

Kit del tutor

Cómo enseñar programación de 6 a 18 años sin depender de la
plataforma

Versión: manual autónomo · Guía para adultos

Para: padres, madres, monitores de ludoteca, entrenadores y docentes

Qué es este documento

Este **kit del tutor** no enseña un lenguaje concreto: explica **cómo organizar** las clases usando los cinco manuales autónomos de la Academia:

Manual	Archivo PDF	Ideal para empezar si...
HTML + CSS	<code>html-css.pdf</code>	Quieres resultados visuales rápido
Python	<code>python.pdf</code>	Prefieres lógica sin navegador
JavaScript	<code>javascript.pdf</code>	Ya saben algo de HTML
TypeScript	<code>typescript.pdf</code>	Ya dominaron JavaScript básico
SQL	<code>sql.pdf</code>	Les gustan los datos y las tablas

Cada manual tiene **3 módulos por edad** (A: 6–12 · B: 12–14 · C: 14–18) con lecciones, ejercicios, proyectos y **soluciones al final**.

Mensaje clave: No hace falta ser programador profesional. Hace falta paciencia, curiosidad y dejar que el alumno **escriba el código con sus manos**.

Antes de la primera sesión

Checklist del espacio

- ☐ Ordenadores o portátiles (1 por alumno o 1 cada 2, máximo)
- ☐ Conexión a internet (opcional si todo está instalado)
- ☐ Proyector o pantalla grande (recomendado)
- ☐ Pizarra o papel para dibujar «cajas» de variables
- ☐ Lista de alumnos con edad y módulo asignado (A/B/C)

Checklist técnico por manual

Manual	Qué instalar	Comprobar que funciona
HTML + CSS	Navegador + editor (VS Code)	Abrir un <code>.html</code> local
Python	Python 3.10+ (Thonny recomendado)	<code>print("Hola")</code>
JavaScript	Navegador + editor	Consola F12 en una página

TypeScript	Node.js 18+ + <code>npm install tsx</code>	<code>npx tsx hola.ts</code>
SQL	SQLite o DB Browser	Ejecutar <code>SELECT 1;</code>

Ludoteca sin permisos de instalación: usa alternativas online indicadas en cada manual (Replit, TypeScript Playground, sqliteonline.com).

Reglas de oro del aula

1. **Un concepto por sesión.** No corras si no está asimilado.
2. **Escribir > leer.** El alumno debe teclear; tú guías, no sustituyes.
3. **Los errores son normales.** Celébralos: «¿Qué crees que dice este mensaje?»
4. **Pregunta antes de corregir:** «¿Qué hace esta línea?»
5. **Sesiones cortas** para 6–12 (30–45 min); más largas para 14–18 (hasta 60 min).

Rutas de aprendizaje recomendadas

Ruta A · «Primera web» (visual, muy motivadora)

1. HTML + CSS — Módulo A
2. JavaScript — Módulo A
3. HTML + CSS — Módulo B
4. JavaScript — Módulo B
5. Proyecto web libre (combinar los dos)

Edades: 8–14 ideal.

Ruta B · «Lógica primero» (sin navegador al inicio)

1. Python — Módulo A
2. Python — Módulo B
3. SQL — Módulo A (datos como tablas)
4. JavaScript — Módulo A (puente a la web)
5. Python — Módulo C o SQL — Módulo B

Edades: 7–16.

Ruta C · «Full stack junior» (14–18)

1. HTML + CSS — Módulos B y C
2. JavaScript — Módulos B y C

3. TypeScript — Módulos A y B
4. SQL — Módulos B y C
5. Proyecto integrador (web + datos)

Edades: 14–18 con experiencia previa.

Ruta D · «Ludoteca express» (10 sesiones × 45 min)

Sesión	Manual	Contenido
1–2	HTML+CSS A	Primera página y colores
3–4	HTML+CSS A	Proyecto tarjeta
5–6	JavaScript A	Botones y contador
7	Python A	<code>print</code> y variables
8	SQL A	<code>SELECT</code> y <code>WHERE</code>
9	Repaso libre	El alumno elige
10	Mini feria	Cada uno enseña su proyecto

Cómo dirigir una sesión (plantilla)

Estructura de 45 minutos

Minutos	Actividad	Tu rol
0–5	Repaso oral del día anterior	Preguntas rápidas
5–15	Leer juntos el concepto del manual	Explicar con analogías
15–30	Código en vivo (tú proyectas, ellos escriben)	No copies por ellos
30–40	Ejercicio del manual	Circular y ayudar
40–45	«¿Qué aprendí hoy?» (1 frase por alumno)	Anotar en pizarra

Frases útiles

- «No lo sé, lo averiguamos juntos.»
- «El error nos está dando una pista.»
- «¿Qué cambiarías para que funcione?»

- «Genial, ahora prueba a romperlo a propósito.»

Señales de que vas demasiado rápido

- Copian sin entender.
- No saben explicar su código.
- Frustración constante sin pausas.
- Piden que tú escribas todo.

