

La arquitectura de software es como la base que sostiene un sistema. Son las decisiones que hacen que el código no solo funcione hoy, sino que también se pueda mantener, mejorar y escalar con el tiempo. Creo que es importante porque define si un producto será fácil de evolucionar o si terminará siendo un problema cada vez que se quiera cambiar algo. Es decir, una buena arquitectura no solo debe servir para que un sistema arranque y funcione, sino para que siga siendo útil y adaptable conforme cambien las necesidades del negocio o la tecnología.

Un punto clave es que la arquitectura no está separada del diseño, sino que ambos forman parte de un mismo todo. Desde los detalles más pequeños en el código hasta la estructura general del sistema, todo suma. Si esos elementos están bien organizados, se logra reducir el esfuerzo que requiere mantener vivo un proyecto. Si están mal, los costos crecen con cada nueva versión, y llega un momento en que hasta lo más simple se vuelve lento, caro y riesgoso de implementar.

### Escalabilidad

Es la capacidad de que un sistema soporte más usuarios, datos o procesos sin fallar. Por ejemplo, si una aplicación pasa de 100 a 100,000 usuarios, la arquitectura debe permitir crecer sin colapsar. Sin escalabilidad, un sistema exitoso se vuelve inútil rápidamente.

### Mantenibilidad

Significa qué tan fácil es corregir errores o agregar nuevas funciones sin romper todo lo demás. Creo que es clave porque un software siempre va a cambiar; si no se puede mantener, se vuelve costoso y lento de mejorar. Una buena arquitectura, en cambio, permite hacer cambios frecuentes y pequeños sin necesidad de reconstruir todo.

### Seguridad

la seguridad es proteger los datos y accesos del sistema contra ataques o mal uso. Un fallo de seguridad puede destruir la confianza de los usuarios y hasta acabar con un negocio, sin importar lo bien que funcione la app. No basta con añadir parches después; la seguridad debe pensarse desde la estructura misma del software.

Pensando en que la arquitectura busca reducir costos de cambio y dar flexibilidad al sistema, ¿es mejor diseñar una buena arquitectura desde el inicio o lanzar rápido y arreglar después?