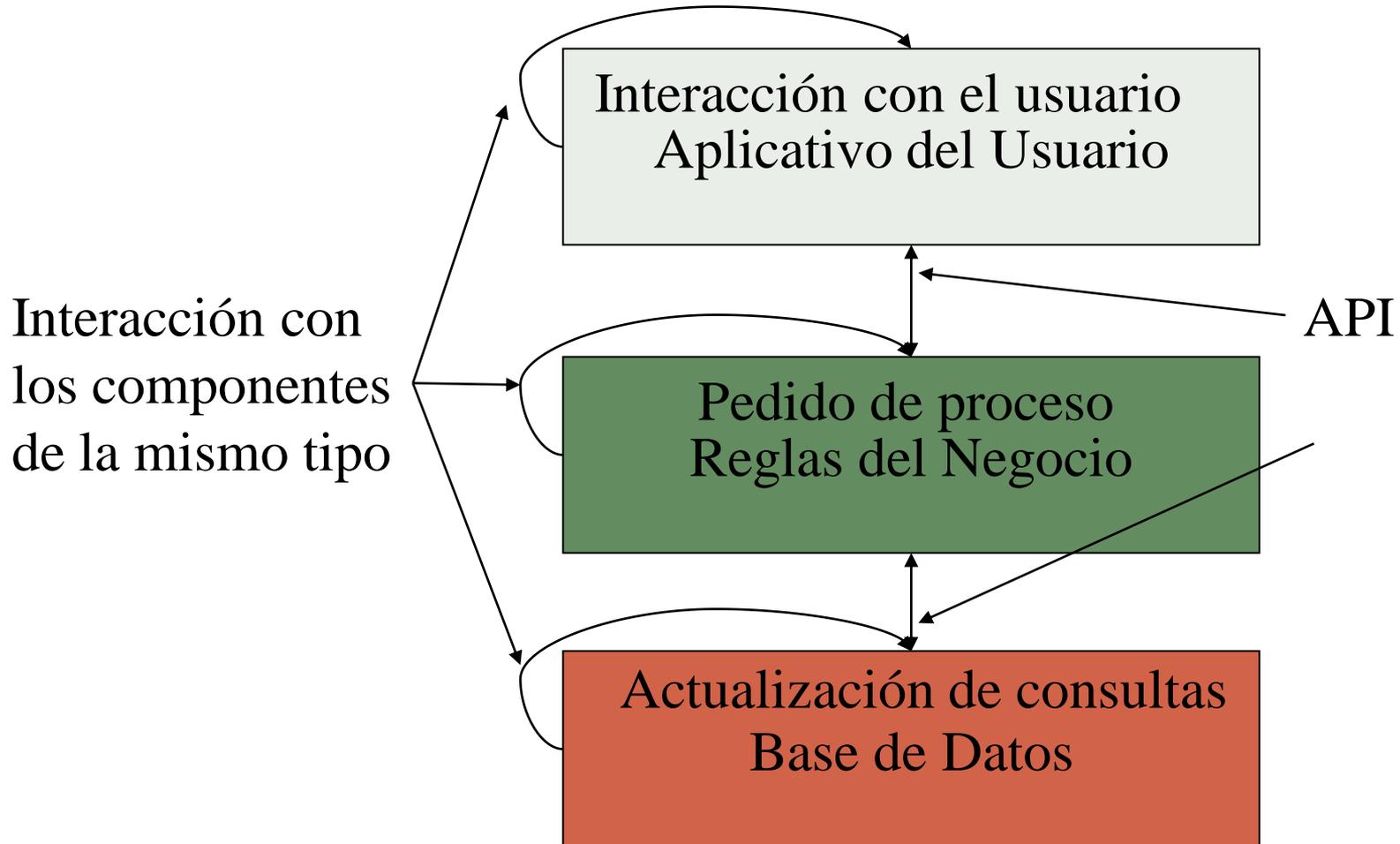
# interfaz de programación de aplicaciones (API)

20

- Una **interfaz de programación de aplicaciones** o **API** (del inglés *application programming interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Usados generalmente en las bibliotecas. interactúan a través de las siguientes funciones:
  - Informa lo que el otro componente debe hacer, pregunta el estado actual del otro componente, recibe el resultado de las operaciones solicitadas.

