**Atributos de calidad**

Desempeño – Calidad de respuesta

Disponibilidad -

Seguridad – Autenticación, autorización

Mantenibilidad

Microservicios es una arquitectura para el despliegue, no mejora el desempeño si no lo separo de otro componente

**Cuando hago una aplicación móvil puedo elegir:**

Tecnología hibrida, tecnología con generadores móviles o con tecnología nativa

Entregar aplicaciones robustas y de calidad

Pensamiento sistémico y estratégico

Los arquitectos elegimos los marcos de arquitectura

Las arquitecturas son evolutivas y dinámicas

**Diseño Top Down.-** Lo más general a lo mas especifico

Identificamos globalmente los riesgos que afecta al sistema

Establecemos Patrones Iniciales de Arquitectura

**Diseño Bottom Up.-** De lo más especifico a lo más general

**Diseño orientado a objetos.-** Parte desde el diagra de dominio de clases y reglas como:

Polimorfismo

Abstracción

Modularización

Herencia

Luego se hace un diagrma de casos de Uso

Luego un diagrama de iteracción

De secuencias

Tiempo de vida

Mensajes

**Desarrollo basado en componentes**: Consta de dos partes

Diseño para una aplicación

Diseño para la reutilización

¿Cómo lo codfican?

¿Cómo los despliegan?

¿Cómo Los mantiene?

¿Quién es el dueño del componente?

¿Cada cuanto se le hace mantenimiento al componente?

¿Cómo se les entrega a los otros desarrolladores?

Repositorios de componentes

**Diseño basado en heurísticas. –**

Es una practica que resulta de la experiencia, ojo no es una buena practica.

Se basa en una documentación implementada durante un proyecto anterior

**Diseño basado en patrones. -** Es un diseño que se basa en todos

Patron DAO, Patron Repository, Patron Storage, Patron Data Acces Componen, Patron Access Object, Patron DTO, Patron Data Assembler,

Patron MVC, basado al diseño arquitectónico

Tener documentados los diferentes patrones que se van a utilizar durante el desarrollo

Los tipo de Patrones son:

Patrones de arquitectura

Patrones de diseño

Patrones de lenguaje de codificación

Arquitectura Monolítica

Arquitectura Cliente Servidor

Arquitectura Tipo Spagetti

**Requerimientos No Funcionales a nivel de agilidad**

A nivel de historias de usuario

A nivel de criterios de aceptación

Criterios de Done

Requisito de la manera común

Tiene que ser

Medible

Lograble

No sea ambiguo

Completo

Trazable

Tester, pueden ser Pruebas Automáticas o Pruebas unitarias

Eliminar dependencias

Pruebas de Integración, ejemplo un módulo con una base de datos

**Calidad del Producto**

Tiempo – Alcance - Recursos

**Cambios**

Cambios por Bugs o defectos

Cambios por extensión de la aplicación

Cambios por flexibilización de la aplicación

La arquitectura es un conjunto de decisiones que va a permitir satisfacer una necesidad

Y todas las decisiones tomadas se convierten en su conjunto en una estrategia de arquitectura

Una estrategia de arquitectura son un conjunto de Tácticas.

Es decir una táctica es una decisión de una arquitectura especifica

Todas esta tácticas dependen de los requisitos no funcionales

Concerns son los intereses y preocupaciones.