



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA DE  
VALPARAÍSO

[pucv.cl](http://pucv.cl)

# Robótica e inteligencia artificial

## Módulo 3

### Programación y lógica de funcionamiento S14

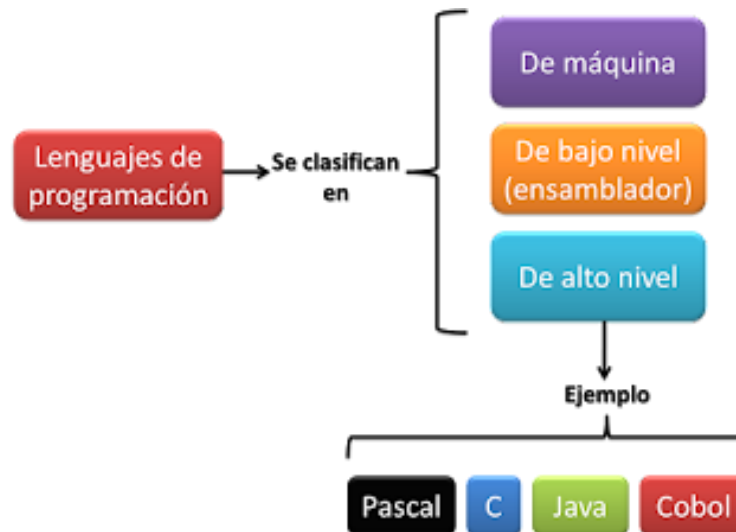
# PROGRAMACIÓN Y LÓGICA DE FUNCIONAMIENTO SESIÓN 14

# Programación y lógica de funcionamiento

## Introducción

Se ha mencionado anteriormente que la comunicación con el robot está fuertemente asociada al nivel de elaboración del robot, dónde se puede identificar comunicaciones altamente elaboradas entre el usuario humano y la máquina robótica, por ejemplo, a través de software con interfaces de operación y programación gráfica como lo es el “Teach Pendant Programming”, entre otros métodos. Luego, hablando de una comunicación menos elaborada con la máquina robótica, se puede encontrar lenguajes que se comunican casi directamente con la unidad de procesos centrales del robot, dando órdenes y cálculos al detalle necesarios para la ejecución del propósito de la máquina.

En cualquiera de estos casos los algoritmos del robot deben funcionar de modo que den cumplimiento al objetivo de éste.



# Programación y lógica de funcionamiento

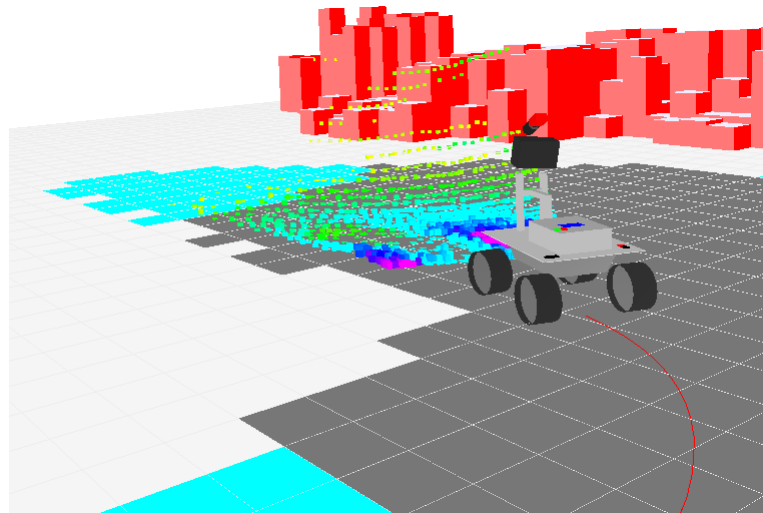
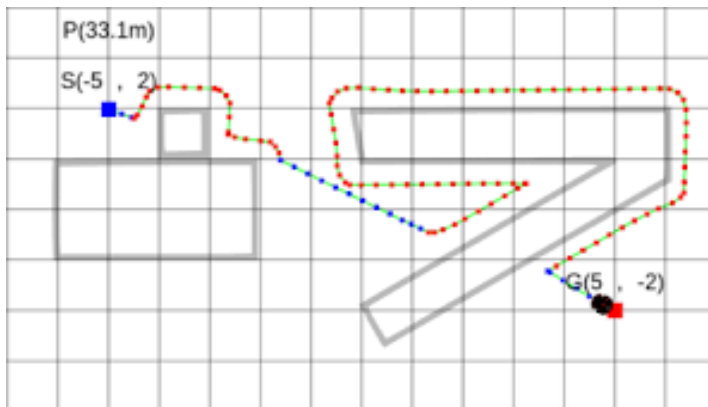
## Introducción

En este módulo de aprendizaje se Abordará el estudio de la capacidad de autonomía de un robot en torno a los procesos centrales y decisiones que este puede tomar para llevar a cabo su cometido.

