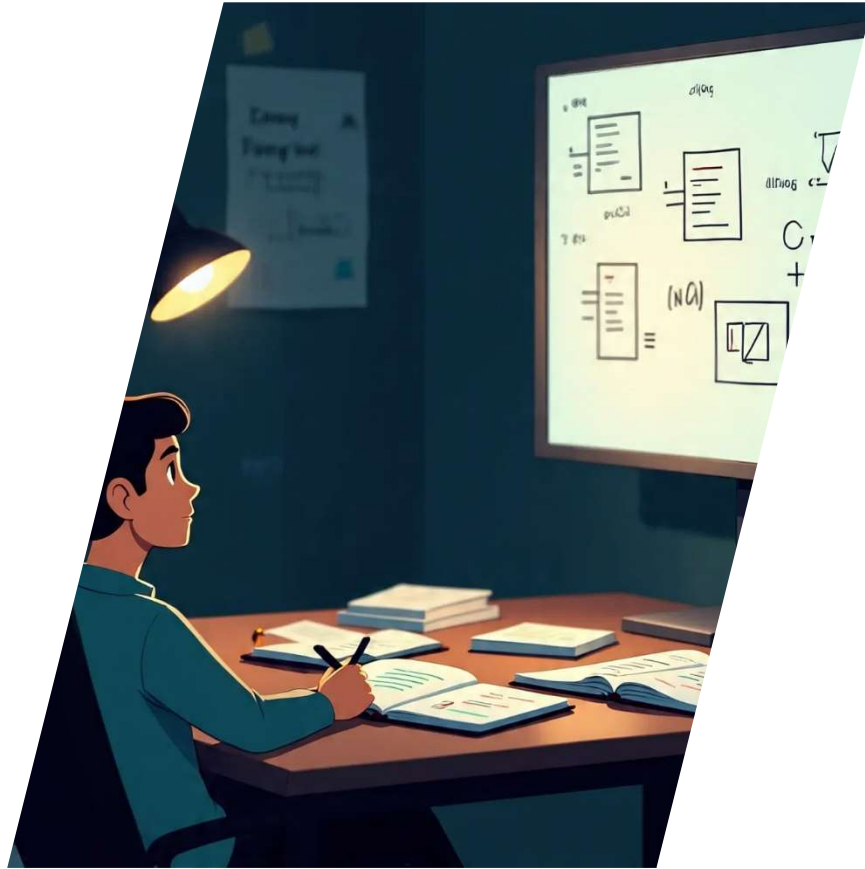


Pensamiento de Programador: Explorando la Lógica de la Programación



Logotipo Logic Programming

Universidad Tecnológica de Pereira
Luis Eduardo Muñoz Guerrero
Agosto 2025



Tabla de Contenido

1. Introducción
2. ¿Qué es la lógica de programación y por qué aprenderla?
3. Requisitos del sistema
4. Primeros pasos: Instalación y apertura de la app
5. Pantalla de inicio: ¿Qué ves y qué hacer?
6. Navegación general: Menú flotante y cómo moverte
7. Módulos y secciones: Cómo elegir y explorar los temas
8. Ejemplos dinámicos: Cómo aprender con casos reales
9. Ejercicios interactivos: Cómo practicar y validar tu lógica
10. Diagrama de flujo y modo paso a paso
11. Uso de Blockly: Programar sin escribir código
12. Selección de opciones en el módulo de variables
13. Consejos para aprender mejor y evitar errores comunes
14. Gestión de progreso y configuración
15. Preguntas frecuentes ampliadas
16. Bibliografía y recursos



Introducción

Pensamiento de Programador: Explorando la Lógica de la Programación es una innovadora aplicación de escritorio diseñada para facilitar el aprendizaje de la lógica de programación de forma visual, interactiva y progresiva. A través de una interfaz intuitiva y herramientas didácticas modernas, el software permite a los usuarios —especialmente a quienes están dando sus primeros pasos en el mundo de la programación— comprender conceptos fundamentales mediante la experimentación directa.

La plataforma integra teoría clara y accesible, ejemplos aplicados a la vida cotidiana y un sistema de resolución de ejercicios mediante bloques visuales tipo Blockly. Esta metodología elimina la barrera del código escrito desde el inicio y promueve un aprendizaje autónomo, permitiendo al usuario construir soluciones lógicas paso a paso mientras desarrolla habilidades clave como el pensamiento algorítmico, la resolución de problemas y la estructuración de procesos.

Ya sea como complemento en entornos educativos o como herramienta de autoformación, Pensamiento de Programador representa una nueva forma de acercarse a la programación, centrada en la lógica y el razonamiento computacional como pilares fundamentales para el desarrollo de habilidades digitales en el siglo XXI.

Requisitos del Sistema

- Windows 10/11 (x64)
- RAM: 2GB mínimo
- Procesador dual-core o superior
- No requiere conexión a internet para uso estándar

¿Qué es la lógica de programación y por qué aprenderla?

La lógica de programación es la base para entender cómo funcionan los programas y cómo resolver problemas paso a paso. Aprender lógica te permite pensar como un programador: analizar situaciones, descomponerlas en partes simples y construir soluciones ordenadas. No importa si nunca has programado: este curso te guía desde cero, usando ejemplos de la vida real y herramientas visuales.

Primeros pasos: Instalación y apertura de la app

Instala la aplicación descargando el archivo ejecutable (.exe) y sigue las instrucciones en pantalla. Una vez instalada, haz doble clic en el icono para abrirla. No necesitas internet para usarla, y tu progreso se guarda automáticamente en tu equipo.

- Tip: Si quieres usar la app en otro computador, copia el archivo .exe y la carpeta de datos.

Pantalla de inicio: ¿Qué ves y qué hacer?

