Robótica



Objetivo Gral. Del curso



Involucrarse en el mundo de la robótica mediante la generación de conocimiento e inmersión de temáticas de física, electrónica y programación.

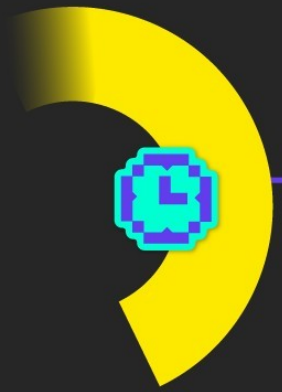
Conocer los conceptos y principios que conlleva la inmersión en el mundo de la robótica, para la automatización de procesos.

Objetivos

- Asimilar los conceptos básicos de la electrónica y mecánica. Identificando los principales componentes.
- Reconocer las herramientas básicas a la hora de querer hacer una máquina mecatrónica.
- Identificar que es un robot y cuales son las principales características.
- Desarrollar prototipos funcionales de robótica.

Horas	
3	Unidades y recursos de programación
2	Introducción a ropa electrónica
2	entradas digitales
2	entradas análogas
2	modelado 3D
6	Taller y presentación proyecto de robótica

Introducción a la robótica



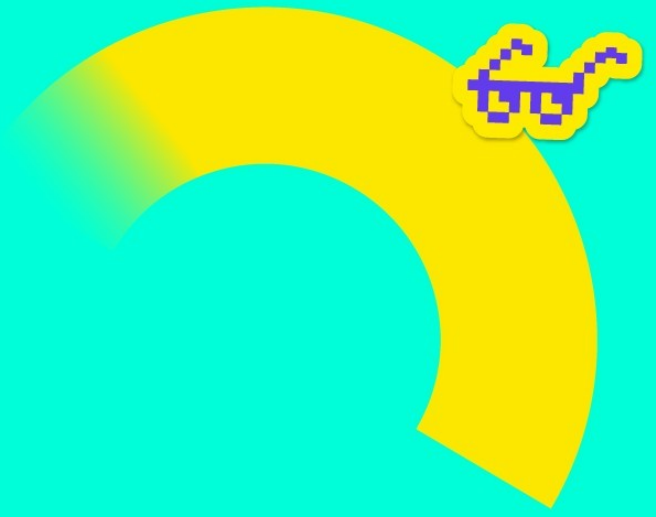
¿Qué es?

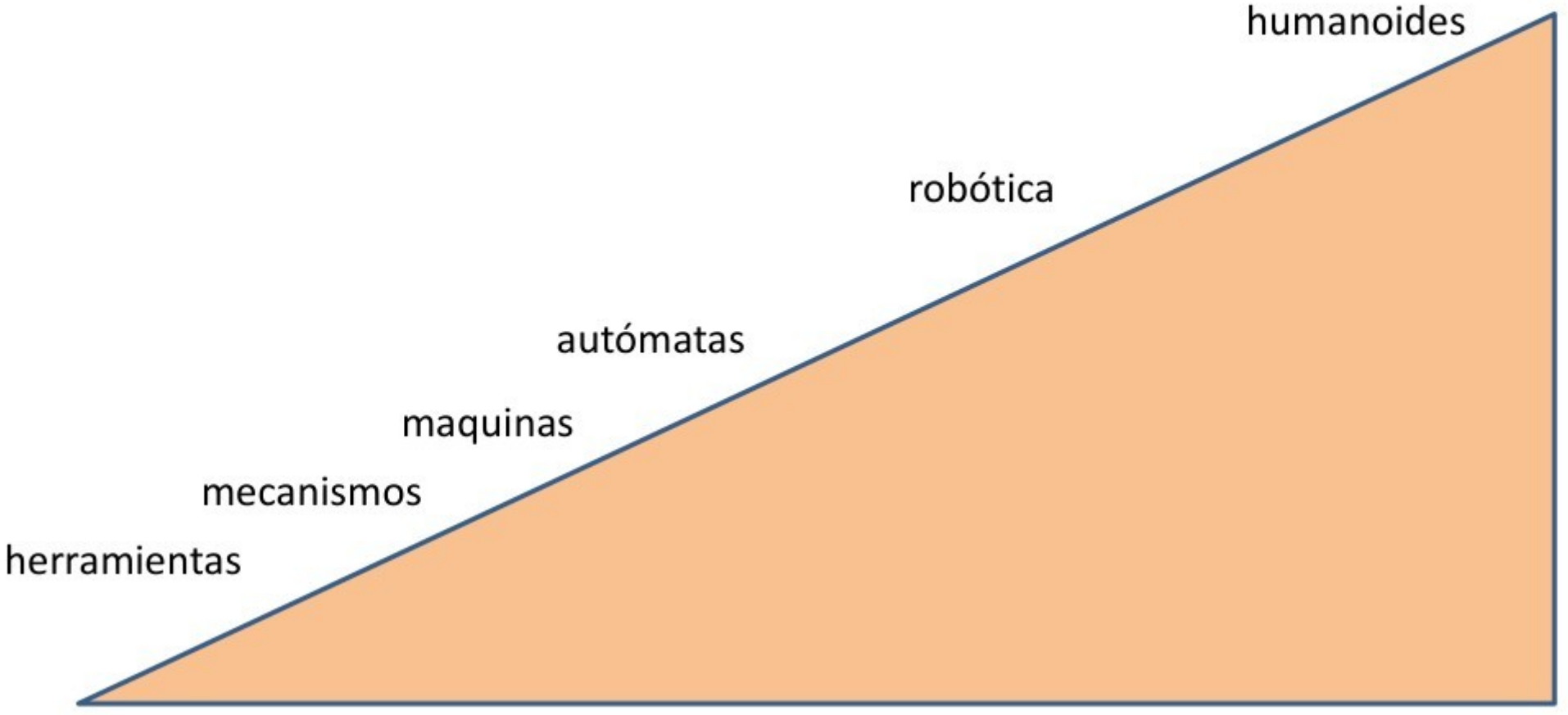


Historia de la robótica

Pero veremos grandes pasos antes de llegar a la robótica, sus antecedentes:

- Herramientas
- mecanismos
- Autómatas
- Mecánicos
- eléctricos
- Robots industriales
- Humanoides



