



### Laboratorio #06

**FECHA Y HORARIO:** Miércoles 07 de abril de 2021, durante el período de laboratorio.

**Instrucciones:** Elabore lo que se le solicita a continuación en galletas de protoboard, haciendo uso de un Raspberry Pi y los componentes electrónicos que considere necesarios.

#### Parte I:

1. Implemente un circuito de detección, usando una pareja de emisor y receptor en su RaspberryPi. Se le recomienda que lo haga funcionar con una fuente de voltaje externa (a 3v) y luego que acople su circuito como entrada a su Raspberry.
2. Codifique una solución en Python dentro de su Raspberry la cual permita la detección de señal proveniente de su pareja emisor-receptor. Su código deberá de recibir este cambio lógico y al momento de recibirlo, enviar una señal de salida hacia un LED el cual se active cuando la señal de emisor-receptor está interrumpida. El LED estará apagado siempre que la pareja emisor-receptor detecte continuidad. Adicional a este cambio físico, en la terminal que esté ejecutando su aplicación dentro de la Raspberry, deberá de mostrar la fecha y hora en la cual se detectó un cambio en el sensor como se muestra:

```
>python3 lab6.py
```

```
Se ha detectado una interrupción de sensor a las 19:25:03.
```

```
Se ha detectado la continuidad del sensor a las 19:25:06.
```

```
Se ha detectado una interrupción de sensor a las 19:26:15.
```

```
Se ha detectado la continuidad del sensor a las 19:26:26.
```

#### Nota adicional y requisito:

1. La salida de la pareja emisor-receptor deberá de estar conectada a un pin del GPIO en Raspberry configurado previamente como entrada y un pin adicional será el que encienda el LED para mostrar si existe o no continuidad en la pareja emisor-receptor.

#### MVP:

Este laboratorio cuenta como mínimo entregable la detección del valor del sensor por la Raspberry.

#### RÚBRICA DE CALIFICACIÓN:

- Posterior a la calificación, deberá de subir al portal del curso en el espacio para "sec03-Lab02" un documento PDF el cual contenga el código realizado en Python para la configuración del dispositivos así como una imagen del funcionamiento alcanzado.

| Elemento   | Ponderación |
|--|-------------|
| Código de detección de señal de entrada en Raspberry                     | 10          |
| Código de respuesta (salida hacia LED) en Raspberry                      | 10          |
| Sensor conectado y funcionando.  | 10          |
| LED conectado y funcionando.   | 10          |
| Impresión en terminal de Raspberry con tiempos de interrupción de señal. | 60          |

## **Laboratorio #07**

**FECHA Y HORARIO:** Miércoles 07 de abril de 2021, durante el período de laboratorio.

**Instrucciones:** Elabore lo que se le solicita a continuación en galletas de protoboard, haciendo uso de un Raspberry Pi y los componentes electrónicos que considere necesarios.

### **Parte I:**

1. Implemente un circuito de detección, usando una pareja de emisor y receptor en su RaspberryPi. Se le recomienda que lo haga funcionar con una fuente de voltaje externa (a 3v) y luego que acople su circuito como entrada a su Raspberry.
2. Codifique una solución en Python dentro de su Raspberry la cual permita la detección de señal proveniente de su pareja emisor-receptor. Su código deberá de recibir este cambio lógico y al momento de recibirlo, enviar una señal de salida hacia un LED el cual se active cuando la señal de emisor-receptor está interrumpida. El LED estará apagado siempre que la pareja emisor-receptor detecte continuidad. Adicional a este cambio físico, en la terminal que esté ejecutando su aplicación dentro de la Raspberry, deberá de mostrar la fecha y hora en la cual se detectó un cambio en el sensor como se muestra:

```
>python3 lab6.py
```

```
Se ha detectado una interrupción de sensor a las 19:25:03.
```

```
Se ha detectado la continuidad del sensor a las 19:25:06.
```

```
Se ha detectado una interrupción de sensor a las 19:26:15.
```

```
Se ha detectado la continuidad del sensor a las 19:26:26.
```

3. Cree una tabla, objeto o similar en una base de datos la cual se encuentre fuera de la Raspberry. Esta base de datos puede ser SQL o NoSQL y de cualquier manejador. A su vez, esta base de datos puede estar alojada en su laptop o bien (preferiblemente) en la nube (como un cloud service) Si tiene dudas sobre ventajas y desventajas sobre un tipo u otro, se puede tratar durante el período de laboratorio.
4. Haga que su dispositivo Raspberry sea capaz de comunicarse con esta base de datos (pruebas de ping y demás)
5. Haga un insert en su base de datos con los campos de: Fecha y hora, tipo de actividad (interrupción y continuidad)
6. Muestre en su computadora u otro dispositivo los registros que se han ido ingresando en su base de datos.

### **Nota adicional y requisito:**

7. La salida de la pareja emisor-receptor deberá de estar conectada a un pin del GPIO en Raspberry configurado previamente como entrada y un pin adicional será el que encienda el LED para mostrar si existe o no continuidad en la pareja emisor-receptor.

**MVP:**

Este laboratorio cuenta como mínimo entregable la comunicación entre su Raspberry y su base de datos.

**RÚBRICA DE CALIFICACIÓN:**

- Posterior a la calificación, deberá de subir al portal del curso en el espacio para “sec03-Lab03” un documento PDF el cual contenga el código realizado en Python para la configuración del dispositivos así como una imagen del funcionamiento alcanzado. Como último punto, screenshot de la base de datos con los valores insertados exitosamente.

| Elemento  | Ponderación |
|---|-------------|
| Código de detección de señal de entrada en Raspberry                                      | 2           |
| Código de respuesta (salida hacia LED) en Raspberry                                       | 2           |
| Sensor conectado y funcionando.   | 3           |
| LED conectado y funcionando.  | 3           |
| Impresión en terminal de Raspberry con tiempos de interrupción de señal.                  | 5           |
| Comunicación entre Raspberry y base de datos  | 20          |
| Insert de los 2 datos hacia un registro en la base de datos (fecha/hora y tipo de evento) | 40          |
| Visualización de eventos almacenados en la base de datos (select a la tabla u objeto).    | 25          |