

# EEST N° 5 “Roberto Noble”



## **MÓDULO TEÓRICO – PRÁCTICO DE TALLER - CICLO BASICO**

## **SISTEMAS TECNOLÓGICOS ELECTROMECANICA Y QUIMICA**

## **NIVEL 3**

*Jefe de departamento: Prof. ZELAYA Marcelo*

# Sistemas Tecnológicos (Electromecánica - Robótica)

## MUY IMPORTANTE:

Obtener el módulo de **SISTEMAS TECNOLOGICOS 2** ya que posee muchos conceptos básicos de electricidad.

## Circuitos eléctricos

### Ley de Ohm

La ley de **Ohm** dice que: "la intensidad de la corriente eléctrica que circula por un conductor eléctrico es directamente proporcional a la diferencia de potencial aplicada e inversamente proporcional a la resistencia del mismo".

$$\mathbf{I} = \frac{\mathbf{V}}{\mathbf{R}}$$

En el Sistema internacional de unidades:

**I** = Intensidad en amperios (A)

**V** = Diferencia de potencial en voltios (V)

**R** = Resistencia en ohmios ( $\Omega$ )

### Fórmulas despejadas de la Ley de Ohm:

$$\mathbf{V} = \mathbf{I} \cdot \mathbf{R}$$

$$\mathbf{I} = \mathbf{V} / \mathbf{R}$$

$$\mathbf{R} = \mathbf{V} / \mathbf{I}$$

**Ejemplo 1:** Calcular la resistencia de un artefacto eléctrico si al someterlo a una tensión de 12V circula una corriente de 0.02<sup>a</sup>.

**Solución:**

$$\mathbf{R} = \mathbf{V} / \mathbf{I}$$

$$\mathbf{R} = 600 \Omega$$

**Ejemplo 2:** Hállese la resistencia de una estufa que consume 3 amperios a una tensión de 120 voltios.

**Solución:** Aplicamos la ley de Ohm:

$$R = V / i$$

$$R = 120 \text{ v} / 3 \text{ A}$$

$$R = 40 \Omega$$

### Ejemplo 3:

- ¿Qué diferencia de potencial hay que aplicar a un reóstato de 30 ohmios para que circulen a través de él 5 amperios?

**Solución:** Aplicamos la ley de Ohm:

$$V = i \cdot R$$

$$V = 5 \cdot 30$$

$$V = 150 \text{ V}$$

## 6- Circuitos de corriente continua

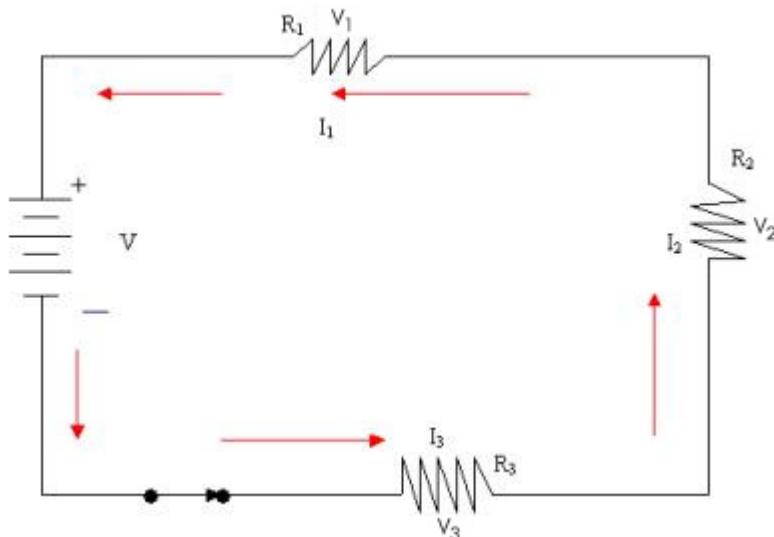
Es la asociación de elementos conductores que hace posible la circulación de una corriente eléctrica.

- En todo circuito eléctrico los consumos o resistencias son elementos que transforman la energía eléctrica en algún otro tipo de energía.
- Los elementos básicos de un circuito eléctrico son: conductor, fuente de energía, y uno o más consumos o resistencias.

Existen tres maneras de conectar resistencias en un circuito: **en serie, en paralelo y en forma mixta**. Dependiendo del tipo de conexión que presenten las resistencias será el comportamiento de la corriente y el voltaje en el circuito.

### a) Circuito en serie

En un circuito en serie las resistencias se conectan en forma sucesiva, de manera que en el camino entre una resistencia y la fuente de alimentación siempre hay otra resistencia que se interpone.



$$V_1 = i \cdot R_1; V_2 = i \cdot R_2; V_3 = i \cdot R_3$$

Donde  **$V_1$ ,  $V_2$ , y  $V_3$**  son las **tensiones** entre sus extremos respectivos e  **$i$**  la **intensidad** de corriente que las atraviesa, igual para todas ellas.

$$V = V_1 + V_2 + V_3 = i \cdot R_1 + i \cdot R_2 + i \cdot R_3 = i \cdot (R_1 + R_2 + R_3)$$

Si se aplica la Ley de Ohm a la asociación en su conjunto, se tiene:

$$V = i \cdot R_e$$

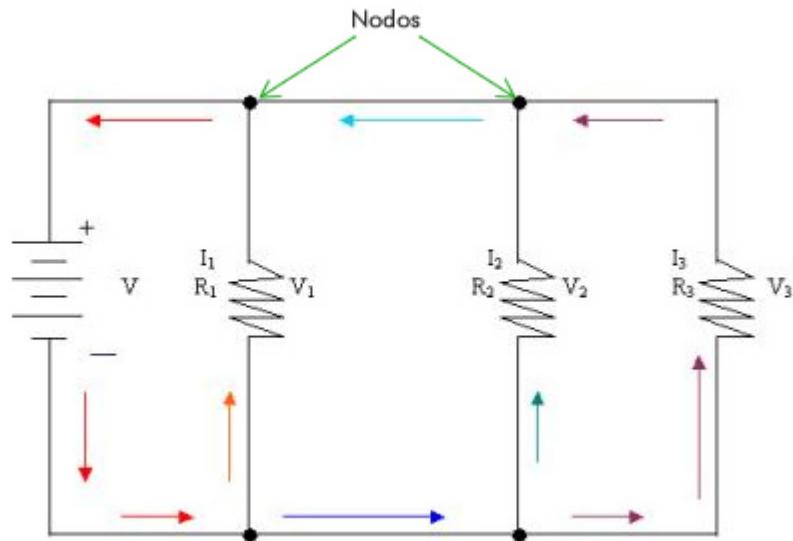
Por lo tanto para, el circuito en serie de resistencias  $R_1$ ,  $R_2$ ,  $R_3$ :

$$R_e = R_1 + R_2 + R_3$$

**Re → Resistencia equivalente:** es la suma de las resistencias individuales. Es decir la **resistencia total** del sistema.

### b) Circuito en paralelo

Este tipo de conexión corresponde a dos o más resistencias cuyos terminales llegan a los mismos nodos en cada extremo. Este tipo de conexión de resistencias se da, por ejemplo, en la instalación domiciliaria de los artefactos eléctricos (estos constituyen una resistencia eléctrica).



En una conexión de resistencias en paralelo:

- La suma de las intensidades que se generen en cada resistencia es igual a la intensidad total del circuito:

$$I_{\text{total}} = I_1 + I_2 + I_3 + \dots + I_n$$

- La tensión es igual en cada resistencia y es la que la fuente proporciona :

$$V_{\text{total}} = V_1 = V_2 = V_3 = \dots = V_n$$

La **resistencia equivalente** es tal, que su valor recíproco es la suma de los valores recíprocos de las resistencias individuales.

$$\frac{1}{R_e} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots + \frac{1}{R_n}$$

### Acometidas eléctricas:

Las acometidas eléctricas se clasifican por dos criterios básicos:



Esquema básico de una acometida eléctrica monofásica aérea en Baja Tensión.

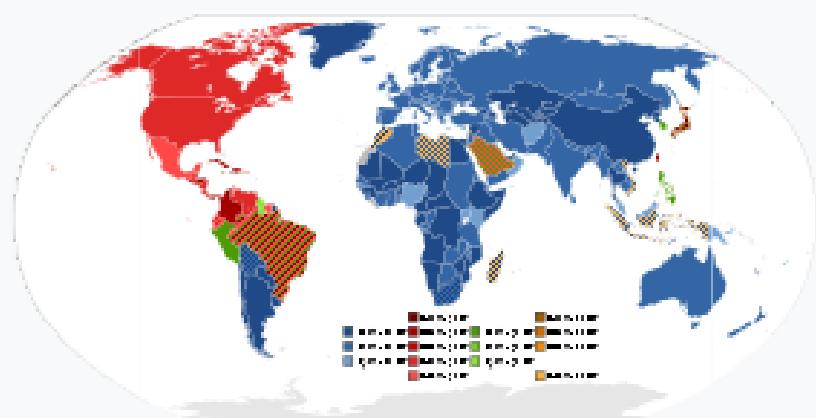
## 1. Según la Tensión:

1. **Baja Tensión**; 127 V, 200 V, 550 V, en general se consideran los límites superiores en 600 o 1000 V dependiendo del país y su normatividad interna.
  2. **Media Tensión**; 5 kV, 25 kV 40 kV, en general se considera el límite inferior en mayor a 600 o 1000 V según la normatividad del país.

## **2. Forma de acometida.**

1. **Acometida aérea**, cuando la entrada de cables del suministrador se da por lo alto de la construcción, normalmente por medio de una mufa y tubo, desde un poste de la red de suministro, en alta tensión los cables del suministro suelen ser llevados al usuario por tuberías enterradas para minimizar los peligros desde las redes aéreas de la empresa suministradora, pero cuando son aéreas es usual el uso de pórticos o torres.
  2. **Acometida subterránea**, cuando la entrada de cables del suministrador se da por debajo de la construcción, desde un registro o pozo de visita de la red de suministro.

## Tensiones en uso:



Tensión eléctrica y frecuencia de uso doméstico en el mundo   $U = 220 \text{ V-240 V}$ ,  $f = 60 \text{ Hz}$    $U = 220 \text{ V-240 V}$ ,  $f = 50 \text{ Hz}$    $U = 100 \text{ V-127 V}$ ,  $f = 60 \text{ Hz}$    $U = 100 \text{ V-127 V}$ ,  $f = 50 \text{ Hz}$

En general, en el contexto de corriente alterna, cuando se habla de tensión se refiere al valor eficaz de tensión salvo que se indique lo contrario. Se sigue esa práctica en este artículo.

Las bases para clavija de uso doméstico varían en forma pero los conductores energizados siempre son dos. Un conductor adicional proporciona conexión a tierra.

Toda Europa, Groenlandia y la mayor parte de América del Sur, junto a casi la totalidad de África, Asia y Oceanía, usan 220 V (con una tolerancia del 10 %). Los demás países, principalmente el resto de América, usan una tensión doméstica de 120 V o 127 V. En Japón excepcionalmente se usa 100 V.

En Japón, la tensión de la electricidad doméstica es 100 V. En el este y el norte de Honshū (incluyendo Tokio) y Hokkaidō se cuenta con una frecuencia de 50 Hz, mientras que al oeste de Honshū (incluyendo Nagoya, Osaka, e Hiroshima), Shikoku, Kyūshū y Okinawa se opera a 60 Hz. Para acomodar estas diferencias, los aparatos vendidos en Japón suelen ofrecer un botón para cambiar entre las dos frecuencias.

## Sistemas monofásicos y trifásicos:

Donde se usa 230 V o 127 V casi siempre se usa un sistema trifásico para distribución eléctrica doméstica y comercial. En Europa, 230 V es la tensión nominal entre fase y neutro, de forma que la tensión entre fases en condiciones ideales es  $230 \text{ V} \times \sqrt{3} \approx 398 \text{ V}$ . En México y otros países de América Latina, para electrificación doméstica y comercial se usa un [sistema trifásico](#) con tensión entre fases de 220 V; entre fase y neutro la tensión es  $220 \text{ V}/\sqrt{3} \approx 127 \text{ V}$ . En ambos sistemas descritos, las bases para clavija se conectan entre fase y neutro. Algunas cargas grandes como aparatos de aire acondicionado se conectan entre dos fases (sin usar el neutro) a 220 V o 230 V (según aplique). Se pueden conectar [motores trifásicos](#) usando las tres fases y sin usar el neutro.

En Estados Unidos el sistema es más complejo y problemático sin que esto resulte en un beneficio: se usan sistemas diferentes para uso doméstico y para uso comercial. Para uso doméstico el sistema es monofásico de fase dividida; la tensión nominal entre fases es 240 V, y entre cualquier fase y neutro es 120 V. Las cargas relativamente grandes como unidades de aire acondicionado se conectan entre fases. No se pueden conectar directamente motores trifásicos ni se puede cambiar el transformador por uno trifásico. Para comercios e industria ligera el sistema es trifásico con tensión entre cualquier fase y neutro de 120 V; entre fases es  $120 \text{ V} \times \sqrt{3} \approx 208 \text{ V}$ . Las cargas domésticas diseñadas para 240 V no se pueden usar en comercios con este sistema salvo que estén diseñadas para aceptar también 208 V o que se use un transformador adicional con el costo y pérdidas de energía que esto conlleva. Los sistemas usados en México y Europa no tienen esa desventaja. Peor aún, en Estados Unidos también existe un sistema eléctrico trifásico asimétrico alimentado por tres transformadores monofásicos en triángulo [aterrizado](#) en el devanado central de uno de ellos; la tensión entre fases es 240 V. La tensión entre fase y neutro es 120 V para dos fases y 208 V para la que resta; entre cualquier fase y neutro solo se puede conectar una carga pequeña.