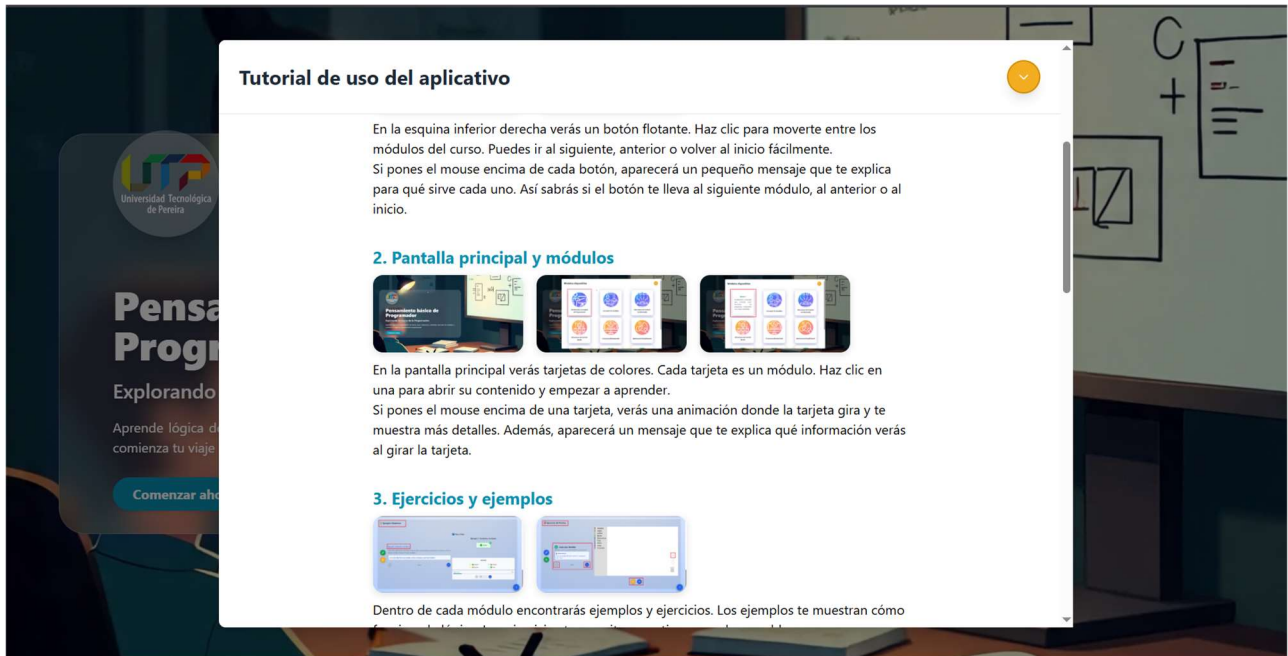
Al abrir la app, verás la pantalla principal con el nombre del aplicativo, el logotipo institucional y un botón destacado para comenzar. Haz clic en "Comenzar ahora" para acceder a los módulos.



- Consejo: Si es tu primera vez, explora el botón "¿Cómo usar el aplicativo?" para ver un tutorial visual.

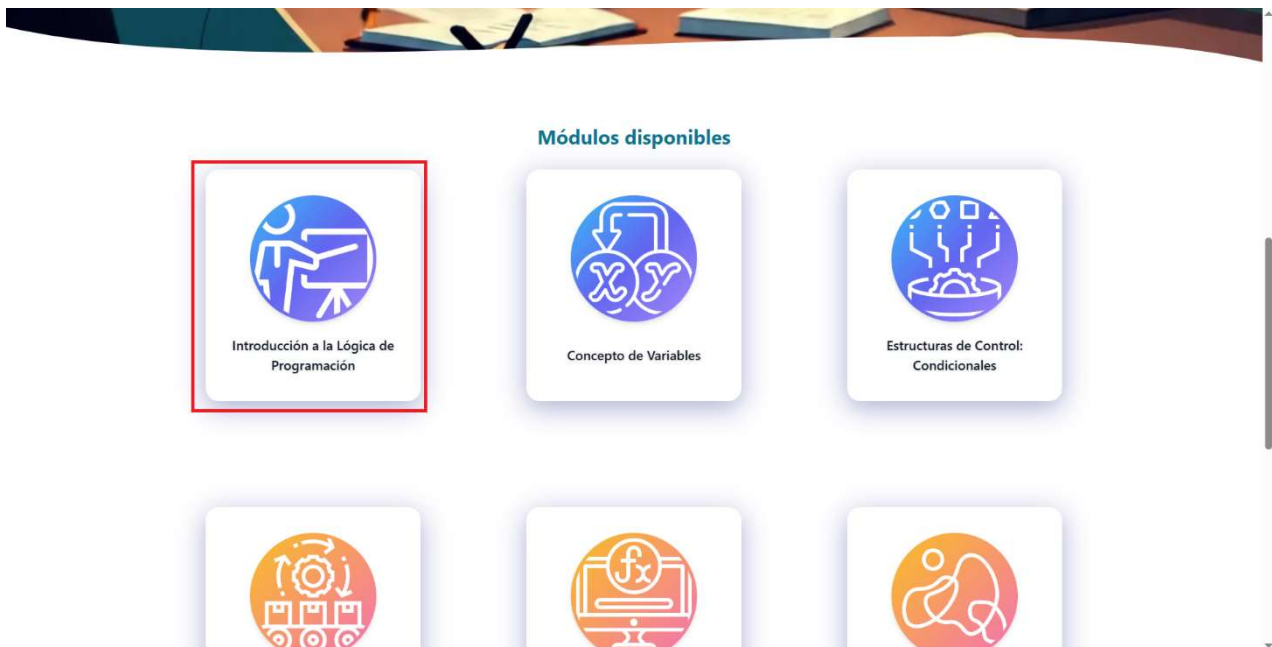
Tutorial de uso del aplicativo.

