



Escuela de Sistemas y Tecnologías

Transparencias de ANALISTA DE SISTEMAS
Edición 2017 – Materia: Diseño de Aplicaciones Web

TEMA: Arquitectura

Plantel y Contactos

➤ **Bedelía:**

- Mail: bedeliasistemas@bios.edu.uy

➤ **Encargado de Sucursal:**

- Pablo Castaño
- Mail: pablocasta@bios.edu.uy

Recursos

➤ Recursos Imprescindibles:

- Sitio Web de material
(comunicarse con Bedelía por usuario/contraseña).
- Transparencias del Curso.
- Contar con el software necesario

Agenda

- ☐ UML
- ☐ Arquitectura
- ☐ Diagramas Componentes

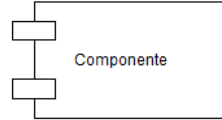
UML

Lenguaje UML

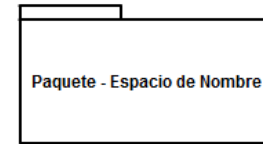
- Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema
- UML ofrece un estándar para describir un "modelo" del sistema.
- UML es un "lenguaje" para especificar y no para describir métodos o procesos.
- Algunos diagramas de Software
 - Diagramas de Clases.
 - Diagramas de Componentes.
 - Diagramas de Paquetes

Notación UML

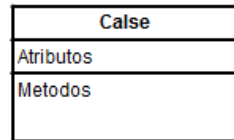
- Componente



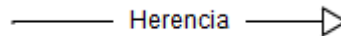
- Paquete – Espacio de Nombre



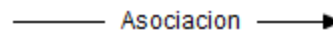
- Clase



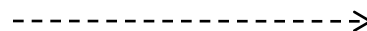
- Herencia



- Asociación



- Usa A



Arquitectura

Introducción

- Es el diseño de más alto nivel de la estructura de un sistema.
- Consiste en un conjunto de técnicas que proporcionan el marco de referencia necesario para guiar la construcción del Software para un sistema de información.
- Establece los fundamentos para que analistas, diseñadores y programadores, trabajen en una línea común que permita alcanzar los objetivos del sistema de información.

Modelos o Vistas

- La visión **estática**: Esta vista describe todos los componentes de la arquitectura.
- La visión **funcional**: Esta vista describe que hace cada uno de los componentes.
- La visión **dinámica**: Esta vista describe como se comportan los componentes a lo largo del tiempo y como interactúan entre sí.

Tipos (1)

- **Monolítica:**
El software se estructura en un mismo componente muy acoplado.
- **Cliente-Servidor:**
El software es repartido entre un cliente y un servidor y el procesamiento de los datos se lleva a cabo en ambos lugares.

Tipos (2)

- **En Capas:**
 - El software se compone en capas, donde cada capa tiene una responsabilidad. La arquitectura mas conocida es la de 3 capas.
 - **Capa de presentación:**

Capa donde se llevara a cabo la programación de la interfaz de usuario.
 - **Capa de Lógica de negocios:**

Esta capa es la responsable de la ejecución de las reglas de negocio.
 - **Capa de Acceso a Datos:**

Esta capa se responsabiliza de manejar los accesos a los datos

Tipos (3)

- **Orientada a Servicios – SOA:**

Es un concepto de arquitectura de software donde se define la utilización de servicios para dar soporte a los procesos de negocio.

 - Diferencias con otras arquitecturas
 - Altamente acoplada
 - Interoperable con otros sistemas.
 - Beneficios
 - Agiliza los cambios en los procesos.
 - Facilidad para la integración con otras tecnologías.

Ejemplos de Distribución

Los siguientes 4 ejemplos muestran las formas básicas en que se puede distribuir la arquitectura de un sistema.

