



## A.3.1 Actividad de aprendizaje

---

Circuito sensor de detección de objetos, utilizando un Arduino, un sensor ultrasonico y un Display I2C LCD 16x2.

---

### Instrucciones

- Realizar un sistema simulado, capaz de detectar la presencia y/o ausencia de un objeto, a través de un circuito electrónico, utilizando un Arduino, y un **Sensor Ultrasonico HC-SR04**.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces, y debe ser nombrado con la nomenclatura **A3.1\_NombreApellido\_Equipo.pdf**.
- Es requisito que el .md contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo **Enlace a mi GitHub** y al concluir el reto se deberá subir a github.
- Desde el archivo **.md** exporte un archivo **.pdf** que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma **oficial** aquí se recibirá la calificación de su actividad.
- Considerando que el archivo .PDF, el cual fue obtenido desde archivo .MD, ambos deben ser idénticos.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

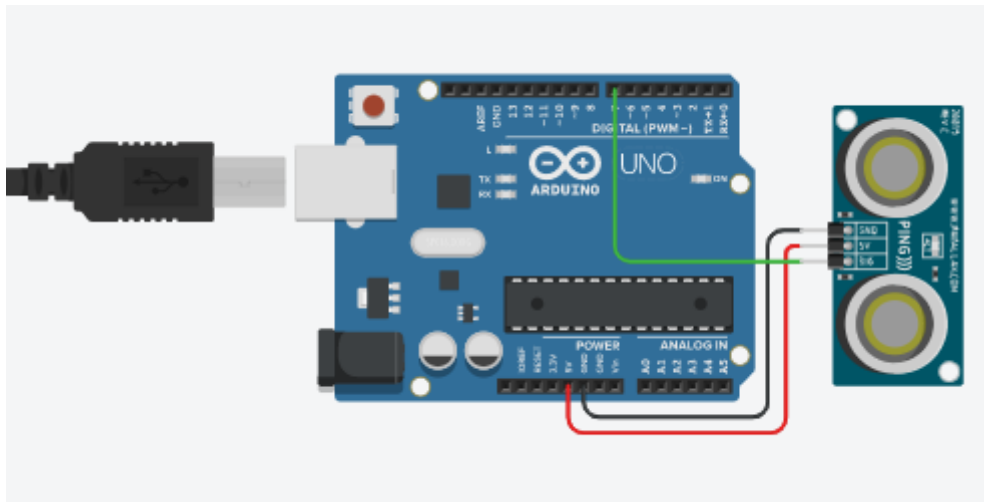
```
* readme.md
- blog
  - C3.1_TituloActividad.md
  - C3.2_TituloActividad.md
  - C3.3_TituloActividad.md
  - C3.4_TituloActividad.md
  - C3.5_TituloActividad.md
  - C3.6_TituloActividad.md
  - C3.7_TituloActividad.md
  - C3.8_TituloActividad.md
- img
- docs
  - A3.1_TituloActividad.md
  - A3.2_TituloActividad.md
  - A3.3_TituloActividad.md
```

