

# KAREL . JS

Karel es un lenguaje de programación educativo que se utiliza para enseñar conceptos básicos de programación y lógica a principiantes.

A medida que avanzan en la programación de Karel, los estudiantes aprenden conceptos clave como secuencias de comandos, bucles, condicionales y abstracciones básicas



## ELEMENTOS BÁSICOS

los elementos básicos de Karel incluyen el robot Karel, el mundo bidimensional compuesto por calles y esquinas, las paredes que limitan el movimiento y los objetos que Karel puede recoger y colocar. Estos elementos forman la base del entorno de programación en Karel y permiten a los estudiantes practicar y experimentar con la programación de manera visual y práctica.

## COMANDOS

### AVANZAR

- Avanzar: El comando "avanzar" permite que Karel se desplace una casilla hacia adelante en la dirección en la que está orientado

### ZUMBADOR

- Los zumbadores son objetos que se pueden colocar en el mundo para que Karel los recoja. Después de recoger un zumbador, este desaparecerá del mundo y Karel lo llevará consigo.



### GIRA IZQUIERDA

- Gira izquierda: Este comando hace que Karel gire 90 grados hacia la izquierda sin moverse de su posición

### COJER/DEJAR ZUMBADOR

- Karel puede recoger un zumbador que se encuentre en la esquina en la que se encuentra.
- El comando "dejar zumbador" permite que Karel coloque un zumbador en la esquina en la que se encuentra. Si Karel tiene zumbadores en su posesión, puede dejar uno en la esquina actual. El zumbador quedará en la esquina y Karel continuará en su posición actual.



## ESTRUCTURAS DE CONTROL

Al igual que en otros lenguajes de programación, se pueden utilizar estructuras de control para controlar el flujo de ejecución de un programa y tomar decisiones en función de ciertas condiciones

### BUCLLES

while: ejecuta un bloque de código siempre que una condición sea verdadera. El bloque de código se repite hasta que la condición se evalúe como falsa.

For: Permite ejecutar un bloque de código un número específico de veces. Se especifica un valor inicial, una condición de finalización y una expresión de incremento o decremento. El bloque de código se ejecuta para cada valor en el rango definido.

### CONDICIONALES

Las estructuras condicionales permiten tomar decisiones basadas en condiciones específicas. En Karel, se utiliza principalmente el condicional "if" para evaluar una condición y ejecutar un bloque de código si la condición es verdadera. If: Evalúa una condición y se ejecuta un bloque de código si la condición es verdadera. Si la condición es falsa, el bloque de código dentro del "if" se omite. También se pueden utilizar cláusulas "else" o "else if" para especificar bloques de código alternativos en caso de que la condición no sea verdadera.

# KAREL . JS

## RESOLUCIÓN DE PROBLEMAS DE KAREL



1. Comprende el problema: Antes de comenzar a escribir código, es importante comprender completamente el problema que se te presenta. Lee atentamente la descripción del problema y asegúrate de entender lo que se te pide.

2. Divide el problema en tareas más pequeñas: Si el problema parece demasiado grande, es útil dividirlo en tareas más pequeñas y abordarlas por separado. Esto facilita el proceso de resolución y te permite concentrarte en una parte específica del problema a la vez.

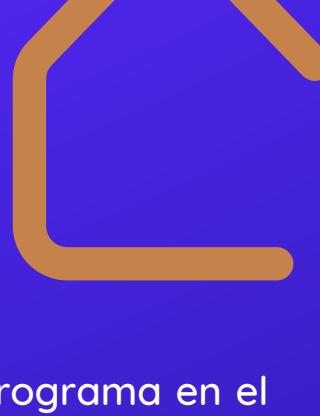
3. Diseña un plan: Una vez que hayas dividido el problema en tareas más pequeñas, diseña un plan general de cómo abordar cada tarea y cómo combinarlas para resolver el problema en su conjunto. Puedes hacer esto mediante un pseudocódigo o un diagrama de flujo.

## EJEMPLOS DE PROYECTOS EN KAREL

1. Crear un laberinto: Diseña un laberinto utilizando las funciones de Karel el Robot. El objetivo es que Karel encuentre la salida del laberinto siguiendo un camino correcto y evitando obstáculos.



2. Recoger basura: Coloca basura en diferentes ubicaciones del mundo de Karel y programa a Karel para que recoja la basura y la lleve a un lugar designado, como una papelera.



3. Construir una casa: Utiliza los comandos de Karel para construir una casa en el mundo. Puedes diseñar el plano de la casa y programar a Karel para que coloque los ladrillos en su lugar.



4. Patrulla de seguridad: Crea un programa en el que Karel patrulle un área específica del mundo y verifique si hay objetos o situaciones sospechosas. Si encuentra algo fuera de lo común, puede dar una alarma o tomar alguna acción específica.



5. Juego del laberinto: Desarrolla un juego en el que el usuario controle a Karel y tenga que guiarlo a través de un laberinto hacia la salida. Puedes diseñar diferentes niveles de dificultad y agregar elementos interactivos al juego.

# KAREL . JS

## PORQUE APRENDER KAREL

1.º te permite tener un conocimiento sobre el programa pascal con sus instrucciones de repetición y de condicionamientos sencillos.

2.º te permite desarrollar un conocimiento básico sobre programación.

3º permite el desarrollo de nuevas habilidades de conocimiento, así como pueden ser de trabajo en equipo.

Además esto nos permite tener una mejor percepción de los procesos lógicos a la hora de programar en otro lenguaje

Aprender a programar puede acelerar la adquisición de conocimientos mediante el desarrollo de habilidades cognitivas. Como resultado, recordamos mucho mejor la información, y el cerebro sometido a un ejercicio constante es mucho más sano.



Conociendo estas herramientas que nos brinda el mundo tecnológico podemos aprender de manera monótona el sistema y la lógica de programación que este programa, conociendo ya que el avance tecnológico que esta teniendo el mundo.

Recordemos que los tiempos cambian por lo que te recomendamos aprender a programar, todo el mundo lo puede hacer.

