2017

Reporte Robótica Educativa

Materia: Informática Educativa

Alumno: Alejandro Rojas

Profesor: Pedro Willging

5-5-2017

Índice

.....0
Introducción.....2
¿Qué es ARDUINO?.....3
¿Qué quiere decir que sea "*Hardware Libre*"?.....3
Tipos de **ARDUINO y variantes**.....4
Placas.....5
ARDUINO Uno R3:.....5
ARDUINO MEGA 2560 Rev3:.....6
ARDUINO Leonardo:.....8
ARDUINO Uno R3 CH340 (versión china):.....9
DuinoBot v2.3 HID:.....10
Sensores:.....11
Sensor de llama KY-026:.....11
Sensor Ultrasónico Hc-sr04:.....11
Sensor de Humo y Gases MQ2:.....12
Software.....13
Visualino:.....13
mBlock **:**.....14
miniBloq:.....14
Arduino IDE:.....15
Incorporación en la Curricula:.....16
Espacio curricular:.....16
Fundamentación:.....16
Actividad:.....17
Experiencia Kit de Robótica **RobotGroup**.....22
Bibliografia:.....32

Introducción

A lo largo del primer cuatrimestre del 2017, tanto en la cursada de Informática Educativa II, y por cuenta propia, hemos experimentado el mundo de la robótica a través del uso de las placas **ARDUINO**. En el siguiente informe se pretende mostrar, nuestra experiencia utilizando **ARDUINO** en festivales de ciencia de la UNLPam. También se realizara una comparativa de las distintas placas que existen en el mercado, tanto originales, como variantes de las mismas.

¿Qué es ARDUINO?

Antes que nada para empezar a hablar de ARDUINO tenemos que hacernos la siguiente pregunta, ¿Qué es **ARDUINO**?

ARDUINO, es una plataforma electrónica de código abierto basada en *hardware* y *software libre* fáciles de usar. Las placas **ARDUINO** son capaces de leer las entradas (luz es un sensor o humo y gases en el ambiente) y convertirlos en una salida (encender un LED, prender un motor, etc.). Compuesta por circuitos impresos que integran un *microcontrolador* y *un entorno de desarrollo (IDE) propio*, en el cual podemos darles ciertas instrucciones a **ARDUINO**.

¿Qué quiere decir que sea "*Hardware Libre*"?

Anteriormente hemos mencionado que **ARDUINO** es una plataforma electrónica basado en hardware libre. Pero, ¿Qué es el Hardware Libre? Podemos llamar a *hardware libre*, *hardware de código abierto* o *electrónica libre*, a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de *acceso público*, ya sea bajo algún tipo de pago o forma gratuita.

De esta forma, al ser hardware libre, es común que en el mercado, podemos encontrarnos **ARDUINO** de todos los tamaños y formas, ya que cualquiera (con los conocimientos adecuados) bajando el diagrama esquemático puede construir su propia placa. Acá tenemos un ejemplo, de un tutorial subido a YouTube, de cómo armar construir una placa **ARDUINO:**

https://www.youtube.com/watch?v=XmNoo0JlYWs&t=570s

Tipos de ARDUINO y variantes.

Como ya hemos mencionado, como ARDUINO es una plataforma tanto de hardware como software libre, existen distintos tipos de placas ARDUINO en el mercado. Algunas son exactamente iguales, otras tienen ligeras modificaciones y/o mejoras.

Podemos encontrar distintas marcas como:

- **Maker Parts**
- **RobotGroup**
- **A-Star (Marca de la Empresa Pololu)**
- **Adafruit**

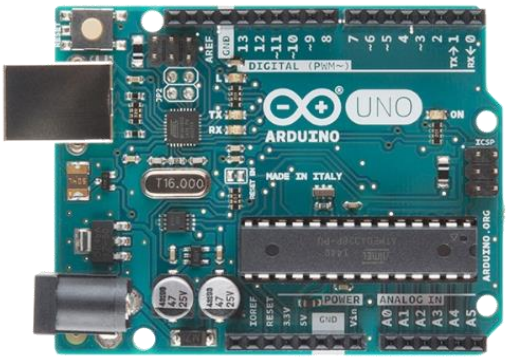
En nuestra experiencia, pudimos contar con los kit de RobotGroup, y placas Arduino Compatibles (Clon de Arduino original, sin marca alguna).

Existe una gran variedad de modelos de placas arduino, en la actualidad. Desde placas para iniciar en la electrónica, como placas especiales para la indumentaria. A continuación, nombraremos los distintos modelos de Arduino que se encuentran disponibles. El siguiente esquema, fue extraído de la página oficial de arduino.

Entre las placas más usadas, para iniciarse en el mundo de la electrónica, tenemos a UNO, MEGA y LEONARDO. Pasaremos a describir las placas que usaremos (y algunas placas similares, con las que se pueden hacer los mismos proyectos):

Placas

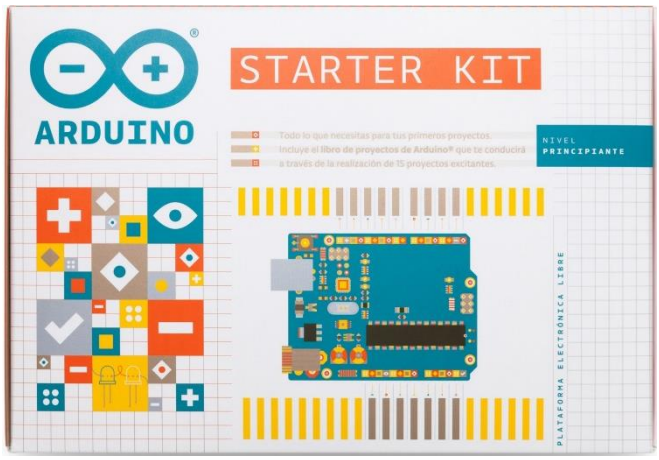
ARDUINO Uno R3:



La placa Arduino Uno, es una de las placas más conocidas, debido a que ha sido la placa, por la cual muchos han descubierto el mundo de la electrónica y la programación. Sirve para realizar proyectos sencillos.

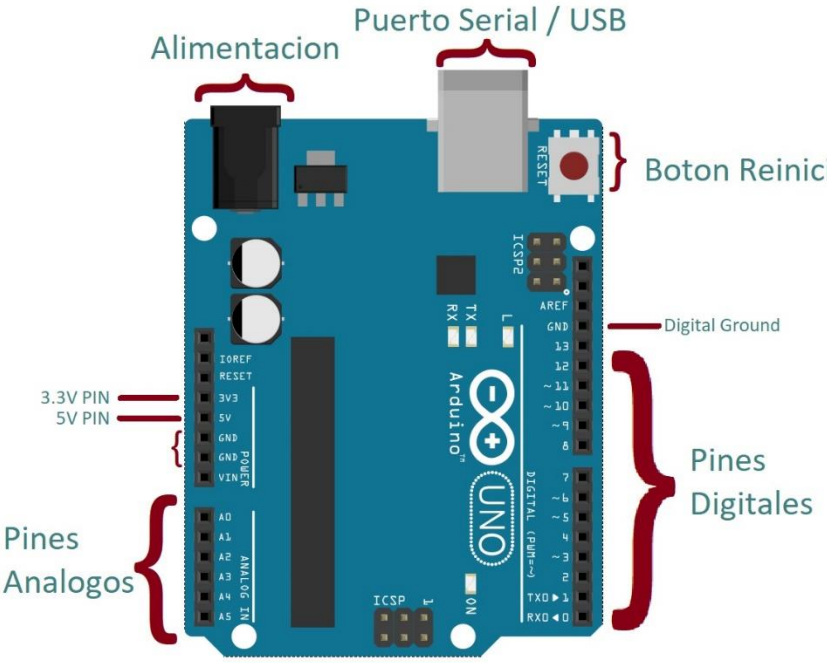
La conexión entre la placa y la computadora, se realiza a través de un puerto serie.

Y como bien aparece en la página oficial de arduino, es una placa para principiantes.



En la página, tenemos la opción de adquirir el starter pack, el cual incluye una Arduino Uno R3.