## Interruptor diferencial



## Disyuntor Diferencial bipolar:

Un **interruptor diferencial (ID)**, también conocido como **RCD**, **RCCB** o **dispositivo diferencial residual (DDR)**, es un dispositivo electromecánico que se instala en las instalaciones eléctricas de [corriente alterna](#) con el fin de proteger a las personas de accidentes directos e indirectos provocados por el contacto con partes activas de la instalación (contacto directo) o con elementos

sometidos a potencial debido, por ejemplo, a una derivación por falta de aislamiento de partes activas de la instalación (contacto indirecto). También protegen contra los incendios que pudieran provocar dichas derivaciones.

Es un dispositivo de protección muy importante en toda instalación, tanto doméstica, como industrial, que actúa conjuntamente con la puesta a tierra de enchufes y masas metálicas de todo aparato eléctrico. De esta forma, el ID desconectará el circuito en cuanto exista una derivación o defecto a tierra mayor que su sensibilidad. Si no existe la conexión a tierra y se produce un contacto de un cable u elemento activo a la carcasa de una máquina, por ejemplo, el ID no se percata hasta que una persona no aislada de tierra toque esta masa, entonces la corriente recorrerá su cuerpo hacia tierra provocando un defecto a tierra y superando ésta la sensibilidad del ID, que disparará protegiendo a la persona y evitando así su electrocución.

## Funcionamiento:

En una instalación residencial o comercial es indispensable un interruptor diferencial de alta sensibilidad.

Si nos fijamos en la Figura 1, vemos que la intensidad ( $I_1$ ) que circula entre el punto a y la carga debe ser igual a la ( $I_2$ ) que circula entre la carga y el punto b ( $I_1 = I_2$ ) y por tanto los campos magnéticos creados por ambas bobinas son iguales y opuestos, por lo que la resultante de ambos es nula. Éste es el estado normal del circuito.

Si ahora nos fijamos en la Figura 2, vemos que la carga presenta una derivación a tierra por la que circula una corriente de fuga ( $I_f$ ), por lo que ahora  $I_2 = I_1 - I_f$  y por tanto menor que  $I_1$ .

Los transformadores de suministro eléctrico sujetos al régimen de neutro TT (95 % en España) tienen conectado a tierra su terminal neutro y por tanto se cierra circuito eléctrico en cuanto se pone en contacto cualquiera de los hilos de fase con tierra. Es aquí donde el dispositivo desconecta el circuito para prevenir electrocuciones, porque hay derivación de corriente hacia la toma de tierra que deben tener todos los elementos metálicos de los aparatos eléctricos.

La diferencia entre las dos corrientes de los hilos del suministro es la que produce un campo magnético resultante, que no es nulo y que por tanto producirá una atracción sobre el núcleo N, desplazándolo de su posición de equilibrio, provocando la apertura de los contactos  $C_1$  y  $C_2$  e interrumpiendo el paso de corriente hacia la carga, en tanto no se rearme manualmente el dispositivo.

Antes de rearmar el dispositivo se recomienda examinar la causa de su actuación y corregirla o habrá riesgo de prolongar una grave situación de inseguridad, de todas formas el sistema de **mecanismo libre** no dejará rearmar el ID hasta que no haya fuga a tierra menor que su sensibilidad ( $I_{\Delta n}$ ).

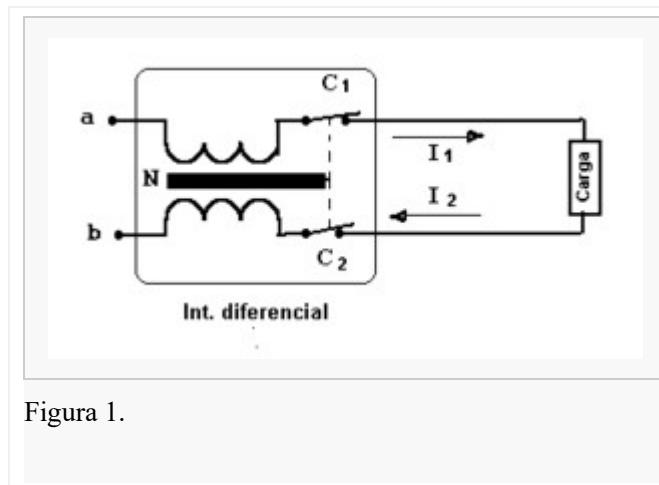


Figura 1.

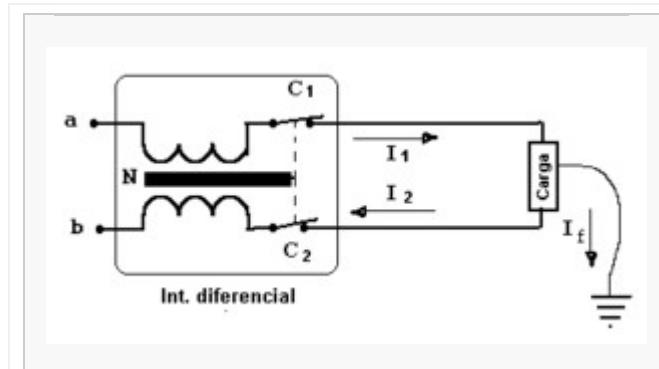
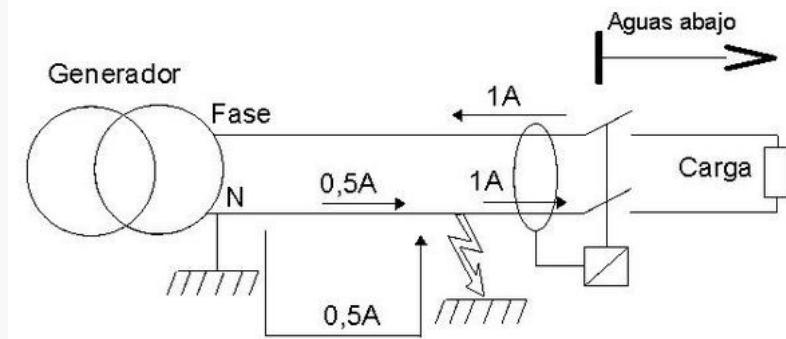


Figura 2.

Hay que tener en cuenta que estos dispositivos solo protegen aguas abajo del mismo, es decir, desde donde se conecte el diferencial hasta la carga. Este hecho lo podemos entender con la siguiente figura:



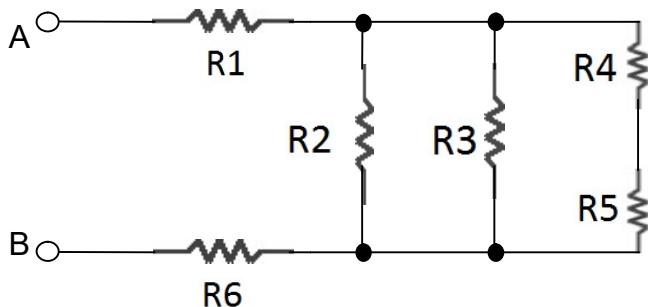
No detección con fallo aguas arriba del diferencial.

Vemos que, por ejemplo, al producirse un fallo en el aislante del cable (representado por un rayo), provoca una derivación a tierra que permitirá la circulación de una corriente desde la tierra conectada al neutro del generador, hasta el fallo producido por este. En el caso de que el fallo se produzca aguas arriba del mismo (entre éste y el transformador), el ID no entraría en funcionamiento, porque las corrientes entrante y saliente seguirían siendo iguales. Por esta razón se debe instalar lo más cerca posible del origen de la fuente de energía eléctrica, que en una vivienda sería el punto de entrada de la derivación individual en el local o la vivienda del usuario, para que la instalación quede totalmente protegida.

### **TRABAJO PRÁCTICO DE CALCULO DE RESISTENCIAS, PROTOBOARD Y MULTÍMETRO:**

Resistencias necesarias:

3 x 10 KΩ	3 x 1 KΩ	3 x 220 KΩ	3 x 47 KΩ
-----------	----------	------------	-----------

Circuito eléctrico a utilizar:

Circuito	R1	R2	R3	R4	R5	R6	RT Calculada [ Ω ]	RT Medida [ Ω ]	Diferencia [ Ω ]
1	10 KΩ	1 KΩ	220 KΩ	47 KΩ	220 KΩ	1 KΩ	11.9 KΩ	11.7 KΩ	0.2 KΩ
2	1 KΩ	10 KΩ	1 KΩ	47 KΩ	47 KΩ	47 KΩ			
3	10 KΩ	220 KΩ	10 KΩ	10 KΩ	1 KΩ	10 KΩ			
4	1 KΩ	1 KΩ	47 KΩ	10 KΩ	47 KΩ	47 KΩ			
5	220 KΩ	10 KΩ	220 KΩ	47 KΩ	10 KΩ	47 KΩ			
6	1 KΩ	1 KΩ	10 KΩ	10 KΩ	1 KΩ	10 KΩ			
7	47 KΩ	47 KΩ	47 KΩ	10 KΩ	220 KΩ	1 KΩ			
8	10 KΩ	220 KΩ	1 KΩ	220 KΩ	1 KΩ	220 KΩ			
9	10 KΩ	47 KΩ	220 KΩ	10 KΩ	47 KΩ	47 KΩ			
10	220 KΩ	10 KΩ	220 KΩ	47 KΩ	1 KΩ	1 KΩ			
11	1 KΩ	1 KΩ	10 KΩ	220 KΩ	47 KΩ	1 KΩ			
12	10 KΩ	220 KΩ	1 KΩ	47 KΩ	47 KΩ	47 KΩ			
13	220 KΩ	47 KΩ	220 KΩ	47 KΩ	1 KΩ	220 KΩ			
14	47 KΩ	220 KΩ	1 KΩ	1 KΩ	1 KΩ	10 KΩ			
15	220 KΩ	220 KΩ	220 KΩ	47 KΩ	47 KΩ	47 KΩ			
16	10 KΩ	47 KΩ	47 KΩ	47 KΩ	1 KΩ	1 KΩ			
17	1 KΩ	220 KΩ	47 KΩ	220 KΩ	47 KΩ	10 KΩ			
18	10 KΩ	1 KΩ	1 KΩ	10 KΩ	220 KΩ	1 KΩ			
19	1 KΩ	47 KΩ	1 KΩ	47 KΩ	47 KΩ	10 KΩ			
20	1 KΩ	47 KΩ	47 KΩ	220 KΩ	47 KΩ	1 KΩ			

Actividad a realizar:

- 1) En el Taller, calcular de manera analítica cada circuito con los valores indicados.
- 2) Luego, realizar el montaje de cada circuito en protoboard y medirlo con multímetro usando alguna de las escalas del óhmetro entre los bornes A y B. Y comparar resultados (calculado y medido).
- 3) Con los datos obtenidos en los puntos 1 y 2 se irá completando las 3 columnas de la derecha de la tabla de datos.

Presentación del trabajo: Todo se realizará en formato A4 **blanco ó cuadriculado**, con recuadro y escrito con letra normalizada en tinta. Dentro de una carpeta de vaina ó folio plástico.

El trabajo es individual y totalmente manuscrito.

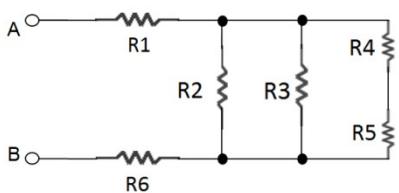
Podrán realizarse preguntas respecto al trabajo: ( cómo calculó?, cómo utilizo el protoboard?, cómo midió con multímetro?, etc)

Contenido: Carátula, Tabla de datos completa y 1 página por cada circuito.

OBSERVACION: Cada página debe contener en esquema del circuito con los valores utilizados y el cálculo de resistencia total, tal cual se muestra en la siguiente hoja.

