



## Curso de robótica de la UTEC

### Código Programado con Arduinoblocks para el carro controlado por Bluetooth

Profesor: Omar Alejandro Bazar

#### Índice:

##### Páginas 1-2: Introducción y Configuración Inicial

- Presentación del proyecto (p.1)
- Acceso a Arduinoblocks (p.1)
- Descripción de la interfaz (panel de bloques, área de programación, consola) (p.1-2)

##### Páginas 2-3: Bloques de Inicialización

- Configuración Bluetooth (Tx: A1, Rx: A2, Baudios: 9600) (p.2-3)
- Definición de variables (VELOCIDAD = 250) (p.3)

##### Páginas 3-4: Bucle Principal y Recepción de Datos

- Estructura del bucle principal (p.3)
- Condicional para datos recibidos (p.3)
- Almacenamiento de órdenes en variable (ordenes) (p.3-4)

##### Páginas 4-5: Bloques de Control de Movimiento

- Comandos: "ADELANTE" y "ATRAS" (activación de motores) (p.4)
- Comandos: "derecha" e "izquierda" (giros) (p.5)
- Comando: "PARAR" (detención de motores) (p.5)

##### Páginas 5-6: Funciones Personalizadas de Motores

- Bloques adelante m1/m2, atras m1/m2, parar m1/m2 (p.5-6)
- Configuración de pines digitales y PWM (p.6)

##### Páginas 6-7: Subida de Código y Pruebas

- Proceso para subir código a Arduino (p.6)
- Pruebas con dispositivo Bluetooth (p.7)
- Consejos para estudiantes (p.7)

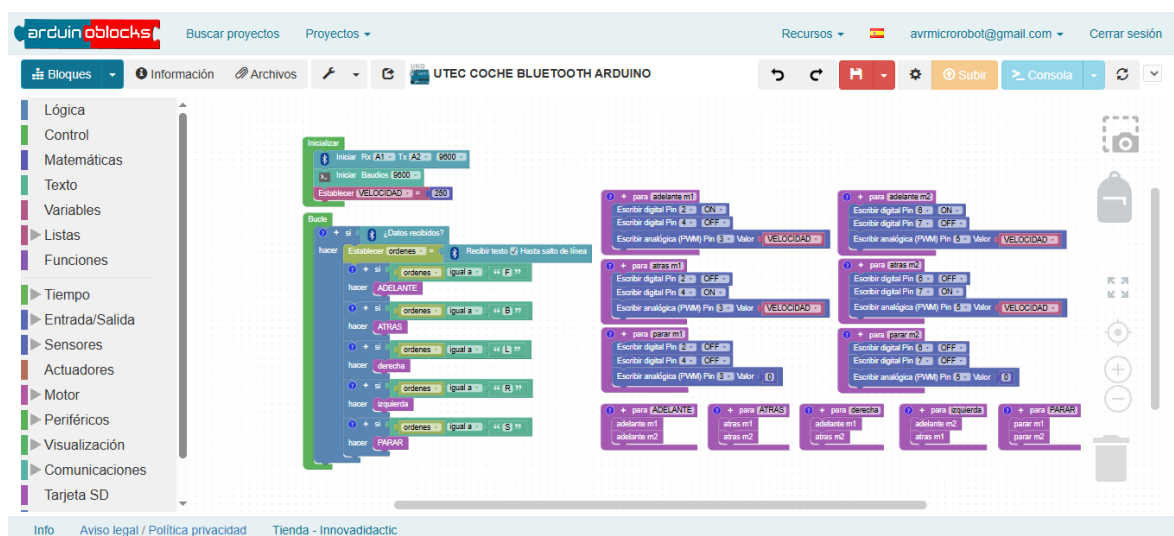
## Cómo Usar el Compilador de Arduinoblocks

Arduinoblocks es un entorno de programación visual que facilita la creación de código para placas Arduino utilizando bloques en lugar de escribir código de texto directamente. Aquí te explico cómo usarlo:

### . Acceder a Arduinoblocks:

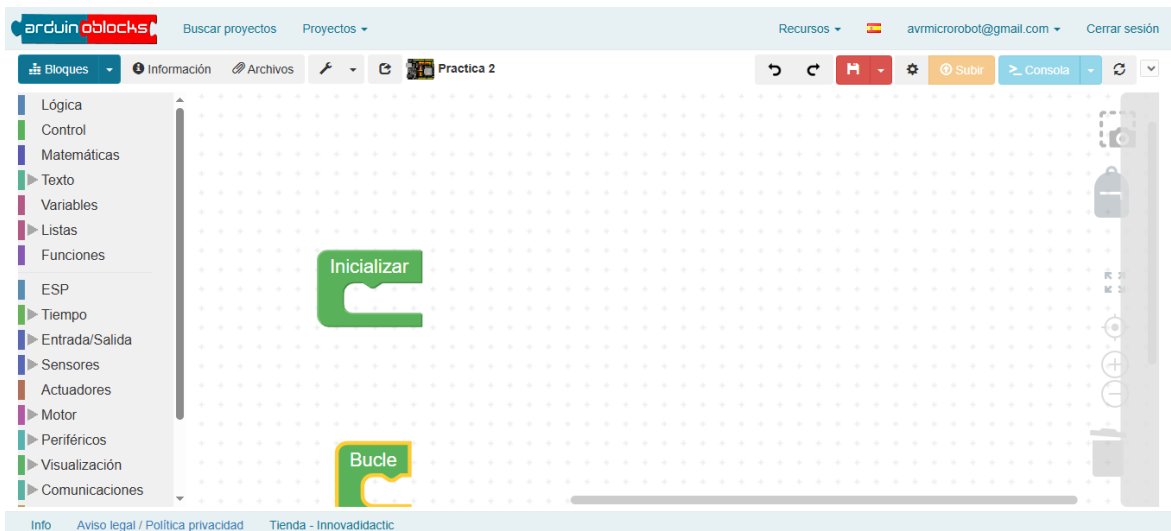
- Generalmente, accedes a Arduinoblocks a través de un navegador web. Busca en internet "Arduinoblocks" o la plataforma específica que estés utilizando (a veces viene integrado en otros entornos o sitios web educativos).

### Familiarizarse con la Interfaz:



- **Panel de Bloques (Izquierda):** Aquí encontrarás diferentes categorías de bloques organizados por su función (Lógica, Control, Matemáticas, Pines, Sensores, Actuadores, etc.). Puedes explorar estas categorías para ver los bloques disponibles.
- **Área de Programación (Centro):** Este es el espacio donde arrastrarás y conectarás los bloques para construir tu programa.
- **Barra de Herramientas (Superior):** Contiene botones para acciones como guardar, abrir, verificar el código, subir el código a la placa Arduino, etc.
- **Consola (Inferior o Lateral):** Puede mostrar mensajes de error, información de compilación y otras salidas.

### Crear un Programa con Bloques:

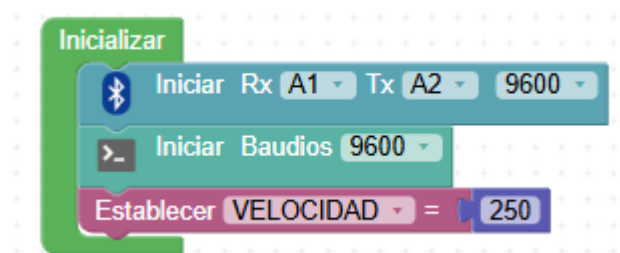


- **Arrastrar y Conectar:** Para programar, simplemente haz clic en un bloque de la paleta izquierda y arrástralo al área de programación. Luego, conecta los bloques entre sí encajando las muescas y las formas. La forma de los bloques indica cómo se pueden conectar (por ejemplo, bloques de condición tienen espacios para bloques de "hacer").
- **Configurar Bloques:** Algunos bloques tienen menús desplegables, campos de texto o números que puedes modificar para personalizar su comportamiento (como seleccionar un pin, establecer un valor o escribir un texto). En el ejemplo de la imagen, puedes ver valores numéricos en los bloques de inicialización del Bluetooth y la velocidad.
- **Organizar el Código:** A medida que tu programa crece, es importante organizarlo de manera lógica para que sea fácil de entender y modificar.