Esquema de Conexiones:



Características UNO R3:

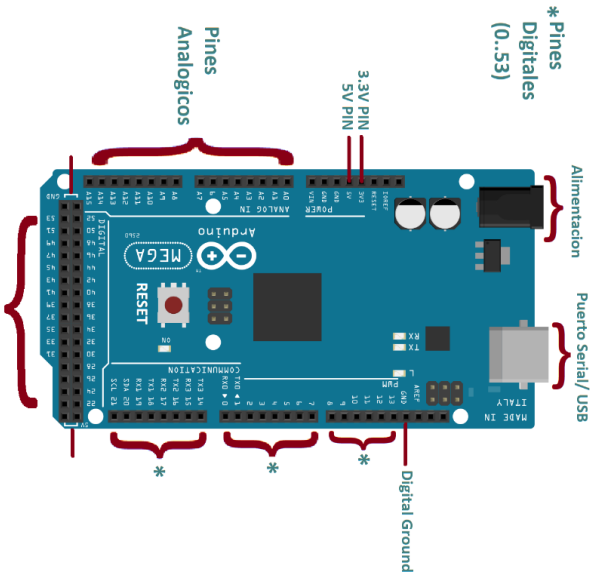
ARDUINO UNO R3	
Microcontrolador	ATmega328
Voltaje de funcionamiento	5v
Pines I/O digitales	14(de los cuales 6 proveen salida PWM)
Pines de entradas análogas	6
Corriente DC por cada pin I/O	40 mA
Corriente DC en el pin de 3.3 V	50 mA
Memoria Flash	32 KH (ATmega328) de los cuales 0.5KB son utilizados por el bootloader
SRAM	2KB
EEPROM	1KB
Velocidad de reloj	16 MHz

ARDUINO MEGA 2560 Rev3:

Podemos decir que el Arduino Mega, es el hermano mayor de Uno.

La UNO, es una placa muy confiable, pero llegara el momento en que tendremos que realizar un proyecto más grande, y vamos a necesitar una cantidad mayor de pines, que las que cuenta la placa UNO, y más memoria para poder cargar grandes programas. Por ello, si quieres una placa con más capacidad, podemos elegir sin lugar a dudas la ¡ARDUINO MEGA! Es una gran opción, a la hora de querer subir un escalón más. De igual forma que su hermana menor, la conexión a la computadora se realizara a través de un puerto serie.

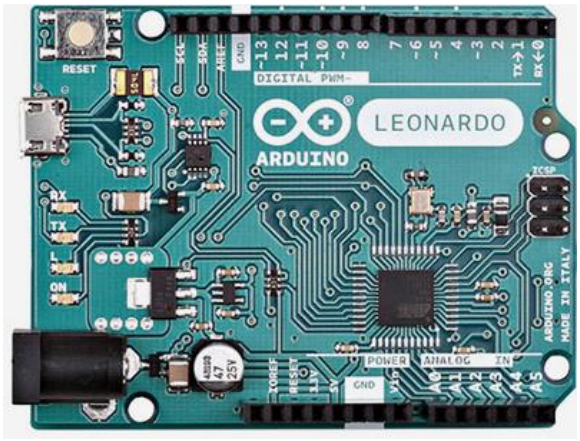
Esquema de Conexiones:



Pasemos a enlistar las características principales de esta placa:

ARDUINO MEGA	
Microcontrolador	ATmega2560
Voltaje de funcionamiento	5v
Pines I/O digitales	54
Pines de entradas análogas	16
Corriente DC por cada pin I/O	40 mA
Corriente DC en el pin de 3.3 V	50 mA
Memoria Flash	256KB de los cuales 8KB son utilizados por el bootloader.
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

ARDUINO Leonardo:



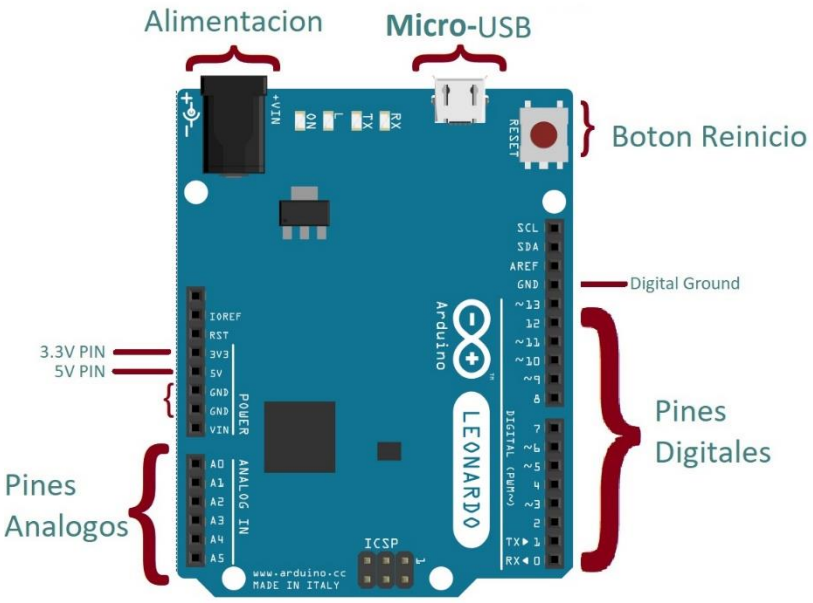
Arduino Leonardo, es como un UNO y con las mismas prestaciones, pero es el primer Arduino que incorpora gestión interna del USB (Y no gestionado por un segundo procesador como en el UNO), lo que le permite presentarse al PC host como si fuera un ratón o un teclado.

A diferencia del puerto serial de la arduino UNO, podemos ver que Leonardo posee una entrada micro-usb.

Características:

ARDUINO LEONARDO	
Microcontrolador	ATmega32u4
Voltaje de funcionamiento	5v
Pines I/O digitales	20
Canales PWM	7
Pines de entradas análogas	12
Corriente DC por cada pin I/O	40 mA
Corriente DC en el pin de 3.3 V	50 mA
Memoria Flash	32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader
SRAM	2 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Velocidad de reloj	16 MHz

Esquema de Conexiones:



ARDUINO Uno R3 CH340 (versión china):

Esta placa, es una copia, versión china, de la arduino uno original, con algunas modificaciones. A simple vista podemos ver que esta ya no tiene el puerto serial, si no, un puerto micro-usb, como vemos que pasa en la arduino Leonardo. Más allá de eso tiene las mismas especificaciones técnicas la arduino uno común.

Esta misma, al tener un puerto micro-usb, a la hora de querer programar en ella, tendremos que instalar los drivers de este mismo puerto. Pero de estas cuestiones hablaremos más adelante en la parte de software.

Nota:

Luego de mostrar las diferencias y similitudes de las placas, UNO LEONARDO Y MEGA, cabe destacar, que las actividades que se mostraran más adelante son realizadas con la placa Arduino UNO R3, sin embargo las mismos, se pueden realizar en cualquier de estas

tres placas. Fueron elegidas, debido a su bajo costo y la gran cantidad de información que existe de ellas, en internet.

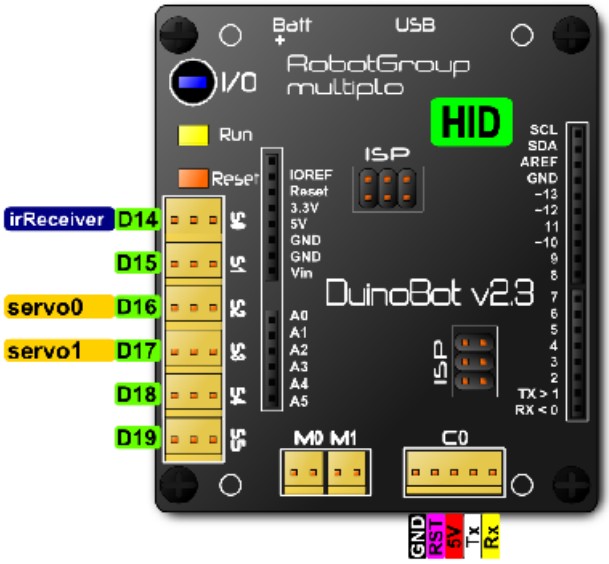
Terminada la parte de Arduino, pasaremos a mostrar, las placas y kits de robotica de la empresa **RobotGroup**.

En una empresa integral de robótica educativa dedicada al diseño, fabricación, investigación y capacitación en robótica cuya misión es la de insertar la robótica como sistema interdisciplinario de aprendizaje en todas las escuelas, colegios y universidades del país.

Aunque **RobotGroup**, no empezó fabricando robots, en el año 2000 se encargaban de dar cursos de Robótica y en el año 2008 emprendieron el reto ser la única empresa en Argentina que produciría íntegramente sus robots a nivel local.

En nuestra experiencia con la robótica, contamos con el kit del robot N8 de RobotGroup, con su respectiva placa DuinoBot v2.3 HID.

