

*Programa Educativo - Educación Primaria*

# Introducción a la Programación, Diseño e Impresión 3D y Robótica

30 Julio, 2015

Iván García Sainz-Aja <ivangsa@gmail.com>

<http://www.unir.net/ingenieria/curso-electronica-robotica-impresion-3d/549200191864/>



Esta obra de Iván García Sainz-Aja se comparte bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

## Tabla de Contenidos

### Tabla de Contenidos

#### Justificación

Conocimiento Previo

Intereses y Motivaciones

Competencias y estrategias de aprendizaje

#### Contextualización

Número de alumnos

Características del grupo

Espacio y tiempo disponible

Materiales y recursos disponibles

#### Objetivos de aprendizaje

#### Planificación

##### Contenidos

Introducción a la Programación por Bloques

Introducción al Dibujo 2D y Diseño 3D

Introducción a la Robótica con Arduino

##### Temporalización

##### Actividades

##### Recursos

#### Creación y desarrollo de las actividades (organización por sesiones)

##### Programación

Introducción a la programación por bloques. Blockly y Scratch.

Bucles y Condicionales. Blockly y Scratch.

Dibujando formas geométricas. Blockly y Scratch.

Formas Geométricas: Aplicando la Abstracción. Scratch.

Sistema de Coordenadas Cartesiano. Posición y Movimiento. Scratch.

Variables. Scratch.

Variables, Bucles y Condicionales. Scratch.

Estrategias para juegos I. Scratch.

Estrategias para juegos II. Scratch.

##### Dibujo 2D y Diseño 3D

Dibujo 2D: Introducción a Processing JS - Líneas y Polígonos

Dibujo 2D: Stroke y Colores. Actividad: Dibujamos una Casa

Dibujo 2D: Control Flujo y Variables en ProcessingJS

Diseño 3D: OpenSCAD Polígonos, Rotacion y Traslacion. Actividad: Muñeco Nieve.

[Diseño 3D: OpenSCAD Booleanas: Union / Diferencia / Interc. Actividad: Ficha Dominó.](#)

[Diseño 3D: OpenSCAD Booleanas: Union / Diferencia / Interc. Actividad: Dado Parchis.](#)

[Diseño 3D: Introducción FreeCAD/TinkerCAD. Entornos de diseño 3D gráficos.](#)

[Diseño 3D: Introducción FreeCAD/TinkerCAD. Entornos de diseño 3D gráficos.](#)  
[impresión: Visita a una copistería 3D e imprimir uno de nuestros diseños.](#)

#### Robótica

[Introducción a Arduino: Pines, Entrada y Salida. Bitbloq.](#)

[Parpadeo Leds. Actividad: Programación Semáforo](#)

[Sensor Luz. Actividad: Farola Automática](#)

[Servo Rotación Continua. Actividad: Coche "Solar"](#)

[Construcción Sensor Flexibilidad: Aluminio / Celo / Carboncillo / Cartón](#)

[Servo Direccional: Actividad: Paso a Nivel](#)

[Robótica: Programando un Robot Sigue líneas](#)

[Robótica: Programando un Robot Sigue/Huye Luz](#)

[Detrás de la cortina: Código detrás de los bloques.](#)

#### Evaluación de todo el programa

[Análisis de Fortalezas/Oportunidades/Debilidades/Amenazas](#)

[Fortalezas \(Internas\)](#)

[Oportunidades \(Externas\)](#)

[Debilidades \(Internas\)](#)

[Amenazas \(Externas\)](#)

## Justificación

Se trata de una actividad extraescolar de introducción a la programación, al diseño e impresión 3D y a la electo-robótica casera con arduino con el objetivo de acercar la tecnología a los alumnos a partir de 8-10 años, como creadores de soluciones tecnológicas, poniendo una semilla de curiosidad, alimentando la creatividad y la alegría por aprender.

Durante esta actividad aprenderemos a *pensar como un ordenador*, organizando nuestros pensamientos como una receta ordenada de instrucciones (programar), a diseñar y crear mediante el ordenador *objetos del mundo físico* (diseño e impresión 3D) y a *darles vida* mediante placas microcontroladoras, sensores y actuadores (electro-robótica).

Tras esta actividad los alumnos serán capaces de comprender mejor en que consiste programar un ordenador, de crear sus primeros videojuegos utilizando lenguajes de programación por bloques, de diseñar e imprimir algún objeto sencillo como pueda ser una ficha de dominó, un dado, un llavero con su nombre.. y de conectar el mundo virtual de los ordenadores con el mundo físico a través de sensores, motores actuadores y placas electrónicas microcontroladoras.

Con esta actividad queremos poner nuestro granito de arena en convertir a una generación de usuarios avanzados de tecnología en una generación de creadores de soluciones tecnológicas, disipando la visión equivocada de la tecnología como una especie de *magia de caja negra* y abriendo un mundo de posibilidades *a otro tipo de magia*: la magia de la curiosidad, la creatividad y la creación.

## Conocimiento Previo

Para participar en la actividad no se requiere conocimiento previo de informática más allá del manejo básico de un navegador web, el ratón y el teclado.

## Intereses y Motivaciones

Curiosidad e interés por la tecnología, puzzles, juegos de lógica, juegos de construcción, manualidades, videojuegos..

## Competencias y estrategias de aprendizaje

Se desarrolla según una estrategia de aprendizaje constructivista basada en resolver problemas y proyectos con elementos de enseñanza virtual de Flipped Classroom, con materiales audiovisuales grabados y disponibles a través del aula virtual.

