

Arquitectura de Software

Sorey Bibiana Garcia Zapata

Para Principiantes





Siguiendo mi *enfoque pedagógico personal*, someteré el tema de la **arquitectura de software** a la ya típica comparación con la ingeniería civil y el rol de los arquitectos de edificaciones.

La razón de este enfoque es *simple*, una realidad poco conocida o comprendida, es más **fácil de asimilar**, cuando es mostrada a través de un símil con otra que **si conocemos**, o bien, con otra que puede intuirse coherentemente usando el sentido común.

Empecemos por un par de ideas simples....



La arquitectura es un instrumento cuya función principal es la de intervenir en favor del hombre.

James M Fitch



La arquitectura debe ser una respuesta, no una imposición. La mayoría de los arquitectos que construyen edificios **extraños** asegura que lo hace porque ahora la tecnología lo hace posible. Eso me parece **absurdo**.

Poder hacer una cosa no legitima hacerla.



Necesitamos soluciones para los problemas reales, no inventar problemas para poder empatar con nuevas soluciones

Glenn Murcutt

Esta última comparación podrá no gustarle a muchos *arquitectos profesionales*, pero en realidad tiene mucho sentido, en tecnología, tanto como en otros escenarios...

“Poder hacer una cosa no legitima hacerla”



Necesitamos soluciones para los problemas reales, no inventar problemas para poder empatar con nuevas soluciones!

Viéndolo de esa forma, en realidad el rol de un arquitecto de edificaciones y un arquitecto de software parece enfrentar los mismos retos



Por lo que es claro, que se debe contar, con un conjunto básico de habilidades y conocimientos para ejercer este tipo de roles





**Y es que no es lo
mismo construir
esto...**



Que esto!

O esto!



Que se necesita para construir algo así?





Que tal esto?

Y esto?



O esto?



**Cada escenario plantea retos,
condiciones y necesidades
diferentes!**

*Que tal si los vemos otra vez,
mientras nos preguntamos...*

Que **herramientas**, **personas**,
presupuesto, **conocimiento** y **tiempo**
necesitamos para cada escenario?



Pues bien, podemos decir sin miedo a equivocarnos, que todas las consideraciones que se nos ocurran con respecto a definir la arquitectura de edificaciones, deberán ser tenidas en cuenta también al momento de definir la arquitectura de software



Conocen la Mansión Winchester?

En 1884, **Sarah L. Winchester** enviuda quedando en sus manos la herencia del acaudalado heredero de los **rifles Winchester**.

Pero tras tan inmensa fortuna, y según una vidente de la época a la que la desdichada viuda consultó, las almas perdidas de aquellos que murieron frente a alguno de estos rifles se encontraban en pena.

Años atrás había perdido también a su hija Annie y ahora, la muerte de su marido terminó por desequilibrarla.

Sarah compró un extenso terreno en California, e hizo levantar una casa que hubo de estar construyéndose hasta el mismo día de su muerte, **38 años después**, en 1922. Ya que según aquella vidente, sólo así las almas encontrarían donde morar mientras no hallaran el reposo eterno.



La arquitectura de esta mansión sorprende a todos y se escapa a los límites de la razón. Y no es tanto porque esta mansión estilo victoriano tenga 160 habitaciones, tres ascensores, 47 chimeneas, sistema de alcantarillado y de calefacción, luces de gas que se encienden apretando un botón, (todo esto adelantado a su época), sino porque en ella se puede encontrar ventanas de dan a paredes, escaleras que no llevan a ninguna parte, puertas en medio de las ventanas en la fachada de los pisos superiores y otras rarezas.





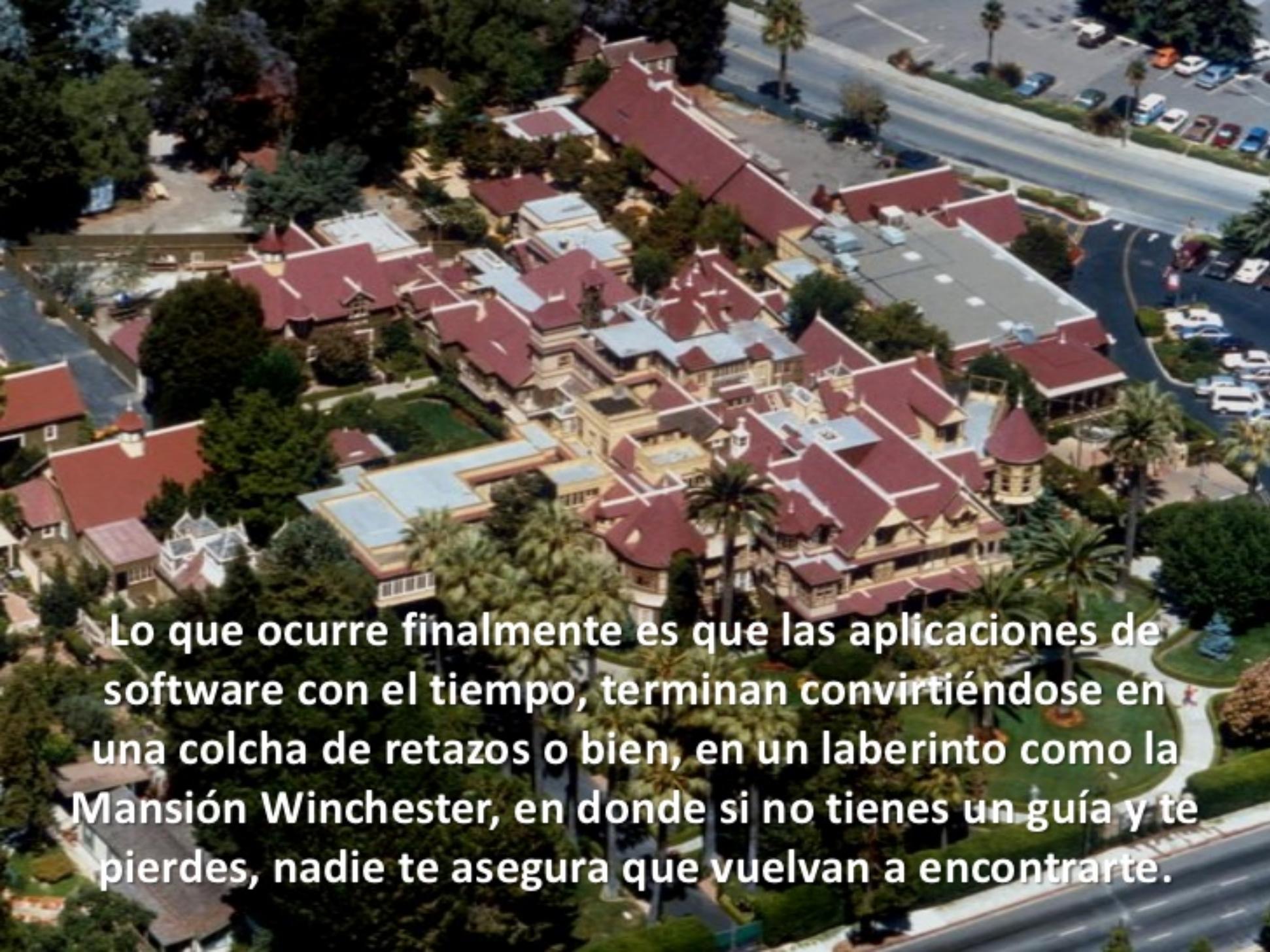
Pero...

