

# JAVA Y ARDUINO

Tutora: Angela Jazmín Miranda Flores



# COMPONENTES A UTILIZAR

Ide de netbeans

Jdk v8 de java oracle

Librería de panamihiteck

Ide arduino

Arduino uno

1 sensor ultrasónico

1 led

1 resistencia de 330 ohm

1 protoboard

jumpers





# ¿CÓMO INTEGRAMOS ARDUINO CON JAVA?



Tutora: Angela Jazmín Miranda Flores

# LIBRERÍA TROUBLESHOOT

## ➤ V1.1.0

- ✓ Contiene la librería RXTX con sus respectivos métodos (arduinoRX, arduinoTX, arduinoRXTX, etc).
- ✓ Debe instalarse manualmente el archivo **RXTXcomm.jar**.

## ➤ V2.2.0

- ✓ A partir de esta versión no es necesario el uso de la librería RXTX, solo se la debe importar dentro del proyecto en NetBeans.
- ✓ Se agrega soporte para la lectura de múltiples mensajes enviados desde arduino.

## ➤ V 2.6.0

- ✓ Luego de esta versión la librería fue nombrada PanamaHiteck\_Arduino.

# LIBRERÍA PANAMAHITEK\_ARDUINO

## ➤ V2.7.0

- ✓ No es necesaria la instalación de los archivos dll. (En Windows ya se almacenan automáticamente).

## ➤ V2.8.0

- ✓ Se ha reemplazado la librería RXTX por la de **Java Simple Serial Connector**.
- ✓ es compatible con Linux, Mac, Solaris y Windows.

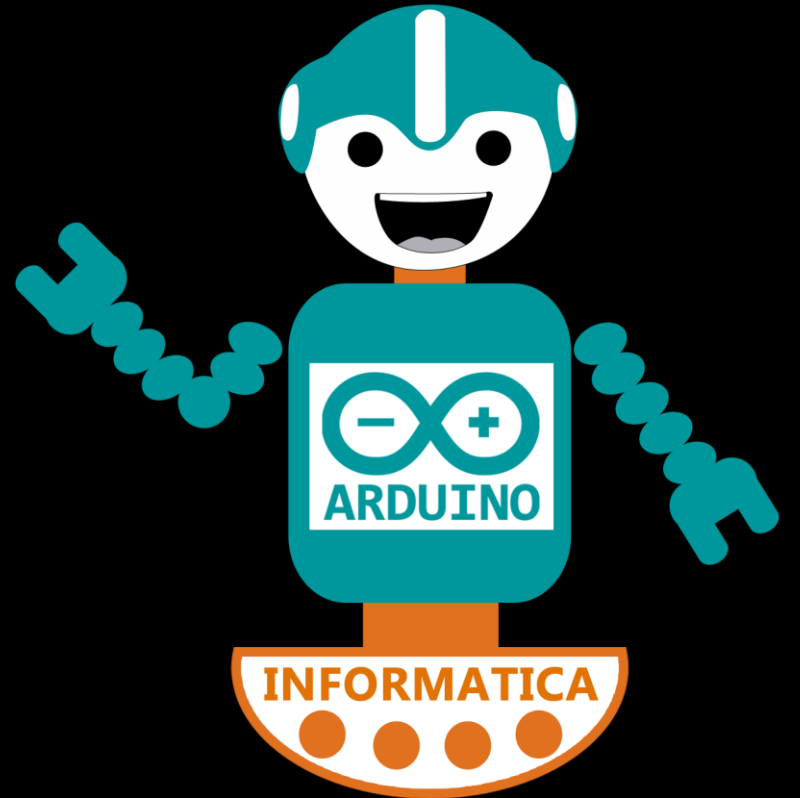
## ➤ V2.8.2

- ✓ Implementa una clase para poder almacenar datos en archivos .xls.

## ➤ V2.8.3

- ✓ Última versión donde se corrigen errores de la V2.8.2.

# ¿CÓMO SE CONECTA ARDUINO CON JAVA?



Tutora: Angela Jazmín Miranda Flores

# CONEXIÓN SERIAL

Para poder realizar la comunicación entre arduino y java es necesario utilizar la comunicación serial.

Es por eso que a la hora de mostrar mensajes en java arduino debe mostrarlos como una cadena por el monitor serial.

Java para poder interactuar con arduino debe mandar caracteres para que este lo pueda interpretar y realizar una acción determinada.

Existen tres formas de comunicarse entre arduino y java.

-ArduinoRX

-ArduinoTX

-ArduninoRXTX

# NETBEANS

## Descargar e instalar NetBeans

Seguro <https://netbeans.org/downloads/>

HOME / Download

### NetBeans IDE 8.2 Download

8.1 | 8.2 | Development | Archive

Email address (optional):

Subscribe to newsletters: ☒ Monthly ☐ Weekly

☒ NetBeans can contact me at this address

IDE Language:  Platform:

Note: Greyed out technologies are not supported for this platform.

#### NetBeans IDE Download Bundles

| Supported technologies *                   | Java SE | Java EE | HTML5/JavaScript | PHP | C/C++ | All |
|--|---------|---------|------------------|-----|-------|-----|
| NetBeans Platform SDK                      | •       | •       |                  |     |       | •   |
| Java SE                                    | •       | •       |                  |     |       | •   |
| Java FX                                    | •       | •       |                  |     |       | •   |
| Java EE                                    |         | •       |                  |     |       | •   |
| Java ME                                    |         |         |                  |     |       | •   |
| HTML5/JavaScript                           |         | •       | •                | •   |       | •   |
| PHP  |         |         | •                | •   |       | •   |
| C/C++                                      |         |         |                  |     | •     | •   |
| Groovy                                     |         |         |                  |     |       | •   |
| Java Card™ 3 Connected                     |         |         |                  |     |       | •   |
| Bundled servers                            |         |         |                  |     |       |     |
| GlassFish Server Open Source Edition 4.1.1 |         | •       |                  |     |       | •   |
| Apache Tomcat 8.0.27                       |         | •       |                  |     |       | •   |

