

# Línea Ares — Actividades de programación por libro

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

Una **actividad de programación por grado** (Transición → 11°), cada una como **historia jugable** anclada al ODS del libro, con el **entorno acorde a la banda**, la **gamificación variada** y el **punteo al código real**. En bandas 1–2 la historia es un cuento; en bandas 3–4 es una **situación real de comunidad**. Operativiza el modelo de los tres productos (doc 22, Acto 1 = software) y la ruta (doc 21). Regla heredada del prototipo: **el algoritmo nunca viene resuelto** — el niño lo construye.

## 1. Entornos por banda (según definición del equipo)

Banda	Grados	Entorno de programación	Forma de “escribir el código”
1 · Exploradores	Trans., 1°, 2°	<b>mBlock</b> (bloques; en Transición, modo ícono / ScratchJr)	Bloques
2 · Constructores	3°, 4°, 5°	<b>mBlock + C++ asomada</b> → inicio de Python	Bloques → primer texto
3 · Inventores	6°, 7°, 8°	<b>Python / MicroPython</b>	Texto

Banda	Grados	Entorno de programación	Forma de “escribir el código”
4 · Innovadores	9°, 10°, 11°	C y Python (firmware + lógica/datos/web)	Texto pleno, varios lenguajes

*Nota de diseño:* en Transición mBlock se usa en su modo más simple (bloques con íconos, pre-lectores). El **concepto** es el mismo en toda la banda; sube la escalera de abstracción (doc 21 §2).

## 2. Tipos de juego (paleta de gamificación)

Las actividades **no son todas iguales**: rotan entre estos tipos para mantener la motivación y entrenar distintas habilidades.

Tipo de juego	Qué entrena	Mecánica
<b>Recorrido / laberinto</b>	secuencia, bucle	llevar un personaje a una meta
<b>Recolección / clasificación</b>	bucle, condicional	juntar u ordenar objetos
<b>Encuentra el error (depuración)</b>	depuración, trazado	arreglar un programa roto
<b>Ordena el código (Parsons)</b>	secuencia, lógica	poner líneas/ bloques en orden
<b>Historia ramificada</b>	condicionales	decisiones cambian el final
<b>Línea de producción / cronometrado</b>	bucle, variable, eficiencia	hacer más con menos pasos
<b>Simulación de datos / panel</b>	listas, datos, funciones	leer datos y decidir/avisar
<b>Control de sistema</b>	clases, eventos asíncronos	manejar un dispositivo que reacciona

Tipo de juego	Qué entrena	Mecánica
Proyecto integrador (“jefe”)	todo	combinar lo aprendido a escala

**Gamificación transversal (doc 19 §10):** rangos (Explorador→Innovador), insignias, XP/niveles, misiones, “jefes” (los proyectos), rachas y **progreso personal** (no ranking tóxico).

### 3. Banda 1 · Exploradores — *entorno: mBlock* (bloques)

#### Transición · ODS 4 (Educación de calidad) — “Robi va a la escuela”

- **Historia:** Robi, un robot pequeñito, quiere llegar a su salón pero no sabe el camino. Ayúdalo dándole órdenes.
- **Concepto:** instrucción · **secuencia** (el orden importa).
- **Tipo de juego:** recorrido en cuadrícula 3×3 con flechas-bloque.
- **Reto:** arrastra las flechas (arriba/derecha) para que Robi llegue al salón. *Empieza vacío: el niño arma el camino.*
- **Gamificación:** estrellas por llegar; insignia “**Primer paso**”; niveles con caminos distintos.
- **Puente:** los bloques son de mBlock; con **Makey Makey**, tocar frutas se vuelve el control de Robi.

#### 1º · ODS 12 (Consumo responsable) — “Reco recicla”

- **Historia:** Reco recoge la basura del patio y debe llevar cada cosa a su cesto: reciclable u orgánico.
- **Concepto:** secuencia · **evento** (“cuando toco...”).
- **Tipo de juego:** recolección + clasificación.
- **Reto:** programa la ruta para recoger 3 objetos y soltarlos en el cesto correcto.
- **Gamificación:** puntos por clasificar bien, “**racha verde**”, insignia “Guardián del patio”.

- **Puente:** mBlock; Makey Makey como botones de los cestos.

## 2° · ODS 7 (Energía) — “Enciende el pueblo”

- **Historia:** al amanecer, Soli el sol debe encender las casas del pueblo, una por una.
- **Concepto:** evento · **bucle simple** (repite N).
- **Tipo de juego:** secuencia rítmica / “enciende”.
- **Reto:** usa **repetir** para “avanzar y encender” 4 casas sin escribirlo cuatro veces.
- **Gamificación:** barra de energía, ciclo día/noche, insignia “Amanecer”.
- **Puente:** mBlock; el bloque **repetir** = el primer bucle.

