

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 1</p>



GUÍA DE LABORATORIO

(formato docente)

INFORMACIÓN BÁSICA					
ASIGNATURA:	TECNOLOGIAS DE INFORMACION				
TÍTULO DE LA PRÁCTICA:	Construcción de un arduino				
NÚMERO DE PRÁCTICA:	01	AÑO LECTIVO:	2023	NRO. SEMESTRE:	2023A
TIPO DE PRÁCTICA:	INDIVIDUAL	X			
	GRUPAL		MÁXIMO DE ESTUDIANTES		
FECHA INICIO:	08/05/2023	FECHA FIN:	12/05/2023	DURACIÓN:	2 horas
RECURSOS A UTILIZAR: Arduino, Tinkercad					
DOCENTE(s): Mg. Maribel Molina Barriga					

OBJETIVOS/TEMAS Y COMPETENCIAS	
OBJETIVOS: <ul style="list-style-type: none"> Utiliza simuladores o controladores para el entorno de desarrollo de un Arduino Crea una simulación de un prototipo electrónico de manera rápida con el propósito de validar alguna propuesta aplicativa. 	
TEMAS: <ul style="list-style-type: none"> Simulador de Arduino 	
COMPETENCIAS	C.e Identifica de forma reflexiva y responsable, necesidades a ser resueltas usando tecnologías de información y/o desarrollo de software en los ámbitos local, nacional o internacional, utilizando técnicas, herramientas, metodologías, estándares y principios de la ingeniería.

CONTENIDO DE LA GUÍA
I. MARCO CONCEPTUAL ¿Qué es Arduino? Arduino es una plataforma de código abierto de prototipos electrónicos que se basa en hardware y software flexibles y fáciles de usar que ponen al alcance de cualquier persona la construcción de circuitos electrónicos/robots. En lo referente a hardware, se basa en placas que se pueden ensamblar a mano o que se pueden comprar directamente preensambladas. Cada una de las placas lleva un microcontrolador en el que se carga el programa software que es necesario desarrollar para “darle vida” a la placa.

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 2

Los pines digitales

- Son las conexiones digitales de los dispositivos conectados en la placa.
- La placa de Arduino cuenta con 14 pines digitales, que van del 0 al 13.
- Una señal digital solo puede tener dos estados:
 - 0 (LOW, bajo, false): Indica 0V de tensión enviados desde la placa.
 - 1 (HIGH, alto, true): Indica 5V de tensión enviados desde la placa.

Por lo tanto, cuando ponemos un pin digital a valor HIGH, la placa suministra 5V de tensión por la salida que hayamos indicado, y si ponemos el valor a LOW suministrará 0V de tensión.

(Ojo: Hay que tener en cuenta que los 5V no siempre son 5 ni los 0 siempre son 0)

Los pines digitales de Arduino pueden ser usados tanto de entrada como de salida.

Pines analógicos

Los pines analógicos pueden medir un rango de valores de voltaje, a diferencia de los digitales que solo entienden dos valores: 0-1, o lo que es lo mismo, 0V o 5V.

Con los pines analógicos vamos a poder leer valores intermedios entre 0V y 5V, representados con un valor entero comprendido entre 0 y 1023, ya que la información se representa en números de 10 bits, y también vamos a poder escribir en los pines valores comprendidos entre 0 y 255, ya que la información se representa en números de 8 bits.

Pines alimentación sensores

Además de los pines de entrada y salida descritos anteriormente, Arduino dispone de pines que nos permiten alimentar componentes externos, concretamente uno con 5V y otro con 3,3V.

