

Universidad Rafael Landívar

Pensamiento computacional

Catedrática: Lic. Cindy García Pérez

## EVALUACION PROYECTO 01

Javier Estrada Núñez

Rubi Paola Sánchez

- ¿Qué acciones debe poder hacer su programa?
  1. Ver información del robot.
    - 1.1. Los detalles del robot (ingresados inicialmente por el usuario).
    - 1.2. El tipo de modo alterno (auto, camión, moto).
    - 1.3. Cantidad de energon.
    - 1.4. Posición.
  2. Cargar Energon.
    - 2.1. El nivel de energon del robot se incrementa en 5%.
    - 2.2. Debe mostrar al usuario el nivel de energon con el que quedó el robot.
  3. Transformarse.
    - 3.1. Cambia de modo robot a modo vehículo y viceversa sucesivamente.
    - 3.2. Al concluir debe indicar el modo actual del robot.
  4. Movilizarse.
    - 4.1. Deberá de preguntar cuanto tiempo (en horas) desea que el robot se mueva.
    - 4.2. Antes de iniciar el movimiento se debe de calcular si el energon será suficiente, si no lo es se debe de informar al usuario.
    - 4.3. Mostrar al usuario la posición y la cantidad de energon que tiene el robot.
  5. Salir.
    - 5.1. Permite que el usuario termine la ejecución del programa.

- ¿Con qué datos trabajará? ¿Qué información debe pedir al usuario?, defina sus datos de entrada y el tipo de dato que utilizará para los datos principales.

R/ Con los datos que trabajare son los siguientes: Nombre del robot, El tipo de modo alterno (auto, camión, moto) nivel inicial de energon y posición inicial.

Estos serán los datos de entrada, para saber la información principal del robot, la cual se presentará en un menú interactivo.

La información que debe pedir al usuario es la siguiente: Nombre, modo, velocidad, nivel y posición.

Los datos principales serán:

El nombre (str)

El modo (str)

Velocidad (float)

El nivel de energon (float)

La posición (float)

- ¿Qué variables utilizará para almacenar la información?

“nombre” = (string) para almacenar el nombre del robot.

“modo” = (string) para almacenar el modo alterno del robot (auto, camión, moto).

“nivel energon” = (float) para almacenar el nivel inicial y actual de energon del robot.

“posición” = (float) para almacenar la posición inicial y actual del robot.

“opción” = (string) para almacenar la opción seleccionada por el usuario en el menú interactivo.

“velocidades” = (float) {'robot': 50, 'auto': 100, 'camión': 85, 'moto': 120}

- ¿Qué condiciones o restricciones debe tomar en cuenta? ¿Qué cálculos debe hacer?

Para empezar, tomando en cuenta los datos y la información presentada las condiciones que debemos tomar en cuenta son las siguientes:

Validación de datos de entrada: El programa debe asegurarse de que el usuario ingrese datos válidos al solicitar información sobre el robot, como un nombre válido, un tipo de modo alternativo válido (auto, camión, moto), un nivel inicial de energon válido y una posición.

Gasto de energon: El programa debe calcular correctamente el gasto de energon durante las diferentes acciones, como cargar energon y moverse.

Transformación: Alternar entre los modos del robot y vehículo, manteniendo actualizado el modo del robot.

Movilización: El código debe calcular si el nivel de energon es apto para el tiempo de uso y la distancia que recorrerá, si el nivel no es suficiente debe informar al usuario y el robot no debe moverse.

Uso lógico: El programa debe seguir cada indicación de manera ordenada, realizando cada acción coherentemente, manteniendo la información actualizada.

Luego, tomando en cuenta los datos y la información presentada las restricciones que debemos tomar en cuenta son las siguientes:

- **El nivel del energon no puede ser negativo en cada acción.**
- **El programa debe permitir al usuario seleccionar opciones válidas del menú interactivo y proporcionar retroalimentación adecuada en caso de que se ingrese una opción inválida.**
- **El programa debe asegurarse de que el usuario ingrese datos válidos al solicitar la cantidad de tiempo para moverse, como un número positivo.**

Por último, pero no menos importante, los cálculos que debemos realizar serán los siguientes:

- Calcular el gasto de energon durante el movimiento en función del tiempo de movimiento y el gasto de energon por hora correspondiente al modo actual del robot. Para esta opción, el programa simplemente necesita aumentar el nivel de energon del robot en un 5% cada vez que el usuario seleccione esta acción.
- Calcular la distancia que recorrerá el robot en función del tiempo y la velocidad.
- Calcular la distancia, se puede multiplicar la velocidad por el tiempo; para calcular el gasto de energon, se puede multiplicar el gasto de energon por hora por el tiempo de movimiento.
- Mostrar al usuario la posición actualizada y el nivel de energon restante después del movimiento.

Algoritmo que implementará en el programa, descrito mediante el Diagrama de Flujo elaborado en Draw.io, para mostrar la lógica de las diferentes acciones.











