**HidroSmart**

**MANUAL TÉCNICO  
  
MATERIA**Programación de Sistemas Telemáticos  
 **TÍTULO**Aplicación y control automatizado de riego  
 **PROFESORA**Msig. Adriana E. Collaguazo Jaramillo  
 **INTEGRANTES**Maria del Cisne Feijoo JaramilloJeremy Carlos León AndradePierina Paulette Pereira CampoverdeYovana Paola Zambrano Ruiz

**Contenido**

[Resumen Ejecutivo 5](#_Toc17407709)

[Introducción 5](#_Toc17407710)

[Descripción del problema 5](#_Toc17407711)

[Objetivos 6](#_Toc17407712)

[Objetivo General: 6](#_Toc17407713)

[Objetivos Específicos: 6](#_Toc17407714)

[Solución Planteada 6](#_Toc17407715)

[Componentes de Software 6](#_Toc17407716)

[Componentes de Hardware 6](#_Toc17407717)

[Diagrama del circuito 7](#_Toc17407718)

[Diagrama Entidad-Relación 8](#_Toc17407720)

[Diagrama UML 8](#_Toc17407721)

[Diagrama de Despliegue 9](#_Toc17407722)

[Set-Up y Código Fuente 9](#_Toc17407723)

[Implementación del Software 10](#_Toc17407724)

[1. Aplicación 10](#_Toc17407725)

[1.1.1. MainActivity 10](#_Toc17407726)

[1.2. Crear Cuenta 13](#_Toc17407727)

[1.3. Inicio 18](#_Toc17407732)

[1.4. Consultar 22](#_Toc17407733)

[1.5. Consumo por Mes 26](#_Toc17407734)

[1.6. Consulta por día 29](#_Toc17407737)

[1.7. Ahorro 33](#_Toc17407738)

[2. Código Arduino 36](#_Toc17407739)

[3. PhpMyAdmin 41](#_Toc17407741)

[Implementación física 48](#_Toc17407742)

[1. Sensor de humedad del suelo 48](#_Toc17407743)

[2. Sensor de humedad y temperatura del aire 49](#_Toc17407744)

[3. Relé 50](#_Toc17407745)

[Implementación final del circuito 51](#_Toc17407746)

[Implementación de maqueta para simular el sistema de riego 52](#_Toc17407747)

[Diagrama de Gantt 53](#_Toc17407748)

[Análisis de presupuesto 53](#_Toc17407749)

[Conclusiones 54](#_Toc17407750)

[Referencias bibliográficas 55](#_Toc17407751)

[Anexos: 55](#_Toc17407752)

# 

# Resumen Ejecutivo

Año tras año, el desperdicio de agua que se produce en la industria agrícola aumenta debido al uso de sistemas ineficientes de riego, por lo cual en este proyecto se ha diseñado un sistema automatizado que garantice el riego adecuado de cultivos, en este caso de flores “Chavelitas”. Este sistema se encuentra conformado por distintos dispositivos como Arduino, Raspberry Pi, bomba y sensores de humedad del suelo y temperatura del aire, las cuales nos permitirá conocer cuáles son las condiciones del ambiente y del suelo, para activar y desactivar la bomba, debido a que la bomba solo deberá ser activada cuando se registre una temperatura del aire menor a 25° , y la humedad del suelo sea menor de 40%, es decir cuando no haya sol ya que los estomas de la planta a pleno sol están cerrados para no deshidratare, con el agua de riego se abren y la planta puede deshidratarse. Además este sistema podrá ser controlado por medio de una aplicación creada con la herramienta Android Studio, la cual permitirá observar en tiempo real datos, estadísticas y demás información sobre el sistema, mediante el uso de una Base de datos, que almacenará los datos recopilados por los sensores. La implementación de este proyecto tiene un costo total de $106.60.

IntroducciónEn la actualidad las nuevas tendencias tecnológicas son un factor fundamental en el desarrollo de distintas áreas, una de ellas es la agricultura, ya que permite mejorar no solo los procesos de preparación y cuidado de los cultivos, sino también en su cosecha y distribución, permitiendo así brindar productos de mejor calidad, y asegurando la optimización de recursos naturales. Uno de los grandes problemas que se pueden evidenciar en la agricultura tradicional es el gran desperdicio de agua que se produce, se estima que aproximadamente el 70% del agua extraída es utilizada en la industria agrícola, y que de esta más del 50% se desperdicia debido al uso de sistemas de riego ineficientes. Este proyecto tiene como finalidad evitar el desperdicio de agua producido al utilizar sistemas de riego tradicionales, para lo cual plantea la implementación de un sistema de riego automatizado que permita asegurar el suministro de agua solo cuando y en la cantidad que el cultivo lo necesite.

# Descripción del problema

La escasez de agua es cada vez más eminente, actualmente 3600 de agua dulce son extraídos al año para el consumo humano de los cuales el 70% de toda la extracción es consumida por el área de la agricultura, y se estima que más de la mitad de ese suministro de agua se desperdicia al momento del riego de cultivos, debido al uso de sistemas de riego ineficientes, que además de producir desperdicio de agua, producen daños en los cultivos y en la tierra.

Un riego adecuado es fundamental no solo para mantener en buenas condiciones el cultivo, sino también porque permite optimizar recursos naturales y con ellos asegurar el cuidado del ambiente, debido a que esta nos permite poder brindar un riego eficiente, según las necesidades de cada cultivo y las condiciones ambientales en las que se encuentran, lo que con lleva a un ahorro significativo de recursos, sobre todo el agua.

# Objetivos

## Objetivo General:

* Garantizar el riego adecuado a una plantación de flores “Chavelitas”, para evitar desperdicio de agua, a través de un sistema de control automático, cumpliendo los estándares recomendados para el cuidado de cultivos.

## Objetivos Específicos:

* Aprovechar al máximo los caudales de agua disponible, para lograr un ahorro significativo de recursos en comparación con un sistema tradicional.
* Utilizar sistemas de riego con programación de control automático para asegurar el suministro de agua cuando y en la cantidad que la planta necesite.
* Crear una aplicación móvil que permita al usuario poder acceder en tiempo real a la información del sistema de riego

# Solución Planteada

Se propone implementar un sistema de control automatizado de riego para plantaciones de flores, que mediante el uso de sensores se pueda detectar las condiciones del ambiente como la humedad, temperatura y luz solar, para poder determinar cuál es el mejor momento para realizar el riego, esto además de asegurar un mejor cuidado del cultivo y de la tierra, permite aprovechar al máximo los caudales de agua disponibles y un ahorro significativo de recursos.

Para una mayor facilidad y comodidad de los usuarios, este sistema podrá ser controlado desde un aplicativo móvil compatible con sistemas operativos Android, que nos permitirá controlar el encendido y apagado del sistema, y además nos permitirá observar en tiempo real el estado del sistema, así como un historial, mediante gráficas, tablas que permitirán observar los datos de una forma más dinámica y organizada.

# Componentes de Software

* Android Studio 3.4.2
* MySQL 10.1.8-MariaDB
* Raspbian 4.19

# Componentes de Hardware

* 1 Raspberry PI 3 modelo B++
* 1 módulo de wifi ESP8266
* 1 sensor de humedad del suelo Yl-69
* 1 sensor de temperatura y humedad relativa en el aire DHT11
* 1 mini Bomba De Agua De Diafragma 6-12v R385 Acuario
* 1 relé
* 1 fotorresistencia LDR
* 15 jumpers
* 1 Fuente de poder

# Diagrama del circuito

# 

Figura 1. Implementación final del circuito para el sistema de riego

# Diagrama Entidad-Relación

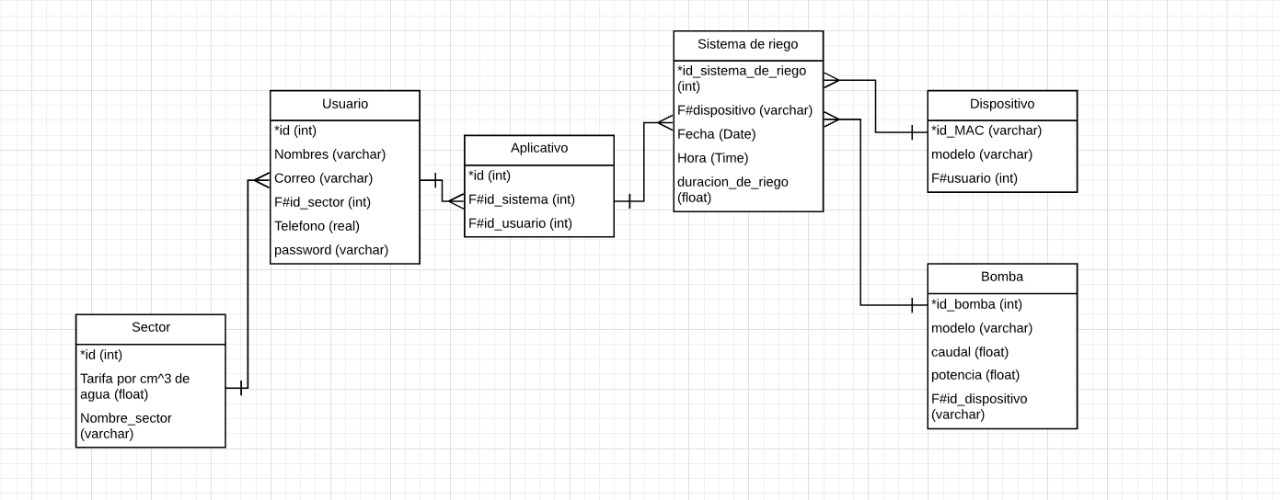


Figura 2. Diagrama Entidad-Relación

# 

# Diagrama UML

Figura 3. Diagrama UML

# Diagrama de Despliegue

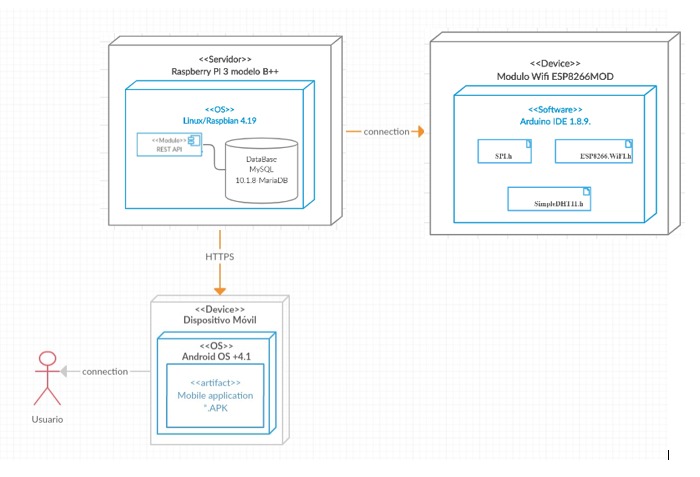


Figura 4.-Diagrama de Despliegue

# Set-Up y Código Fuente

# Implementación del Software

## Aplicación

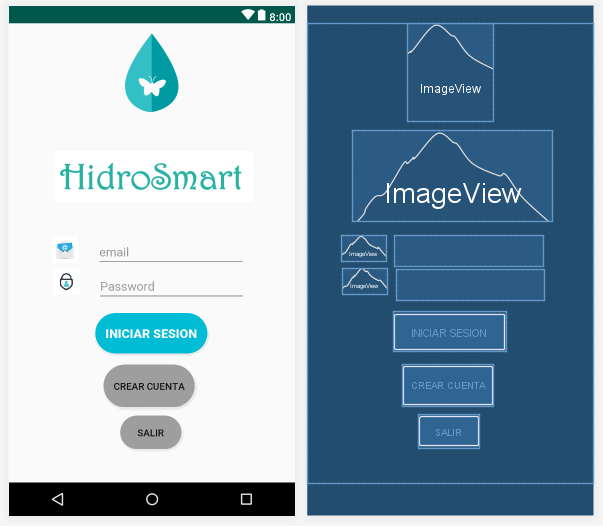
### MainActivity

En el código del archivo MainActivity.java, se importaron las librerías correspondientes y se crearon dos funciones, que están enlazadas a los botones correspondientes del archivo XML, la función Crear Cuenta dirige al usuario a una activity nueva donde podrá ingresar sus datos para crear una cuenta nueva, en el caso de tener una cuenta, con la función Tabla1, el usuario será dirigido a una activity que contiene los datos de su sistema de riego.