---

## 4. Banda 2 · Constructores — *entorno: mBlock + C++/inicio de Python*

### 3° · ODS 11 (Ciudades sostenibles) — “Tito ordena la ciudad”

- **Historia:** Tito, un camioncito, mantiene limpia y ordenada su ciudad, pero las calles tienen obstáculos.
- **Concepto:** **bucle + condicional** (“si hay obstáculo, gira”).
- **Tipo de juego:** recorrido con decisiones (ramifica según el camino).
- **Reto:** recorre todas las calles recogiendo, evitando obstáculos con una decisión. *Editor en blanco.*
- **Gamificación:** misiones por barrio, medalla “**Urbanista**”, mapa que se desbloquea.
- **Puente:** mBlock → **C++ asomada** (mostrar el `for / if` equivalente, solo leer).

### 4° · ODS 6 (Agua limpia) — “Gota y la huerta” (*prototipo ya construido*)

- **Historia:** la huerta de doña Rosa se seca; Gota debe regar las plantas con sed.
- **Concepto:** **bucle** (+ condicional “si está seca”).
- **Tipo de juego:** recolección / riego + reto de eficiencia.

- **Reto:** riega las 6 plantas; **reto estrella:** hacerlo en 4 líneas (obliga al bucle).
- **Gamificación:** reto estrella (eficiencia), insignia “Riego justo”.
- **Puente:** mBlock → **MicroPython** ( `for ... : avanzar(); regar()` ); en el Acto 3 mueve el robot real.

## 5° · ODS 9 (Industria e innovación) — “La fábrica de Mía”

- **Historia:** Mía maneja una banda transportadora y debe programar el brazo que empaca.
- **Concepto:** bucle condicional + variable (contador de cajas).
- **Tipo de juego:** línea de producción / cronometrado (eficiencia).
- **Reto:** empaca 10 cajas llevando la cuenta en una **variable**; no empacar de más.
- **Gamificación:** récord de cajas/min, niveles de fábrica, insignia “Ingeniera de procesos”.
- **Puente:** mBlock → **inicio de Python** ( `while` , variable contador).

## 5. Banda 3 · Inventores — *entorno: Python / MicroPython · situaciones de comunidad*

En esta banda la “historia” madura a **situación real**: un problema de la comunidad que se resuelve con código y datos. (Conecta con el producto de software del libro, doc 22.)

## 6° · ODS 13 (Acción por el clima) — “Estación climática del barrio”

- **Situación:** el barrio se inunda porque nadie sabe cuándo lloverá fuerte.
- **Concepto:** función + lista (serie de lecturas).
- **Tipo de juego:** simulación de datos / panel.
- **Reto:** escribe una **función** que recorra una **lista** de lecturas y avise cuando una supere el umbral de alerta.
- **Gamificación:** misiones de pronóstico, puntos por acierto, insignia “Guardián del clima”.

- **Puente: MicroPython** en Raspberry Pi Pico + sensor (lluvia/humedad).

### 7° · ODS 2 (Hambre cero) — “Reparte la cosecha”

- **Situación:** la huerta comunitaria cosechó y hay que repartir de forma justa entre las familias.
- **Concepto:** listas + bucle `for` + condicional (equidad).
- **Tipo de juego:** lógica / optimización.
- **Reto:** programa el reparto de N alimentos entre M familias de la forma más equitativa.
- **Gamificación:** niveles de “equidad”, progreso personal de eficiencia, insignia “Cosecha justa”.
- **Puente: Python** (luego escala a sensores de la huerta).

### 8° · ODS 12 (Consumo responsable) — “Cazadores de desperdicio”

- **Situación:** el colegio bota mucha comida y nadie sabe cuánta.
- **Concepto:** funciones + diccionario (intro a estructuras de datos).
- **Tipo de juego:** recolección de datos + dashboard.
- **Reto:** registra el desperdicio por día en un **diccionario**, suma el total y encuentra el peor día.
- **Gamificación:** retos semanales de reducción, racha, insignia “Cero desperdicio”.
- **Puente: Python/MicroPython** + báscula/registro.

---

## 6. Banda 4 · Innovadores — *entorno: C y Python · sistemas para la comunidad*

Aquí las situaciones son **sistemas** que atienden una necesidad real de la comunidad, con datos, red y a veces web. El producto de software ya es parte de un sistema (doc 22, banda 4).

### 9° · ODS 7 (Energía asequible) — “Monitor de energía del colegio”

- **Situación:** el colegio paga mucha luz y no sabe dónde se va.