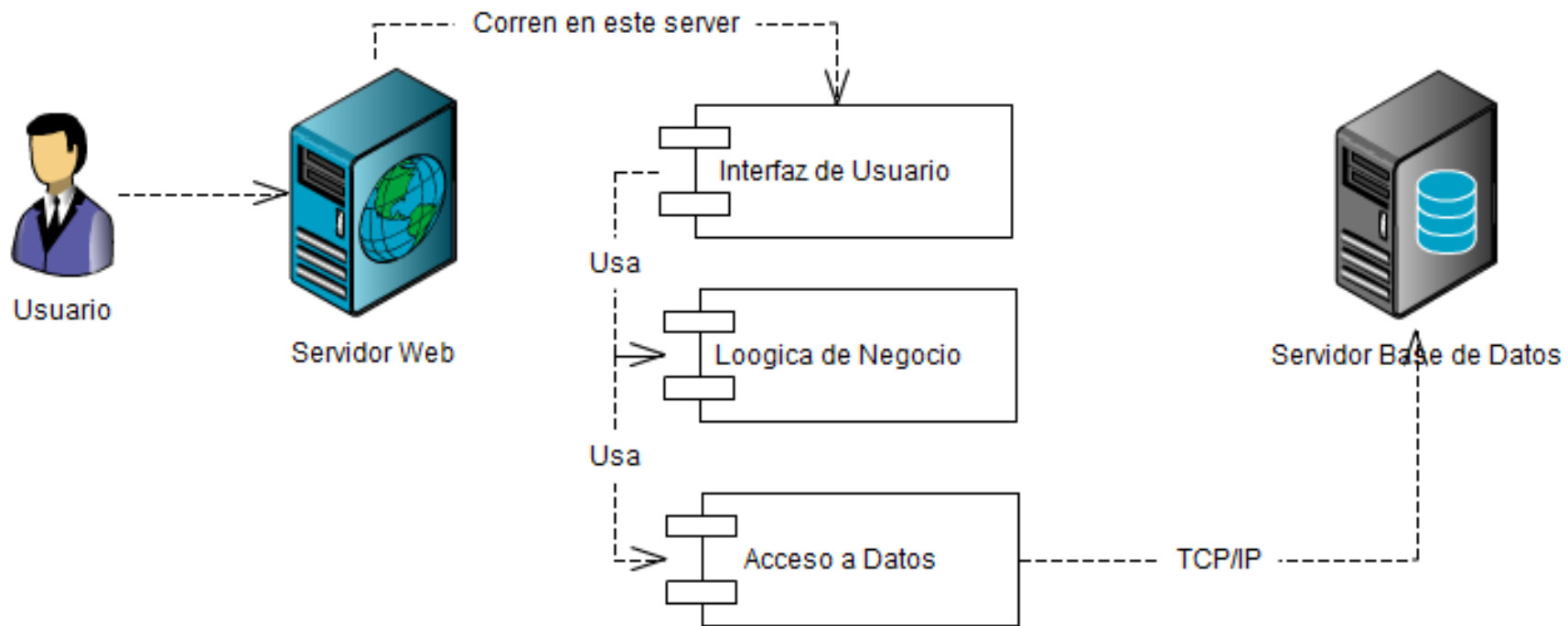
Tomar en cuenta que solo se basan en el uso de sitios web, no se contempla la posibilidad de la existencia de aplicaciones escritorio.



Único Servidor (2)

- Se utiliza arquitectura en 3 capas
- El usuario se comunica con el sitio publicado en el Servidor Web a través de internet
- Todos los componentes están alojados en el Servidor Web
- Motor de base de datos dentro del Servidor Web

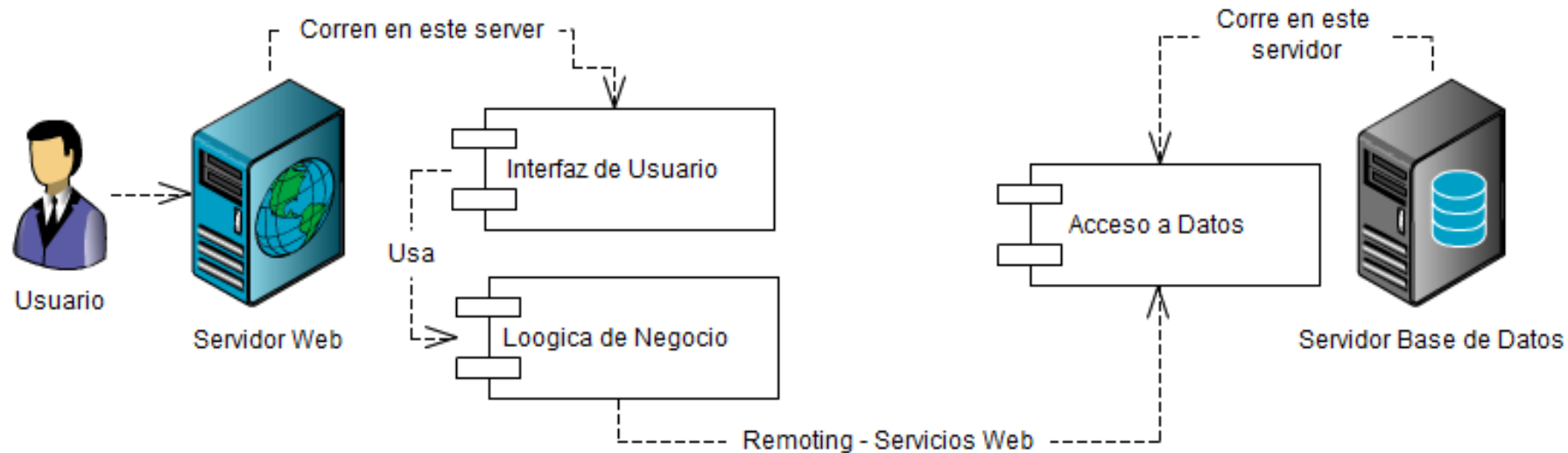
Dos Servidores (1)