**package** com.example.pst3\_1t2019\_aplicacionderiegoycontrolautomatizado;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
 **public void** tabla1(View view){  
 Intent i = **new** Intent(**this**,consumo\_por\_mes.**class**);  
 startActivity(i);  
 }  
 **public void** crearCuenta(View view){  
 Intent i = **new** Intent(**this**,CrearCuenta.**class**);  
 startActivity(i);  
  
 }  
}

* + 1. **Diseño XML:**

Para la interfaz gráfica, en la sección Design, se agregó el logo de la aplicación, así también 2 TextView , 2 EditText y 2 botones, los botones ayudaran al usuario a entrar al sistema, ya sea ingresando sus credenciales en los EditText o en el caso de no tener cuenta, en el botón Crear cuenta, el usuario será dirigido a una activity donde podrá crear una.

* + 1. **Código XML**

<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorAccent"  
 tools:context=".MainActivity"**>  
  
 <**TextView  
 android:id="@+id/textView"  
 style="@android:style/Widget.Holo.Light.TextView"  
 android:layout\_width="174dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="108dp"  
 android:fontFamily="cursive"  
 android:text="@string/hidrosmart"**

**android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="36sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView"** />  **android:id="@+id/imageView"  
 android:layout\_width="124dp"  
 android:layout\_height="142dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="136dp"  
 android:contentDescription="@string/logo"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/logo"** />  
  
 <**EditText  
 android:id="@+id/editText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="96dp"  
 android:ems="10"  
 android:inputType="textEmailAddress"  
 android:text="@string/correo\_electr\_nico"  
 android:textColor="#BC6B5D5D"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView"  
 android:autofillHints=""** />  
  
 <**EditText  
 android:id="@+id/editText3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="96dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:text="@string/contrase\_a"  
 android:textColor="#BE6B5D5D"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText"  
 android:autofillHints=""** />

<**Button  
 android="@+id/button4"  
 android:layout\_width="161dp"  
 android:layout\_height="58dp"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="120dp"  
 android:background="#00BCD4"  
 android:onClick="tabla1"  
 android:text="@string/iniciar\_sesi\_n"  
 android:textColor="#F3F6F7"  
 android:textColorHighlight="#00BCD4"  
 android:textColorHint="#00BCD4"  
 android:textColorLink="#00BCD4"  
 \_toBottomOf="@+id/editText3"** />

<**Button  
 android:id="@+id/button3"  
 android:layout\_width="236dp"  
 android:layout\_height="33dp"  
 android:layout\_marginTop="12dp"  
 android:layout\_marginEnd="96dp"  
 android:background="#FAF7F7"  
 android:onClick="table"  
 android:text="@string/olvidaste\_tu\_contrase\_a"  
 android:textAllCaps="false"  
 android:textColorHighlight="#00FFFFFF"  
 android:textColorHint="#00FFFFFF"  
 android:textColorLink="#00FFFFFF"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button4"** />

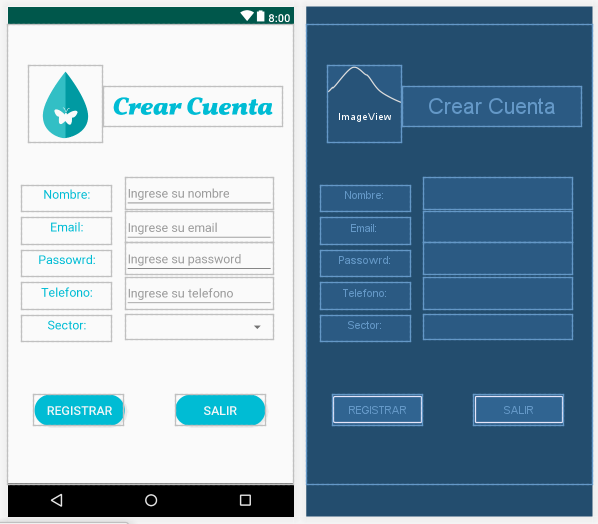
## Crear Cuenta

En el código del archivo CrearCuenta.java, se importaron las librerías correspondientes y se inicializo una variable para crear un Spinner, para poder ingresar las opciones en el Spinner se agrega al mismo una lista con las opciones a través de un ArrayAdapter. Además se agregó el comando Toast, el mismo que aparecerá si la ejecución fue exitosa

**import** android.os.Bundle;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Spinner;  
**import** android.widget.Toast;  
  
**public class** CrearCuenta **extends** AppCompatActivity {  
 **private** Spinner **spinner1**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_crear\_cuenta***);  
  
 **spinner1**= (Spinner)findViewById(R.id.***spinner***);  
  
 String [] opciones= {**"Sector Norte(Guayaquil)"**,**"Daule"**};  
 ArrayAdapter<String> adapter= **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_item***, opciones);  
 **spinner1**.setAdapter(adapter);  
  
 String seleccion= **spinner1**.getSelectedItem().toString();  
 **if** (seleccion.equals(**"Sector Norte(Guayaquil)"**)){  
  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 } **else if** (seleccion.equals(**"Daule"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }  
  
 }  
}

* + 1. **Diseño XML**

Para la interfaz gráfica, en la sección Design, se agregó el logo de la aplicacion,asi también 7 TextView , 4 EditText, 1 Spinner y 2 botones, en los campos del EdiTexts el usuario ingresara sus datos personales, el Spinner ayudara al usuario a seleccionar el sector donde desea instalar el sistema de riego, el botón Registrar se encargara de subir los datos del usuario a la base de datos, el usuario siempre tendrá la opción de salir de la aplicación con el botón Salir.



* + 1. **Código XML:**

<**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="257dp"  
 android:layout\_height="58dp"  
 android:layout\_marginTop="28dp"  
 android:layout\_marginEnd="28dp"  
 android:fontFamily="sans-serif"  
 android:text="@string/crear\_cuenta"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="36sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />

# 

<**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="257dp"  
 android:layout\_height="58dp"  
 android:layout\_marginTop="28dp"  
 android:layout\_marginEnd="28dp"  
 android:fontFamily="sans-serif"  
 android:text="@string/crear\_cuenta"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="36sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
<**ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="105dp"  
 android:layout\_height="111dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="28dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/logo"  
 android:contentDescription="@string/name1"** />  
  
<**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="4dp"  
 android:background="#00BCD4"  
 android:text="@string/salir"  
 android:textColor="#F3F6F7"  
 android:textColorHighlight="#F3F6F7"  
 android:textColorHint="#F3F6F7"  
 android:textColorLink="#F3F6F7"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"** />

<**Button  
 android:id="@+id/button6"  
 android:layout\_width="146dp"  
 android:layout\_height="58dp"  
 android:layout\_marginStart="132dp"  
 android:layout\_marginBottom="80dp"  
 android:background="#00BCD4"  
 android:text="@string/registrar"  
 android:textColor="#F3F6F7"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"** />  
  
<**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="129dp"  
 android:layout\_height="37dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="52dp"  
 android:text="@string/nombre"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"** />

# 

<**TextView  
 android:id="@+id/textView9"  
 android:layout\_width="186dp"  
 android:layout\_height="40dp"  
 android:layout\_marginStart="152dp"  
 android:text="@string/registrate"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="24sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2"** />  
  
<**TextView  
 android:id="@+id/textView4"  
 android:layout\_width="129dp"  
 android:layout\_height="37dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="28dp"  
 android:text="@string/email"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3"** />  
  
<**TextView  
 android:id="@+id/textView5"  
 android:layout\_width="129dp"  
 android:layout\_height="37dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/usuario"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4"** />  
  
<**TextView  
 android:id="@+id/textView7"  
 android:layout\_width="129dp"  
 android:layout\_height="37dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="20dp"  
 android:text="@string/contase\_a"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView5"** />

<**TextView  
 android:id="@+id/textView8"  
 android:layout\_width="130dp"  
 android:layout\_height="57dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/confirma\_la\_contase\_a"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView7"** />

# 

<**TextView  
 android:id="@+id/textView11"  
 android:layout\_width="130dp"  
 android:layout\_height="57dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="96dp"  
 android:text="Sector:"  
 android:textAlignment="center"  
 android:textColor="#00BCD4"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView7"** />  
  
 <**EditText  
 android:id="@+id/editText6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="64dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/ingrese\_su\_nombre"  
 android:inputType="textPersonName"  
 app:layout\_constraintStart\_toEndOf="@+id/textView3"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView9"** />  
  
 <**EditText  
 android:id="@+id/editText5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="16dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/ingrese\_un\_usuario"  
 android:inputType="textPersonName"  
 app:layout\_constraintStart\_toEndOf="@+id/textView5"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText9"** />  
  
 <**EditText  
 android:id="@+id/editText7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="12dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="Ingrese su contraseña "  
 android:inputType="textPassword"  
 app:layout\_constraintStart\_toEndOf="@+id/textView7"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText5"** />

<**EditText  
 android:id="@+id/editText9"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginTop="12dp"  
 android:autofillHints=""  
 android:ems="10"**

# 

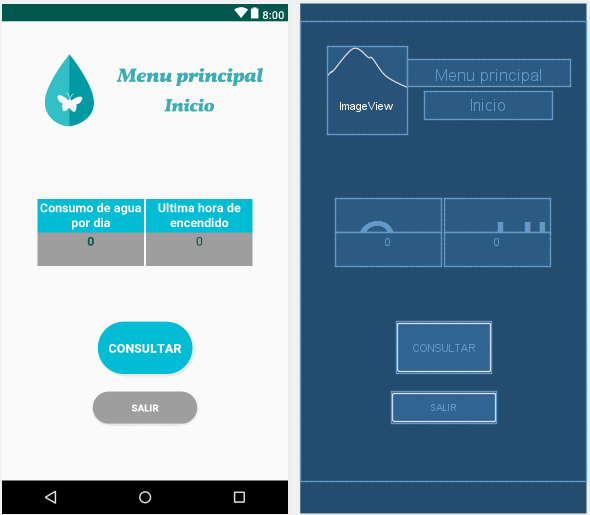
**android:hint="@string/ingrese\_su\_email"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintStart\_toEndOf="@+id/textView4"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText6"** />  
  
 <**Spinner  
 android:id="@+id/spinner2"  
 android:layout\_width="203dp"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="40dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText8"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

## Inicio

En el código del archivo Inicio.java, primero se importó las librerías correspondientes y se inicializo las variables, consumo y horas de tipo TextView, además se crearon dos funciones que están enlazadas a los botones del archivo XML, la función salir, para salir de la ventana actual y la función Consultar, que será activada cuando el usuario presione el botón Consultar, de esta manera el usuario será dirigido a una nueva ventana donde podrá revisar la información de su sistema de riego.

**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
  
**public class** Inicio **extends** AppCompatActivity {  
 TextView **consumo**,**horas**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_inicio***);  
 **consumo** = (TextView) findViewById(R.id.***txt\_consumo2***);  
 **horas**= (TextView) findViewById(R.id.***txt\_hora2***);  
 }  
 **public void** Consultar(View view){  
 Intent i = **new** Intent(**this**, Consultar.**class**);  
 startActivity(i);  
 }  
  
 **public void** Salir (View view){  
 finish();

* + 1. **Diseño XML:**

Para la interfaz gráfica, en la sección Design, se agregó el logo de la aplicación, así también 6 TextView, 4 de ellos que funcionan como etiquetas, títulos, los dos restantes serán declarados variables en el archivo java, además la interfaz tiene 2 botones, uno para consultar y otro para salir de la ventana.