Se estudiarán diferentes plataformas de desarrollo utilizadas recurrentemente en la robótica, además de sus ventajas y desventajas.

Se estudiará el detalle el funcionamiento de actuadores y sensores en relación con una CPU.

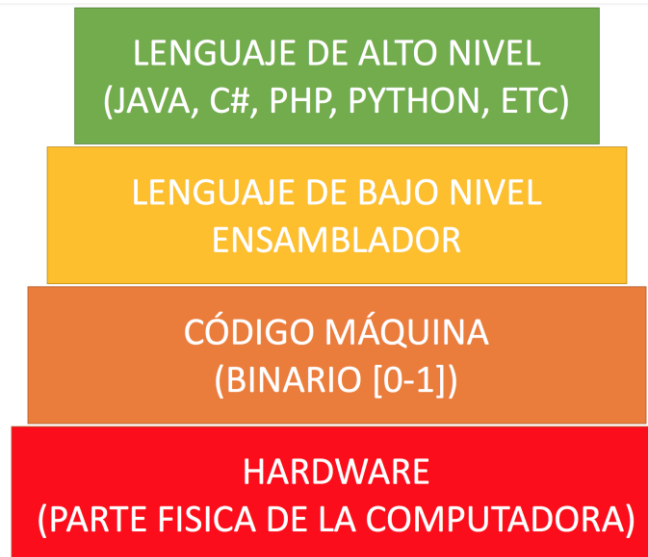
Finalmente, se estudiarán aplicaciones comunes de programación enfocadas a la percepción y navegación de un robot.



# Programación y lógica de funcionamiento

## Niveles de lenguajes de programación

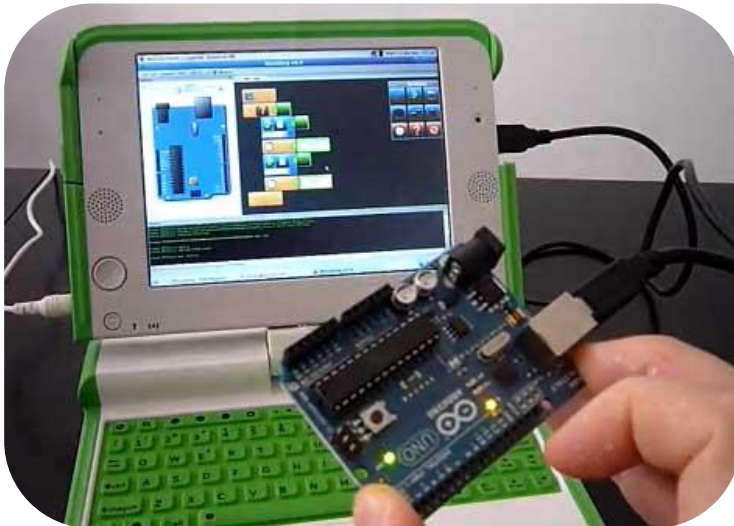
Existen lenguajes de bajo nivel los cuales se comunican casi directamente con los registros de un procesador, mientras que otros lenguajes de alto nivel poseen mayor cantidad de capas de comunicación entre software o algoritmos de un computador para llegar a comunicarse con la máquina.



En la robótica es posible encontrar distintos niveles de comunicación con la máquina, siempre dependiendo de la aplicación final de esta.

# Programación y lógica de funcionamiento

## Niveles de lenguajes de programación



Programación de menor nivel



Programación de mayor nivel

Al usar simuladores se identifica que no hay comunicación directa con una máquina real, y que el lenguaje de programación utilizado en el simulador es comúnmente de alto nivel, ya que no está enfocado en dar funcionamiento a la unidad de procesos centrales de la máquina robótica en cuestión.

