

Integrantes:

Giovanny Mite

Darien Bustos

Cristoffer Montenegro

Erik Pulley

Avance 2 Proyecto PST

- **[AVANCE 2] Resultados: Explicación detallada de las pruebas del funcionamiento del proyecto incluyendo imágenes y video demostrativo. Usen estadística descriptiva para mostrar los resultados de las pruebas realizadas indicando los aciertos o fallos obtenidos.**

Respecto al software, el desarrollo de la aplicación móvil se volvió compleja al realizar el respectivo inicio de sesión, su creación de usuario, su elección de acciones para el usuario y finalmente el porcentaje de llenado para que el usuario tenga una idea.

La conexión con la base de datos se dificultó para el inicio de sesión de los usuarios, así mismo se dificultó la selección de colores estéticamente correctos para la aplicación.

Por el lado del hardware, se realizó un sistema mediante una conexión en arduino, donde por medio de un módulo web, se nos permitió la conexión directa a una página web la cual nos permitió la interacción directa de usuario y control respecto al funcionamiento principal de nuestro sistema.

El sistema poseerá un motor reductor el cual girará de una manera controlada para abrir la compuerta del flujo de comida, este funcionamiento de apertura irá vinculado directamente con el control del usuario hacia la aplicación, que a su vez esta indicará que tan llena está la comida, donde finalmente dependiendo del valor de llenado, el usuario deberá proceder a la acción.



Figure 1 Pagina web para manejo de hardware



Figure 2 Inicio Splash de la aplicación



Figure 3 Inicio de Sesión de Usuario



Figure 4 Recolección de datos para la creación de usuario.



Figure 5 Selección de operación a realizar



Figure 6 Porcentajes informativos.

- **[AVANCE2] Conclusiones: Acerca del desarrollo del proyecto.**
 - El vínculo principal entre la aplicación y el hardware es la base de datos que ambas partes modificarán, esto permitirá el intercambio de datos directo para la interacción inalámbrica entre el usuario con su aplicación en su celular, y el hardware en casa.
 - Podemos concluir que el modulo wifi ESP8266 permite acceder a una red wifi comunicando mediante diversos sensores, cuanto se llena el dispensador, y cuando está vacío.
 - Se pudo crear una aplicación mediante Android Studio que nos permitió realizar la interfaz grafica para el usuario, con su respectiva creación de usuario y su inicio de sesión correspondiente.
- **[AVANCE 2] Presupuesto: Tabla con la descripción de las partes, cantidad, vendedor, precios y precio total**

Partes	Precio	Descripción
Modulo WIFI ESP8266 V3	\$8.00	Equipo que permite al Arduino conectarse a una red para el intercambio de datos.
Motorreductor DC	\$2.00	Motor que gira con una potencia controlada con una fuente de voltaje de 5VDC

Sensor ultrasónico	\$2.00	Sensor detector de aproximación.
Rele DC	\$3.00	Sistema periférico de accionamiento del motor.
Precio Final	\$15.00	

Table 1 Precios de los elementos usados para el Desarrollo del hardware

- **[AVANCE 2] Apéndice: Código fuente de hardware y software. Referencias bibliográficas. Código Hardware**

```
#include <ESP8266WiFi.h>
```

```
const char* ssid = "A51";
const char* password = "herculesn";
```

```
int ledPin = 13; // GPIO13
WiFiServer server(80);
```

```
void setup() {
  Serial.begin(115200);
  delay(10);
```

```
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
```

```
  // Connect to WiFi network
  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");
```

```
  // Start the server
  server.begin();
  Serial.println("Server started");
```

```
  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
```

```

}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  // Match the request

  int value = LOW;
  if (request.indexOf("/LED=ON") != -1) {
    digitalWrite(ledPin, HIGH);
    value = HIGH;
  }
  if (request.indexOf("/LED=OFF") != -1) {
    digitalWrite(ledPin, LOW);
    value = LOW;
  }

  // Set ledPin according to the request
  //digitalWrite(ledPin, value);

  // Return the response
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println(""); // do not forget this one
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");

  //client.print("Led pin is now: ");

  // if(value == HIGH) {
  //   client.print("On");

```

```

// } else {
//   client.print("Off");
// }
client.println("<head>");
client.println("</head>");
client.println("<body>");
client.println("<h1 align='center'>DISPETSER</h1><h3 align='center'>Dispensador de
comida</h3>");
//Creamos los botones. Para enviar parametres a través de HTML se utiliza el metodo URL
encode. Los parámetros se envían a través del símbolo '?'
client.println("<div style='text-align:center;'>");
client.println("<button onClick=location.href='./LED=ON\' style='margin:auto;background-color:
#84B1FF;color: snow;padding: 10px;border: 10px solid #3F7CFF;width:65px;'>");
client.println("SERVIR");
client.println("</button>");
client.println("<button onClick=location.href='./LED=OFF\' style='margin:auto;background-
color: #84B1FF;color: snow;padding: 10px;border: 10px solid #3F7CFF;width:65px;'>");
client.println("PARAR");
client.println("</button>");
client.println("<br /><br />");
client.println("<b> %COMIDA = 50% ");
client.println("</b><br />");
client.println("</b></body>");
client.println("</html>");
delay(1);
Serial.println("Client disconnected");
Serial.println("");
}

```

Código software Login

```
package com.example.dispetser;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.EditText;
```

```
import android.widget.Toast;
```

```
public class Login extends AppCompatActivity {
```

```
    EditText username, password;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_login);

        username = (EditText) findViewById(R.id.username);
        password = (EditText) findViewById(R.id.password);

    }

    public void registro(View view){
        Intent i = new Intent(this, Register.class );
        i.putExtra("username", username.getText().toString());
        i.putExtra("password", password.getText().toString());
        startActivity(i);
        username.setText("");
        password.setText("");
        finish();
    }

    public void acciones (View view){
        if(!username.getText().toString().equals("") && !password.getText().toString().equals("")){
            Intent i = new Intent(this, Actions.class );
            i.putExtra("username", username.getText().toString());
            startActivity(i);
            username.setText("");
            password.setText("");
            finish();}
        else{ Toast.makeText(this, "Ingrese los datos
completos", Toast.LENGTH_SHORT).show();}
        Intent i = new Intent(this, Actions.class );
        if(username.equals("Grupo_1") && password.equals("12345")){
            startActivity(i);

        }
    }

}

```

Referencias Bibliográficas.

<http://internetofthinking.blogspot.com/2015/12/control-led-from-webserver-using.html>
<http://diymakers.es/crear-servidor-web-con-arduino/>
https://www.youtube.com/watch?v=noT_Mrfstdw

- **[AVANCE 2] Crear un video demostrativo del proyecto en que cada integrante del equipo explicará las funciones del proyecto usando diversos escenarios. El archivo del video debe ser proporcionado a los ayudantes vía Github o enlace, para subirlo en el canal de PST (YouTube).**

<https://youtu.be/eTqBk9-TVmI>

- **[AVANCE 2] En el repositorio de GitHub se almacenará el proyecto, colocando el código fuente (hardware y software), informe técnico, diagramas, y el video.**
- [https://github.com/ersepull/Proyecto PST G1](https://github.com/ersepull/Proyecto_PST_G1)