## Descripción de la Programación en Bloques con Arduinoblocks

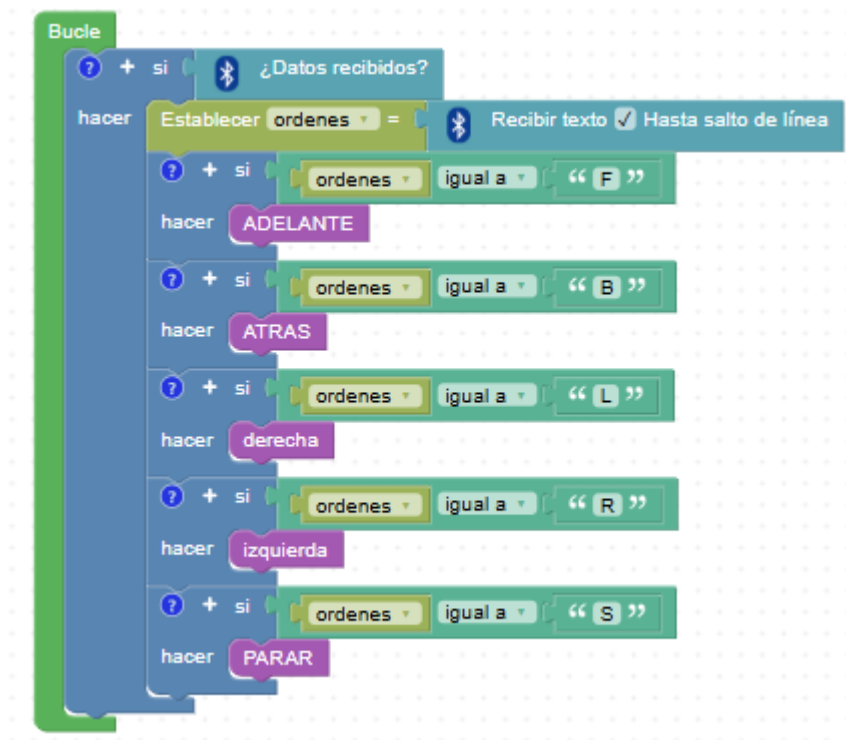
El programa en Arduinoblocks está diseñado para controlar un coche Bluetooth Arduino. Podemos identificar las siguientes secciones y funcionalidades:

### Bloques de Inicialización:



- **Iniciar BT Tx: A1 Rx: A2 Baudios: 9600:** Este bloque configura la comunicación Bluetooth. Indica que el pin digital A1 se utiliza para transmitir datos (Tx) y el pin digital A2 para recibir datos (Rx), con una velocidad de transmisión de 9600 baudios. Esto es esencial para que el Arduino se comunique con un dispositivo Bluetooth (como un teléfono o una computadora).
- **Establecer VELOCIDAD v = 250:** Este bloque probablemente inicializa una variable llamada VELOCIDAD con el valor de 250. Esta variable se utilizará más adelante para controlar la velocidad de los motores.