## Contextualización

Se trata de una actividad extraescolar de carácter anual con sesiones semanales de 2 horas de duración a realizar en el propio centro educativo en el aula de informática.

### Número de alumnos

La actividad se desarrollará en grupos de entre 10 y 20 alumnos.

### Características del grupo

Los grupos se esperan que sean relativamente homogéneos en cuanto a los conocimientos y motivaciones con la principal diferencia entre alumnos pueda pequeñas diferencias de edad.

### Espacio y tiempo disponible

La actividad se desarrollará en sesiones semanales de 2 horas a lo largo de un curso académico en el aula de informática del propio centro.

### Materiales y recursos disponibles

En cuanto a los materiales será necesario disponer de un ordenador por cada dos alumno, con conexión a internet, y una pantalla proyector o pizarra digital en la que poder visualizar contenidos compartidos.

Para la actividad de diseño e impresión 3D visitaremos una tienda local en la que imprimimos alguno de nuestros diseños.

Para la actividad de robótica necesitaremos un conjunto de elementos: placa Arduino por cada dos alumnos, surtido de componentes electrónicos, cables, tuercas, tornillos, destornilladores... así como otros elementos de manualidades como goma eva, cartón, cartulina, tijeras, pegamento..

## Objetivos de aprendizaje

1. Programación y pensamiento computacional
  - a. Lenguajes de programación por bloques
  - b. Bucles y condicionales
  - c. Variables
  - d. Sistema de coordenadas cartesiano
  - e. Eventos, estados y comunicación entre componentes
2. Comprensión geoespacial en 2D y 3D
  - a. Dibujo de formas geométricas básicas en 2D
  - b. Diseño en 3D combinando formas geométricas básicas y operadores
  - c. impresión 3D: conocer las posibilidades de la impresión en 3D e imprimir alguno de nuestros diseños en una impresora 3D.
3. robótica
  - a. Conocer que es una placa microcontroladora
  - b. Sensores analógicos y digitales
  - c. Actuadores y servomotores
  - d. Uso en la vida diaria de elementos electrónicos y/o de robótica.
4. Comunicación y cooperación
  - a. Trabajo en equipo
  - b. Resolución de proyectos de forma colaborativa
  - c. Introducción al aprendizaje virtualizado

## Planificación

### Contenidos

Durante esta actividad explicaremos los siguientes contenidos:

#### Introducción a la Programación por Bloques

Con Scratch **aprenderemos a pensar como un ordenador**, programando usando bloques ensamblables tipo puzzle que constituyen las instrucciones del programa, para construir videojuegos o contar historias interactivas..

De esta manera **aprenderemos a organizar los pensamientos como una receta de instrucciones concretas y ordenadas, que utilizaremos para comunicarnos con el cerebro de nuestro ordenador**, aprendiendo a resolver problemas complejos dividiéndolos en otros más pequeños y sencillos y a colaborar con nuestros compañeros para ser capaces de resolver problemas cada vez mayores.

Y es que la programación como recurso educativo fomenta y ayuda al desarrollo del pensamiento computacional, el pensamiento estructurado, nos ayuda a ver los sistemas complejos en partes simples sin olvidarnos del sistema en sí. A la vez que se alenta la persistencia, al trabajar con problemas difíciles, la creatividad, la comunicación y el trabajo cooperativo.

Exploraremos los principios básicos de la programación imperativa utilizando bloques ensamblables: resolviendo puzzles con Blockly y programando nuestros primeros videojuegos con Scratch. Aprenderemos las claves prácticas que hacen posible la programación de ordenadores así como sus estructuras básicas: instrucciones, bucles, condicionales y variables.

#### Introducción al Dibujo 2D y Diseño 3D

**Aprenderemos a usar el ordenador para diseñar y crear objetos del mundo físico.** El diseño y la impresión 3D está pasando rápidamente de ser el futuro a ser el presente y diseñar e imprimir un objeto en tres dimensiones ayuda a una mejor comprensión geo espacial.

En este taller introduciremos al alumno al dibujo de formas primitivas (líneas, polígonos, círculos) en 2 dimensiones para pasar luego al diseño espacial en 3 dimensiones, utilizando una impresora 3D para imprimir algunos de nuestros diseños en una impresora 3D. Las posibilidades que nos ofrece esta tecnología son múltiples y en muchos casos estas posibilidades están todavía sin explorar.

Con las herramientas ProcessingJS y OpenSCAD aprenderemos a dibujar formas en 2 y 3 dimensiones utilizando funciones geométricas escribiendo para ello nuestro propio código. Utilizaremos combinaciones de polígonos y líneas para realizar dibujos en 2D y 3D. Visitaremos una copistería local en la que podremos crear una versión física de nuestros diseños digitales, convirtiendo el ordenador en un instrumento para generar objetos del mundo físico.

### Introducción a la Robótica con Arduino

*Instruir a niños sobre el mundo elaborado por seres humanos, es decir, el mundo de la tecnología y la ingeniería, es tan importante como instruirlos sobre el mundo natural, los números y las letras.*

- Marina U. Bers de la Universidad Tufts / Programa TANGIBLE K

Con Bitbloq y Arduino **aprenderemos a “programar cosas”** utilizando bloques ensamblables similares a los usados en Scratch soluciones ingeniosas aderezadas con un toque de manualidades a problemas del día a día como: programar las luces de un semáforo, sensores de luz, sensores de contacto, alarmas, robots sigue líneas, grúas.. Con la aparición y popularización de componentes electrónicos de bajo coste como las placas microcontroladoras Arduino y kits *hazlo-tu-mismo* cada día es más sencillo y accesible adentrarse en el mundo de la electrónica y la robótica casera.