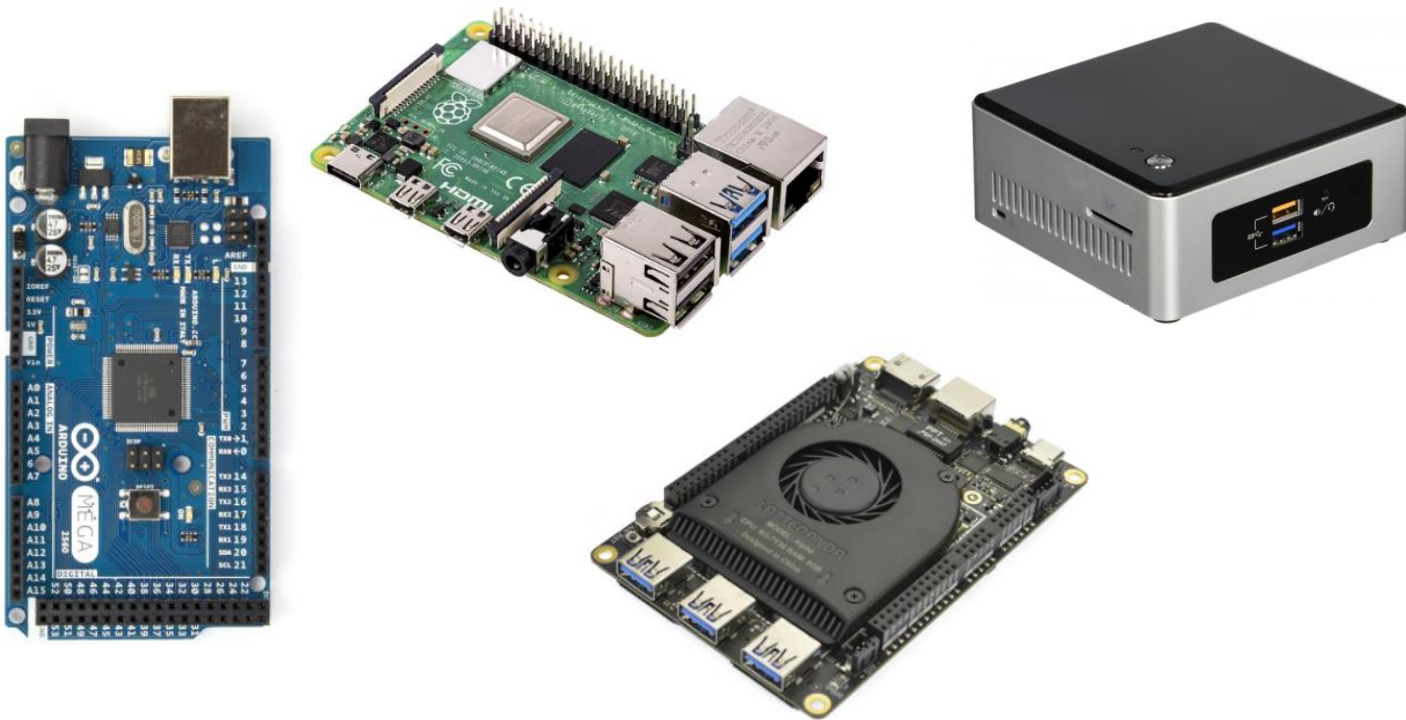
El código del simulador no es el mismo que el que se carga al robot



# Programación y lógica de funcionamiento

## Niveles de lenguajes de programación

En la aplicación real de la robótica educativa o de investigación, la comunicación existente entre el programador y el robot es comúnmente a través de un software compilador instalado en un computador. Luego, el nivel del lenguaje de programación estará directamente relacionado con la plataforma de desarrollo que el usuario haya escogido, identificando que existen gran cantidad de plataformas de desarrollo útiles para la robótica.