DuinoBot v2.3 HID:



La DuinoBot v2.3 HID, es una variante de la Arduino UNO, con algunos agregados tales como :

- Botón de encendido/apagado
- Botón RUN (para iniciar la aplicación cargada)
- Botón Reset ubicado debajo de los anteriores.
- Conectores especiales, sensores (D14...D15), motores (M0, M1), y C0.

Nota: Los programas cargados a la DuinoBot v2.3 HID, son compatibles con la versión v2.3 (no HID).

Sensores:

Sensor de llama KY-026:

El Sensor KY 026 es capaz de detectar llamas y posee una estructura LED detectora de fuego, cuando el circuito capta las ondas emitidas por la llama este enciende una advertencia de tal modo que enciende un red. Este singular sensor está conectado a 3 interfaces digitales del LED.

La longitudes de ondas de llama pueden ser detectadas entre 760nm y los 1100nm, cuando utiliza métodos infrarrojos es más sensible 60 grados.

Este sensor posee salida AO: salida en tiempo real, la cual está estructurada para enviar señales de voltaje análogas.

Generalmente es utilizado en circuitos y su función consiste en enviar señales de advertencia mediante un LED indicador, si el sensor no detecta ningún tipo de señal sencillamente el LED no encenderá.

CARACTERISTICAS GENERALES

- Voltaje funcional: 0 – 15V DC
- Tamaño: 36mm x 16mm
- Angulo de detección: 60°C
- Ajustes por potenciómetro
- Salida analogíca de cantidad
- LED indicador de alimentación.

Sensor Ultrasónico Hc-sr04:

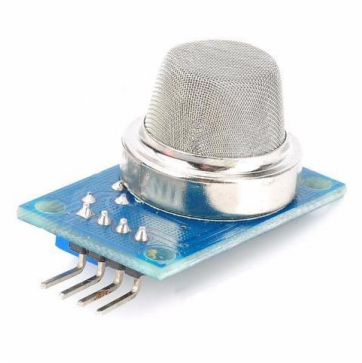
El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo

consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes.

Características:

- Voltaje de alimentación: 5V DC.
- Corriente en reposo: <2mA.
- Angulo de cobertura: <15°.
- Rango de distancia: 2cm – 500 cm.
- Resolución: 0.3 cm.
- Frecuencia ultrasónica: 40k Hz.

Sensor de Humo y Gases MQ2:



La serie MQ de sensores usan un pre-calentador en su interior y un sensor electroquímico. Son sensible a una gran cantidad de gases. El MQ2 es particularmente sensible al gas natural de hornallas lo que o hace ideal para domótica. También detecta LPG butano, propano, metano, alcohol, hidrógeno y "humo". (GNC metano hidrogeno LPG monóxido CO propano).

Características:

- Alimentación: 5V.
- Consumo: 122mA.
- Salida Digital y Analógica.
- Led de encendido.
- Led de accionamiento salida digital.

Software

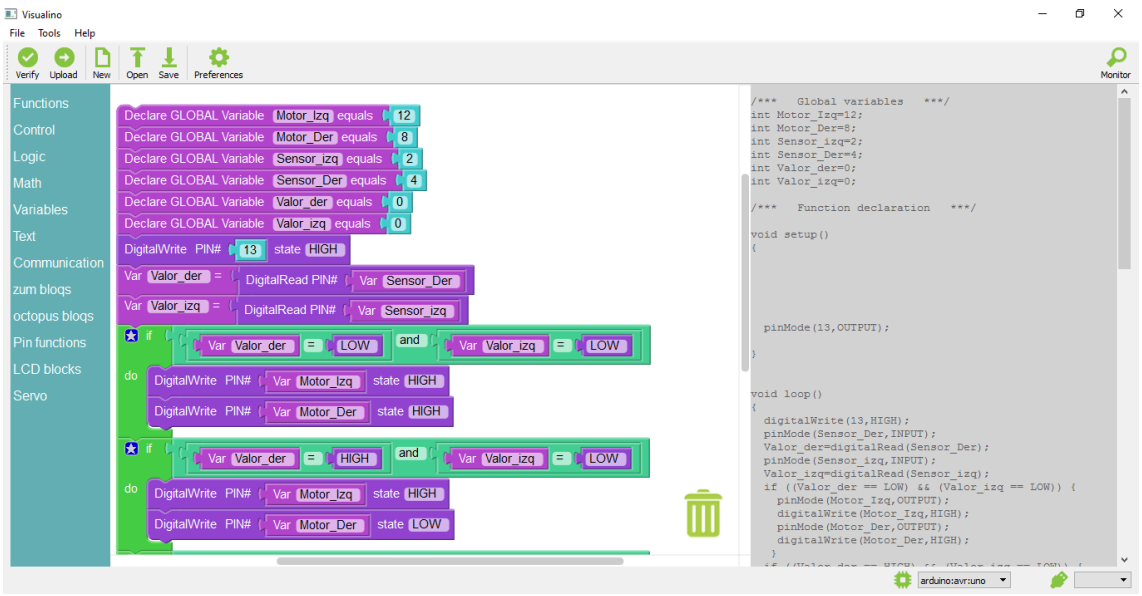
En este apartado vamos a mencionar características e impresiones, sobre los distintos entornos de desarrollo (IDE), en los cuales programar las placas nombradas anteriormente.

IDE Testeados:

- Visualino
- Arduino IDE
- Minibloq (para Arduino y DuinoBot)
- Mblock

Visualino:

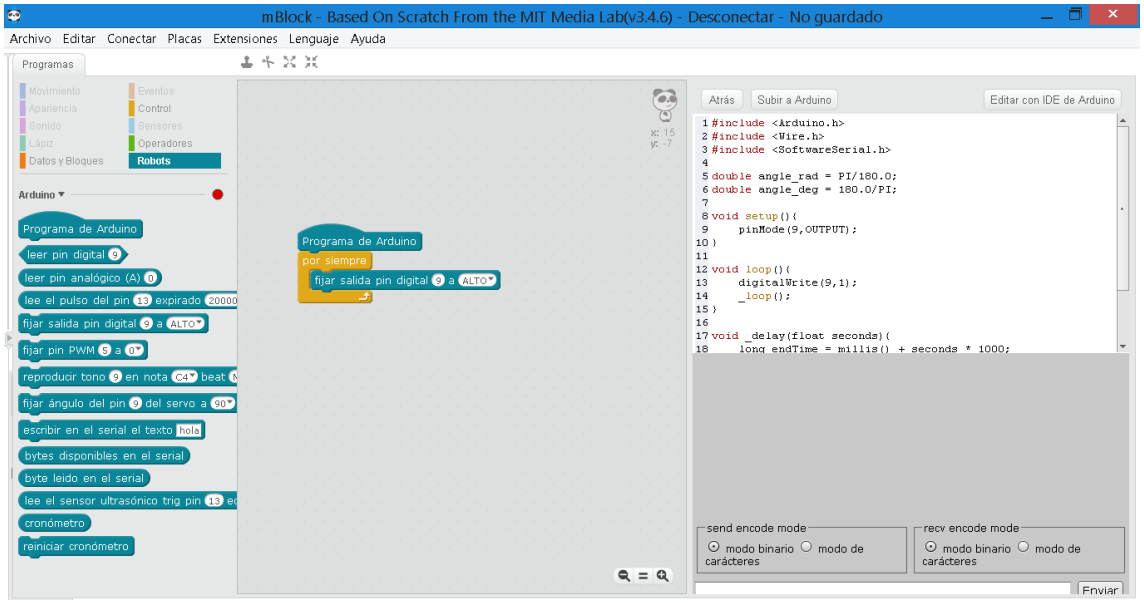
Visualino, es un entorno de desarrollo, ideal para la programación de placas Arduino (no compatible con DuinoBot). Está inspirado en scratch, ya que se enfoca en la programación visual, a través de unos bloques que tienen que ir encajando los unos con los otros. Está orientado a aquellas personas que quieren aprender a usar arduino y a su vez se inician en el mundo de la programación.



mBlock :

