

# Ejercicio

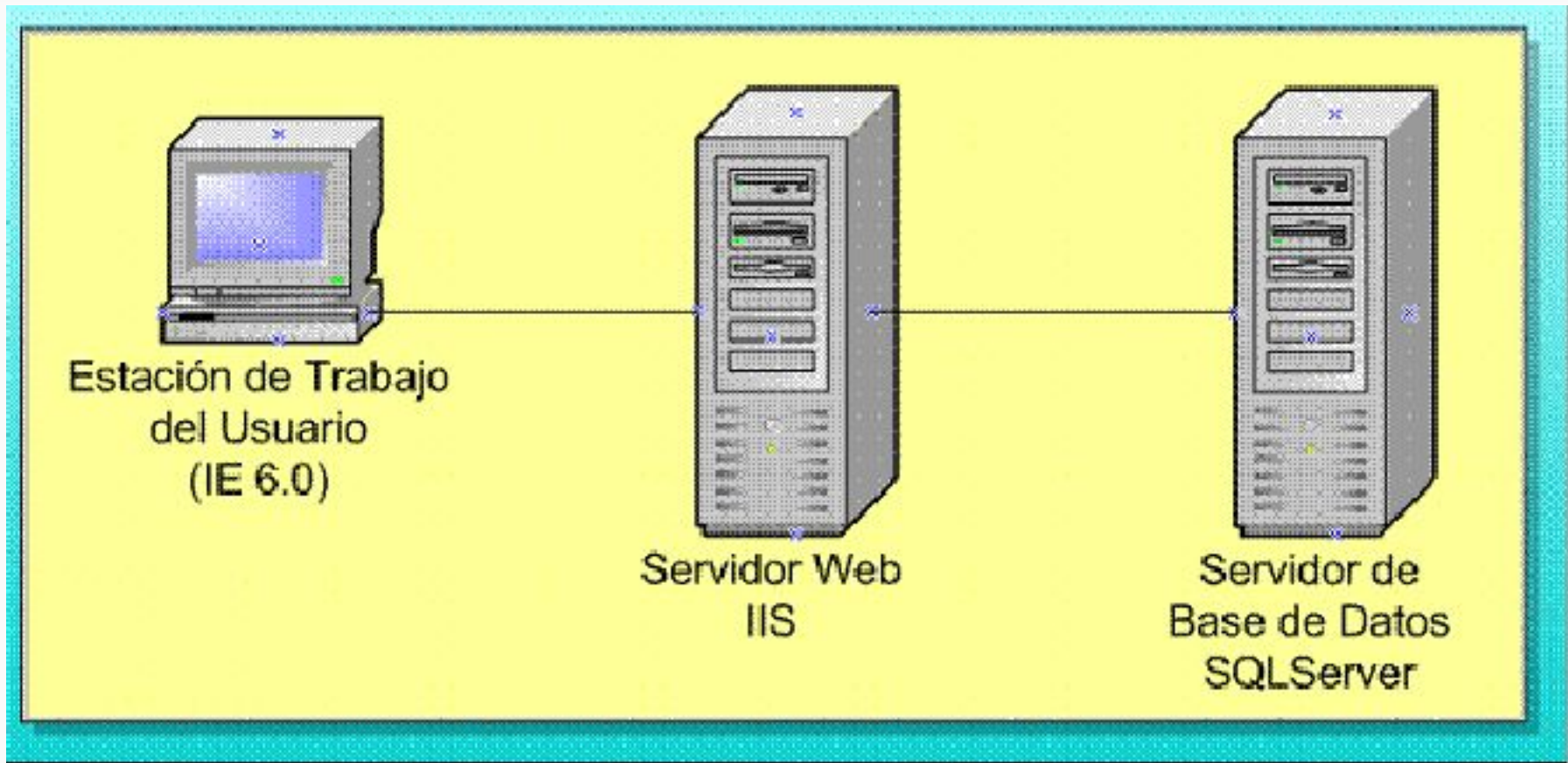
Si nos pidieran contar y describir la arquitectura del último sistema en que hemos participado, que haríamos?

