

# REGATA DE VELA POR RELEVOS



## STEMJAM Teaching Guide

Developing make spaces to promote creativity  
around STEM in schools

Acronym: STEMJAM

Project no. 2016-1-ES01-KA201-025470

[www.stemjam.eu](http://www.stemjam.eu)



Co-funded by the  
Erasmus+ Programme  
of the European Union

# REGATA DE VELA POR RELEVOS

## RESUMEN

La actividad consiste en la disputa de una competición por relevos en el mar, con barcos de vela, basada en la capacidad de los estudiantes para responder preguntas científicas.

La carrera se disputará entre dos equipos, que navegarán al mismo tiempo. Cada equipo de estudiantes tiene tres veleros: MB01, MB02, MB03 (tres mBots). A medida que avanza la carrera, el programa solicita a los estudiantes preguntas STEM en la pantalla de la computadora: si el equipo responde correctamente, su próximo barco puede comenzar a navegar y continuar la carrera. El equipo que llegue primero a la meta, ganará.

## OBJETIVOS DIDÁCTICOS

### GENERAL:

- ❖ Aprender a desarrollar el algoritmo de una actividad.
- ❖ Adquirir habilidades en la resolución de problemas.
- ❖ Adquirir motivación para el estudio y comprensión de los contenidos de las diferentes áreas de STEM.
- ❖ Aprender el funcionamiento de diferentes sensores y componentes.
- ❖ Desarrollar el pensamiento computacional.
- ❖ Aprender a trabajar y cooperar en grupo.

### TECNOLOGÍA:

- ❖ Aprender a programar la comunicación entre robots, usando mensajes infrarrojos.
- ❖ Aprender a programar en Arduino.
- ❖ Tener los primeros contactos con el lenguaje de programación Arduino.
- ❖ Hacer un programa que combine la operación de un robot vinculado a un programa de computadora, con otros robots autónomos, todos trabajando en equipo.
- ❖ Aprender a usar una pantalla de 7 segmentos, luces LED, componentes del ventilador y función de cronómetro.
- ❖ Cálculo de la corriente que fluye a través de una bombilla, cuando está conectada a una batería.
- ❖ Saber solventar problemas.

### MATEMÁTICAS:

- ❖ Resolver porcentajes y proporciones.

### CIENCIA:

- ❖ Cálculo de la velocidad en función del espacio y el tiempo.
- ❖ Cálculo de la densidad en función de la masa y el volumen.

Materia STEM:                      Ciencia                       Tecnología                       Ingeniería                       Matemáticas

Nivel educativo:                      12-14 años                       14-16 años



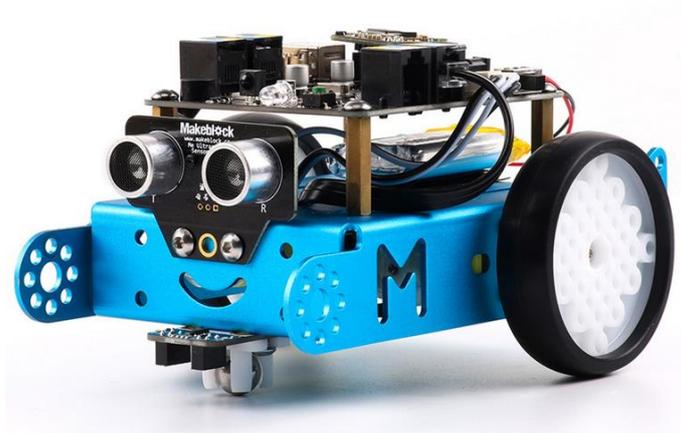
## PLANTEAMIENTO DEL PROBLEMA

Combinaremos diferentes programas y rutinas en uno. Por lo tanto, el diagrama de flujo es muy importante para el éxito de la actividad:

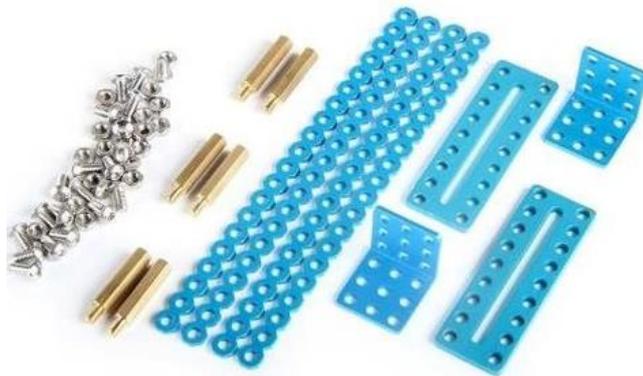
- ❖ Enfoque aleatorio a preguntas de diferentes sujetos.
- ❖ Acciones condicionadas a eventos.
- ❖ Movimiento, efectos luminosos, cronómetro, abanico...
- ❖ Robots autónomos con su programa grabado en la placa Arduino trabajando en equipo con robots vinculados a un programa de PC.
- ❖ Comunicación entre robots mediante el envío de mensajes infrarrojos.

