

Proyecto de grado

SIREI

SISTEMA DE INFORMACIÓN PARA RESULTADOS DE EXÁMENES IMAGENOLÓGICOS

Autores

Rafael Mártony
María Noel Tamayo



Tutor

Ing. Raúl Ruggia

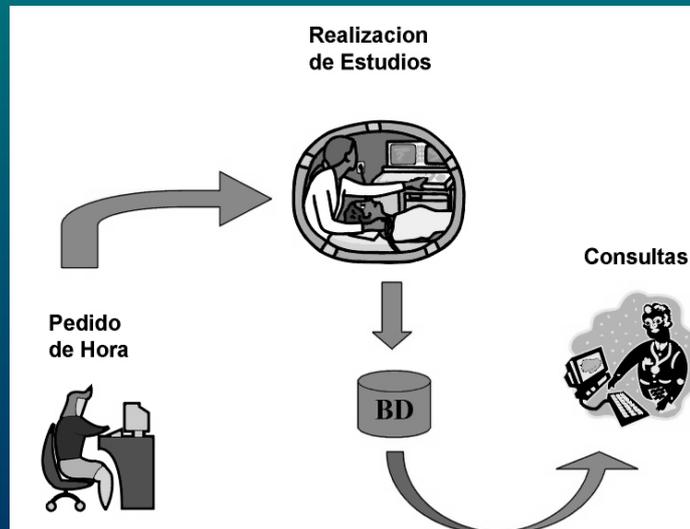
Facultad de Ingeniería
Universidad de la República
Agosto 2002

Introducción

¿Qué es Sirei?

- Es un Sistema de Información de Resultados de Exámenes Imagenológicos, en el cual se integran datos textuales e imágenes.
- Este sistema permite la realización de Altas, Bajas y Modificaciones de registros de pacientes, estudios, diagnósticos y análisis Imagenológicos.

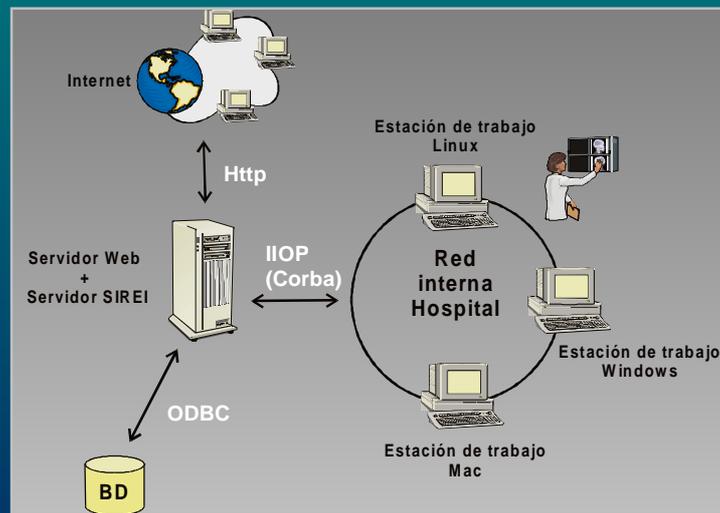
¿Qué es Sirei?



¿Qué es eSirei?

- Este proyecto apunta a desarrollar un Sistema de Información de Resultados de Exámenes Imagenológicos, cliente/servidor multiusuario con interfase Web, en el cual se integran datos textuales e imágenes.
- La idea del sistema propuesto es que se pueda acceder a un servidor, que se encarga de manejar la información, desde varios clientes independientemente de los sistemas operativos que corran en ellos.

Esquema del sistema



Fases del trabajo

- Estudio y comprensión del sistema Sirei anteriormente desarrollado.
- Transformar el sistema anterior a una arquitectura cliente-servidor de tres capas.
- Realizar una interfase web.
- Mejorar calidad del software, incorporar nuevas funcionalidades.
- Optimizar y verificar el software.