## **Temporalización**

La actividad anual está dividida en 3 bloques de aproximadamente un trimestre cada uno. Cada uno de los bloques contará con un número de actividades estructuradas, aproximadamente entre 8 o 9 dejando espacio al final de cada trimestre de algunas sesiones para la improvisación y la exploración creativa combinando los conocimientos adquiridos durante el bloque y de las preferencias e intereses mostradas por los alumnos.



## Actividades

<b>Programación</b>	
	Introducción a la programación por bloques. Blockly y Scratch.
	Bucles y Condicionales. Blockly y Scratch.
	Dibujando formas geométricas. Blockly y Scratch.
	Formas Geométricas: Aplicando la Abstracción. Scratch.
	Sistema de Coordenadas Cartesiano. Posicion y Movimiento. Scratch.
	Variables. Scratch.
	Variables, Bucles y Condicionales. Scratch.
	Estrategias para juegos I. Scratch.
	Estrategias para juegos II. Scratch.

<b>Dibujo 2D y Diseño 3D</b>	
	Dibujo 2D: Introducción a Processing JS - Líneas y Polígonos
	Dibujo 2D: Stroke y Colores. Actividad: Dibujamos una Casa
	Dibujo 2D: Control Flujo y Variables en ProcessingJS
	Diseño 3D: OpenSCAD Polígonos, Rotacion y Traslacion. Actividad: Muñeco Nieve.
	Diseño 3D: OpenSCAD Booleanas: Union / Diferencia / Interc. Actividad: Ficha Dominó.
	Diseño 3D: OpenSCAD Booleanas: Union / Diferencia / Interc. Actividad: Dado Parchis.
	Diseño 3D: Introducción FreeCAD/TinkerCAD. Entornos de diseño 3D graficos.
	Diseño 3D: Introducción FreeCAD/TinkerCAD. Entornos de diseño 3D graficos.

	impresión: Visita a una copistería 3D e imprimir uno de nuestros diseños.
--	---

<b>Robótica con Arduino</b>	
	Introducción a Arduino: Pines, Entrada y Salida. Bitbloq.
	Parpadeo Leds. Actividad: Programación Semáforo
	Sensor Luz. Actividad: Farola Automatica
	Servo Rotacion Continua. Actividad: Coche "Solar"
	Construcción Sensor Flexibilidad: Aluminio / Celo / Carboncillo / Cartón
	Servo Direccional: Actividad: Paso a Nivel
	Robótica: Programando un Robot Sigue lineas
	Robótica: Programando un Robot Sigue/Huye Luz
	Detrás de la cortina: Código detrás de los bloques.

## Recursos

Para el desarrollo de la actividad serán necesarios los siguientes recursos:

- Aula de informatica:
  - Un ordenador por cada dos alumnos
    - Software: Navegador web moderno con los plugins habituales (Java, Flash)
    - Software: OpenSCAD programa de software libre disponible para distintos sistemas operativos.
  - Pizarra digital o proyector
  - Conexion a internet
- Componentes Electro-robótica
  - Placa controladora Arduino, preferiblemente Freaduino o Zum, por cada dos alumnos

- Surtido de componentes electrónicos, bread boards, cables usb, tuercas, tornillos, llaves allen, destornilladores..
- Materiales para manualidades: carton, cartulina, goma eva, tijeras, pegamento...
- Plataforma Virtual
  - Moodle: plataforma virtual para la centralización de los contenidos virtuales como videoclases, cuestionarios, tarjetas de referencia, glosario...
- Impresora 3D: no es necesario disponer de una impresora 3D en el propio aula. Podremos visitar alguna copistería local en la que imprimir algunos de nuestros diseños.

## Creación y desarrollo de las actividades (organización por sesiones)

### **Programación**

## Introducción a la programación por bloques. Blockly y Scratch.

### Objetivos trabajados:

- Introducción a la programación por bloques
- Ensamblar bloques de instrucciones
- Introducción informal a los bucles y condicionales
- Introducción al *lingo* de la programación por bloques y en especial de Scratch.

### Requisitos: (Conocimientos previos necesarios)

- No se requieren conocimientos previos más allá del manejo básico del ratón y de un navegador web
- Es interesante que los alumnos hayan registrado una cuenta de usuario gratuita en el sitio web de Scratch <http://scratch.mit.edu> para poder guardar así sus avances en la nube

### Actividad 1: *El Laberinto de Blockly Games*

Esta actividad permite a los alumnos iniciarse de manera informal a la programación por bloques, las instrucciones ensamblables, y gracias a que se trata de una actividad con una herramienta guiada les permite usar de una manera intuitiva los bucles y condicionales.

**Recursos:** Blockly

**Evaluación:** Es necesario que todos los alumnos sean capaces de resolver el nivel 7 del laberinto antes de continuar a la siguiente actividad.

### Actividad 2: *Introducción al editor de Scratch*

Introducción al editor de Scratch.

- Iniciar sesión.
- Presentación de las distintas áreas y utilidades del editor Scratch.
- Creación de un juego sencillo en el cambiaremos el fondo de pantalla, añadiremos un nuevo personaje y haremos que se mueva por la pantalla
- Renombrar el proyecto
- Guardarlo y volver a abrirlo

**Recursos:** Scratch

### Evaluación:

Los alumnos serán capaces de iniciar sesión en la página de Scratch, crear un nuevo proyecto, cambiarle el nombre, guardarlo y volver a abrirlo.