Download Download Download x86 Download x86 Download x86 Download

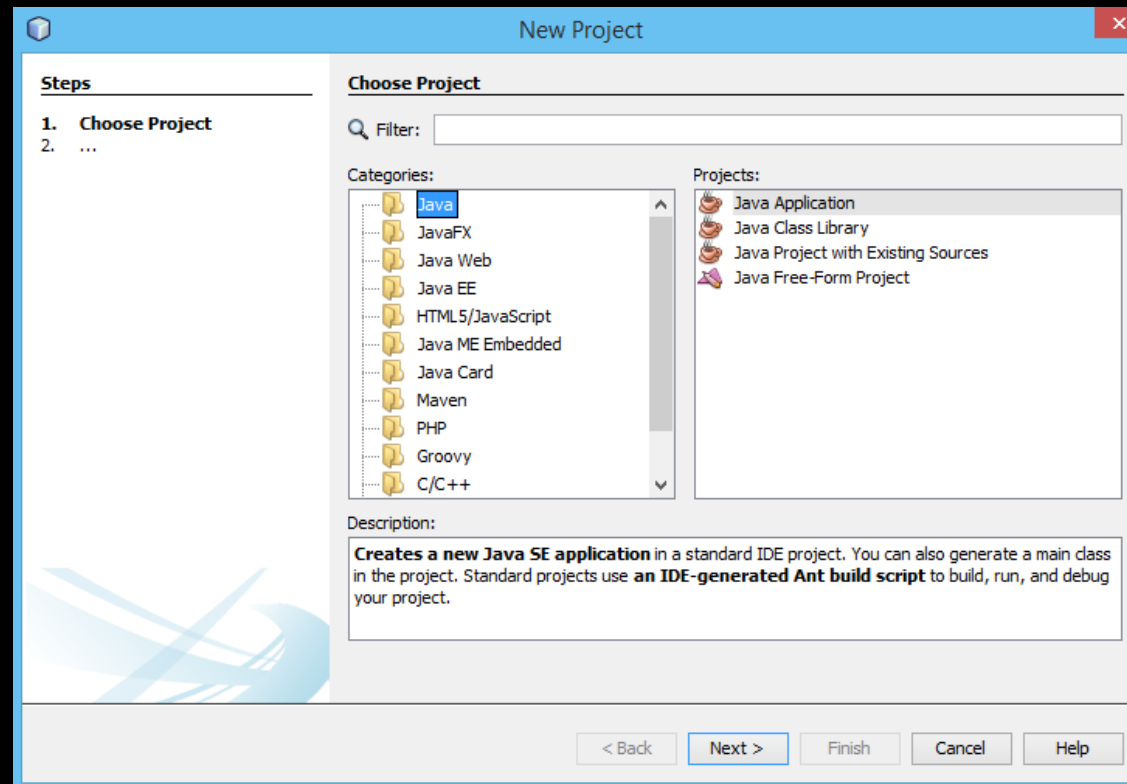
Download x64 Download x64 Download x64

Free, 95 MB Free, 197 MB Free, 108 - 112 MB Free, 108 - 112 MB Free, 107 - 110 MB Free, 221 MB