Circuito Nº 1:

R	Valor [KΩ]	Valor [Ω]
10		



$R1$	$10K\Omega$	$10000\Omega$
$R2$	$1K\Omega$	$1000\Omega$
$R3$	$220K\Omega$	$220000\Omega$
$R4$	$47K\Omega$	$47000\Omega$
$R5$	$220K\Omega$	$220000\Omega$
$R6$	$1K\Omega$	$1000\Omega$

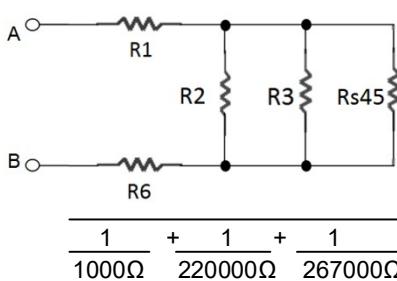
Cálculo de la resistencia equivalente entre  $R4$  y  $R5$  (la llamaremos  $Rs45$ )

$$Rs45 = R4 + R5$$

$$Rs45 = 47000\Omega + 220000\Omega$$

$$Rs45 = 267000 \Omega$$

Redibujando en circuito:



Se calcula la resistencia equivalente en paralelo entre  $R2$ ,  $R3$  y  $Rs45$ .

$$Rp = \frac{1}{\frac{1}{R2} + \frac{1}{R3} + \frac{1}{Rs45}}$$

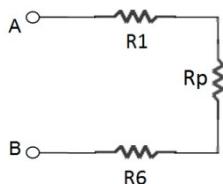
$$Rp = \frac{1}{0.001 \Omega + 0.00000454 \Omega + 0.00000374 \Omega}$$

1

$$Rp = \frac{1}{0.00100828\Omega}$$

$$Rp = 991.7 \Omega$$

Redibujando en circuito:



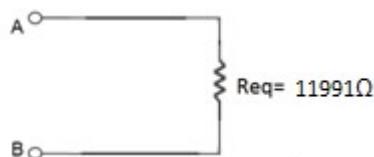
Finalmente en el circuito quedan 3 resistencias en serie. La resistencia equivalente total se calcula de la siguiente manera:

$$Req = R1 + Rp + R6$$

$$Req = 10000 \Omega + 991 \Omega + 1000 \Omega$$

$$Req = 11991 \Omega = 11.9 \text{ K}\Omega$$

El circuito equivalente final queda de la siguiente manera:



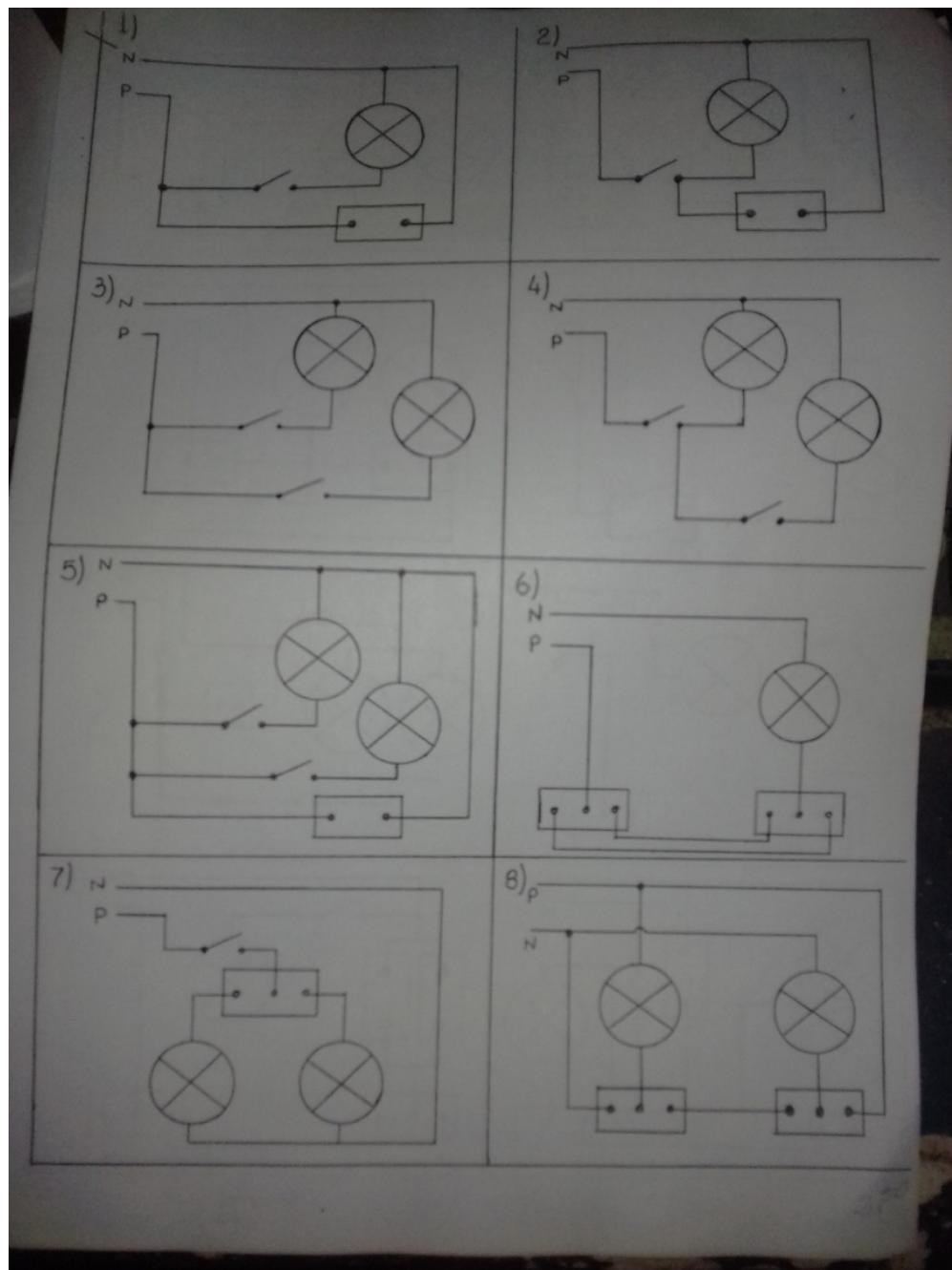
**OBSERVACIÓN:** Algunos valores pueden ser diferentes debido a la cantidad de decimales utilizados. Usar luego de la coma 3 decimales.

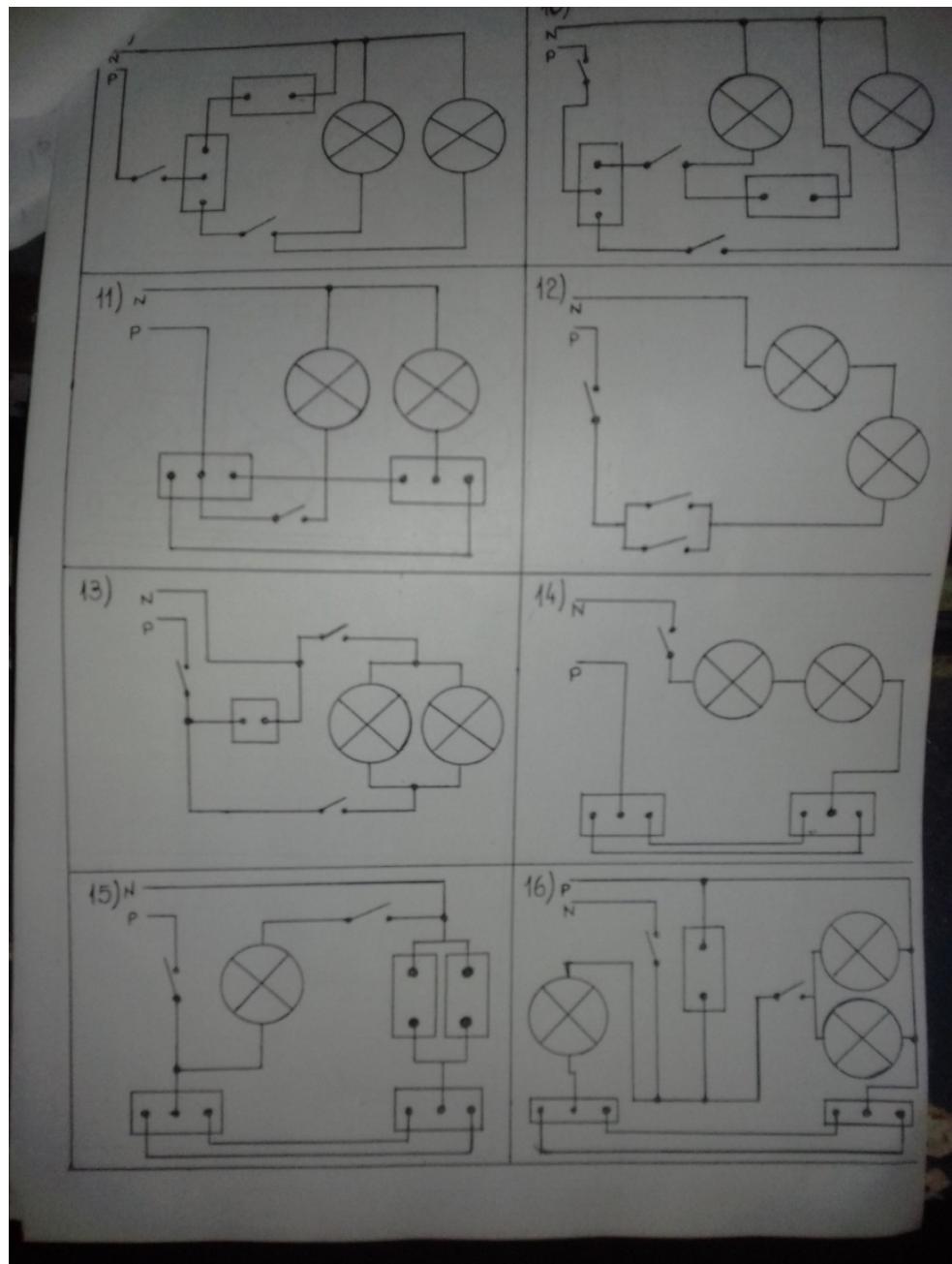
## **INTERPRETACIÓN, DISEÑO Y MONTAJE DE CIRCUITOS ELÉCTRICOS:**

Para cada circuito realizar los planos correspondientes que contengan el esquema eléctrico y el esquema de cableado. Además para cada circuito debe realizar el montaje eléctrico para su respectiva evaluación.

Los planos realizarlos en formato A4.

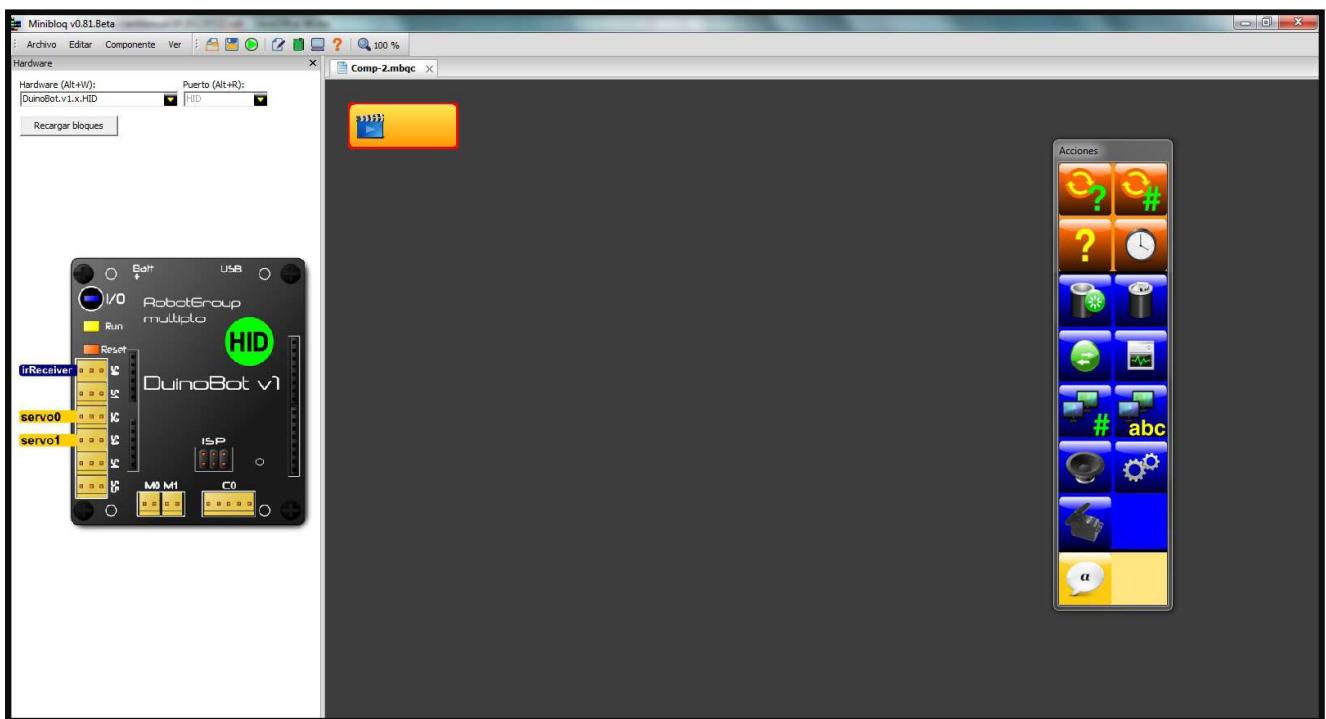
Recordar que la alimentación ingresa por la caja octogonal de techo, las lámparas se colocan dentro de cajas octagonales, tomacorrientes y teclas en caja rectangulares.



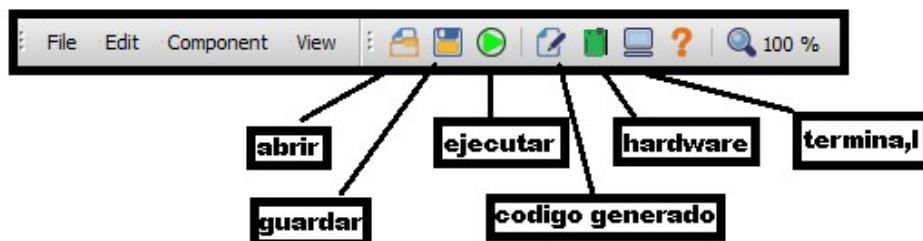


## ROBÓTICA - MINIBLOQ.v0.8 1.Beta .

A continuación se puede ver una captura de pantalla del entorno cuando recién inicia. Según las diferentes versiones podrán cambiar levemente pero en general son prácticamente iguales.