Solución: vuelve a la lección anterior o divide el ejercicio en micro-pasos.

Módulos por edad: qué esperar

Módulo A · 6–12 años

- Analogías concretas (cajas, listas, botones).
- Mucho refuerzo visual.
- Sesiones de 30–45 min máximo.
- Celebrar cada pequeño logro.

Manuales más adecuados: HTML+CSS A, Python A, JavaScript A, SQL A.

Módulo B · 12–14 años

- Introducen lógica (`if` , bucles, funciones).
- Pueden trabajar más autónomos.
- Comparaciones entre lenguajes («en Python es `for` , en JS también»).

Manuales: cualquiera en módulo B según la ruta elegida.

Módulo C · 14–18 años

- Proyectos más largos (1–3 sesiones).
- Diseño de solución antes de codear.
- Documentación y comentarios en el código.

Manuales: módulo C de JS, TS, SQL o HTML+CSS según ruta.

Seguimiento del progreso (sin plataforma)

Ficha de alumno (imprimible)

Nombre: _____ Edad: ____ Módulo: A / B / C

Fecha	Manual	Lección	¿Completada?	Notas
-----	-----	-----	-----	-----
			☐	

Criterios simples de «lección completada»

- ☐ El alumno ejecutó el ejemplo del manual sin ayuda para escribir.
- ☐ Completó la práctica (aunque con errores corregidos).
- ☐ Puede explicar en una frase qué hace su programa.

Proyectos finales por trimestre

Sugiere **un proyecto demostrable** cada 8–12 sesiones:

- Ludoteca: feria de proyectos con familias.
- Centro deportivo: póster digital con HTML o contador de puntos en JS.
- Casa: enseñar a un familiar lo que aprendió.

Grupos mixtos (varias edades)

Situación	Estrategia
8 y 13 años juntos	Mismo manual, módulos distintos (A y B)
Principiantes y avanzados	Mismo tema, ejercicios extra para avanzados
Solo 2 alumnos	Turnos de «profesor»: uno explica al otro
1 solo alumno	Sigue el manual al ritmo del alumno; más proyectos libres

Problemas frecuentes y soluciones

«No tengo internet»

- HTML/JS: funciona offline con archivos locales.
- Python: IDLE/Thonny offline.
- SQL: SQLite offline.
- TypeScript: tras instalar Node, offline con `npx tsx`.

«El código no hace nada»

1. ¿Guardó el archivo?
2. ¿Recargó el navegador (F5)?
3. ¿Miró la consola (F12)?
4. ¿Ejecutó el comando correcto (`python` , `tsx` , `sqlite3`)?

«Un alumno va mucho más rápido»

- Ejercicios extra del manual (prácticas sin solución inmediata).
- Que ayude a un compañero (con supervisión).
- Módulo siguiente o proyecto libre documentado.

«Un alumno se bloquea»

- Reduce el ejercicio a una sola línea nueva.
- Dibuja en papel antes de codear.
- Cambia de manual temporalmente (más visual → HTML).

Uso opcional de la plataforma Academia

Los manuales funcionan **solos**. La plataforma es un **refuerzo**:

Recurso	URL	Cuándo usarlo
Editor	<code>/student/editor</code>	Practicar con plantillas
Cursos	<code>/courses/*-starter</code>	Gamificación y seguimiento
Biblioteca PDF	<code>/student/guides</code>	Descargar manuales

Recomendación: primeras 4–6 sesiones **sin plataforma** para que entiendan que programar no depende de una web concreta.

Plan trimestral ejemplo (ludoteca, 12 sesiones)

Grupo: 9–13 años · 12 sesiones de 45 min · Ruta A

#	Manual	Lección / actividad
1	HTML+CSS	A1–A2: primera página

2	HTML+CSS	A3: imágenes y colores
3	HTML+CSS	Proyecto A: tarjeta personal
4	JavaScript	A1–A2: consola y variables
5	JavaScript	A3–A4: botón y contador
6	JavaScript	Proyecto A: tarjeta interactiva
7	HTML+CSS	B1: listas y enlaces
8	JavaScript	B1: condicionales
9	Python	A1: primer <code>print</code> (cambio de aires)
10	SQL	A1: <code>SELECT</code> (datos como tabla)
11	Libre	Mejorar proyecto favorito
12	Feria	Presentaciones 3 min por alumno

Materiales a imprimir

1. Esta guía (kit del tutor) — para el adulto.
2. Ficha de seguimiento por alumno (sección anterior).
3. Chuleta de la sesión (copiar del manual del día).
4. **No imprimas las soluciones** hasta después del ejercicio.

Contacto y recursos

- Repositorio y PDFs: biblioteca en `/student/guides`
- Regenerar PDFs localmente: `cd academy-platform && npm run guides:build`
- Cada manual incluye apéndice de errores y glosario

Recuerda: Tu objetivo no es formar programadores en 10 sesiones, sino **despertar curiosidad**, confianza y pensamiento lógico. El resto llegará con el tiempo.