

# Contenido de Pensamiento Computacional, Algoritmos y Programación

## Banda Constructores (3° · 4° · 5°)

ROBOTSchool · Línea Ares · Versión 0.1 (borrador para edición) · Junio 2026 · *(salida del PROMPT MAESTRO)*

Banda Constructores: **Arduino + mBlock** (alterna: micro:bit + MakeCode). Es donde la programación toma estructura: **bucles, condicionales, variables** y la **asomada a funciones/C++**. Entorno de práctica: **Karel / Reeborg's World** (bloques). Principio: *primero el algoritmo, luego el código*. La habilidad se acumula del grado anterior.

# GRADO 3° - Proyecto: Semáforo inteligente (ODS 11)

**Foco:** del bloque a la lógica — **bucles** y **condicionales**.

## 1. Malla de habilidades (ISTE 1.5 · CSTA 1B-AP)

Competencia de PC	Qué se desarrolla en 3°
Descomposición	Separar una tarea en pasos y un programa en bloques
Patrones	Reconocer lo que se repite → <b>bucle</b>
Algoritmos	Secuencia + <b>repetición</b> + <b>decisión</b>
Depuración	Encontrar y corregir un error simple
Estructuras	Secuencia y eventos (repaso) · <b>bucles</b> (nuevo) · <b>condicionales</b> (nuevo) · variables (introducción)

## 2. Conceptos con ejemplo resuelto

**Bucle (repetición).** *Ejemplo cotidiano: dar 10 saltos.* En vez de escribir “salta” 10 veces:

```
repetir 10 veces:  
  saltar
```

*El robot dibuja un cuadrado:*

```
repetir 4 veces:  
  avanzar 50  
  girar 90°
```

**Condicional (decisión).** *Ejemplo: cruzar la calle.*

```
mirar a ambos lados  
SI viene un carro:  
  esperar  
SI NO:  
  cruzar
```

**Aplicado al semáforo:** el **ciclo** de luces es un bucle; el **botón del peatón** es un condicional (ver §9).

### 3. Banco de ejercicios (con solución)

#	Concepto	Enunciado	Solución
1	Secuencia	Ordena los pasos para hacer una limonada (desordenados).	exprimir → echar agua → echar azúcar → revolver → servir
2	Bucle	Algoritmo para dar 5 vueltas a la cancha.	repetir 5: correr una vuelta
3	Bucle	¿Cómo le dices a un robot que dibuje un cuadrado?	repetir 4: avanzar, girar 90°
4	Condicional	Algoritmo “si llueve, lleva sombrilla”.	SI está lloviendo: llevar sombrilla / SI NO: salir normal
5	Condicional	Diagrama de flujo “si la nota $\geq 3$ aprobó, si no refuerzo”.	rombo ¿nota $\geq 3$ ? → Sí: “aprobó” / No: “refuerzo”
6	Bucle + condicional	Pseudocódigo del ciclo del semáforo.	por siempre: verde(4s); amarillo(2s); rojo(4s)
7	Bucle (Karel)	Lleva a Karel hasta el muro.	mientras (no hay muro al frente): avanzar
8	Variable	Lleva un marcador: suma 1 cada gol.	fijar puntos=0; cada gol: puntos = puntos + 1
9	Depuración	Un robot gira 60° para hacer un cuadrado y no cierra. ¿Qué corriges?	girar 90°, no 60°
10	Reto	Algoritmo para que el robot recoja 3 pelotas.	repetir 3: avanzar hasta pelota, recoger

#### 4. Actividades desenchufadas

- **Robot humano + laberinto:** marcar un laberinto en el piso; un niño da instrucciones paso a paso a otro (secuencia y depuración).
- **Tarjetas de bucle:** representar “repetir N veces” con una tarjeta que envuelve varios pasos.
- **“Simón dice... SI”:** juego de condicionales (“si tienes zapatos rojos, salta”).

#### 5. Retos en Reeborg’s World (bloques)

1. **Llega a la meta** (secuencia simple).
2. **El laberinto** — usa `mientras` para avanzar hasta el muro (bucle).
3. **Recolector** — recoge todos los objetos del camino (bucle + condición).
4. **El cuadrado** — haz que Reeborg recorra un cuadrado (bucle con giro).

#### 6. Hitos de 3º (es capaz de...)

- Ordenar y ejecutar una secuencia de pasos.
- Usar un **bucle** para repetir acciones.
- Usar un **condicional** para decidir.
- Encontrar y corregir un **error simple**.