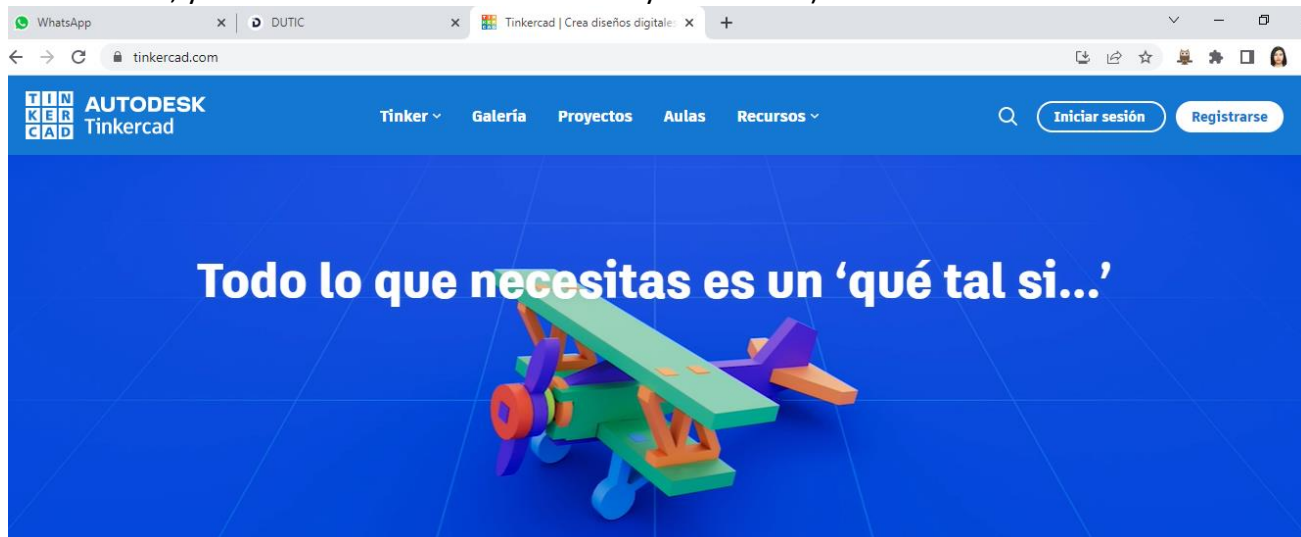
Arduino y robótica



Arduino es considerado una de las bases sobre las que empezar a trabajar en temas relacionados con robótica, ya que, por su simplicidad, permite adquirir conocimientos básicos para dar el paso posteriormente a tecnologías más complejas y completas.

II. EJERCICIO/PROBLEMA RESUELTO POR EL DOCENTE

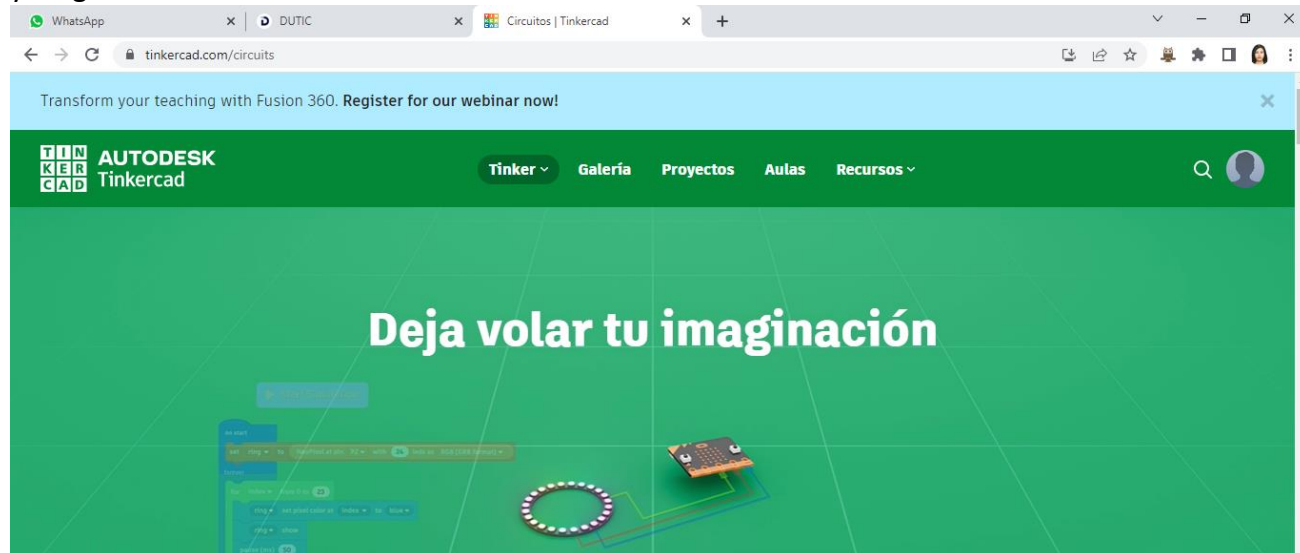
1º Accede a <http://www.tinkercad.com>

2º Registrarse lo cual es muy fácil y rápido (tendremos que añadir el país y nuestra edad, le daremos a SIGUIENTE, y finalmente un correo electrónico y contraseña)