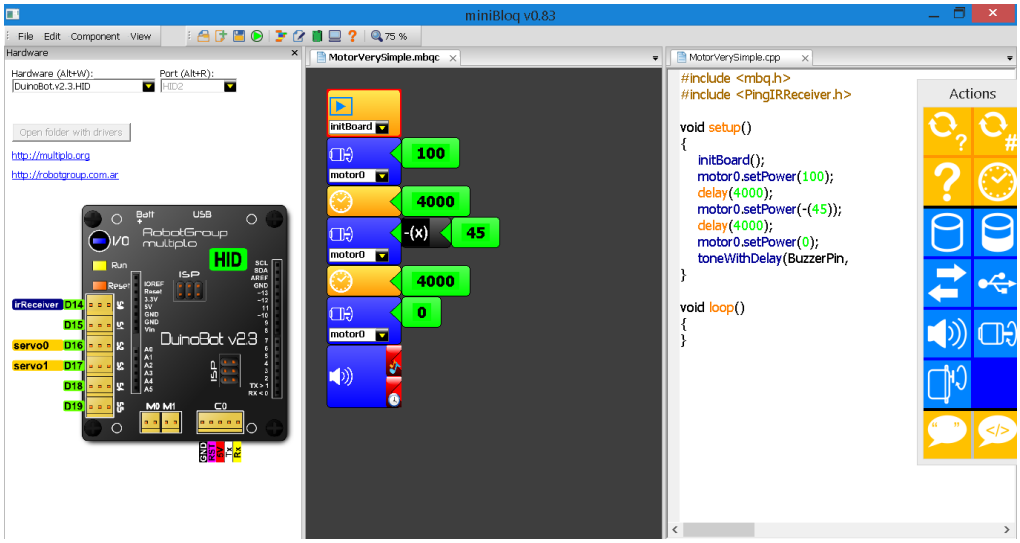
mBlock, es una versión modificada de scratch, añadiendo la opción de programar robots, utilizando placas arduino o mBot. Es un entorno de desarrollo amigable y de fácil comprensión. Una gran ventaja de este IDE, es la de además de ver el código en bloques, podemos ver cómo quedaría el código en c++.

Nota: mBlock suele fallar a veces a la hora de cargar un programa en la placa, se recomienda reiniciar tanto la placa como el IDE, si se presentan problemas.



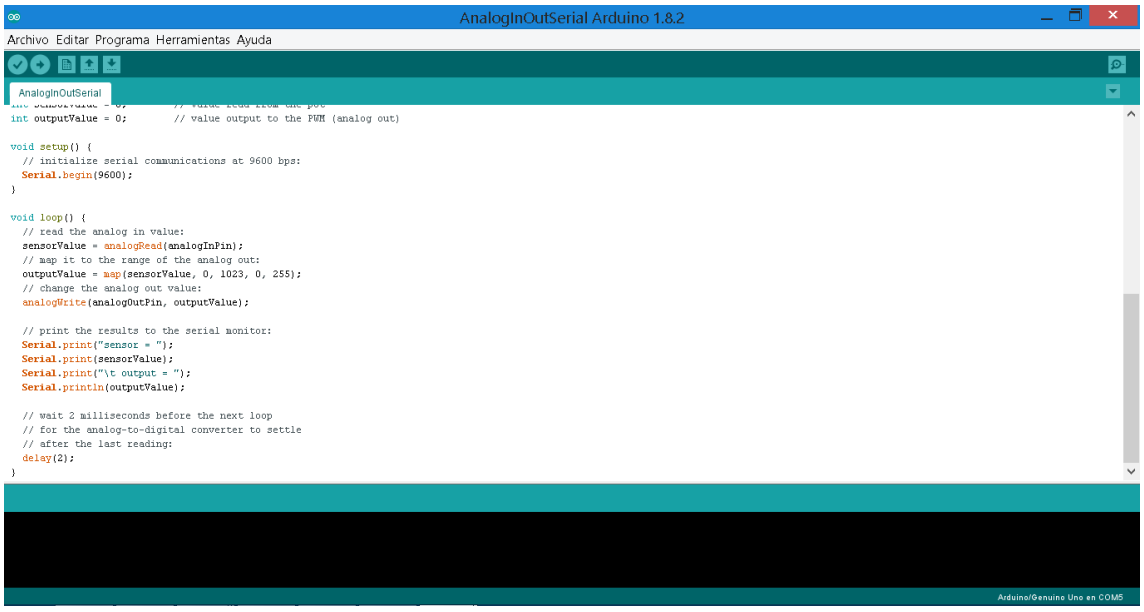
miniBloq:

Si hablamos de RobotGroup y sus placas DuinoBot, no podemos olvidarnos de miniBloq.¿Qué es miniBloq? Es un entorno de desarrollo, diseñado para las placas DuinoBot. Tiene una interfaz sencilla, la cual permite un aprendizaje rápido. Cabe destacar que no todas las versiones de miniBloq, sirven para todas las placas. En el caso de la DuinoBot v2.3 HID, tuvimos que bajar la versión 0.82 Beta de miniBloq para que funcionara adecuadamente.



Arduino IDE:

Es el IDE oficial para trabajar con Arduino o DuinoBot. El mismo no es para personas que recién empiezan en la programación, se requieren conocimientos de C++ para poder usarlo. Al ser el IDE oficial, a la hora de cargar programas, no nos encontraremos con ningún error.



Incorporación en la Curricula:

Espacio curricular:

Tecnología de los Sistemas Informáticos.

Fundamentación:

Eje Tecnología aplicada al Hardware

El desarrollo de este eje permite conocer los distintos dispositivos electrónicos que conforman los sistemas tecnológicos de procesamiento de información y de la comunicación. Este eje se propone analizar cómo se interrelacionan los mismos, para su correcto funcionamiento y comparar estos sistemas informáticos con sistemas tecnológicos que tengan la misma función.

Asimismo, admite la posibilidad de clasificar los sistemas tecnológicos y sus componentes según función y uso.

El tratamiento de los saberes que se proponen en este eje es fundamental para la orientación, proporciona la oportunidad de conocer y analizar la diversidad tecnológica en la que se encuentran inmersos nuestros alumnos.

¿Por qué utilizar Arduino, DuinoBot en Tecnología de los Sistemas Informáticos?

Como podemos ver en uno de los ejes de la materia, se propone que se conozcan los distintos dispositivos electrónicos que conforman los sistemas tecnológicos de procesamiento de información y de la comunicación. Una forma de hacerlo, con un costo bajo, es utilizando las placas este tipo de placas y sensores.

Generalmente, cuando en una clase se enseñan temas, como los sistemas informáticos, lo hacen a través de definiciones, pero los alumnos, no ven la comunicación directa entre el hardware y el software. Una forma muy sencilla de mostrar la comunicación entre la parte tangible e intangible de la computadora, es utilizando este tipo de placas y sensores.

De una forma muy simple y entretenida, pueden ver la diferencia entre hardware y software, cuales son dispositivos de salida (LED's , pantallas LCD) , de entrada (Sensores de Llama, humo), los distintos tipos de software (Sistemas, aplicación y programación).

En nuestra experiencia con la robotica, en festivales de ciencia, hemos notado que los chicos se muestran interesados a la hora de trabajar con placas como las Arduino o DuinoBot. De esta forma, los alumnos, van a poder entender los conceptos y tener resultados de una forma inmediata.

Actividad:

Las siguientes actividades, fueron testeadas en festivales de ciencia organizados por la UNLPam (Universidad Nacional de La Pampa) e Interactuando con la Ciencia (programa de comunicación científica), contando con un total de 60 minutos para desarrollar las actividades.

1era Parte Introducción a mBlock:

Luego de mostrar el entorno de desarrollo y explicar cómo acomodar los bloques, se les pide a los alumnos que realicen las siguientes actividades:

I. Hacer que el panda camine 10 pasos.
II. Hacer que el camine 3 veces 10 pasos. *(Repetición)*
III. Hacer que el panda camine 3 veces 10 pasos y a su vez diga ¡Hola!

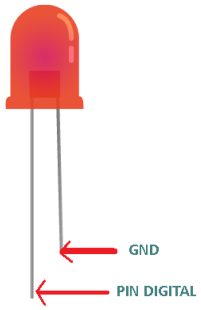
2da Parte Introducción a Arduino y los bloques de Robot de mBlock

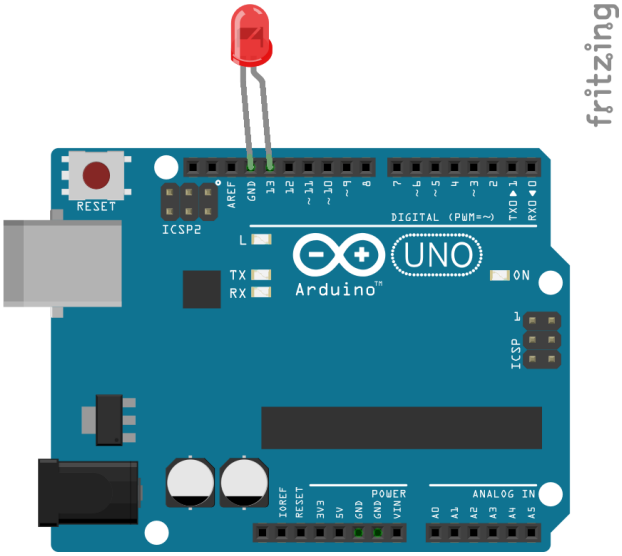
Breve introducción sobre Arduino.