Los alumnos serán capaces de identificar las distintas secciones del editor y familiarizarse con el *lingo* propio de Scratch.

## Bucles y Condicionales. Blockly y Scratch.

### Objetivos trabajados:

- Bucles: los bloques repetir
- Bucles Anidados
- Bucles y Condicionales: introducción a los *sensores* de Scratch

### Requisitos: (Conocimientos previos necesarios)

- Es necesario haber completado las actividades de la sesión anterior.

### Actividad 1: Bloques Repetir

Usando las instrucciones del area sonido: *tocar nota, tocar tambor y tocar sonido*:

- Se explorarán las distintas posibilidades de los bloques repetir.
- Se explorarán bloques repetir dentro de bloques repetir (bucles anidados), así como instrucciones dentro y fuera de bloques repetir anidados.
- Se explorarán el uso de *sensores* y condicionales en combinación con bucles *repetir por siempre*.

**Recursos:** Blockly, Scratch, [Auriculares para cada PC](#)

### Evaluación:

Los alumnos serán capaces de predecir el número de veces que una determinada instrucción se va a ejecutar en función del bloque repetir en que se encuentra embebida.

Los alumnos serán capaces de identificar aquellos bloques de código que solo se ejecutarán en caso de que haya una condición cuyo valor que sea cierto

## Dibujando formas geométricas. Blockly y Scratch.

### Objetivos trabajados:

- Aprender a dibujar formas geométricas de manera programática.
- Uso de bucles y repeticiones con un propósito concreto.

### Requisitos: *(Conocimientos previos necesarios)*

- Haber completado la actividad de bucles y condicionales.

### Actividad 1: *La Tortuga de Blockly Games*

Actividad de los Juegos Blockly para dibujar formas geométricas usando bloques de instrucciones. Al ser una actividad guiada con un número limitado de opciones de las que escoger los alumnos son capaces de encontrar soluciones de manera intuitiva al problema que se plantea

**Recursos:** Blockly Games

**Evaluación:** Es necesario que todos los alumnos sean capaces de resolver el nivel **X** de la tortuga antes de continuar a la siguiente actividad.

### Actividad 2: *Dibujando formas geométricas con Scratch*

Utilizaremos lo aprendido durante la Actividad 1 para aprender a dibujar formas con Scratch. El gran número de opciones disponibles en Scratch hace que los alumnos se sientan abrumados al enfrentarse a esta actividad por lo que aprovecharemos la ocasión para mostrarles que ambos juegos tienen las mismas opciones disponibles aunque a veces bajo nombres distintos y que adaptando las soluciones implementadas en uno de los juegos nos pueden servir en el otro entorno.

**Recursos:** Scratch

**Evaluación:** Los alumnos serán capaces de dibujar distintos polígonos: cuadrado, pentagono, hexagono... con un número creciente de lados.

## Formas Geométricas: Aplicando la Abstracción. Scratch.

### Objetivos trabajados:

- Dibujar formas geométricas de forma programática
- Capacidad de abstracción y búsqueda de patrones

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de la sesión anterior.

### Actividad 1: *Del Triángulo a la Circunferencia*

Continuando con la actividad anterior de dibujar formas geométricas reflexionaremos sobre la relación que existe entre los ángulos de cada arista y el número de lados de cada polígono, así como el número de total de grados que es necesario girar para dibujar cada polígono.

Esto nos llevará a buscar la forma de crear una *función* parametrizable (Custom Block en el *lingo* de Scratch) que nos permita dibujar polígonos regulares de cualquier número de lados, incluida la circunferencia

**Recursos:** Scratch

### Evaluación:

Los alumnos serán capaces ver que, independientemente del número de lados, para dibujar un polígono cerrado es necesario girar 360 grados.

Los alumnos serán capaces de ver la circunferencia como un caso especial de polígono con un número de lados *muy grande* (infinito).

Los alumnos crearán sus primeras funciones (Custom Blocks) aplicando el concepto de abstracción, y serán capaces de comprobar las ventajas de disponer de una solución genérica.

## Sistema de Coordenadas Cartesiano. Posición y Movimiento. Scratch.

### Objetivos trabajados:

- Sistema de coordenadas cartesiano para representar puntos en el plano y movimientos.

### Requisitos: (Conocimientos previos necesarios)

- Es necesario haber completado las actividades de la sesión anterior.

### Actividad 1: *El sistema de coordenadas cartesiano*

Jugaremos a colocar el personaje de Scratch en en los distintos cuadrantes del escenario haciendo uso de la instrucción *ira a x:\_y:\_*.

Usaremos la instrucción de *número al azar* para deslizar el personaje por una área determinada del escenario.

### Actividad 2: *Moviendo a un personaje con las flechas del teclado*

Utilizaremos las instrucciones *cambiar x* y *cambiar y* para cambiar la posición del personaje en la cuatro direcciones: izquierda, derecha, arriba y abajo.

Explicaremos la diferencia de las instrucciones *fijar x/y* y *cambiar x/y*

**Recursos:** Scratch

### Evaluación:

Los alumnos serán capaces ver de comprender el sistema de coordenadas cartesiano para referirnos a puntos del plano, en este caso del escenario de Scratch.

Los alumnos serán capaces de hacer uso de sensores y condicionales dentro de un bucle infinito para leer las pulsaciones de las flechas del teclado y mover a un personaje por el escenario de Scratch.



## Variables. Scratch.

### Objetivos trabajados:

- Variables como espacio de memoria para guardar datos

### Requisitos: (Conocimientos previos necesarios)

- Es necesario haber completado las actividades de Posición y Movimiento de la sesión anterior.