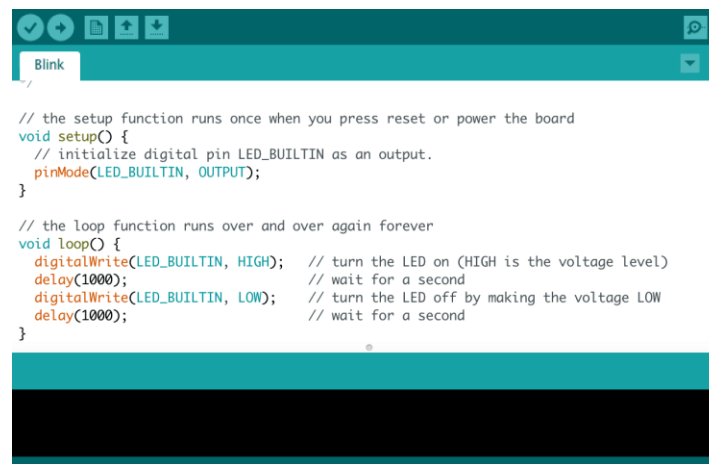
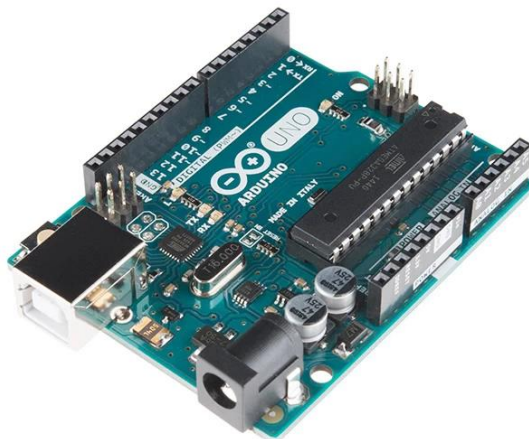
# Plataformas de desarrollo (Arduino)

Para la robótica e investigación en conjunto con una gran cantidad de aplicaciones electrónicas, se identifica el Arduino como una popular plataforma de desarrollo.

Se les llama tarjeta Arduino a un board que cuenta con un microcontrolador como unidad de procesos centrales y que cuenta con un diseño enfocado en facilitar la conexión de comunicación entre el computador programador y la CPU, además de la etapa de potencia o energía, y una disposición de conexión rápida de los pines del microcontrolador.

Arduino también ha facilitado un software programador o entorno de desarrollo (IDE) que utiliza el lenguaje de programación Derivado del C++.

Con ayuda del puerto de comunicación USB de la placa y el software programador, se considera sencillo programar una rutina básica en esta plataforma de desarrollo.