I. Encender un Led
II. Encender y apagar un led, de forma intermitente.
III. Encender más de un LED (Protoboard).

Para la primera actividad, se les pide que hagan el “hola mundo” de arduino, el cual consiste en encender un led.

Para esto vamos a colocar el led directo a la placa, en el pin 13 y al lado de ese pin, se encuentre el GND. Recordemos que el led tiene dos patas, la más larga seria el pin digital (en nuestro caso sería el pin 13) y la más corta es el GND.





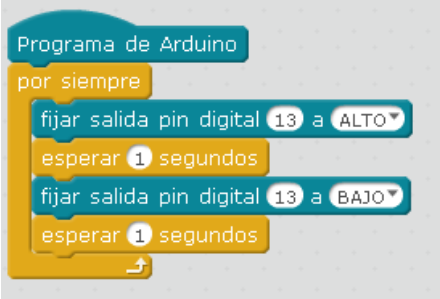
Como un led es un dispositivo de salida, tenemos que decirle al led que mostrar, para eso utilizamos el bloque “Fijar salida del pin digital”, donde enviamos ALTO, para indicar que el led debe encenderse, de lo contrario para apagarlo tenemos que enviar BAJO. El código quedaría de la siguiente forma:

Parece estar todo bien, pero aunque no parezca algo falta. La placa arduino está constantemente recibiendo y enviando información, eso que quiere decir que todo programa cargado, tendría que estar repitiéndose para siempre. Lo que falta, es el bloque de repetición por siempre.

De esta forma escribimos código en arduino, siempre se necesita un bloque de repetición *por siempre*.

En la 2da Actividad, se les pide que enciendan y apaguen un led, de forma intermitente.

Para esto necesitamos determinar un tiempo para el apagado y encendido. Con el bloque de *esperar* podemos determinar una espera en segundos. Fijamos que el led en ALTO, y que luego de unos segundos quede en BAJO.

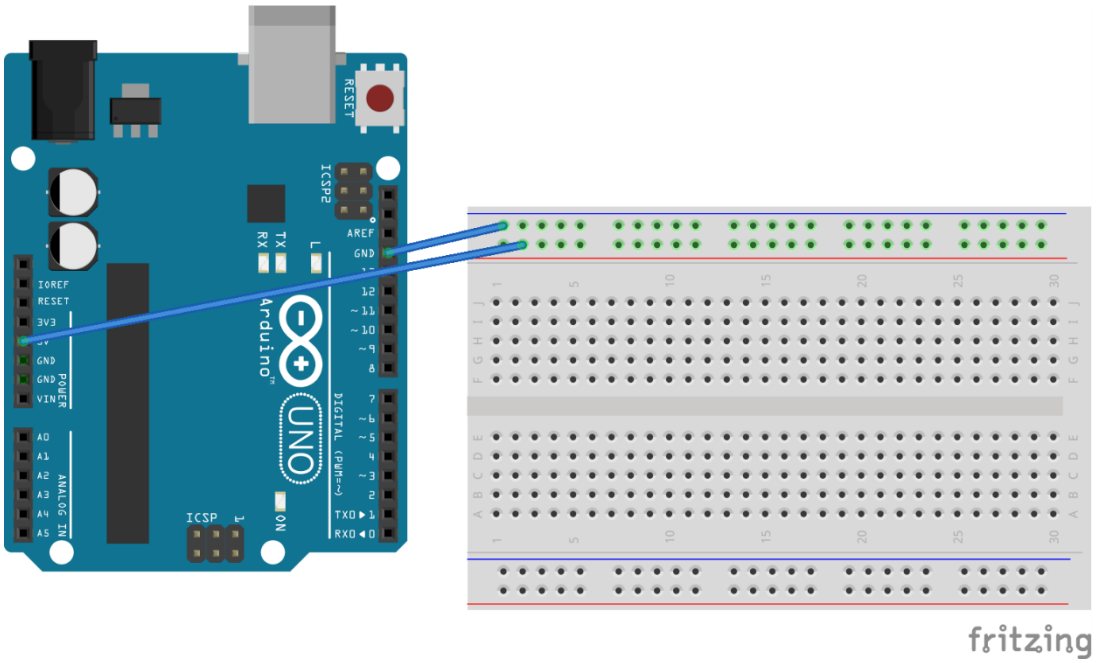


De esta forma, repitiendo por siempre, cada 1 segundo el led se prendera y apagará.

3era Actividad. En esta actividad, tenemos que mostrar el uso de uno Protoboard. Sirve para poder extender el número de conexiones en la placa, es una herramienta muy útil, a la hora de trabajar con varios sensores, leds, etc.

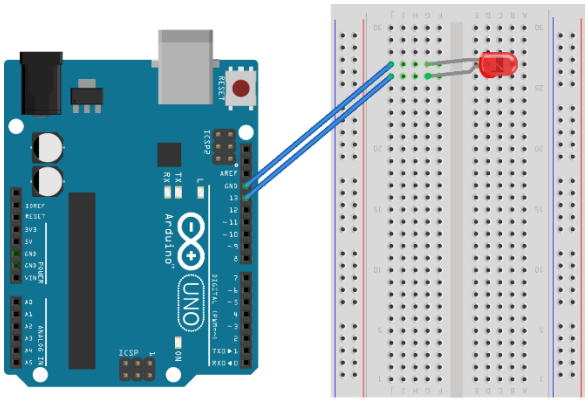
¿Cómo podemos usar la protoboard?

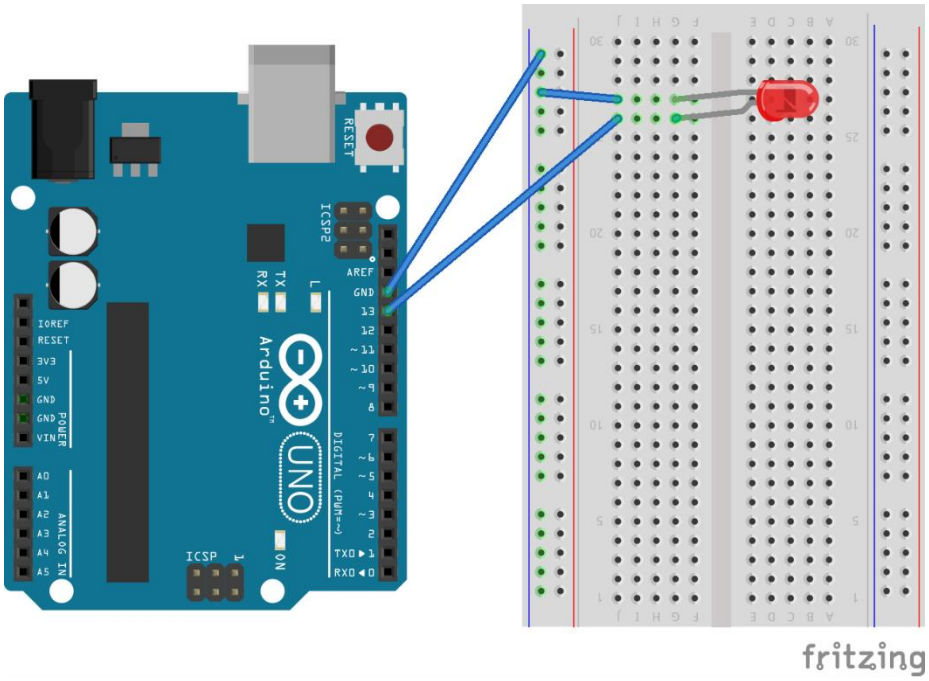
Podemos observar que en la protoboard, hay dos líneas de colores diferencias en dos columnas. La azul indica GND, es decir, podemos conectar un cable al GND de la placa arduino, a uno de los orificios de GND de la protoboard. Al conectarlo, ese GND va a alimentar a todo el resto de la línea, es decir que podemos conectar varios GND. Lo mismo ocurre con la parte roja, en ella conectamos la energía, en este caso 5V o 3.3v, y de igual forma podemos alimentar a toda la columna.