### Menú y barra rápida de herramientas



### Hardware

Esta ventana permite al usuario seleccionar el tipo de placa o controlador que se desea programar (con lo cual Minibloq carga automáticamente los bloques que son compatibles con el hardware seleccionado). Es también en esta ventana donde el usuario selecciona el puerto de comunicaciones (que puede ser cualquier puerto serie tanto físico como virtual montado sobre un bus USB o sobre Bluetooth) para interactuar con el hardware. Verifique que sea el hardware seleccionado el mismo que la placa que trabaje. DINOBOT 1.2 HID ó DUINOBOT 2.3 HID.

### Editor de Componentes

Los programas realizados con Minibloq se llaman *Componentes*. El Editor de Componentes es por tanto la zona donde el usuario puede agregar los bloques que forman el *Componente*. En la misma es posible cortar y pegar bloques, desplazarlos, hacer zoom, etc. El Editor de Componentes soporta navegación por teclado para seleccionar el bloque actual y otras operaciones, así como también permite realizar ciertas operaciones (Principalmente zoom y desplazamiento) utilizando la rueda del mouse y diferentes combinaciones de teclas. Los bloques son además inventados automáticamente a medida que el usuario los va colocando.



### Selector de acciones (Actionspicker)



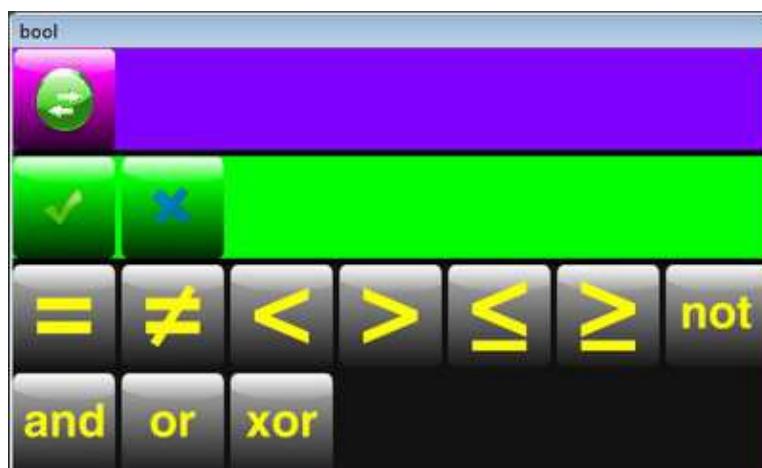
Es la barra con bloques que permite ir agregando al programa (o componente) los bloques que representan comandos imperativos, o acciones (como por ejemplo la escritura de una salida digital, o de una salida analógica, o el establecimiento de la velocidad de un motor). Entre otras cosas, muestra pequeños “tooltips” de ayuda cuando el usuario se posiciona sobre un bloque por cierto corto intervalo de tiempo con el cursor, y sus botones cambian de color al moverse sobre ellos con el mouse. Esta ventana es flotante, y está siempre visible. Cada nuevo bloque se insertará con un solo click de mouse, y se hará a continuación del “bloque actual”, que es el seleccionado con un pequeño rectángulo rojo en el Editor de componentes.

### Electores contextual es (Contextualpickers)

Una de las características más importantes de Minibloq es que presenta los bloques de forma *contextual*. Esto significa que cuando el usuario va a agregar un bloque como parámetro de otro, Minibloq sólo le mostrará aquellos del mismo tipo de datos. De esta forma, no sólo la interfaz de usuario es más limpia y clara, sino que se reducen las posibilidades de errores y el tiempo de búsqueda de cada bloque, ya que el usuario no debe seleccionar el bloque entre todos los bloques, sino sólo entre los que podría colocar. A continuación se muestra el selector contextual para bloques que devuelven números (tipo de datos *number*, en Minibloq):



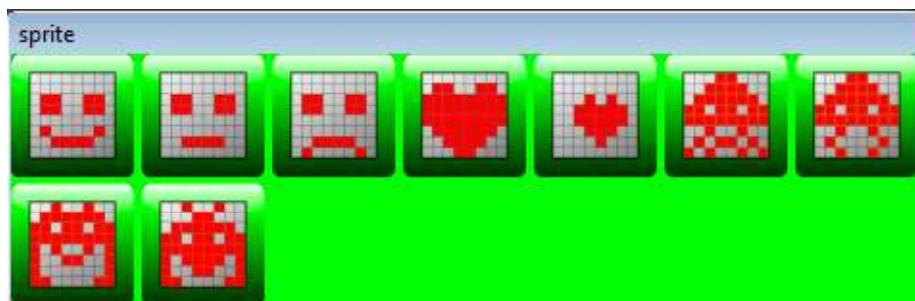
Y el siguiente es el selector de bloques que devuelven un valor booleano (tipo de datos *bool*):



Para poder enviar texto por el puerto serie, la versión v0.81.Beta dispone del selector de constantes de texto. Además de texto en sí

mismo, es posible seleccionar “emoticones”, los cuales son mostrados gráficamente por la terminal de Minibloq.v0.81.Beta:

Por último, en placas con display de matriz de puntos, tales como DuinoBot.Kids, existe también un selector para los llamados *sprites* gráficos:

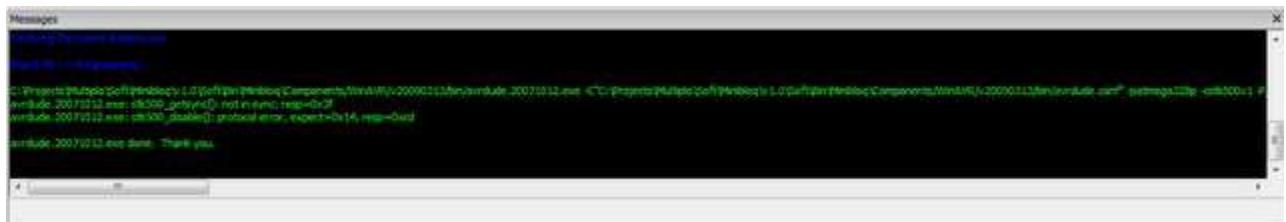


Algunas características importantes de los selectores (tanto del de acciones como de los contextuales), es que dividen automáticamente por color los bloques de acuerdo a la función que cumplen. Por ejemplo en el de bloques numéricos, la línea violeta es la de los “getters” (bloques que devuelven un valor de un método de una instancia, o de funciones del sistema), la línea verde es de constantes, la gris oscura de operaciones, y la gris clara de sensores específicos. Esta división el sistema la hace en *run-time*, durante la carga inicial de los archivos XML que describen a los bloques

### Ventana de mensajes

La ventana de mensajes tiene 2 zonas: Una de texto para los mensajes en sí, y una barra de progreso, para indicar el avance de tareas como la carga de bloques al seleccionar otra placa controladora, etc. Esta ventana muestra el texto en diferentes colores:

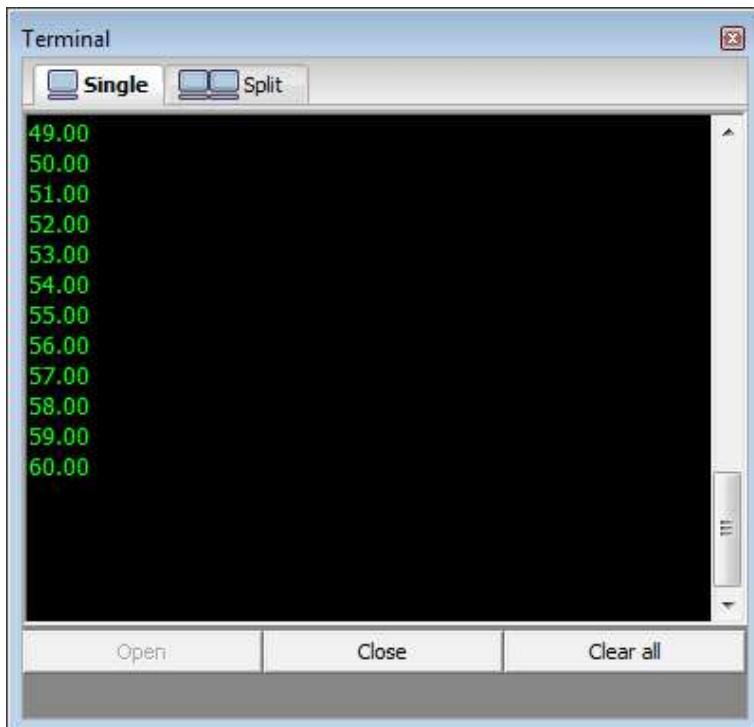
- Los comandos generados o ejecutados por Minibloq, tales como una llamada a un compilador se muestran en verde.
- La salida devuelta por herramientas externas llamadas por Minibloq se presenta en azul cuando fue exitosa.
- Si hubo errores en la salida producida por herramientas externas, el texto es rojo.



La ventana de mensajes además cuenta con zoom (utilizando la rueda del mouse y la tecla **Ctrl**). Para ocultar o visualizar la ventana de mensajes se puede, o bien presionar **Alt+M**, o bien utilizar el menú **Ver->Mensajes**.

### Terminal

Es frecuente que los programas (componentes) envíen datos por el puerto USB (generalmente configurado como puerto serie virtual, o dispositivo USB CDC en las placas controladoras soportadas por Minibloq). Para visualizar estos datos en tiempo real, Minibloq cuenta con un terminal embebido:

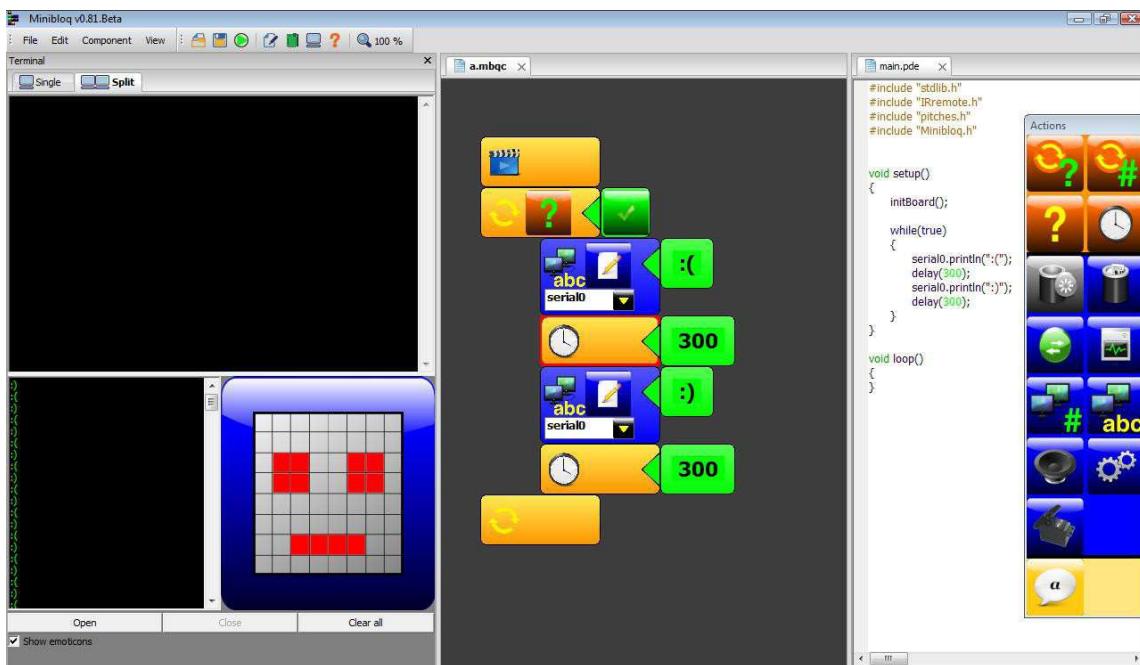


El terminal puede ser visualizado ya sea utilizando las teclas **Alt + T**, o mediante el menú **Ver->Terminal**. La misma (como se puede ver en la figura) tiene botones para abrir y cerrar el puerto, lo que permite dejar el control del puerto a otras aplicaciones sin cerrar el terminal.

Además, al bajar un nuevo programa a una placa, la terminal se desconecta automáticamente del dispositivo (cierra el puerto), de modo que su uso está completamente integrado al entorno, resultando así muy cómoda. Algunas características adicionales de la ventana de terminal son las siguientes:

- Muestra el texto recibido en verde y el enviado en azul.
- Posee zoom (rueda del mouse combinada con la tecla **Ctrl**).
- Presenta dos formas de visualizar los datos: Una donde el texto enviado y recibido están en la misma área de texto, y otra (*split*) donde están divididas (algunos usuarios pueden preferir una u otra, dependiendo también de lo que estén realizando).