* + 1. **Código XML**

<**ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="114dp"  
 android:layout\_height="128dp"  
 android:layout\_marginStart="32dp"  
 android:layout\_marginTop="36dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/logo"** />  
  
<**TextView  
 android:id="@+id/txt\_consumo3"  
 android:layout\_width="153dp"  
 android:layout\_height="48dp"  
 android:layout\_marginStart="48dp"  
 android:layout\_marginTop="248dp"  
 android:background="#3DADB1"  
 android:text="Consumo de agua por dia"  
 android:textAlignment="center"**

**android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />

<**TextView  
 android:id="@+id/txt\_consumo2"  
 android:layout\_width="153dp"  
 android:layout\_height="48dp"  
 android:layout\_marginStart="48dp"  
 android:layout\_marginTop="296dp"  
 android:background="#3DADB1"  
 android:text="0"  
 android:textAlignment="center"  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
<**TextView  
 android:id="@+id/txt\_hora"  
 android:layout\_width="153dp"  
 android:layout\_height="48dp"  
 android:layout\_marginTop="248dp"  
 android:layout\_marginEnd="56dp"  
 android:background="#3DADB1"  
 android:text="Ultima hora de encendido"  
 android:textAlignment="center"  
 android:textColor="#FFFFFF"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
<**TextView  
 android:id="@+id/txt\_hora2"  
 android:layout\_width="153dp"  
 android:layout\_height="48dp"  
 android:layout\_marginTop="296dp"  
 android:layout\_marginEnd="56dp"  
 android:background="#3DADB1"  
 android:text="0"  
 android:textAlignment="center"  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
<**Button  
 android:id="@+id/btn\_salir"  
 android:layout\_width="150dp"  
 android:layout\_height="46dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="376dp"  
 android:layout\_marginEnd="8dp"  
 android:background="#3DADB1"  
 android:onClick="Salir"  
 android:text="Salir"  
 android:textColor="#FDFFFFFF"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"** />

<**Button  
 android:id="@+id/btn\_consultar"  
 android:layout\_width="136dp"  
 android:layout\_height="75dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="228dp"  
 android:layout\_marginEnd="8dp"  
 android:background="#3DADB1"  
 android:text="Consultar"  
 android:textColor="#FDFFFFFF"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.532"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"** />  
  
<**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="184dp"  
 android:layout\_height="42dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="104dp"  
 android:layout\_marginEnd="48dp"  
 android:text="Inicio"  
 android:textAlignment="center"  
 android:textColor="#3DADB1"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.953"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
<**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="185dp"  
 android:layout\_height="36dp"  
 android:layout\_marginEnd="56dp"  
 android:layout\_marginBottom="12dp"  
 android:text="Menu principal"  
 android:textAlignment="center"  
 android:textColor="#3DADB1"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toTopOf="@+id/textView2"  
 app:layout\_constraintEnd\_toEndOf="parent"** />

## Consultar

En el archivo Consultar.java, se importó las librerías necesarias para esta clase, además se inicializo una variable para crear un Spinner, para poder ingresar las opciones en el Spinner se crea una lista a través de un ArrayAdapter, la cual será agregada al Spinner. Además se creó una función Consultar, en este método se tiene una condición if para validar la seleccion, dentro de cada condición el usuario será dirigido a una nueva activity donde estarán los datos correspondientes a ese mes, además un Toast aparecerá en el caso de que la ejecución haya sido exitosa. La función ahorro que esta enlazada al botón ahorro, va a dirigir al usuario a una activity el cual muestra un diagrama de tipo pastel donde se compara el consumo de agua con el sistema de riego tradicional y con el propuesto, mostrando así el ahorro mensual de agua.

**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Spinner;  
**import** android.widget.Toast;  
  
**public class** Consultar **extends** AppCompatActivity {  
 **private** Spinner **spinner1**;

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_consultar***);  
 **spinner1**= (Spinner)findViewById(R.id.***spinner***);

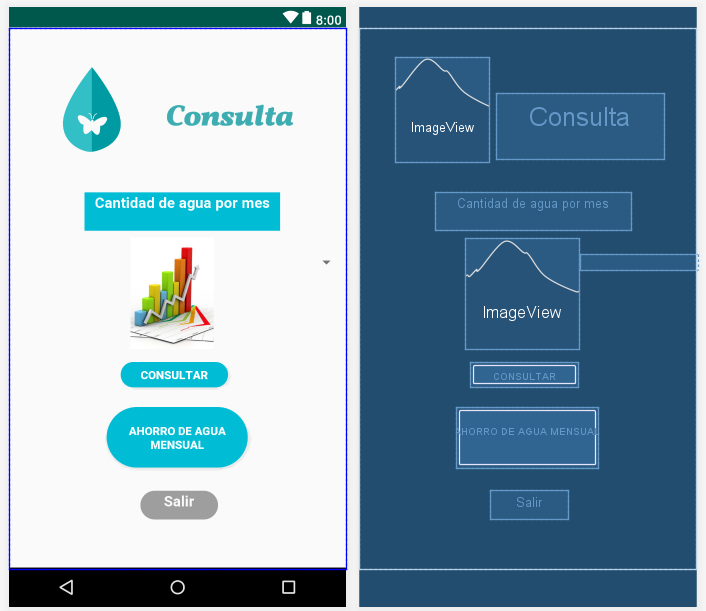
String [] opciones= {**"Enero"**,**"Febrero"**,**"Marzo"**,**"Abril"**,**"Mayo"**,**"Junio"**,**"Julio"**,**"Agosto"**,**"Septiembre"**,**"Optubre"**,**"Noviembre"**,**"Diciembre"**};  
 ArrayAdapter<String> adapter= **new** ArrayAdapter<String>(**this**,   
}

**public void** Consultar(View view){  
 String seleccion= **spinner1**.getSelectedItem().toString();  
 **if** (seleccion.equals(**"Enero"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 } **else if** (seleccion.equals(**"Febrero "**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Marzo"**)){  
  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Abril"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Mayo"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Junio"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Julio"**)){

Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Agosto"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Septiembre"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Octubre"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Noviembre"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }**else if** (seleccion.equals(**"Diciembre"**)){  
  
 Toast.*makeText*(**this**, **"Selección Exitosa"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
  
 **public void** Ahorro(View view){  
 Intent i = **new** Intent();  
 startActivity(i);  
 }  
 **public void** Salir (View view){  
 finish();  
  
 }

* + 1. **Diseño XML**

Para la interfaz gráfica, en la sección Design, se agregó el logo de la aplicación, así también 2 TextView y 3 botones, dos de ellos para ingresar a los datos del sistema de riego, los cuales serán mostrados en una gráfica de barras y en un diagrama pastel, el botón restante será utilizado para salir de la ventana actual.



* + 1. **Código XML**

<**ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="139dp"  
 android:layout\_height="136dp"  
 android:layout\_marginStart="132dp"  
 android:layout\_marginTop="284dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/esta"** />  
  
<**TextView  
 android:id="@+id/textView4"  
 android:layout\_width="145dp"  
 android:layout\_height="45dp"  
 android:layout\_marginTop="76dp"  
 android:layout\_marginEnd="68dp"  
 android:text="Consulta"  
 android:textAlignment="center"  
 android:textColor="#3DADB1"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />

**TextView  
 android:id="@+id/textView5"**

**android:layout\_width="226dp"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginEnd="80dp"  
 android:background="#3DADB1"  
 android:text="Cantidad de agua por mes"  
 android:textAlignment="center"**

**android:textColor="#F7FFFFFF"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4"** />

<**TextView  
 android:id="@+id/textView5"  
 android:layout\_width="226dp"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginEnd="80dp"  
 android:background="#3DADB1"  
 android:text="Cantidad de agua por mes"  
 android:textAlignment="center"  
 android:textColor="#F7FFFFFF"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4"** />

**Button  
 android:id="@+id/btn\_ahorro"  
 android:layout\_width="168dp"  
 android:layout\_height="43dp"  
 android:layout\_marginTop="84dp"  
 android:layout\_marginEnd="120dp"  
 android:background="#3DADB1"  
 android:text="Ahorro de agua mensual"  
 android:textAlignment="center"  
 android:textColor="#F5FFFFFF"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button2"** />

<**Spinner  
 android:id="@+id/spinner"  
 android:layout\_width="143dp"  
 android:layout\_height="20dp"  
 android:layout\_marginTop="264dp"  
 android:layout\_marginEnd="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="131dp"  
 android:layout\_height="31dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="140dp"  
 android:background="#3DADB1"  
 android:onClick="Consultar"  
 android:text="Consultar"  
 android:textAlignment="center"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"** />

### Consumo por Mes

En el código del archivo consumo\_por\_mes.java, primero se importó las librerías correspondientes y se declaró la variable tipo TextView, la variable spinner tipo Spinner y por último el LineChartView, las opciones del spinner fueron creadas con una lista de tipo Arrarlist,

**package** com.example.proyecto;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Spinner;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**import** lecho.lib.hellocharts.model.Axis;  
**import** lecho.lib.hellocharts.model.AxisValue;  
**import** lecho.lib.hellocharts.model.Line;  
**import** lecho.lib.hellocharts.model.LineChartData;  
**import** lecho.lib.hellocharts.model.PointValue;  
**import** lecho.lib.hellocharts.model.Viewport;  
**import** lecho.lib.hellocharts.view.LineChartView;

**public class** consumo\_por\_mes **extends** AppCompatActivity {  
  
 **private** TextView **titulo**;  
 *//data falsa de prueba* **private** Spinner;  
 **private** LineChartView;  
 List **yAxisValues** = **new** ArrayList();  
 List **axisValues** = **new** ArrayList();  
 **final private** String[] **dias** =**new** String[]{**"1"**,**"2"**,**"3"**,**"4"**,  
 **"5"**,**"6"**,**"7"**,**"8"**,**"9"**,**"10"**,**"11"**,**"12"**,**"13"**,**"14"**,**"15"**,**"16"**,  
 **"17"**,**"18"**,**"19"**,**"20"**,**"21"**,**"22"**,**"23"**,**"24"**,**"25"**,**"26"**,**"27"**,  
 **"28"**,**"29"**,**"30"**,**"31"**};  
  
 **private int**[]**valores**=**new int**[]{10,25,30,27,16,44,21,15,28,22,10,25,30,  
 27,16,44,21,15,28,22,10,25,30,27,16,44,21,15,28,22,10};  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_consumo\_por\_mes***);

*//AQUI DEBE SETEARSE EL STRING CONSUMO DE AGUA Y EL MES SELECCIONADO***titulo** = (TextView)findViewById(R.id.***txt\_titulo***);  
**titulo**.setText(**"..."**);

# 

ArrayAdapter<String> adapter= **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_item***, **dias**);  
 **spinner** = (Spinner)findViewById(R.id.***spinner***);  
 **spinner**.setAdapter(adapter);  
  
 **lineChartView**=(LineChartView)findViewById(R.id.***lineChart***);  
 createchart();  
}  
**private void** createchart(){  
 Line line=**new** Line(**yAxisValues**).setColor(Color.*parseColor*(**"#9C27B0"**));  
 **for**(**int** i=0;i<**dias**.**length**;i++){  
 **axisValues**.add(i, **new** AxisValue(i).setLabel(**dias**[i]));  
 }  
 **for**(**int** i=0;i<**valores**.**length**;i++){  
 **yAxisValues**.add(i, **new** PointValue(i,**valores**[i]));  
 }  
 List lines = **new** ArrayList();  
 lines.add(line);  
 LineChartData data = **new** LineChartData();  
 data.setLines(lines);  
 **lineChartView**.setLineChartData(data);  
  
 Axis = **new** Axis();  
 axis.setValues(**axisValues**);  
 data.setAxisXBottom(axis);  
 Axis yAxis = **new** Axis();  
 data.setAxisYLeft(yAxis);  
 axis.setTextSize(16);  
 axis.setTextColor(Color.*parseColor*(**"#03A9F4"**));  
 yAxis.setTextColor(Color.*parseColor*(**"#03A9F4"**));  
 yAxis.setTextSize(16);  
 yAxis.setName(**"Consumo en cm3"**);  
 Viewport viewport = **new** Viewport(**lineChartView**.getMaximumViewport());  
 viewport.**top** =110;  
 **lineChartView**.setMaximumViewport(viewport);  
 **lineChartView**.setCurrentViewport(viewport);

}  
 **public void** tabla2(View view){  
 Intent i = **new** Intent(**this**,consulta\_por\_dia.**class**);  
 startActivity(i);  
 }  
 **public void** Salir (View view){  
 finish();  
  
 }  
}

* + 1. **Diseño XML**

Para la interfaz gráfica de esta activity, se le añadió una gráfica de barras y dos botones, de los cuales, el botón consulta por día, dirige al usuario a una nueva activity donde podrá observar los datos obtenidos por los sensores, el botón salir ayuda al usuario a salir de la ventana actual.