### Actividad 1: *Las variables*

Exploraremos las siguientes funcionalidades:

- Definir una variable: diferencias entre variable global y local
- Modificar una variable: diferencias entre *fijar* y *cambiar* una variable

### Actividad 2: *Juego: La Macedonia*

En esta actividad pondremos en práctica los conocimientos que hemos aprendido acerca de posición y movimiento y el uso de variables.

Consiste en implementar un juego que haga que caigan *objetos* fruta desde el borde superior del escenario, en una posición horizontal aleatoria y que se desplacen verticalmente (caigan) hasta tocar el borde inferior o el usuario haga click sobre ellos.

Si el usuario hace click sobre un objeto fruta este desaparece del escenario y se incrementa en uno la variable frutas.

Adicionalmente se usará un objeto con dibujo de taza que seguirá al cursor, dando la apariencia de que se están recogiendo las frutas para preparar una macedonia.

**Recursos:** Scratch

### Evaluación:

Los alumnos serán capaces de definir un punto aleatorio en la esquina superior.

Los alumnos serán capaces de emular la caída de los objetos moviendolos horizontalmente hacia abajo a través de sus coordenadas cartesianas.

Los alumnos serán capaces de hacer uso de los sensores y eventos de Scratch para detectar cuando se hace clic sobre un objeto.

Los alumnos serán capaces de definir, fijar y cambiar el valor de una variable, en este caso una variable local.

## Variables, Bucles y Condicionales. Scratch.

### Objetivos trabajados:

- Pensamiento computacional
- Datos, Bucles y Condiciones como elementos base de la programación

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de Posición y Movimiento y Variables de la sesiones anteriores.

### Actividad 1: *Modificando variables dentro de bucles anidados*

Exploraremos la ejecución de programas consistentes en bucles anidados y condicionales a través del valor de variables que se modifican durante la ejecución, con el objetivo de predecir los valores que las distintas variables toman a lo largo de la ejecución del programa.

**Recursos:** Scratch

### Evaluación:

Los alumnos serán capaces de trazar la ejecución de un programa compuesto por bucles anidados y condicionales y decidir cuál será el valor final de las variables del programa.

## Estrategias para juegos I. Scratch.

### Objetivos trabajados:

- Pensamiento computacional
- Estrategias para juegos

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de Posición y Movimiento, Variables, Bucles y Condicionales de la sesiones anteriores.

### Actividad 1: *Implementación de un juego de naves espaciales. Invasores del Espacio.*

Durante esta actividad implementaremos un juego consistente en tres objetos:

- Plátano volante: se desliza aleatoriamente por las posiciones de la mitad superior del escenario
- Cohete: se desplaza de izquierda a derecha por la base del escenario bajo el control de las teclas flecha izquierda y derecha del teclado
- Proyectoil: una pelota que cuando se pulsa la barra espaciadora se coloca detrás del cohete y se mueve en dirección vertical, si el proyectil toca al Plátano se incrementará el valor de una variable puntuación.

**Recursos:** Scratch

### Evaluación:

Los alumnos serán capaces de poner en práctica lo aprendido durante las sesiones anteriores implementando este clásico de los videojuegos.

## Estrategias para juegos II. Scratch.

### Objetivos trabajados:

- Pensamiento computacional
- Estrategias para juegos

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de Posición y Movimiento, Variables, Bucles y Condicionales de la sesiones anteriores.

### Actividad 1: *Exploraremos distintas estrategias comunes en la implementación de videojuegos*

Durante esta actividad exploraremos distintas estrategias presentes para la creación de videojuegos:

- cambio de pantalla
- detección de colisiones
- salto y caída parabólicas
- tamaños para dar sensación de perspectiva y profundidad

**Recursos:** Scratch

### Evaluación:

Los alumnos serán capaces de usar el entorno de programación de Scratch con fluidez para seguir las instrucciones de esta actividad y explorar su propia creatividad.

## Dibujo 2D y Diseño 3D

## Dibujo 2D: Introducción a Processing JS - Líneas y Polígonos

### Objetivos trabajados:

- Pensamiento geo espacial
- Geometria

### Requisitos: *(Conocimientos previos necesarios)*

- No son necesarios conocimientos previos de informática más allá del manejo básico del ratón, el teclado y un navegador web.
- Es necesario conocer el sistema de coordenadas cartesiano, que se explica en las sesiones de programación.

### Actividad 1: *Exploración del entorno de programación de ProcessingJS*

Durante esta actividad exploraremos el entorno de programación del lenguaje Processing en su versión Javascript:

- crear y guardar un nuevo proyecto
- dibujando líneas
- dibujando un rectángulo
- dibujando elipses y círculos

**Recursos:** Processing JS

### Evaluación:

Los alumnos serán capaces acceder, crear, guardar y volver a abrir proyectos en ProcessingJS.

Los alumnos serán capaces de dibujar líneas y formas geométricas básicas.

## Dibujo 2D: Stroke y Colores. Actividad: Dibujamos una Casa

### Objetivos trabajados:

- Pensamiento geo espacial
- Geometria

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario conocer el sistema de coordenadas cartesiano y la sesión anterior de ProcessingJS.

### Actividad 1: *Líneas, Poligonos, Trazos y Colores: Dibujando una casa.*

Durante esta actividad propondremos a los alumnos dibujar una casa utilizando las formas geométricas que hemos aprendido a dibujar en la sesión anterior: rectas, triángulos, rectángulos, elipses... Aprenderemos a manejar el grosor de los trazos y el color tanto de las líneas como del relleno.