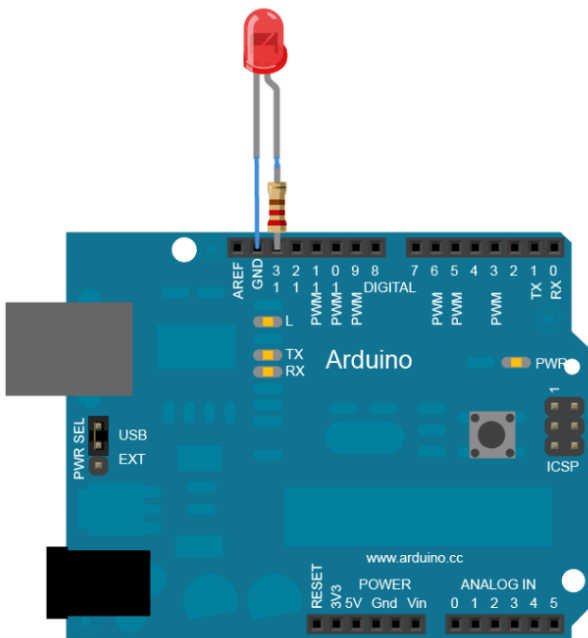


```
// Blink
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

# Plataformas de desarrollo (Arduino)

Con esta plataforma fácilmente se puede construir un circuito y programar la rutina deseada. En el siguiente ejemplo se muestra una construcción de circuito básica y una programación denominada “blink”.



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

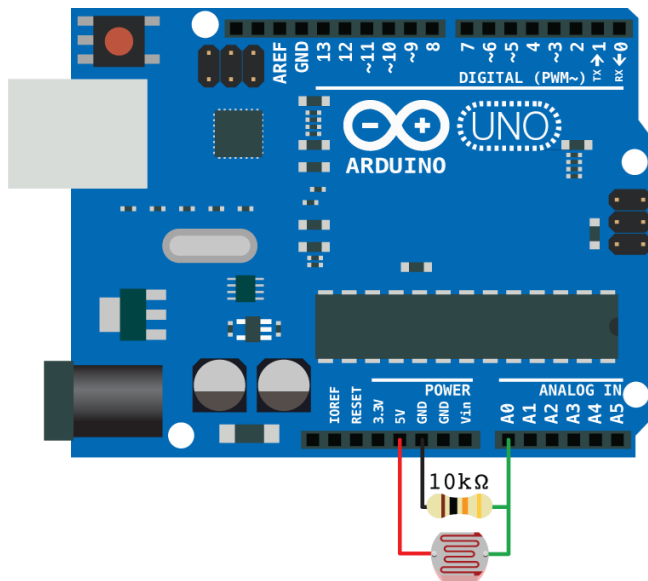
Ya que el modelo Arduino UNO cuenta con 14 salidas digitales, Para este ejercicio es posible conectar mayor cantidad de leds y programar la secuencia con que estos serán encendidos o apagados.

# Plataformas de desarrollo (Arduino)

Arduino también cuenta con entradas de información llamadas puertos análogos, ya que cuentan con un circuito ADC que permite tomar información proveniente de distintos sensores.

Algunos sensores sencillos como lo son los LDR, sensores de luz resistivos, permiten tomar una medición de la luz ambiental presente En el circuito, para luego programar una rutina que relacione esta medición.

Es importante considerar que los sensores de funcionamiento básico no entregan información interpretada. es trabajo del programador realizar una interpretación de los valores medidos y asociarles una unidad de medida en caso de ser necesario.

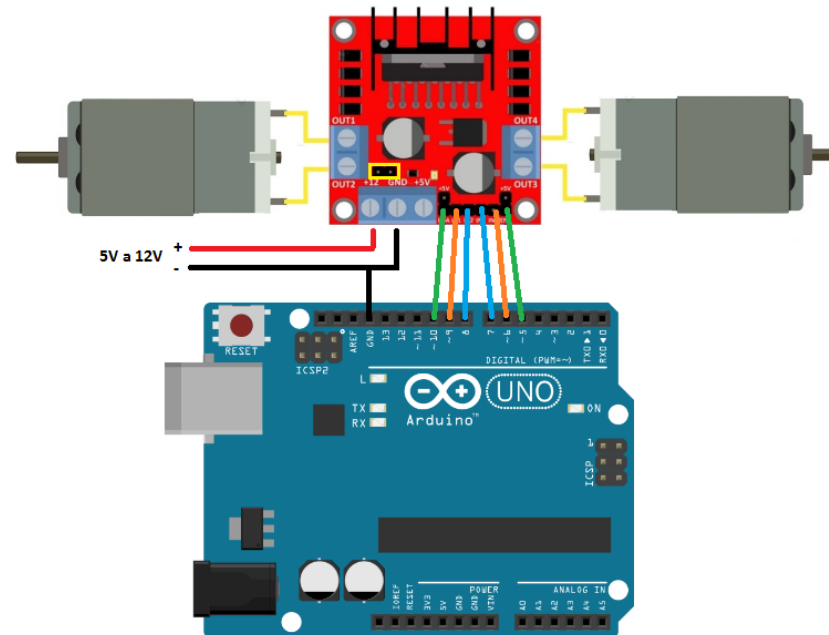


```
void loop() {  
  
    int ldrStatus = analogRead(ldrPin);  
  