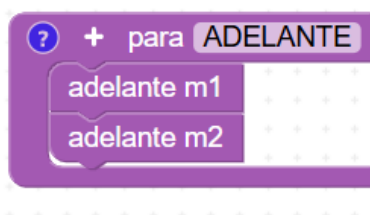
## Bucle Principal (Bucle):

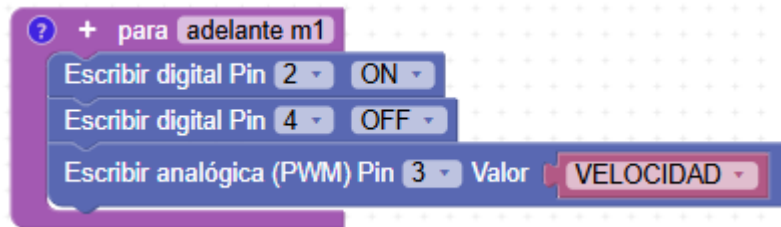


- **Hacer:** Este bloque indica el inicio del bucle principal del programa, que se ejecutará continuamente una vez que el Arduino se enciende.
- **si Datos recibidos?:** Este es un bloque de condición. El programa verifica si se han recibido datos a través de la conexión Bluetooth.
  - **Hacer:** Si se reciben datos, se ejecutan los bloques dentro de este "hacer".
  - **Establecer ordenes = Recibir texto ( ) Hasta salto de línea:** Este bloque lee el texto recibido a través de Bluetooth hasta que encuentra un salto de línea y lo almacena en una variable llamada ordenes. Se espera que estas "órdenes" sean comandos de texto enviados por el dispositivo Bluetooth para controlar el coche.

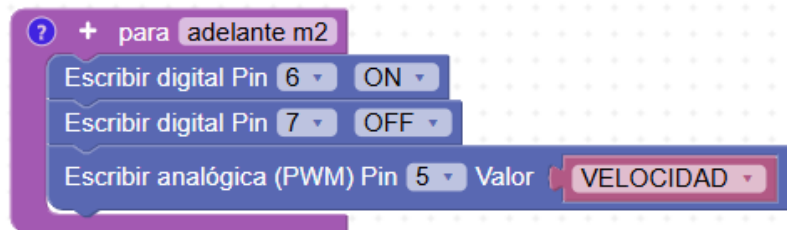
## Bloques de Control de Movimiento (Condicionales):

- Se utilizan varios bloques de condición si para interpretar las órdenes recibidas:
  - **si ordenes = "ADELANTE":** Si la orden recibida es la palabra "ADELANTE", se ejecutan los siguientes bloques para hacer que el coche avance:



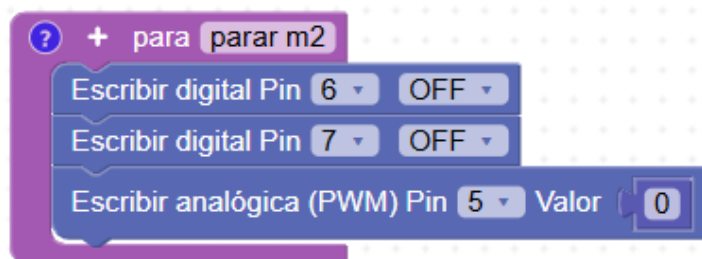
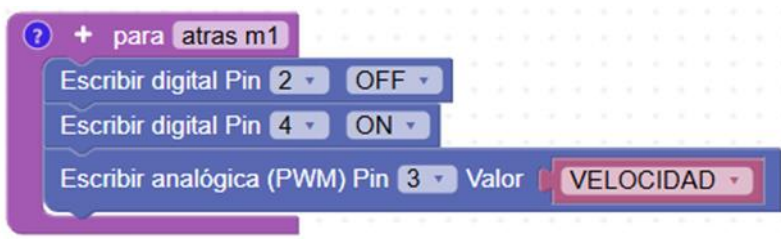


- **para adelante m1:** Activa el motor 1 para que gire hacia adelante. Esto se logra configurando los pines digitales 2 y 3 en OFF y ON respectivamente, y enviando una señal PWM (Pulse Width Modulation) al pin 6 con el valor de la variable VELOCIDAD (250). La modulación PWM permite controlar la velocidad del motor variando el ancho de los pulsos eléctricos.

