
UAEH
Universidad Autónoma del Estado de Hidalgo

PREPA TRES

# 3.2 Diseño de Soluciones computacionales

M.C.C.C. Olivia Vázquez Bautista

El pensamiento algorítmico se refiere al desarrollo y uso de algoritmos que puedan ayudar a resolver un tipo especifico de problema o a realizar un tipo especifico de tarea.[1]

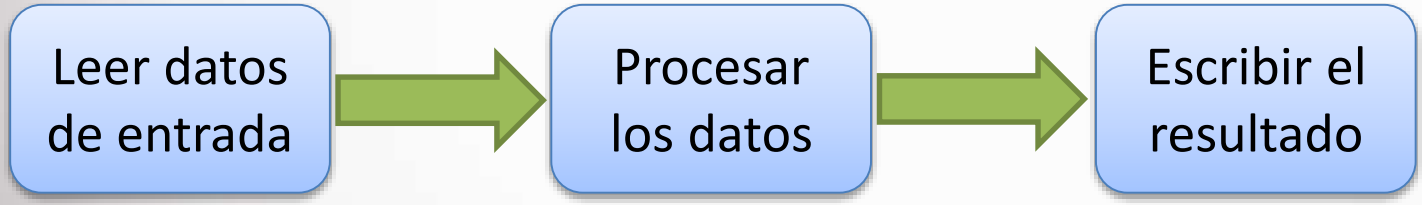
## Algoritmo

Serie de pasos, instrucciones u operaciones detallados y no ambiguos que permiten resolver un problema determinado.[1]

Los algoritmos deben ser: [2]

1. **Preciso.** Los pasos que conducen a la solución del problema, deben estar ordenados.
2. **Definido.** Indica que el resultado nunca debe cambiar bajo las mismas condiciones del problema, éste siempre debe ser el mismo.
3. **Finito.** Debe tener un fin.

Un algoritmo consiste en tres etapas:[2]



Los algoritmos se representan mediante herramientas o técnicas de programación, siendo las más utilizadas:

1. **Pseudocódigo.**
2. **Diagramas de flujo.**

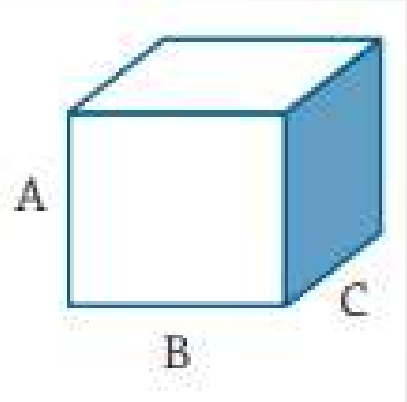
## Pseudocódigo.

Es un “falso lenguaje” (pseudo = falso) mezcla de instrucciones de programación y de palabras del lenguaje natural. Es una herramienta muy efectiva para el seguimiento de la lógica de un algoritmo y para transformarlo con facilidad a un programa informático. [3]

## Ejemplo de un pseudocódigo:

Pseudocódigo para determinar el volumen de una caja de dimensiones A, B y C se puede establecer de la siguiente forma:

1. Inicio.
2. Leer las medidas A, B y C.
3. Realizar el producto de A * B * C y guardarlo en V (V ⇘A * B * C).
4. Escribir el resultado V.
5. Fin.



## Diagramas de flujo.

Es la representación gráfica de un algoritmo. Para su construcción se usan símbolos estandarizados en el que cada uno tiene un significado. Los pasos del algoritmo se escriben dentro del símbolo correspondiente y se unen por flechas, denominadas “líneas de flujo”, que indican el orden en que los pasos deben de llevarse a cabo. [3]

# Diagramas de flujo - Simbología. [3]

| SÍMBOLO | NOMBRE | FUNCIÓN |
|---|---|---|
|  | Terminal / inicio | Indica el comienzo o el final de un diagrama de flujo. |
|  | Entrada/salida de datos | Utilizado para solicitar datos o representar la salida de datos por monitor o medios que NO sean de impresión. |
|  | Proceso | Empleado para indicar las operaciones. |
|  | Imprimir resultados | Indica la salida de un resultado por la impresora. |