#### 7. Indicadores de rúbrica en 3º

Criterio	1	2	3	4
Descomposición	no separa pasos	con ayuda	separa en bloques claros	bloques reutilizables
Lógica (bucle/condicional)	no la usa	con errores	la usa bien	la combina (bucle+condición)
Representación	no dibuja	parcial	diagrama antes de codear	diagrama claro y completo
Depuración	no halla el error	con ayuda	halla y corrige	anticipa el error
Autonomía	depende	con guía	trabaja solo	ayuda a otros

## 8. Gamificación de 3°

- **Insignias:** “Repetidor” (domina bucles) · “Decididor” (condicionales) · “Cazador de bugs I”.
- **Misiones:** los 4 retos de Reeborg.
- **Jefe del grado:** programar el **semáforo** completo.

## 9. Conexión con el proyecto

Antes de armar el semáforo, el estudiante **dibuja el diagrama de flujo** del ciclo de luces y el **condicional** del botón peatonal (paso “Diseña tu algoritmo” del libro de 3°). El código es la traducción de ese diagrama.

---

# GRADO 4° - Proyecto: Riego automático (ODS 6)

---

**Foco (ampliación): variables y umbral** — decidir con un número que cambia (sensor).

## 1. Malla

Repaso de bucles/condicionales. **Nuevo:** variables que guardan la lectura de un sensor; **comparaciones** (<, >); condicional con **umbral**.

## 2. Conceptos con ejemplo resuelto

**Variable.** *Una caja con un número que puede cambiar.* Ej.: el marcador de un partido (puntos). **Umbral y comparación.** *Ejemplo: termostato.*

```
fijar limite = 30
SI temperatura > limite:
    prender ventilador
SI NO:
    apagar ventilador
```

### Aplicado al riego:

```
fijar umbral = 400
por siempre:
    fijar humedad = leer sensor
    SI humedad < umbral: # tierra seca
        regar
    SI NO:
        no regar
```

### 3. Banco de ejercicios (con solución)

#	Concepto	Enunciado	Solución
1	Variable	Lleva el puntaje de un juego y muéstralo.	<code>puntos=0; al anotar: puntos=puntos+1; mostrar puntos</code>
2	Umbral	“Si la temperatura pasa de 30, prende el ventilador”. Diagrama de flujo.	rombo ¿temp>30? → Sí: ventilador ON / No: OFF
3	Umbral	Pseudocódigo del riego (humedad < umbral).	ver §2
4	Comparación	¿Qué umbral de humedad usarías para un cactus vs una lechuga? Justifica.	cactus umbral bajo (riega poco); lechuga alto
5	Variable + bucle	Cuenta cuántas veces se regó en el día.	<code>riegos=0; cada riego: riegos=riegos+1</code>
6	Depuración	El riego nunca para de regar. El umbral está en 1000 (máx). ¿Qué pasa?	umbral muy alto: siempre “seco”; bajarlo
7	Reeborg	Usa una variable para contar los pasos hasta el muro.	<code>pasos=0; mientras no muro: avanzar; pasos+=1</code>
8	Reto	“Avisa con un LED si la planta lleva 3 días sin agua”.	variable <code>días</code> ; SI días≥3: encender LED

### 4. Desenchufados

- **El termostato humano:** con tarjetas de temperatura, los niños deciden “ventilador ON/OFF” según un umbral.
- **La caja-variable:** una caja física donde guardan/cambian un número (puntos, conteo).
- **¿Mayor o menor?:** juego de comparaciones con cartas numéricas.

### 5. Retos en Reeborg’s World

1. **Contador de pasos** (variable que aumenta).
2. **Recoge solo si hay** (condición sobre el entorno).

3. **Patrullaje** (repetir con condición de parada por variable).

## 6. Hitos de 4°

- Usa **variables** para guardar y cambiar un valor.
- Decide con un **umbral** (comparaciones).
- Lee un **sensor** y actúa según su valor.
- Ajusta el umbral para mejorar el resultado (depuración con sentido).

## 7. Indicadores de rúbrica en 4°

Igual estructura que 3°, subiendo la vara: en “Lógica”, *Logrado* = usa variable + umbral correctamente; *Destacado* = elige y justifica el umbral.

## 8. Gamificación de 4°

- **Insignias:** “Maestro de variables” · “Sensor & umbral” · “Cazador de bugs II”.
- **Misiones:** los retos de Reeborg con variable.
- **Jefe del grado:** programar el **riego automático**.