**Recursos:** Processing JS

### Evaluación:

Los alumnos serán capaces de utilizar líneas y formas geométricas básicas, trazos y colores para dibujar una casa.

## Dibujo 2D: Control Flujo y Variables en ProcessingJS

### Objetivos trabajados:

- Pensamiento geo espacial
- Geometria
- Pensamiento computacional y programación

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario conocer el sistema de coordenadas cartesiano y haber completado la actividad de la sesión anterior de ProcessingJS.

### Actividad 1: *Programando en Javascript, Control de Flujo y Variables: Dibujando una casa II.*

Durante esta actividad continuaremos con el dibujo de la casa de la actividad anterior introduciendo algunos conceptos de programación en javascript.

Usaremos:

- variables para configurar colores y grosor de trazos
- bucles for en javascript para dibujar formas repetidas: por ejemplo una valla..

**Recursos:** Processing JS

### Evaluación:

Los alumnos serán capaces de utilizar líneas y formas geométricas básicas, trazos y colores para dibujar una casa.

Los alumnos serán capaces de definir variables e iterar usando bucles *for* en javascript.

**Diseño 3D: OpenSCAD Polígonos, Rotacion y Traslacion. Actividad: Muñeco Nieve.**

**Objetivos trabajados:**

- Pensamiento geo espacial
- Geometria 3D
- Pensamiento computacional y programación

**Requisitos:** *(Conocimientos previos necesarios)*

- Es necesario conocer el sistema de coordenadas cartesiano y es interesante haber completado la actividad de las sesiones anteriores de ProcessingJS.

**Actividad 1:** *OpenSCAD: Polígonos, Rotacion y Traslacion: Creando un muñeco de nieve*

Durante esta actividad aprenderemos a dibujar con OpenSCAD formas geométricas en 3D:

- cuadrados y rectángulos
- esferas
- cilindros

Aprenderemos a desplazar y rotar dichas esferas para construir un muñeco de nieve compuesto de esferas y cilindros

**Recursos:** OpenSCAD programa libremente instalable en múltiples sistemas operativos.

**Evaluación:**

Los alumnos serán capaces de crear y guardar proyecto en OpenSCAD

Los alumnos serán capaces de dibujar formas geométricas en 3D usando las funciones de OpenSCAD.

Los alumnos serán capaces de trasladar y rotar dichas formas geométricas para crear la figura de un muñeco de nieve.



**Diseño 3D: OpenSCAD Booleanas: Union / Diferencia / Interc. Actividad: Ficha Dominó.**

**Objetivos trabajados:**

- Pensamiento geo espacial
- Geometria 3D
- Pensamiento computacional y programación

**Requisitos:** *(Conocimientos previos necesarios)*

- Es necesario haber completado la actividad de la sesion anterior de OpenSCAD.

**Actividad 1:** *OpenSCAD: Unión / Diferencia / Intersección: Dibujando una Ficha de Dominó.*

Durante esta actividad aprenderemos a aplicar operaciones de union, diferencia e intersección de las formas geométricas básicas que hemos aprendido en las sesiones anteriores.

Aprenderemos a dibujar una ficha de dominó usando la union/diferencia de rectangulos y esferas.

**Recursos:** OpenSCAD programa libremente instalable en múltiples sistemas operativos.

**Evaluación:**

Los alumnos serán capaces de dibujar, rotar y trasladar formas geométricas en 3D usando las funciones de OpenSCAD.

Los alumnos serán capaces de realizar operaciones de diferencia/unión/intersección sobre dichas formas geométricas para crear una ficha de dominó.

**Diseño 3D: OpenSCAD Booleanas: Union / Diferencia / Interc. Actividad: Dado Parchis.**

**Objetivos trabajados:**

- Pensamiento geo espacial
- Geometria 3D
- Pensamiento computacional y programación

**Requisitos:** *(Conocimientos previos necesarios)*

- Es necesario haber completado la actividad de la sesion anterior de OpenSCAD.

**Actividad 1:** *OpenSCAD: Unión / Diferencia / Intersección: Dibujando un Dado de Parchis.*

Durante esta actividad profundizaremos en la aplicacion operaciones de unión, diferencia e intersección de las formas geométricas básicas que hemos aprendido en las sesiones anteriores.

Aprenderemos a dibujar una dado de parchis usando la unión/diferencia/intersección de rectangulos y esferas.

**Recursos:** OpenSCAD programa libremente instalable en múltiples sistemas operativos.

**Evaluación:**

Los alumnos serán capaces de dibujar, rotar y trasladar formas geométricas en 3D usando las funciones de OpenSCAD.

Los alumnos serán capaces de realizar operaciones de diferencia/unión/intersección sobre dichas formas geométricas para crear una ficha de dominó.

**Diseño 3D: Introducción FreeCAD/TinkerCAD. Entornos de diseño 3D gráficos.**

**Objetivos trabajados:**

- Pensamiento geo espacial
- Geometria 3D
- Pensamiento computacional y programación

**Requisitos:** *(Conocimientos previos necesarios)*

- Es importante haber completado las actividades de las sesiones anteriores de OpenSCAD.

**Actividad 1:** *FreeCAD/TinkerCAD. Entornos de diseño 3D gráficos.*

Durante esta actividad exploraremos los conceptos que hemos aprendido en OpenSCAD en un entorno de diseño 3D gráficos como FreeCAD o TinkerCAD

**Recursos:** FreeCAD o TinkerCAD

**Evaluación:**

Los alumnos serán crear y guardar proyectos sencillos en la herramienta gráfica FreeCAD o TinkerCAD.