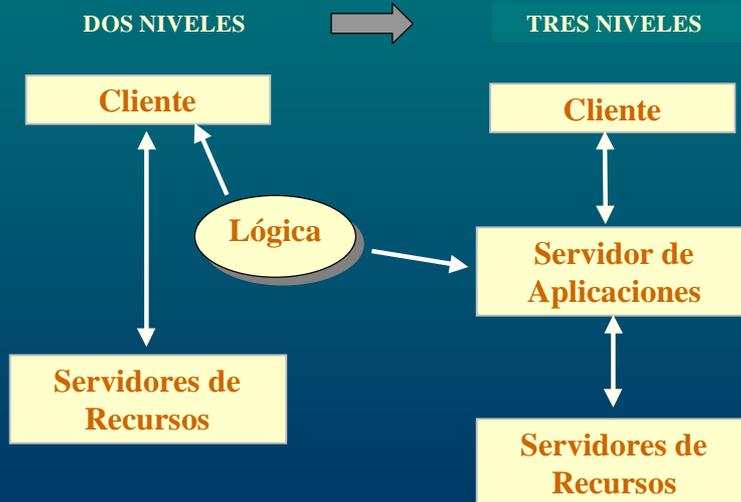
SIREI previamente desarrollado

- Situación anterior
- Sistema desarrollado en dos capas
 - Acceso directo a base de datos, muchas sentencias SQL enviadas por la red.
 - Lógica en el cliente a mantener y administrar.
 - Seguridad a nivel de base de datos.
- Diseño satisfactorio, buena modularización.
- Problemas de rendimiento: gran cantidad de accesos a base de datos, creación y destrucción de objetos innecesarios. Inicialización del programa lento.
- Problemas de manejo de memoria: goteo.
- Difícil detección de errores ya que no había manejo de excepciones.

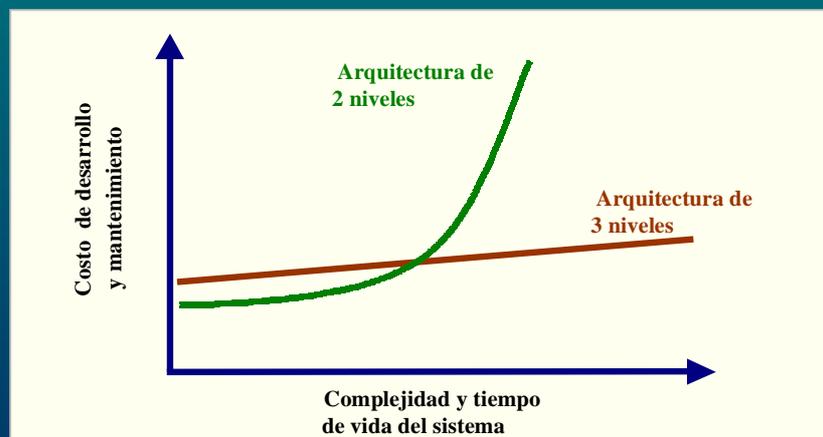
Nuevo SIREI

- Situación actual
- Sistema desarrollado en tres capas
 - Encapsulamiento de datos, no se accede directamente a la base de datos. Implementación en **CORBA**.
 - Lógica en el cliente sencilla.
 - Se puede acceder al servidor desde múltiples plataformas.
- Mejoras en rendimiento: a pesar del peso de CORBA se mejoró sensiblemente el desempeño de la aplicación.
- Manejo de memoria eficiente, utilización de estructuras de datos (hash, listas) para mejorar eficiencia. Utilización de punteros inteligentes para no “perder” memoria.
- Manejo de excepciones para permitir detectar en dónde y por qué sucedió un error.