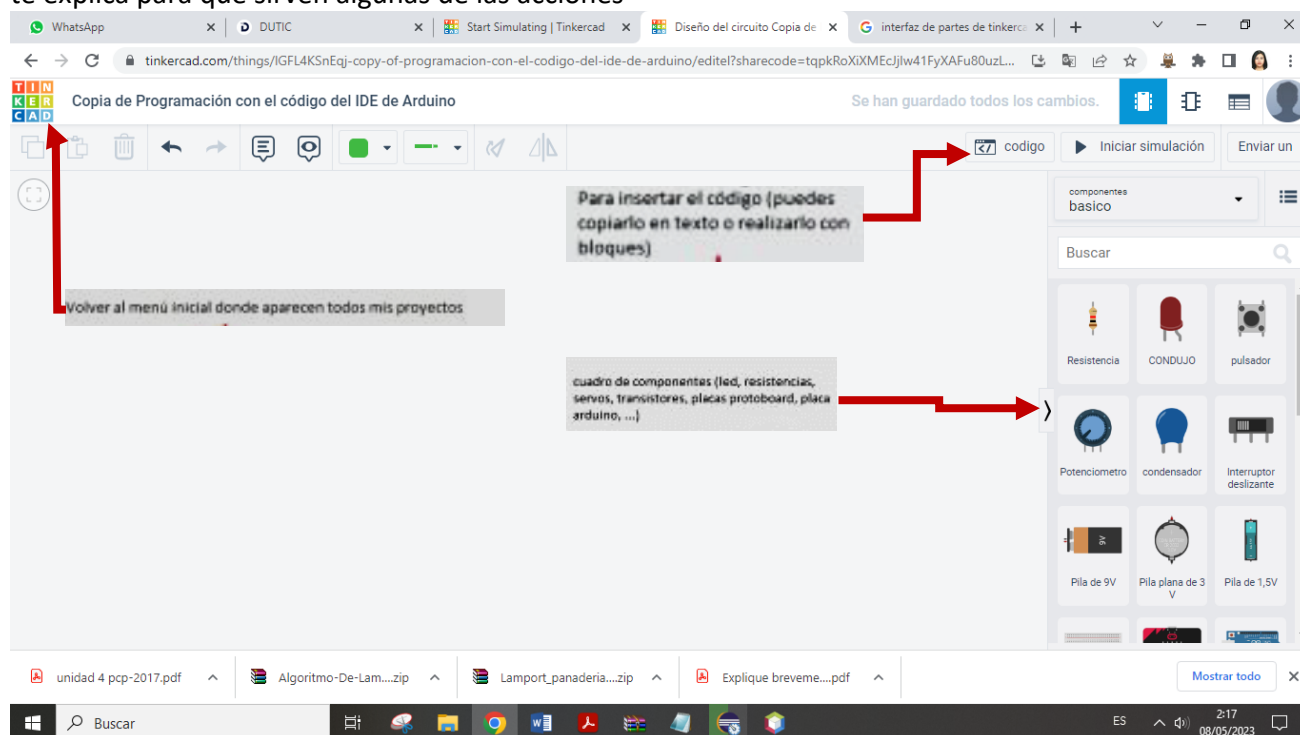


	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 3



3º Una vez registrados ya accedemos a la pantalla/interfaz del programa y hacemos click en TINKER y luego clic en CIRCUITS



4º Le damos a crear nuevo circuito y comenzamos a montar y programar nuestros circuitos digitales. Aquí te explica para que sirven algunas de las acciones



- Volver al menú inicial donde aparecen todos mis proyectos
- Rotar/Girar componentes (sobre todo válido para resistencias)
- Deshacer o rehacer la última acción realizada
- Para insertar el código (si ya lo tienes lo puedes copiar en «texto» sino lo puedes hacer por bloques)
- Cuadro de componentes electrónicos (hay una opción en la que se visualizan solamente los básicos y otra opción para que aparezcan todos los componentes posibles).