Si presionaste la opción de "¿Cómo usar el aplicativo? Se abrirá un pequeño tutorial de uso el cual es breve y sencillo con tan solo 8 pasos

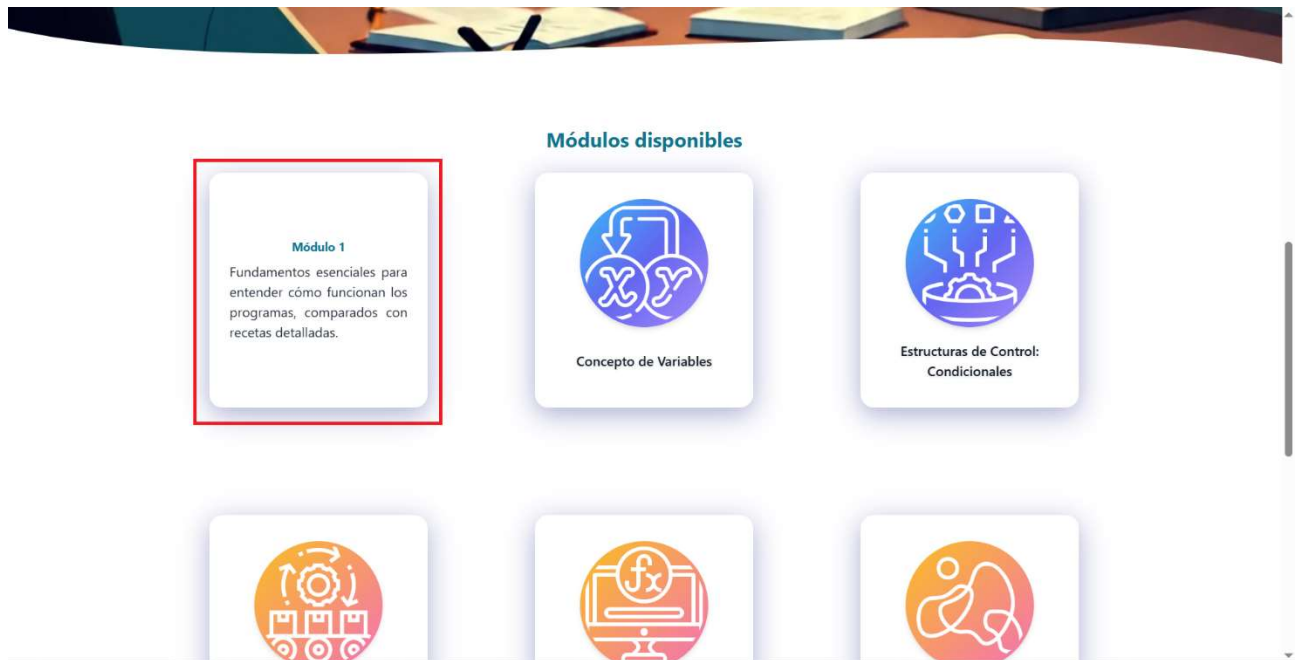


Módulos y secciones: Cómo elegir y explorar los temas

En la pantalla de módulos, verás tarjetas de colores. Cada tarjeta representa un tema clave de la lógica de programación. Haz clic en una para abrir su contenido.



Al pasar el mouse sobre una tarjeta, esta gira y muestra una breve descripción del módulo. Así puedes decidir cuál te interesa más antes de entrar.



Módulos disponibles:

- Introducción a la Lógica de Programación



- Concepto de Variables



Concepto de Variables

Módulo 2

Explicación sobre cómo las variables almacenan datos como números, palabras o imágenes, esenciales en programación.

- Estructuras de Control: Condicionales



Estructuras de Control:
Condicionales

Módulo 3

Uso de estructuras para tomar decisiones en programas según condiciones específicas.

- Estructuras de Control: Bucles



Estructuras de Control:
Bucles

Módulo 4

Repetición eficiente de acciones en programas usando bucles 'for' y 'while'.

- Funciones/Modularidad



Funciones/Modularidad

Módulo 5

Organización del código mediante funciones para facilitar su mantenimiento y reutilización.

- Abstracción/Simplificación

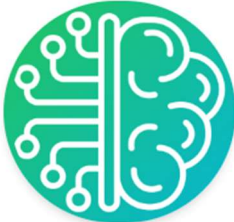


Abstracción/Simplificación

Módulo 6

Métodos para dividir problemas complejos en partes más simples y manejables.

- Pensamiento lógico y deductivo

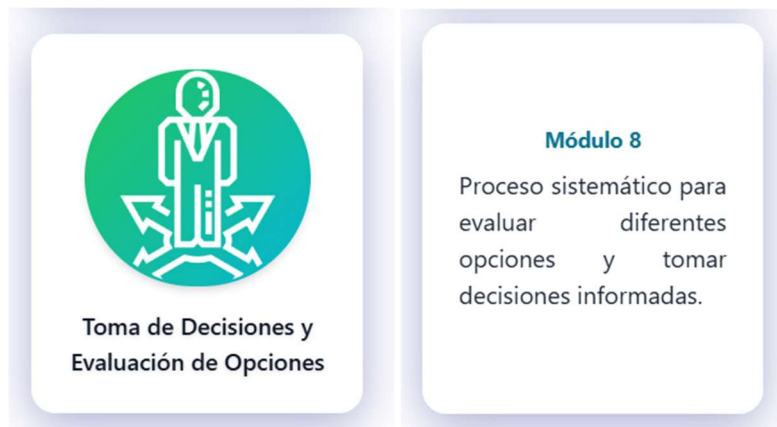


Pensamiento Lógico y Deductivo

Módulo 7

Aplicación de razonamiento lógico para resolver problemas usando premisas y conclusiones.

- Toma de decisiones y evaluación de opciones



- Solucion creativa de problemas



- Tip: Puedes cambiar de módulo en cualquier momento usando el menú flotante.

Navegación general: Menú flotante y cómo moverte

La navegación es sencilla gracias a un menú flotante en la esquina inferior derecha. Este menú te permite avanzar, retroceder o volver al inicio en cualquier momento.