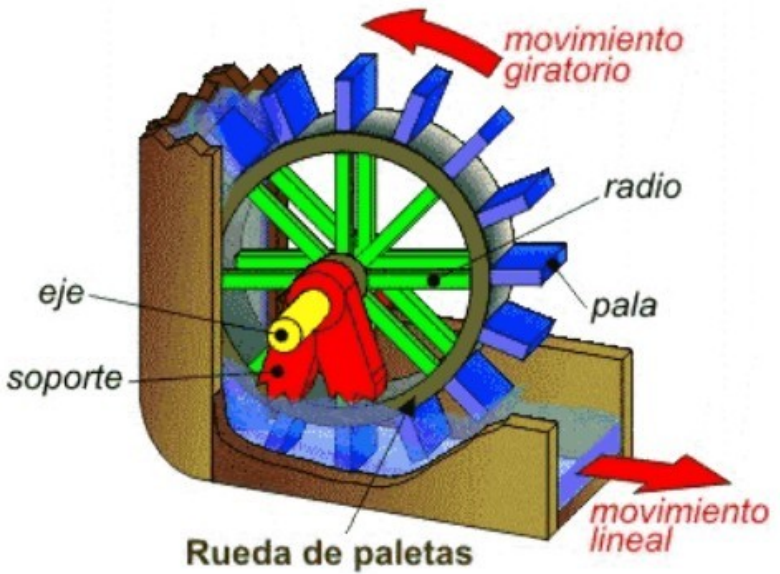

humanoides
robótica
autómatas
maquinas
mecanismos
herramientas

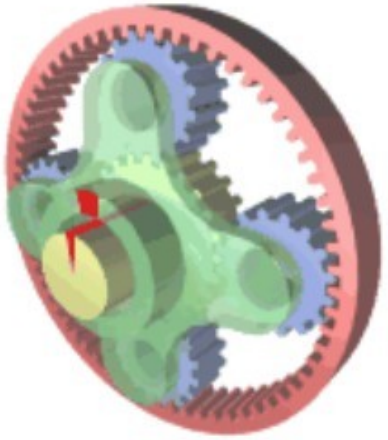
Herramientas

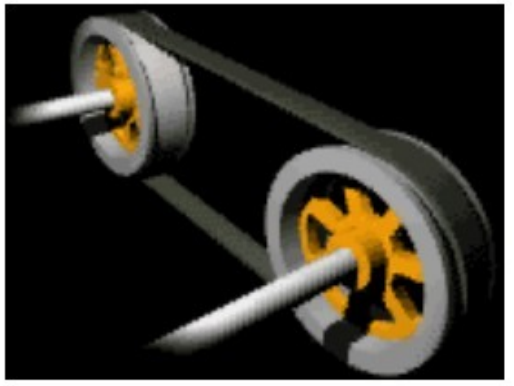


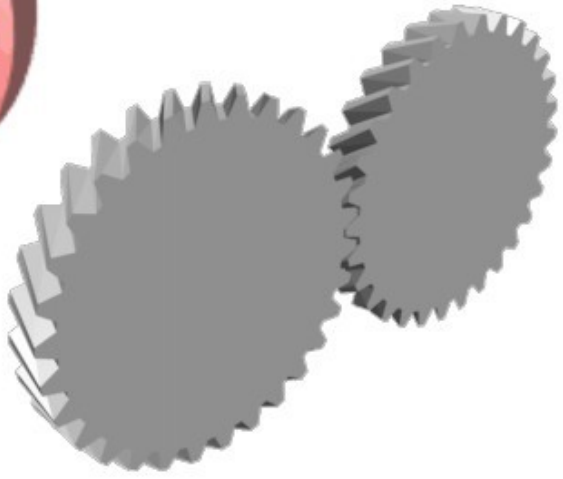
Mecanismos




movimiento giratorio
radio
pala
eje
soporte
movimiento lineal
Rueda de paletas



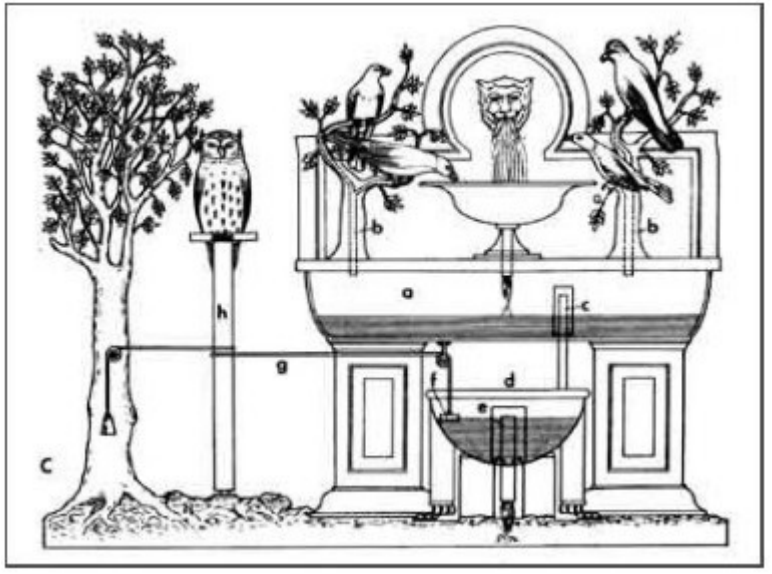








Automatas



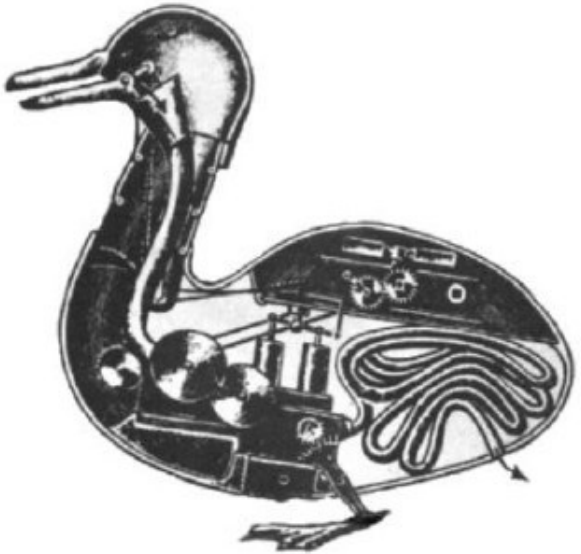


Leonardo Da Vinci construye en el año 1500 un león automático en honor de Luis XII que actúa en la entrada del Rey de Milán.



Además de haber inventado el antecesor del engranaje actual







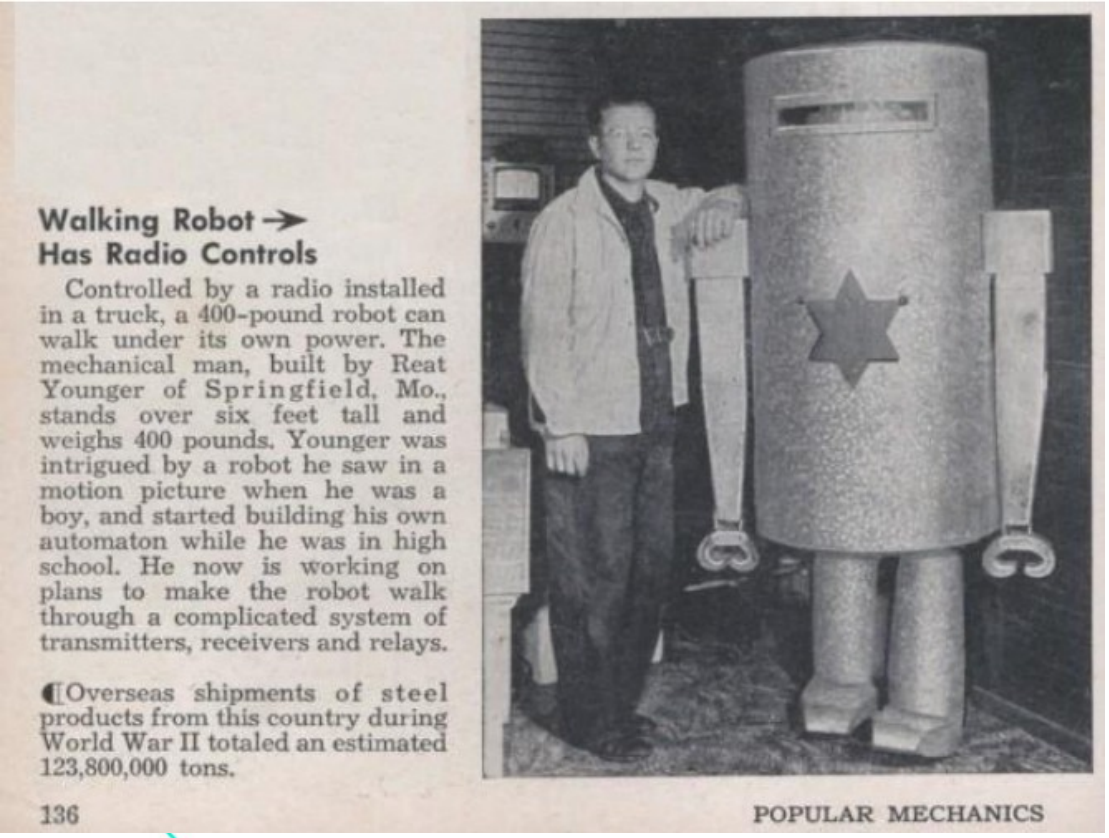
Jacques de Vaucanson, construye el pato, el autómata más conocido; un pato hecho de cobre, que bebe, come, grazna, chapotea en el agua y digiere su comida como un pato real. Previamente construye un flautista y un tamborilero en 1738; el primero consistía en un complejo mecanismo de aire que causaba el movimiento de dedos y labios, como el funcionamiento normal de una flauta.

Automatización



Maquina vapor de James Watt

El abuelo de los robots



Walking Robot → Has Radio Controls

Controlled by a radio installed in a truck, a 400-pound robot can walk under its own power. The mechanical man, built by Reat Younger of Springfield, Mo., stands over six feet tall and weighs 400 pounds. Younger was intrigued by a robot he saw in a motion picture when he was a boy, and started building his own automaton while he was in high school. He now is working on plans to make the robot walk through a complicated system of transmitters, receivers and relays.

◐Overseas shipments of steel products from this country during World War II totaled an estimated 123,800,000 tons.

136 POPULAR MECHANICS

Esta foto corresponde al año 1948, fue publicada en la revista “Mecánica Popular“. Este singular robot era controlado por una radio instalada en un camión, el robot que pesaba unas 400 libras y media unos seis pies de alto, podía caminar por sus propios medios.