# NETBEANS

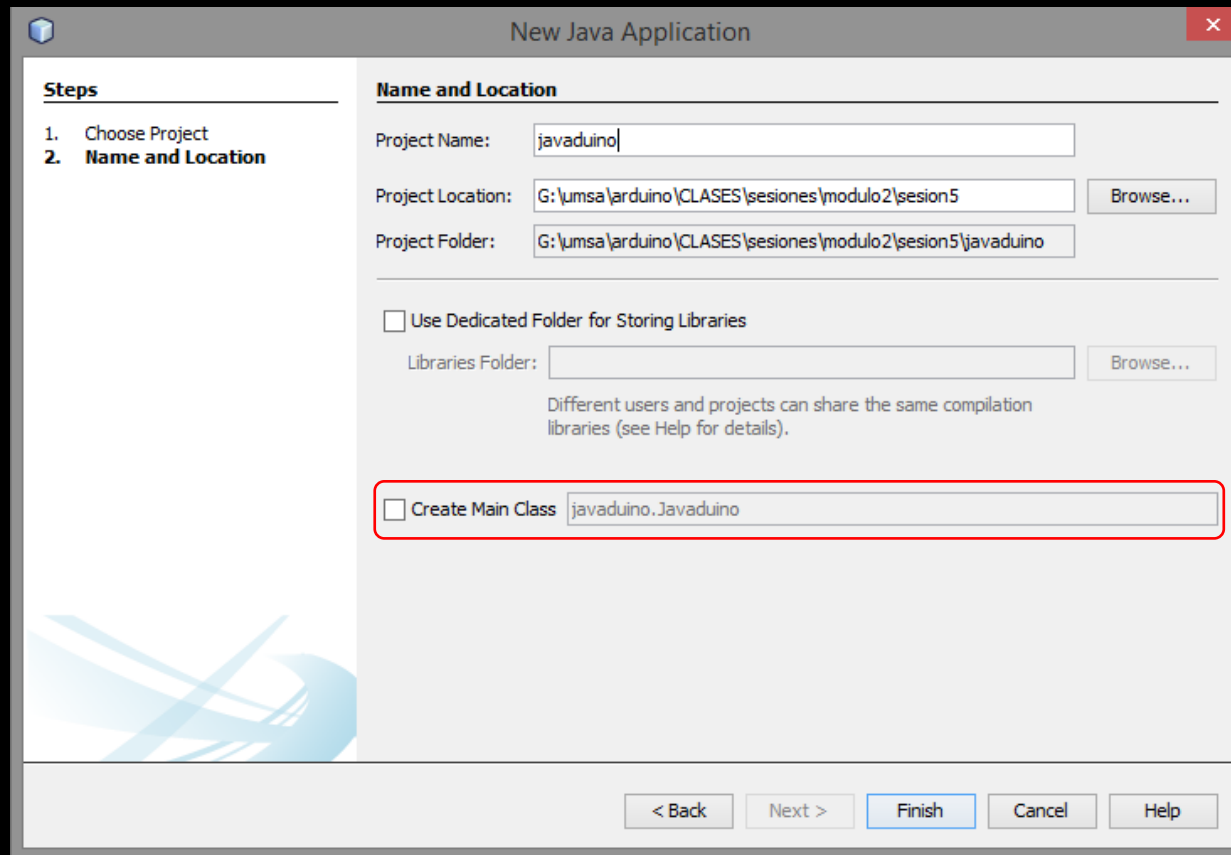
Creamos un nuevo proyecto



Tutora: Angela Jazmín Miranda Flores

# NETBEANS

Le colocamos un nombre a nuestro proyecto



**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name:

Project Location:  Browse...

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:  Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > Finish Cancel Help

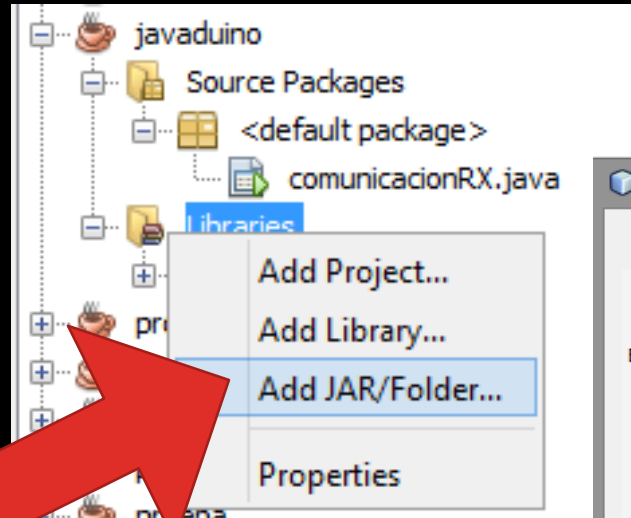
NOTA: No crear el proyecto con la clase principal.



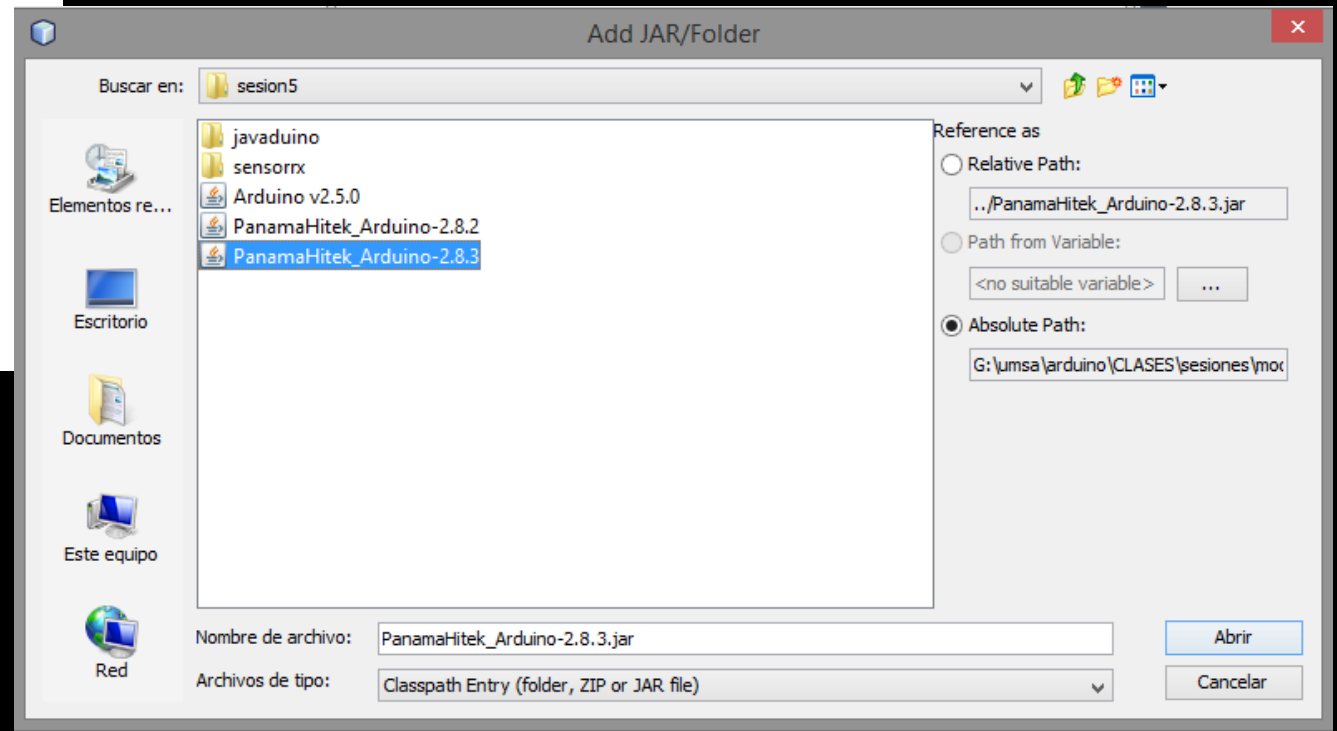
Tutora: Angela Jazmín Miranda Flores

# NETBEANS

Agregamos la librería de panama\_Hiteck -2.8.3



NOTA: Se sugiere que la librería se encuentre dentro la misma carpeta de nuestro proyecto.