## Diseño 3D: Introducción FreeCAD/TinkerCAD. Entornos de diseño 3D gráficos.

### Objetivos trabajados:

- Pensamiento geo espacial
- Geometría 3D
- Pensamiento computacional y programación

### Requisitos: *(Conocimientos previos necesarios)*

- Es importante haber completado las actividades de las sesiones anteriores de OpenSCAD.

### Actividad 1: *FreeCAD/TinkerCAD. Entornos de diseño 3D gráficos.*

Durante esta actividad exploraremos los conceptos que hemos aprendido en OpenSCAD en un entorno de diseño 3D gráficos como FreeCAD o TinkerCAD

**Recursos:** FreeCAD o TinkerCAD

### Evaluación:

Los alumnos serán crear y guardar proyectos sencillos en la herramienta gráfica FreeCAD o TinkerCAD.

**impresión: Visita a una copistería 3D e imprimir uno de nuestros diseños.**

**Objetivos trabajados:**

- Creatividad e impresión 3D

**Requisitos:** *(Conocimientos previos necesarios)*

- Es importante haber completado las actividades de las sesiones anteriores de OpenSCAD.

**Actividad 1:** *Visita a una copistería 3D.*

Durante esta actividad aprenderemos a exportar alguno de nuestros diseños 3D de las sesiones anteriores: Una Ficha de Dominó o Un Dado de Parchís.

Visitaremos una copistería 3D en la que imprimimos uno de nuestros diseños.

**Recursos:** FreeCAD o TinkerCAD

**Evaluación:**

Los alumnos serán capaces de exportar sus diseños 3D y ordenar una impresión en una copistería local.

## **Robótica**

## Introducción a Arduino: Pines, Entrada y Salida. Bitbloq.

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción al entorno de programación para Arduino por bloques Bitbloq.

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de programación por bloques con Blockly y Scratch.

### Actividad 1: *Introducción a las placas microcontroladoras*

Veremos lo que son las placas microcontroladoras,

- entradas: sensors
- salidas: salidas motores y actuadores
- pines digitales y analogicos
- distintos tipos de sensores

### Actividad 2: *Introducción a Bitbloq*

Crearemos una cuenta de usuario para escribir nuestro primer programa Arduino

- registrar una cuenta de usuario en Bitbloq
- instalación de drivers
- instalacion serial-uploader
- creación de un programa que encienda un led

**Recursos:** Navegador Chrome, Drivers, Serial Uploader, cuenta usuario en BitBloq

### Evaluación:

Los alumnos serán capaces de entender y distinguir que es una placa controladora Arduino, los elementos de entrada y salida, pines digitales y analogicos asi como una lista de sensores y actuadores de uso común.

Los alumnos serán capaces de crear su primer programa sencillo en Bitbloq e instalarlo en una placa Arduino a través del serial uploader.

## Parpadeo Leds. Actividad: Programación Semáforo

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción a la programación de Arduino con Bitbloq.

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de programación por bloques con Blockly y Scratch y las sesiones anteriores de Arduino y Bitbloq

### Actividad 1: *Estados y Algoritmos: Programando un Semáforo*

En esta actividad introduciremos a los alumnos a las máquinas de estados y los algoritmos a través de una simple actividad en la que programaremos las luces de un semáforo.

**Recursos:** Placa Arduino, Led, Bitbloq.

### Evaluación:

Los alumnos serán capaces de programar con Bitbloq las luces de un semáforo.

Los alumnos tendrán una visión genérica de lo que son las máquinas de estados y los algoritmos.



## Sensor Luz. Actividad: Farola Automatica

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción a los sensores de luz

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de las sesiones anteriores de Arduino y Bitbloq

### Actividad 1: *Estados y Algoritmos: Programando un una Farola Automática*

En esta actividad introduciremos a los alumnos a los sensores de luz e implementaremos una farola automática:

- utilizaremos un nivel mínimo de luz para encender un led
- usaremos dos niveles de luz: uno para encender la farola y otro distinto para apagarlo
- usaremos un nivel de luz mínimo y un temporizador para apagar la farola *a la mañana siguiente*.
- 

**Recursos:** Placa Arduino, Leds, Sensores de Luz, Bitbloq

### Evaluación:

Los alumnos serán capaces de usar los valores de un sensor de luz para encender un led.

Los alumnos tendrán una visión genérica de lo que son las máquinas de estados y los algoritmos.

## Servo Rotacion Continua. Actividad: Coche "Solar"

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción a los sensores de luz
- Introducción a los servos de rotación continua.

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de las sesiones anteriores de Arduino y Bitbloq

### Actividad 1: *Haciendo avanzar los servo motores según nivel de luz. El Coche Solar.*

En esta actividad introduciremos a los alumnos a los servomotores de rotación continua en combinación con los sensores de luz que habíamos visto en la sesión anterior.

Construiremos un coche con el chasis de un printbot y dos servos de rotación continua.

El coche avanzara en línea recta si el nivel de luz es mayor que un determinado umbral.

**Recursos:** Placa Arduino, Leds, Sensores de Luz, Bitbloq

### Evaluación:

Los alumnos serán capaces de usar servos de rotacion continua con Bitbloq.

Los alumnos serán capaces de distinguir el sentido de rotación de cada servo es distinto para que el coche avance.

Los alumnos tendrán una visión genérica de lo que son las máquinas de estados y los algoritmos.