Por último, queremos mencionar que el terminal de Minibloq soporta algunos "emoticones" gráficos. La idea de incluir esto surgió de la buena experiencia en aula que produjo el controlador DuinoBot.Kids, que cuenta con un pequeño display gráfico de matriz de LEDs. Al no contar con esto la mayoría de los controladores soportados por Minibloq, se decidió emular esta capacidad, al menos en lo que respecta a los emoticones ("sprites", como se los llama en el display). Aquí se puede ver la terminal mostrando estos gráficos. Para esto debe seleccionarse la solapa "Doble" (o "Split" en la versión en inglés del software):



### Visor del código generado

```
main.pde x
#include "stdlib.h"
#include "IRremote.h"
#include "pitches.h"
#include "Minibloq.h"

void setup()
{
    initBoard();
    float count = 0;
    while(true)
    {
        serial0.println(count);
        count = (count+1);
        delay(300);
    }
}

void loop()
```

El usuario puede ver el código generado siempre que lo desee, en su editor de texto preferido, ya que todos los archivos de código creados automáticamente son de texto plano (Minibloq genera tanto los archivos de texto como los binarios -hex- y los coloca en un subdirectorio del directorio donde el usuario ha guardado el componente en el que está trabajando). Pero además, Minibloq brinda la posibilidad de visualizar todo el código generado en **tiempo real**. Ésta (la generación en tiempo real del código) es una característica única con respecto a otros entornos open source de programación visual

para computación física (hasta donde sabemos al menos). La ventana del visor de código generado es también acopiable, y se la puede ver/ocultar con las teclas rápidas **Alt + G** o con el menú **Ver->Código generado**. A continuación se puede ver una captura de pantalla:

Como se puede apreciar en la imagen, la ventana cuenta con coloreo de sintaxis, y marca además con puntos verdes sobre el margen izquierdo el último código que fue modificado (coloca un punto verde por cada línea entre la primer línea modificada y la última). También es posible hacer zoom, dando foco a la ventana con un click de mouse y luego utilizando la rueda del mouse en combinación con la tecla **Ctrl**.

### Ciclos



**While:** El ciclo *while* de Minibloq es equivalente al *while* de C/C++, recibiendo como único parámetro una expresión booleana. Todos los bloques entre él y el próximo bloque de fin de ciclo se repetirán mientras se cumpla la condición especificada aquí. El siguiente ejemplo incrementa una variable, que luego utiliza para generar una frecuencia en el buzzer, mientras dicha variable sea menor que 25:

The screenshot shows the Minibloq software interface. On the left, a Scratch-like script is displayed with various blocks: a green control block, a blue control block with a repeat loop containing a yellow sound block and a blue control block, and a blue control block with a note output. On the right, the generated C++ code is shown:

```
#include "stdlib.h"
#include "IRremote.h"
#include "pitches.h"
#include "Minibloq.h"

void setup()
{
    initBoard();

    float note = 0;
    while((note<25))
    {
        toneWithDelay(BuzzerPin, (note*50), 200);
        note = (note+1);
    }
}

void loop()
{}
```

 **Repeat:** El ciclo repeat se asemeja al *for* de C/C++ (y de hecho genera un *for* en el código), pero no es directamente equivalente. Este parámetro puede de todos modos no ser un simple número sino una expresión numérica más compleja, donde también podrían utilizarse variables que sean modificadas adentro del propio ciclo. Pero en general es utilizado para repeticiones sencillas, donde se conoce a priori cuántas iteraciones se harán. Todos los bloques entre él y el próximo bloque de fin de ciclo se repetirán tantas veces como indique el parámetro numérico. A continuación puede verse un ejemplo sencillo con el código que genera:

The screenshot shows the Minibloq software interface. On the left, a Scratch-like script is displayed with a green control block, a blue control block with a repeat loop containing a yellow sound block, and a blue control block with a note output. On the right, the generated C++ code is shown:

```
#include "stdlib.h"
#include "IRremote.h"
#include "pitches.h"
#include "Minibloq.h"

void setup()
{
    initBoard();

    for(unsigned int _i=0; _i<(unsigned int)(5); _i++)
    {
        toneWithDelay(BuzzerPin, NOTE_C4, 300);
        toneWithDelay(BuzzerPin, NOTE_B4, 200);
    }
}

void loop()
{}
```

La variable `_i` del código es interna y el usuario no puede accederla en la versión actual de Minibloq, decisión de diseño tomada en base a la idea de mantener la sencillez del entorno, y al hecho de que el bloque repeat es independiente del lenguaje que se genera (como se dijo antes, Minibloq podría generar otras sintaxis más allá del C/C++ en el futuro). De esta forma, el repeat no es un ciclo *for*, sino precisamente un *repeat*. El *for*

visto aquí en el código generado es simplemente la **implementación C/C++** de este concepto.



**Decisiones (if):** Los bloques de decisiones en base a condiciones booleanas (if) son equivalentes a las construcciones *if-else* de C/C++. La única salvedad es que generan siempre el *else* en la versión actual (de forma similar a como lo hacen otros entornos de programación gráfica). A continuación se puede ver el bloque en uso, junto con el código que genera:

The screenshot shows two windows side-by-side. On the left is the Minibloq interface with a project titled "IRMotorControl.mblk". The workspace contains several blocks: an "init" block, a "code" block, a decision block with conditions "code == 10" and "code > 0", and a "motor0" block. On the right is a code editor window titled "main.pde" containing the generated C++ code:

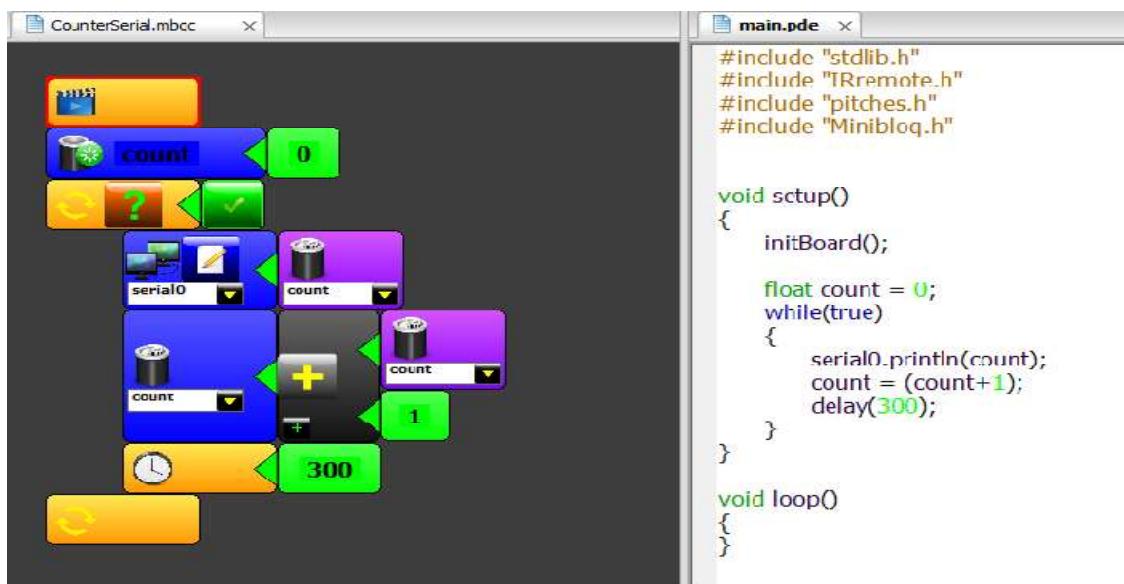
```
#include "stdlib.h"
#include "IRremote.h"
#include "pitches.h"
#include "Minibloq.h"

void setup()
{
    initBoard();
    float code = 0;
    while(true)
    {
        code = irReceiver.getIRRemoteCode();
        if((int)(code)==(int)(10))
        {
            motor0.setSpeed(0);
        }
        else
        {
            if((code>0))
            {
                motor0.setSpeed((max(code, 1)*10));
            }
            else
            {
            }
        }
        delay(100);
    }
}

void loop()
{}
```

Numero. Minibloq posee en su versión actual variables numéricas. Se han tomado algunas decisiones de diseño con respecto al sistema de variables, para impulsar buenas prácticas de programación. Por ejemplo:

Las variables requieren **declaración e inicialización**. Como se puede ver en la imagen a continuación, hay un bloque específico para declarar variables, y éste requiere inicialización obligatoria. Como ya se explicó antes, Minibloq tiene varias características para ayudar al usuario cuando renombra una variable, o si borra una declaración, renombrando automáticamente las demás ocurrencias de la variable en el programa, o notificando de errores (por borrado por ejemplo) en tiempo real. La siguiente imagen muestra un sencillo programa con una variable que funciona como contador. Allí se puede apreciar el bloque de declaración / inicialización (o *varInit*), y los bloques de asignación y acceso a la variable:



The screenshot shows the MBCC (Microcontroller Block-based Code) IDE interface. On the left, a Scratch-like script titled "CounterSerial.mbcc" is displayed. It starts with a green flag, followed by a "set count to 0" block, a "when green flag is pressed" hat block containing a "repeat until [count >= 100]" loop. Inside the loop, there's a "print count" block, an "add (1) to count" block, and a "wait (300)" block. On the right, the corresponding Arduino code is shown in the "main.pde" file:

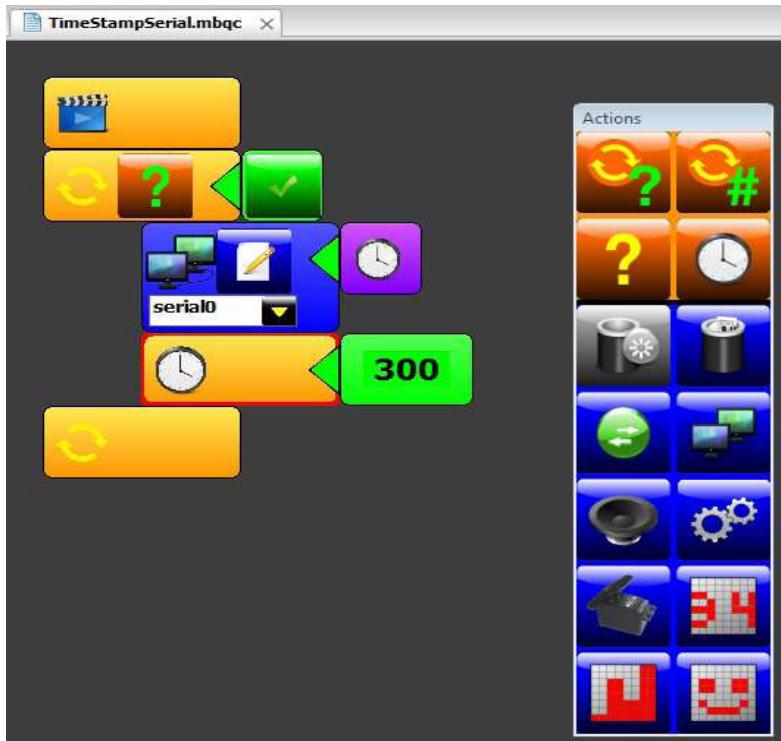
```
#include "stdlib.h"
#include "TRremote.h"
#include "pitches.h"
#include "Minibloq.h"

void setup()
{
    initBoard();

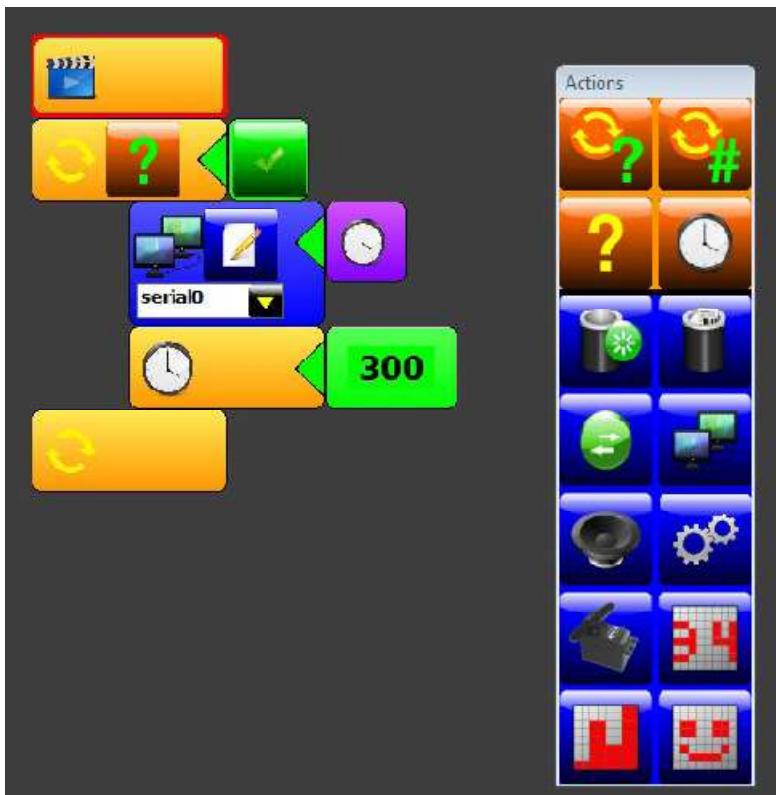
    float count = 0;
    while(true)
    {
        serial0.println(count);
        count = (count+1);
        delay(300);
    }
}

void loop()
{}
```