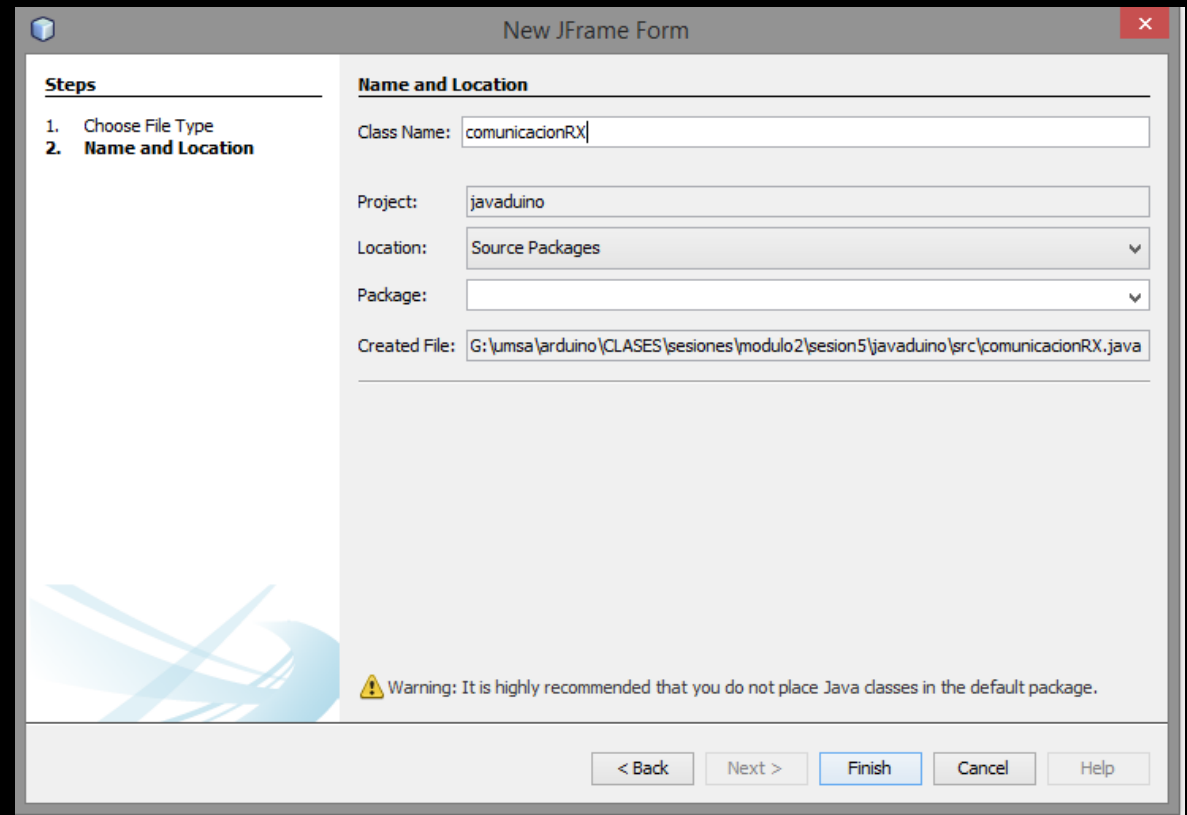
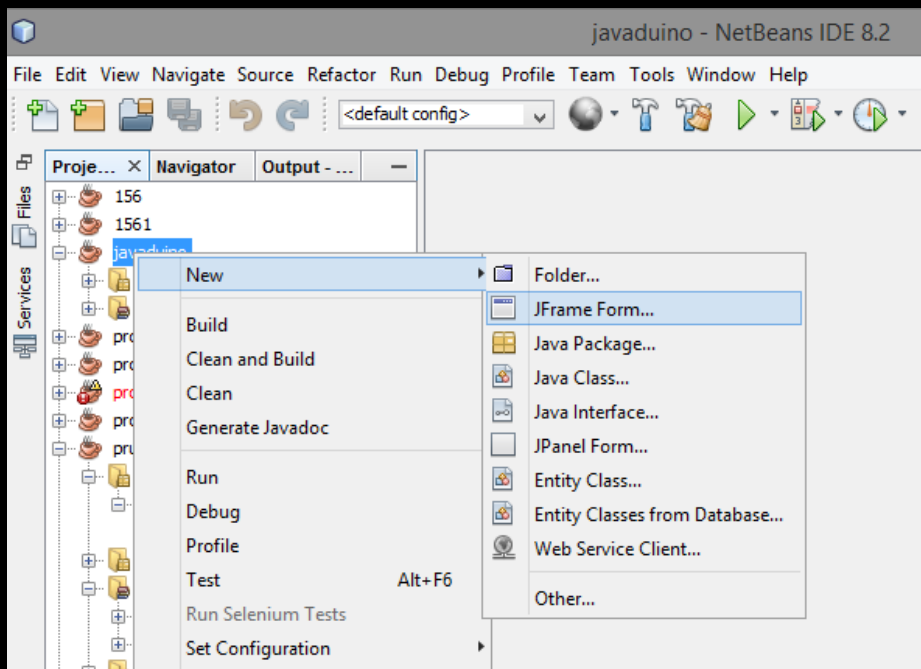
# COMUNICACIÓN RX

- ✓ `SerialPortEventListener();` => enviar datos desde arduino.
- ✓ `SerialEvent();` => se ejecuta cada vez que se recibe un dato en el puerto serie.
- ✓ `MessageAvailable();` => devuelve true cuando se termina de recibir el mensaje desde arduino.
- ✓ `PrintMessage();` => mensaje recibido como String desde arduino.
- ✓ `arduinoRX("puerto com de nuestro arduino", "velocidad de comunicación del puerto serial", "evento que escucha nuestro puerto serial");`



# COMUNICACIÓN RX

Añadimos un JFrame Form al proyecto, lo nombraremos “comunicacionRX”.



# COMUNICACIÓN RX

Para el JFrame utilizaremos 3 label, las cuales solo debemos arrastrar de la paleta de componentes.