Concepto de Variables

¿Qué es una variable?

En programación, una variable es un espacio de memoria reservado para guardar datos que pueden cambiar durante la ejecución de un programa. Estos datos pueden ser números, textos, imágenes, sonidos, valores booleanos (verdadero/falso), entre otros.

Elementos fundamentales de una variable

Una variable tiene tres elementos fundamentales:

- **Nombre:** Es el identificador único de la variable. Como una etiqueta que permite acceder a su contenido.
- **Tipo de dato:** Define el tipo de información que la variable puede almacenar. Ejemplos: entero, texto, decimal, imagen.
- **Valor:** Es el contenido actual que guarda la variable.

¿Por qué se llaman "variables"?

Porque su valor puede cambiar

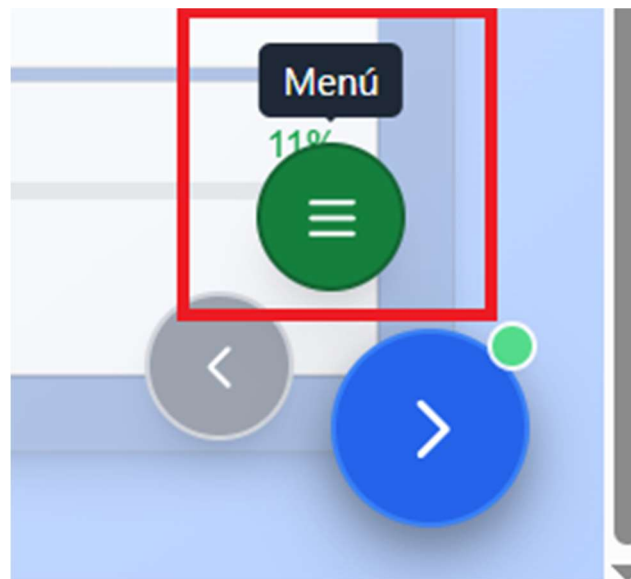
Por ejemplo, si declaras una variable llamada edad, podrías modificar su valor así:

```
edad = 15
# Más tarde en el programa...
edad = 16
```

Metáfora: Caja con etiqueta

- Botón azul: Avanza al siguiente módulo o sección.
- Botón amarillo: Retrocede al anterior.
- Botón verde: Regresa al menú principal.

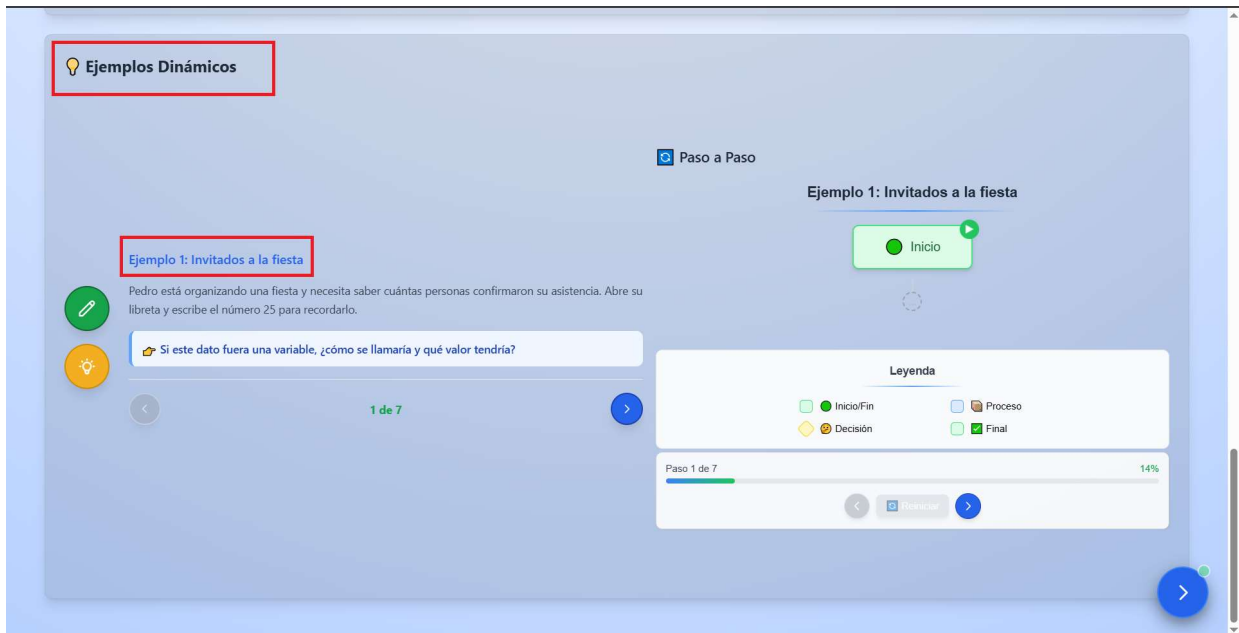
Al pasar el mouse sobre cada botón, verás un mensaje que explica su función.