Las declaraciones e inicializaciones van obligatoriamente en el principio del programa, tras el bloque de inicialización. El entorno habilita o deshabilita los bloques del selector de acciones también en relación al contexto del bloque seleccionado actualmente. Por ejemplo, si el usuario tiene seleccionado un bloque cualquiera del programa, que no sea el primer bloque, no podrá agregar un bloque de declaración/inicialización, puesto que las variables sólo pueden declararse al principio. Nótese en la siguiente imagen que el bloque de declaración/inicialización está deshabilitado (gris) en el Selector de acciones, ya que el bloque seleccionado (bloque con el borde rojo) en el Editor de componentes es un bloque intermedio (tras el cual no está permitido insertar declaraciones de variables):



En cambio, cuando el usuario selecciona el primer bloque, se habilita también el botón del bloque de Declaración/Inicialización de variables:



Ahora bien, si se agrega un bloque de Declaración/Inicialización, entre éste y el bloque de inicio, sólo podría agregarse otro bloque de Declaración/Inicialización. Si fuera de otro modo, el usuario podría declarar una variable y luego colocar bloques antes de la declaración, lo cual como se explicó antes, no se permite en Minibloq por no considerarse una buena práctica de programación. Esta situación también es detectada por el entorno, de modo que cuando hay bloques de declaración al principio, y la posición seleccionada de inserción es **antes** de uno de éstos, el Selector de acciones deshabilita automáticamente todos los bloques, excepto los de declaración:

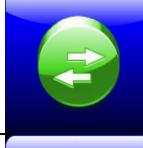
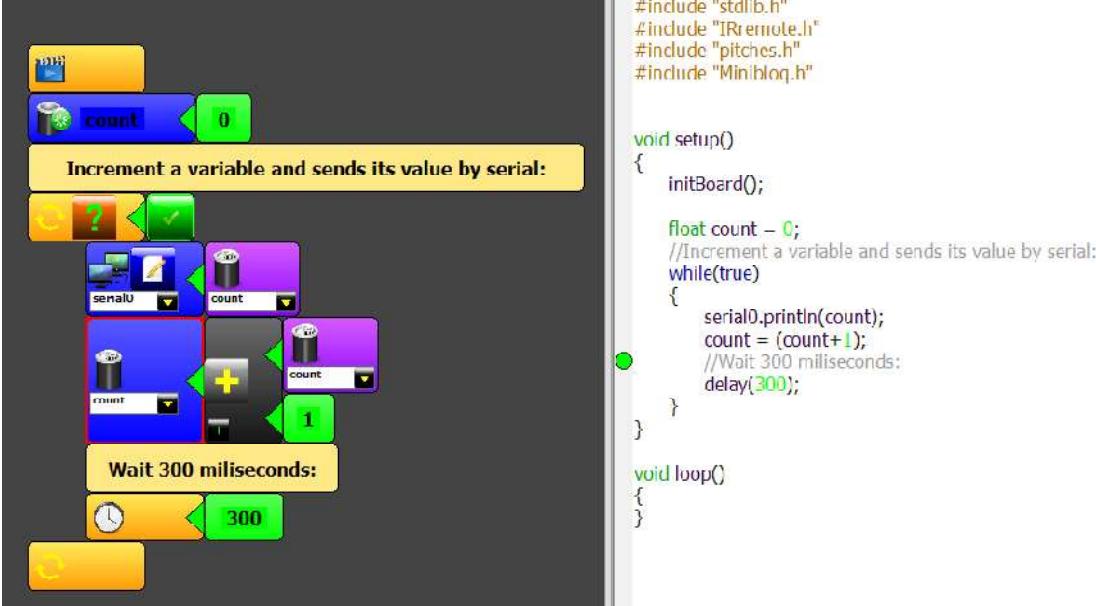
- No hay variables globales. Si en el futuro se incorporan bloques de usuario (a modo de procedimientos y funciones), las variables permanecerán siendo locales. En la versión actual, las variables tienen alcance local dentro de la función setup() del código generado.
- Por último, cabe destacar que Minibloq lista automáticamente sólo los nombres de variables declaradas en los bloques de acceso y asignación, de modo que el usuario no puede ingresar a mano allí nombres inválidos, protegiéndolo de este modo de posibles errores. Nótese el menú contextual en la siguiente imagen, que sólo lista las variables "Code" y "Vel", declaradas antes:

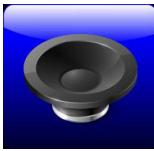
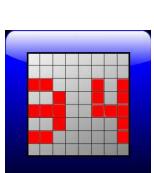
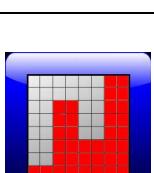
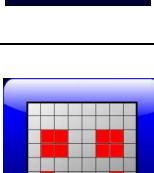
## Bloques

A continuación se describen brevemente los bloques de Minibloq y sus parámetros. En algunos casos se ha optado por no traducir el nombre del bloque, sobre todo en aquellos ligados a hardware (como por ejemplo "IOPin").

### Selector de acciones

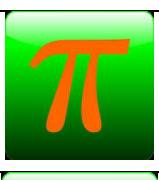
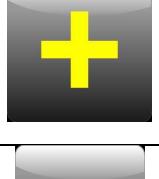
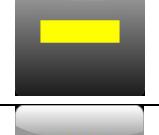
	<b>Esperar:</b> Este bloque hace que el programa espere durante el tiempo especificado (en milisegundos).
--	---

	<b>Variables (crear):</b> Este bloque permite crear e inicializar una variable, para almacenar en ella un número o el resultado de una expresión, y de este modo poder utilizar dicho valor en otras partes del programa.
	<b>Variable (escritura):</b> Este bloque permite asignar un valor a la variable Seleccionada. Numérico.
	<b>IOpin (escritura):</b> Este bloque permite establecer el estado de un pin de salida digital del controlador. Booleano.
	<b>AnalogWrite (escritura analógica):</b> Este bloque permite controlar una salida analógica (PWM). Numero.
	<b>SerialNumber (escritura):</b> Este bloque permite enviar datos (valores numéricos) por el puerto USB (o por un puerto serie, dependiendo del modelo de controlador) del controlador a la computadora.
	<p><b>Comentarios:</b> Además de la posibilidad de comentar el código, Minibloq permite también agregar comentarios arbitrarios, para documentar el programa. Esto se realiza por medio del bloque de comentarios, en el selector de acciones: Dicho bloque permitirá agregar comentarios de una sola línea entre los bloques. Y como se puede ver en la siguiente imagen, Minibloq automáticamente agregará el comentario al código:</p>  <pre>#include "stdlib.h" #include "IRremote.h" #include "pitches.h" #include "Minibloq.h"  void setup() {     initBoard();      float count = 0;     //Increment a variable and sends its value by serial:     while(true)     {         serial0.println(count);         count = (count+1);         //Wait 300 milliseconds:         delay(300);     } }  void loop() {}</pre>
	<b>SerialText (escritura):</b> Este bloque permite enviar datos (texto) por el puerto USB (o por un puerto serie, dependiendo del modelo de controlador) del controlador a la computadora.

	Buzzer (escritura): Este bloque permite emitir sonidos. El primer parámetro determina la frecuencia (nota) y el segundo la duración. <b>nota (Hz)</b> : número. <b>duración(ms)</b> : número
	Motor (escritura): Este bloque permite controlar un motor eléctrico conectado a una de las salidas de motor del controlador. <b>Potencia (-100 a 100)</b> : número
	ServoRC (escritura): Este bloque permite establecer la posición (en grados) de un servo R/C conectado a una de las salidas digitales del controlador.
	ScreenNumber (escritura): Este bloque permite mostrar números de -99 a +99 en la pantalla del controlador.
	ScreenBars (escritura): Este bloque permite mostrar 4 valores numéricos (que pueden ir de 0 a 100) en forma de barras en la pantalla del controlador. <b>barra0 (0 a 100)</b> : número. <b>barra1 (0 a 100)</b> : número. <b>barra2 (0 a 100)</b> : número. <b>barra3 (0 a 100)</b> : número.
	ScreenSprite(escritura): Este bloque permite mostrar en la pantalla del controlador uno de los dibujos pre instalados.

### Selector contextual numéricico

	<b>Variable (lectura):</b> Este bloque devuelve el valor almacenado en la variable seleccionada
	<b>Motor (lectura):</b> Este bloque devuelve el estado de un motor eléctrico conectado a una de las salidas de motor del controlador.
	<b>ServoRC (lectura):</b> Este bloque devuelve la posición (en grados) de un servo R/C conectado a una de las salidas digitales del controlador.

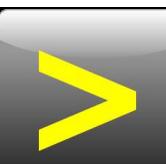
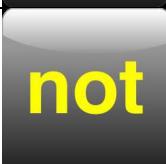
	<b>PulseIn:</b> Este bloque devuelve el valor de la entrada analógica (o "entrada de sensor") seleccionada.
	<b>timeStamp:</b> Este bloque devuelve la cantidad de milisegundos desde que el programa comenzó.
	<b>Número pseudoaleatorio:</b> Este bloque devuelve un número pseudoaleatorio entre 0 y 100.
	<b>buzzerNote (constante):</b> Este bloque devuelve la frecuencia correspondiente a la nota seleccionada
	<b>número (constante):</b> Este bloque devuelve un número constante.
	<b>pi (constante):</b> Este bloque devuelve la constante pi con un número limitado de decimales (3.14159265358979323846).
	<b>e (constante):</b> Este bloque devuelve la constante e (o "número de Euler") con un número ilimitado de decimales (2.7182818284590452354).
	<b>Sumar:</b> Este bloque permite sumar números, variables y otras expresiones numéricas.
	<b>Sustraer:</b> Este bloque permite sustraer números, variables y otras expresiones numéricas.
	<b>Multiplicar:</b> Este bloque permite multiplicar números, variables y otras expresiones numéricas.
	<b>Dividir:</b> Este bloque permite dividir números, variables y otras expresiones numéricas. También se puede utilizar para expresar números como fracciones.

	<b>Negativo</b> (signo negativo unario): Este bloque hace negativo al número o a la expresión numérica a su izquierda.
	<b>Potencia:</b> Este bloque permite elevar el primer parámetro (base) a la potencia dada por el segundo parámetro (exponente).
	<b>Valor absoluto:</b> Este bloque retorna el valor absoluto del número o de la expresión numérica a su izquierda.
	<b>resto (de la división):</b> Este bloque devuelve el resto de la división entre sus 2 parámetros.
	<b>Mínimo:</b> Este bloque retorna el mínimo de dos números, variables u otras expresiones numéricas.
	<b>Máximo:</b> Este bloque retorna el máximo de dos números, variables u otras expresiones numéricas.
	<b>Mapear:</b> Este bloque permite mapear linealmente (realizando una regla de tres) un valor numérico desde un rango de valores hacia otro rango.
	<b>Limitar:</b> Este bloque permite limitar un número, una variable u otra expresión numérica a un valor entre un mínimo y un máximo.
	<b>Seno:</b> Este bloque retorna el valor del seno del ángulo dado por el número o la expresión numérica a su izquierda (en radianes).
	<b>Coseno:</b> Este bloque retorna el valor del coseno del ángulo dado por el número o la expresión numérica a su izquierda (en radianes).
	<b>Tangente:</b> Este bloque retorna el valor de la tangente del ángulo dado por el número o la expresión numérica a su izquierda (en radianes).
	<b>Arcoseno:</b> Este bloque retorna el ángulo (en radianes) cuyo seno se está dado por el número o la expresión numérica a su izquierda.
	<b>Arcocoseno:</b> Este bloque retorna el ángulo (en radianes) cuyo coseno es dado por el número o la expresión numérica a su izquierda.

	<b>Arcotangente:</b> Este bloque retorna el ángulo (en radianes) cuya tangente es está dada por el número o la expresión numérica a su izquierda.
	<b>IRRemote:</b> Este bloque devuelve el número obtenido de un sensor de control remoto infrarrojo.
	<b>Ping:</b> Este bloque devuelve la distancia aproximada (en cm) medida con un sensor ultrasónico Parallax's PING)) [TM], Seeed Studio SEN136B5Bm, Seeed Studio Grove Ultrasonic Ranger, o similar.
	<b>IRRanger (10 a 80 cm):</b> Este bloque devuelve la distancia aproximada (en cm) medida con un sensor infrarrojo de distancia SharpGP2Y0A21YK0F (10 a 80 cm aprox.).
	<b>IRRanger (20 a 150cm):</b> Este bloque devuelve la distancia aproximada (en cm) medida con un sensor infrarrojo de distancia Sharp GP2Y0A02YK0F (20 a 150 cm aprox.).

### Selector Contextual BOOLEANO

	<b>IOPin (lectura):</b> Este bloque devuelve el estado de un pin digital del controlador.
	<b>verdadero (constante):</b> Este bloque devuelve siempre "verdadero" (true).
	<b>falso (constante):</b> Este bloque devuelve siempre "falso" (false).
	<b>Igual:</b> Este bloque permite comparar números, y devuelve "verdadero" (true) si son iguales.
	<b>Distinto:</b> Este bloque permite comparar números, y devuelve "verdadero" (true) si son diferentes.