    if (ldrStatus <= 400) {  
        digitalWrite(ledPin, HIGH);  
        Serial.print("Its DARK, Turn on the LED : ");  
        Serial.println(ldrStatus);  
    } else {  
        digitalWrite(ledPin, LOW);  
        Serial.print("Its BRIGHT, Turn off the LED : ");  
        Serial.println(ldrStatus);  
    }  
}
```

\*Código incompleto

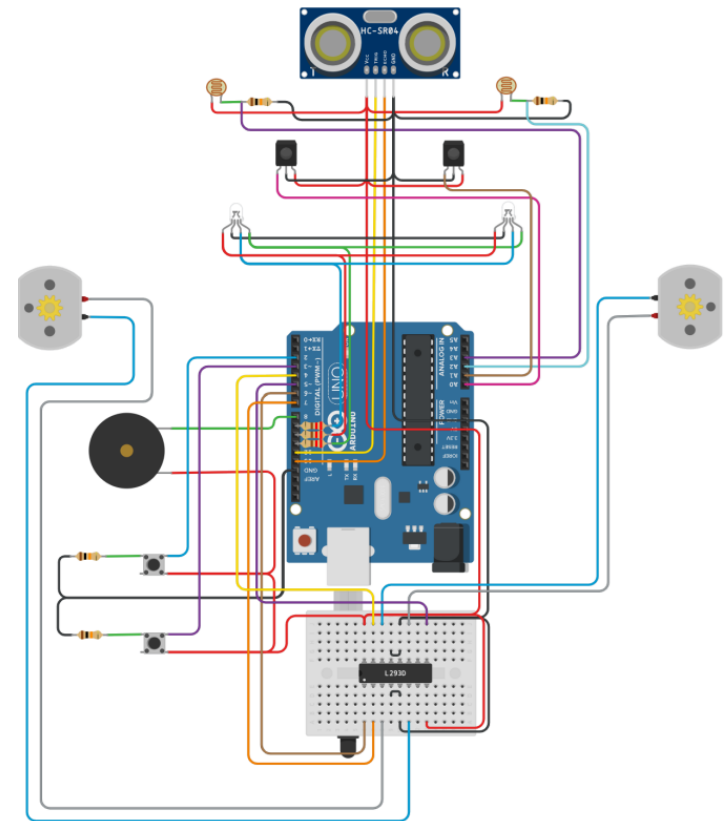
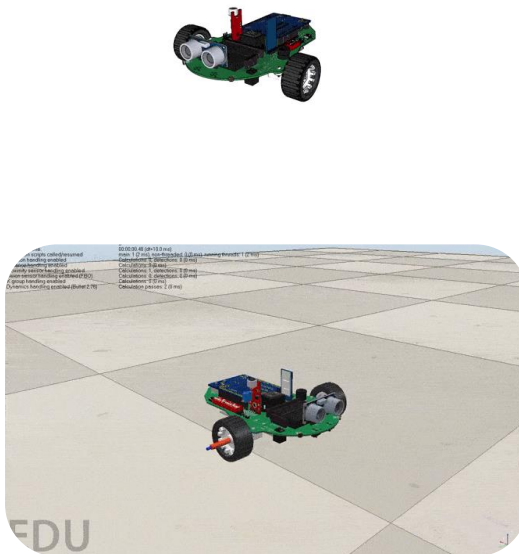
# Plataformas de desarrollo (Arduino)

Como se ha mencionado, la plataforma Arduino también puede ser utilizada para la robótica básica, implementando los principios de funcionamiento de un robot, tales como señales de entrada, procesamiento de información y ejecución de acciones.



# Plataformas de desarrollo (Arduino)

En CoppeliaSim podemos encontrar robots con un Arduino como CPU, pero el lenguaje de programación de este software no se enfoca en la programación de la plataforma de desarrollo, sino más bien en la ejecución de los movimientos y la acción de los actuadores proveídos por el software.



# Plataformas de desarrollo (Raspberry)

Otra plataforma muy utilizada la robótica educativa y de investigación es la plataforma Raspberry Pi, la cual es mucho más elaborada que un Arduino, ya que está puede soportar un sistema operativo Y por ende soportar otros lenguajes de programación.

Esta plataforma de desarrollo también cuenta con unidad de procesos gráficos que puede ayudar en el procesamiento de información proveniente de cámaras, logrando implementar incluso sistemas de inteligencia artificial en ella.

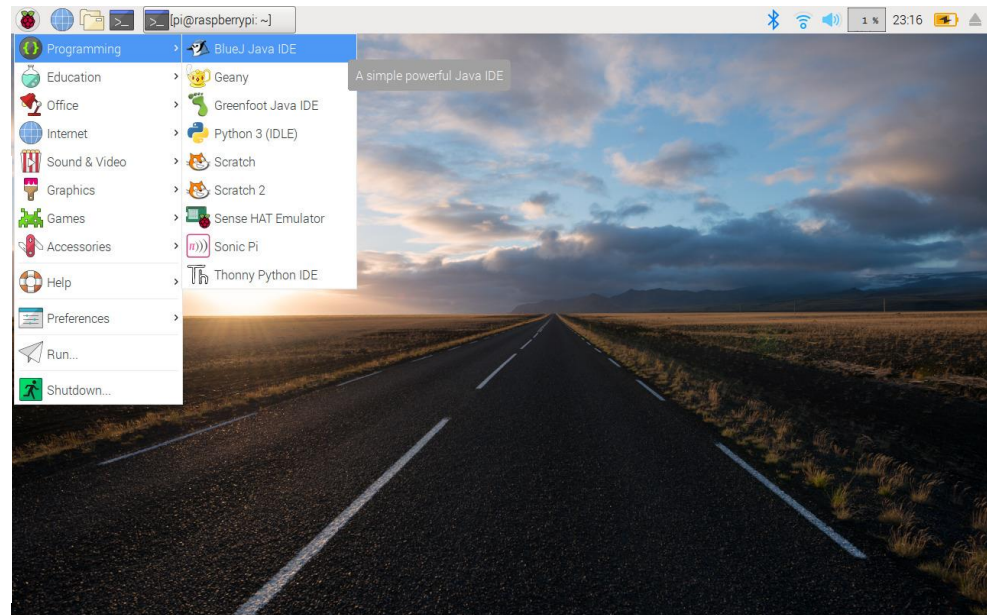
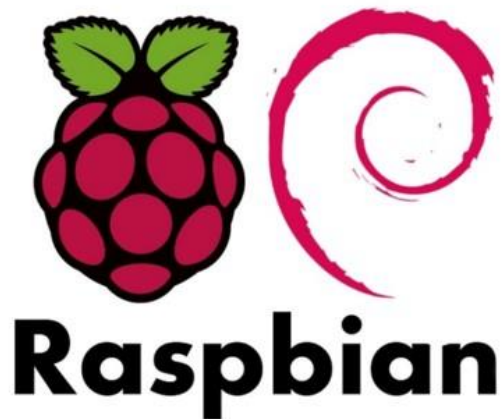




# Plataformas de desarrollo (Raspberry)

El principio de funcionamiento de las raspberry es muy parecido al de Arduino, disponiendo de ciertas conexiones tipo pinheader para que el usuario pueda conectar distintos tipos de sensores o actuadores a esta plataforma.