# Diagramas de flujo - Simbología. [3]

| SÍMBOLO | NOMBRE | FUNCIÓN |
|---|---|---|
|  | Flujo de datos | Unen los diferentes bloques del diagrama e indican la secuencia de las instrucciones. |
|  | Conector dentro de página | Se emplea en la conexión de bloques, dentro de la misma página. Generalmente se usan letras o números para identificarlos. |
|  | Conector fuera de página | Usado para conectar bloques de página a página. También usan letras o números para identificarlos |
|  | Decisión | Se utiliza para representar una pregunta o interrogante que tiene al menos dos respuestas posibles. |

# 3.2.2 Construcción de soluciones computacionales


Scratch es un entorno de programación desarrollado por un grupo de investigadores, bajo la dirección del Dr. Michael Resnick, en una de las universidades más prestigiosas de Estados Unidos: MIT. [4]

# 3.2.2 Construcción de soluciones computacionales


Este entorno de programación fue diseñado como medio de expresión para ayudarte a expresar tus ideas de forma creativa, al tiempo que desarrollas habilidades de pensamiento algorítmico y de aprendizaje del Siglo XXI. [5]

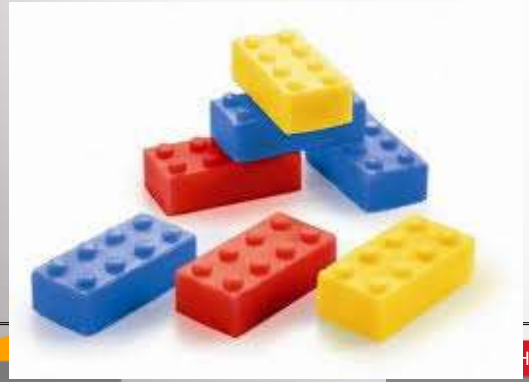
Puedes crear historias interactivas, juegos, animaciones y compartirlas con otras personas en Internet.

Los proyectos de Scratch son hechos con objetos llamados **SPRITES.**

Puedes cambiar la vista de un Sprite dándole diferentes apariencias. Un Sprite puede ser una persona, un tren una mariposa o alguna cosa.[5]




La gramática de Scratch se basa en un conjunto de “**bloques gráficos de programación**” que puedes ensamblar para crear programas. Tal como con las fichas de LEGO, conectores en los bloques sugieren de qué manera puedes ensamblarlos.[5]