# Modelo de Vistas de Arquitectura 4+1

Philippe Kruchten

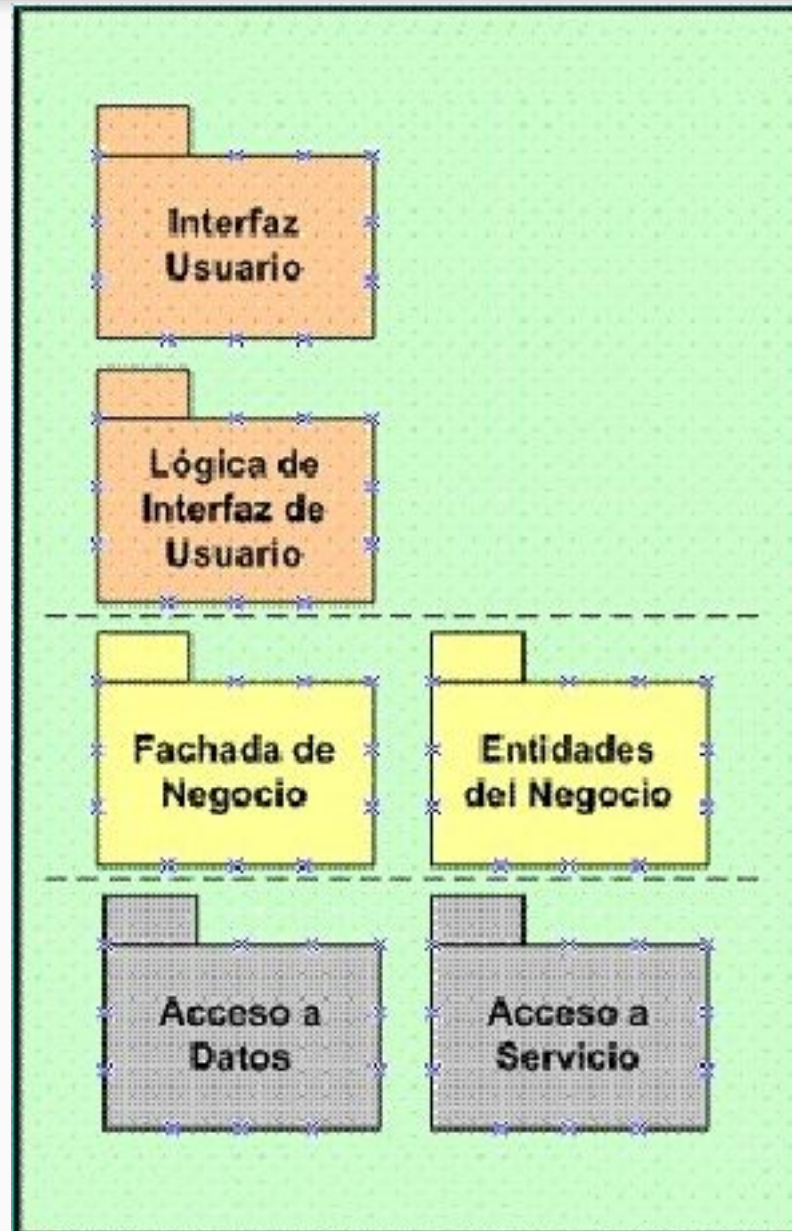


Podemos encontrarnos con cosas como:

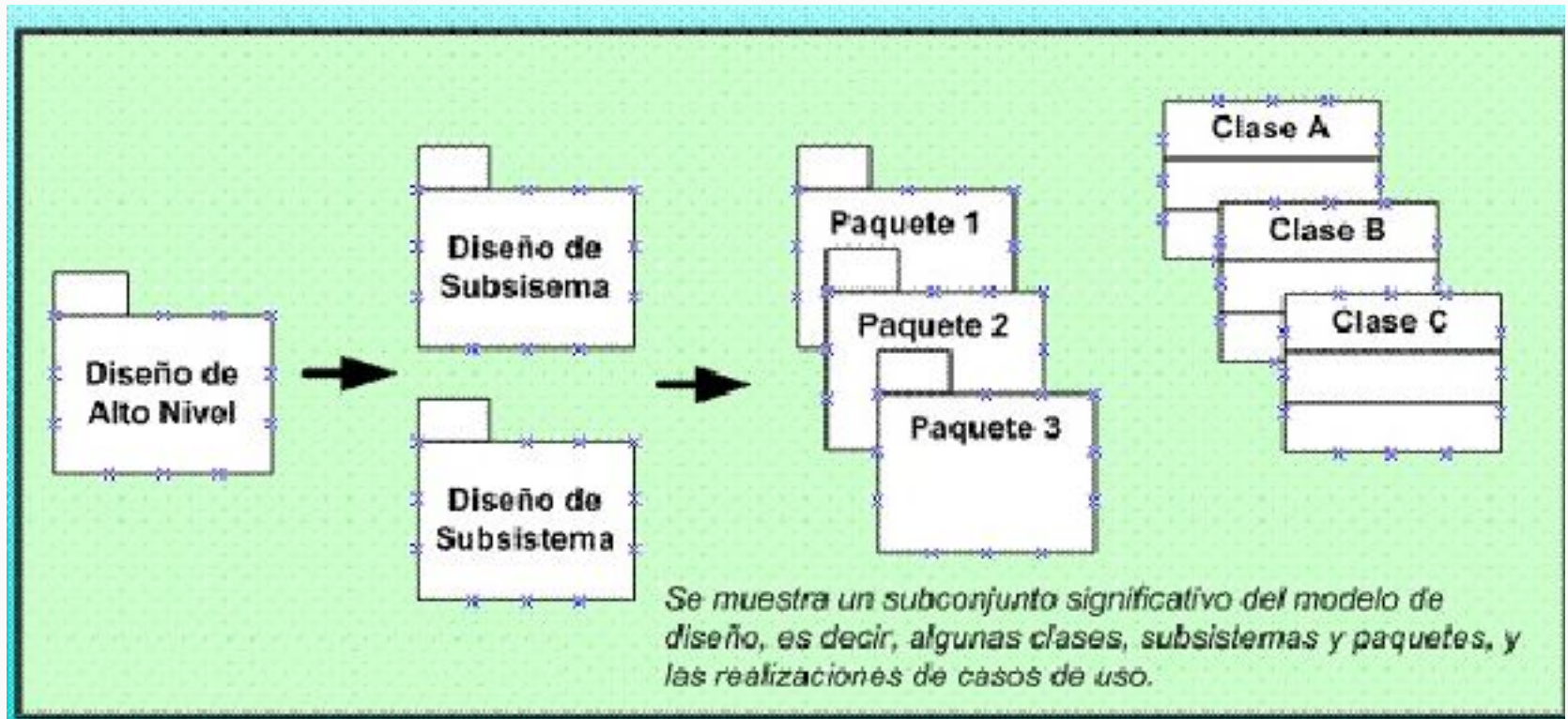


El sistema tendría una Arquitectura Web basada en plataforma Microsoft, o sea, un cliente usando un Internet Explorer 6.0, un servidor corriendo IIS y los componentes de sistema desarrollados en ASP.NET accediendo a otro servidor que contendría los datos dentro de un SQL Server 2008