Pasaje a modelo tres capas

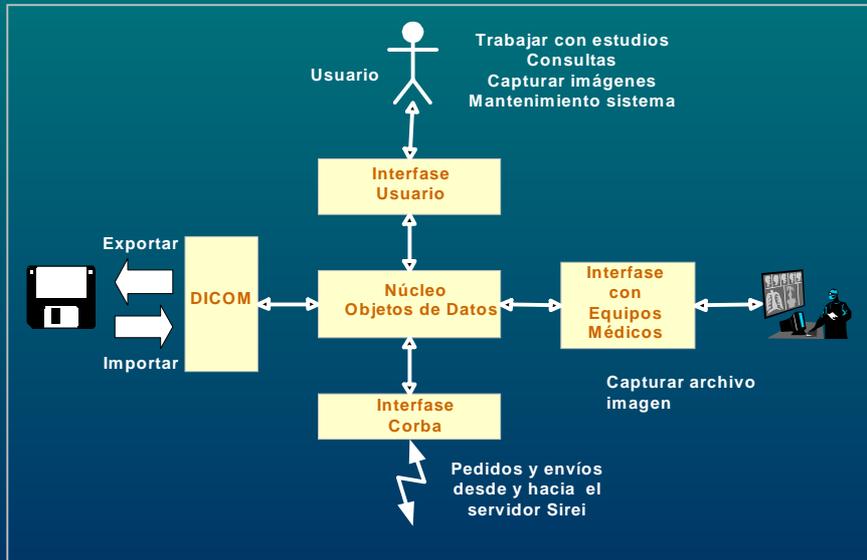


Arquitectura de tres niveles vs. dos niveles

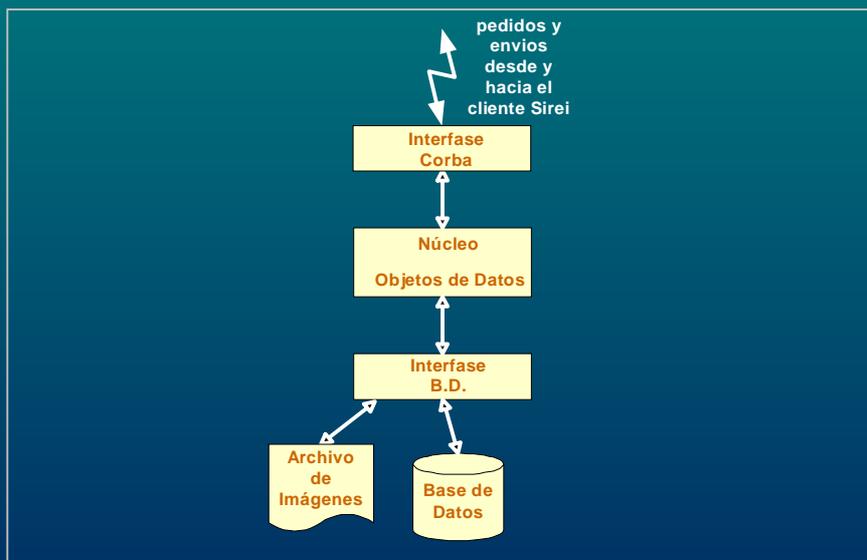


(Fuente: Gartner Group)