## 9. Conexión con el proyecto

Antes de construir el riego, el estudiante escribe el **pseudocódigo** con su umbral y lo prueba en simulación; el libro de 4° trae ese paso y la pregunta “¿qué umbral elegiste y por qué?”.

---

# GRADO 5° - Proyecto: Brazo / mano robótica (ODS 9)

---

**Foco (ampliación): funciones** (agrupar pasos con nombre) y **secuencias de movimiento**; **asomada a C++**.

## 1. Malla

Repaso de variables/condicionales/bucles. **Nuevo: funciones** (definir una rutina y reutilizarla); secuencias precisas (ángulos del servo); ver el mismo programa en **C++** (asomada, sin exigir dominio).

## 2. Conceptos con ejemplo resuelto

**Función.** *Un grupo de pasos con un nombre, para reutilizarlo.* Ej.: “preparar la mesa”.

```
definir poner_puesto():  
    poner plato  
    poner cubiertos  
    poner vaso  
  
# usarla 4 veces:  
repetir 4: poner_puesto()
```

**Aplicado al brazo (secuencia de ángulos):**

```
definir agarrar():  
    mover servo a 10° # cerrar  
    esperar 1 s  
definir soltar():  
    mover servo a 90° # abrir  
  
agarrar()  
mover el brazo  
soltar()
```

### 3. Banco de ejercicios (con solución)

#	Concepto	Enunciado	Solución
1	Función	Define una función “saludar” que encienda un LED 3 veces.	<code>definir saludar(): repetir 3: LED on/ off</code>
2	Función	Convierte “dibujar un cuadrado” en una función y dibuja 3.	<code>definir cuadrado(): repetir 4: avanzar, girar90; luego repetir 3: cuadrado()</code>
3	Secuencia	Escribe la secuencia “agarrar → mover → soltar” con ángulos.	ver §2
4	Descomposición	Divide “el brazo recoge un objeto y lo deja en una caja” en funciones.	<code>agarrar(); girar(); soltar()</code>
5	Función + variable	Función “abrir_grados(n)” que reciba el ángulo.	<code>definir abrir(n): mover servo a n</code>
6	C++ asomada	Escribe en bloques y observa el equivalente en C++ de “encender pin 13”.	<code>digitalWrite(13, HIGH);</code>
7	Depuración	El brazo suelta antes de mover. Ordena la secuencia.	agarrar → mover → soltar
8	Reto	Programa una rutina que salude, agarre y guarde un objeto usando 3 funciones.	combinar <code>saludar(); agarrar(); guardar()</code>

### 4. Desenchufados

- **La receta-función:** escribir una rutina (“hacer un sándwich”) como función y “llamarla” varias veces.
- **Órdenes con nombre:** un niño define una “jugada” (secuencia) y el grupo la ejecuta al nombrarla.
- **Ángulos con el brazo:** estimar ángulos (0°, 90°, 180°) con el propio brazo.

## 5. Retos en Reeborg's World (con funciones)

1. Define `girar_derecha()` (si el entorno solo trae girar izquierda) y úsala.
2. `subir_escalon()` como función y repítela para subir una escalera.
3. **Misión modular:** resuelve un mapa usando al menos 2 funciones propias.

## 6. Hitos de 5°

- Define y **reutiliza funciones**.
- Programa una **secuencia precisa** de movimientos (ángulos).
- Reconoce el **equivalente en C++** de un bloque.
- Descompone un problema en **funciones**.

## 7. Indicadores de rúbrica en 5°

En “Descomposición”, *Destacado* = resuelve con funciones reutilizables; en “Representación”, incluye el pseudocódigo con funciones antes de codear.

## 8. Gamificación de 5°

- **Insignias:** “Arquitecto de funciones” · “Asomado a C++” · “Cazador de bugs III”.
- **Misiones:** los retos modulares de Reeborg.
- **Jefe del grado:** programar el **brazo robótico** con funciones (agarrar/soltar).

## 9. Conexión con el proyecto

El brazo se piensa como **funciones** (`agarrar`, `soltar`) antes de programar; el libro de 5° trae ese diseño y la pregunta “¿por qué conviene agrupar los pasos en una función?”.

## Cierre de la banda

Al terminar Constructores, el estudiante **piensa con bucles, condicionales, variables y funciones, representa** sus algoritmos (diagrama/pseudocódigo), **practica** en Reeborg y **programa** robots reales — listo para el salto a MicroPython en Inventores (con BIPES de puente).

Generado con `PROMPT_desarrollo_PC_programacion.md`. Pendiente: ampliar el banco a 12–15 ejercicios por grado y crear el arte de las insignias.