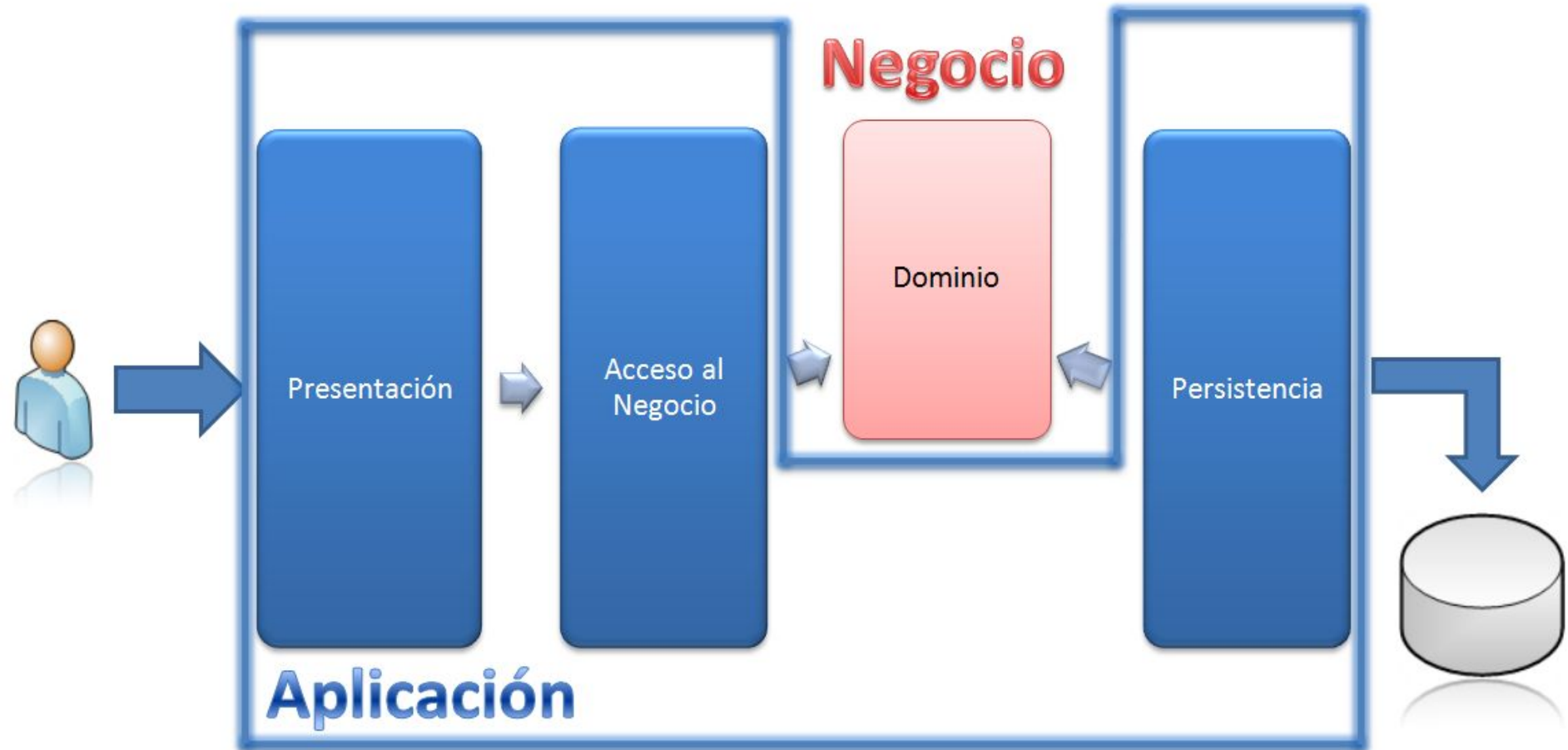
Podemos encontrarnos con cosas como:



Podemos encontrarnos con cosas como:

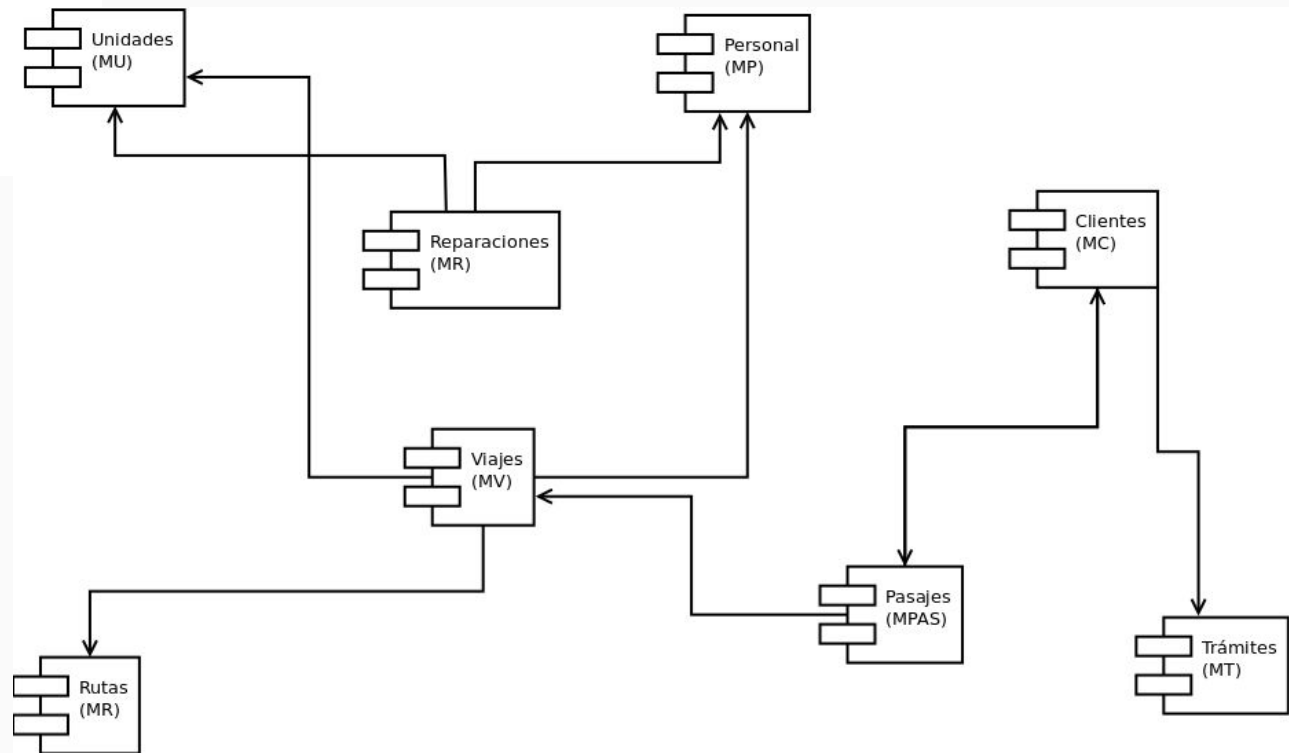
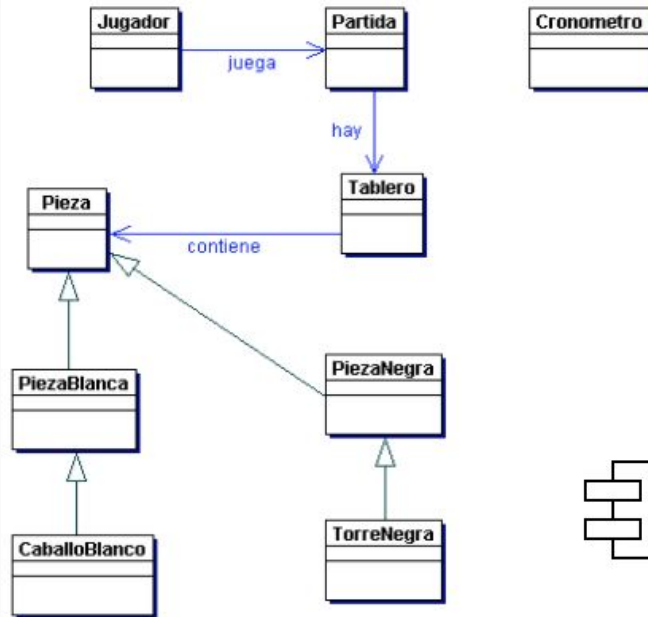


Podemos encontrarnos con cosas como:





Podemos encontrarnos con cosas como:



# Arquitectura

- ieee-1471 defining-architecture

Conceptos o propiedades fundamentales de un sistema en su entorno encarnado en sus elementos, relaciones y en los principios de su diseño y evolución

- La arquitectura de software representa la estructura o las estructuras del sistema, que consta de componentes de software, las propiedades visibles externamente y las relaciones entre ellas.