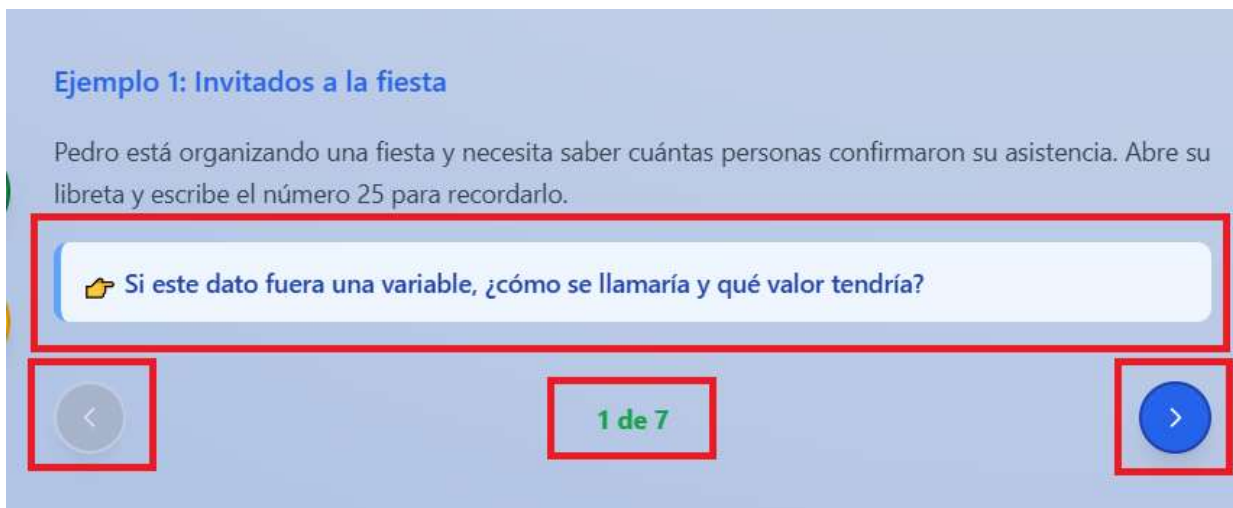
- Tip: Si te pierdes, usa el botón verde para volver al menú principal sin perder tu progreso.

Ejemplos dinámicos: Cómo aprender con casos reales

Cada módulo incluye ejemplos inspirados en situaciones cotidianas. Estos ejemplos te muestran cómo aplicar la lógica de programación para resolver problemas reales.



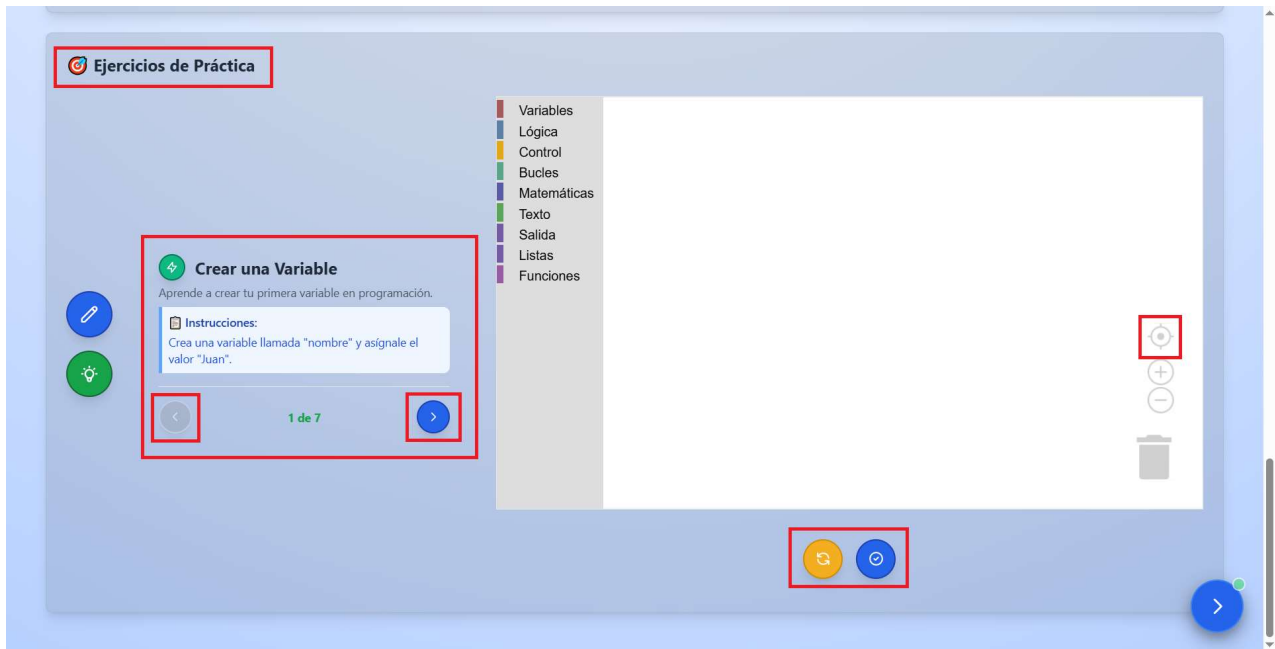
Los ejemplos se presentan paso a paso. Puedes avanzar o retroceder entre los pasos usando los botones azules y amarillos. Reflexiona sobre las preguntas interactivas para fortalecer tu comprensión.



- Tip: Si no entiendes un paso, vuelve atrás y repítelo. La clave es practicar y analizar cada parte del proceso.

Ejercicios interactivos: Cómo practicar y validar tu lógica

Pon a prueba lo aprendido resolviendo ejercicios prácticos. Cada ejercicio te plantea un reto y te guía para resolverlo usando bloques visuales.