## LISTADO DE MATERIALES

- (x6) mBots => Ref. 90054



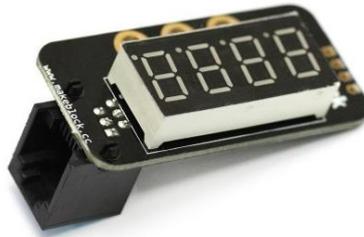
- ❖ Diferentes estructuras:



❖ (x6) Me 130 DC Motor Pack:



❖ (x2) Me Display 7 segmentos (4 dígitos. Rojo):



❖ Resto de Atrezzo: boyas para indicar el inicio y final de la etapa. Banderas para barcos.

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Am	Az	Bl	Am	Az	Bl	Am	Az	Bl	Ng	Am	Az	Bl	Ng		
Mbot Robot 2'4G			6																
Motor 1	Bl*																Bl*		
Motor 2	Bl*																	Bl*	
ME 130 motor pack 5V/10000rpm (FAN)	Az	6	6		Az														
ME 7-Segment serial display	Az	2	2											Az					
RJ25 cables			8																
Structures																			
Beam 0808-040			6																
Cutttable Linkage 080			6																
Bracket P1			8																
Laptops			2																
Atrezzo (not essential)																			

## DESCRIPCIÓN DE LA ACTIVIDAD

En esta actividad, cada equipo debe responder correctamente dos preguntas STEM. Si lo hacen, el primer velero (MB1) comenzará a navegar. Si no dan la respuesta correcta, el programa les hará otra pregunta. Este procedimiento continuará hasta que dos preguntas sean respondidas correctamente.

El tiempo comenzará a contar desde el inicio del programa. El tiempo se reflejará en una pantalla ubicada en el tercer mBot (MB3). Este tercer robot es el que cruzará la línea de meta.

El proceso se repite dos veces más:

- ❖ Después de acertar 2 preguntas más, el MB2 avanzará hacia el MB3.
- ❖ Finalmente, al responder las 2 últimas preguntas, el MB3 avanzará hacia la línea de meta. Cuando llegue, el cronómetro se detendrá.

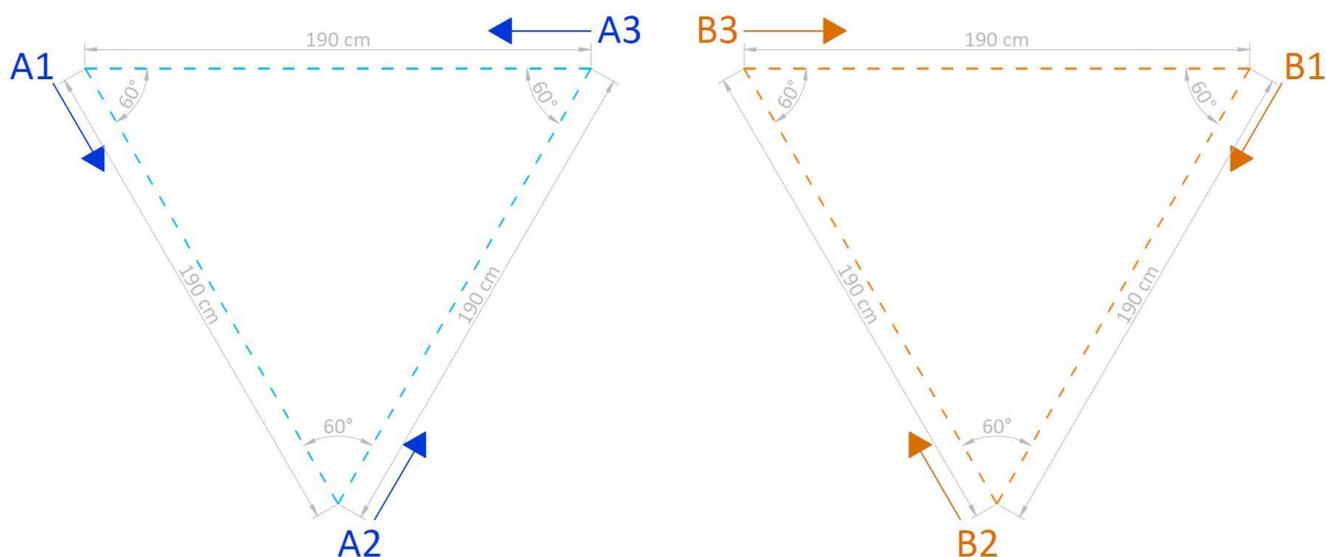
El programa principal se ejecutará desde un PC y se conectará al MB1 a través de la conexión Wi-Fi o *Bluetooth*.

Los robots MB2 y MB3 funcionarán de forma autónoma. Sus propios códigos tienen que ser grabados en sus placas Arduino.