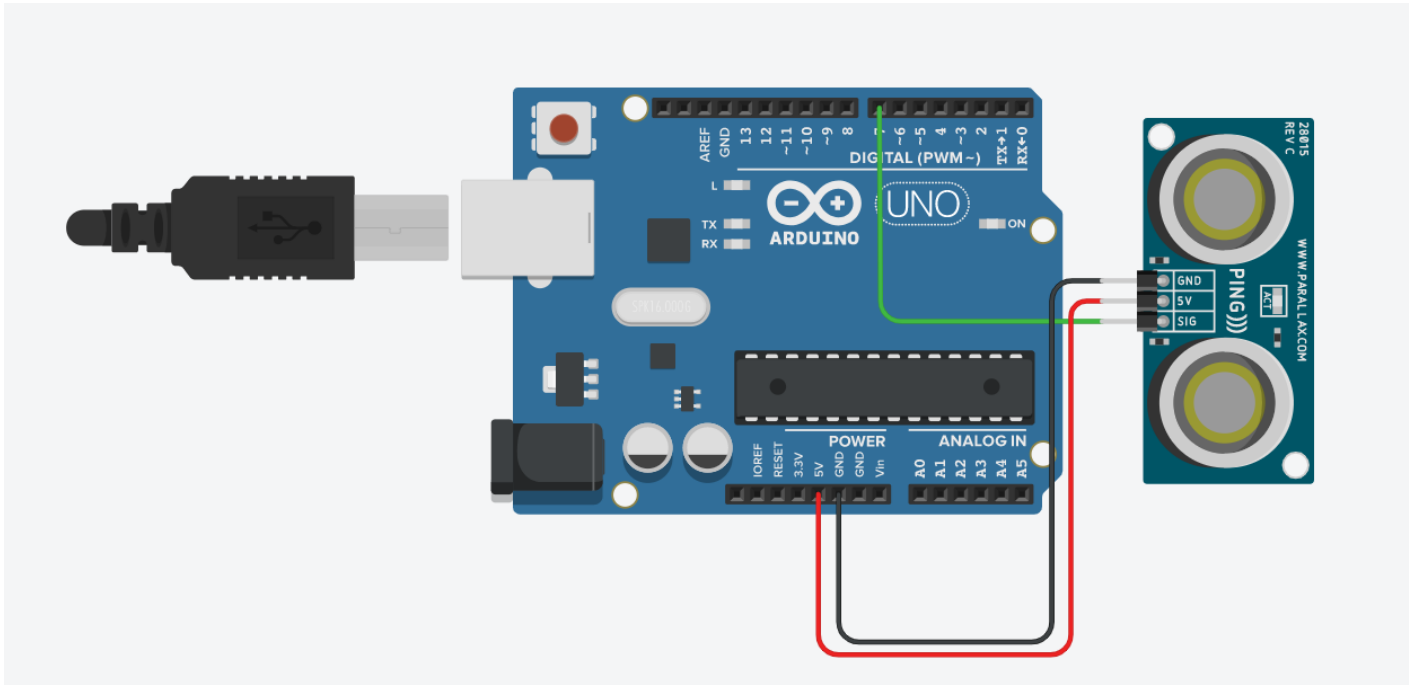
## Desarrollo

1. Utilizar el siguiente listado de materiales para la elaboración de la actividad

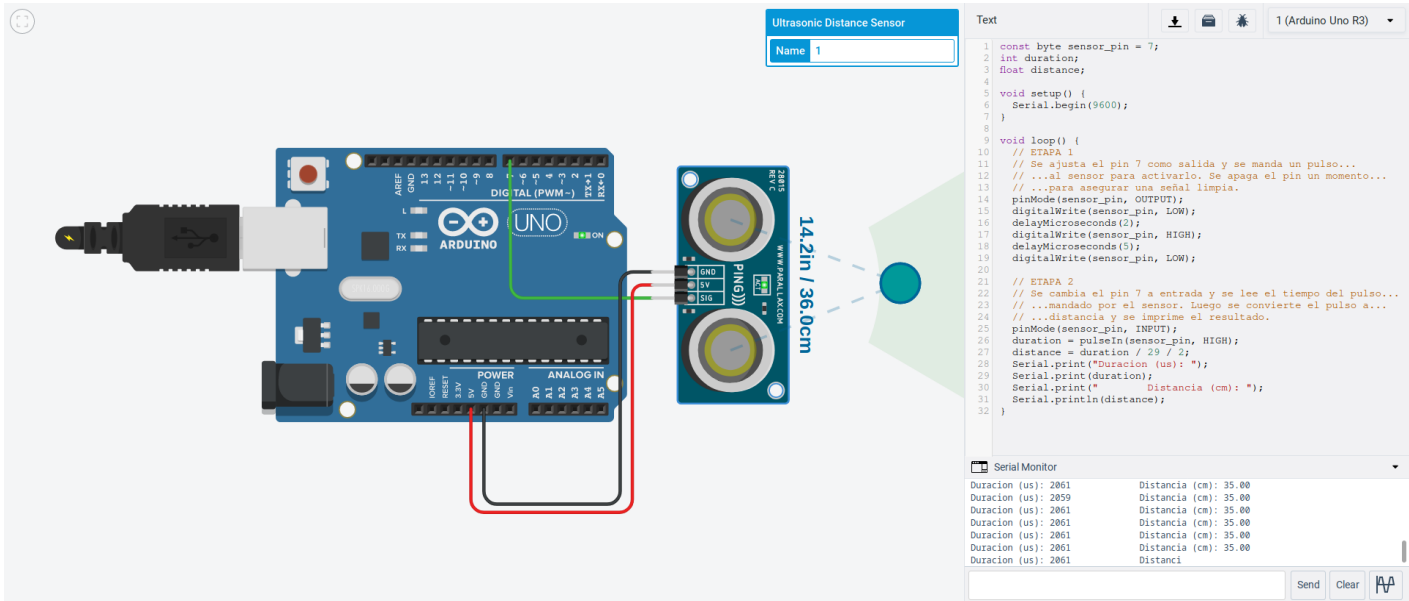
Cantidad	Descripción
1	Sensor Ultrasonico HC-SR04
1	Display LCD de 16x2
1	Fuente de voltaje de 5V
1	1 Potenciómetro 10k
1	Arduino UNO