**Qué tiene que ver
esta historia con la
Arquitectura de
Software?**

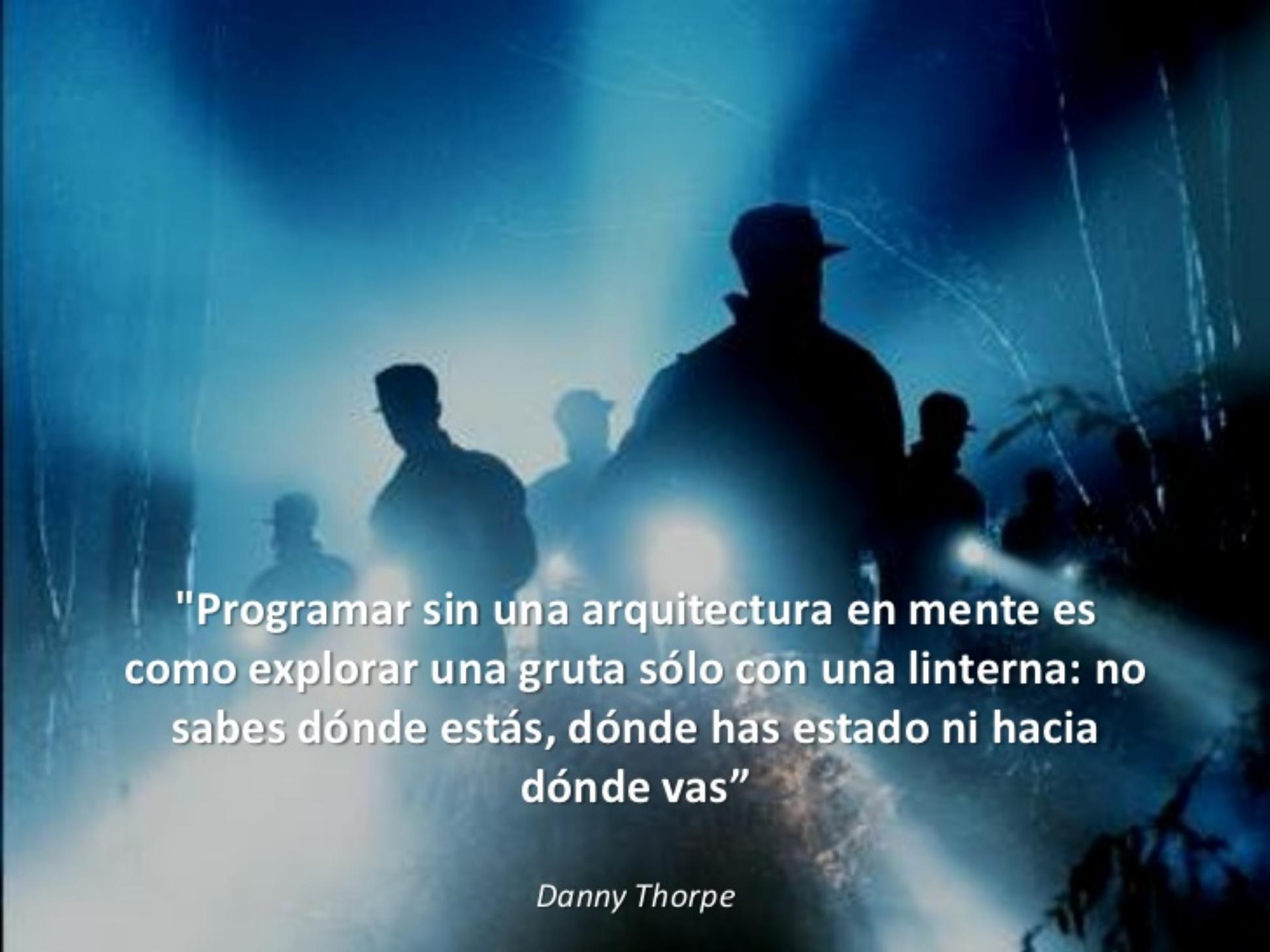
Pues bien, resulta que esta situación en el contexto del desarrollo de software, es más común de lo que debería...

Cuando un desarrollador es asignado a la tarea de mantener y evolucionar un sistema legado, cuya arquitectura tiene fallas o incluso que simplemente no está documentada, elige reconstruir partes o crear sus propias rutas dentro del código...