Nótese que en la imagen, cuando conectamos el GND a la línea azul, se resalta con verde el resto de la columna, eso quiere decir que la placa esta alimentado a toda esa línea horizontal. Lo mismo ocurre con 5V pin.

Formas alternativas de conectar un led con una protoboard.





3era Parte Sensores.

En esta instancia, luego de una breve introducción a los sensores, vamos a encender un led a partir de un sensor de llama. ¿Cómo funciona esto?

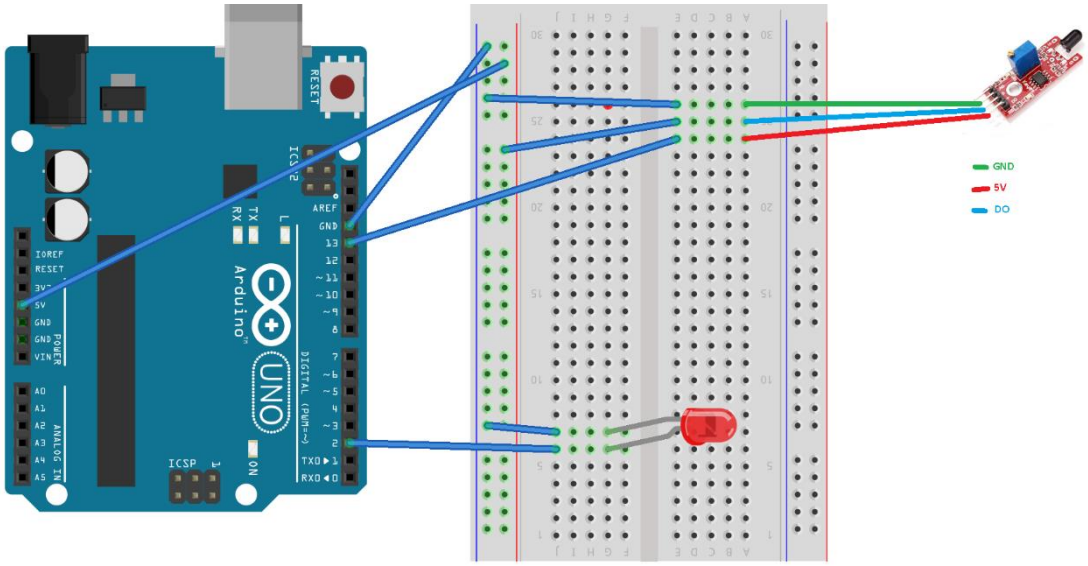
Un sensor de llama, es un dispositivo de entrada, el cual nos envía una información cuando detecta la luz de una llama a través del infrarrojo. Utilizando el Pin digital (DO), el sensor envía pulsos digitales en forma de 0 o 1 (LOW, HIGH).

Como es un dispositivo de entrada, tendremos que leer del pin digital. Apenas se recibe información sea 1 o 0, deberá encender o apagar el led. Si la lectura del sensor de llama, es ALTO, quiere decir que hay una llama cerca, por lo tanto debemos encender un LED, caso contrario queda apagado.

Para este programa, utilizaremos los bloques de leer pin digital, fijar salida pin digital, y el bloque de condición SI..entonces, y el operador igual.

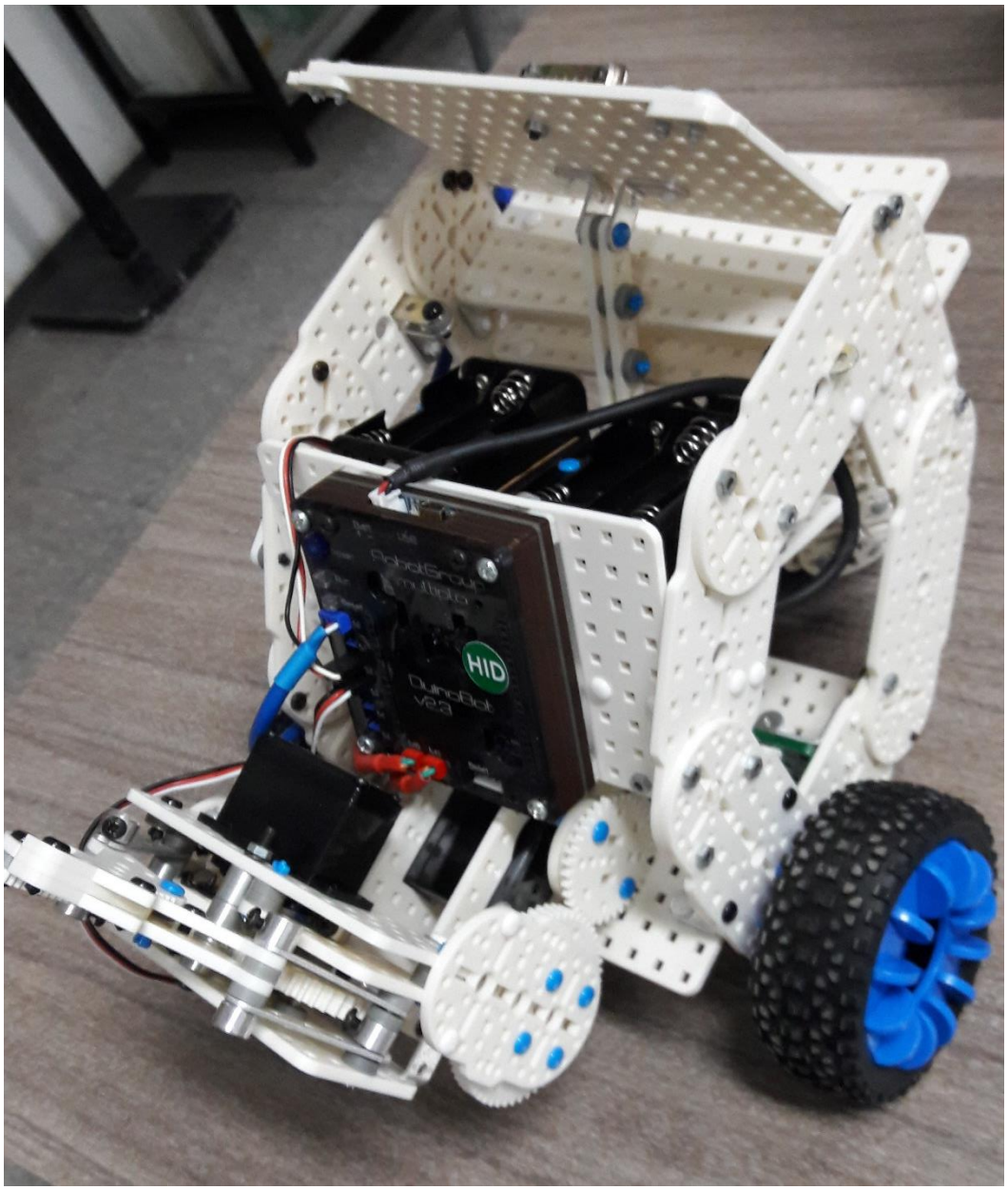
Código:

Conexión:



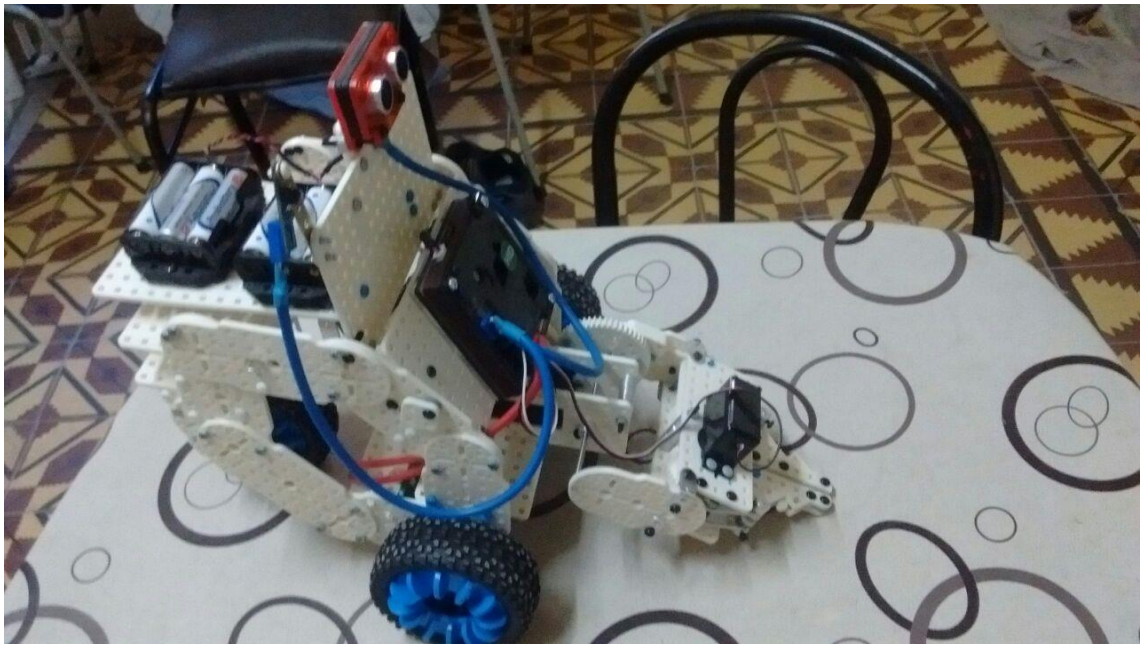
Experiencia Kit de Robótica RobotGroup

Durante la cursada de Informática Educativa 2, tuvimos la oportunidad de experimentar con un kit de robótica de la empresa RobotGroup, el robot n8, el cual posee una placa DuinoBot v2.3 HID.