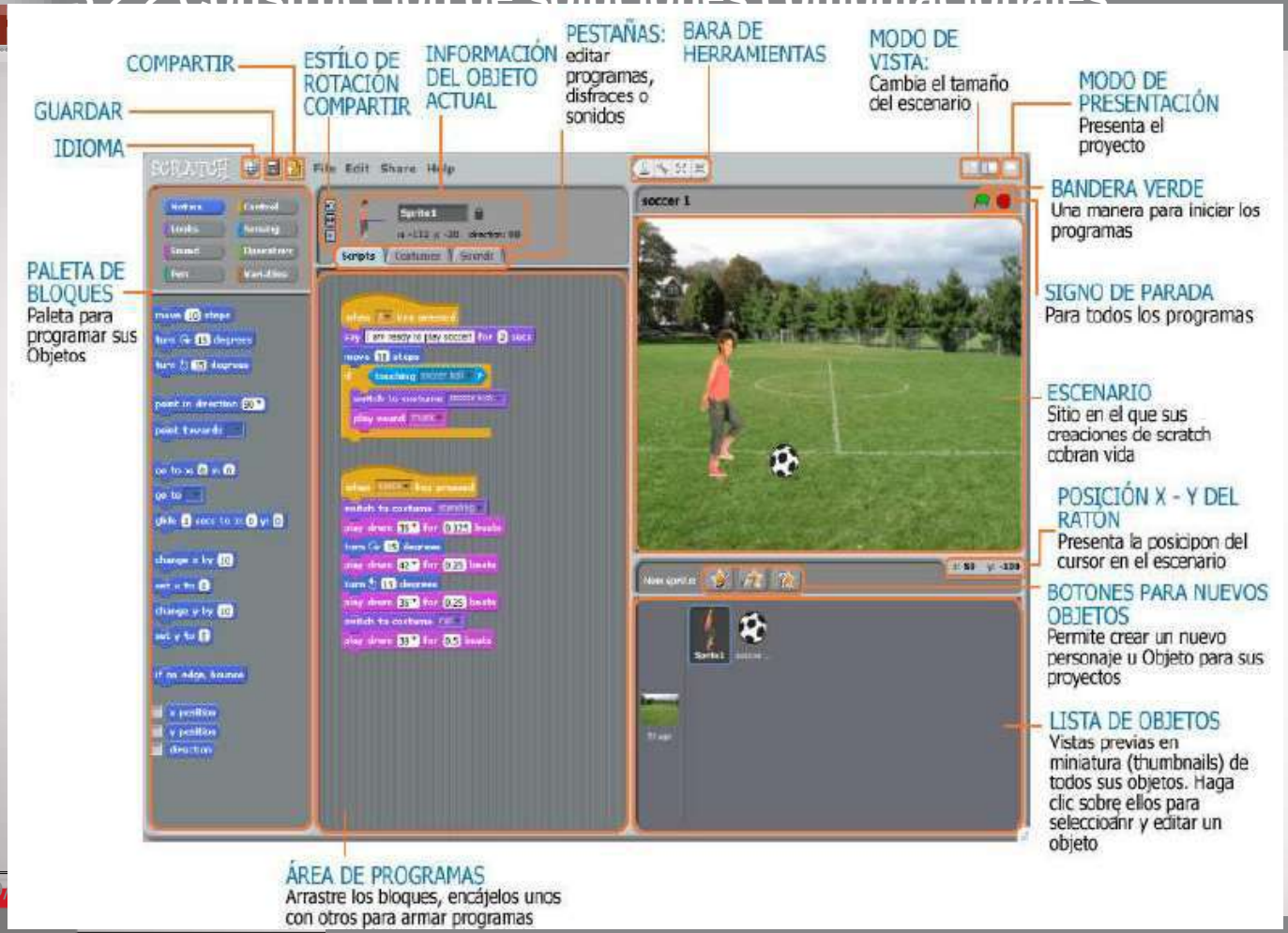
# 3.2.2 Construcción de soluciones computacionales

El **escenario** es donde puedes ver tus historias, juegos y animaciones dándoles vida. Los Sprites se mueven e interactúan con otros Sprites sobre el escenario.[5]

El escenario es de 480 unidades de ancho y de 360 unidades de alto. Por dentro esta dividido en una grilla de X – Y. A la mitad del escenario están ubicadas las coordenadas X = 0 y Y= 0.

# 3.2.2 Construcción de soluciones computacionales

## INTERFAZ DE SCRATCH



## Conceptos básicos de programación en Scratch.

**Identificadores:** son nombres que se dan a los elementos utilizados para resolver un problema y poder diferenciar unos de otros. [6]

Al asignar nombres (identificadores) a variables y procedimientos se deben tener en cuenta algunas reglas[6]:

- Los nombres pueden estar formados por una combinación de letras y números (saldoMes, salario,fecha2, baseTriángulo, etc).

## Conceptos básicos de programación en Scratch.

- El primer carácter de un nombre debe ser una letra.
- La mayoría de los lenguajes de programación diferencian las mayúsculas de las minúsculas.
- Los nombres deben ser nemotécnicos, con solo leerlos se puede entender lo que contienen. Deben ser muy descriptivos; no utilizar abreviaturas, a menos que se justifique plenamente.
- Es conveniente utilizar una sola palabra

## Conceptos básicos de programación en Scratch.

- No utilizar caracteres reservados (%, +, /, >, etc). Scratch admite letras acentuadas (á, é, í, ó, ú). Pero algunos lenguajes de programación no admiten las tildes.
- No utilizar palabras reservadas por los lenguajes de programación.
- Para cumplir con convenciones ampliamente utilizadas (Jiménez, 2002), los nombres deben empezar con minúscula. Ejemplo, fecha, suma, etc. Si es un nombre compuesto por varias palabras, cada una de las palabras (con excepción de la primera) deben empezar con mayúscula. Ejemplo:fechaInicial, baseTriángulo, etc.