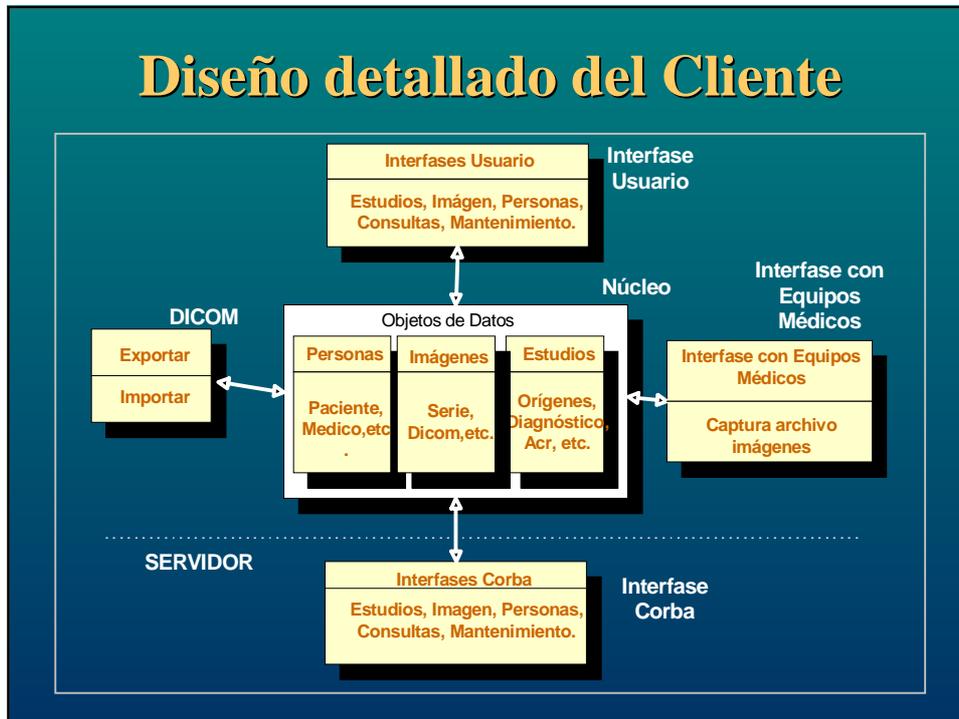
Esquema Cliente Sirei



Esquema Servidor Sirei



Diseño detallado del Cliente



Diseño detallado del Servidor



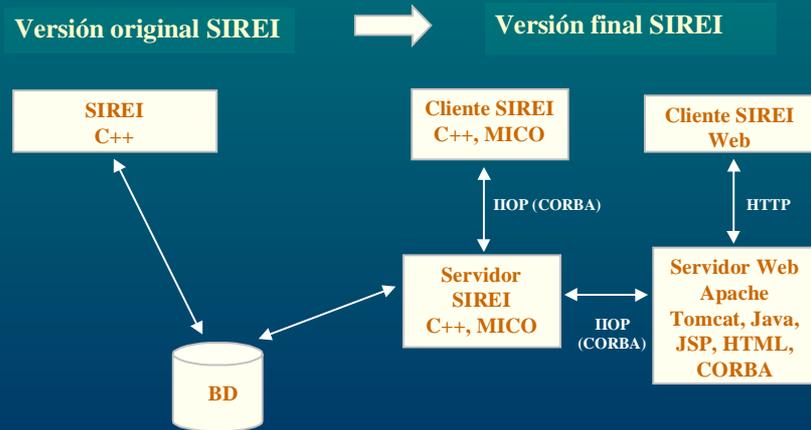
¿ Por qué CORBA ?

- Proporciona la infraestructura que permite la 'conversación' entre objetos, independientemente de la plataforma específica y de las técnicas utilizadas en la implementación de los objetos.
- Logra portabilidad e interoperabilidad de objetos en una red de sistemas heterogéneos.
- CORBA reside en su carácter de especificación abierta y existen implementaciones libres como MICO y JacOrb.

Implementación

- Se utilizó una metodología de desarrollo incremental, ya que el proceso se realizó mediante aproximaciones sucesivas.
- La implementación se dividió en tres etapas:
 - Servidor Sirei → MICO 2.3.7
 - Cliente Sirei → MICO 2.3.7
 - Cliente Web → páginas JSP, J2SDK 1.4

Implementación



Implementación Cliente y Servidor Sirei

- Se utilizó MICO, una implementación en C++ completamente compatible y totalmente libre del estándar CORBA.
- La interfase CORBA de cada objeto se especificó mediante el lenguaje declarativo IDL (Interface Definition Language)
- Una vez definidas las interfases se implementó la lógica del servidor.

Implementación Cliente Web

- Para la implementación de la interfase se eligió la tecnología JSP.
- El cliente web realizado se conecta al servidor Sirei a través de CORBA.
- Permite acceder al sistema en forma remota a través de Internet.

Lenguajes utilizados

- Visual C++ 6
- Java
 - Acceso al servidor Sirei a través de CORBA.
 - Utilizado en páginas JSP.
 - Lenguaje portable, multi-plataforma, orientado a objetos
 - Sintácticamente parecido a C++ y bastante conocido.
 - Amigable para el trabajo en red.

Causas de la elección de la tecnología JSP

- Las páginas JSP correctamente sobre un servidor web Apache con Tomcat, ambos de dominio público.
- “Escribir una vez, ejecutar en cualquier lugar”.
- JSP soporta la separación de roles: diseñadores gráficos y programadores.
- Acceso web para arquitectura de aplicaciones de N capas.
- Otra Opción:
 - Páginas ASP (Microsoft, IIS)

Modificaciones y correcciones a la versión anterior

- Los códigos ACR se cargan a medida que se necesitan.
- Se agregaron más códigos ACR.
- Se agregó manejo de excepciones.
- Se utilizaron estructuras de HASH para mejorar el rendimiento.

Modificaciones y correcciones a la versión anterior

- Impresión de estudios y diagnósticos.
- Errores al agregar médicos y técnicos existentes.
- Se realizaron ajustes en la interfase gráfica del cliente C++.
- Se ingresó la lista de médicos en la base de datos.
- Se realizó un nuevo manual de usuario.
- Se agregó el menú para salir e ingresar con otro usuario.

Futuro y Posibles mejoras

- Vincular Sirei con programas accesorios para el manejo de imágenes en 3D.
- Adaptar e integrar el sistema a otros sistemas dentro del hospital.
- Intercambiar datos almacenados con otras instituciones o profesionales.
- Implementar la función administrativa “Pedido de Hora”.
- Realización de más consultas con respecto a los estudios, diagnósticos, insumos, etc.
- Continuar el desarrollo del cliente web.

Conclusiones

- La información es accesible, incluso en forma remota a través de Internet, para la realización de consultas estadísticas y el conocimiento del hospital.
- La arquitectura cliente-servidor de tres niveles brinda escalabilidad y buen rendimiento.
- Se utilizaron de herramientas de dominio público, no hace falta comprar software.

Muchas Gracias

¿Preguntas?