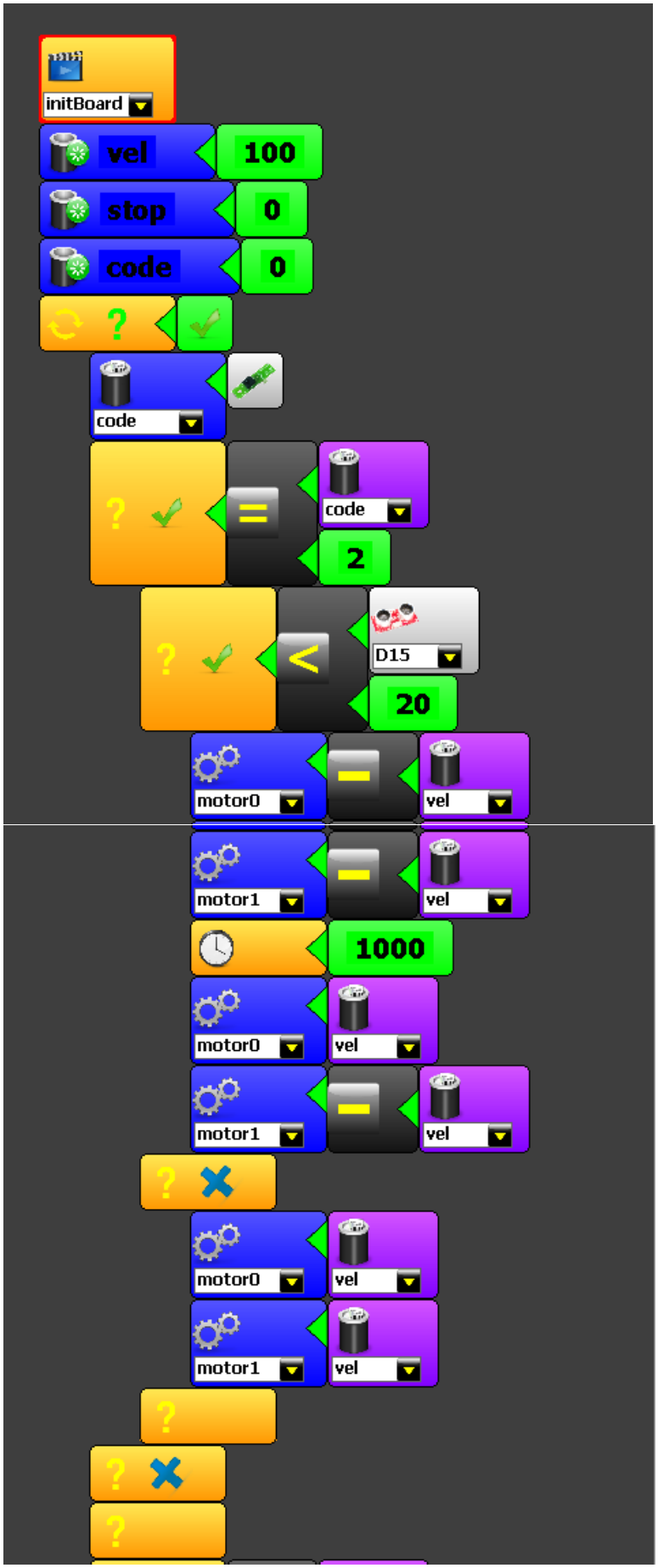


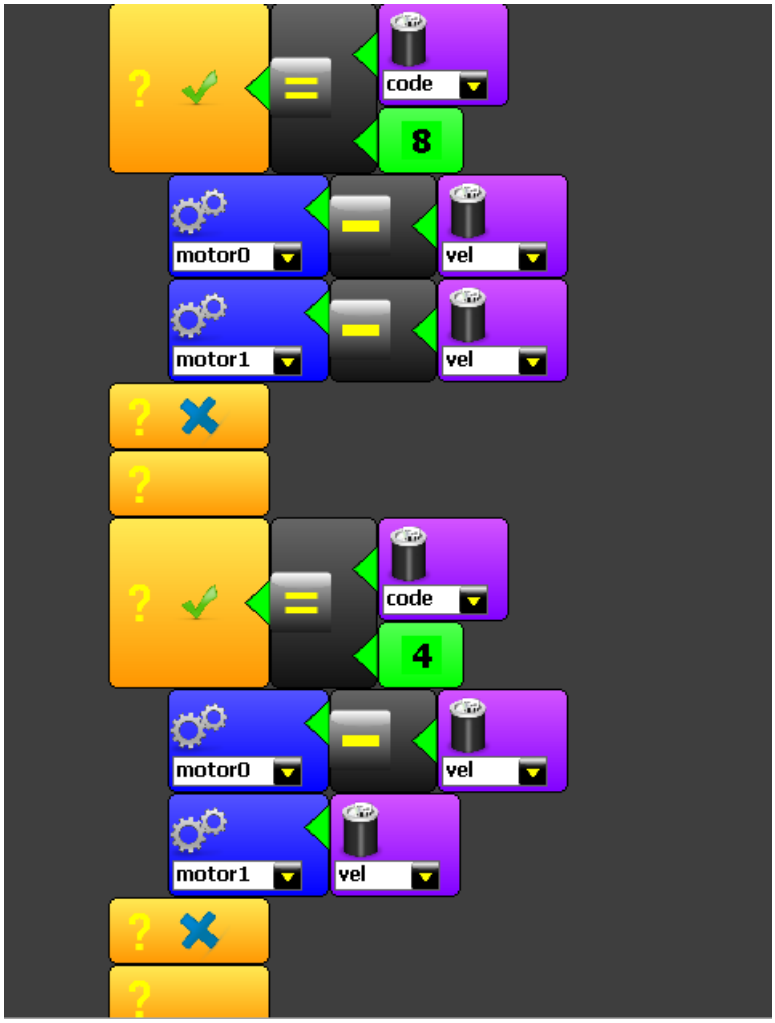
Al principio, nuestro avance fue algo lento, ya que al utilizar un IDE reciente, como lo es la versión 0.83, no funcionaba de forma correcta con la placa. Por eso, tuvimos que descargar la versión 0.82 Beta. A partir de allí, el aprendizaje fue mucho más sencillo, al no tener complicaciones con el IDE. El proyecto consistía en hacer el que el robot, se mueva a partir de las instrucciones dadas por un control remoto, y a su vez, si encuentra un obstáculo en su camino, poder retroceder.