# 

* + 1. **Código XML**

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".consumo\_por\_mes"**>  
  
 <**TextView  
 android:id="@+id/txt\_titulo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="8dp"  
 android:text="Consumo de agua en "  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**/>  
 <**lecho.lib.hellocharts.view.LineChartView  
 android:id="@+id/lineChart"  
 android:layout\_width="398dp"  
 android:layout\_height="391dp"  
 android:layout\_marginStart="4dp"  
 android:layout\_marginTop="28dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_weight="1"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_titulo"** />

<**Spinner  
 android:id="@+id/spinner"  
 android:layout\_width="188dp"  
 android:layout\_height="32dp"  
 android:layout\_marginTop="24dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/lineChart"** />  
  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="28dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:onClick="tabla2"  
 android:text="Consultar por dia"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/button5"  
 app:layout\_constraintHorizontal\_bias="0.47"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/spinner"  
 app:layout\_constraintVertical\_bias="0.0"** />  
  
 <**Button  
 android:id="@+id/button5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="28dp"  
 android:layout\_marginEnd="60dp"  
 android:layout\_marginBottom="8dp"  
 android:onClick="Salir"  
 android:text="volver"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/spinner"  
 app:layout\_constraintVertical\_bias="0.0"** />  
</**androidx.constraintlayout.widget.ConstraintLayout**>

### Consulta por día

En el código del archivo consumo\_por\_dia.java, primero se importó las librerías correspondientes , las cuales permiten obtener una gráfica que muestran los datos obtenidos por los sensores en el día especificado por el usuario.

**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.graphics.Color;  
**import** android.os.Bundle;  
**import** android.view.View;  
  
**import** com.github.mikephil.charting.charts.BarChart;  
**import** com.github.mikephil.charting.charts.Chart;  
**import** com.github.mikephil.charting.components.Legend;  
**import** com.github.mikephil.charting.components.LegendEntry;  
**import** com.github.mikephil.charting.components.XAxis;  
**import** com.github.mikephil.charting.components.YAxis;  
**import** com.github.mikephil.charting.data.BarData;  
**import** com.github.mikephil.charting.data.BarDataSet;  
**import** com.github.mikephil.charting.data.BarEntry;  
**import** com.github.mikephil.charting.data.DataSet;  
**import** com.github.mikephil.charting.formatter.IndexAxisValueFormatter;  
  
**import** java.util.ArrayList;  
  
**public class** consulta\_por\_dia **extends** AppCompatActivity {  
  
 **private** BarChart **barChart**;  
  
 *//FALTA PASAR LA SELECCION DEL DIA DE LA ACTITIVITY ANTERIOR  
 //VALIDAR LA SELECCION EN LA BASE DE DATOS Y LLENAR LA TABLA* **final private** String[] **horas** =**new** String[]{**"18"**,**"19"**,**"20"**,**"21"**,**"22"**,**"23"**,  
 **"24"**,**"1"**,**"2"**,**"3"**,**"4"**,**"5"**,**"6"**};  
  
 **private int**[]**valores**=**new int**[]{10,25,30,27,16,44,21,15,28,22,24,  
 30,25};  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_consulta\_por\_dia***);  
 **barChart**=(BarChart)findViewById(R.id.***lineChart***);  
 createCharts();  
 }

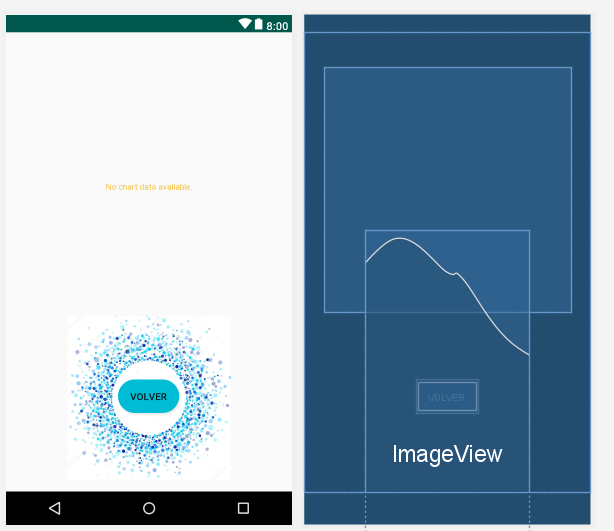
**private** Chart getSameChart(Chart chart, String descripcion, **int** textColor, **int** backgroung, **int** animateY){  
 chart.getDescription().setText(descripcion);  
 chart.getDescription().setTextSize(15);  
 chart.setBackgroundColor(backgroung);  
 chart.animateY(animateY);  
 legend(chart);  
 **return** chart;  
}

**private void** legend(Chart chart){  
 Legend legend=chart.getLegend();  
 legend.setForm(Legend.LegendForm.***DEFAULT***);  
 legend.setHorizontalAlignment(Legend.LegendHorizontalAlignment.***CENTER***);

ArrayList<LegendEntry> entries=**new** ArrayList<>();  
 **for**(**int** i = 0; i< **horas**.**length**; i++){  
 LegendEntry entry=**new** LegendEntry();  
 entry.**label**= **horas**[i];  
 entries.add(entry);  
 }  
 legend.setCustom(entries);  
}

**private** ArrayList<BarEntry>getBarEntries(){  
 ArrayList<BarEntry> entries = **new** ArrayList<>();  
 **for**(**int** i=0; i<**valores**.**length**;i++){  
 entries.add(**new** BarEntry(i,**valores**[i]));  
 }  
 **return** entries;  
 }  
 **private void** axisX(XAxis axis){  
 axis.setGranularityEnabled(**true**);  
 axis.setPosition(XAxis.XAxisPosition.***BOTTOM***);  
 axis.setValueFormatter(**new** IndexAxisValueFormatter(**horas**));  
 }  
  
 **private void** axisLeft(YAxis axis){  
 axis.setSpaceTop(35);  
 axis.setAxisMinimum(0);  
 }  
 **private void** axisRight(YAxis axis){  
 axis.setEnabled(**false**);  
 }  
  
 **public void** createCharts(){  
 **barChart**=(BarChart)getSameChart(**barChart**, **"Consumo por mes"**,  
 Color.***BLUE***,Color.***LTGRAY***,3000);  
 **barChart**.setDrawGridBackground(**true**);  
 **barChart**.setData(getBarData());  
 axisX(**barChart**.getXAxis());  
 axisLeft(**barChart**.getAxisLeft());  
 axisRight(**barChart**.getAxisRight());  
 }  
 **private** DataSet getData(DataSet dataSet){  
 dataSet.setValueTextSize(10);  
 **return** dataSet;  
 }  
 **private** BarData getBarData(){  
 BarDataSet barDataSet=(BarDataSet)getData(**new** BarDataSet(getBarEntries(),**""**));  
 barDataSet.setBarShadowColor(Color.***GRAY***);  
 BarData barData=**new** BarData(barDataSet);  
 barData.setBarWidth(0.45f);  
 **return** barData;  
 }  
 **public void** Salir (View view){  
 finish();  
 }

* + 1. **Diseño XML**



<**Button  
 android:id="@+id/button6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="96dp"  
 android:layout\_marginBottom="8dp"  
 android:onClick="Salir"  
 android:text="Volver"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/lineChart"  
 app:layout\_constraintVertical\_bias="0.0"** />  
</**androidx.constraintlayout.widget.ConstraintLayout**>

* + 1. **Código XML**

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".consulta\_por\_dia"**>  
  
 <**com.github.mikephil.charting.charts.BarChart  
 android:id="@+id/lineChart"  
 android:layout\_width="398dp"  
 android:layout\_height="391dp"  
 android:layout\_marginStart="4dp"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginEnd="4dp"  
 android:layout\_weight="1"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_titulo"** />

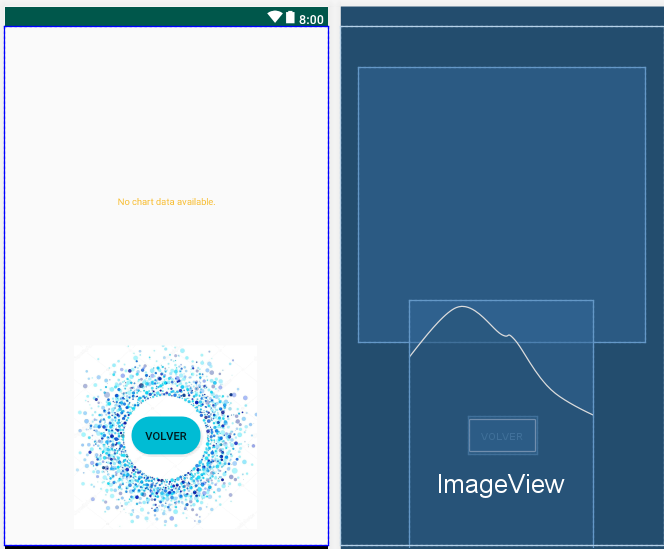
## Ahorro

En el código del archivo consumo\_por\_dia.java, primero se importó las librerías correspondientes , las cuales permiten obtener una gráfica pastel, en la cual se compara el consumo de agua de un sistema de riego tradicional y el consumo con el sistema de control automatizado.