El hombre mecánico, fue construido por un joven de Springfield, Missouri, quien fue inspirado por un robot que vio en una película cuando era un niño, y comenzó a construir su propio autómata. Según dice el artículo su creador tiene planes de mejorarlo por medio de un complicado sistema de transmisores, receptores y repetidores.

Enlace: modernmechanix.com

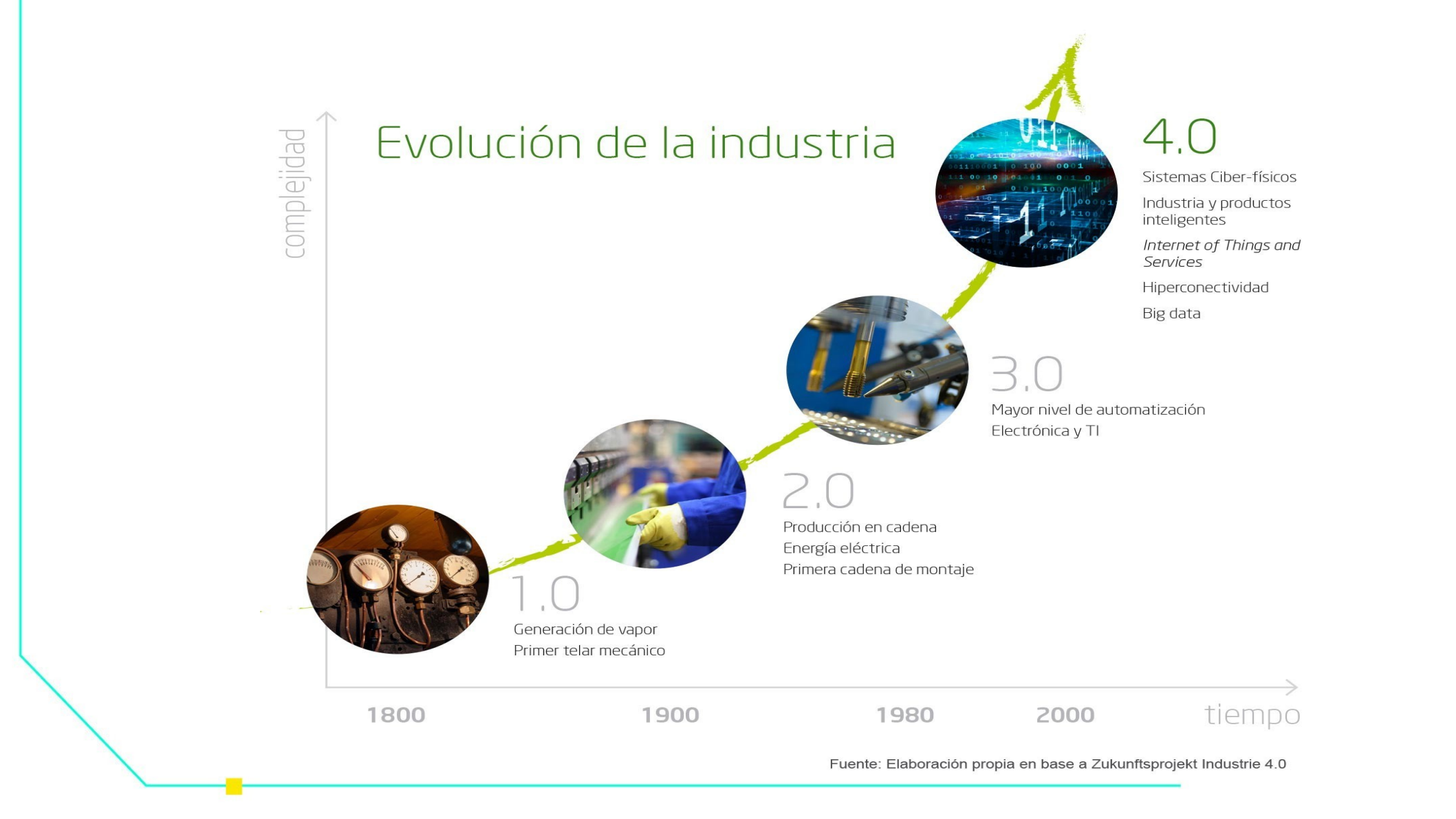

ANDRÉS
OPPENHEIMER
¡SÁLVESE
QUIEN PUEDA!
El futuro del trabajo en la
era de la AUTOMATIZACIÓN