Ya que raspberry soporta un sistema operativo y cuenta con la posibilidad de conectar está a un monitor, no se consideran software intermedios para que sea programada.



# Software de simulación

La robótica también es utilizado ampliamente las capacidades de los software de simulación, ya que presentan ventajas cómo la posibilidad de diseñar modelos, probarlos y optimizarlos, conocer el comportamiento de los mismos antes de su puesta en marcha, rediseñar procesos fuera de línea de fabricación y anticipar errores mejorando la calidad y la precisión.

Al igual que los robots, los software de simulación también cuentan con determinados objetivos, desarrollando ciertas áreas De especificación es posible encontrar software enfocados en la programación, o en el armado de robots, o en el cálculo dinámico del movimiento, entre otras cosas.



CoppeliaSim  
from the creators of V-REP



ISAAC

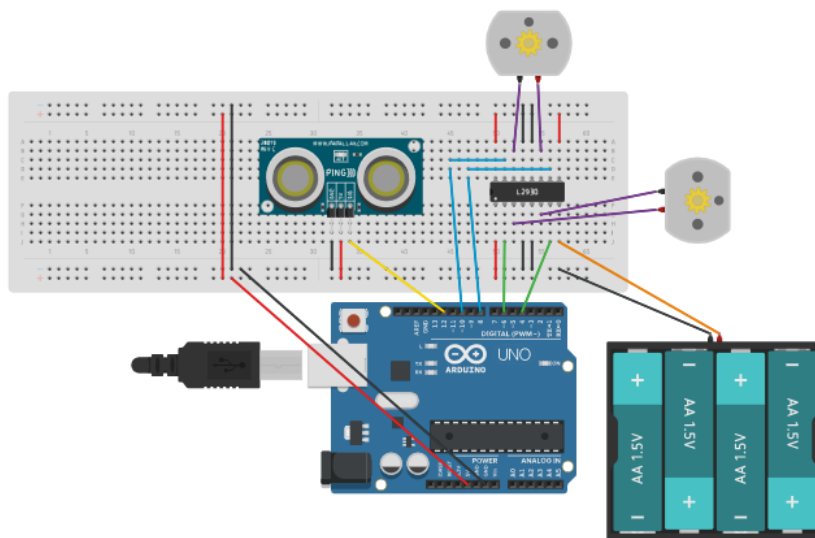


# Software de simulación (Tinkercad)

El software tinker CAD perteneciente a la empresa autodesk, es bastante sencillo y el público objetivo son jóvenes mayores a 10 años. tinker CAD posee un apartado de simulación de circuitos electrónicos, permitiendo programar un Arduino UNO con lenguaje de programación gráfico o directamente escribiendo el código.

Para la robótica educativa este simulador es una buena herramienta, ya que permite tomar el código generado y subirlo a Arduino a través de su IDE.

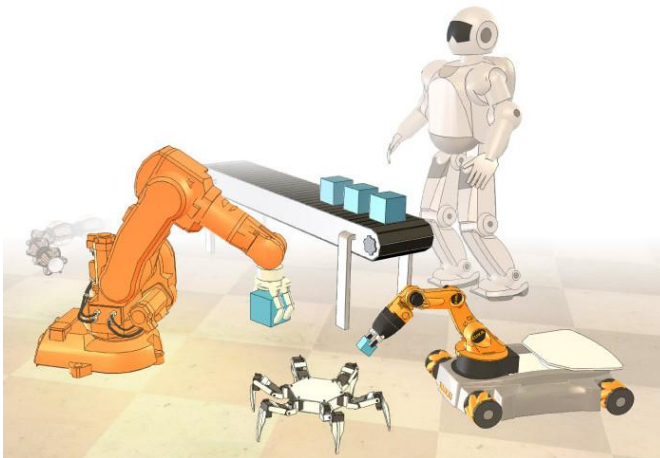
El simulador se enfoca en el funcionamiento del circuito y no en el movimiento de los actuadores, por lo que presenta ventajas y desventajas en su uso.