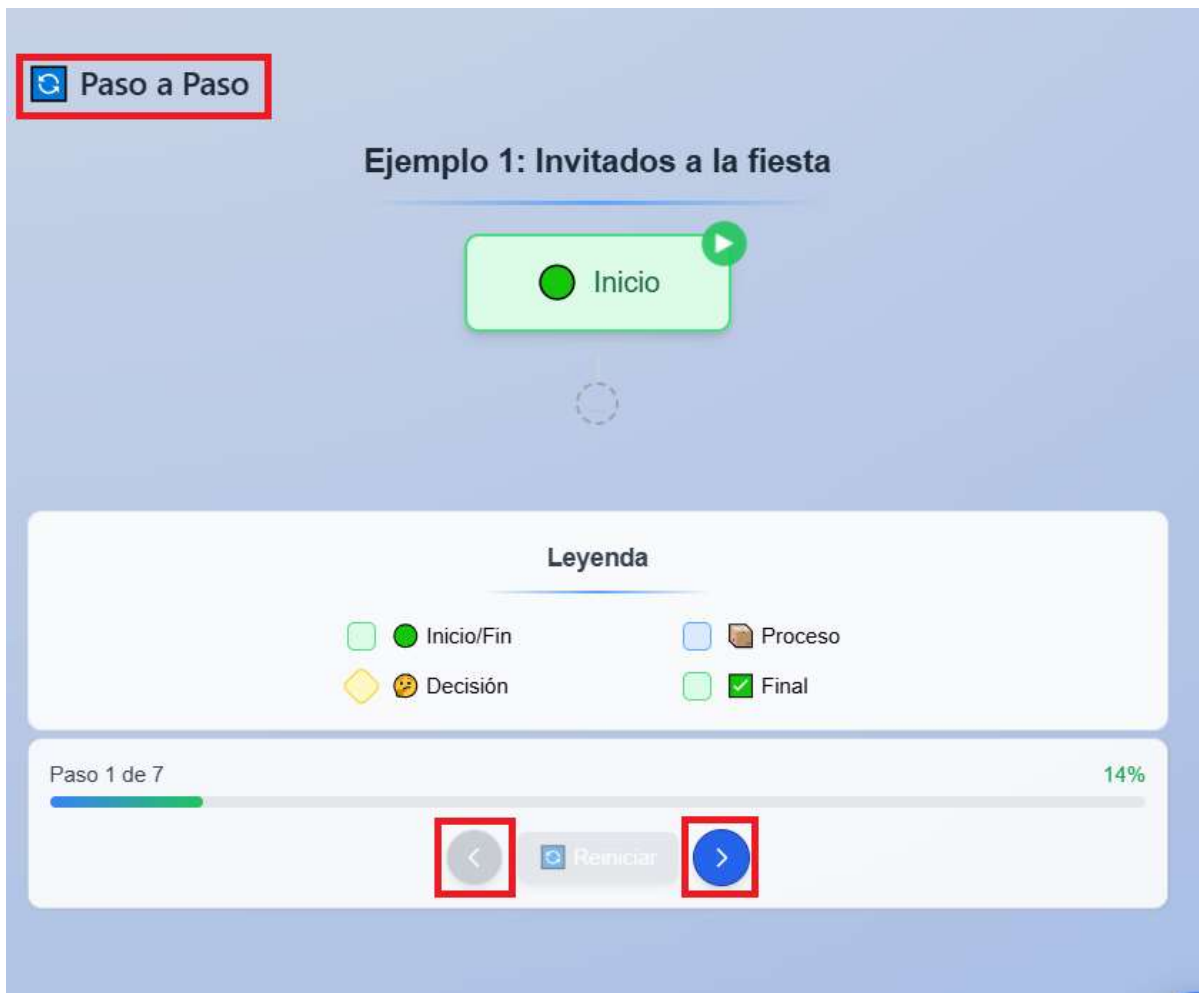


Lee las instrucciones, arrastra los bloques y construye tu solución. Usa los botones de navegación para avanzar o retroceder entre ejercicios. Puedes reiniciar el ejercicio en cualquier momento.

- Tip: Si te equivocas, no te preocupes. Reinicia y vuelve a intentarlo. Aprender de los errores es parte del proceso.

Diagrama de flujo y modo paso a paso

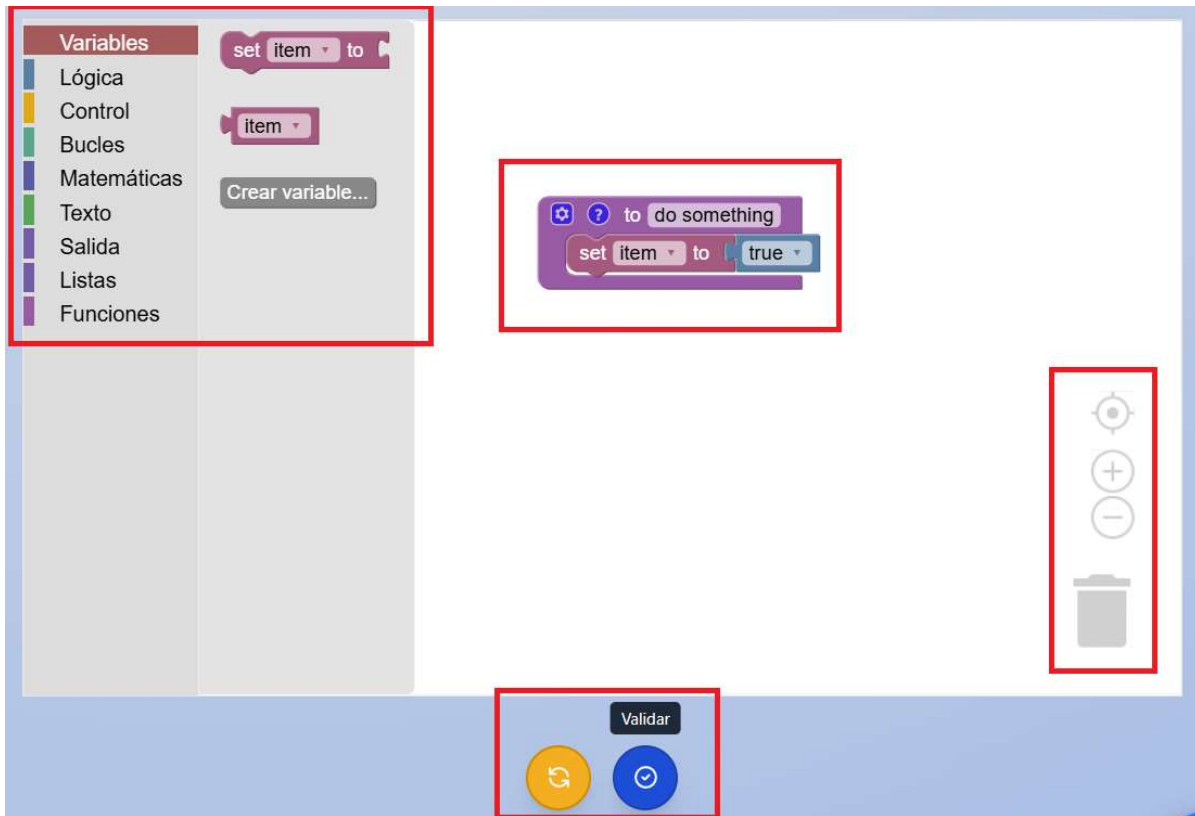
Muchos ejercicios y ejemplos incluyen diagramas de flujo. Estos diagramas te ayudan a visualizar la lógica y el orden de los pasos. Observa cómo se conectan los bloques y sigue el flujo para entender el proceso.