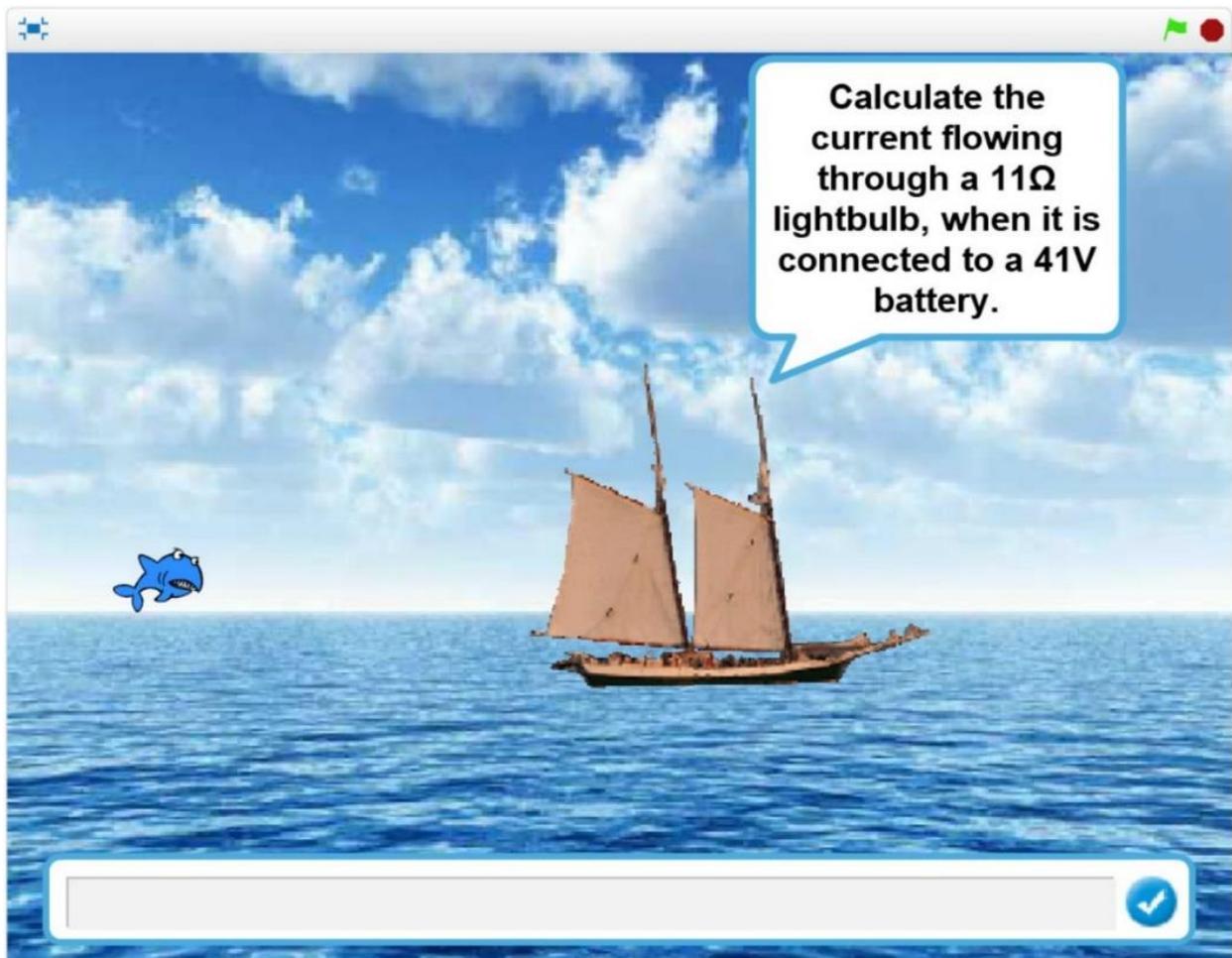
El MB1 realizará una conexión entre el programa de PC y los robots autónomos (MB2 y MB3) enviándoles mensajes para que ejecuten sus rutinas en el momento adecuado.

La configuración de movimientos y tiempos de espera se ha realizado de acuerdo con la ruta de la regata, que podemos ver en el siguiente esquema.

Cualquier cambio en la distancia de separación entre mBots significa que debemos ajustar estos parámetros en las rutinas de código afectadas.



PANTALLA PRINCIPAL – PROGRAMA PC:



Scratch 2.0 interface showing the code for the sailboat project. The code is organized into several message handlers:

- when green flag clicked:** Sets LED colors (red, green, blue) and sends a message to 'Message1'. It also sets initial variables for speed, acceleration, and velocity.
- Message1 received:** Adjusts the waiting time and sends a message to 'Message2'.
- Message2 received:** Turns the sailboat so that the message matches the LED color.
- Message3 received:** Sends a message to 'Message4'.
- Message4 received:** Sends a message to 'Message5'.

The code uses various Scratch blocks including 'set led on board', 'send mbot:amessage', 'set rotation style', and 'set rotation to'. It also includes a 'wait' block to control the timing of the messages.

La complejidad del código reside en cómo unir las diferentes rutinas sin afectar el rendimiento del programa.

Hay **3 programas básicos para cada equipo** (uno para cada uno de los tres robots de un equipo). Los programas de cada equipo deben ser diferentes, de modo que no haya interferencia entre los mensajes de los dos equipos.