# Software de simulación (CoppeliaSim)

CoppeliaSim Es un software de simulación de máquinas robóticas que se enfoca en el cálculo cinemático de las variables físicas que afectan a un cuerpo. Específicamente es posible encontrar gran cantidad de configuraciones asociadas a parámetros físico mecánico como torques, velocidades, índices de fricción, fuerzas, entre muchos otros.

La programación de este software se enfoca en el funcionamiento de los actuadores y sensores que él mismo provee, por lo que el código generado no servirá para implementarlo en una máquina robótica real, pero permitirá probar algoritmos que se encuentren relacionados con el funcionamiento físico mecánico del robot en cuestión.



CoppeliaSim  
from the creators of V-REP

# Software de simulación (Nvidia ISAAC)

La plataforma de simulación NVIDIA ISAAC es una herramienta moderna para La simulación y el desarrollo de la robótica industrial y comercial. Permitiendo abordar desafíos de robótica con soluciones integrales para reducir los costos, simplificar el desarrollo y acelerar el tiempo de comercialización.

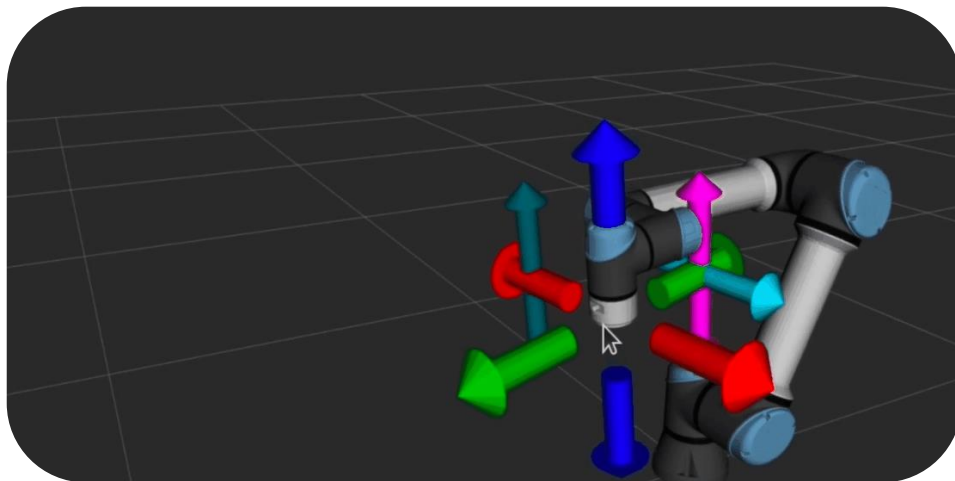
NVIDIA Busca reducir las barreras para integrar la inteligencia artificial en máquinas autónomas, por lo que el simulador se encuentra habilitado con bibliotecas y algoritmos para GPU, Utilizados ampliamente en la implementación de IA.



# Herramientas de programación (ROS)

El sistema operativo de robot (ROS) es un conjunto de bibliotecas de software y herramientas que ayudan a crear aplicaciones de robótica. Desde controladores hasta algoritmos de última generación y potentes herramientas de desarrollo.

ROS Es un software código abierto que posee interfaces, herramientas y componentes útiles para aplicaciones robóticas. Se considera una de las herramientas más completas para la robótica industrial y es altamente utilizada.



 ROS