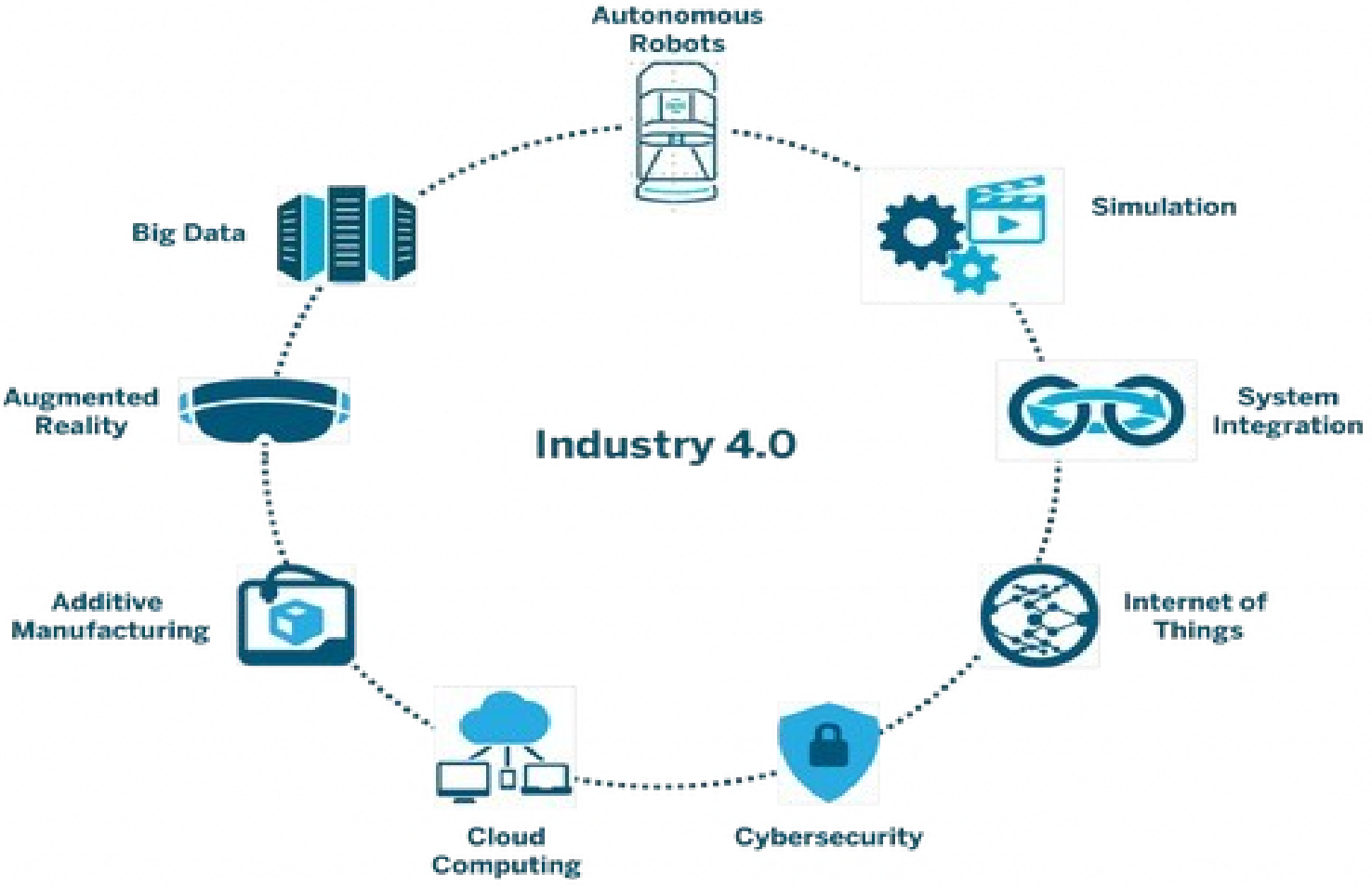


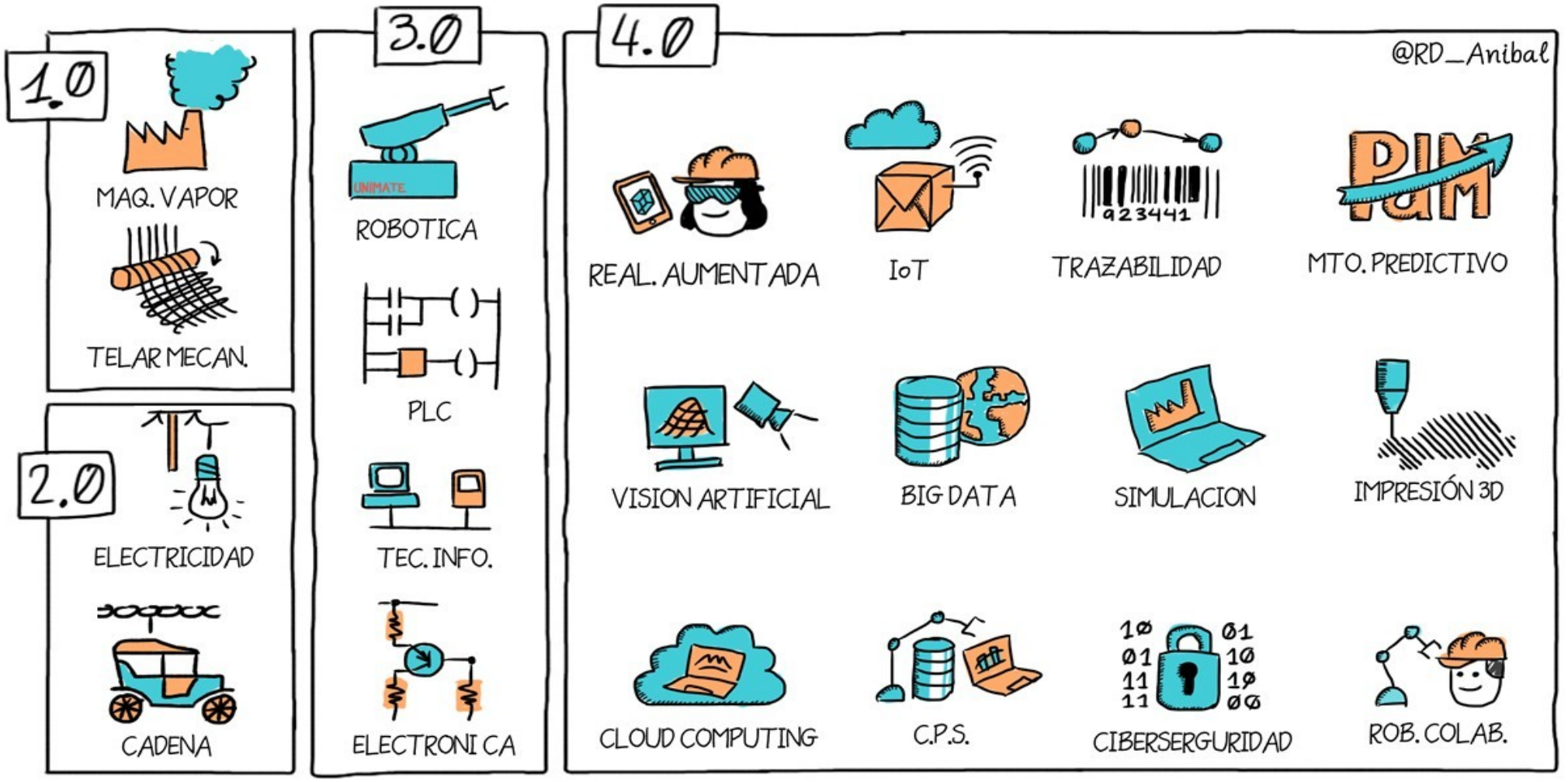


Evolución de la industria



Fuente: Elaboración propia en base a Zukunftsprojekt Industrie 4.0


Autonomous Robots
Simulation
System Integration
Internet of Things
Cybersecurity
Cloud Computing
Additive Manufacturing
Augmented Reality
Big Data
Industry 4.0

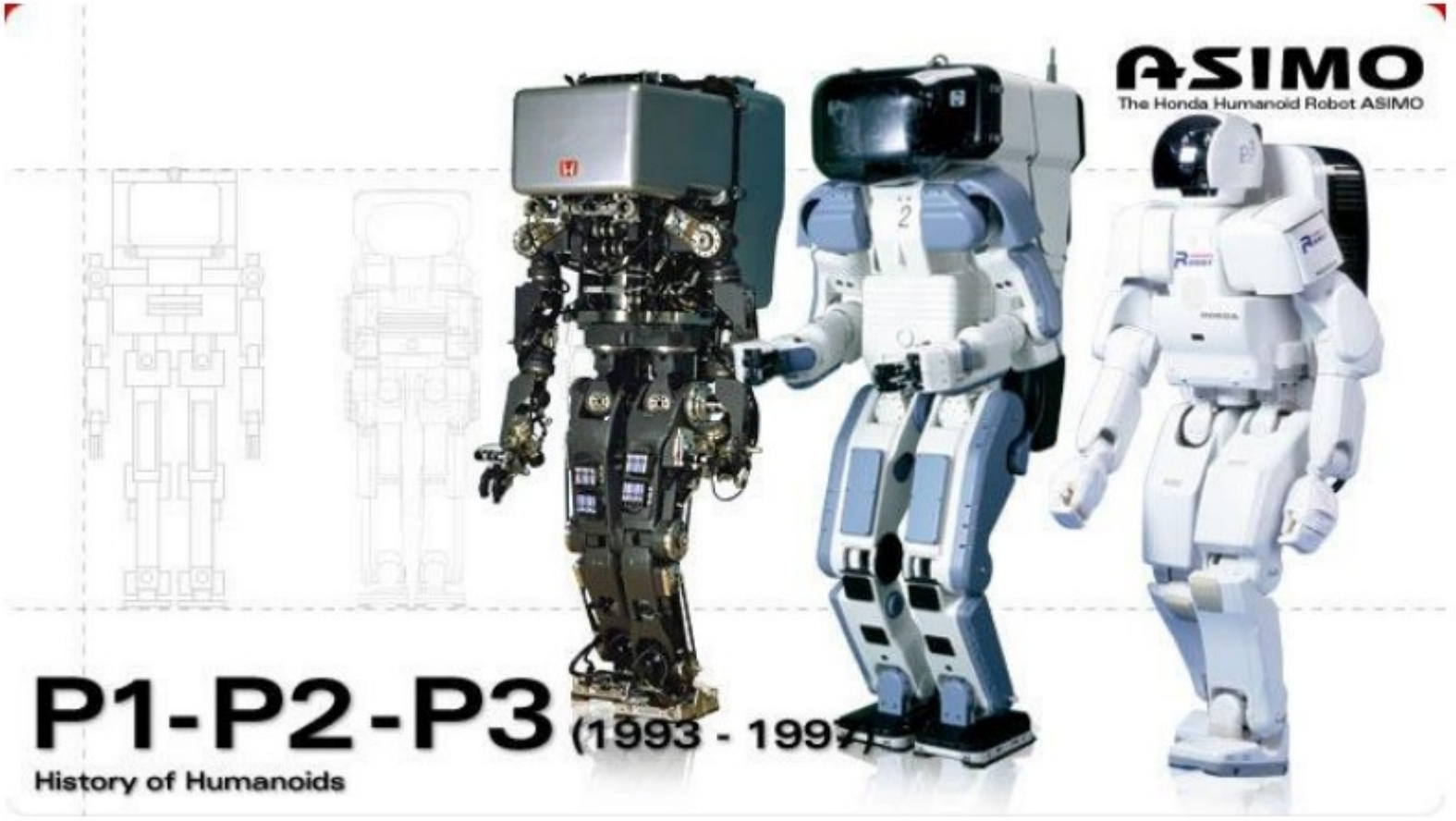

1.0
MAQ. VAPOR
TELAR MECAN.
2.0
ELECTRICIDAD
CADENA
3.0
UNIMATE
ROBOTICA
PLC
TEC. INFO.
ELECTRONICA
4.0
@RD_Anibal
REAL. AUMENTADA
IoT
TRAZABILIDAD
PLM
MTO. PREDICTIVO
VISION ARTIFICIAL
BIG DATA
SIMULACION
IMPRESIÓN 3D
CLOUD COMPUTING
C.P.S.
CIBERSERGURIDAD
ROB. COLAB.