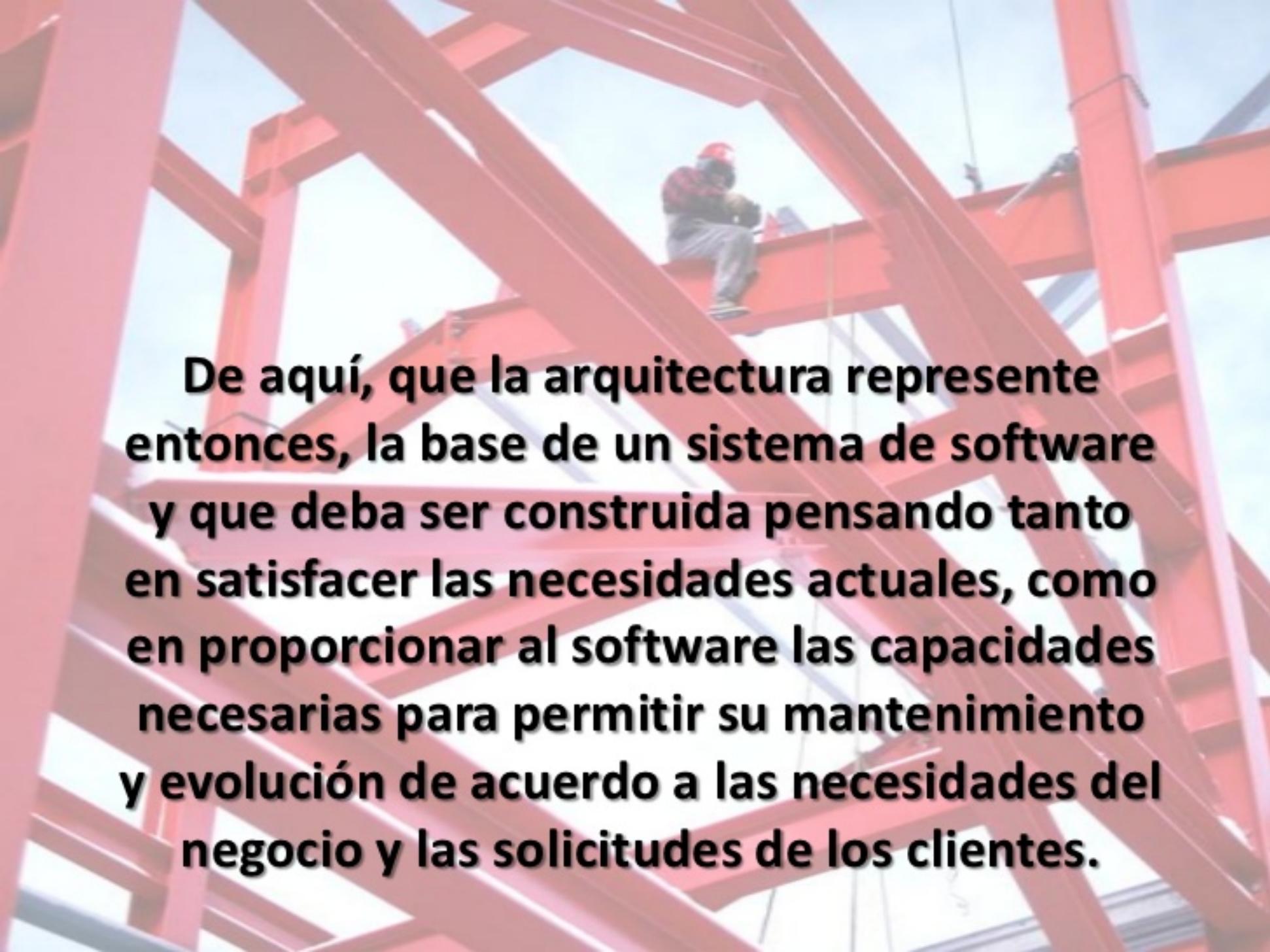


Lo que ocurre finalmente es que las aplicaciones de software con el tiempo, terminan convirtiéndose en una colcha de retazos o bien, en un laberinto como la Mansión Winchester, en donde si no tienes un guía y te pierdes, nadie te asegura que vuelvan a encontrarte.

A photograph of several silhouetted figures wearing hats, standing in a dark, rocky environment. A single beam of light from a flashlight cuts through the darkness, illuminating the textured rock walls and floor. The scene serves as a metaphor for programming without a clear architectural plan.

**"Programar sin una arquitectura en mente es
como explorar una gruta sólo con una linterna: no
sabes dónde estás, dónde has estado ni hacia
dónde vas"**

Danny Thorpe



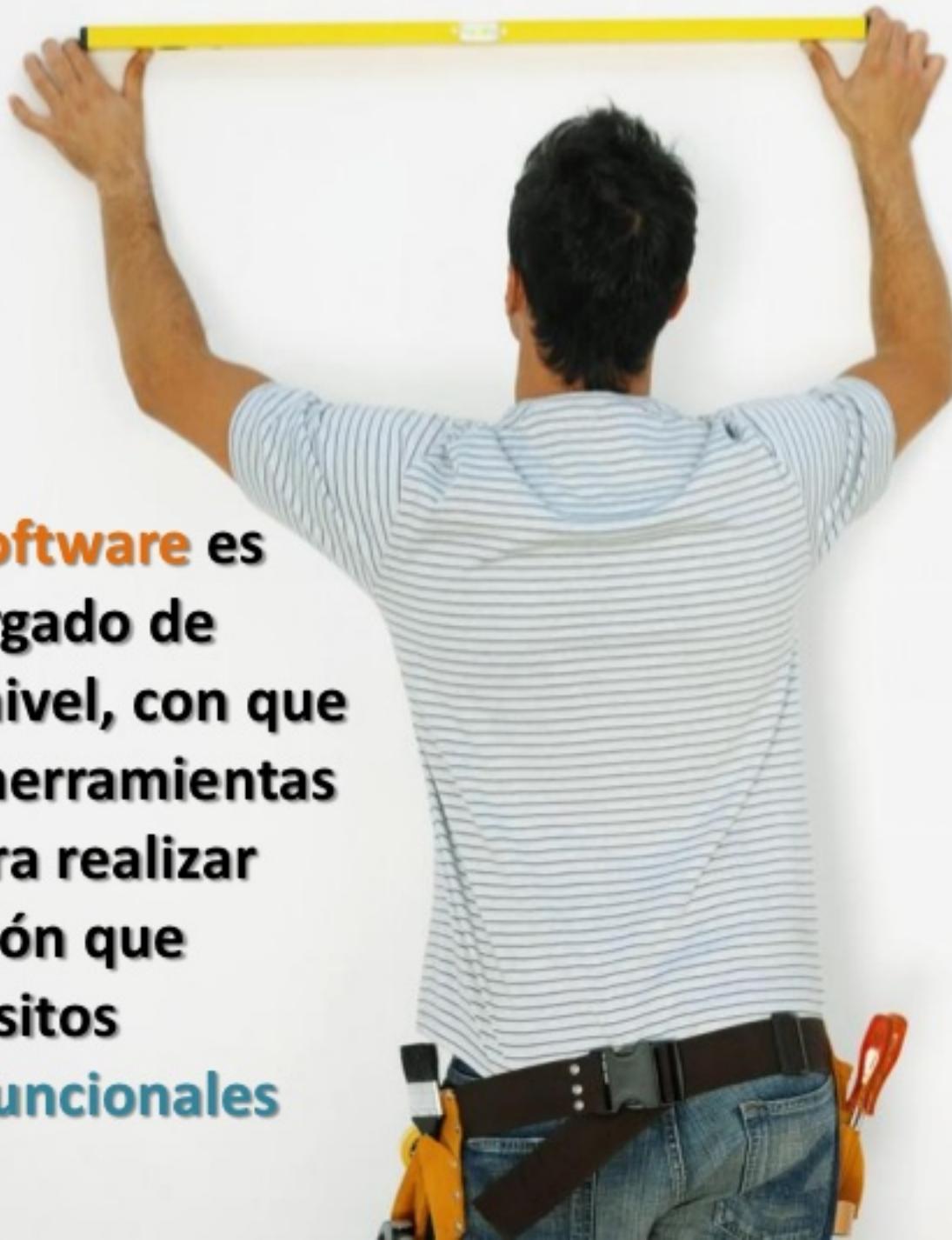
De aquí, que la arquitectura represente entonces, la base de un sistema de software y que deba ser construida pensando tanto en satisfacer las necesidades actuales, como en proporcionar al software las capacidades necesarias para permitir su mantenimiento y evolución de acuerdo a las necesidades del negocio y las solicitudes de los clientes.