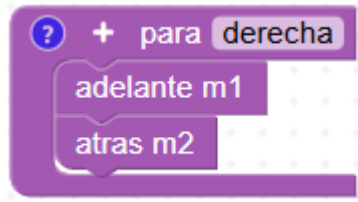


- **para adelante m2:** Similar al anterior, activa el motor 2 para que gire hacia adelante, utilizando los pines digitales 4 y 5 (OFF y ON) y el pin PWM 9 con la misma VELOCIDAD.

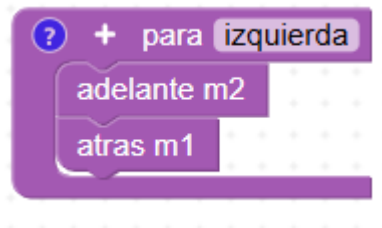
- **si ordenes = "ATRAS":** Si la orden es "ATRAS", se activan los motores en la dirección opuesta (pines 2 y 3 en ON y OFF para m1, y 4 y 5 en ON y OFF para m2), manteniendo la misma velocidad.



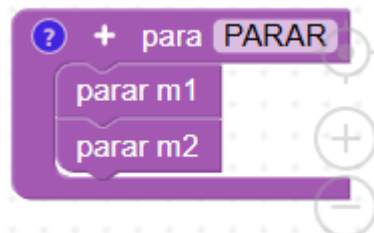
- **si ordenes = "derecha"**: Para girar a la derecha, se hace que un motor avance (adelante m1) y el otro retroceda (atras m2).



- **si ordenes = "izquierda"**: Para girar a la izquierda, se hace que un motor retroceda (atras m1) y el otro avance (adelante m2).



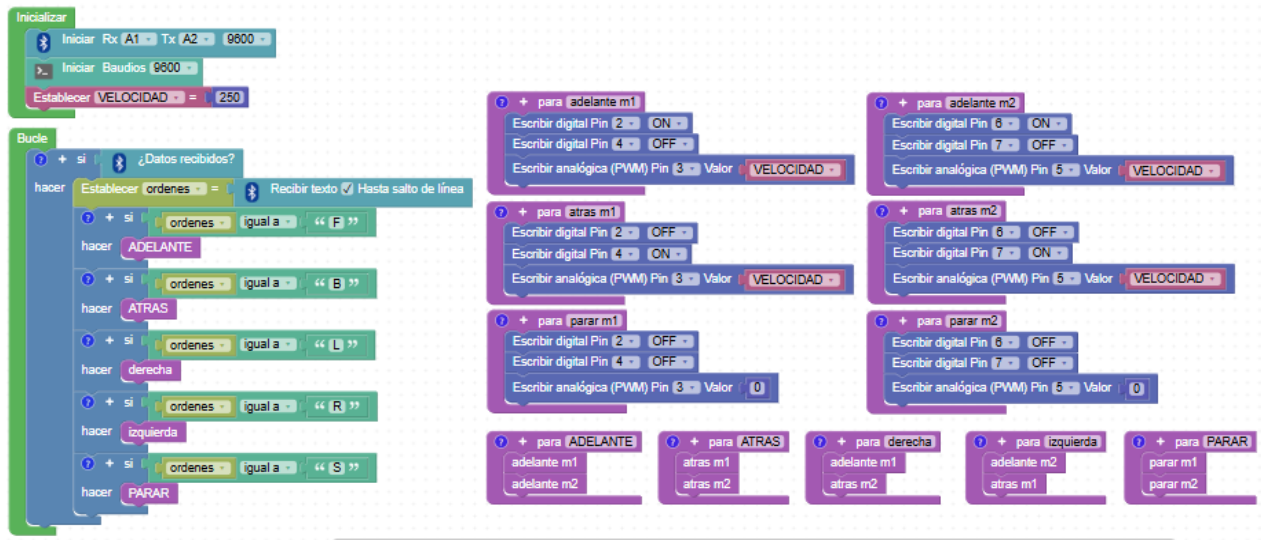
- **si ordenes = "PARAR"**: Para detener el coche, ambos motores se detienen (parar m1 y parar m2). Esto se logra configurando ambos pares de pines de control de dirección en OFF.