# Arquitectura

Fowler:

La arquitectura de software es aquellas decisiones que son importantes y difíciles de cambiar.

# Arquitectura

- ¿Cómo contamos la arquitectura de un sistema?
- ¿Cómo diseñamos una posible arquitectura para un sistema como para hacer una prueba de concepto?
- ¿Cómo documentamos la arquitectura de un sistema?

# Un paréntesis..

- ¿Que debemos tener en cuenta cuando escribimos un documento, un informe o queremos transmitir alguna información?

# Un paréntesis..

Entonces, a quién está dirigido un Documento de Arquitectura?

En realidad existen muchos interesados o stakeholders de un documento de arquitectura, e incluso estos pueden tener distintos intereses y conocimientos.

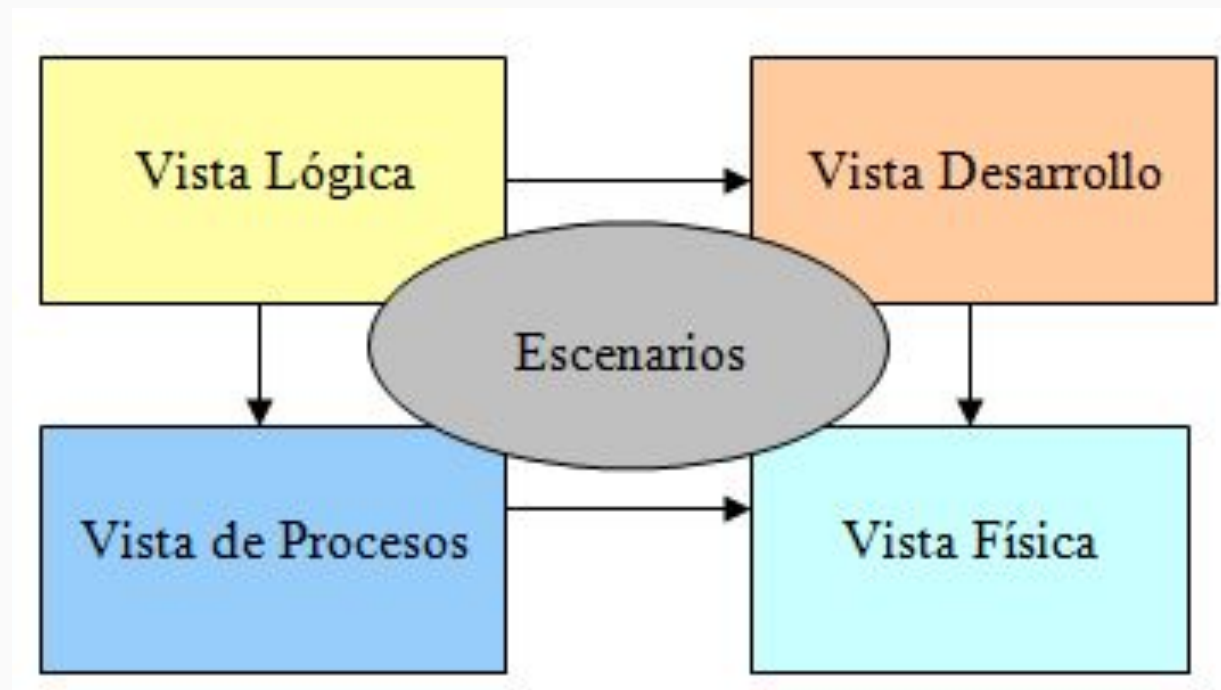
Dentro de un equipo de trabajo podemos encontrar: analistas, desarrolladores, testers, líderes de proyectos, arquitectos, clientes, gente de infraestructura, de despliegue, usuarios, administradores de bases de datos, etc.