**public class** Ahorro **extends** AppCompatActivity {  
  
 **private** PieChart;  
 **final private** String[] **horas** =**new** String[]{**"Sistema tradicional"**,**"Sistema automatizado"**};  
 **private int**[] **colors**=**new int**[]{Color.***GREEN***,Color.***YELLOW***};  
 **private int**[]**valores**=**new int**[]{10,25};  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_ahorro***);  
 **pieChart**=(PieChart)findViewById(R.id.***pieChart***);  
 createCharts();  
 }  
 **private** Chart getSameChart(Chart chart, String descripcion, **int** textColor, **int** backgroung, **int** animateY){  
 chart.getDescription().setText(descripcion);  
 chart.getDescription().setTextSize(15);  
 chart.setBackgroundColor(backgroung);  
 chart.animateY(animateY);  
 legend(chart);  
 **return** chart;  
 }**private void** legend(Chart chart){  
 Legend legend=chart.getLegend();  
 legend.setForm(Legend.LegendForm.***DEFAULT***);  
 legend.setHorizontalAlignment(Legend.LegendHorizontalAlignment.***CENTER***);

ArrayList<LegendEntry> entries=**new** ArrayList<>();  
 **for**(**int** i = 0; i< **horas**.**length**; i++){  
 LegendEntry entry=**new** LegendEntry();  
 entry.**formColor** = **colors**[i];  
 entry.**label**= **horas**[i];  
 entries.add(entry);  
 }  
 legend.setCustom(entries);  
 }  
 **private** ArrayList<PieEntry>getBarEntries(){  
 ArrayList<PieEntry> entries = **new** ArrayList<>();  
 **for**(**int** i=0; i<**valores**.**length**;i++){  
 entries.add(**new** PieEntry(**valores**[i]));  
 }  
 **return** entries;  
 }  
  
 **public void** createCharts(){  
 **pieChart**=(PieChart)getSameChart(**pieChart**,**"Ahorro mensual"**,Color.***GRAY***,Color.***CYAN***,3000);  
 **pieChart**.setHoleRadius(2);  
 **pieChart**.setData(getBarData());  
 **pieChart**.invalidate();  
 }  
 **private** DataSet getData(DataSet dataSet){  
 dataSet.setColors(**colors**);  
 dataSet.setValueTextSize(Color.***WHITE***);  
 dataSet.setValueTextSize(10);  
 **return** dataSet;  
 }  
 **private** PieData getBarData(){  
 PieDataSet pieDataSet=(PieDataSet) getData(**new** PieDataSet(getBarEntries(),**""**));  
 pieDataSet.setSliceSpace(1);  
 pieDataSet.setValueFormatter(**new** PercentFormatter());  
  
 **return new** PieData(pieDataSet);  
 }  
  
 **public void** Salir (View view){  
 finish();  
 }  
}

* + 1. **Diseño XML**



* + 1. **Código XML**

<**com.github.mikephil.charting.charts.PieChart  
 android:id="@+id/pieChart"  
 android:layout\_width="365dp"  
 android:layout\_height="349dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginBottom="8dp"  
 android:layout\_weight="1"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.151"** />  
  
<**Button  
 android:id="@+id/button6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="8dp"  
 android:background="@drawable/boton\_redondo"  
 android:onClick="Salir"  
 android:text="Volver"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/pieChart"  
 app:layout\_constraintVertical\_bias="0.467"** />

<**ImageView  
 android:id="@+id/imageView5"  
 android:layout\_width="233dp"  
 android:layout\_height="347dp"  
 android:layout\_marginStart="88dp"  
 android:layout\_marginLeft="88dp"  
 android:layout\_marginTop="348dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/ima"** />  
</**androidx.constraintlayout.widget.Constrai**

# Código Arduino

Para el código en Arduino se importó 3 librerías, estas librerías permiten el uso del módulo de wifi, de los sensores de humedad y temperatura y la utilización de la mini bomba de agua Dc, mediante el módulo wifi se realizó la conexión a un servidor, y mediante al método Post envía al servidor los datos recolectados por los sensores a la base de datos. Además en el código se valida la activación del sistema de riego dadas ciertas condiciones ambientales, las cuales son registradas por el sensor DHtY1 y YL-69, las cuales determinan el encendido y apagado de la bomba.

#incluye <SimpleDHT.h>

// Librería <SimpleDHT.h> del sensor de temperatura y humedad relativa

#include <SPI.h>

#include <ESP8266WiFi.h>

//Librería <SPI.h> del motor DC compatible con la minibomba de agua DC

#define humidity\_sensor\_pin A0

//Se define variable del sensor de humedad en el suelo en el pin A0

// for DHT11,

// VCC: 5V or 3V

// GND: GND

// DATA: 2

int pinDHT11 = 2;

// Se declara la variable pinDHT11 y lo asocio al pin 2

SimpleDHT11 dht11;

int water\_pump\_pin = 14;

//Se declara la variable mini bomba de agua y lo asocio al pin 3

int water\_pump\_speed = 255;

int bandera=0;

//Velocidad de la minibomba de agua oscila entre 100 como mínimo y 255 como máximo. Yo he //elegido 255 pero ustedes pueden elegir la que estimen conveniente. A más velocidad, mayor //bombeo de aguaconst char\* ssid = "Internet Zambrano";  
const char\* password = "zambrano\_ruiz";  
const char\* host = "192.168.0.109";

# 

float tiempo\_inicial=0;

float tiempo\_final=0;

float duracion\_encendido=0;

const float caudal\_bomba=2;

static char outstr[15];

String mac\_adress="";

void setup() {

Serial.begin(9600);

pinMode(water\_pump\_pin,OUTPUT);

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

} Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

mac\_adress= WiFi.macAddress();

Serial.println(mac\_adress);

}

void loop() {

// start working…

// Mide la temperatura y humedad relativa y muestra resultado

Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Serial.println("Sample DHT11…");

// read with raw sample data.

byte temperature = 0;

byte humidity\_in\_air = 0;

byte data[40] = {0};

if (dht11.read(pinDHT11, &temperature, &humidity\_in\_air, data)) {

Serial.print("Read DHT11 failed")

}Serial.print("Sample RAW Bits: ");

for (int i = 0; i < 40; i++) {

Serial.print((int)data[i]);

if (i > 0 && ((i + 1) % 4) == 0) {

Serial.print(' ');

} }Serial.println("");

Serial.print("Sample OK: ");

Serial.print("Temperature: ");Serial.print((int)temperature); Serial.print(" \*C, ");

Serial.print("Relative humidity in air: ");Serial.print((int)humidity\_in\_air); Serial.println(" %");

// DHT11 sampling rate is 1HZ.

// Ground humidity value in %

// Mide la humedad en el suelo en % y muestra el resultado

int ground\_humidity\_value = map(analogRead(humidity\_sensor\_pin), 450, 1023, 100, 0);

Serial.print("Ground humidity: ");

Serial.print(ground\_humidity\_value);

Serial.println("%");

if( ground\_humidity\_value <= 50 && digitalRead(ldr\_pin) == LOW){

Serial.println("low");}

if( ground\_humidity\_value <= 50 && temperature < 30) {

bandera+=1;

if( bandera==1) {

tiempo\_inicial=mills();

}digitalWrite(water\_pump\_pin, HIGH);

Serial.println("Irrigate");

//analogWrite(water\_pump\_pin, water\_pump\_speed);

//El motor de la bomba de agua arranca con la velocidad elegida anteriormente en el código

else{

digitalWrite(water\_pump\_pin, LOW);

Serial.println("Do not irrigate");

//El motor de la bomba de agua se para y no riega

if (bandera==1){

bandera=0;

tiempo\_final=millis();

duracion\_encendido=((tiempo\_final-tiempo\_inicial)\*caudal\_bomba)/(100\*60);

String duracion\_final = String(duracion\_encendido);

Serial.print("connecting to ");

Serial.println(host)

// Use WiFiClient class to create TCP connections

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;

}

// We now create a URL for the request

String url = "/dht11.php";

String duracion = "?duracion\_riego=";

String dispositivo= "&dispositivo=";

Serial.print("Requesting URL: ");

Serial.println(url);

client.print(String("GET ") + url + duracion + duracion\_final + dispositivo + mac\_adress+ " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

unsigned long timeout = millis();

while (client.available() == 0) {

if (millis() - timeout > 5000) {

Serial.println(">>> Client Timeout !");

client.stop();

return;

}

}

while (client.available()) {

String line = client.readStringUntil('\r');

Serial.print(line);

}

Serial.println();

Serial.println("closing connection");

}

}// Ejecuta el código cada 100 milisegundo

delay(2000);

}

## PhpMyAdmin

Para la base de datos, se procedió a crear 5 tablas en las cuales se almacenará la información recopilada por los sensores.