# Tipos de interfaces

21



# Interfaz Base de Datos

22

- **Funciones de las transacciones:**
  - Ejecutar actualizaciones consistentes con los datos.
  - Imponer reglas básicas del negocio.
  - Evitar cambios no autorizados o inválidos
  
- **Funciones de las consultas:**
  - Facilitar consultas complejas
  - Asegurar consistencia
  - Garantizar seguridad

# Base de Datos independientes de las Reglas del Negocio

23

- Projete la base de datos usando un modelo de planeamiento.
- Desarrolle consultas y transacciones que ofrezcan un buen acceso a la base de datos.
- Solo permita que las transacciones bien elaboradas actualicen la base de datos.
- Aisle a los usuarios de los detalles y de la localización de las base de datos fundamentales.

# Interfaz de las reglas del proceso del negocio

24

- **Inter-operabilidad:**
  - Capacidad de compartir trabajo, compartir software y hacer cosas consistente para toda la organización.
- **Reutilización:**
  - Uno de los principales motivos que imposibilita la reutilización de códigos es la falta de separación entre la interfaz del usuario, las reglas del negocio, y la administración de la base de datos en un mismo programa.

# Flexibilidad de las Reglas del Negocio

25

- Un único módulo de software trata de una tarea específica.
- Este módulo es independiente de la BD y de la interfaz del Usuario.
- Todas las aplicaciones de la organización utilizan este módulo para realizar esa misma tarea.

# Interfaz de la Aplicación del Usuario

26

- Libertad para los Usuarios:
  - Modelan el sistema según sus necesidades sin afectar las reglas del negocio y la base de datos.
- Libertad para la Organización:
  - La aplicación del usuario envía solicitudes de procesos formales para ejecutar las reglas del negocio, transacciones y consultas en la Base de Datos.
  - Las Reglas del Negocio y la Base de Datos quedan aislados de los cambios en la interfaz de los Usuarios.

# Interfaz Grafica de Usuario

27



The image shows the classic 1997 Google logo, where each letter is a different color (blue, red, yellow, blue, green, red) and has a 3D effect with a shadow.

Search the web using Google!

10 results ▾

Google Search

I'm feeling lucky

*Index contains ~25 million pages (soon to be much bigger)*

[About Google!](#)

[Stanford Search](#) [Linux Search](#)

Get Google! updates monthly!

your e-mail

Subscribe

[Archive](#)

Copyright ©1997-8 Stanford University

Google 1997



The image shows the Google logo for the Spanish market, with the word "Google" in its multi-colored font and "España" in black below it.

Buscar con Google

Voy a tener suerte

¡Nuevo! [Viaja por el mundo](#) desde tu página de inicio de Google

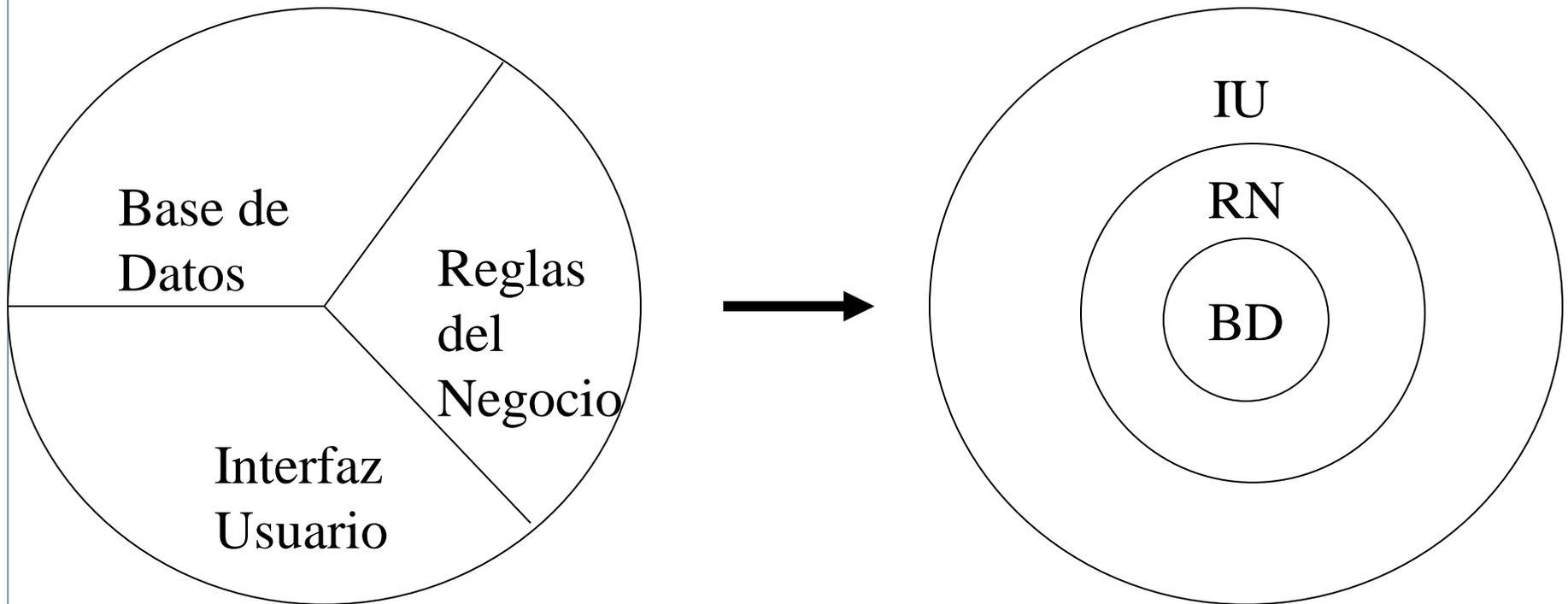
Google.es ofrecido en: [català](#) [galego](#) [euskara](#)  
[Programas de publicidad](#) [Soluciones Empresariales](#) [Todo acerca de Google](#) [Google.com in English](#)