# COMUNICACIÓN RX

```
import com.panamahitek.ArduinoException;
import com.panamahitek.PanamaHitek_Arduino;
import java.util.logging.Level;
import java.util.logging.Logger;
import jssc.SerialPortEvent;
import jssc.SerialPortEventListener;
import jssc.SerialPortException;
public class comunicacionRX extends javax.swing.JFrame {
    //variable de tipo cadena que almacenara el valor enviado desde el sensor
    String msensor="";
    //llamamos a nuestra libreria
    PanamaHitek_Arduino sensor=new PanamaHitek_Arduino();
```

# COMUNICACIÓN RX

```
//escucha lo que se manda a traves del puerto serial
SerialPortEventListener lista=new SerialPortEventListener() {
    public void serialEvent(SerialPortEvent spe){
        try {
            if(sensor.isMessageAvailable()){
                //se almacena el valor que envia arduino
                msensor=sensor.printMessage();
                muestra(msensor);
            }
        } catch (SerialPortException ex) {
            Logger.getLogger(comunicacionRX.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(comunicacionRX.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
};

public void muestra(String cadena){
    this.jLabel2.setText(cadena);
}
```

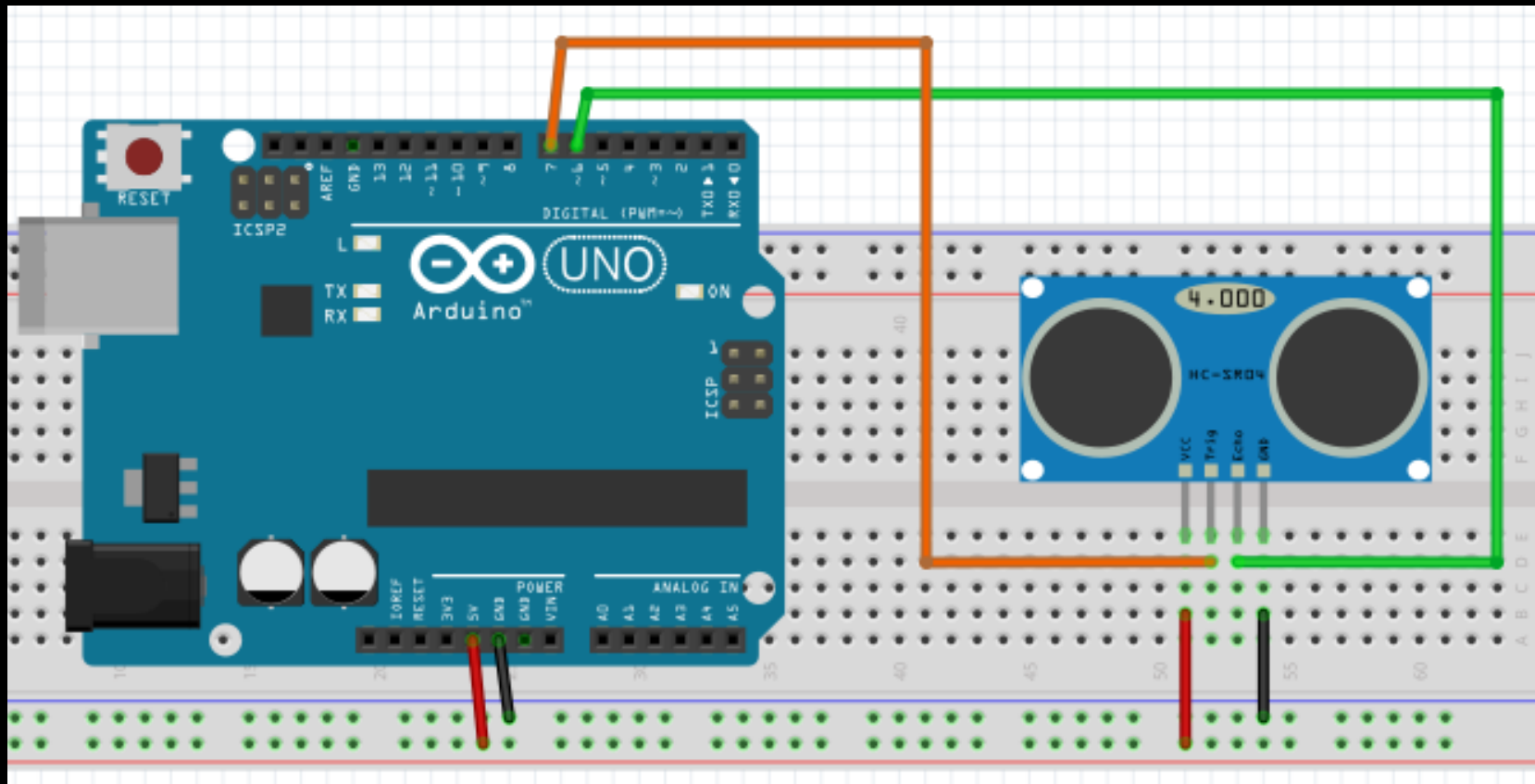


# COMUNICACIÓN RX

En la clase donde se inicializa nuestro jframe añadimos las siguientes líneas de código.

```
public comunicacionRX() {  
    try{  
        initComponents();  
        //arduinoRX("PUERTO COM EN EL QUE SE ENCUENTRA NUESTRO ARDUINO",  
        // VELOCIDAD DE COMUNICACION DEL PUERTO SERIAL, VARIABLE ASIGNADA QUE ESCUCHA EL EVENTO DEL PUERTO SERIAL);  
        sensor.arduinoRX("COM7", 9600, lista);  
  
    } catch (ArduinoException ex) {  
        Logger.getLogger(comunicacionRX.class.getName()).log(Level.SEVERE, null, ex);  
    } catch (SerialPortException ex) {  
        Logger.getLogger(comunicacionRX.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

# ESQUEMA DE CONEXIÓN



# CÓDIGO EN ARDUINO