La única diferencia entre los programas de los dos equipos son los mensajes que se envían. Todos los mensajes enviados por el robot MB1 del Equipo 1 terminan en 1 ("chrono1", "a1", "b1", "end1"). Lo mismo para el equipo 2, ("chrono2", "a2", "b2", "end2").

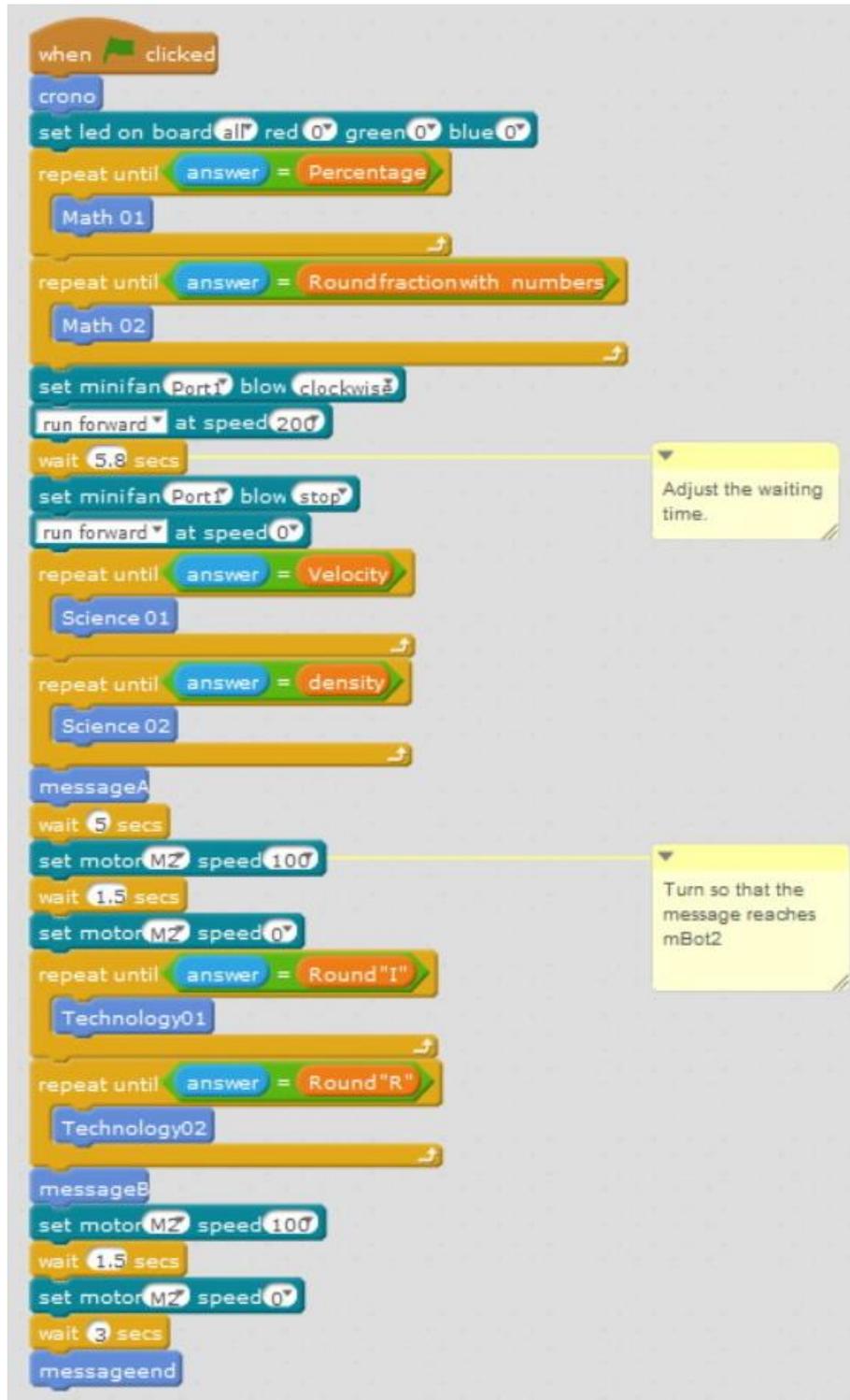
El programa 1, llamado "starter", es el programa principal que ejecutará el PC. Este programa gestiona las preguntas del concurso y las reacciones del MB1.

Los programas 2 y 3 se grabarán en las placas Arduino de MB2 y MB3. La ejecución de las rutinas de estos dos programas, depende de los mensajes enviados por el MB1, (se comunicará con ellos a través del puerto de infrarrojos). A continuación, se mostrarán las diferentes rutinas en orden de ejecución.

Los códigos mostrados son los del equipo 1. Los códigos del equipo 2 son idénticos y la única diferencia es que los mensajes terminan en "2" en lugar de en "1" y la orientación del turno de MB1.



