

# Workflow de AI-Driven Development

---

- Agentes con habilidades realizando tareas siguiendo especificaciones.

## Programación Inteligente

---

| Por [Alberto Basalo@AICode.Academy](mailto:AlbertoBasalo@AICode.Academy)

## 4.1. Análisis de requerimientos y arquitectura del sistema

---

- El análisis y la arquitectura son la **base sólida** del desarrollo.
- Proporcionan el **contexto necesario** para que la IA genere código adecuado.
- Invertir tiempo aquí **ahorra tiempo** en fases posteriores.

## 4.2. SDD, especificaciones funcionales detalladas

---

- Las especificaciones detalladas son el **contrato** del desarrollo.
- Permiten que la IA genere código **correcto desde el inicio**.
- Son el **punto de referencia** para verificación y validación.

## 4.3. Ramas, planes y tareas de codificación

---

- La planificación convierte especificaciones en **trabajo ejecutable**.
- Las tareas pequeñas y controlables permiten **verificación continua**.
- El flujo estructurado mantiene la **calidad y el control** del desarrollo.

## 5.1. Generar código es la parte fácil

---

- La generación de código se vuelve una **commodity**.
- El valor está en **definir QUÉ hacer** (specs) y **guiar CÓMO hacerlo** (skills).
- La eficiencia aumenta, pero nunca a costa de la calidad.

## 5.2. Verificar que funciona es la parte difícil

---

- La **verificación** consume más tiempo que la codificación.
- Es la **red de seguridad** necesaria para aprovechar la velocidad de la IA.
- Las pruebas del código son el punto de partida de un producto mantenible.

## 5.3. Documentar los cambios es la parte crítica

---

- La documentación **actualizada** es clave para la mantenibilidad.
- Saber *Qué he hecho y por qué*.
- Documentar es **invertir en sostenibilidad**.

# Conclusión

---

- El **análisis** y las **especificaciones** guían a la IA.
- La **codificación** es rápida, pero requiere **verificación**.
- La **documentación** asegura la **sostenibilidad** del proyecto.