## Conceptos básicos de programación en Scratch.

**Variables:** son espacios de trabajo(contenedores) reservados para guardar datos (valores). El valor de una Variable puede cambiar en algún paso del Algoritmo o permanecer invariable; por lo tanto, el valor que contiene una variable es el del último dato asignado a esta.[6]

## Conceptos básicos de programación en Scratch.

**Constante:** se crean de la misma forma que las variables y consisten en datos que, luego de ser asignados, no cambian en ninguna instrucción del Algoritmo. [6]

**Acumuladores:** consisten en guardar en una variable A el valor de ella misma, más otro valor variable B. [6]

## Conceptos básicos de programación en Scratch.

**Contadores:** consisten en almacenar en una variable (A) el valor de ella misma más un valor constante (1). [6]

Es muy útil para controlar el número de veces que debe ejecutarse un grupo de instrucciones.

## Conceptos básicos de programación en Scratch.

**Palabras reservadas (primitivas).** Todos los lenguajes de programación definen unas palabras para nombrar sus comandos, instrucciones y funciones. Un identificador definido por el usuario no puede tener el nombre de una palabra reservada. [6]

| Algunas palabras reservadas de Scratch | | | |
|---|---|---|---|
| adelante (ad) | atrás (at) | muestra | rumbo |
| derecha (de) | repite | para | cp |
| izquierda (iz) | da | limpia | sp |

## Conceptos básicos de programación en Scratch.

**TIPOS DE DATOS.** Scratch solo tiene tres tipos de datos[6]:

1. **Números.** se utilizan como entradas en las operaciones matemáticas. Cuando se utilizan los signos positivo (+) o negativo (-), estos deben estar pegados al número. Scratch acepta tanto el punto como la coma para escribir números decimales (3,14=3.14).

## Conceptos básicos de programación en Scratch.

2. **Palabras.** Las palabras están formadas por letras y/o números. Una palabra está delimitada por espacios en blanco; sin embargo, si se quiere tener un texto conformado por dos o más palabras, este debe encerrarse entre barras (|palabra1 palabra2|). [6]

## Conceptos básicos de programación en Scratch.

3. **Listas**. una secuencia de palabras puede manipularse igual que una sola palabra mediante el uso de listas. Una lista es una secuencia de palabras separadas por espacios en blanco y encerrada entre corchetes. Las palabras en una lista no necesitan comillas y los espacios en blanco se ignoran. [6]

## Conceptos básicos de programación en Scratch.

**Operadores.** Son símbolos que sirven para manipular datos. En scratch es necesario dejar un espacio en blanco a cada lado del signo aritmético. Los operadores y las operaciones que se pueden realizar con ellos se clasifican en [6]:

- **Aritméticos**: Posibilitan las operaciones entre datos de tipo numérico y dan como resultado otro valor de tipo numérico. Ejemplo: (^); (*); (/); (+); (-);asignación (“).

## Conceptos básicos de programación en Scratch.

- **Alfanuméricos:** Permiten operar con datos de tipo carácter o cadenas. Y debe utilizarse, para concatenar, el operador &.
- **Relaciónales**: Permiten la comparación entre datos del mismo tipo y dan como resultado dos valores posibles: Verdadero o Falso. Ejemplo: igual a (=);menor que (<); mayor que (>).
- **Lógicos**: Posibilitan la evaluación lógica de dos expresiones de tipo lógico. Dan como resultado uno de dos valores posibles: Verdadero o Falso. Ejemplo: negación (no); conjunción (y); disyunción(o).

## APP INVENTOR[5]

Es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación.

Sólo necesitas un navegador web y un télefono o tablet Android (si no lo tienes podrás probar tus aplicaciones en un emulador). [7]

**Solución de problemas con programación en Scratch.**

Escribir un procedimiento que se llame triangulo para hallar el área de un triángulo rectángulo cuya Base mide 3 cm, la Altura 4 cm y la Hipotenusa 5 cm.