## RUTINA 1: CÓDIGO PRINCIPAL (PROGRAMA PC)

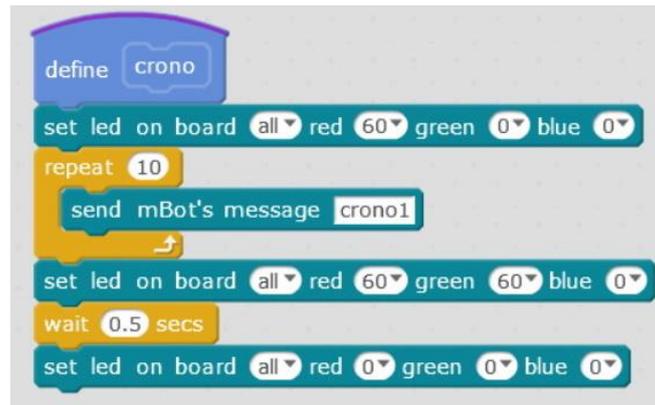


```
when clicked
  crono
  set led on board all red 0 green 0 blue 0
  repeat until answer = Percentage
    Math 01
  repeat until answer = Roundfractionwith numbers
    Math 02
  set minifan Port1 blow clockwise
  run forward at speed 200
  wait 5.8 secs
  set minifan Port1 blow stop
  run forward at speed 0
  repeat until answer = Velocity
    Science 01
  repeat until answer = density
    Science 02
  messageA
  wait 5 secs
  set motor M2 speed 100
  wait 1.5 secs
  set motor M2 speed 0
  repeat until answer = Round "I"
    Technology01
  repeat until answer = Round "R"
    Technology02
  messageB
  set motor M2 speed 100
  wait 1.5 secs
  set motor M2 speed 0
  wait 3 secs
  messageend
```

Adjust the waiting time.

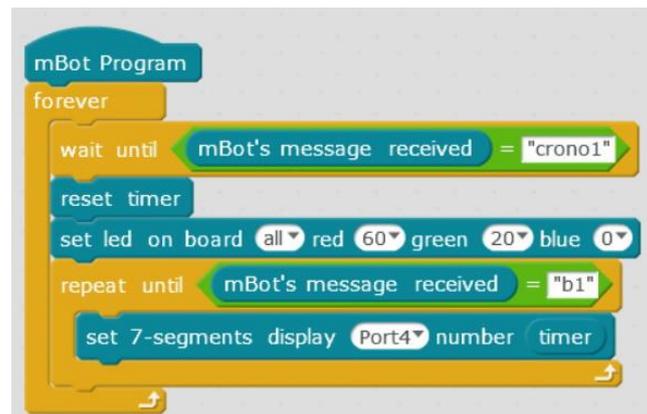
Turn so that the message reaches mBot2

## RUTINA 2: MENSAJE "crono" ENVÍA MBOT1 AL MBOT3



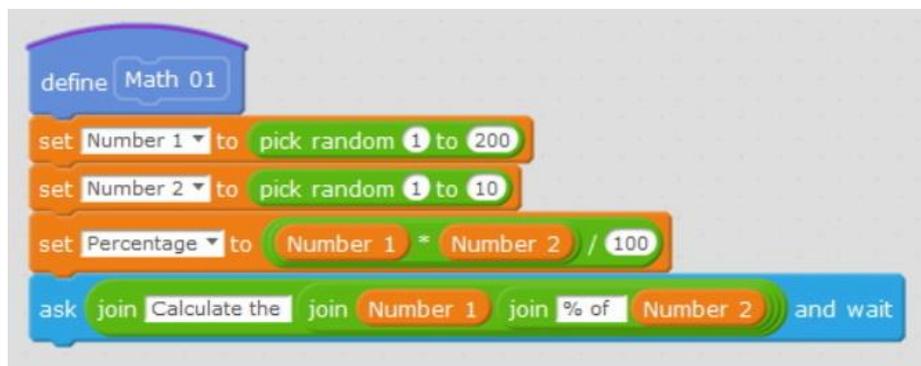
```
define crono
  set led on board all red 60 green 0 blue 0
  repeat 10
    send mBot's message crono1
  set led on board all red 60 green 60 blue 0
  wait 0.5 secs
  set led on board all red 0 green 0 blue 0
```

## RUTINA 3: CÓDIGO ARDUINO MBOT3 – PARTE 1 – MENSAJE "crono" RECIBIDO



```
mBot Program
forever
  wait until mBot's message received = "crono1"
  reset timer
  set led on board all red 60 green 20 blue 0
  repeat until mBot's message received = "b1"
    set 7-segments display Port4 number timer
```

## RUTINA 4: PREGUNTA MATH1



```
define Math 01
  set Number 1 to pick random 1 to 200
  set Number 2 to pick random 1 to 10
  set Percentage to Number 1 * Number 2 / 100
  ask join Calculate the join Number 1 join % of Number 2 and wait
```

## RUTINA 5: PREGUNTA MATH2

```
define Math 02
  set Numerator to pick random 1 to 10
  set Denominator to pick random 1 to 10
  set Number 3 to pick random 10 to 20
  set Fraction with numbers to Numerator * Number 3 / Denominator
  set Round fraction with numbers to round Fraction with numbers * 100 / 100
  ask join If I have join Numerator join / join Denominator join of join Number 3 , How much do I have? and wait
```