Robótica
Impresión 3D
AI
IoT
Sistemas físicos
i4.0
Sistemas virtuales
Sistemas digitales
Sistemas autónomos
Nano-tecnología
Blockchain
Ciberseguridad

Entertainment Robot AIBO

AIBO will win your heart and entertain your mind.



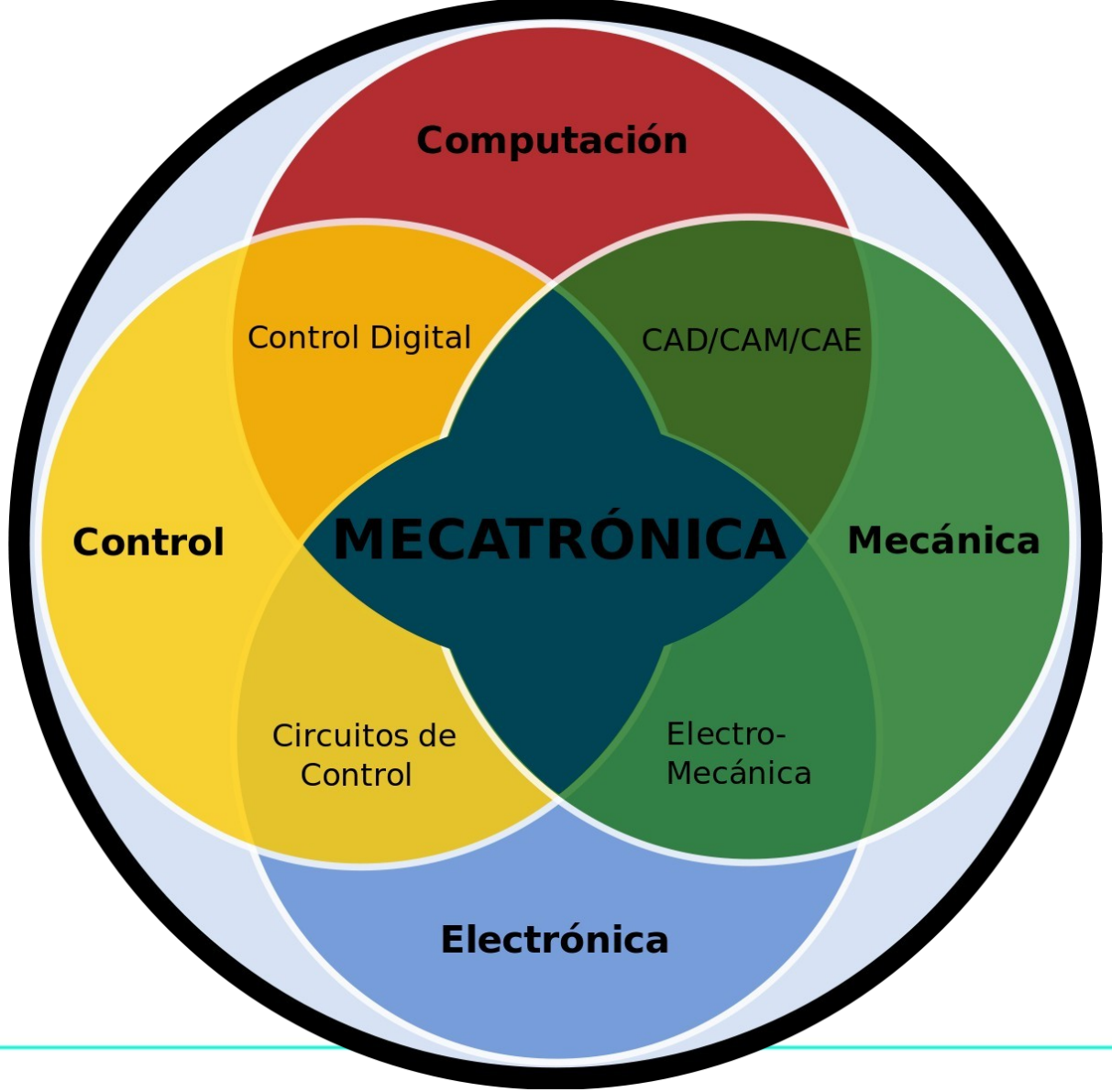

ASIMO
The Honda Humanoid Robot ASIMO
P1-P2-P3 (1993 - 199
History of Humanoids

¿Qué es un robot?

“Máquina controlada por ordenador y programada para moverse, manipular objetos y realizar trabajos a la vez que interacciona con su entorno. Los robots son capaces de realizar tareas repetitivas de forma más rápida, barata y precisa que los seres humanos”.

¿Qué es la robótica?

Básicamente, la robótica se ocupa de todo lo concerniente a los robots, lo cual incluye el control de motores, mecanismos automáticos neumáticos, sensores, sistemas de cómputos


Computación
Control Digital
CAD/CAM/CAE
Control
MECATRÓNICA
Mecánica
Circuitos de Control
Electro-Mecánica
Electrónica