2. Basado en la imagen que se muestra, ensamble dentro del simulador a utilizar, el circuito electrónico indicado en la **Figura 1**.





3. Realice el programa que permita a través de una de las entradas del Arduino, recibir el valor que registra el **Sensor Ultrasonico** al acercarse un objeto a distintas distancias.

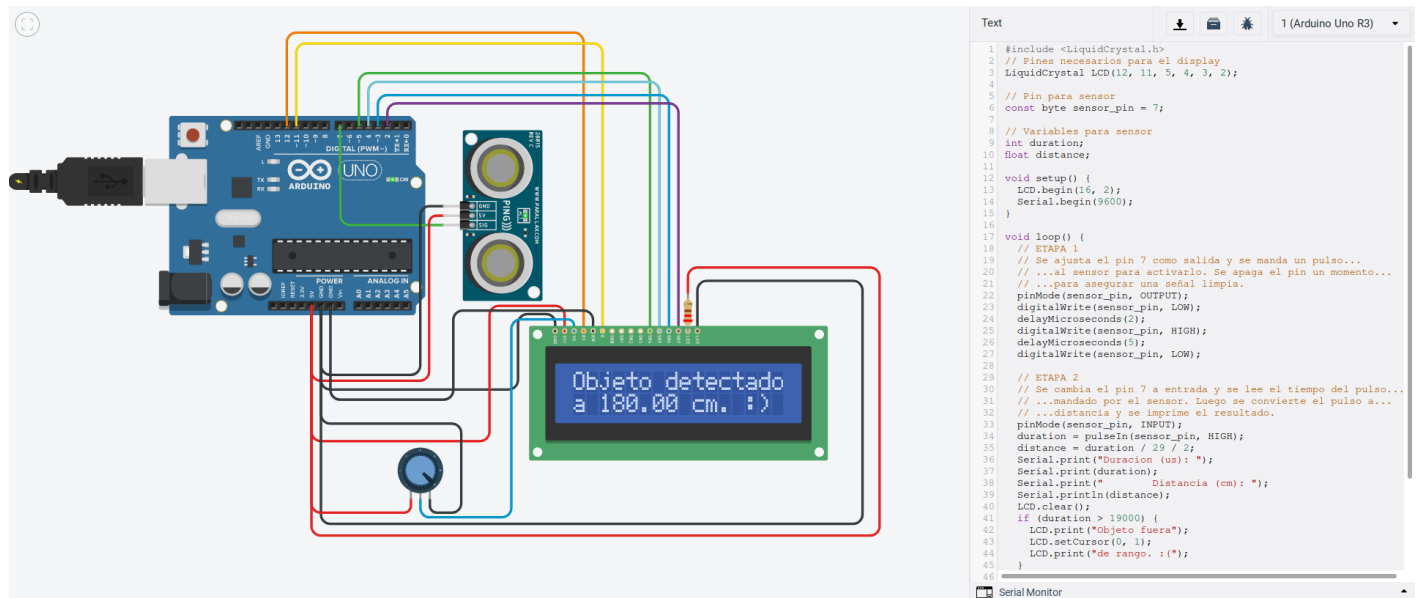


4. Considerando que el sensor ultrasonico tiene un rango mínimo y máximo de detección basado en el tiempo de retorno de la señal sonica, que valores se obtienen en la simulación bajo las **siguientes condiciones:** **Nota:** **Esta actividad se hizo con el sensor ultrasónico PING))) en lugar del HC-SR04.**