- **Concepto: estructuras de datos + manejo de errores** ( `try/except` ).
- **Tipo de juego:** sistema con datos + dashboard.
- **Reto:** lee lecturas (protegidas ante fallos), calcula promedio y **alerta picos** de consumo.
- **Gamificación:** retos de optimización (bajar el consumo simulado), tabla de logros, insignia “Eficiencia”.
- **Puente: Python** (lógica/datos) + **C** (firmware ESP32 que mide).

### 10° · ODS 11 (Ciudades sostenibles) — “Semáforo inteligente del barrio”

- **Situación:** un cruce peligroso del barrio necesita un semáforo que reaccione a los peatones.
- **Concepto: clases/objetos + eventos asíncronos** (interrupción de sensor).
- **Tipo de juego:** control de sistema / simulación.
- **Reto:** crea una **clase Semáforo** con estados; que reaccione al sensor de peatón sin bloquearse.
- **Gamificación:** niveles de tráfico, misiones de flujo, insignia “Movilidad”.
- **Puente: C** (ESP32, tiempo real) + **Python** (lógica y simulación).

### 11° · ODS 2 + 11 (Proyecto integrador) — “Red de huertos del barrio (IoT)”

- **Situación:** varias familias tienen huertos urbanos; juntos pueden dar seguridad alimentaria si se coordinan.
- **Concepto: integración total** — clases + **red** + **SQL** + **dashboard**.
- **Tipo de juego:** proyecto integrador / “**jefe final**” multi-nodo.
- **Reto:** varios nodos (ESP32) reportan humedad/cosecha a un panel central; el sistema alerta y coordina el riego.
- **Gamificación:** el **jefe final** del programa Ares; portafolio completo; insignia máxima “**Innovador**”.
- **Puente: C** (nodos) + **Python** (servidor/lógica) + **web/SQL** (panel).

## 7. Cómo se produce cada actividad (formato común)

Cada actividad, en cualquier banda, se entrega en **tres piezas** (como Gota):

1. **Historia jugable** (pantalla, Academy): editor en blanco → el niño programa → el personaje/sistema actúa → puente al código real. *El algoritmo nunca viene resuelto.*
2. **Página impresa** (libro): historia + define-en-flujo + mundo para trazar a mano + líneas para escribir + tabla de trazado + puente + reto + QR.
3. **Aplicación al robot** (Acto 3): el mismo código mueve el dispositivo físico del proyecto.

Sigue la **plantilla de 12 momentos** del doc 20 y se evalúa con la rúbrica (doc 04 / 19 §8).

## 8. Resumen (matriz de actividades)

Grado	ODS	Historia / situación	Concepto	Tipo de juego	Entorno
Trans.	4	Robi va a la escuela	secuencia	recorrido	mBlock (ícono)
1°	12	Reco recicla	secuencia, evento	recolección/ clasificación	mBlock
2°	7	Enciende el pueblo	evento, bucle	secuencia rítmica	mBlock
3°	11	Tito ordena la ciudad	bucle, condicional	recorrido c/ decisiones	mBlock→C++
4°	6	Gota y la huerta	bucle	recolección/ eficiencia	mBlock→MicroPython
5°	9	La fábrica de Mía	bucle, variable	línea de producción	mBlock→Python
6°	13	Estación climática	función, lista	simulación de datos	MicroPython

Grado	ODS	Historia / situación	Concepto	Tipo de juego	Entorno
7°	2	Reparte la cosecha	listas, for, condicional	lógica/ optimización	Python
8°	12	Cazadores de desperdicio	función, diccionario	datos + dashboard	Python/MicroPython
9°	7	Monitor de energía	estructuras de datos, errores	sistema + dashboard	C + Python
10°	11	Semáforo inteligente	clases, eventos asíncronos	control de sistema	C + Python
11°	2+11	Red de huertos IoT	integración total	proyecto "jefe"	C + Python + web/ SQL

## 9. Pendiente

1. Construir las **historias jugables** (HTML/Academy) por grado, empezando por las bandas 1–2 (como Gota).
2. Diseñar la **página impresa** de cada actividad (como la de Gota).
3. Desarrollar la **lección de 12 momentos** de cada concepto (doc 20) usando estas actividades como juego ancla.
4. Definir el **arco narrativo global** que conecte a Robi, Reco, Soli, Tito, Gota, Mía... a lo largo de las bandas.
5. Programar el **set de niveles** de cada juego y su gamificación en la plataforma.

*Fin del documento v0.1. Operativiza la ruta (21) y el modelo de tres productos (22); cada actividad es el Acto 1 (software) de su libro.*