Tipos de robots

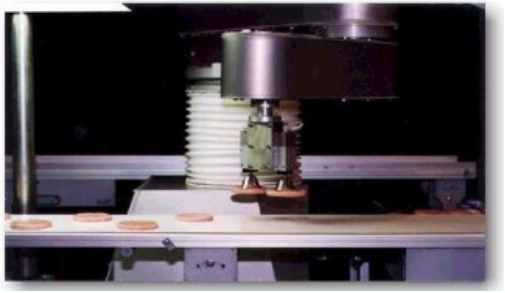


Manipuladores



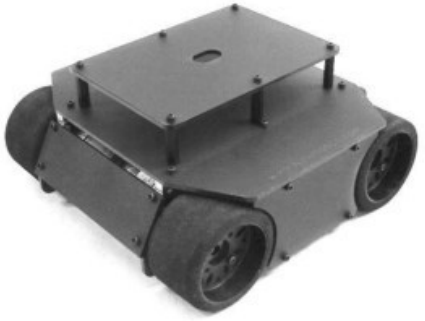






Terrestres







Aéreos















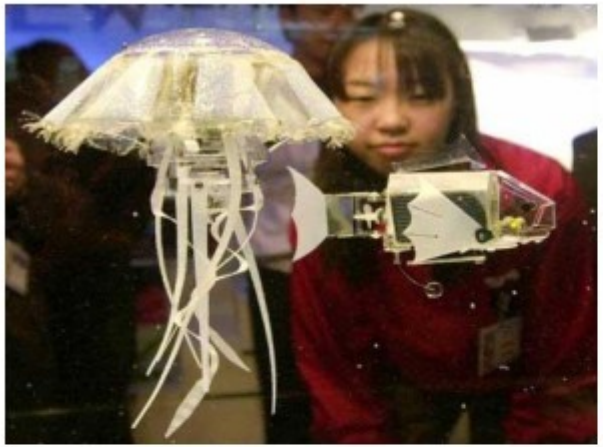


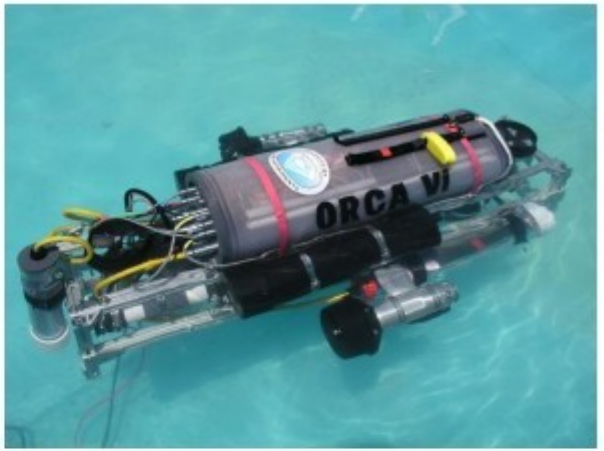


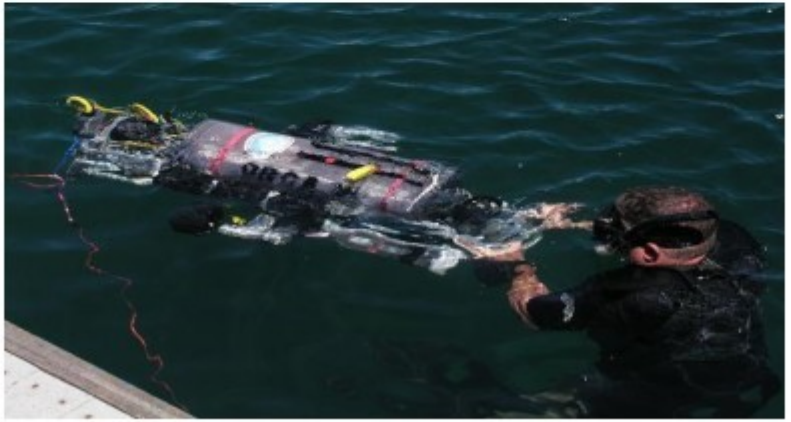




Acuáticos







Espaciales



OPPORTUNITY



ROCKY IV





Didácticos y/o entretenimiento












mindstorms




OLO







BIOLOID
PREMIUM Kit



Prótesis







Sociales y/o doméstico



EMIEW



PAPERO

Androides

ASIMO





HOAP-1



Mini-Robots









Mini-Robots



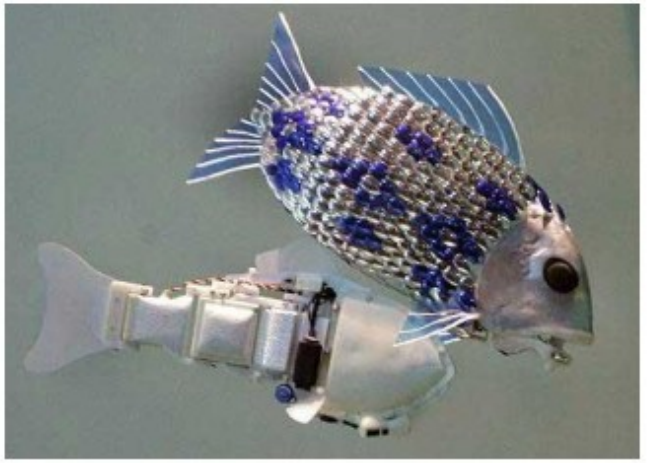


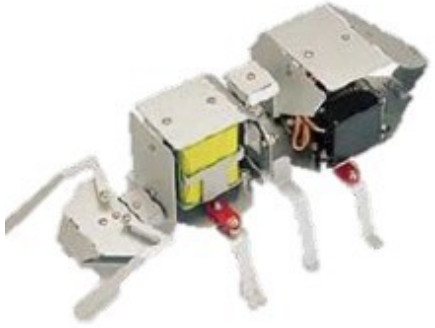




Bio-inspirados





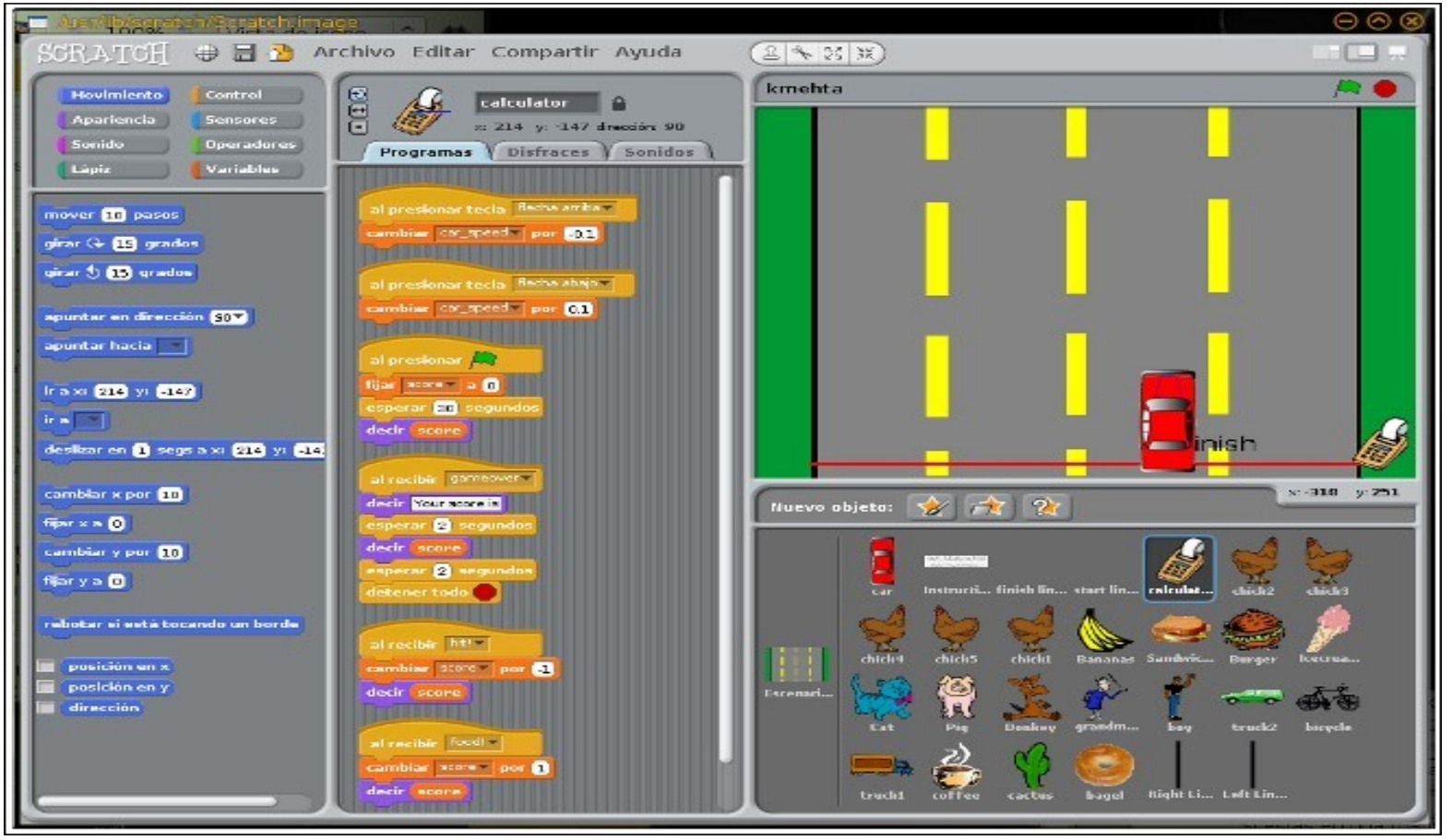




Programación




por siempre
imagina
programa
comparte
SCRATCH


SCRATCH
Archivo Editar Compartir Ayuda
Movimiento
Control
Apariencia
Sensores
Sonido
Operadores
Lápiz
Variables
calculator
Programas
Disfraces
Sonidos
mover 10 pasos
girar 15 grados
girar 15 grados
apuntar en dirección 90
apuntar hacia
ir a x: 214 y: -147
ir a
deslizar en 1 segs a x: 214 y: -14
cambiar x por 10
fijar x a 0
cambiar y por 10
fijar y a 0
rebotar si está tocando un borde
posición en x
posición en y
dirección
al presionar tecla Flecha arriba
cambiar car_speed por -0.1
al presionar tecla Flecha abajo
cambiar car_speed por 0.1
al presionar
fijar score a 0
esperar 30 segundos
decir score
al recibir gameover
decir Your score is
esperar 2 segundos
decir score
esperar 2 segundos
detener todo
al recibir hit
cambiar score por -1
decir score
al recibir Food!
cambiar score por 1
decir score
kmehta
Nuevo objeto:
car
calculat...
chick2
chick3
chick4
chick5
chick1
Bananas
Burger
Escenari...
Cat
Pig
Donkey
bay
truck2
bicycle
truck1
coffee
cactus
bagel