## Construcción Sensor Flexibilidad: Aluminio / Celo / Carboncillo / Cartón

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción a los componentes electrónicos hechos a mano.

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de las sesiones anteriores de Arduino y Bitbloq

### Actividad 1: *Construiremos un sensor de flexibilidad con materiales de manualidades.*

En esta actividad a los alumnos a la creación de componentes electrónicos hechos con materiales caseros.

Crearemos un sensor de flexibilidad usando cartulina, papel de aluminio, carboncillo o lápiz, dos latiguillos de cable.

Una tira de cartulina doblada de modo que en un lado pegaremos el papel de aluminio y del otro la pintaremos con carboncillo de modo que al doblar la tira ambos lados, el papel de aluminio y el carboncillo, estén en contacto aunque ligeramente separados.

A continuación pegaremos los latiguillos de cable a cada uno de los extremos de modo que queden en contacto cada uno con su conductor: el aluminio o el carboncillo.

Teniendo el sensor “cerrado” con los lados enfrentados, cuanto mayor sea la curvatura que apliquemos mayor será el contacto del carboncillo y el aluminio disminuyendo la resistencia y aumentando el paso de corriente, que usaremos para medir el grado de curvatura.

**Recursos:** Cartulina, papel de aluminio, carboncillo o lápiz, pegamento, dos latiguillos de cable. Placa Arduino.

### Evaluación:

Los alumnos serán capaces de crear y usar con una placa Arduino un sensor de flexibilidad hecho con materiales de manualidades.

## Servo Direccional: Actividad: Paso a Nivel

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción a los servos de rotación por pasos.

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de las sesiones anteriores de Arduino y Bitbloq

### Actividad 1: *Usando un servomotor de rotación controlada construiremos un Paso a Nivel.*

En esta actividad introduciremos a los alumnos a los servomotores de rotación por pasos y modificaremos nuestro programa del semáforo para construir un Paso a Nivel

**Recursos:** Placa Arduino, Leds, Sensores de Luz, Servo rotation por pasos, Bitbloq

### Evaluación:

Los alumnos serán capaces de usar servos de rotación por pasos con Bitbloq.

Los alumnos serán capaces de modificar un programa completado anteriormente para añadir nueva funcionalidad manteniendo la funcionalidad antigua.

## Robótica: Programando un Robot Sigue líneas

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción a los sensores de infrarrojos.
- Introducción a la robótica.

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de las sesiones anteriores de Arduino y Bitbloq

### Actividad 1: *Programando un Robots Sigue Líneas*

En esta actividad introduciremos a los alumnos a los sensores digitales infrarrojos y con dos servomotores de rotación continua y el chasis de un printbot construiremos un robot siguelineas.

**Recursos:** Placa Arduino, Sensores de Infrarrojo, Servo rotación continua Bitbloq

### Evaluación:

Los alumnos serán capaces de usar sensores digitales de infrarrojos con Bitbloq.

Los alumnos serán capaces de programar un robot sigue líneas.

## Robótica: Programando un Robot Sigue/Huye Luz

### Objetivos trabajados:

- Introducción a la electro-robótica y placas microcontroladoras Arduino
- Introducción a los sensores de luz.
- Introducción a la robótica.

### Requisitos: *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de las sesiones anteriores de Arduino y Bitbloq

### Actividad 1: *Programando un Robots Sigue/Huye Luz*

En esta actividad usaremos dos sensores de luz y con dos servomotores de rotación continua y el chasis de un printbot construiremos un robot sigue/huye luz.

**Recursos:** Placa Arduino, Sensores de Infrarrojo, Servo rotación continua Bitbloq

### Evaluación:

Los alumnos serán capaces de usar sensores de luz con Bitbloq.

Los alumnos serán capaces de programar un robot sigue/huye luz con Bitbloq.

**Detrás de la cortina: Código detrás de los bloques.**

**Objetivos trabajados:**

- Introducción a los lenguajes de programación textuales

**Requisitos:** *(Conocimientos previos necesarios)*

- Es necesario haber completado las actividades de las sesiones anteriores de programación con Blockly y Scratch así como Arduino y Bitbloq

**Actividad 1:** *Comparando lenguajes de programación*

En esta actividad compararemos distintos lenguajes de programación “profesionales” con los lenguajes por bloques que hemos aprendidos:

Compararemos Blockly, Scratch y Bitbloq con Python, Java y C

**Recursos:**

**Evaluación:**

Los alumnos serán capaces encontrar las similitudes y diferencias entre los distintos lenguajes de programación comparados, y serán capaces de apuntar algunas ventajas e inconvenientes de cada uno.

## Evaluación de todo el programa

### **Análisis de Fortalezas/Oportunidades/Debilidades/Amenazas**

#### **Fortalezas (Internas)**

- Novedoso
- Innovador
- Divertido
- Creativo
- Valor formativo
- Material audiovisual disponible
- Aula virtual
- Fomenta la interacción padre/hijo.
- 

#### **Oportunidades (Externas)**

- Ampliación a más contenidos
- Colaboraciones y sinergias
- Día de la ciencia en la calle
- 

#### **Debilidades (Internas)**

- Primera vez que se realiza la actividad
- Poca experiencia en actividades tan largas
- Capacidad para equilibrar el valor formativo y la diversión, para mantener viva la atención de los alumnos.
- Número de alumnos.

#### **Amenazas (Externas)**

- Expectativa, ideas preconcebidas o desconocimiento.
- Compatibilización de horarios con otras actividades.
- Horarios y disponibilidad de las instalaciones del centro educativo