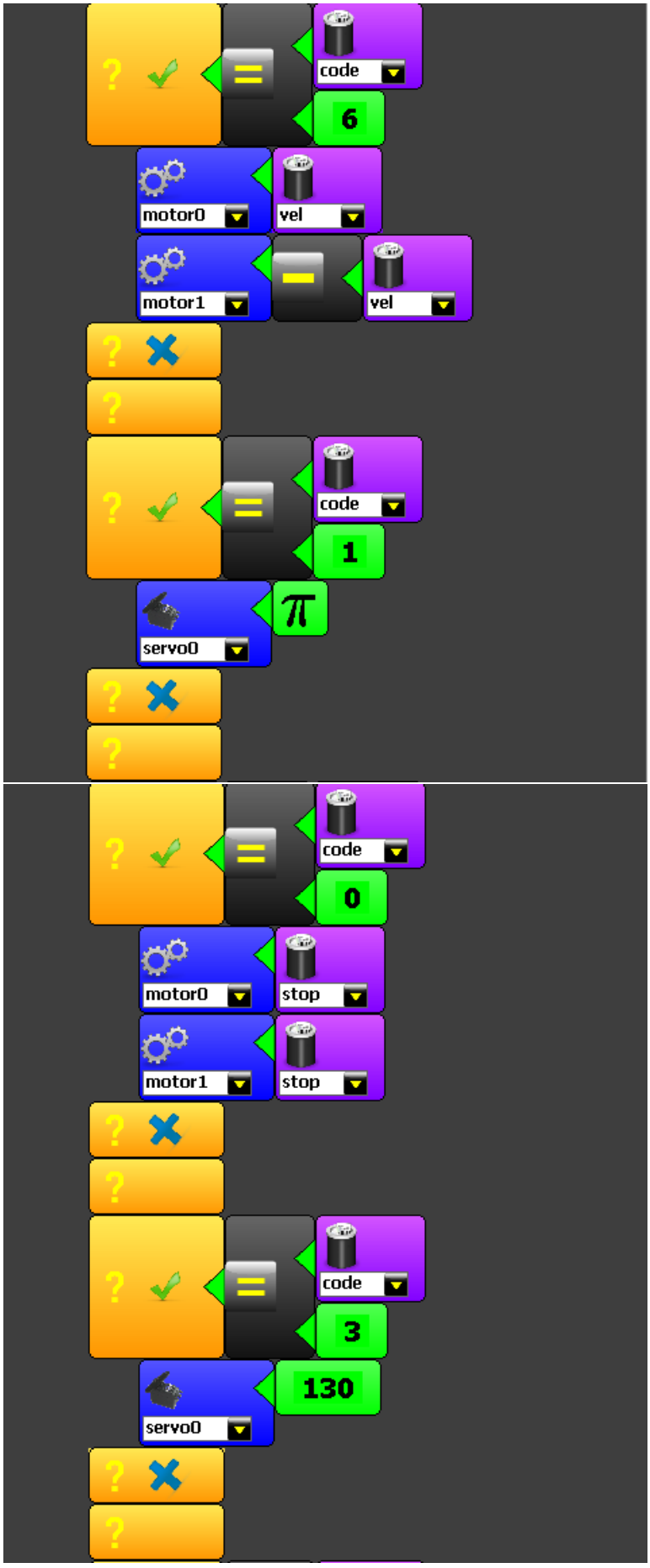
Luego de unas ligeras modificaciones, a la ubicación de las pilas, el sensor ultrasónico y la pinza, el robot quedo de la siguiente manera:

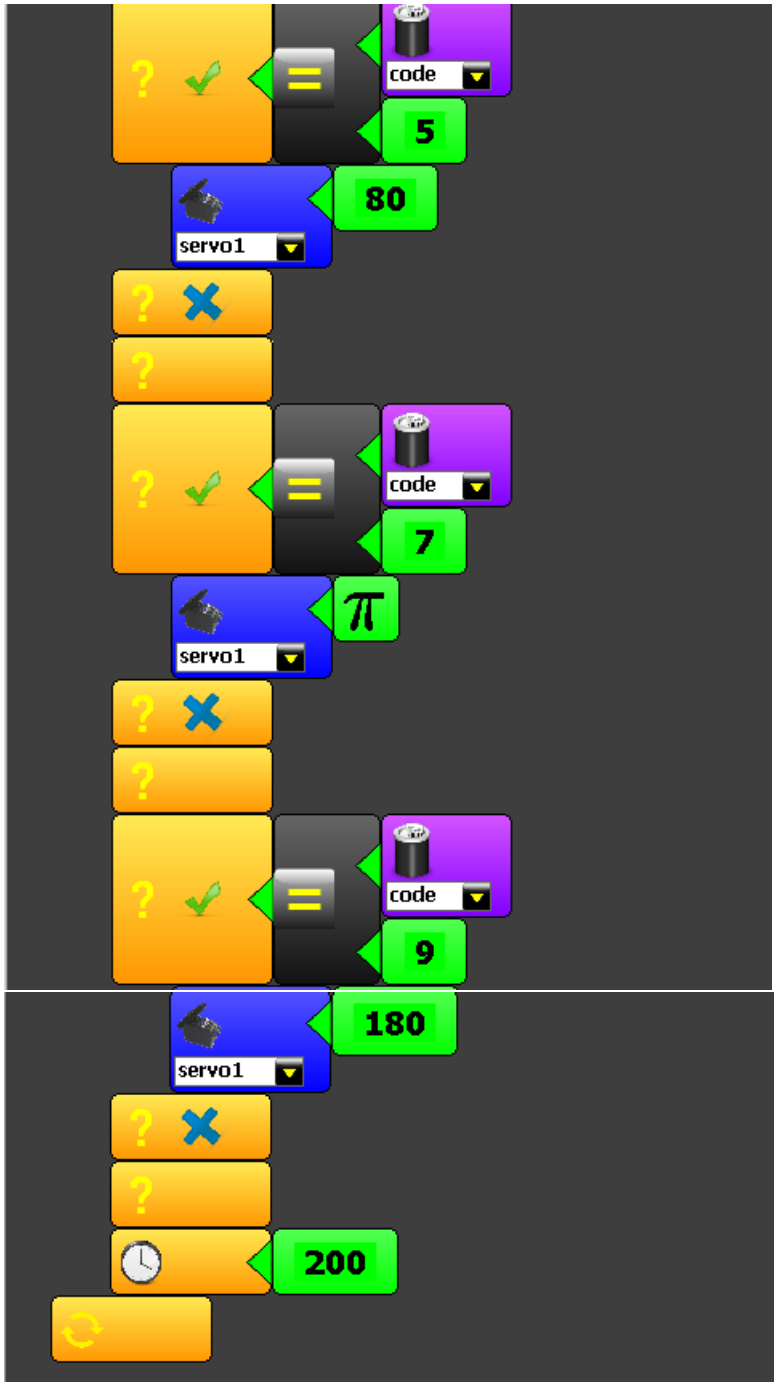


Codigo miniBloq:


initBoard
vel
100
stop
0
code
0
code
code
2
D15
20
motor0
vel
motor1
vel
1000
motor0
vel
motor1
vel
motor0
vel
motor1
vel


code
8
motor0
vel
motor1
vel
code
4
motor0
vel
motor1
vel


code
6
motor0
vel
motor1
vel
code
1
servo0
π
code
0
motor0
stop
motor1
stop
code
3
130
servo0


code
5
80
servo1
code
7
π
servo1
code
9
180
servo1
200

Codigo en C:

```
#include <mbq.h>
#include <PingIRReceiver.h>

void setup()
{
	initBoard();
	float vel = 100;
	float stop = 0;
	float code = 0;
	while(true)
	{
		code = irReceiver.getIRRemoteCode();
		if(((int)(code)==(int)(2)))
		{
			if((pingMeasureCM(D15)<20))
			{
				motor0.setPower(-(vel));
				motor1.setPower(-(vel));
				delay(1000);
				motor0.setPower(vel);
				motor1.setPower(-(vel));
			}
			else
			{
				motor0.setPower(vel);
				motor1.setPower(vel);
			}
		}
		else
		{
```

```
}
if(((int)(code)==(int)(8)))
{
        motor0.setPower(-(vel));
        motor1.setPower(-(vel));
}
else
{
}
if(((int)(code)==(int)(4)))
{
        motor0.setPower(-(vel));
        motor1.setPower(vel);
}
else
{
}
if(((int)(code)==(int)(6)))
{
        motor0.setPower(vel);
        motor1.setPower(-(vel));
}
else
{
}
if(((int)(code)==(int)(1)))
{
        servo0.attachAndWrite(M_PI);
}
else
{
```

```
}
if(((int)(code)==(int)(0)))
{
        motor0.setPower(stop);
        motor1.setPower(stop);
}
else
{
}
if(((int)(code)==(int)(3)))
{
        servo0.attachAndWrite(130);
}
else
{
}
if(((int)(code)==(int)(5)))
{
        servo1.attachAndWrite(80);
}
else
{
}
if(((int)(code)==(int)(7)))
{
        servo1.attachAndWrite(M_PI);
}
else
{
}
if(((int)(code)==(int)(9)))
```

```
                {
                        servo1.attachAndWrite(180);
                }
                else
                {
                }
                delay(200);
        }
}
void loop()
{
}
```

Bibliografia:

www.arduino.cc

www.prometec.net

http://www.mblock.cc/

http://www.visualino.net/index.es.html

http://robotgroup.com.ar/es/