Dos Servidores (1)

- Se utiliza arquitectura en 3 capas
- El usuario se comunica con el sitio publicado en el Servidor Web a través de internet
- Todos los componentes están alojados en el Servidor Web
- Motor de base de datos alojado en Servidor de Datos
- Componente Acceso a datos se comunica con el Servido de datos a través de ADO.NET.
- La carga esta distribuida entre ambos servidores

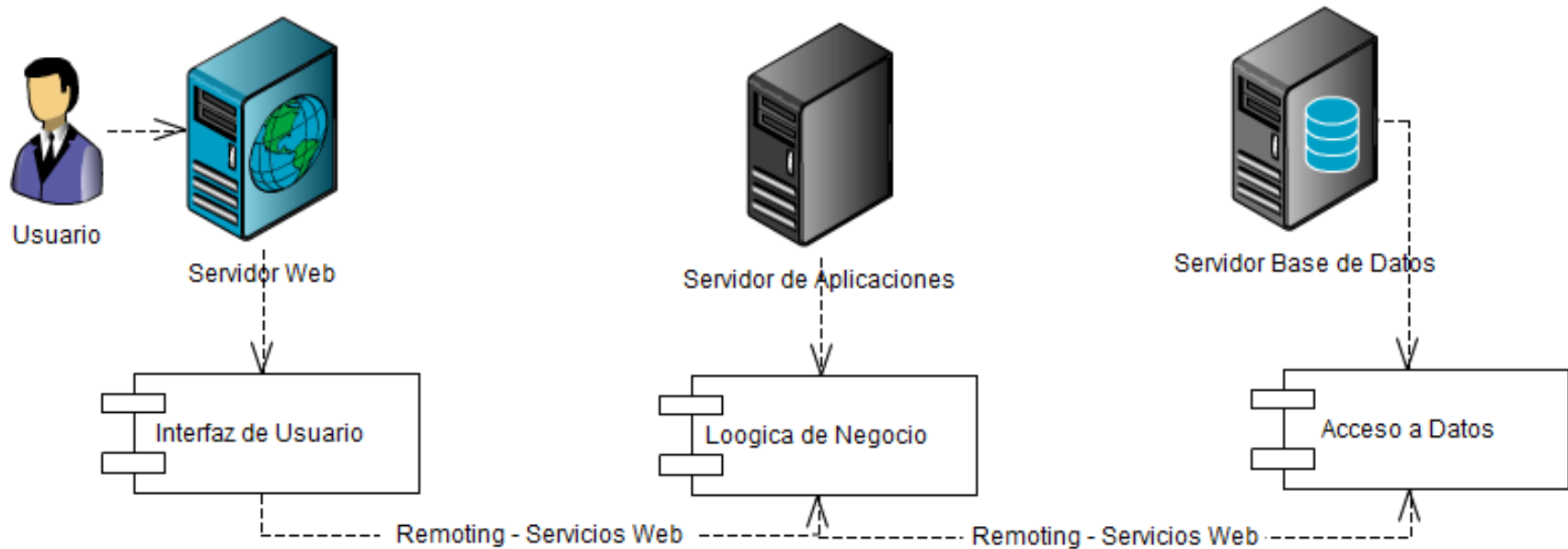
Servidores específicos(1)



Servidores específicos(2)