Entonces cómo escribir un único documento o diagrama que pueda explicar la arquitectura propuesta para resolver cierto problema?

# Modelo de Vistas 4+1



# Modelo de Vistas 4+1 - Kruchten





## VISTA LOGICA

La vista lógica apoya principalmente los requisitos funcionales –lo que el sistema debe brindar en términos de servicios a sus usuarios.

Aquí se aplican los principios de abstracción, encapsulamiento y herencia. Esta descomposición no sólo se hace para potenciar el análisis funcional, sino también sirve para identificar mecanismos y elementos de diseño comunes a diversas partes del sistema

## VISTA PROCESOS

La vista de procesos toma en cuenta algunos requisitos no funcionales tales como el rendimiento y la disponibilidad.

Se enfoca en asuntos de concurrencia y distribución, integridad del sistema, de tolerancia a fallas.

Un proceso es una agrupación de tareas que forman una unidad ejecutable. Los procesos representan el nivel al que la arquitectura de procesos puede ser controlada tácticamente (i.e., comenzar, recuperar, reconfigurar, y detener). Además, los procesos pueden replicarse para aumentar la distribución de la carga de procesamiento, o para mejorar la disponibilidad.

## VISTA DESARROLLO (o de COMPONENTES)

La vista de desarrollo se centra en la organización real de los módulos de software en el ambiente de desarrollo del software.

El software se empaqueta en partes pequeñas –bibliotecas de programas o subsistemas– que pueden ser desarrollados por uno o un grupo pequeño de desarrolladores.

La vista de desarrolla tiene en cuenta los requisitos internos relativos a la facilidad de desarrollo, administración del software, reutilización y elementos comunes, y restricciones impuestas por las herramientas o el lenguaje de programación que se use.

## VISTA FISICA (o de DESPLIEGUE)

La vista física toma en cuenta primeramente los requisitos no funcionales del sistema tales como la disponibilidad, confiabilidad (tolerancia a fallas), rendimiento (throughput), y escalabilidad.

El software ejecuta sobre una red de computadores o nodos de procesamiento. Los variados elementos identificados –redes, procesos, tareas y objetos– requieren ser mapeados sobre los nodos. Esperamos que diferentes configuraciones puedan usarse: algunas para desarrollo y pruebas, otras para mostrar el sistema en varios sitios para distintos usuarios. Por lo tanto, la relación del software en los nodos debe ser altamente flexible y tener un impacto mínimo sobre el código fuente.

## ESCENARIOS

Los elementos de las cuatro vistas trabajan conjuntamente en forma natural mediante el uso de un conjunto pequeño de escenarios relevantes.

Los escenarios son de alguna manera una abstracción de los requisitos más importantes.

Sirve a dos propósitos principales:

- Como una guía para descubrir elementos arquitectónicos durante el diseño de arquitectura
- Como un rol de validación e ilustración después de completar el diseño de arquitectura, en el papel y como punto de partido de las pruebas de un prototipo de la arquitectura.

## CONCLUSIONES

- Permite a través de diferentes vistas analizar distintas perspectivas del problema, focalizándose en el problema en cuestión
- Concentra en un único documento las principales decisiones tomadas sobre el sistema
- Permite a nuevos integrantes del equipo entender la arquitectura del sistema y ubicarse dentro de la solución
- Permite discutir con todos los stakeholders las distintas decisiones y validarlas en una etapa temprana

# Bibliografía

- [4+1view-architecture-Krutchen.pdf](#)
- [Making Architecture Matter - Martin Fowler Keynote](#)
- <http://www.iso-architecture.org/ieee-1471/defining-architecture.html>
- <https://github.com/7510-tecnicas-de-disenio/material-clases/tree/master/lecturas-recomendadas>