```
//AUTORA:Angela Miranda Flores
//CONTACTO angiejazminmiranda@gmail.com
#define echo 6
#define trig 7
long duracion,distancia;
void setup() {
  pinMode(echo,INPUT);
  pinMode(trig,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  String cad="";
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);    // genera el pulso de trigger por 10ms
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duracion = pulseIn(echo, HIGH);
  distancia = (duracion/2) / 29;
  cad.concat(distancia);
  cad.concat(" cm");
  Serial.println(cad);
  delay(1000);
}
```

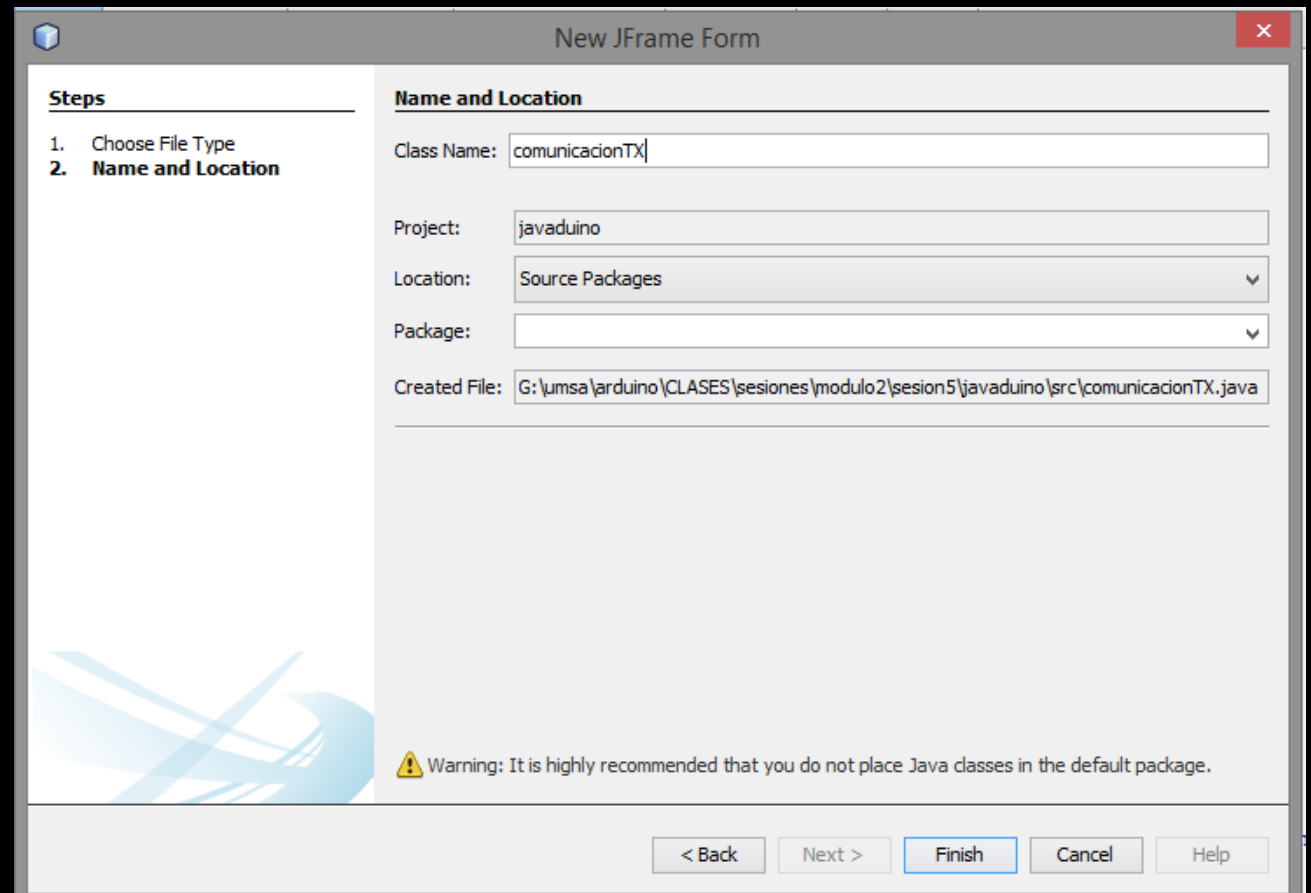
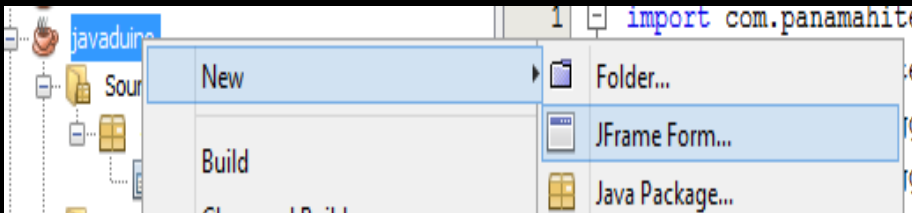
# COMUNICACIÓN TX

- ✓ `arduinoTX("puerto com en el que esta nuestro arduino", "velocidad de comunicación del puerto serial")`
- ✓ `SendData ("valor");` => envía un carácter a arduino mediante el puerto serial para que este lo interprete y ejecute una determinada acción.



# COMUNICACIÓN TX

Añadimos un JFrame al cual nombraremos “comunicacionTX”.

A screenshot of the 'New JFrame Form' dialog box. The dialog has a 'Steps' section on the left with '1. Choose File Type' and '2. Name and Location'. The 'Name and Location' section on the right contains the following fields: 'Class Name' with the text 'comunicacionTX', 'Project' with 'javaduino', 'Location' with 'Source Packages', and 'Package' which is empty. The 'Created File' field shows the path 'G:\umsa\arduino\CLASES\sесiones\modulo2\sesion5\javaduino\src\comunicacionTX.java'. At the bottom, there is a warning message: 'Warning: It is highly recommended that you do not place Java classes in the default package.' and buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Tutora: Angela Jazmín Miranda Flores

# COMUNICACIÓN TX

Para el JFrame utilizaremos 1 label y 2 botones, los cuales solo debemos arrastrar de la paleta de componentes.



# COMUNICACIÓN TX

