

Línea Ares — Estándar de profundidad de contenido

ROBOTSchool · Documento maestro (fuente única) Versión 0.1 —
Junio 2026

El problema que resuelve este documento: los libros **dicen qué hacer pero no enseñan a hacerlo** — son un mapa, no un maestro. Aquí se define qué significa **llevar de la mano** en cada hilo, con criterios observables, una **plantilla de andamiaje** del microproyecto y una **lista de chequeo** de calidad. Es la vara contra la que se mide (y se reescribe) todo contenido Ares.

1. El principio: cada paso explica, muestra, anticipa y verifica

Una instrucción profunda nunca es una orden suelta. Siempre tiene cuatro capas:

1. **Explica el porqué** — para qué sirve este paso / concepto.
2. **Muestra el cómo** — la acción concreta, real, hacible a esa edad (una acción por paso).
3. **Anticipa el error** — qué suele salir mal y cómo notarlo/arreglarlo.
4. **Verifica** — “¿cómo sabes que quedó bien?” (un checkpoint observable).

Regla de oro: **un concepto nuevo a la vez**, de lo **concreto a lo abstracto**, con la **definición en el flujo** (1 frase + ejemplo + uso), nunca como glosario.

Prueba rápida de profundidad: tapa la imagen y dale el texto a un niño de la edad objetivo. ¿Podría hacerlo sin que el docente complete los huecos? Si no, falta profundidad.

2. Estándar por hilo (criterios observables)

2.1 Conceptual (pensamiento)

- Ancla en algo **concreto y cotidiano** antes de nombrar el término.
- **Define en el flujo:** una frase + ejemplo + uso inmediato.
- Hace **predecir** antes de revelar (“¿qué crees que pasará si...?”).
- Conecta el concepto con el **ODS** (por qué importa en la vida real).
- Cierra con una **verificación de comprensión** (aplicar, no memorizar).

2.2 Programación

- **Primero el algoritmo** (palabras / diagrama / pseudocódigo) **antes** del código.
- **PRIMM:** muestra un programa que funciona → predecir → correr → investigar → modificar → crear.
- **Explica cada bloque/línea nueva** (qué hace y por qué), no solo la lista.
- Incluye un **trazado** (seguir el programa a mano al menos una vez).
- Incluye una **depuración:** un error típico y cómo aislarlo.
- El editor/actividad **nunca viene resuelto**.

2.3 Construcción / robotización

- Pasos **accionables a la edad:** un verbo concreto, una acción por paso.
- Cada conexión (eléctrica/mecánica): **qué va dónde + cómo verificar** que quedó bien.
-

Figura de referencia en cada paso clave (marcar [INSERTAR FIGURA]).

- **Errores comunes** y cómo evitarlos/arreglarlos (p. ej. LED al revés no enciende).
- **Seguridad** cuando aplica (corriente, herramientas, corte).
- **Checkpoint** de funcionamiento (“¿cómo sabes que funcionó?”).

2.4 Diseño

- El diseño como **proceso**, no decoración: necesidad → idea → boceto → criterio → prueba.
- **Criterios de diseño explícitos** (firmeza, tamaño, legibilidad, ergonomía...).
- Para esquemático/CAD: **qué representa cada símbolo** y por qué importa.
- **Iteración**: “mejora una cosa de tu diseño y di por qué”.
- Conexión **forma** ↔ **función** (el diseño sirve al objetivo del ODS).

2.5 Transversal (convergencia · IA · gamificación)

- Hace visible cómo el hilo **se conecta con los otros** (no en silos).
- Un **momento de IA** cuando aporta (explicar, comparar, generar idea), con criterio.
- **Gamificación**: el paso emite un **hito** + una **insignia**.

3. Plantilla de andamiaje de un microproyecto profundo

Todo microproyecto se reescribe siguiendo esta anatomía (no todos los momentos pesan igual, pero el orden se mantiene):

#	Momento	Qué contiene
1	¿Para qué? (gancho)	Conecta con el reto y el ODS: por qué este paso importa
2	Concepto en el flujo	Predecir → definir (1 frase + ejemplo) → usar
3	Diseña antes de construir	Boceto/esquema + criterio de diseño explícito
4	Construye paso a paso	Cada paso: porqué + cómo + figura + error común + checkpoint
5	Programa	Algoritmo → PRIMM → explica cada línea → traza → depura
6	Integra y prueba	Júntalo, pruébalo: “¿funcionó?” con evidencia
7	Convergencia + IA + reflexión	Cómo se unieron los hilos; uso opcional de IA; pregunta de reflexión
8	Gamificación + comprensión	Hito + insignia; preguntas que verifican comprensión

4. Lista de chequeo de calidad (autochequeo antes de publicar)

Un microproyecto está listo **solo si** cumple:

- ¿Un niño de la edad podría hacerlo **sin** que el docente complete los huecos?
- ¿Cada concepto nuevo está **definido en el flujo** con ejemplo?
- ¿La programación va **algoritmo** → **código**, con explicación, trazado y un bug?
- ¿Cada paso de construcción tiene **cómo + error común + checkpoint**?
- ¿El diseño es **proceso con criterio**, no una línea decorativa?

- ¿Se ve la **convergencia** de los hilos y hay **hito + insignia**?
 - ¿Hay **figuras** marcadas donde se necesitan?
 - ¿El lenguaje es **apropiado a la edad**, sin jerga innecesaria?
-

5. Cómo se usa este estándar

1. Es la **vara** para auditar cualquier libro existente (empezando por ODS 11).
2. Es la base del **skill de producción de contenido Ares** (lo encapsula para producir a profundidad de forma repetible).
3. Acompaña a la **plantilla de lección** (doc 20) y la **ruta** (doc 21): el estándar dice *qué tan hondo*, la plantilla dice *en qué orden*, la ruta dice *qué y cuándo*.

Fin del documento v0.1. Vara de profundidad para todo contenido Ares; se prueba primero en el MP1 de ODS 11 y luego se vuelve skill.