Retos con Scratch

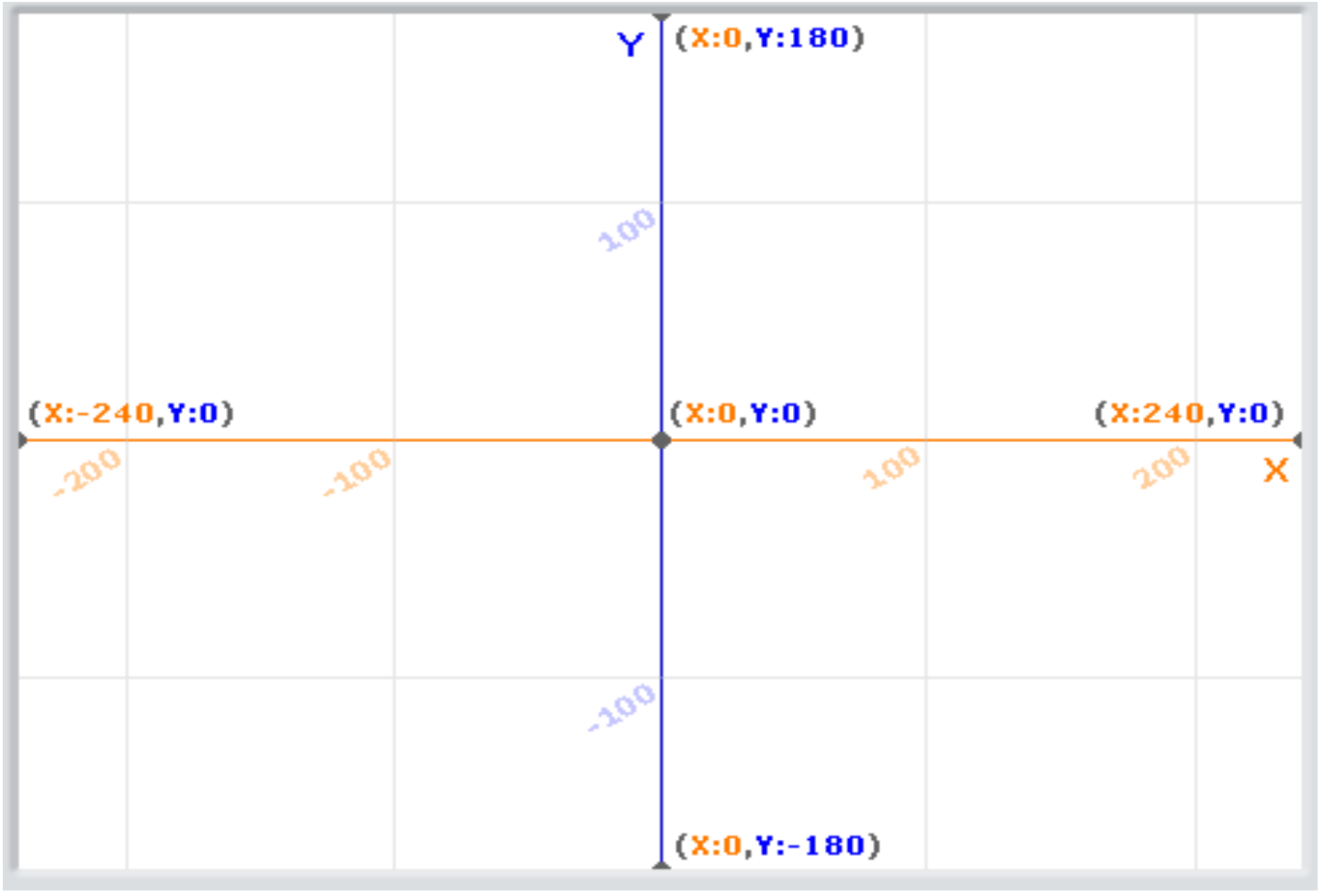


- Saludar

- Mover 10 pasos

- Mover 100 pasos

- Empezar desde (-300,0)


Y
(X:0,Y:180)
100
(X:-240,Y:0)
(X:0,Y:0)
(X:240,Y:0)
-200
-100
100
200
X
-100
(X:0,Y:-180)

Ahora vamos a ver **estructuras de control**

Estructuras de control



- Mover 100 pasos y
- rebotar si toca el borde

- Mover 100 pasos
- rebotar si toca el borde
- Cambiar de color

- Mover 100 pasos
- rebotar si toca el borde
- esperar 1 segundo
- Cambiar de color

- Mover 100 pasos
- rebotar si toca el borde
- esperar 1 segundo
- Cambiar de color
- Que el gato maulle

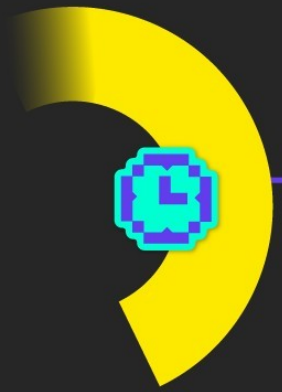
- guardar proyecto

- Realizar:
  - 1. Cuadrado
  - 2. Triángulo equilátero
  - 3. Círculo
  - 4. Pentágono
  - 5. Hexágono
  - 6. Octágono

- 1. Cuadrado
- 2. Triángulo equilátero
- 3. Círculo
- 4. Pentágono
- 5. Hexágono
- 6. Octágono

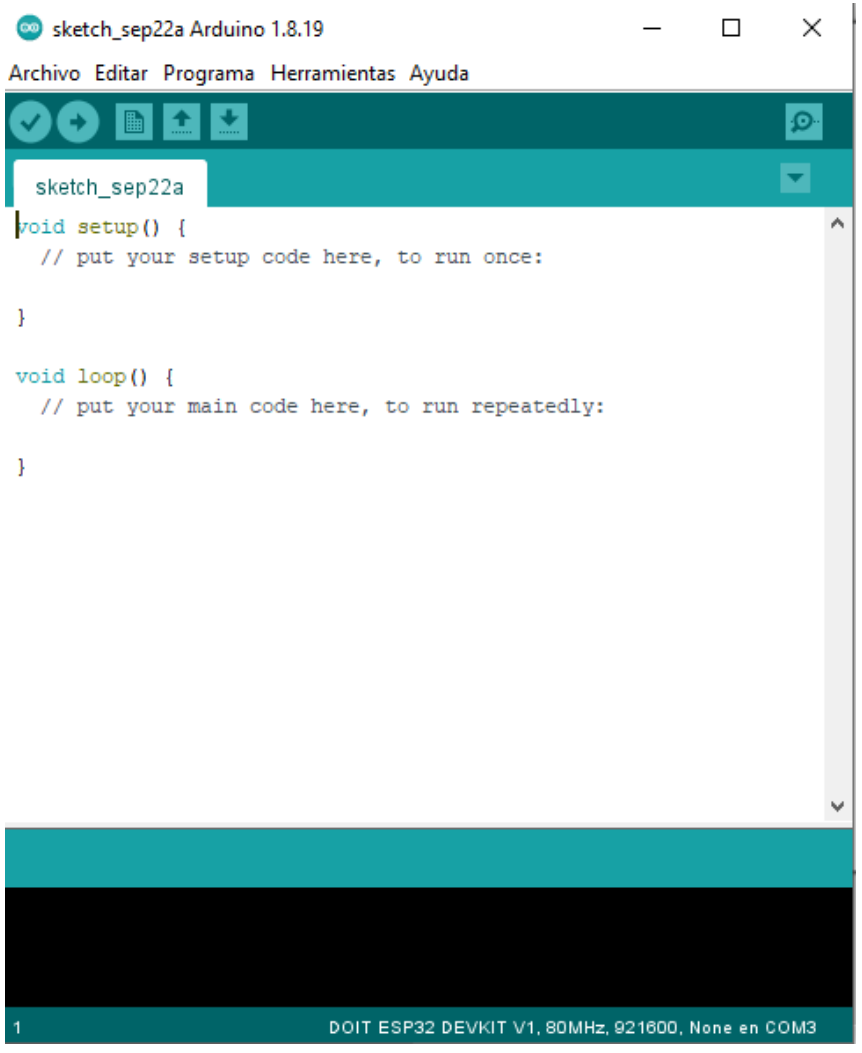
- + Empezar con teclado y terminar con teclado c/u

Programación con código



Instalación IDE Arduino

https://www.arduino.cc/en/software


sketch_sep22a Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
sketch_sep22a
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}
1
DOIT ESP32 DEVKIT V1, 80MHz, 921600, None en COM3

IDE Arduino

Bloques del IDE de Arduino



Bloque 0 – Comentarios (OPCIONAL)

Bloque 1 – Declaración de las variables que vamos a utilizar

Bloque 2 – Configuración inicial del programa

Bloque 3 – Bucle infinito que contiene el conjunto de instrucciones que se repiten constantemente

Primer programa en el IDE de Arduino

1)**Conecta la placa Arduino a tu computador** y ejecuta el IDE oficial.

2)Selecciona (si no lo está ya) el tipo de placa adecuado (en el menú Tools->Board) y

3)El puerto USB utilizado (en el menú Tools->Serial port).