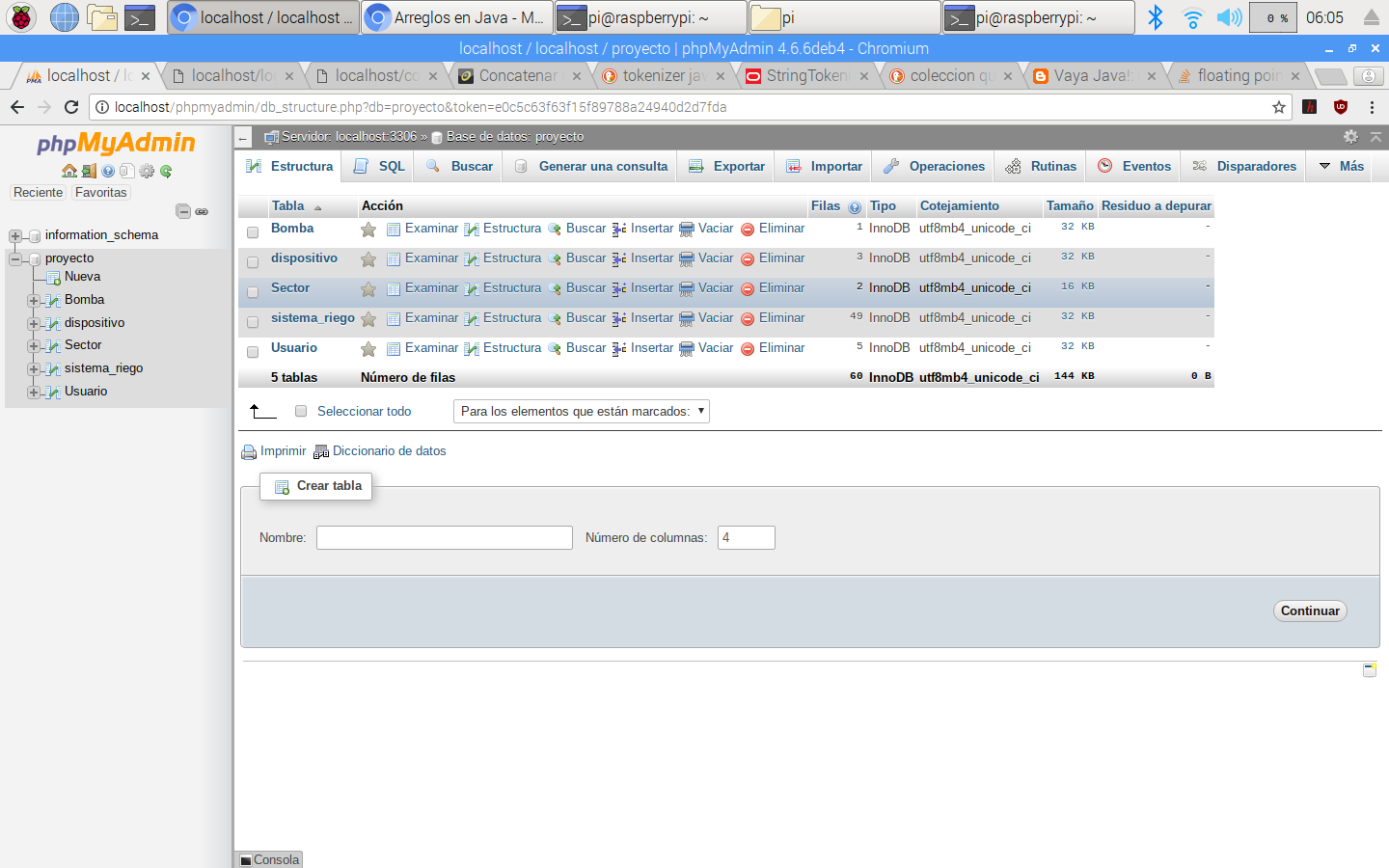


Figura 5. Creación de base de datos phpMyAdmin

**Tablas:**

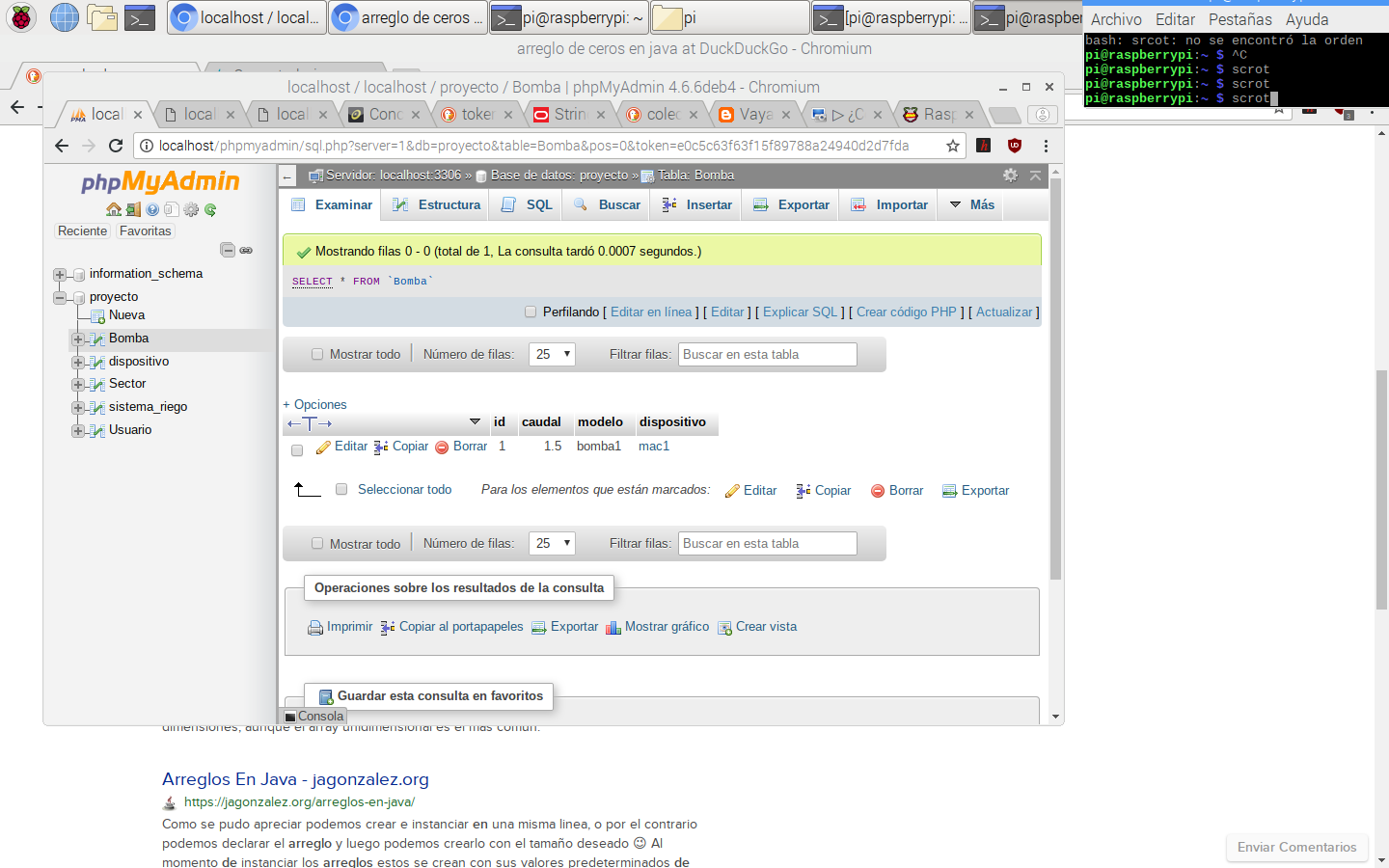


Figura 6. Creación tabla bomba con sus respectivos parámetros:

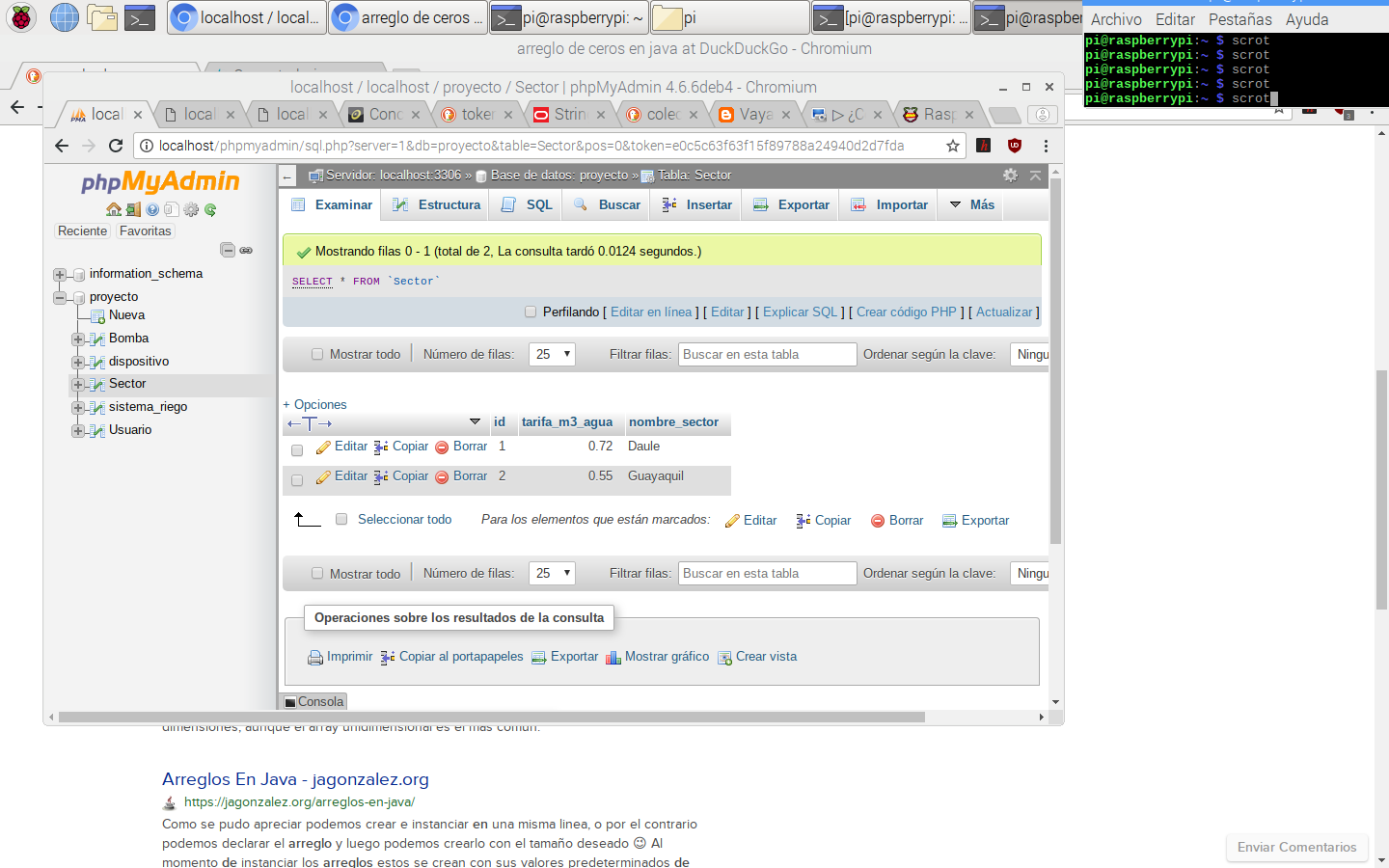


Figura 7. Creación de tabla sector con sus respectivos parámetros.



Figura 8. Creación de tabla Usuario con sus respectivos parámetros

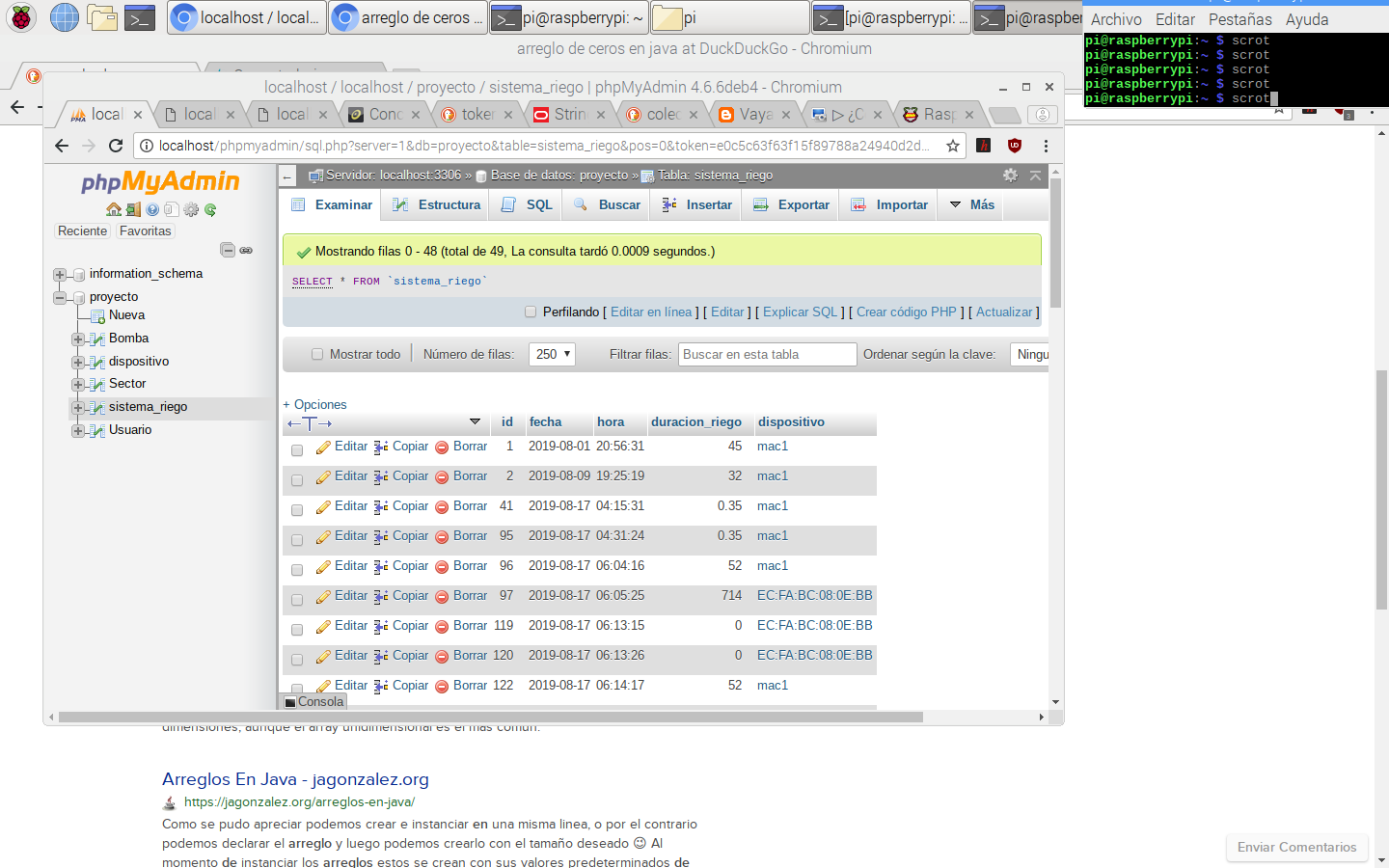


Figura 9. Creación de tabla sistema\_riego con sus respectivos parámetros

**Código SQL**

-- phpMyAdmin SQL Dump-- version 4.6.6deb4

-- https://www.phpmyadmin.net/

--

-- Servidor: localhost:3306

-- Tiempo de generación: 20-08-2019 a las 17:38:55

-- Versión del servidor: 10.1.38-MariaDB-0+deb9u1

-- Versión de PHP: 7.0.33-0+deb9u3

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Base de datos: `proyecto`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Bomba`