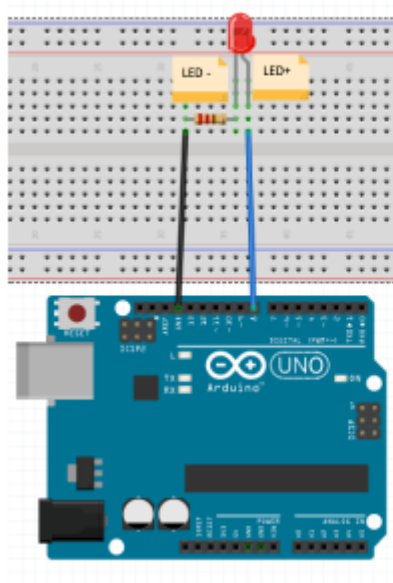
	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 4

Tienes que saber que al hacer click en algunos componentes te da la opción de cambiar sus características, por ejemplo el color de los LED, o los ohmnios en las resistencias.

1. LED INTERMITENTE

Se recuerda que la resistencia es de 220 Ohmnios.

ESQUEMA:



PROGRAMA:

```
#define pinLED 8

void setup() {
  pinMode(pinLED, OUTPUT);
}

void loop() {
  digitalWrite(pinLED, HIGH); //
  enciende el LED.
  delay(500); // retardo en
  milisegundos
  digitalWrite(pinLED, LOW); // apaga
  el LED.
  delay(500);
}
```

1. LED INTERMITENTE

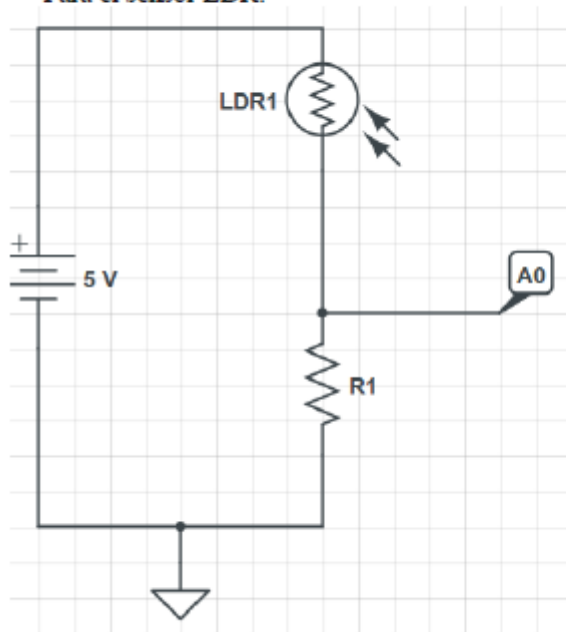
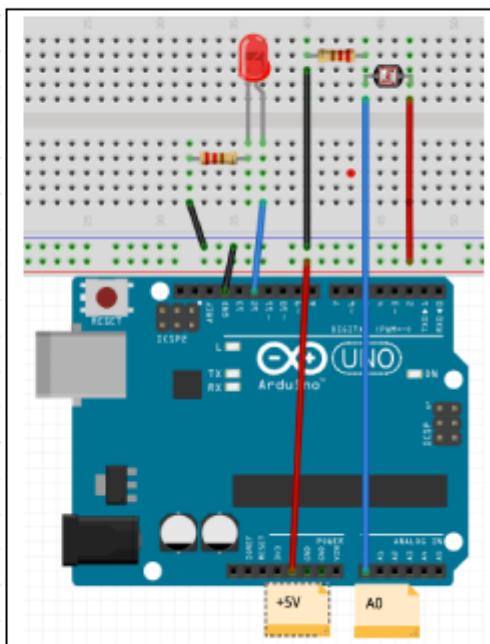
Se recuerda que la resistencia es de 220 Ohmnios.