## RUTINA 6: PREGUNTA SCIENCE1

```
define Science 01
  set Space to pick random 1 to 100
  set Time to pick random 1 to 20
  set Velocity to round Space / Time * 100 / 100
  ask join If an object does join Space join m. every join Time seconds, what is its velocity? and wait
```

## RUTINA 7: PREGUNTA SCIENCE2

```
define Science 02
  set mass to pick random 100 to 200
  set volume to pick random 20 to 60
  set density to round mass / volume * 100 / 100
  ask join If an object has a join mass join kg mass and a join volume m3 volume, what is the value of its density? and wait
```

### RUTINA 8: MENSAJE "a" ENVIADO DEL MBOT1 AL MBOT2

```
define messageA
  set led on board all red 60 green 0 blue 0
  repeat 10
    send mBot's message a1
  set led on board all red 0 green 0 blue 20
  wait 0.5 secs
  set led on board all red 0 green 0 blue 0
```

### RUTINA 9: CÓDIGO ARDUINO MBOT2 - MENSAJE "a" RECIBIDO

```
mBot Program
  forever
    set led on board all red 0 green 0 blue 0
    wait until mBot's message received = "a1"
    set led on board all red 0 green 150 blue 0
    wait 0.5 secs
    set mini fan Port1 blow clockwise
    run forward at speed 200
    wait 6.5 secs
    set mini fan Port1 blow stop
    run forward at speed 0
```

### RUTINA 10: PREGUNTA TECHNOLOGY1

```
define Technology01
  set V to pick random 1 to 100
  set R to pick random 1 to 12
  set I to V / R
  set Round "I" to round I * 100 / 100
  ask join Calculate the current flowing through join R join Ω lightbulb, when it is connected to join V V battery. and wait
```

## RUTINA 11: PREGUNTA TECHNOLOGY2

```
define Technology 02
  set Force to pick random 1 to 200
  set Effort Arm to pick random 1 to 12
  set Resistance Arm to pick random 1 to 12
  set Resistance to Force * Effort Arm / Resistance Arm
  set Round "R" to round Resistance * 100 / 100
  ask join How much load weight can an effort of join Force join N lift with a classone lever, if the effort arm is join Effort Arm join metres long and the load arm is join Resistance Arm metres long? and wait
```

## RUTINA 12: MENSAJE "b" ENVAIDO DESDE MBOT1 AL MBOT3

```
define messageB
  set led on board all red 60 green 0 blue 0
  repeat 10
    send mBot's message b1
  set led on board all red 20 green 0 blue 150
  wait 0.5 secs
  set led on board all red 0 green 0 blue 0
```

Set the time to the advance time of the mbot3.

## RUTINA 13: CÓDIGO ARDUINO MBOT3 – PARTE 2 – MENSAJES "b" O "end" RECIBIDOS

```
mBot Program
  forever
    set led on board all red 0 green 0 blue 0
    wait until mBot's message received = "b1"
    set led on board all red 0 green 150 blue 60
    wait 0.5 secs
    set mini fan Port1 blow clockwise
    run forward at speed 200
    wait 6.5 secs
    set mini fan Port1 blow stop
    run forward at speed 0
    repeat until mBot's message received = "end1"
      set 7-segments display Port4 number timer
```

#### RUTINA 14: MENSAJE "end" ENVIA DESDE MBOT1 A MBOT3

```
define message end
set led on board all red 60 green 0 blue 0
repeat 10
  send mBot's message end1
set led on board all red 0 green 20 blue 150
wait 0.5 secs
set led on board all red 0 green 0 blue 0
stop all
```

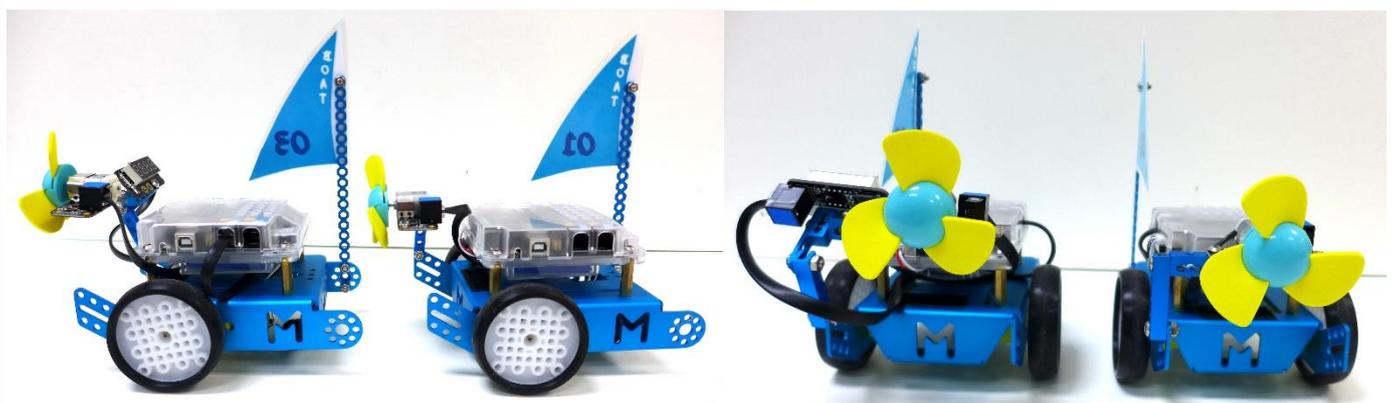
Grabaremos el código de mBot2 y mBot3 en sus placas Arduino. De esta manera, los mBots funcionarán independientemente de la computadora y solo necesitarán dos computadoras portátiles para controlar seis robots.