- Se utiliza arquitectura en 3 capas
- El usuario se comunica con el sitio publicado en el Servidor Web a través de internet
- El componente de Lógica de negocio se aloja en el Servidor Web, mientras que el componente de Acceso a datos se aloja en el mismo servidor que el Motor de base de datos (Servidor de Datos)
- El Servidor Web se centraliza en procesar las reglas de negocio e interactuar con el usuario
- El componente de Lógica de Negocio utiliza Remoting o Servicios Web para acceder a los datos
- La carga esta distribuida entre ambos servidores

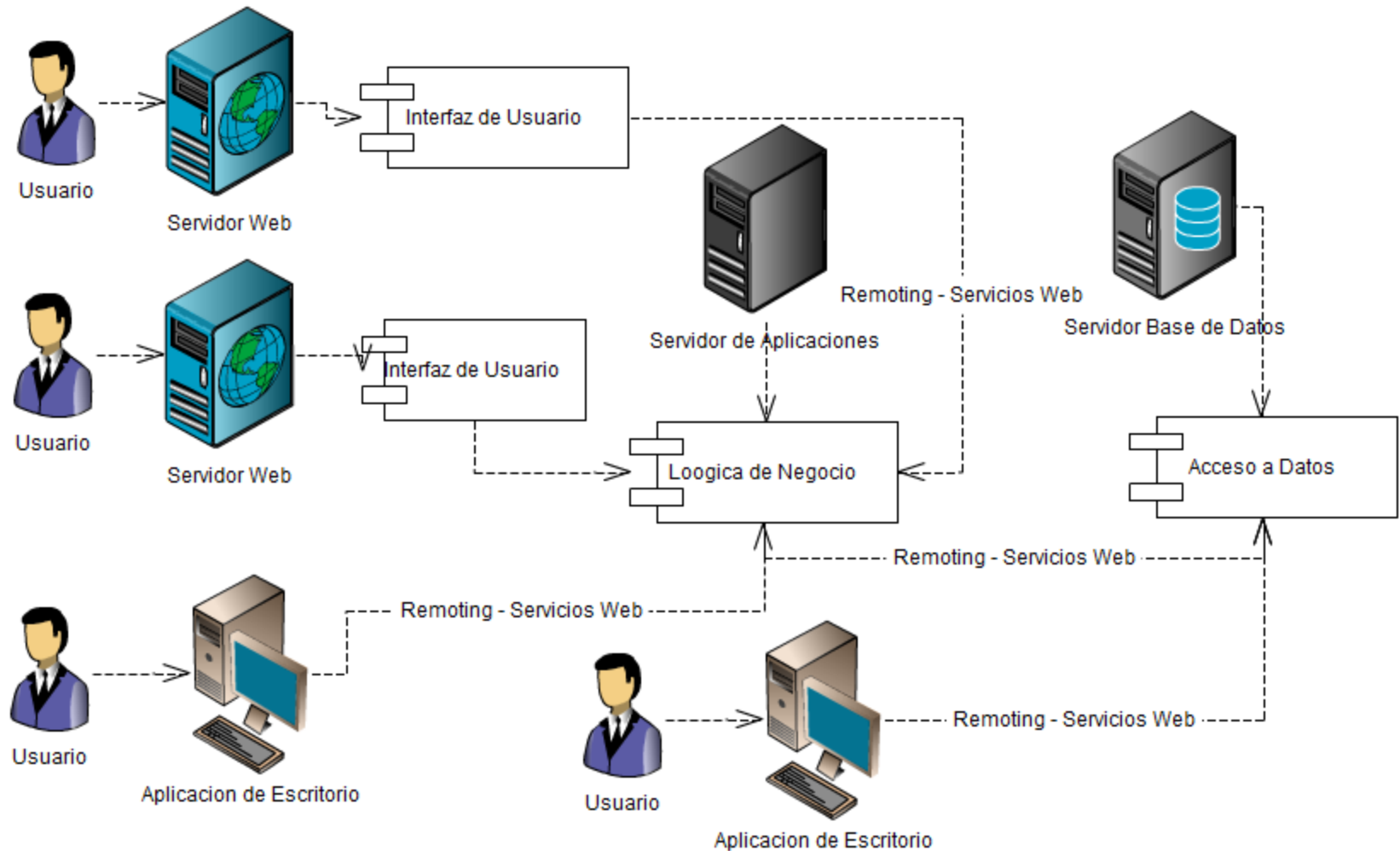
Distribuir Todo (1)