## ANÁLISIS DEL PROBLEMA

**Formular el problema:** Ya se encuentra claramente planteado.

**Resultados esperados:** El área de un triángulo rectángulo.

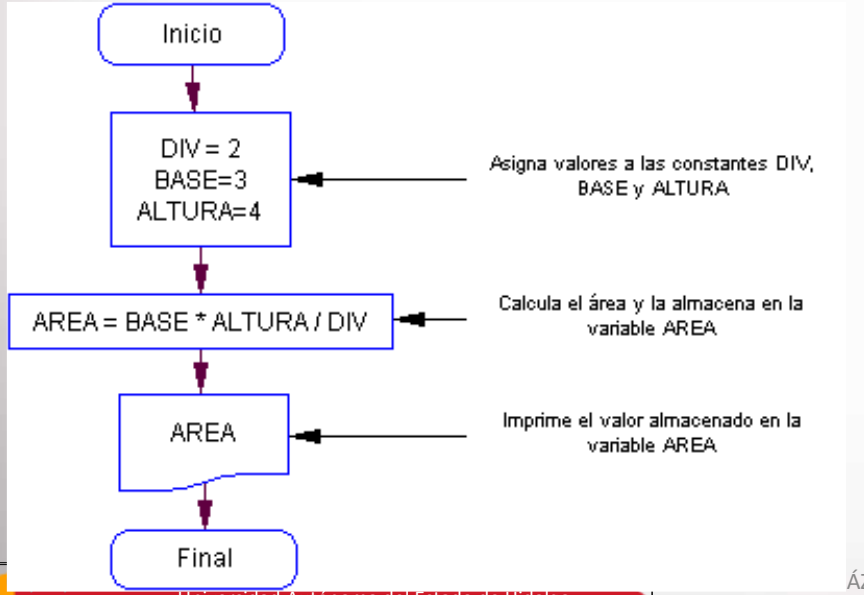
**Datos disponibles:** Base, Altura, Hipotenusa, tipo de triángulo. La incógnita es el área y todos los valores son constantes. El valor de la hipotenusa se puede omitir.

## ANÁLISIS DEL PROBLEMA

**Restricciones:** Utilizar las medidas dadas.

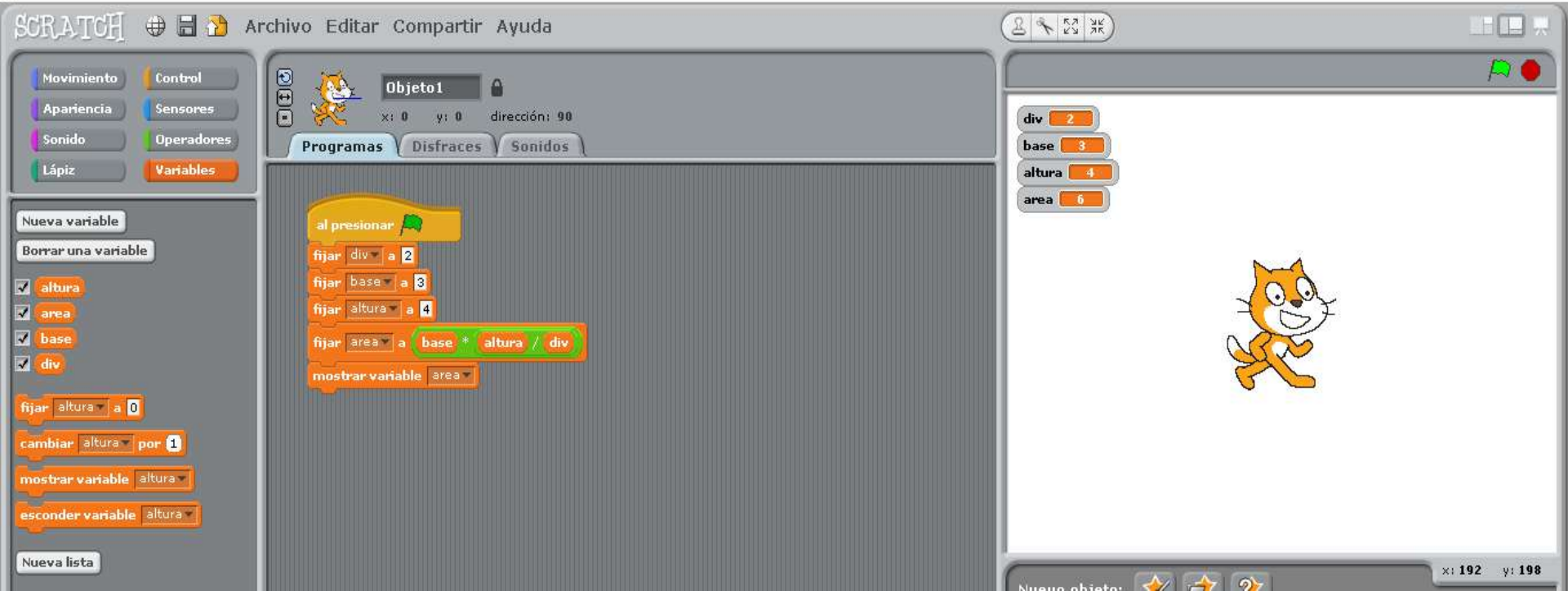
**Procesos necesarios:** Guardar en dos variables (BASE y ALTURA)los valores de Base y Altura; Guardar en una constante (DIV) el divisor 2; aplicar la fórmula BASE*ALTURA/DIV y guardar el resultado en la variable AREA; mostrar el resultado (AREA).