Numero	Condición 1	Condición2	El objeto es detectado?
1	5 cm de distancia al sensor	0 grados al eje perpendicular del sensor	El objeto es detectado

Numero	Condición 1	Condición2	El objeto es detectado?
2	50 cm de distancia al sensor	35 grados al eje perpendicular del sensor	El objeto no es detectado
3	100 cm de distancia al sensor	-35 grados al eje perpendicular del sensor	El objeto no es detectado
4	5 cm de distancia al sensor	90 grados al eje perpendicular del sensor	El objeto no es detectado
5	50 cm de distancia al sensor	-60 grados al eje perpendicular del sensor	El objeto no es detectado
6	350 cm de distancia al sensor	0 grados al eje perpendicular del sensor	El objeto no es detectado

5. Una vez completados los puntos anteriores, agregue a la Figura 1, **un Display I2C 16x2 LCD**, y coloque la imagen del circuito completado hasta este apartado.



6. Al haber completado la integración del Display I2C, ajuste el programa que le permita a través del display mostrar el siguiente mensaje, **"Objetivo detectado a ? cm** , y en caso de no lograr la detección indicar el mensaje **"Objetivo fuera de rango"**

```

// Librería para uso del display
#include <LiquidCrystal.h>

```

```
// Pines necesarios para el display
LiquidCrystal LCD(12, 11, 5, 4, 3, 2);
```

```
// Pin para sensor
const byte sensor_pin = 7;
```

```
// Variables para sensor
int duration;
float distance;
```

```
void setup() {
  LCD.begin(16, 2);
  Serial.begin(9600);
}
```

```
void loop() {
  // ETAPA 1
  // Se ajusta el pin 7 como salida y se manda un pulso...
  // ...al sensor para activarlo. Se apaga el pin un momento...
  // ...para asegurar una señal limpia.
  pinMode(sensor_pin, OUTPUT);
  digitalWrite(sensor_pin, LOW);
  delayMicroseconds(2);
  digitalWrite(sensor_pin, HIGH);
  delayMicroseconds(5);
  digitalWrite(sensor_pin, LOW);

  // ETAPA 2
  // Se cambia el pin 7 a entrada y se lee el tiempo del pulso...
  // ...mandado por el sensor. Luego se convierte el pulso a...
  // ...distancia y se imprime el resultado.
```

```
pinMode(sensor_pin, INPUT);
duration = pulseIn(sensor_pin, HIGH);
distance = duration / 29 / 2;
Serial.print("Duracion (us): ");
Serial.print(duration);
Serial.print("      Distancia (cm): ");
Serial.println(distance);
LCD.clear();
if (duration > 19000) {
    LCD.print("Objeto fuera");
    LCD.setCursor(0, 1);
    LCD.print("de rango. :(");
}
else {
    LCD.print("Objeto detectado");
    LCD.setCursor(0, 1);
    LCD.print("a ");
    LCD.print(distance);
    LCD.print(" cm. :)");
}
}
...
```

7. Coloque aqui evidencias que considere importantes durante el desarrollo de la actividad.



**CAROLINA DOMINGUEZ CRUZ** 6:48 PM

oki

[https://www.tinkercad.com/things/4idL2eug43U-ingenious-jaagub-bigery/editel?sharecode=geyF0ZPxhhY9hy8j\\_UjxG7NQZQPO2\\_RWIHSANdTPcuE](https://www.tinkercad.com/things/4idL2eug43U-ingenious-jaagub-bigery/editel?sharecode=geyF0ZPxhhY9hy8j_UjxG7NQZQPO2_RWIHSANdTPcuE)



**Tinkercad**

Tinkercad | From mind to design in minutes

Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding.