**Cuando se piensa en la arquitectura de un sistema,
debe tomarse en cuenta primero, lo más crítico e
importante...**



**Y luego complementar para satisfacer las necesidades
generales del sistema**

El arquitecto de software es entonces, el encargado de establecer a que nivel, con que estrategia, y que herramientas son necesarias para realizar una implementación que satisfaga los requisitos funcionales y no funcionales de los sistemas.



A professional man in a light blue dress shirt and a dark tie is leaning forward, intently focused on a white computer monitor. He is positioned on the left side of the frame, with his head tilted down towards the screen. The monitor has a perforated metal grille on its front panel. The background is plain white.

Y además que debe ser una persona capaz de identificar las necesidades de los **negocios**, las habilidades de su **equipo de trabajo** y la viabilidad de las **tecnologías** disponibles para el desarrollo de software

Un buen arquitecto debe estar en capacidad de entender todas las **condiciones a las que se verá sometido un sistema y proponer una solución acorde a cada **escenario** en particular.**



Por tanto, la madurez de un arquitecto dará a las aplicaciones de software que tenga a su cargo, una especificación coherente, para enfrentar un conjunto de riesgos mucho más reducido, que en el caso de un arquitecto aprendiz

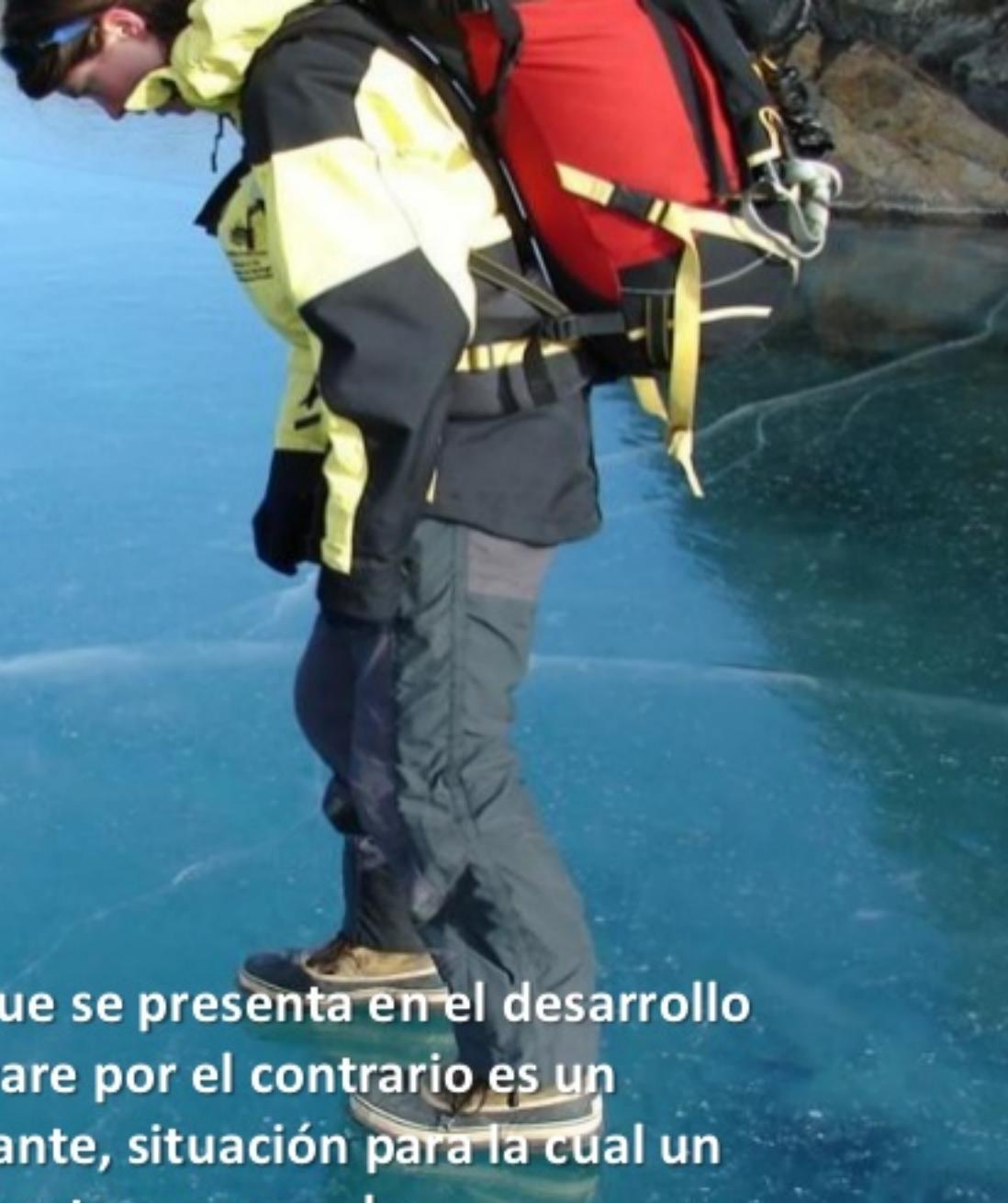


Y es que...