### Bloques de Control de Motor (Funciones Personalizadas):

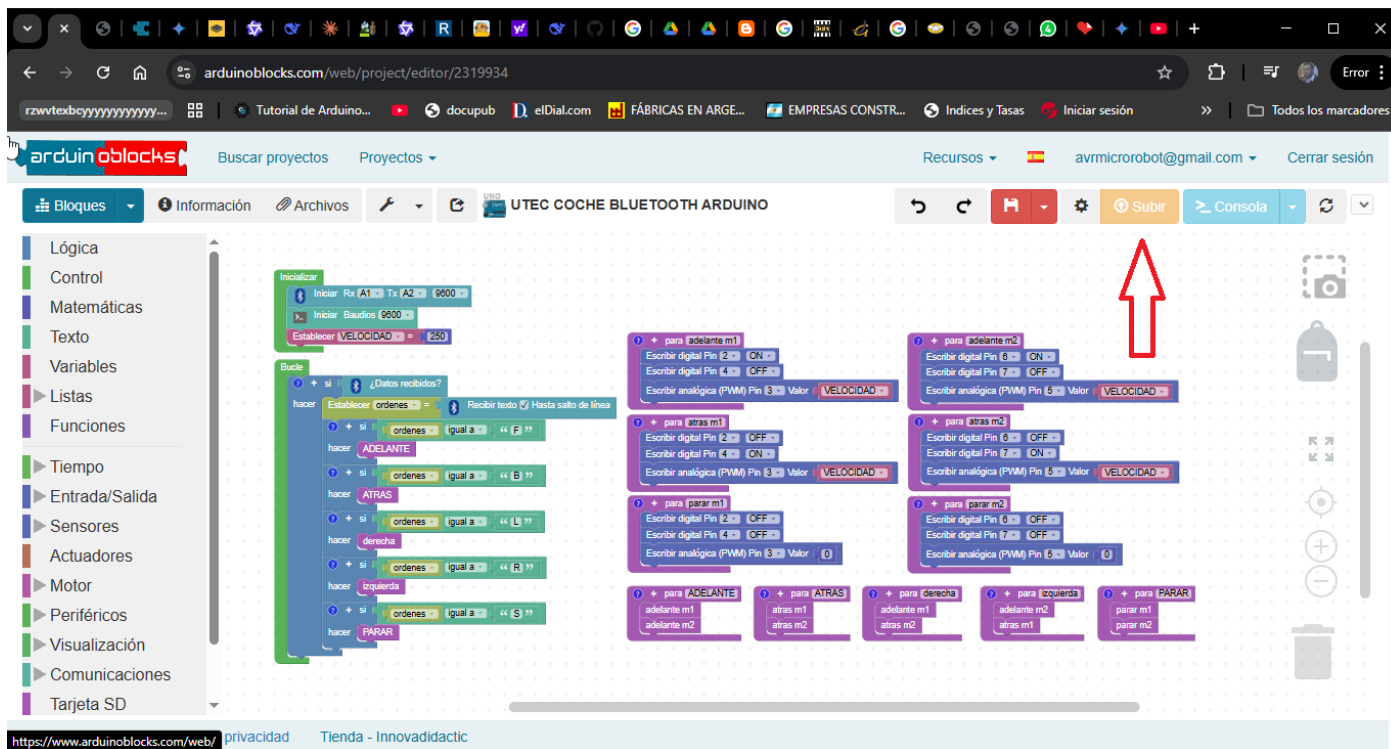
- Los bloques como para adelante m1, para atras m1, parar m1, para adelante m2, para atras m2, y parar m2 son probablemente funciones o bloques personalizados definidos dentro del entorno de Arduinoblocks. Estos bloques encapsulan la lógica para controlar la dirección y la activación/desactivación de cada motor individualmente. Internamente, como se ve en su expansión, manipulan los pines digitales y envían señales PWM a los pines de control de los motores (probablemente a través de un driver de motor).