© 2010 - Privacidad

Google 2010

# Flexibilidad=Interoperabilidad + reutilizabilidad

28



Independencia y modularidad de los procesos  
Reutilización del código

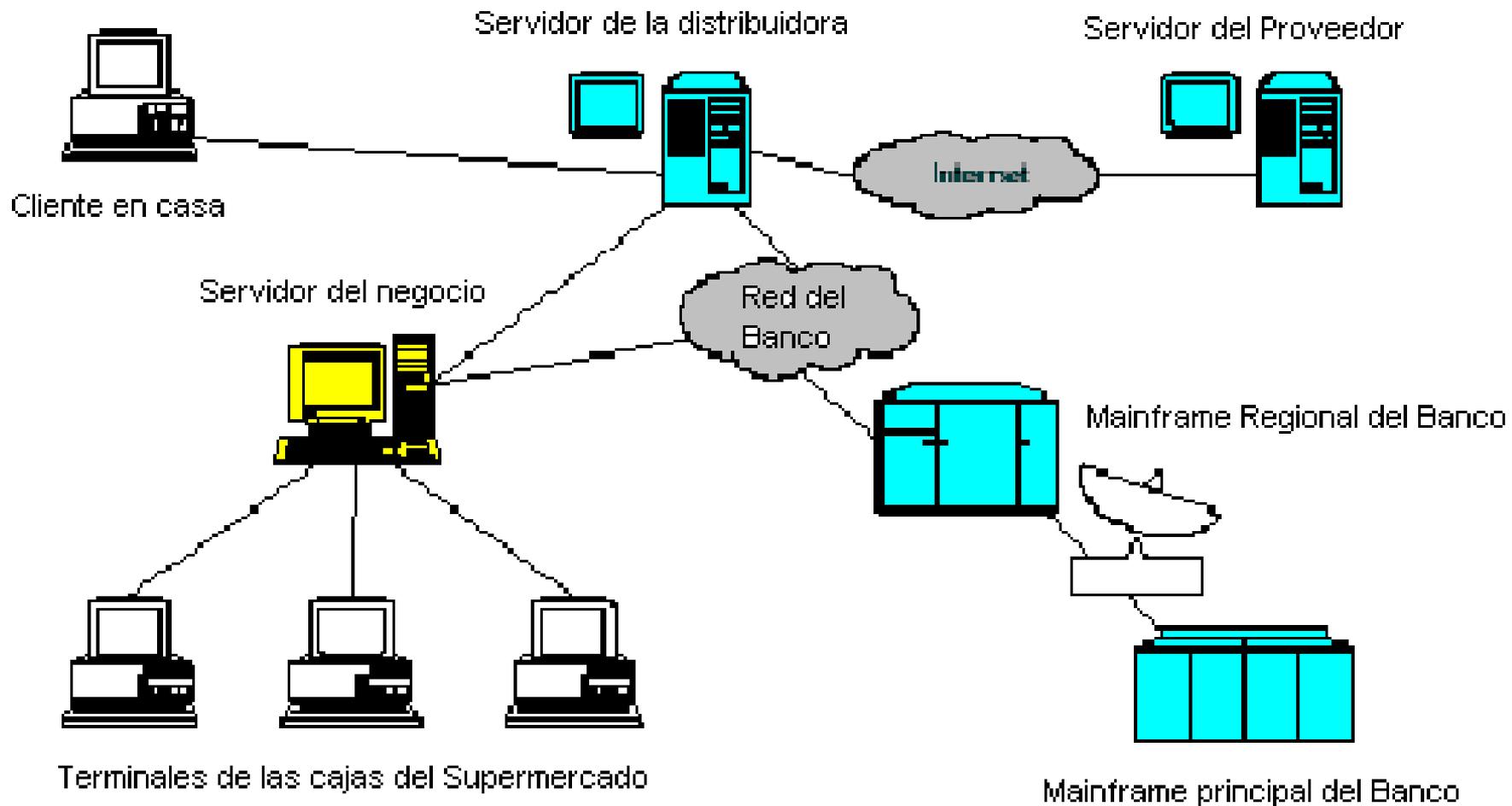
# Es posible desarrollar un Sistema Distribuido