"Desarrollar software de calidad basado en requisitos y caminar sobre el agua son cosas muy fáciles...

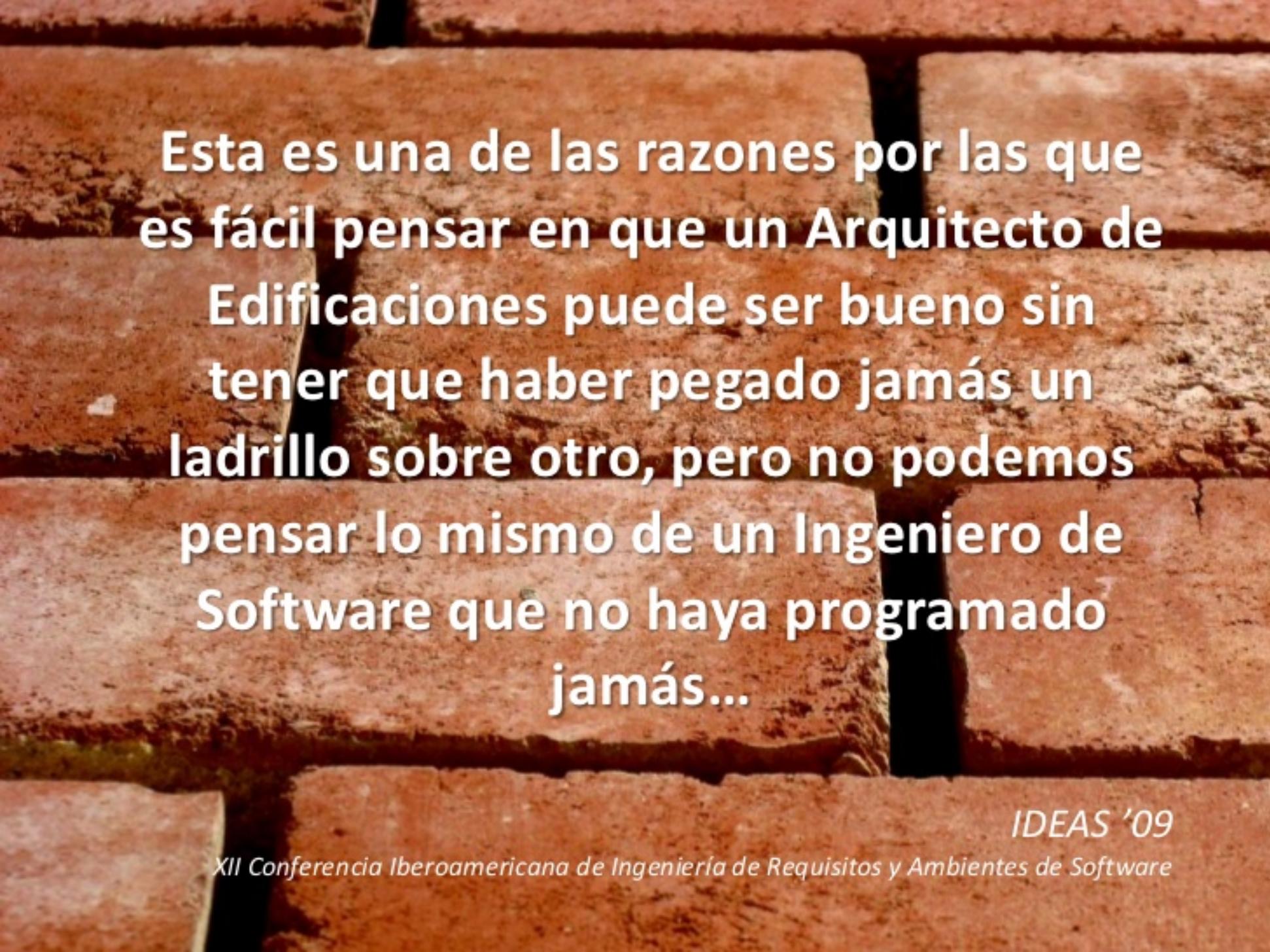
si ambos están congelados"

Pero esa no es la situación que se presenta en el desarrollo de software; construir software por el contrario es un escenario evolutivo y cambiante, situación para la cual un arquitecto de software debe estar preparado.



Ahora bien, a diferencia de los arquitectos de edificaciones, los arquitectos de software son los encargados de construir la base de la arquitectura de los proyectos...



The background of the slide features a stack of reddish-brown clay bricks, some of which are partially broken or missing, creating a textured, layered effect.

**Esta es una de las razones por las que
es fácil pensar en que un Arquitecto de
Edificaciones puede ser bueno sin
tener que haber pegado jamás un
ladrillo sobre otro, pero no podemos
pensar lo mismo de un Ingeniero de
Software que no haya programado
jamás...**

IDEAS '09

XII Conferencia Iberoamericana de Ingeniería de Requisitos y Ambientes de Software

Ahora bien, al igual que los trabajadores de una obra, **un equipo de desarrollo** debe tener las habilidades necesarias para realizar desarrollo del sistema y seguir la indicaciones del **arquitecto**, buscando el éxito del proyecto.



Otra parte de nuestro símil está a en los **planos** y **maquetas** usados por los arquitectos para especificar y detallar la forma en que las edificaciones serán construidas.



A photograph showing a person from behind, sitting at a desk covered with architectural drawings and blueprints. In the foreground, a large-scale architectural model of a building is visible, featuring intricate details and multiple levels. The person appears to be working on the project.

Al igual que ellos, los arquitectos de software usan modelos que representan las especificaciones y necesidades técnicas de los sistemas.

A photograph showing a person from the side, wearing a dark apron over a tank top, working at a light-colored wooden desk. They are holding a white plastic bottle with a purple label, possibly containing glue or paint. On the desk, there is a large sheet of paper with a grid pattern, several small wooden sticks, a roll of black tape, a yellow box labeled 'Tesa', and a bottle of water. The background is a plain grey wall.