- Tip: El avance se muestra en porcentaje. Puedes reiniciar el ejercicio en cualquier momento para practicar desde el inicio.

Uso de Blockly: Programar sin escribir código

Blockly es una herramienta visual que te permite programar arrastrando y uniendo bloques. Así puedes centrarte en la lógica sin preocuparte por la sintaxis.

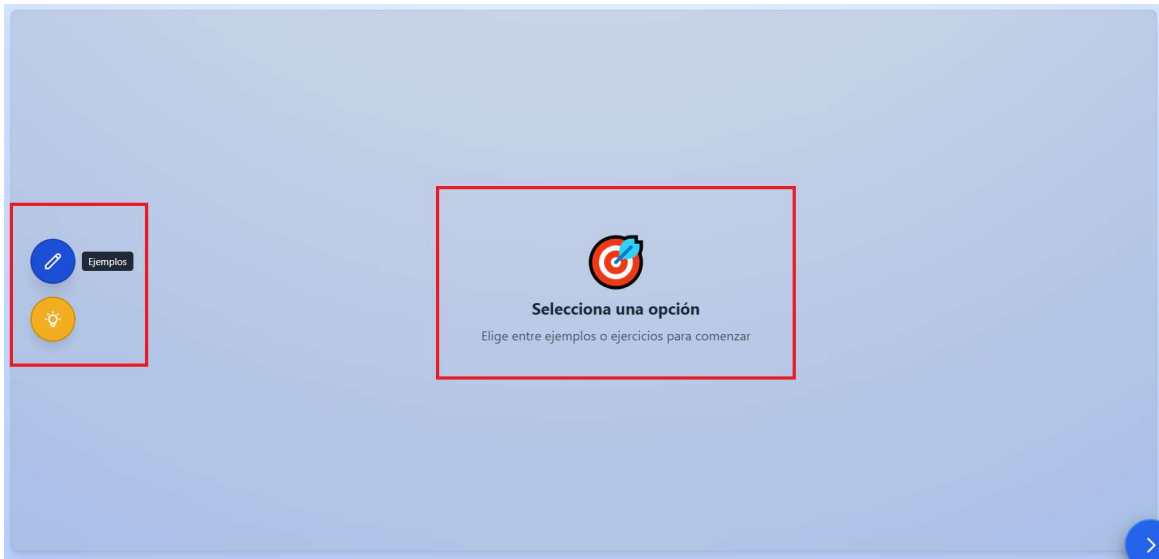


Selecciona bloques de la barra lateral (variables, lógica, bucles, etc.), arrástralos al área de trabajo y únelos para construir tu solución. Haz clic en Validar para comprobar si tu lógica es correcta.

- Tip: Usa las herramientas de la derecha para centrar, acercar, alejar o eliminar bloques según lo necesites.

Selección de opciones en el módulo de variables

En el módulo de variables, puedes elegir entre diferentes ejemplos y ejercicios específicos. Haz clic en la opción deseada para explorar el contenido relacionado.



- Tip: Usa los botones de la izquierda para alternar entre ejemplos y ejercicios.

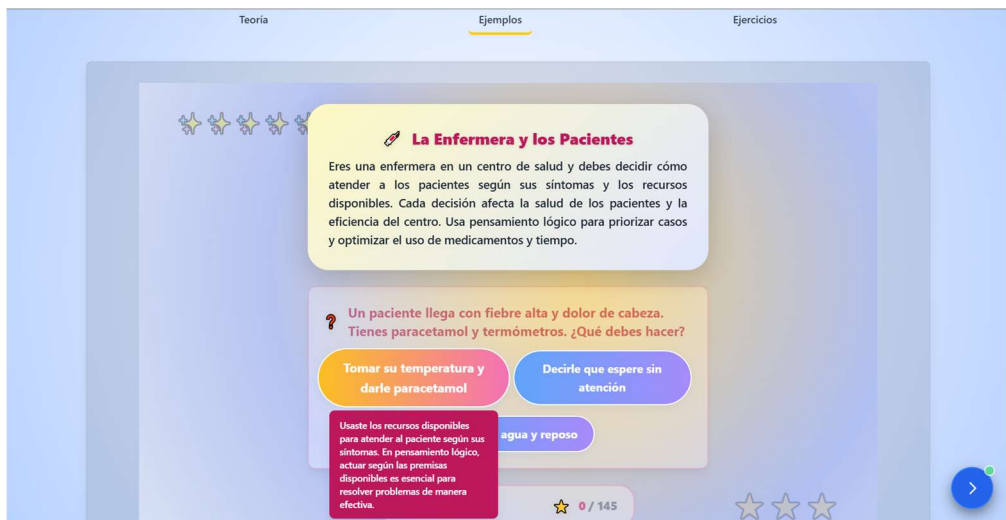
Juegos como metodología de enseñanza.