```
import com.panamahitek.ArduinoException;
import com.panamahitek.PanamaHitek_Arduino;
import java.util.logging.Level;
import java.util.logging.Logger;
import jssc.SerialPortException;

public class comunicacionTX extends javax.swing.JFrame {
    // llamamos a nuestra libreria
    PanamaHitek_Arduino arduino=new PanamaHitek_Arduino();
    public comunicacionTX() {
        initComponents();
        try{
            arduino.arduinoTX("COM7", 9600);
        } catch (ArduinoException ex) {
            Logger.getLogger(comunicacionTX.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

# COMUNICACIÓN TX

En nuestros botones añadimos las siguientes líneas de código.

Botón 1

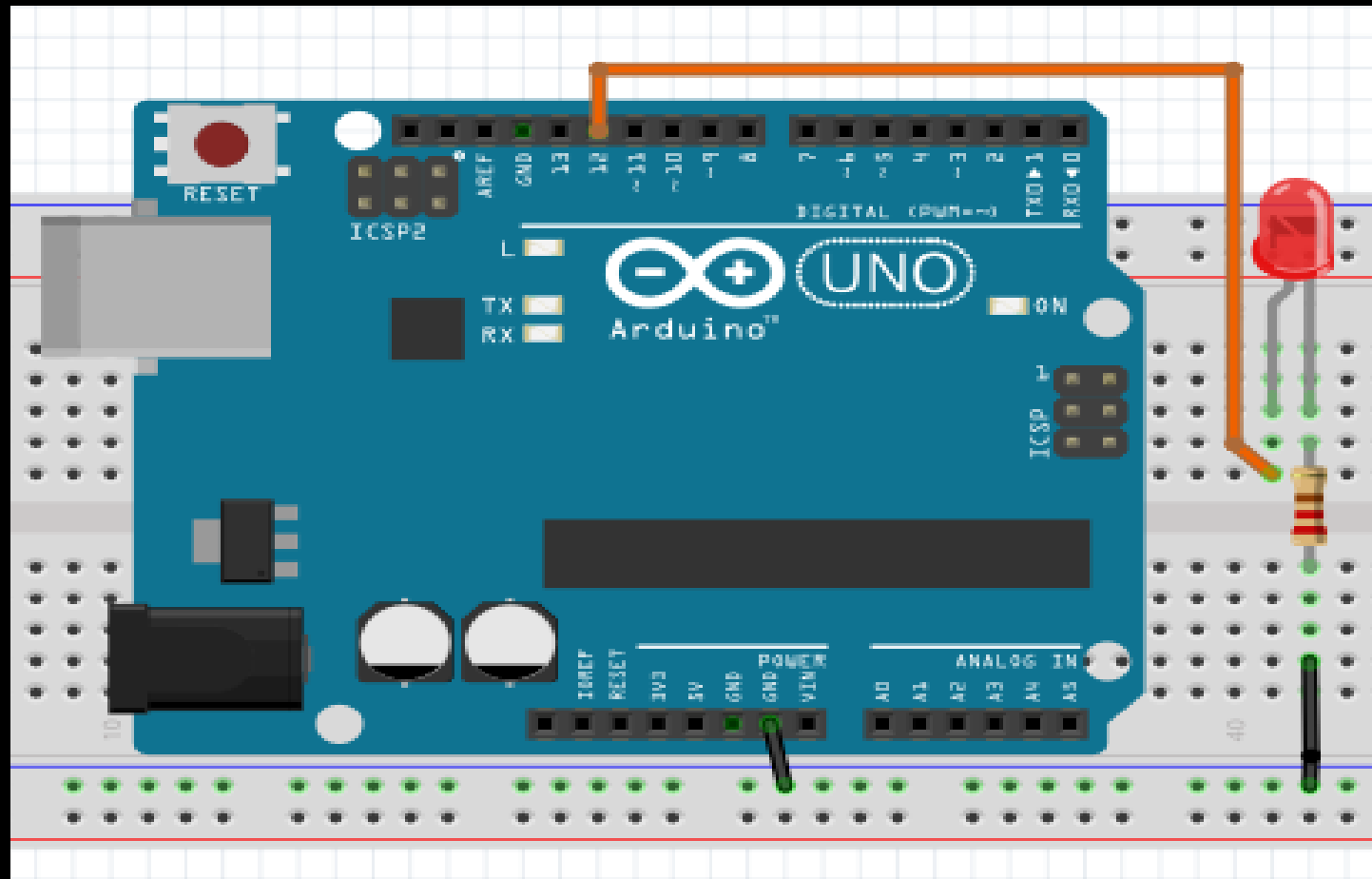
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        //mandamos el mismo caracter con el que configuramos nuestro arduino  
        arduino.sendData("h");  
    } catch (ArduinoException ex) {  
        Logger.getLogger(comunicacionTX.class.getName()).log(Level.SEVERE, null, ex);  
    } catch (SerialPortException ex) {  
        Logger.getLogger(comunicacionTX.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

Botón 2

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        //mandamos el mismo caracter con el que configuramos nuestro arduino  
        arduino.sendData("l");  
    } catch (ArduinoException ex) {  
        Logger.getLogger(comunicacionTX.class.getName()).log(Level.SEVERE, null, ex);  
    } catch (SerialPortException ex) {  
        Logger.getLogger(comunicacionTX.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```