	<b>menor que:</b> Este bloque permite comparar números, y devuelve "verdadero" (true) si el primer parámetro es menor al segundo parámetro.
	<b>mayor que:</b> Este bloque permite comparar números, y devuelve "verdadero" (true) si el primer parámetro es mayor al segundo parámetro.
	<b>menor o igual que:</b> Este bloque permite comparar números, y devuelve "verdadero" (true) si el primer parámetro es menor o igual al segundo parámetro.
	<b>mayor o igual que:</b> Este bloque permite comparar números, y devuelve "verdadero" (true) si el primer parámetro es mayor o igual al segundo parámetro.
	<b>negación lógica:</b> Este bloque permite negar expresiones lógicas (booleanas).
	<b>y (función lógica):</b> Este bloque realiza un "Y" lógico (booleano) y devuelve el resultado.
	<b>o (función lógica):</b> Este bloque realiza un "O" lógico (booleano) y devuelve el resultado
	<b>o-exclusivo (función lógica):</b> Este bloque realiza un "O-Exclusivo" lógico (booleano) y devuelve el resultado

## TP MINIBLOQ:

- 1) Activar M0 a una velocidad de 100%
- 2) Activar M1 a una velocidad de 40%
- 3) Activar el motor M0 al 100%, que funcione por 4 segundos y luego se detenga.
- 4) Activar el motor M0 al 100%, que funcione por 3 segundos, luego funcione al 60% por 2 segundos, luego al 30% por 5 segundos y luego se detenga.
- 5) Activar M0 al 100% durante 5 segundos, se detenga por 2 segundos y gire en sentido contrario al 100% por 3 segundos y luego se detenga.

- 6) Activar M0 al 100% durante 5 segundos, se detenga por 2 segundos y gire en sentido contrario al 20% por 3 segundos y luego se detenga.
- 7) Al comenzar que se emita un sonido en nota A1 durante 1 segundo, espere 3 segundos y arranque el motor M0 al 100% por 3 segundos y finalmente se detenga.
- 8) Al comenzar que se emita un sonido en nota A2 durante 2 segundos, espere 3 segundos y arranque el motor M0 al 100% y M1 al 30% por 3 segundos y finalmente se detenga.
- 9) Al comenzar que se emita un sonido en nota A2 durante 2 segundos, espere 3 segundos y arranque el motor M0 al 100% por 3 segundos, luego M1 al 30% en sentido contrario por 3 segundos y finalmente se detenga.
- 10) Programar un tono de audio de Nota A1 con una duración de 500 mili segundos.
- 11) Programar que se emitan dos tonos, el primero de Nota A1 con duración de 500 mili segundo y el otro tono de Nota B5 con 500 mili segundos de duración.
- 12) Programar que se emita sonido del ejercicio anterior y al terminar este, comience a girar M0 a 100 % durante 1 segundo y luego se detenga.
- 13) Programar que se emita sonido del ejercicio anterior y al terminar este, comience a girar M0 a 100 % durante 1 segundo y luego invierta su marcha al 100% durante 2 segundos y finalmente se detenga.
- 14) Programar un sonido en donde el tono y la duración están asignadas a variables. Tono 3000 Hertz y duración 1 segundo.
- 15) Activar el motor M0 al 100%, que funcione por 4 segundos y luego se detenga, los valores de velocidad y tiempo de funcionamiento estarán asignados por variables.
- 16) Marcar en el piso 2 puntos (A y B) distanciados aproximadamente 70 cm. Programar que el auto N6 emita un sonido por 1 segundo, espere 2 y luego salga del punto A y llegue al punto B y luego se detenga.
- 17) Con esos 2 puntos que al llegar retorne al punto A, marcha atrás.
- 18) Que salga de A, llegue a B, gire 180 grados y retorne al punto inicial A.
- 19) Se marcaran 3 puntos de manera triangular (A, B, C). El auto debe salir de A, llegar a B y luego llegar a C.
- 20) Cuando llega a C, volver a A marcha atrás.
- 21) Salir de A hasta C, girar 180º y volver al punto A.
- 22) Se marcaran 4 puntos de manera rectangular (A, B, C, D). El auto debe salir de A, llegar a B y luego llegar a C y finalmente a D.
- 23) Cuando llega a D, volver a A marcha atrás.
- 24) Salir de A hasta D, girar 180º y volver al punto A.
- 25) Analizar los valores de manera digital y analógica que devuelve el sensor LDR utilizando el comando serial.
- 26) Analizar los valores de manera digital y analógica que devuelve el sensor infrarrojo al presentarle diferentes colores.
- 27) Analizar los valores que devuelve el sensor de control remoto al presionar diferentes teclas del control remoto.
- 28) Emitir un sonido cada vez que se ilumine un LDR,
- 29) Encender un LED cuando se ilumina un LDR y al volver a iluminarlo que se apague.
- 30) Activar un motor cuando se ilumina un LDR.
- 31) Activar un motor al iluminar un LDR, al volver a iluminarlo que se detenga el motor.

- 32) Activar un motor con velocidad 10 al iluminar un LDR. Cada vez que se ilumine la velocidad crece en 10. Cuando llegue a 100 que se detenga.
- 33) Hacer que un auto gire sobre el eje de una de sus ruedas durante 5 segundos.
- 34) Hacer que un auto gire sobre el eje del auto durante 5 segundos.
- 35) Utilizando el control remoto realizar que un auto se mueva para adelante, atrás, doble a izquierda y derecha, se detenga y toque bocina.
- 36) Conectar la placa de LED y encenderlos secuencialmente de a uno con intervalos de 1 segundos.
- 37) Conectar la placa de LED y encenderlos secuencialmente de a uno con intervalos de 1 segundos, finalmente esperar 3 segundos y apagarlos en sentido inverso.
- 38) Programar una barra de 5 LED que se encienda por 1 segundo, que se apague por  $\frac{1}{2}$  segundo de manera cíclica.
- 39) COCHE FANTASTICO. Encender y apagar 7 led secuencialmente, el tiempo de encendido y apagado de 100 ms, luego encender y apagar. Luego de manera inversa encender y apagar con retardo de 100 ms, así indefinidamente logrando un efecto de ida y vuelta.
- 40) Investigando sobre el código Morse, Codificar el mensaje internacional de ayuda SOS. Para la raya se asignara un tiempo de 1 segundo, para el punto  $\frac{1}{2}$  segundo y  $\frac{1}{2}$  segundo para los tiempos entre letra y letra. Se utilizará el brillo de un LED.
- 41) Idem anterior, pero utilizando sonidos, por ejemplo nota A1.
- 42) Utilizando código Morse. Codificar palabras (utilizar solo letras) por medio de sonido y el resto de los compañeros debe descodificar que ha codificado cada uno. Irán ejecutando de a uno. El que descodifica primero se le asigna 10 puntos que se irá anotando en una tabla de control. Y finalmente se asignará un ganador del concurso.
- 43) Realizar un telégrafo utilizando un pulsador. Mientras se mantiene presionado suena cuando se suelta se apaga el sonido. Una vez logrado eso, cada integrante emitirá palabras para que el resto descodifique.
- 44) Idem anterior, pero que se utilice el control remoto, y por ejemplo con 2 sea punto y con la raya. Utilizar sonido y LED.
- 45) Hacer que un auto arranque hacia adelante sobre un piso oscuro, y cuando llegue a una línea blanca se detenga. (puede ser piso claro y línea oscura).
- 46) Idem anterior, pero que al llegar se detenga, emita un sonido audible, gire 180° sobre el eje del auto y funcione de vuelta durante 4 segundos.
- 47) Realizar que un auto se mueva, cuando esté a 5 cm de un obstáculo se detenga y emita un sonido de 5000 Hz durante 3 segundos.
- 48) Realizar que un auto se mueva, cuando esté a 10 cm de un obstáculo gire 180° y se mueva por 2 segundos.
- 49) Realizar que un auto se mueva, cuando esté a 10 cm de un obstáculo lo esquive y retorne a su trayectoria.
- 50) Realizar que un auto se mueva de manera permanente dentro de una pista cuadrada o circular.
- 51) Realizar un auto que siga una línea blanca en un trayecto cuadrado o circular.
- 52) En la salida hay una raya clara, en la llegada también. Arranca el auto y cuando llega a una línea gira 180°, sigue moviéndose y llega a la otra raya, gira 180° y va hacia la otra, y así sucesivamente.

- 53) Investigar cómo se realiza un informe técnico. Realizar un informe técnico **detallado** de todos los comandos utilizados hasta el momento. Recordar que un informe técnico debe servir como guía de aprendizaje para otras personas. Por eso debe ser muy detallado.

# Sistemas Tecnológicos (Química)

## Introducción:

El mundo que nos rodea contiene todo tipo de objetos (mesas, piedras, mares, etc.). Estos cuerpos sufren cambios, transformaciones que son estudiados por las ciencias naturales, como por ejemplo la química, física y biología, etc. Éstas son ciencias experimentales pues se basan en hechos comprobables. El componente común es la materia. Todo ente material ocupa un lugar en el espacio (tiene volumen) y posee masa.

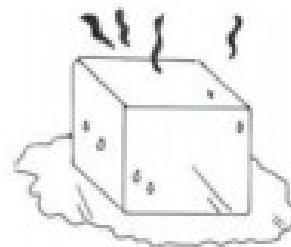
Existen distintos tipos de materiales que forman los cuerpos. La química se ocupa principalmente de la composición, propiedades y transformaciones de los materiales así como también se ocupa de los cambios energéticos

## Sistemas Materiales:

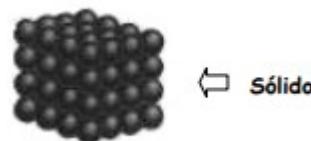
Un sistema material es todo cuerpo o conjunto de cuerpos que nos interesa estudiar. Los cuerpos están formados por materia, y la materia es todo aquello que nos rodea, ocupa un lugar en el espacio y es perceptible por nuestros sentidos. Toda porción limitada de materia recibe el nombre de cuerpo. Por ejemplo, una manzana, una pelota, un vaso. Además, la materia está compuesta por partículas como átomos, moléculas, iones. Estas partículas que componen la materia son muy pero muy chiquitas. Para que te des una idea, el radio del átomo más chiquito, el de hidrógeno mide  $5,29 \cdot 10^{-11} m$ , o sea, ¡0,0000000529 cm! Son tan pero tan chiquitos que son imposibles de ver ni siquiera con el microscopio más potente.

## Estados de agregación o estados físicos de la materia:

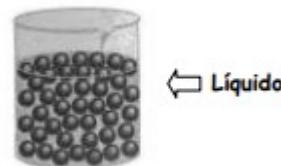
Una de las cosas más importantes cuando hablamos de la materia es su estado físico. En general se la puede encontrar en 3 estados diferentes: sólido, líquido y gaseoso. Por ejemplo, un cacho de carbón en la parrilla está en estado sólido. El alcohol en el baño está en estado líquido. El gas de la cocina está en estado gaseoso. Ahora: ¿Cuáles son las características de cada estado?

**Sólidos:**

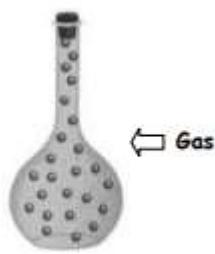
Los sólidos tienen forma y volumen propios y no se los puede comprimir (son incompresibles). A nivel microscópico es el estado más ordenado. Las partículas que lo forman se encuentran de manera muy compacta, por lo que no pueden moverse. Todas estas propiedades hacen que los sólidos sean capaces de mantener su estructura. Así como los dejamos, así se quedan. Por ejemplo, si ejercemos una fuerza sobre un bloque de hierro no se va a comprimir.

**Líquidos:**

Si bien tienen volumen propio, toman la forma del recipiente que los contiene: si pasamos un líquido de un vaso a una taza, el líquido va a tomar la forma de la taza. Este estado es un poco más desordenado que el estado sólido. Quiere decir que, si bien Sólido sus partículas siguen muy juntitas entre sí, pueden deslizarse unas sobre otras permitiéndoles fluir. Pero ojo: como sus partículas siguen muy cerca, se los puede comprimir muy poco.

**Gases:**

Los gases no tienen forma ni volumen propios. Ocupan todo el espacio del recipiente que los contiene y se los puede comprimir fácilmente. Esto es porque sus partículas no están juntas, si no que se encuentran muy dispersas entre sí, moviéndose al azar. Es el estado de máximo desorden.



Como los **gases** y los **líquidos** pueden **fluir** se los suele llamar fluidos.

### Cambios de Estado:

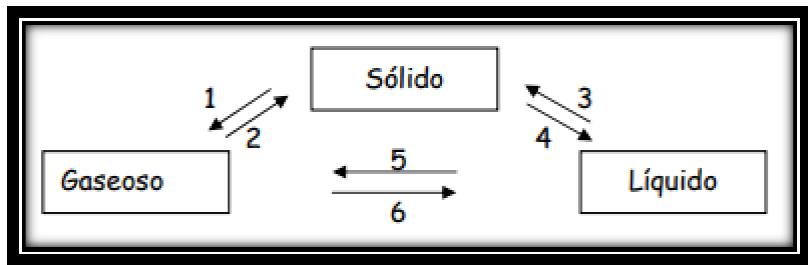
Si tomamos una olla con agua y la calentamos, en cierto momento empiezan a salir burbujas. ¿Qué está pasando? Lo que pasa es que el agua en estado líquido pasa a ser vapor de agua (estado gaseoso). Esto, que lo ves todos los días es un cambio de estado de la materia. Estos cambios son el resultado de la “absorción” o la “liberación” de calor por parte del material. Otro ejemplo: si ponemos un vaso con agua caliente en el freezer lo que pasa es que el agua caliente libera el calor que tenía hacia el resto del freezer y, como su temperatura disminuye hasta menos de 0 °C, se congela.