Uno de los juegos es arrastrar y soltar. Se entregan las instrucciones y mediante lógica el estudiante intenta resolver el reto, lo ejemplos cuentan con un paso a paso guiado, en cambio los ejercicios no, para aumentar su dificultad, la idea es que el estudiante pueda resolver los ejercicios sin ayuda y mejorar su abstracción y simplificación de problemas



El siguiente juego es un juego de elecciones y consecuencias, el cual consiste en que el estudiante deberá toma decisiones de acorde a su razonamiento lógico, en el caso de los ejemplos. Cada botón de respuesta tiene un mensaje emergente el cual explica lógicamente que puede pasar si se elije esta opción, para el caso de los ejercicios no cuenta con esta

opción, para que disponga de un reto elevado para así fomentar el pensamiento lógico en los estudiantes.



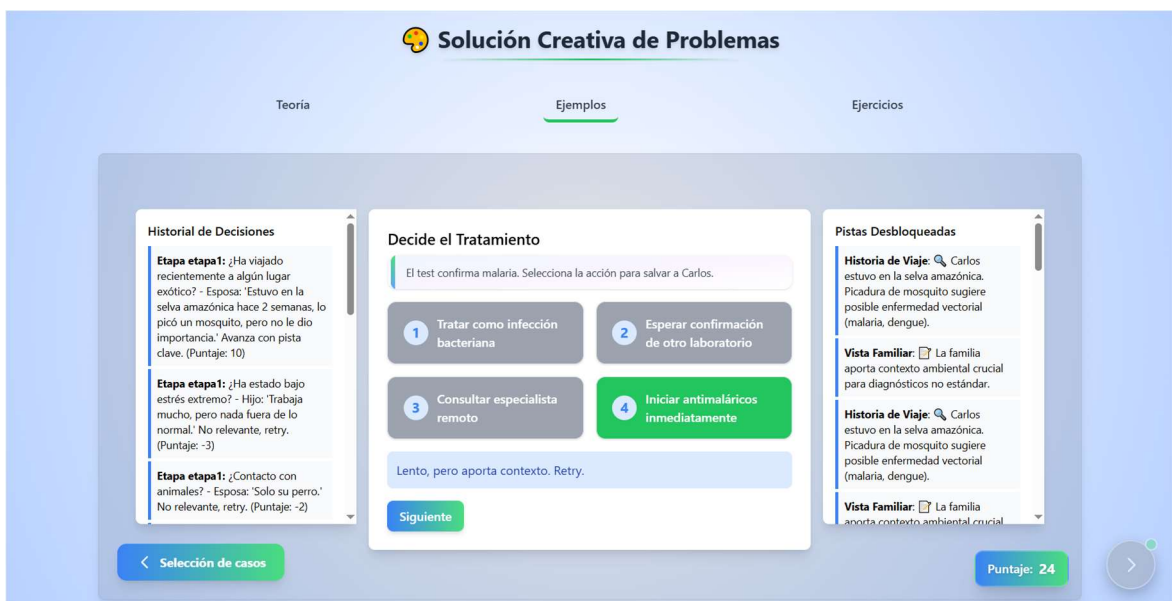
Para el siguiente juego **“EL CARRO RECOLECTO”** consta de un mini juego con varios mapas en los cuales el estudiante debe de recoger las mascotas y llevarlas a sus respectivas casas, el carro tiene gasolina por lo que debe de evaluar sus opciones y luego tomar una decisión para evitar quedarse sin gasolina y no poder llevar la mascota a su respectiva casa



Por último, tenemos el juego. **“EL DETECTIVE DE PREMISAS”** el cual consiste en resolver varias misterio para así fomentar el pensamiento creativo para la solución de problemas, el estudiante se pone en los zapatos de un detective y mediante pistas, preguntas y respuestas evaluar cual es la solución del misterio que paso, por que ocurrió



La pantalla del detective cuenta con ventanas, Historial de decisiones, decisión a tomar y las pistas desbloqueadas con lo cual podrá ir resolviendo el misterio.



Consejos para aprender mejor y evitar errores comunes

- Lee cada explicación con calma y sigue los pasos sugeridos.
- Practica los ejercicios varias veces para dominar la lógica.
- Usa los diagramas de flujo para entender el proceso antes de programar.



- No tengas miedo de equivocarte, cada error es una oportunidad para aprender.
 - Si tienes dudas, revisa este manual o consulta a tu profesor.
 - Si un bloque no encaja, revisa su forma y color: cada tipo de bloque tiene su función.
 - Si la pantalla se ve incompleta, ajusta el zoom o maximiza la ventana.
- Advertencia: No cierres la app mientras resuelves un ejercicio, ya que podrías perder el avance de ese ejercicio en particular.

Preguntas Frecuentes

¿Necesito internet para usar la app?

No, funciona completamente offline.

¿Pierdo mi progreso si cierro la app?

No, el progreso se guarda localmente.

¿Puedo usar la app en varios computadores?

Sí, solo debes copiar el archivo .exe y la carpeta de datos si quieres mantener tu progreso.

¿Cómo accedo a los módulos de teoría, ejemplos y ejercicios?

Desde el menú principal, haz clic en la tarjeta del módulo que deseas explorar. Usa los botones flotantes para cambiar de módulo en cualquier momento.

¿Qué hago si no veo los bloques en los ejercicios?

Verifica que la ventana esté maximizada o ajusta el zoom desde la configuración.

¿Para qué sirve el botón de menú flotante?

Permite regresar al menú principal desde cualquier módulo.

¿Qué significan los colores y los iconos de los botones?

El color azul indica avance, el amarillo retroceso, el verde acceso al menú. Los iconos de flecha y menú ayudan a identificar la función de cada botón.



Bibliografía

American Psychological Association. (2020). *Publication manual of the American Psychological Association* (7th ed.). <https://doi.org/10.1037/0000165-000>

Angular. (2025). Documentación oficial. <https://angular.io/>

Electron. (2025). Documentación oficial. <https://www.electronjs.org/docs>

Blockly. (2025). Documentación oficial. <https://developers.google.com/blockly>

Repositorio del proyecto: <https://github.com/Napssters/logic-programming>