**Luis Alejandro Sanchez Gallegos** 6:49 PM



entonces creo que van agrupadas las partes 2 con la 3 y la 4 con la 5

o si Cesar prefiere me puedo encargar de la 2, 4, y 5 porque son todas de programación y él de la 3 que son pruebas

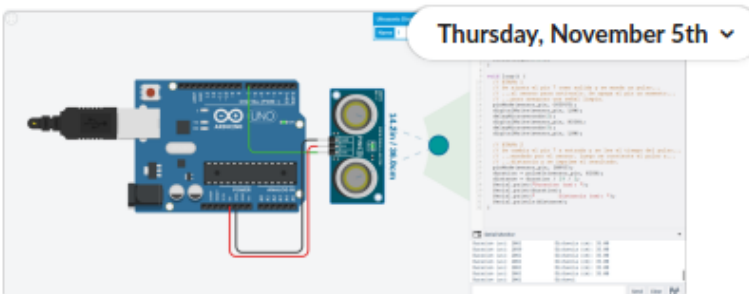
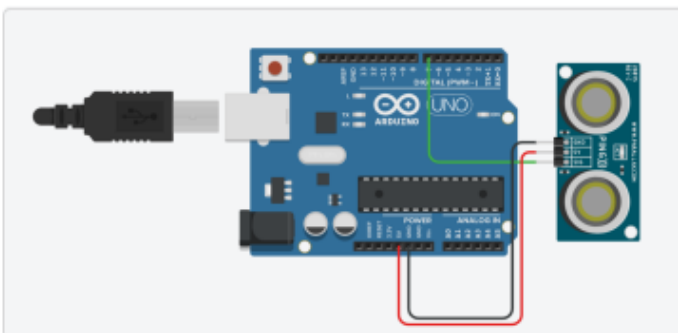
@CAROLINA DOMINGUEZ CRUZ manda mensaje al grupo de messenger cuando esté listo el punto 1

por favor



**CAROLINA DOMINGUEZ CRUZ** 6:54 PM

A3.1\_SensorUltrasonicoParte1.PNG ▾



aquí está el punto 2

me basé en este tutorial oficial de Arduino para el sensor que estamos usando:

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Ping>



**Luis Alejandro Sanchez Gallegos** 8:00 PM

<https://www.arduino.cc/en/Tutorial/LibraryExamples/LiquidCrystalAutoscroll>

encontré esta liga también para controlar el display

me parece que Tinkercad no tiene el control de tipo I2C para el display, pero se puede hacer de otra forma



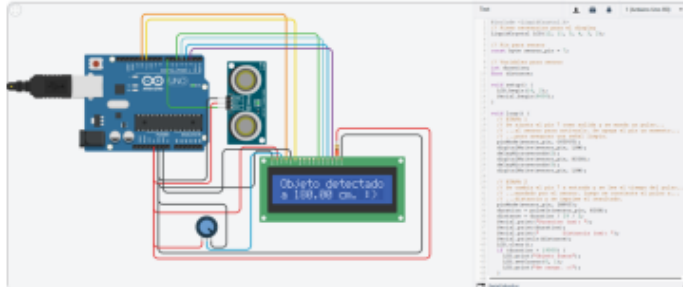
**Luis Alejandro Sanchez Gallegos** 8:15 PM

<https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld>



**Luis Alejandro Sanchez Gallegos** 8:28 PM

image.png ▾



me parece que ya serían los puntos 4 y 5

Luis Alejandro Sanchez Gallegos está presentando

Meet - xylz-asp x Messenger Slack | electro\_modi Circuit design Ingenio Meet - dlu-gmty x alra-gallegos-tes/Sli x

https://www.tinkercad.com/things/4idL2uqf3U-ingenious-jaagub-bigery/edit?sharecode=geyf0ZPhHY9w8j\_UjG7NQZQPO2\_RWHSANdTPuE