29

- Distribución de Datos
- Distribución de Procesamiento
- Interfaz gráfica con el usuario

# Existen los Sistemas Distribuidos

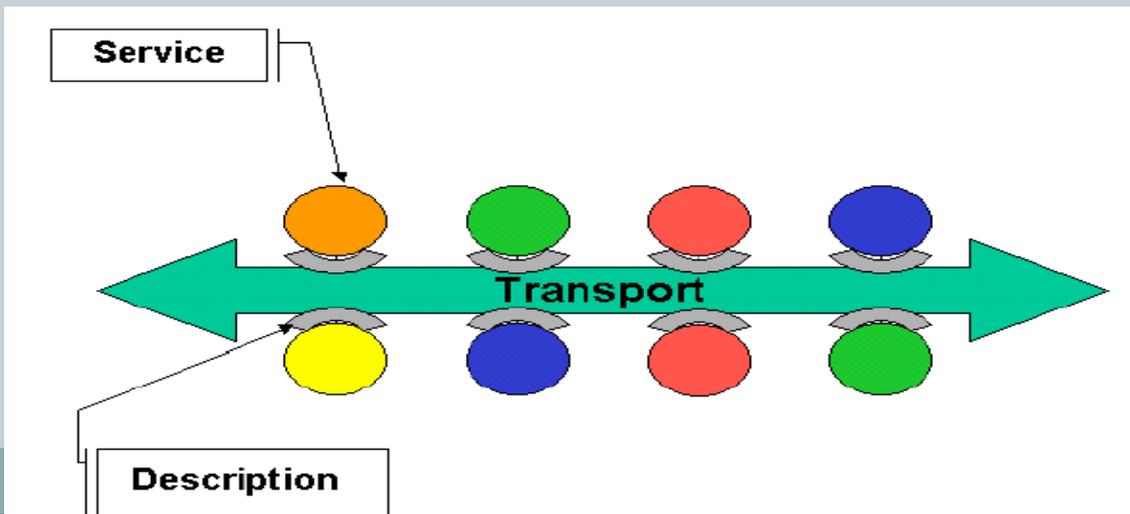
30



# Aquitectura SOA

31

- La arquitectura orientada a servicios es tanto un marco de trabajo para el desarrollo de software como un marco de trabajo de implementación. El Desarrollo se basa en crear servicios comunes que son orquestados por clientes o middleware para implementar los procesos de negocio.



# ARQUITECTURAS

32

IN73J – Arquitectura, Diseño y  
Construcción de un Negocio con  
Apoyo TI

Carlos Reveco D.

[creveco@dcc.uchile.cl](mailto:creveco@dcc.uchile.cl)

**Magíster**

en Ingeniería de Negocios con TI

TECNOLOGÍA AL SERVICIO DE LA GESTIÓN



**INGENIERIA INDUSTRIAL**  
UNIVERSIDAD DE CHILE