# ESQUEMA DE CONEXIÓN



Tutora: Angela Jazmín Miranda Flores

# CÓDIGO EN ARDUINO

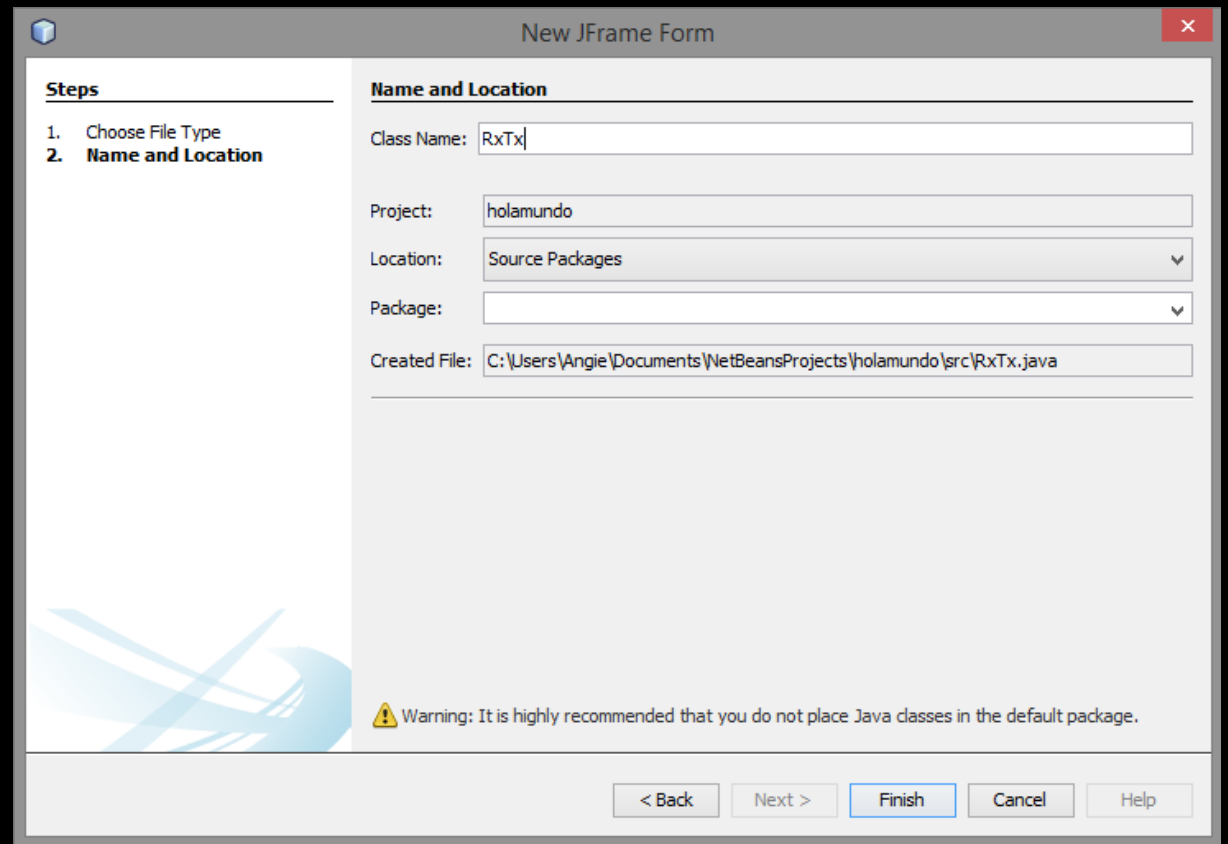
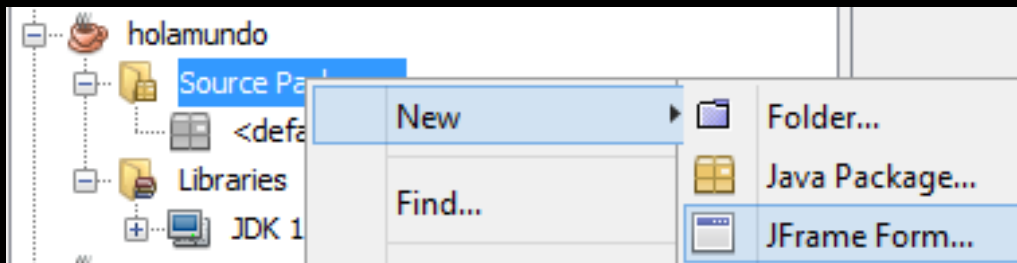
```
//AUTORA:Angela Miranda Flores
//CONTACTO angiejazminmiranda@gmail.com
char estado=' ';
void setup() {
    Serial.begin(9600);
    pinMode(12,OUTPUT);
}
void loop() {
    if(Serial.available()>0){
        estado = Serial.read();
    }
    if(estado== 'h' ){
        digitalWrite(12,HIGH);
    }
    if(estado== 'l' ){
        digitalWrite(12,LOW);
    }
}
```

# COMUNICACIÓN RXTX

- ✓ `arduinoRXTX` ("puerto com en el que se encuentra arduino", "velocidad de conexión del puerto serial", "evento que escucha nuestro puerto serial");
- ✓ `SerialPortEventListener()`; => enviar datos desde arduino.
- ✓ `SerialEvent()`; => se ejecuta cada vez que se recibe un dato en el puerto serie.
- ✓ `MessageAvailable()`; => devuelve true cuando se termina de recibir el mensaje desde arduino.
- ✓ `PrintMessage()`; => mensaje recibido como String desde arduino.
- ✓ `SendData ("valor")`; => envía un carácter a arduino mediante el puerto serial para que este lo interprete y ejecute una determinada acción.

# NETBEANS

Agregamos un jframe a nuestro proyecto, como nombre le podremos RxTx



Tutora: Angela Jazmín Miranda Flores

# EJERCICIO DE APLICACIÓN

Realizar la conexión RXTX entre java y arduino, EL JFrame debe quedar de la siguiente manera:



Tutora: Angela Jazmín Miranda Flores



# ESQUEMA DE CONEXIÓN