2. DETECTOR DE LUZ

a) Realiza un montaje con Arduino para detectar cambios en el nivel de luminosidad del aula donde te encuentras.

b) Para ello utiliza un sensor de luz LDR (light-dependent resistor), una resistencia eléctrica y un LED. La idea es que cuando la intensidad luminosa disminuya un cierto umbral, el LED se active.

c) El sensor LDR es un sensor resistivo (fotoresistor), es decir que su resistencia eléctrica varía en función de la luz que recibe. Es un sensor analógico por lo que, para leer sus medidas, tenemos que conectarlo a una entrada analógica de Arduino. Si tapamos o acercamos el sensor a la luz veremos cómo cambian las medidas a través del monitor serie. El valor de la resistencia (R1) marcará la sensibilidad de las medidas y dependerá también del rango de resistencias que nos proporcione el LDR. Puedes empezar un una resistencia de 1KOhmio.

ESQUEMA:
Para el sensor LDR:

Circuito completo:

PROGRAMA:

```
// Detector de luz con LDR

#define pinLED 12

void setup() {
    pinMode(pinLED, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    int v = analogRead(A0);
    // El valor 600 (siguiente línea) se debe de ajustar dependiendo de la luz en el
    local
    // y del valor de la resistencia pull-down
    // poca luz -> v pequeño, mucha luz -> v grande.
    if (v < 600) digitalWrite(pinLED, HIGH);
    else digitalWrite(pinLED, LOW);
    Serial.println(v);
}
```



III. EJERCICIOS/PROBLEMAS PROPUESTOS

Revisar y probar las herramientas de simulación de un microcontrolador, elaborar los siguientes ejercicios:

1. CONTROL DE SERVO

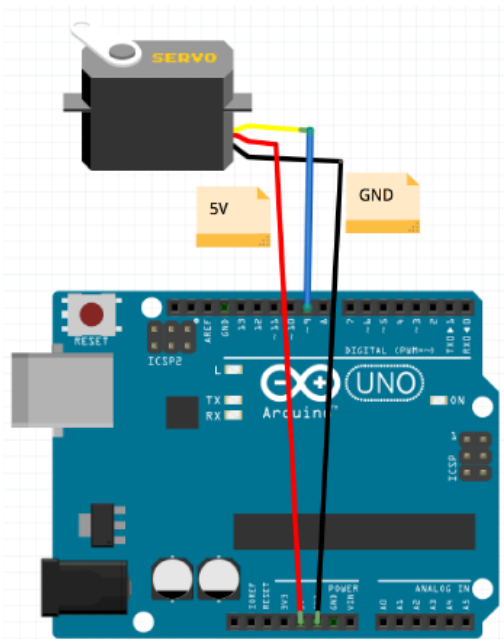
Realizar una práctica que haga girar un motor de rotación continua una vez con la siguiente secuencia: giro durante 2 segundos, parada durante 2 segundos y giro en sentido contrario durante 2 segundos.

Un servomotor de rotación se programa de forma muy similar a los servomotores vistos anteriormente. La diferencia

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 6

respecto a estos es que el servo de rotación continua puede girar (como su nombre indica) los 360 grados de forma continua. Hay que recordar que un servomotor sólo podía girar de 0 a 180 grados. Los servos de rotación continua llevan una reductora y proporcionan un buen par motor. Incluyen un circuito interno de control y las conexiones se realizan a través de 3 cables: Alimentación (+Vcc), Tierra (GND) y señal de control

ESQUEMA:



PROGRAMA:

```
#include <Servo.h>
Servo servoRotCont; // crea los objetos para
controlar los servomotores

void setup() {
  servoRotCont.attach(9);
  servoRotCont.write(0); //clockwise
  delay(2000);
  servoRotCont.write(90); //stop (el valor 90
  depende del motor.
  //Es conveniente probar valores por encima o por
  debajo
  //de 90 hasta comprobar que se para el
  servomotor.
  delay(2000);
  servoRotCont.write(180); //counter-clockwise
  delay(2000);
  servoRotCont.write(90); //stop
}

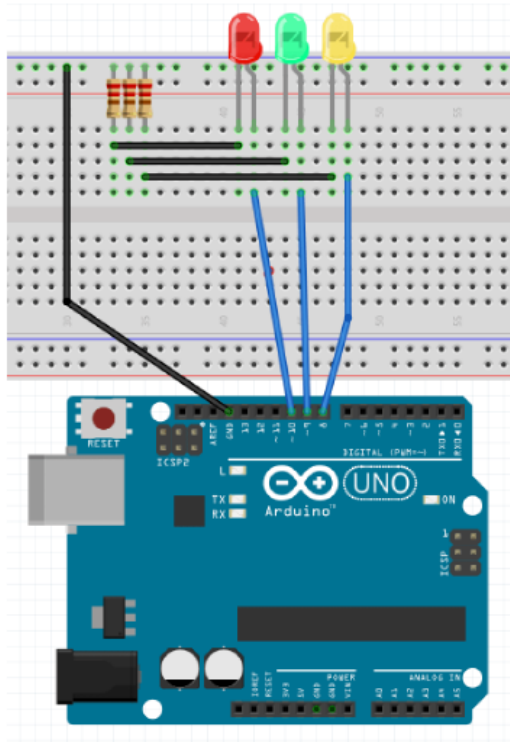
void loop() {
}
```