4)Nuevo sketch

Reto 1

```
/*Declaración e inicialización de una
variable global llamada “mivariable” */
int mivariable=555;
void setup() {
Serial.begin(9600);
}
void loop() {
Serial.println(mivariable);
mivariable=mivariable+1;
}
```

Reto 2

- int numero=100;
- long resultado;
- void setup() {
- Serial.begin(9600);
- resultado=numero*1000;
- Serial.println(resultado);
- }
- void loop() {}

- Tiempo

Reto 3

```
unsigned long time;
void setup(){
Serial.begin(9600);
}
void loop(){
time = micros();
Serial.println(time);
delay(1000);
}
```

Acordeón Arduino

Tomado mayormente de la Referencia de Lenguaje Arduino:
http://arduino.cc/en/Reference/HomePage

Estructura y flujo

Estructura básica del programa

```
void setup() {
  // Corre una vez cuando el
  // programa inicia
}
void loop() {
  // Se ejecuta repetidamente
```

Estructuras de control

```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
do { ... } while ( x < 5);
for (int i = 0; i < 10; i++) { ... }
break;  //sale del bucle inmediatamente
continue; //va a la siguiente iteración
switch (miVariable) {
   case 1:
     ...
     break;
   case 2:
     ...
     break;
   default:
     ...
}
return x; // o "return;" para vacíos
```

Operadores

Operadores generales

```
=  (operador de asignación)
+  (adición)     -  (sustracción)
*  (multiplicación)
/  (división)    %  (módulo)
== (igual a)     != (desigual a)
<  (menor que)   >  (mayor que)
<= (igual o menor que)
>= (mayor o igual que)
&& (y)    || (ó)    ! (negación)
```

Operadores compuestos

```
++ (incremento)
-- (decremento)
+= (suma compuesta)
-= (resta compuesta)
*= (multiplicación compuesta)
/= (división compuesta)
&= (AND binario compuesto)
|= (OR binario compuesto)
```

Operadores a nivel de bit

```
& (AND binario)    | (OR binario)
^ (XOR binario)    ~ (NOT binario)
<< (desplazamiento a la izquierda)
>> (desplazamiento a la derecha)
```

Variables, vectores y datos

Tipos de datos

```
void          vacío
boolean       (0, 1, true, false)
char          (ej. 'a'  -128 a 127)
int           (-32768 a 32767)
long          (-2147483648 a 2147483647)
unsigned char (0 a 255)
byte          (0 a 255)
unsigned int  (0 a 65535)
word          (0 a 65535)
unsigned long (0 a 4294967295)
float         (-3.4028e+38 a 3.4028e+38)
double        (igual que los flotantes)
```

Calificadores

```
static   //persiste entre llamadas
volatile //usa la RAM
const    //sólo lectura
PROGMEM  //usar la flash
```

Vectores y matrices

```
int myInts[6]; //vector de 6 enteros
int myPins[]={2, 4, 8, 3, 6};
int mySensVals[6]={2, 4, -8, 3, 2};
myInts[0]=42;  //asigna al primero
               //en el índice
myInts[6]=12;  //ERROR! El índice va
               //de 0 a 5
```

Constants

```
HIGH | LOW
INPUT | OUTPUT
true | false
143          //Decimal
0173         //Octal (comenzando en 0)
0b11011111 //Binario
0x7B         //Hex (hexadecimal)
7U           //forzar unsigned
10L          //forzar long
15UL         //forzar long unsigned
10.0         //forzar floating point
2.4e5        //240000
```

Punteros

```
&  (referencia: obtener puntero)
*  (valor: seguir puntero)
```

Cadenas

```
char S1[8] =
 {'A','r','d','u','i','n','o'};
 //cadena sin terminación
 //puede producir error
char S2[8] =
 {'A','r','d','u','i','n','o','\0'};
 //incluye terminación nula \0
char S3[]="arduino";
char S4[8]="arduino";
```

Funciones incluídas

E/S Digital

```
pinMode(pin,[INPUT, OUTPUT])
digitalWrite(pin, valor)
int digitalread(pin)
  //Escribe HIGH en entradas para
  //usar los pull-ups
```

E/S Analógicas

```
analogReference([DEFAULT,
    INTERNAL, EXTERNAL])
int analogRead(pin)
analogWrite(pin, valor)  //PWM
```

Advanced I/O

```
tone(pin, freqhz)
tone(pin, freqhz, duracion_ms)
noTone(pin)
shiftOut(pinDatos, pinReloj,
  [MSBFIRST,LSBFIRST], valor)
unsigned long pulseIn(pin,
  [HIGH,LOW])
```

Tiempo

```
unsigned long millis()
  //desbordamiento en 50 días
unsigned long micros()
  //desbordamiento en 70 minutos
delay(ms)
delayMicroseconds(us)
```

Matemáticas

```
min(x, y)    max(x, y)    abs(x)
sin(rad)     cos(rad)     tan(rad)
sqrt(x)      pow(base, exponente)
constrain(x, valMin, valMax)
map(val, deBAJO, deALTO,
aBAJO,aAlto)
```

Números aleatorios

```
randomSeed(semilla) //long ó int
long random(max)
long random(min, max)
```

Bits y Bytes

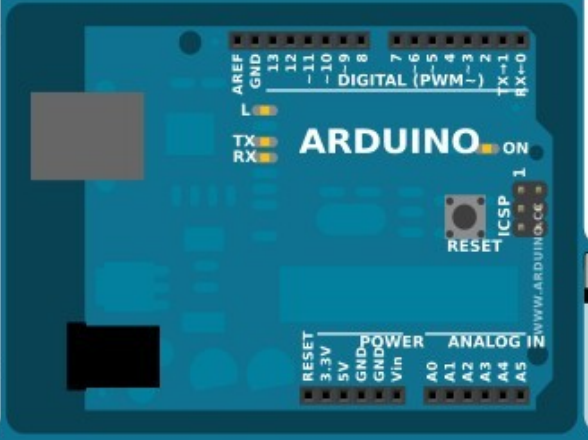
```
lowByte(x)    highByte(x)
bitRead(x, bitn)
bitWrite(x, bitn, bit)
bitSet(x, bitn)
bitClear(x, bitn)
bit(bitn)  // bitn: 0=LSB 7=MSB
```

Conversiones

```
char()        byte()
int()         word()
long()        float()
```

Interrupciones Externas

```
attachInterrupt(interrup, func,
 [LOW, CHANGE, RISING, FALLING])
detachInterrupt(interrupción)
interrupts()
noInterrupts()
```



Bibliotecas

Serie

```
begin([300, 1200, 2400, 4800,
       9600, 14400, 19200, 28800,
       38400, 57600, 115200])
     //Puede ser cualquier número
end()
int available()
byte read()
byte peek()
flush()
print(misDatos)
println(misDatos)
write(misBytes)
```

EEPROM (#include <EEPROM.h>)

```
byte read(dirInterna)
write(dirInterna, miByte)
```

Servo (#include <Servo.h>)

```
attach(pin, [min_uS, max_uS])
write(ángulo) // 0, 180
writeMicroseconds(uS)
     //1000-2000; 1500 es en medio
read()       //0 - 180
attached()   //regresa boleano
detach()
```

SoftwareSerial(RxPin, TxPin)
(#include <softwareSerial.h>)

```
begin(long velocidad) //hasta 9600
char read()     //espera los datos
print(misDatos)
println(misDatos)
```

Wire (#include <Wire.h>) //para I²C

```
begin()        //se une a maestro
begin(addr)    //se une a esclavo @dir
requestFrom(dirección, cuenta)
beginTransmission(dir)  // Paso 1
send(miByte)            // Paso 2
send(char * miCadena)
send(byte * datos, tamaño)
endTransmission()       // Paso 3
byte available()   // Num de bytes
byte receive() //Regresa el sig byte
onReceive(manejador)
onRequest(manejador)
```


CC BY SA

por Mark Liffiton

Traducción de Antonio Maldonado

Adaptado de:
- Idea original por Gavin Smith
- Versión SVG por Frederic Dufourg
- Dibujo del Arduino de Fritzing.org

retroalimentación

¿Qué aprendí? 	¿Qué desaprendí? 
Lo que mas te gusto - aspectos positivos 	Lo que se podría mejorar 
Cosas que no entendieron 	Nuevas ideas y conclusiones 

Bibliografía

- Robótica básica https://www.slideshare.net/jegcgarcas/taller-de-robotica-basica

- Tecnologías 4.0 para niños y no tan niños
- https://www.slideshare.net/jegcgarcas/tecnologas-40-para-nios-y-no-tan-nios