

# PROCESO CON ENFOQUE ADD Y CLEAN ARCHITECTURE



Jesús Estenllos Loaiza Serrano 2313021

Desarrollo de Software III

Docente: Ing. Álvaro Salazar

Grupo 50

Sede Tuluá

17/06/2025

### 1. ¿Qué es Attribute-Driven Design (ADD) y cuál es su propósito en el diseño de software?

**Attribute-Driven Design (ADD)** es un método estructurado para diseñar arquitecturas de software basándose en los **atributos de calidad** del sistema (como rendimiento, disponibilidad, seguridad, escalabilidad, etc.), así como en **requisitos funcionales y restricciones**.

El propósito de ADD es tomar decisiones arquitectónicas que aseguren que la arquitectura cumpla con los objetivos del sistema desde el principio del desarrollo.

### 2. ¿Cómo se relaciona ADD con Clean Architecture en el proceso de diseño de sistemas?

ADD define **qué decisiones** arquitectónicas deben tomarse en función de los atributos de calidad. **Clean Architecture**, por su parte, proporciona un **modelo de capas** que guía **cómo estructurar** el código para mantener la independencia entre lógica de negocio, infraestructura y frameworks.

ADD ayuda a establecer los requisitos y prioridades arquitectónicas, mientras que Clean Architecture los implementa de manera ordenada y sostenible.

### 3. ¿Cuáles son los pasos principales del método ADD para definir una arquitectura de software?

1. **Recolección de requisitos:** funcionales, no funcionales y restricciones.
2. **Selección de módulos y patrones arquitectónicos** basados en atributos de calidad.
3. **Asignación de responsabilidades** a módulos y componentes.
4. **Identificación de relaciones** entre componentes.
5. **Refinamiento de módulos** y evaluación de decisiones previas.
6. **Validación contra atributos de calidad** y posibles escenarios.
7. **Documentación de decisiones arquitectónicas.**

### 4. ¿Cómo se identifican los atributos de calidad en ADD y por qué son importantes?

Se identifican mediante:

- Análisis de requisitos del cliente.
- Escenarios de uso críticos.
- Consultas a stakeholders clave (usuarios, clientes, negocio, devs).

Son importantes porque **guían las decisiones arquitectónicas** y garantizan que el sistema no solo cumpla funciones, sino que lo haga con características como rendimiento adecuado, facilidad de mantenimiento o seguridad robusta.

### 5. ¿Por qué Clean Architecture complementa ADD en la implementación de una solución?

Porque mientras ADD se enfoca en el **qué y por qué** de la arquitectura (atributos y decisiones), Clean Architecture se enfoca en el **cómo**, ofreciendo:

- Separación clara de responsabilidades.
- Independencia de frameworks y bases de datos.
- Facilidad para aplicar pruebas, cambiar tecnologías, y escalar el sistema.

**6. ¿Qué criterios se deben considerar al definir las capas en Clean Architecture dentro de un proceso ADD?**

- **Aislamiento de lógica de negocio** respecto a frameworks o bases de datos.
- **Facilidad de prueba y mantenimiento.**
- **Cumplimiento de atributos de calidad** (por ejemplo, capas separadas pueden mejorar seguridad y escalabilidad).
- **Dependencias unidireccionales** (de afuera hacia adentro).
- **Adaptabilidad a cambios tecnológicos.**

**7. ¿Cómo ADD ayuda a tomar decisiones arquitectónicas basadas en necesidades del negocio?**

ADD obliga a considerar los **atributos de calidad que más valoran los stakeholders**, y a diseñar la arquitectura para satisfacerlos. Esto alinea la estructura del software con los **objetivos estratégicos del negocio**, como mejorar la experiencia de usuario, reducir tiempo de respuesta o permitir despliegues frecuentes.

**8. ¿Cuáles son los beneficios de combinar ADD con Clean Architecture en un sistema basado en microservicios?**

- **Alineación con requisitos de negocio y calidad** desde el diseño.
- **Independencia entre servicios**, mejorando el mantenimiento y escalabilidad.
- **Claridad en los límites de contexto**, lo cual es clave en microservicios.
- **Facilidad de prueba e implementación continua.**
- **Robustez frente a cambios tecnológicos**, gracias a la separación de capas.

**9. ¿Cómo se asegura que la arquitectura resultante cumpla con los atributos de calidad definidos en ADD?**

- **Escenarios de validación:** se evalúan contra atributos de calidad con pruebas, revisiones o simulaciones.
- **Revisión arquitectónica periódica.**

- **Prototipos (spikes)** para probar decisiones críticas.
- **Monitoreo y métricas** en ejecución para validar rendimiento, disponibilidad, etc.

**10. ¿Qué herramientas o metodologías pueden ayudar a validar una arquitectura diseñada con ADD y Clean Architecture?**

- **ATAM (Architecture Tradeoff Analysis Method).**
- **ADR (Architectural Decision Records).**
- **Modelado con C4 o UML.**
- **Pruebas automatizadas** por capa.
- **Herramientas de análisis estático** (SonarQube, ArchUnit).
- **Simulación de cargas** (JMeter, Gatling).
- **Monitoreo de atributos en producción** (Prometheus, Grafana).