La construcción de estos artefactos, en ambos contextos requiere de herramientas, habilidades y conocimientos, y servirá como guía de la en el proceso de construcción

**Y así como los planos, diseños y
maquetas muestran diferentes
vistas de un mismo lugar...**





...de forma global...

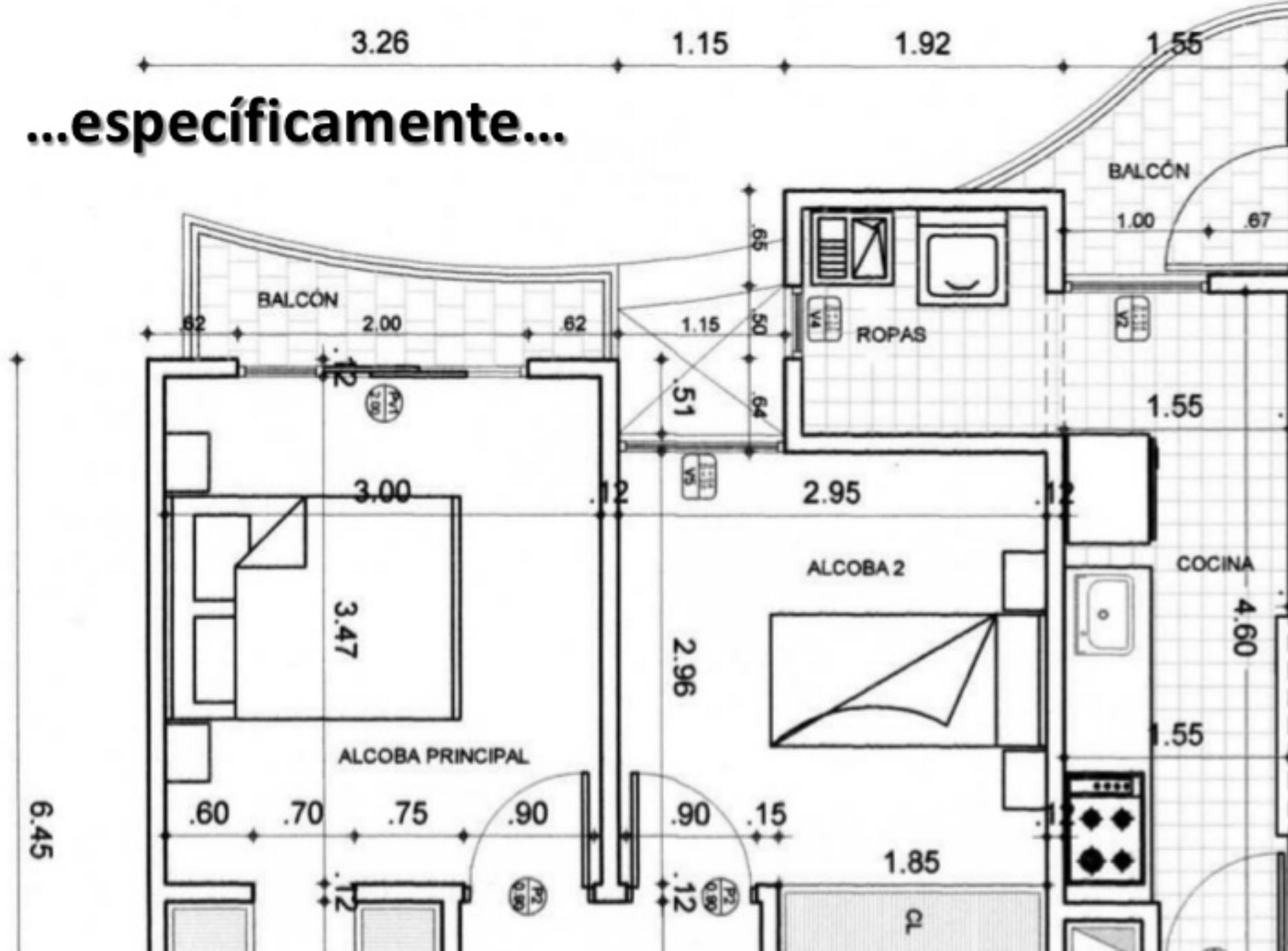


...como una visión completa...



...parcialmente...

...específicamente...