### Código completo del carro controlador por Bluetooth



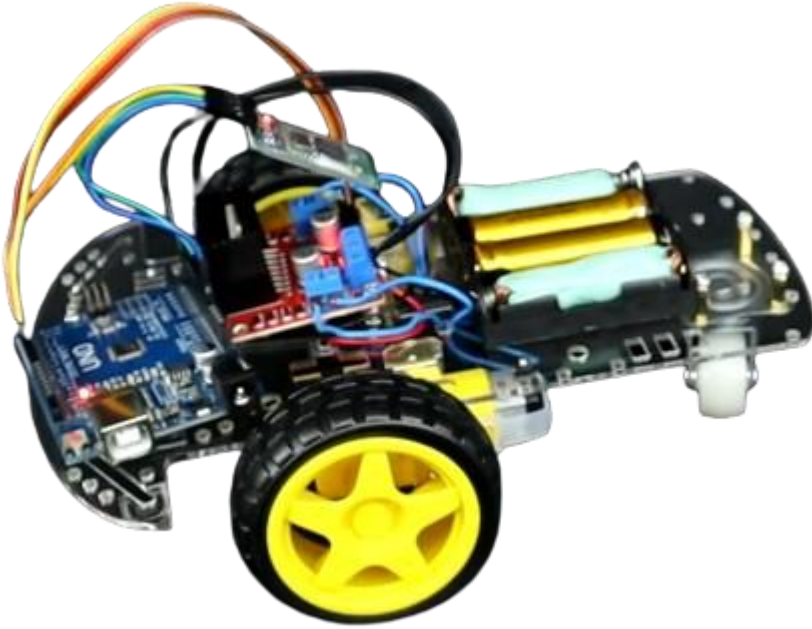
## Subir el Código a la Placa Arduino:

- Una vez que el código esté verificado y no haya errores, puedes subirlo a la placa Arduino. Busca un botón que diga "Subir" o tenga un icono de una flecha apuntando hacia la derecha. Durante el proceso de subida, verás luces parpadeando en tu placa Arduino. Una vez que termine, el programa se estará ejecutando en la placa.



**Ahora a Probar tu Proyecto del Carro controlado por Bluetooth !**





- Después de subir el código, desconecta la placa Arduino de la computadora (si es necesario para tu proyecto, como en el caso de un coche Bluetooth con alimentación externa) y prueba su funcionamiento. En el caso del coche Bluetooth, necesitarás una aplicación en tu teléfono o computadora que se conecte por Bluetooth al Arduino y le envíe los comandos ("ADELANTE", "ATRÁS", "derecha", "izquierda", "PARAR").

#### Consejos para los Alumnos de la UTEC que empiezan con Arduinoblocks:

- **Empieza con lo básico:** Comienza con proyectos sencillos como hacer parpadear un LED o controlar un pequeño motor.
- **Experimenta:** No tengas miedo de probar diferentes bloques y ver qué hacen. ¡La experimentación es una excelente forma de aprender!
- **Lee la documentación y busca tutoriales:** Hay muchos recursos en línea (tutoriales en video, guías escritas, foros) que te pueden ayudar a entender conceptos y resolver problemas.
- **Pide ayuda si te atascas:** No dudes en preguntar a profesores, amigos o en foros en línea si tienes alguna duda o problema.
- **¡Diviértete!** La programación puede ser muy creativa y gratificante. Disfruta del proceso de hacer que tus ideas cobren vida.

¡Felicitaciones para todos!