CREATE TABLE `Bomba` ( `id` int(11) NOT NULL,

`caudal` float DEFAULT NULL,

`modelo` varchar(20) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`dispositivo` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `Bomba`

--

INSERT INTO `Bomba` (`id`, `caudal`, `modelo`, `dispositivo`) VALUES

(1, 1.5, 'bomba1', 'mac1');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `dispositivo`

--

CREATE TABLE `dispositivo` (

`id\_MAC` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`modelo` varchar(30) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`usuario` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `dispositivo`

--

INSERT INTO `dispositivo` (`id\_MAC`, `modelo`, `usuario`) VALUES

('EC:FA:BC:08:0E:BB', 'arduino3', 1),

('mac1', 'arduino1', 1),

('mac2', 'arduino2', 2);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Sector`

--

CREATE TABLE `Sector` (

`id` int(11) NOT NULL,

`tarifa\_m3\_agua` float NOT NULL,

`nombre\_sector` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

---- Volcado de datos para la tabla `Sector`

--

INSERT INTO `Sector` (`id`, `tarifa\_m3\_agua`, `nombre\_sector`) VALUES(1, 0.72, 'Daule'),

(2, 0.55, 'Guayaquil');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `sistema\_riego`

--

CREATE TABLE `sistema\_riego` (

`id` int(11) NOT NULL,

`fecha` date DEFAULT NULL,

`hora` time DEFAULT NULL,

`duracion\_riego` float DEFAULT NULL,

`dispositivo` varchar(30) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `sistema\_riego`

--

INSERT INTO `sistema\_riego` (`id`, `fecha`, `hora`, `duracion\_riego`, `dispositivo`) VALUES

(1, '2019-08-01', '20:56:31', 45, 'mac1'),

(2, '2019-08-09', '19:25:19', 32, 'mac1'),

(41, '2019-08-17', '04:15:31', 0.35, 'mac1'),

(95, '2019-08-17', '04:31:24', 0.35, 'mac1'),

(96, '2019-08-17', '06:04:16', 52, 'mac1'),

(97, '2019-08-17', '06:05:25', 714, 'EC:FA:BC:08:0E:BB'),

(119, '2019-08-17', '06:13:15', 0, 'EC:FA:BC:08:0E:BB'),

(120, '2019-08-17', '06:13:26', 0, 'EC:FA:BC:08:0E:BB'),

(122, '2019-08-17', '06:14:17', 52, 'mac1'),

(123, '2019-08-17', '06:19:14', 4216, 'EC:FA:BC:08:0E:BB'),

(124, '2019-08-17', '06:19:26', 15542, 'EC:FA:BC:08:0E:BB'),

(125, '2019-08-17', '06:19:37', 26867, 'EC:FA:BC:08:0E:BB'),

(180, '2019-08-17', '06:40:04', 147948, 'EC:FA:BC:08:0E:BB'),

(185, '2019-08-17', '06:41:01', 204573, 'EC:FA:BC:08:0E:BB'),

(186, '2019-08-17', '06:41:18', 6098, 'EC:FA:BC:08:0E:BB'),

(187, '2019-08-17', '06:44:26', 6098, 'EC:FA:BC:08:0E:BB'),

(188, '2019-08-17', '06:44:37', 17423, 'EC:FA:BC:08:0E:BB'),

(234, '2019-08-17', '14:03:56', 0.35, 'mac1'),(235, '2019-08-18', '02:27:02', 424.77, 'EC:FA:BC:08:0E:BB'),

(236, '2019-08-18', '02:28:38', 84.97, 'EC:FA:BC:08:0E:BB'),

(237, '2019-08-18', '02:29:14', 24.29, 'EC:FA:BC:08:0E:BB'),

(238, '2019-08-18', '02:31:14', 109.24, 'EC:FA:BC:08:0E:BB'),(239, '2019-08-18', '02:36:23', 291.28, 'EC:FA:BC:08:0E:BB'),

(240, '2019-08-18', '02:53:03', 12.43, 'EC:FA:BC:08:0E:BB'),

(241, '2019-08-18', '02:53:13', 2.09, 'EC:FA:BC:08:0E:BB'),(242, '2019-08-18', '02:55:39', 2.09, 'EC:FA:BC:08:0E:BB'),

(243, '2019-08-18', '02:56:49', 51.72, 'EC:FA:BC:08:0E:BB'),(244, '2019-08-18', '02:58:25', 6.22, 'EC:FA:BC:08:0E:BB'),

(245, '2019-08-18', '02:58:42', 2.09, 'EC:FA:BC:08:0E:BB'),

(246, '2019-08-18', '02:59:02', 2.09, 'EC:FA:BC:08:0E:BB'),

(247, '2019-08-18', '02:59:20', 10.36, 'EC:FA:BC:08:0E:BB'),

(248, '2019-08-18', '03:00:27', 12.43, 'EC:FA:BC:08:0E:BB'),

(249, '2019-08-18', '03:00:44', 2.09, 'EC:FA:BC:08:0E:BB'),

(250, '2019-08-18', '03:01:08', 2.09, 'EC:FA:BC:08:0E:BB'),

(251, '2019-08-18', '03:01:17', 2.09, 'EC:FA:BC:08:0E:BB'),

(252, '2019-08-18', '03:02:50', 82.76, 'EC:FA:BC:08:0E:BB'),

(253, '2019-08-18', '03:05:11', 134.43, 'EC:FA:BC:08:0E:BB'),

(254, '2019-08-18', '03:05:42', 12.43, 'EC:FA:BC:08:0E:BB'),

(255, '2019-08-18', '03:22:03', 7090.46, 'EC:FA:BC:08:0E:BB'),

(256, '2019-08-18', '03:26:54', 2.58, 'EC:FA:BC:08:0E:BB'),

(257, '2019-08-18', '03:32:00', 0.62, 'EC:FA:BC:08:0E:BB'),

(258, '2019-08-18', '03:32:38', 0.78, 'EC:FA:BC:08:0E:BB'),

(259, '2019-08-18', '03:33:12', 0.52, 'EC:FA:BC:08:0E:BB'),

(260, '2019-08-18', '05:45:09', 0.05, 'EC:FA:BC:08:0E:BB'),

(261, '2019-08-18', '05:51:19', 0.1, 'EC:FA:BC:08:0E:BB'),

(262, '2019-08-18', '05:51:30', 0.05, 'EC:FA:BC:08:0E:BB'),

(263, '2019-08-18', '05:52:41', 0.41, 'EC:FA:BC:08:0E:BB'),

(264, '2019-08-18', '05:53:13', 0.31, 'EC:FA:BC:08:0E:BB'),

(265, '2019-08-18', '05:55:23', 0.62, 'EC:FA:BC:08:0E:BB'),

(266, '2019-08-18', '09:15:34', 0.05, 'EC:FA:BC:08:0E:BB'),

(267, '2019-08-20', '16:16:25', 35.31, 'EC:FA:BC:08:0E:BB'),

(268, '2019-08-20', '16:17:05', 33.11, 'EC:FA:BC:08:0E:BB'),

(269, '2019-08-20', '16:19:04', 111.68, 'EC:FA:BC:08:0E:BB'),

(270, '2019-08-20', '16:21:42', 148.9, 'EC:FA:BC:08:0E:BB'),

(271, '2019-08-20', '16:22:00', 10.36, 'EC:FA:BC:08:0E:BB'),

(272, '2019-08-20', '16:26:38', 270.92, 'EC:FA:BC:08:0E:BB'),

(273, '2019-08-20', '16:26:50', 4.17, 'EC:FA:BC:08:0E:BB'),

(274, '2019-08-20', '16:27:03', 6.24, 'EC:FA:BC:08:0E:BB'),

(275, '2019-08-20', '16:27:25', 14.49, 'EC:FA:BC:08:0E:BB'),

(276, '2019-08-20', '16:33:17', 0.35, 'mac1');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `Usuario`

--

CREATE TABLE `Usuario` (

`id` int(11) NOT NULL,

`nombres` varchar(50) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`correo` varchar(30) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`password` varchar(30) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`telefono` int(11) DEFAULT NULL,

`id\_sector` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

---- Volcado de datos para la tabla `Usuario`

--

---- Filtros para la tabla `Bomba`

--

ALTER TABLE `Bomba`

ADD CONSTRAINT `Bomba\_ibfk\_1` FOREIGN KEY (`dispositivo`) REFERENCES `dispositivo` (`id\_MAC`);

--

-- Filtros para la tabla `dispositivo`

--

ALTER TABLE `dispositivo`

ADD CONSTRAINT `dispositivo\_ibfk\_1` FOREIGN KEY (`usuario`) REFERENCES `Usuario` (`id`);

--

-- Filtros para la tabla `sistema\_riego`

--

ALTER TABLE `sistema\_riego`

ADD CONSTRAINT `sistema\_riego\_ibfk\_1` FOREIGN KEY (`dispositivo`) REFERENCES `dispositivo` (`id\_MAC`);

--

-- Filtros para la tabla `Usuario`

--

ALTER TABLE `Usuario`

ADD CONSTRAINT `Usuario\_ibfk\_1` FOREIGN KEY (`id\_sector`) REFERENCES `Sector` (`id`);

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

# Implementación física

Para la implementación física del proyecto, en primer lugar, se debió comprar todos los dispositivos, y se procedió a conectar cada uno de los sensores.

Cada uno de los sensores cuenta con distintos requerimientos por lo cual, a continuación, se detalla el proceso de conexión de cada uno de ellos.

## Sensor de humedad del suelo

Este sensor nos permite obtener una medida análoga de la humedad del suelo, además consta con un módulo incluido el yl-38, el cual actúa como un amplificador y acondicionador de señal.

El yl-38 consta de 2 pines de conexión hacia el módulo YL-69, y 4 pines más para la conexión con el módulo Wifi ESP8266MOD, dos para alimentación Vcc y Gnd, y 2 pines para datos de salida D0 (salida digital) y A0 (salida analógica).

Para nuestro caso solo es necesaria utilizar la salida analógica, dado que solo se requiere obtener los datos de humedad del suelo de forma precisa. Dado que el módulo wifi utilizado solo cuenta con una salida analógica A0, el sensor debe ser conectado en ese pin, a continuación, se muestra la conexión implementada para este sensor.