Pero hay que tener en cuenta que los cambios de estado son cambios físicos, es decir son cambios donde no se ve afectada la estructura química del material (esto significa que no existe reacción química).

En estado líquido como en estado gaseoso, el agua tiene la misma composición química ( $H_2O$ ), o sea sigue siendo ¡agua!



Los cambios de estado tienen varios nombres: sublimación, fusión, condensación.  
Veamos un dibujito para recordarlos:



Con números están marcados los distintos cambios físicos.

- 1- **Volatilización:** Es el cambio de estado que se da cuando una sustancia pasa del estado sólido al estado gaseoso (Esto pasa con la naftalina y con el hielo seco).
- 2- **Sublimación:** Es el cambio de gas a sólido sin pasar por líquido.
- 3- **Solidificación:** Es el cambio de estado que se da cuando una sustancia pasa de estado líquido a sólido. Por ejemplo, el congelamiento de los lagos en invierno.
- 4- **Fusión:** Es el pasaje del estado sólido al estado líquido. A la temperatura de fusión, el sólido comienza a cambiar su estado de sólido a líquido. Lo mismo va para los otros cambios de estado. Veamos en un ejemplo estos últimos cambios de estado que te conté, la fusión y la solidificación. Al prender una vela vemos que la cera que está cerca de la llama empieza a derretirse. (es decir se vuelve líquida, se funde. O sea, ha alcanzado el punto de fusión). Cuando la cera líquida se aleja de la llama, disminuye su temperatura y vuelve al estado sólido. (es decir se solidifica, o sea, alcanza el punto de solidificación). También podemos ver el fenómeno de fusión cuando se derrite el hielo o un helado.
- 5- **El cambio de líquido a gas puede ocurrir de dos maneras distintas:**
  - a. **Ebullición:** Es el pasaje del estado líquido a vapor que se hace en la superficie y en toda la masa líquida. Es lo que pasa al hervir agua. La temperatura a la que hierve se llama temperatura de ebullición.
  - b. **Vaporización:** Es el pasaje del estado líquido a vapor, pero que se da solamente en la superficie del líquido. Si dejamos un vaso con agua a temperatura ambiente sin tapar, vas a ver que después de unos días hay menos agua. Lo que falta es lo que se evaporó.
- 6- **Condensación:** Es el cambio de estado que ocurre cuando una sustancia pasa del estado de vapor al líquido, por disminución de la temperatura. Esto pasa cuando el vapor, al chocar con una superficie más fría, disminuye su temperatura y vuelve al estado líquido. Por ejemplo si ponemos a calentar agua en una olla tapada, después de un rato en la tapa aparecen gotitas de agua. Otro ejemplo es cuando se empañá el vidrio del auto en un día de lluvia. Esto pasa porque el agua fría de la lluvia enfria al vidrio, entonces el vapor que está dentro del auto se condensa.

### Propiedades Intensivas y Extensivas de la Materia:

Los cuerpos tienen propiedades que pueden ser apreciadas por los sentidos, o bien medidas con instrumentos en el laboratorio, que nos permiten distinguirlos entre sí. Podemos clasificar a estas propiedades en dos grandes grupos:

- a) **Propiedades intensivas:** Son las propiedades que **NO** varían con la cantidad de materia considerada. Caracterizan a los materiales desde un punto de vista macroscópico (o sea, a lo grande); pero están relacionadas con la naturaleza de la materia (es decir, con su estructura química). Por esto, se las puede medir en cualquier porción del sistema. Son, por ejemplo, el punto de ebullición, el calor específico y la densidad. Pensemos en la fusión del agua (pasaje de sólido a líquido). A presión normal, la temperatura a la que esto pasa (el tan famoso punto de fusión) va a ser 0 °C sin importar si tenemos a un cubito de hielo o al iceberg que hundió al Titanic. Esto es porque el punto de fusión también es una propiedad intensiva. Las propiedades intensivas te pueden servir para identificar y caracterizar una sustancia pura, es decir, aquella que está compuesta por un solo tipo de molécula. Por ejemplo, si tenemos un líquido transparente, incoloro, insípido, al cual le calculamos la densidad y nos da igual a 1 g/cm<sup>3</sup>, tenemos buenas razones para creer que ese líquido es, ni más ni menos que agua.
- b) **Propiedades extensivas:** Son las que **SÍ** varían con la cantidad de materia. Por ejemplo: la masa, el volumen, el peso, la longitud. A diferencia de las propiedades intensivas, las extensivas no permiten identificar un material: dos cuerpos pueden pesar lo mismo o tener el mismo volumen y ser, sin embargo, totalmente diferentes.

## Propiedades Extensivas importantes:

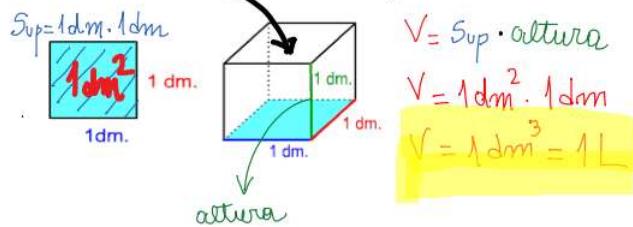
**Masa:** Podríamos decir que la masa es la reina de las propiedades extensivas, ya que se define como la cantidad de materia contenida en un cuerpo. En general en química la expresamos en gramos (g).

La masa de un cuerpo es la cantidad de materia que ese cuerpo tiene.

**Volumen:** Es la cantidad de espacio que ocupa un cuerpo. Generalmente se pone en centímetros cúbicos ( $cm^3$ ), aunque puede estar en  $dm^3$  o litros.

Nota aclaratoria: 1 cubo de 10 cm = 1 dm de arista tiene una capacidad de 1

*¡aca cabe un litro!*



**Curiosidad:** Observemos esto. Existe otra propiedad, la densidad delta ( $\delta$ ), que se define así:

$$\text{densidad} = \frac{\text{masa}}{\text{volumen}}$$

Dijimos que masa y volumen eran propiedades extensivas, o sea, que dependen de la cantidad de materia. Pero con la densidad que tiene el agua en un vasito y la que tiene en la pileta del club. En el vasito hay menos masa y menos volumen que en la pileta. Pero la relación entre ambas va a ser igual en los dos casos: la densidad del agua del vasito y de la pileta va a ser la misma. Esto quiere decir que la densidad es una propiedad intensiva, porque no depende de la cantidad de materia. Esto es un hecho curioso porque es el cociente entre dos propiedades extensivas.

La unidad que se usa comúnmente para expresar la densidad es gramos por o sobre centímetros cúbicos  $\frac{g}{cm^3}$

## Sistemas Homogéneos y Heterogéneos

Para llevar adelante los estudios químicos debemos definir perfectamente la parte del universo de cosas que queremos estudiar. Retomemos y ampliemos el concepto de sistema material: la porción o parte del universo que constituye nuestro objeto de estudio lo llamamos nuestro **sistema material**.

Un sistema material puede interactuar con el medio o el entorno, existiendo la posibilidad que intercambie con éste materia y/o energía.

- Si intercambia materia y energía lo denominamos **sistema abierto**.
- Cuando no pueda intercambiar materia pero sí energía lo llamamos **sistema cerrado**.
- Cuando no intercambia ni materia ni energía lo denominamos **sistema aislado**.



Un sistema material puede contener uno o más cuerpos o partes de cuerpos. Además un sistema material puede estar formado por uno o varios componentes (o sustancias). Si en un sistema existe más de un componente decimos que es una mezcla. Por ejemplo:

- Agua (un componente)
- Trozos de cobre (un componente)
- Agua salada (mezcla)
- Suspensión de talco en agua (mezcla)

**Sistemas homogéneos:** Podemos decir intuitivamente que los sistemas homogéneos son aquellos que son uniformes y no presentan ningún límite de separación. Pero hay algo todavía más importante: **las propiedades intensivas en los sistemas homogéneos se mantienen constantes a lo largo de todo el sistema**. Imaginemos un vaso de agua con azúcar disuelta. Si medimos la densidad en la superficie del vaso (parte de arriba) va a ser la misma que en la parte de abajo. Otros sistemas homogéneos son, por ejemplo, una barra de oro puro, una muestra de agua salada o el gas de una garrafa.

**Sistemas heterogéneos:** Ahora imaginemos si al vaso con agua azucarada del ejemplo anterior le tiramos una bolita de vidrio. ¿Las propiedades intensivas se van a mantener constantes a lo largo de todo el sistema?, evidentemente no. La densidad del agua dulce va a ser igual que antes, pero la canica va a tener su propia densidad, diferente a la del agua. Entonces, podemos decir que los sistemas heterogéneos son aquellos en los cuales las propiedades intensivas cambian según la posición del sistema que consideramos. Algo curioso es que también podemos tener un sistema heterogéneo formado por la misma sustancia. Por ejemplo imaginemos un vaso de agua con un hielo flotando. El hielo flota porque el agua en estado líquido es más densa que el agua en estado sólido (en esto el agua es un caso particular, porque en general es al revés). Como hay dos densidades diferentes, las propiedades intensivas no son las mismas. Entonces decimos que se trata de un sistema heterogéneo.

Analizando este tipo de sistemas vemos que no son uniformes, si no que presentan "partes" separadas (el hielo y el agua en el ejemplo anterior). Cada una de estas partes tiene las mismas propiedades intensivas y se las llama, comúnmente, fases.

**Fase:** es cada porción de un sistema material con los mismos valores de propiedades intensivas.

Entonces un sistema va a ser heterogéneo cuando tenga dos o más fases (se los llama "polifásicos").

Como en los sistemas homogéneos las propiedades intensivas se mantienen constantes durante todo el sistema podemos decir que están formados por una sola fase (son monofásicos). Observemos este ejemplo:



Acá tenemos un claro ejemplo de sistema heterogéneo. ¿Cuántas fases tiene? Y... dos seguro. El agua y el barquito. Pero fíjemonos que se pueden ver más fases. El hierro que forma el ancla va a tener distintas propiedades intensivas que la madera que forma la base del barco, por lo que van a ser fases diferentes, etcétera. Así que hay que tener cuidado con esto.

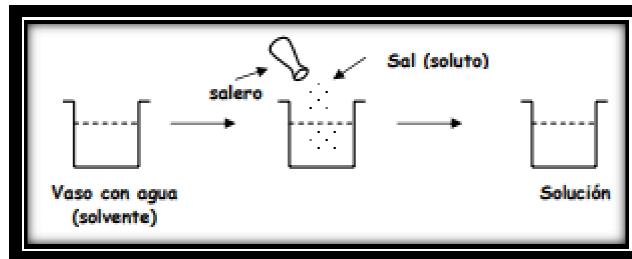
Pero debemos mencionar que el criterio con el que uno dice si un sistema es homogéneo o heterogéneo, depende del poder de resolución del instrumento que se usa para estudiar el sistema. El sistema va a ser heterogéneo si la observación a simple vista, con lupa, microscopio o lo que sea que se use muestra diferentes fases. Aunque a simple vista un sistema te parezca homogéneo, éste puede ser heterogéneo. Un ejemplo de esto es la leche. Parece un sistema homogéneo, pero cuando se la analiza bien, se ve que tiene un montón de cosas flotando por ahí,

como proteínas y polisacáridos bastante grandes. Pero, el agua azucarada también tiene un montón de cosas flotando, el azúcar y es un sistema homogéneo. Dejamos para más adelante ampliar estas cuestiones.

Para finalizar es importante recalcar que los sistemas homogéneos de más de un componente (como el agua azucarada) y los heterogéneos forman mezclas, y que existen diferentes métodos físicos para fraccionar o separar a los distintos componentes, sin que cambie en nada la química de cada uno.

### Solución y Sustancia

¿Qué es una sustancia? Es un sistema homogéneo que posee un sólo componente con propiedades intensivas constantes. Por ejemplo: agua líquida, el hielo y el vapor de agua tienen en común la sustancia agua. ¿Y una solución? Una solución es un sistema homogéneo constituido por dos o más sustancias puras o especies químicas. Cuando una sustancia sólida se mezcla con un líquido de tal forma que no puede distinguirse de él, se dice que la sustancia ha sido disuelta por el líquido. A la mezcla homogénea así formada se la llama solución. Usemos un ejemplillo para describir cómo está compuesta una solución. Supongamos que tenemos un vaso con agua pura a una cierta temperatura, y le agregamos sal. Si revolvemos un poco vemos que la sal “desaparece”. Lo que pasó es que la sal se disolvió. Lo que antes era agua pura, se “convirtió” en agua salada. Al componente mayoritario (en el caso anterior, el agua) la vamos a llamar solvente o disolvente (sv). Al componente minoritario (la sal), soluto (st). Y, a la suma de solvente más soluto (agua salada), la llamamos solución (sc).



Otro ejemplo, el perfume es una disolución en agua de alcohol y de ciertas esencias. Como no es posible determinar dónde está la parte de alcohol, dónde la de agua y dónde la de esencia. Entonces, las soluciones al igual que las sustancias puras en un estado de agregación determinado, están formadas por UNA ÚNICA FASE.