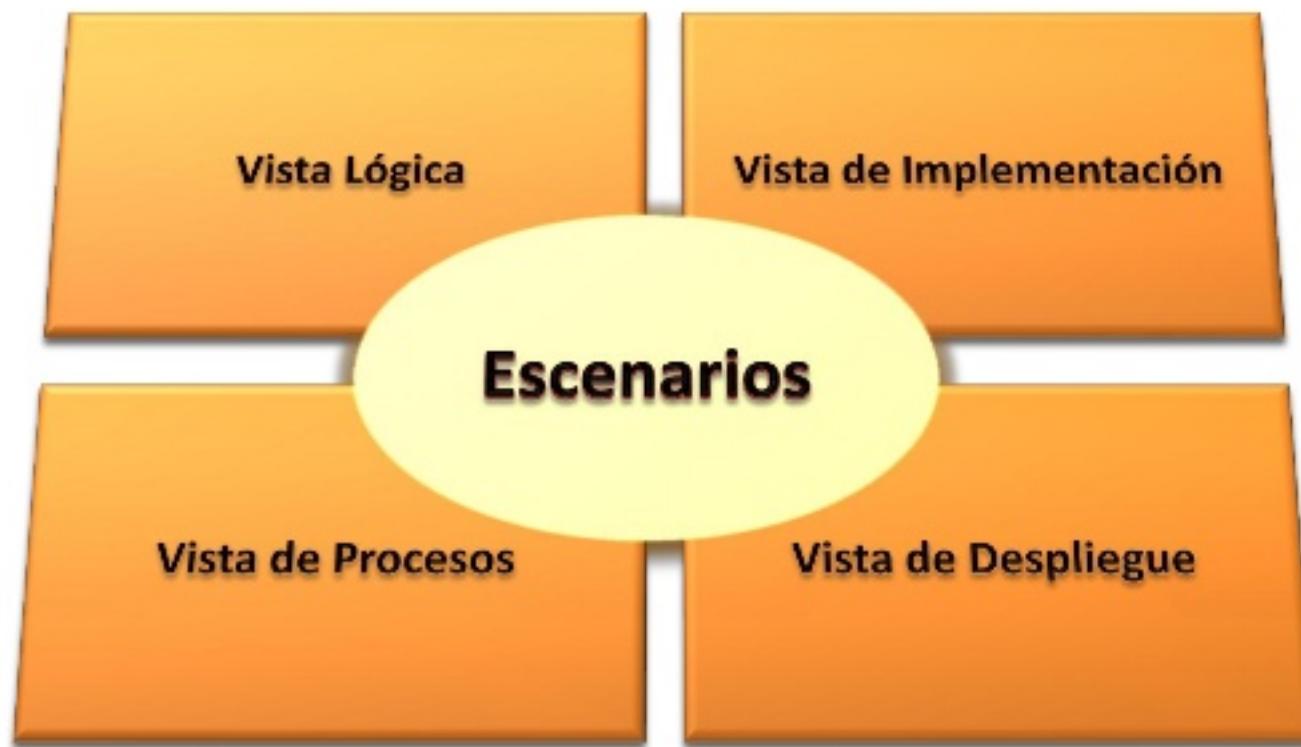


o bien,
con un alto nivel de detalle.

...los diagramas a través de los cuales se representa el diseño y distribución del software, pueden mostrar diferentes vistas de un mismo sistema y de las condiciones que existen en el entorno donde se despliega



El modelo 4 + 1 vistas, es una propuesta que establece las diferentes perspectivas a través de las cuales se puede representar el diseño y arquitectura de un sistema de software



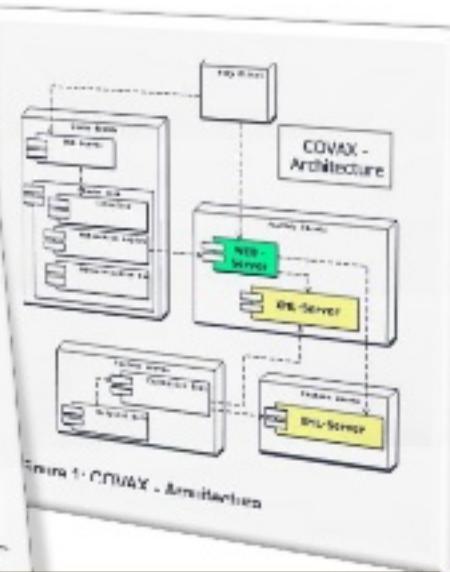
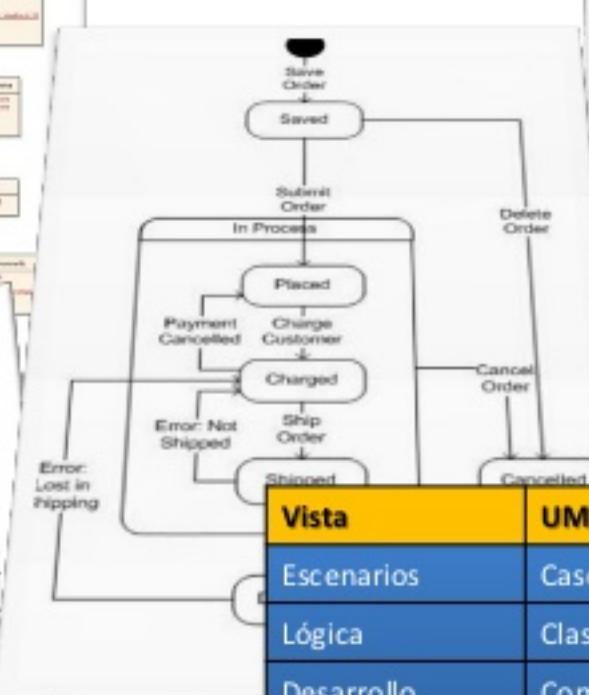
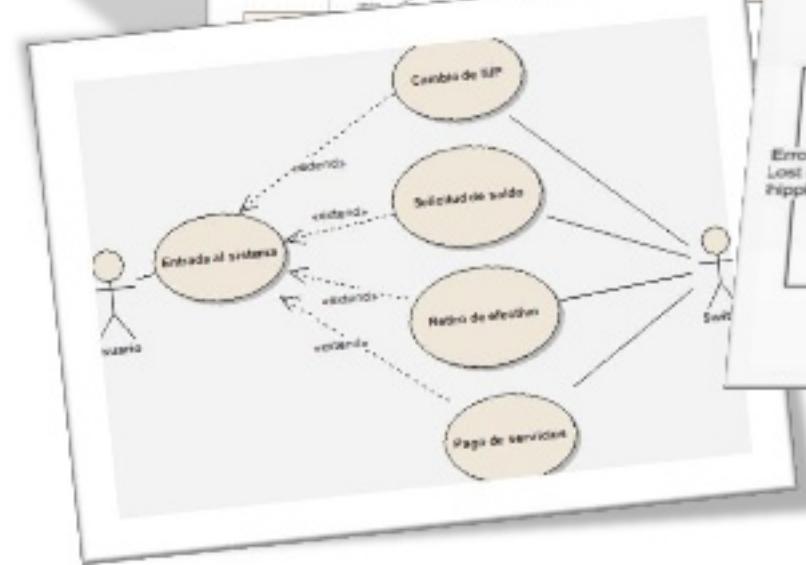
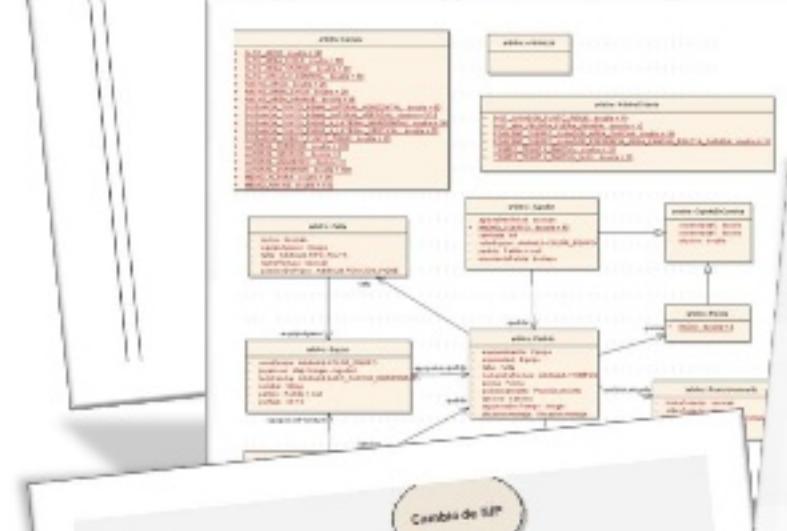
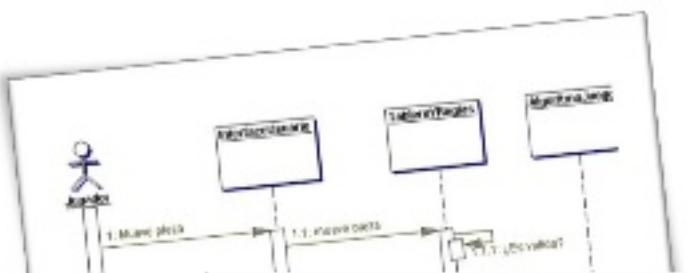
Kruchten P. Architectural Blueprints—The “4+1” View Model of Software Architecture. IEEE Software,
November 1995, 12 (6), pp.42-50.