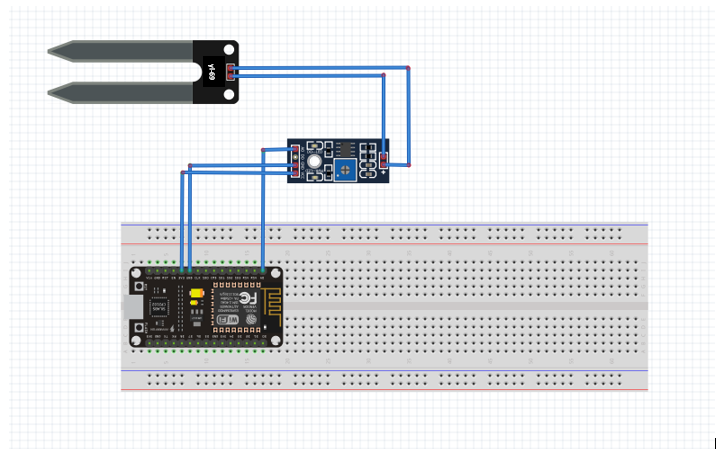


Figura 10. Conexión de sensor de humedad yl-69

## Sensor de humedad y temperatura del aire

El DHT11 es un sensor digital de temperatura y humedad relativa de bajo costo y fácil uso. Este sensor consta de 3 pines lo cuales dos de ellos son para la alimentación es decir Vcc y Ground, y un pin digital para datos, que en nuestro proyecto este ha sido implementado en el pin D4 del módulo wifi. En la figura adjunta a continuación se muestra la implementación final de este sensor. *Intagro, C (2016).*

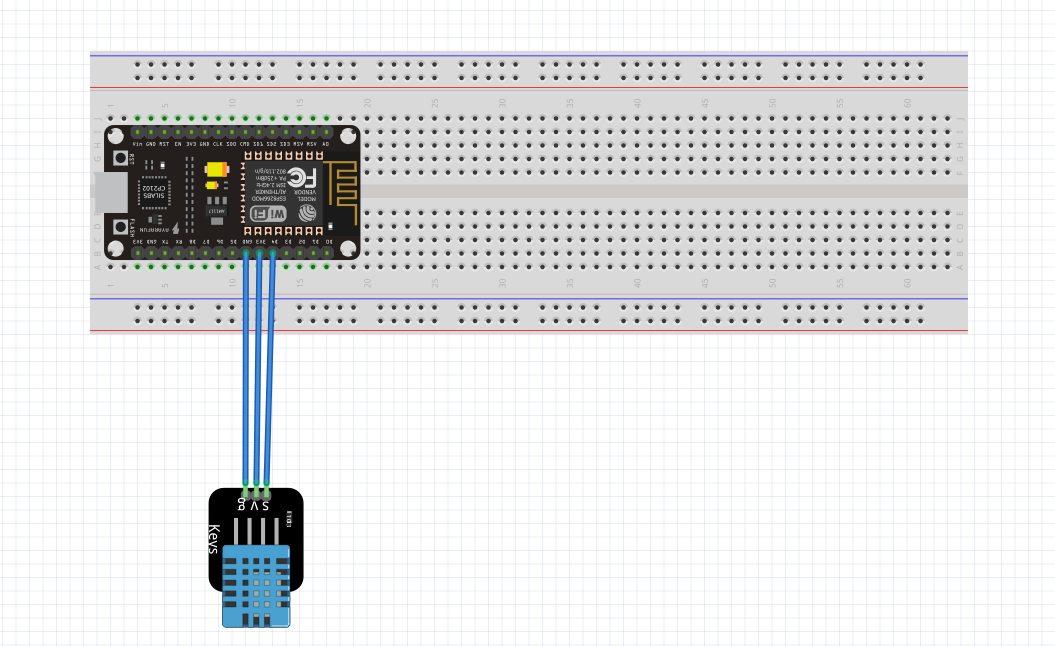


Figura 11. Conexión del sensor de humedad y temperatura del aire DHT11

## Relé

Para el control de la bomba, en nuestro proyecto utilizaremos Relé de 5V DC Simple Inversor, el cual permitirá activar o desactivar la bomba, según las condiciones del ambiente y del suelo. Para su implementación, debido a que la corriente y el voltaje entregado por la salida digital del módulo wifi, no era suficiente para activar el relé, se utilizó el transistor PN 2222. , para poder aumentar la corriente y que se activara el relé. La configuración realizada, se puede observar en la figura adjunta.

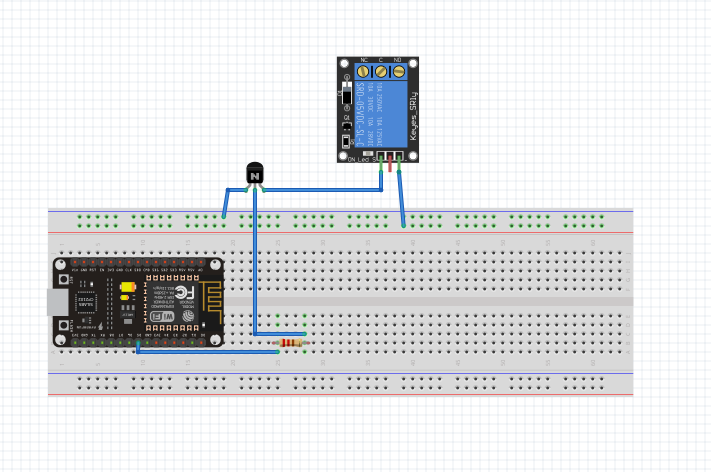


Figura 12. Conexión del relé para el control de la bomba

## Implementación final del circuito

Una vez, ya comprobado el correcto funcionamiento de cada uno de los sensores, se procede a realizar la implementación final, para lo cual se conecta todos los sensores en un mismo circuito y utilizando una fuente de alimentación de 5V para alimentar el circuito, se puede observar la implementación final a continuación. *Gavilán, P., Martinez-Ferri, E., & Lozano, D. (2018).*

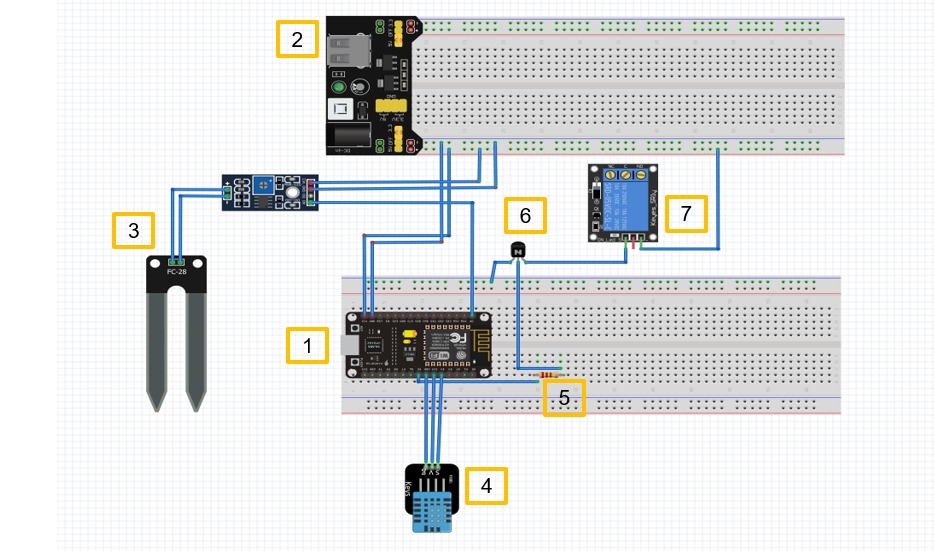


Figura 13. Implementación final del circuito para el sistema de riego

**Descripción de los componentes del circuito:**

1. Modulo Wifi ESP8266
2. Fuente de alimentación para protoboard de 5V YWRobot.
3. Sensor de humedad del suelo yl-69.
4. Sensor de temperatura y humedad del aire dht11.
5. Resistencia de 1k ohmios.
6. Transistor PN2222
7. Relé inversor de 3.3 V

# Implementación de maqueta para simular el sistema de riego

1. En primer lugar, se mandó hacer una maceta de madera de 1x0.5x.0.15 m, a la cual se le instalo una manguera de media pulgada, en su centro, por donde pasar el agua, la cual será enviada por la bomba, desde un repositorio.



Figura 14. Maceta de madera



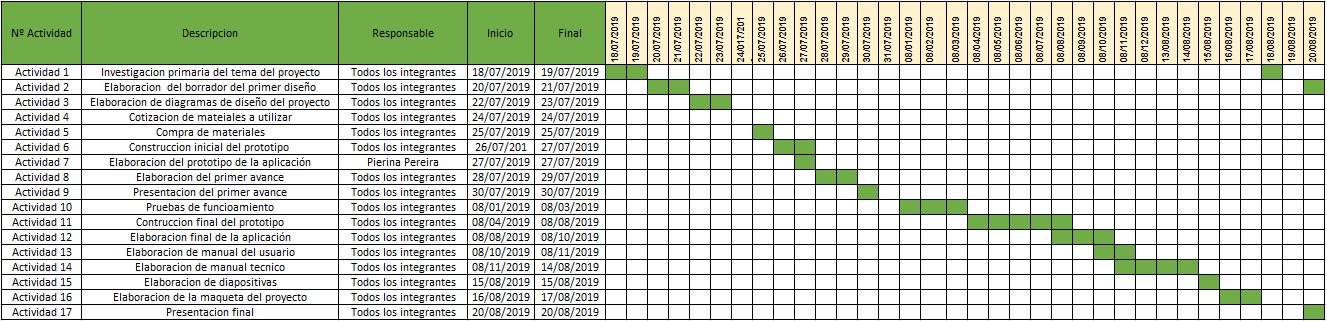
Figura 15. Implementación de manguera para sistema de riego

1. Para el reservorio donde se almacenará el agua, se utilizó una tina, a la cual se instaló una manguera, la cual se conecta a la bomba, para que se realice el riego cada vez que la bomba sea activada.



Figura 16. Pruebas de funcionamiento

# Diagrama de Gantt

****

# 

# Análisis de presupuesto

|  |  |  |  |
| --- | --- | --- | --- |
| **Cantidad** | **Material** | **Precio Unidad** | **Precio Final** |
| **1** | Raspberry PI 3 modelo B++ | $ 60 | $ 60 |
| **1** | Protoboard | $ 4.00 | $ 4.00 |
| **1** | Mini Bomba De Agua De Diafragma 6-12v R385 Acuario | $8.00 | $8.00 |
| **1** | Sensor de humedad del suelo Yl-69 | $2.00 | $2.00 |
| 1 | Sensor de temperatura y humedad relativa en el aire DHT11 | $2.00 | $2.00 |
| 1 | Transistor PN2222 | $0.50 | $0.50 |
| 1 | Resistencia de 1 K Ohmios | $0.10 | $0.10 |
| 1 | Relé | $1.00 | $1.00 |
| 15 | Jumpers | $0.10 | $ 1.50 |
| 1 | Módulo de wifi ESP8266 | 10.00 | $10.00 |
| 1 | Fuente de poder | 6.00 | $6.00 |
| 2 | Madera | $5 | $10 |
| 1 | Manguera | $4 | $4 |
| 1 | Codo | $0.5 | $0.5 |
| 1 | Tina | $2 | $2 |
|  |  | **TOTAL** | **$ 106.60** |

# 

# Conclusiones

* Con la implementación de este proyecto, se puede concluir que un sistema de riego automatizado permite un ahorro significativo de recursos, además garantiza un cuidado adecuado del cultivo, ya que el sembrío es regado según lo requiera dependiendo de las condiciones ambientales.
* La aplicación diseñada brinda la posibilidad al usuario de acceder en tiempo real y desde cualquier lugar, a la información relevante sobre el sistema de riego, dado al uso de una base de datos la cual permite almacenar información, la mismo que será enviada a la aplicación.
* Mediante la programación en Arduino IDE, se puede manipular la activación de la bomba, con la finalidad de logar un riego adecuado y un ahorro considerable de recursos.

# Referencias bibliográficas

* Intagro, C (2016). *Uso de sensores de Humedad para definir riego.* Obtenido de intagri: https://www.intagri.com/articulos/agua-riego/uso-de-sensores-de-humedad-para-definir-riego
* Lasmonedasdejudas. (2017, 01 31). *Sistema de riego hecho con arduino y varios sensores de luz, temperatura y humedad que usa una minibomba de agua. .* Retrieved from Las Monedas de Judas: <https://lasmonedasdejudas.wordpress.com/2017/01/31/sistema-de-riego-hecho-con-arduino-y-varios-sensores-de-luz-temperatura-y-humedad-que-usa-una-minibomba-de-agua/>
* Gavilán, P., Martinez-Ferri, E., & Lozano, D. (2018). *Consumo de agua e indicadores de riego del cultivo de la frambuesa en Huelva*. (October). Retrieved from <https://www.researchgate.net/publication/328138706>

# 

# Anexos:



Figura 17. Exposición de proyecto



Figura 18. Circuito y maqueta final



Figura 19. Presentación final



Figura 20. Prueba de funcionamiento



Figura 21. Integrantes del grupo