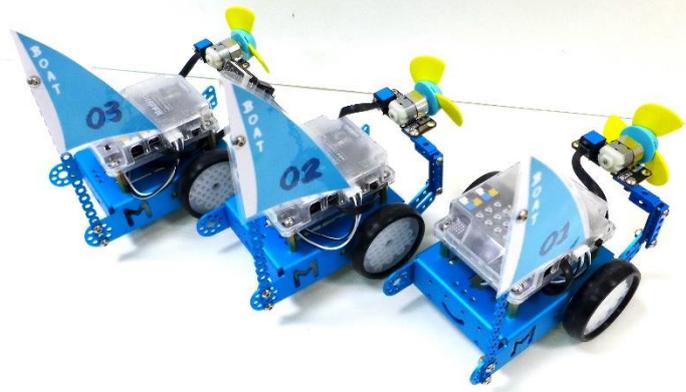
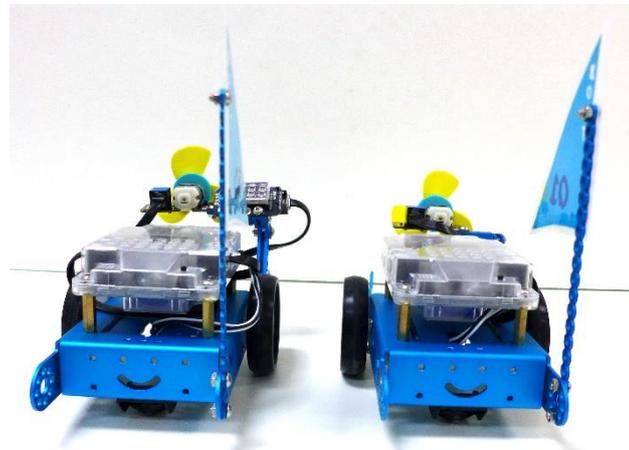
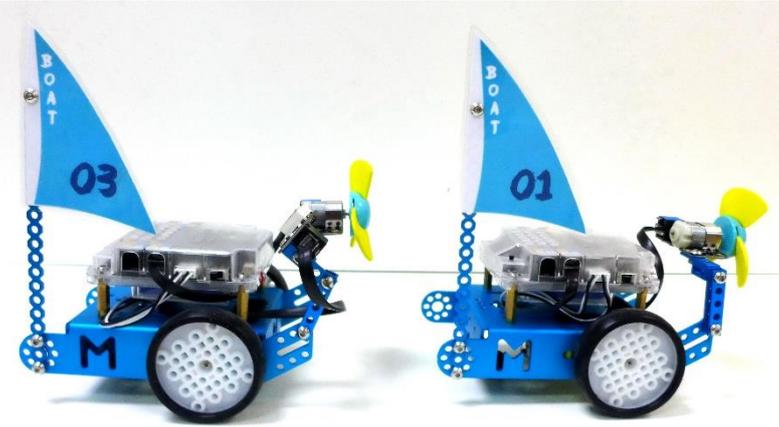
#### Cómo cargar un programa en la placa Arduino mBot usando mBlock:

El proceso será el siguiente:

1. Elija mBot en la pestaña *BOARDS* del menú mBlock.
2. Conecte el USB y elija "Puerto serie" en la pestaña de conexión.
3. En la pestaña de edición, elija "Modo Arduino" (en el programa que vamos a cargar, en lugar de la bandera verde, pondremos el comando azul "programa mBot").
4. Se abrirá una ventana con el código para grabarla en la placa Arduino de mBot. Usted puede, si lo desea, modificar su programa. Finalmente, haga clic en Cargar en Arduino.
5. Si no ha habido errores, se enviará un mensaje informando que el programa se ha grabado correctamente. En este momento, podrá comenzar a disfrutar el programa introducido en el robot, sin la computadora encendida. Para hacer esto, debe desconectar el cable USB y conectar las baterías (o batería de litio) del robot. Verás que tu mBot funciona independientemente.