## Diseño de DFD y pseudocódigo




Inicio.
div ⇦2
base ⇦3
altura ⇦4
area ⇦base * altura/div
impirmir area
Fin.

## Solución scratch

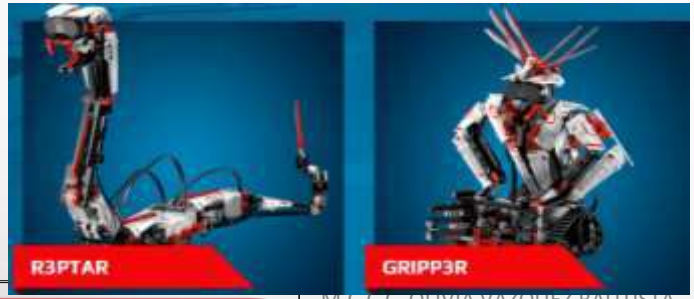


**Mindstrom Lego EV3 (armado y programación).** [8]

Es un set de robótica que te permite construir, programar y controlar tus propios robots LEGO del modo más inteligente, rápido y divertido. Solo debes seguir los pasos y, estarás controlando robots que se mueven, disparan, reptan, caminan, golpean y giran, por mencionar sólo algunas de las acciones.

## strom Lego EV3 (armado y programación).

Para construir un robot solo necesitas una caja: los bricks, motores y sensores te permiten construir cualquiera de los 17 robots EV3. Sólo tienes que elegir uno



**Mindstrom Lego EV3 (armado y programación).**

Usa la app gratuita EV3 Programmer para dar vida a tu robot con tu tableta. Elige un programa, pulsa el botón de ejecución y descubre cómo se comporta tu robot. O bien, descarga el software de programación en tu PC/Mac y disfruta de funciones de programación más avanzadas

https://www.lego.com/es-es/mindstorms/learn-to-program

# Referencias

1. Pinales Delgado F.J.; César Eduardo Velázquez Amador C.E. *Algoritmos Resueltos con Diagramas de Flujo y Pseudocódigo*. Primera edición. Impreso en México.
2. Elizondo, C. R. A. (2002). *Informática 2.* México DF, Grupo Editorial Patria.
3. Dirección Académica del Colegio de Bachilleres del Estado de Sonora. (2015). *Informática 2.* 1ª edición. Primera edición. Impreso en México.
4. http://scratch.mit.edu/
5. http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf
6. López García J.C.() ALGORITMOS Y PROGRAMACIÓN. SEGUNDA EDICIÓN, Fundación Gabriel Piedrahita Uribe www.eduteka.org
7. appinventor.mit.edu/
8. https://www.lego.com/es-es/mindstorms