2. SECUENCIA LED

Realizar un montaje con 3 LEDs (rojo, verde y amarillo) que realice la siguiente secuencia con un intervalo de tiempo de 0,5 segundos entre cada uno ('1' indica encendido y '0' apagado):

- 100 (LED rojo encendido y el resto apagado).
- 010 (LED verde encendido y resto apagado).
- 001 (LED amarillo encendido y resto apagado).

ESQUEMA:



PROGRAMA:

```
#define pinLED1 8
#define pinLED2 9
#define pinLED3 10

void setup() {

  pinMode(pinLED1, OUTPUT);
  pinMode(pinLED2, OUTPUT);
  pinMode(pinLED3, OUTPUT);

}

void loop() {

  digitalWrite(pinLED1, HIGH); // enciende
  digitalWrite(pinLED2, LOW); // apaga
  digitalWrite(pinLED3, LOW); // apaga
  delay(500);
  digitalWrite(pinLED1, LOW);
  digitalWrite(pinLED2, HIGH);
  digitalWrite(pinLED3, LOW);
  delay(500);
  digitalWrite(pinLED1, LOW);
  digitalWrite(pinLED2, LOW);
  digitalWrite(pinLED3, HIGH);
  delay(500);

}
```

IV. CUESTIONARIO



1. Explique ¿Por qué es importante el funcionamiento del simulador Tinkercad de Arduino?
2. Investigar sobre los simuladores de Arduino y luego realizar un cuadro comparativo de Tinkercad con otros simuladores existentes.

V. REFERENCIAS Y BIBLIOGRAFÍA RECOMENDADAS:

- [1] Alejandro Peña Ayala, Tecnologías de la Información: su alineamiento al negocio de las organizaciones, Primera Edición, 2006.
- [2] Gómez, G., Planeación y Organización de Empresas, MC Graw Hill, 8va Edición. México. 1994.
- [3] Rodríguez, J., Cómo aplicar la planeación estratégica a la pequeña y mediana empresa, Thomson Learning, México, 2001.
- [4] Porte R., Millar, R., Tecnologías de Información en las Empresas, MC Graw Hill, 2da Edición. México, 2000.
- [5] Doherty R., Designing Business Intelligence Solutions, Microsoft. USA. 1999, pp.8.

URL's:

- Simulador Tinkercad: <https://www.tinkercad.com>
<https://www.tinkercad.com/things/IGFL4KSnEqj-copy-of-programacion-con-el-codigo-del-ide-de-arduino/editel?sharecode=tqpkRoXiXMEcJlW41FyXAFu80uzLPay9RIybaX1qLU>
- Video: <https://youtu.be/axUACNSQSYM>
- Video: <https://youtu.be/MojSo7OtF9w>
- Video: <https://youtu.be/X8dHbdhnGKY>

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p align="center">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLD-001</p>	<p>Página: 8</p>

TÉCNICAS E INSTRUMENTOS DE EVALUACIÓN	
<p>TÉCNICAS: <i>Problemas /Ejercicios propuestos</i> <i>/ Preguntas formuladas /</i> <i>Resolución de casos</i></p>	<p>INSTRUMENTOS: <i>Lista de cotejo</i></p>
<p>CRITERIOS DE EVALUACIÓN</p> <ul style="list-style-type: none"> • Identifica a los simuladores online de Arduino. • Utiliza el Arduino para resolver problemas. 	