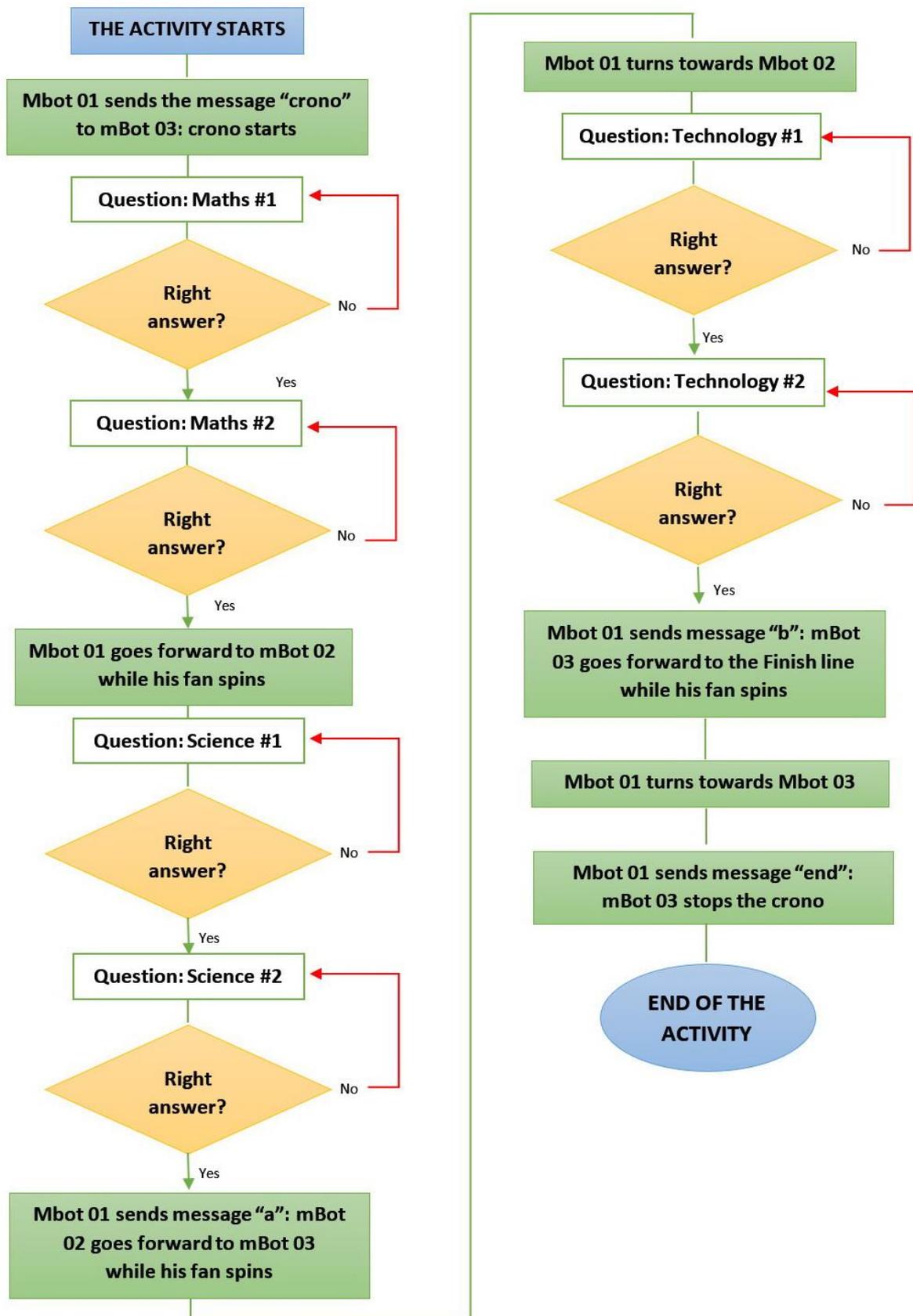
Una vez que se termina la programación, comenzamos a construir LA ESTRUCTURA, donde se establecerán todos los elementos mecánicos. También los elementos electrónicos.





# DIAGRAMA DE FLUJO

## SAILBOAT RELAY REGATTA



## EVALUACIÓN DE LOS ESTUDIANTES

Para la evaluación de los estudiantes en esta actividad, use la rúbrica de evaluación diseñada para este proyecto.

### MÁS INFORMACIÓN

#### **DIFICULTADES:**

1. Encontrar la manera de combinar las diferentes rutinas sin afectar el rendimiento del programa.
2. Transformar la idea inicial en un diagrama de flujo y éste en un código que funcione bien.
3. Hacer que 3 robots trabajen en equipo usando solo un PC.
4. Combinar seis preguntas aleatorias de diferentes temas en un solo código.
5. En muchas ocasiones algunos mBots no se mueven en línea recta.
6. La velocidad del mBot depende del nivel de carga de sus baterías.
7. Para que los mensajes de infrarrojos se reciban, deben enviarse repetidamente durante un período de tiempo (consulte el código). Si se envía un mensaje infrarrojo de manera oportuna, es normal que el receptor no lo reciba.
8. La señal infrarroja es muy sensible a la luz intensa. Para habilitar la recepción de mensajes de infrarrojos, la actividad mBot debe realizarse en un lugar sin luz solar directa y con una intensidad de luz moderada..
9. El transmisor / receptor de infrarrojos se encuentra en la parte frontal del mBot. Para habilitar la recepción de los mensajes, los robots deben colocarse en una posición lo más cerca posible (ver el boceto). Según nuestras pruebas, un ángulo de 60° y una distancia de 1'90 metros son parámetros que funcionan.
10. Durante la actividad, el transmisor mBot debe cambiar su posición para cumplir siempre con las condiciones anteriores.