Esta propuesta presenta su propio esquema de modelado, pero como bien sabemos, la notación más reconocida para el modelamiento de sistemas de software es el **UML** *



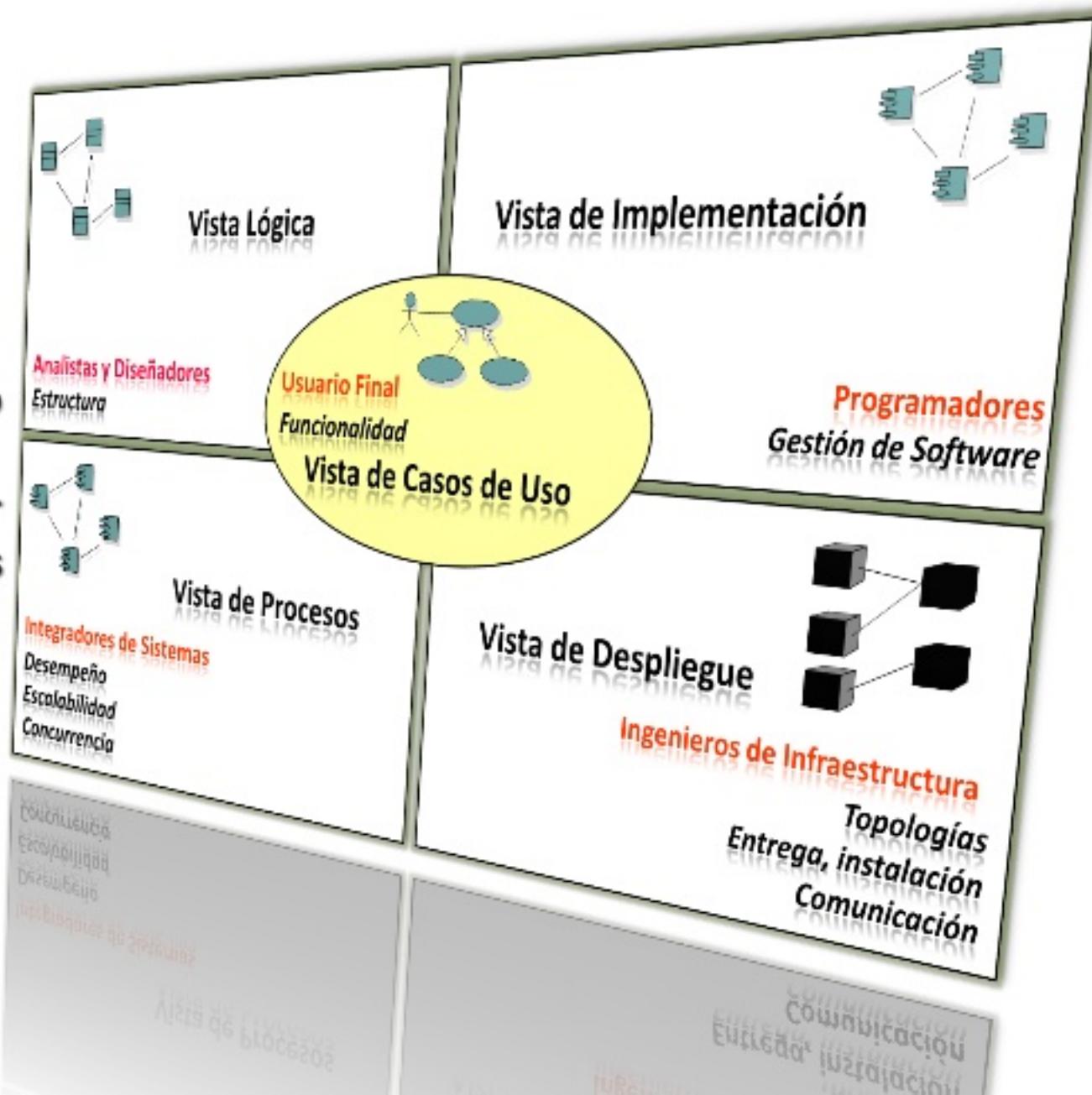
* UML es un lenguaje de modelado visual que se usa para *especificar, visualizar, construir y documentar artefactos de un sistema de software*, y se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

Sin embargo, cada una de las vistas del modelo 4 + 1, puede equiparse contra uno o varios diagramas de UML



Vista	UML
Escenarios	Casos de Uso
Lógica	Clases, de Estados y Colaboración
Desarrollo	Componentes
Física	Despliegue
Procesos	Actividad, Estados, Secuencia

De esta forma es posible representar las diferentes perspectivas de un sistema desde 2 propuestas de trabajo conocidas y enfocadas a satisfacer las necesidades de los diferentes participantes del proceso de desarrollo





Esta es una visión inicial acerca de la arquitectura de software, para profundizar en ella debemos detallar de una forma mas específica, cada uno de los conceptos que hemos mencionado.

Sin embargo eso será en una próxima oportunidad

A close-up photograph of a child's hands playing with colorful wooden blocks on a light-colored table. The child is wearing a grey and white textured sweater with red cuffs. The blocks are various colors including blue, red, yellow, and green, and are being used to build a small structure.

Gracias!!!

No olvides visitar mi blog y dejarme tus comentarios

<http://soreygarcia.blogspot.com>

Hasta pronto!