Distribuir Todo (2)

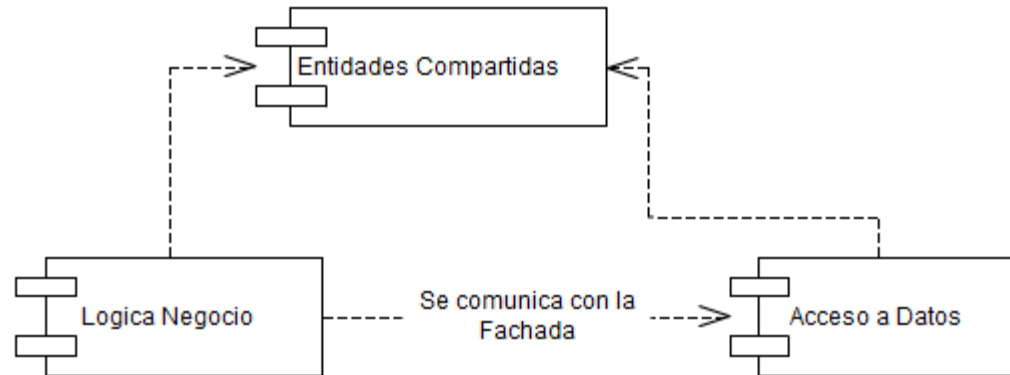
- Componentes distribuidos en 3 servidores especializados para cada función.
- Comunicación entre componentes mediante Remoting o Servicios Web
- La carga esta distribuida en 3 servidores.
- El servidor Web interactúa con el usuario.
- El servidor de aplicaciones solo procesa reglas de negocio
- El servidor de acceso a datos se encarga de manejar la base de datos.

Ejemplo Distribución Completo



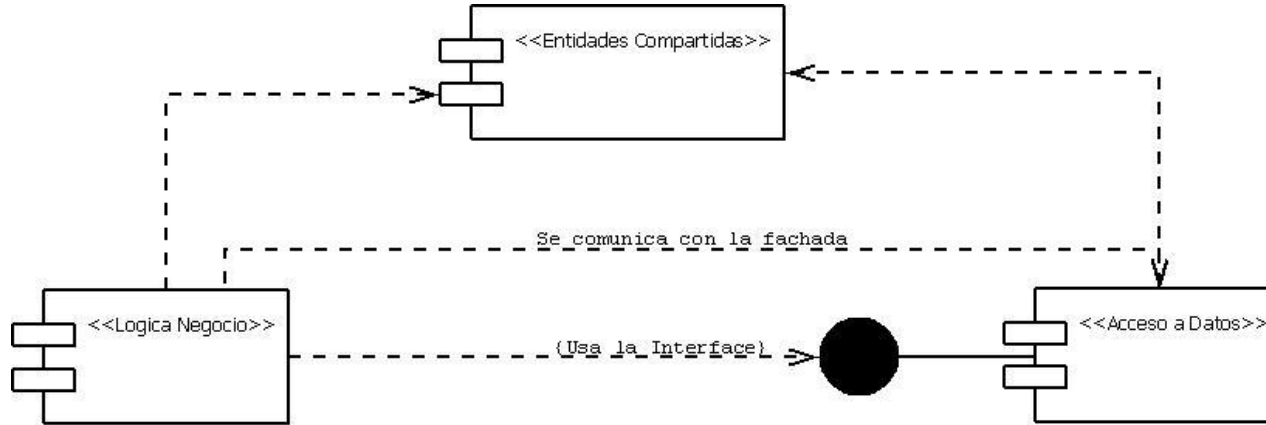
Diagramas Componentes

Diagrama de Componentes (1)



- Lógica de negocio dialoga con Acceso a datos mediante su fachada.
- Dicha fachada ejecuta la operación y retorna el resultado.
- La información se comunica mediante objetos de negocios definidos en Entidades Compartidas

Diagrama de Componentes (2)



- Lógica de negocio dialoga con Acceso a datos mediante su fachada (basada en Patrón Factory).
- La fachada devuelve la instancia en el nombre de la interface para ser usada.
- La Lógica de negocio ejecuta las reglas del negocio utilizando la interfaz directamente
- La información se comunica mediante objetos de negocios definidos en Entidades Compartidas