Ingenious Jaagub-Bigery

All changes saved

Code Start Simulation Export Share

Test

```

1 // Libreria para uso del display
2 #include <LiquidCrystal.h>
3
4 // Pines de conexión para el display
5 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
6
7 // Pin para sensor
8 const byte sensor_pin = 7;
9
10 // Variables para sensor
11 int duration;
12 float distance;
13
14 void setup() {
15   lcd.begin(16, 2);
16   Serial.begin(9600);
17 }
18
19 void loop() {
20   // ETAPA 1
21   // Se activa el pin 7 como salida y se manda un pulso...
22   // ...el sensor para activarlo; se apaga el pin un momento...
23   // ...para escuchar una señal limpia.
24   pinMode(sensor_pin, OUTPUT);
25   digitalWrite(sensor_pin, LOW);
26   delayMicroseconds(2);
27   digitalWrite(sensor_pin, HIGH);
28   delayMicroseconds(10);
29   digitalWrite(sensor_pin, LOW);
30
31   // ETAPA 2
32   // Se cambia el pin 7 a entrada y se lee el tiempo del pulso...
33   // ...mandado por el sensor. Luego se convierte el pulso a...
34   // ...distancia y se imprime el resultado.
35
36   pinMode(sensor_pin, INPUT);
37   duration = pulseIn(sensor_pin, HIGH);
38   distance = duration * 0.0343 / 2;
39   Serial.print("Duración (us): ");
40   Serial.print(duration);
41   Serial.print("Distancia (cm): ");
42   Serial.print(distance);
43   Serial.println();
44 }

```

Serial Monitor

Duración (us)	Distancia (cm)
2084	46.00
2078	46.00
2093	46.00
2088	46.00
2078	46.00
2091	46.00

Send Clear

CESAR ISAAC SOTO GARCIA

Luis Alejandro Sanchez Gallegos se ha unido

8. Incluya las conclusiones individuales y resultados observados durante el desarrollo de la actividad.

- Carolina Dominguez Cruz:** La primera parte del circuito, es la conexión del sensor al arduino Ultrasonico. Lo utilizamos para detectar objetos en que pasan dentro de su rango. Nuestra segunda parte es colocar el potenciómetro y LDC para poder presentar el mensaje o aviso. Que ambos componentes estaran conectados al arduino. Se codifica la distancia del objeto, duración. También se codifica al LDC un if else, para que en caso de que el objeto este fuera de rango (del área del sensor), muestre el mensaje de "Objeto fuera de rango", y lo contrario cuando este dentro del rango.
- Cesar Isaac Soto García:** En esta actividad aprendí a utilizar el sensor ultrasonico en el simulador, y también el alcance que este sensor tiene, y el como medir la posición del objeto y el como trabaja el sensor, mas que nada el como mandar y recibir datos en un Display y el como se realiza la conexión, también un poco mas en lo personal aprendí a programar un poco mas en el arduino puesto que no conozco mucho de el aun.



- **Luis Alejandro Sanchez Gallegos:** El sensor utilizado en esta práctica no devuelve ninguna información de manera concisa: no tiene una escala predefinida y no utiliza in protocolo de comunicación o algo similar. El sensor simplemente utiliza un pulso de electricidad. Sin embargo, la placa Arduino es capaz de registrar este pulso y transformarlo en información útil para nosotros con un simple cálculo matemático. De igual manera, el display LCD trabaja con registros de memoria individuales a los cuales se debe transmitir información de manera manual. Sin embargo, existe una librería de código para transmitir información a este display de manera sencilla. **En cierta manera la placa Arduino parece ser un puente que comunica a dos diferentes dispositivos que hablan idiomas diferentes, o, vaya, no hablan realmente ningún idioma.**
- 



## Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	10
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	60
Demostración	El alumno se presenta durante la explicación de la funcionalidad de la actividad?	20
Conclusiones	Se incluye una opinión personal de la actividad por cada uno de los integrantes del equipo?	10



[Volver al Índice](#)



[Repositorio en GitHub de Carolina Dominguez Cruz](#)



[Repositorio en GitHub de Cesar Soto García](#)



[Repositorio en GitHub Luis Alejandro Sanchez Gallegos](#)