MECAGUYS

AQUAFRESH

DISPENSADOR AUTOMATICO DE BEBIDAS

RESUMEN DE CONTENIDOS

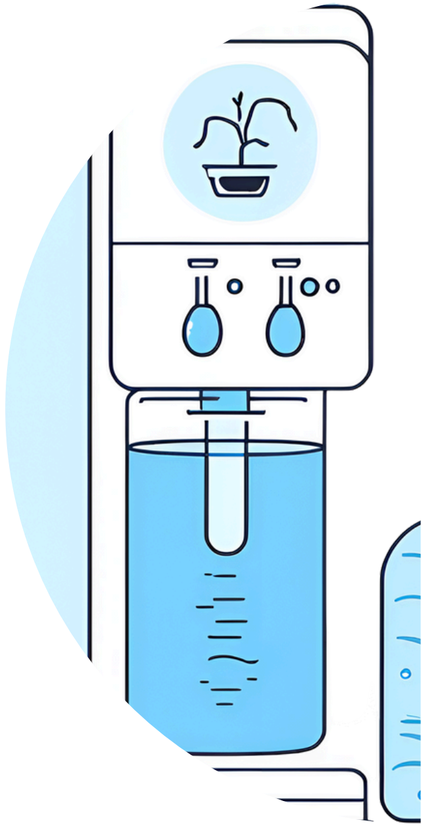
INTRODUCCION

OBJETIVOS

ALCANCE

RESUMEN TÉCNICO

INTRODUCCIÓN



Aquafresh es un dispensador de bebidas a elección del usuario que automatiza este proceso y lo hace fácil e intuitivo para cualquier persona.

OBJETIVOS

OBJETIVO COMERCIAL

Proveer una solución económica y eficiente para el llenado automatizado de líquidos.

OBJETIVO MODERNO

Diseñar un sistema de dispensado automatizado que funcione con sensores inductivos y un puente Wheatstone.

OBJETIVO ECOLÓGICO

Reducir el impacto ambiental al minimizar el desperdicio de botellas mediante un sistema automatizado de autollenado.

OBJETIVO DE LA MATERIA

Poder implementar una interfaz grafica que guarde y muestre los datos del proyecto, ademas de un control del prouecto mediante git.

ALCANCE

El primer prototipo busca sentar la base y demostrar que el proyecto se podría producir en grandes cantidades, al ser portátil, barato y de fácil mantenimiento. El proyecto está enfocado a sectores como: empresas de alimentos y bebidas, espacios públicos como oficinas y hospitales, además de consumidores individuales interesados en soluciones prácticas para el hogar.

RESUMEN TÉCNICO

Python

Botones y pantallas

```
#login
mensaje_error = ctk.StringVar()
titulo_login = ctk.CTkLabel(pagina_login, text="Inicio de sesión", font=('Yeah Papa', 55), fg_color="#32ade6", text_color="black")
titulo_login.place(relx=0.5, rely=0.35, anchor='center')
mensaje = ctk.CTkLabel(pagina_login, textvariable=mensaje_error, font=('Zen Dots', 14), text_color="white", fg_color="#32ADE6", bg_color="#32ADE6")
mensaje.place(relx=0.5, rely=0.4, anchor='center')
entrada_usuario = ctk.CTkEntry(pagina_login, placeholder_text="Ingrese datos de Usuario", width=300, corner_radius=10, font=('Squada One', 20), fg_color="white",
                               , border_width=3)
entrada_usuario.place(relx=0.5, rely=0.5, anchor='center')
entrada_password = ctk.CTkEntry(pagina_login, placeholder_text="Ingrese datos de su Password", width=300, corner_radius=10, font=('Squada One', 20), fg_color="wh
                                , border_width=3)
entrada_password.place(relx=0.5, rely=0.6, anchor='center')
boton_iniciar_sesion = ctk.CTkButton(pagina_login, text="Iniciar Sesión", width=250, height=50, corner_radius=10,  command=procesar_inicio_sesion, font=('Squada
boton_iniciar_sesion.place(relx=0.5, rely=0.735, anchor='center')
boton_registrarse = ctk.CTkButton(pagina_login, text="Registrarse", width=250, height=50, corner_radius=10, command=pantalla_registro, font=('Squada One', 22), b
boton_registrarse.place(relx=0.5, rely=0.8, anchor='center')

#registro
titulo_registro = ctk.CTkLabel(pagina_registro, text="Craación de cuenta", font=('Yeah Papa', 55, 'bold'), fg_color="#32ade6", text_color="black")
titulo_registro.place(relx=0.21, rely=0.3)
registro_usuario = ctk.CTkEntry(pagina_registro, placeholder_text="Registre datos de Usuario", width=300, corner_radius=10, font=('Squada One', 20), fg_color="wh
                                , border_width=3)
registro_usuario.place(relx=0.3, rely=0.5, anchor='center')
registro_password = ctk.CTkEntry(pagina_registro, placeholder_text="Registre datos de su Password", width=300, corner_radius=10, show="*", font=('Squada One', 20
                                 , border_width=3)
registro_password.place(relx=0.3, rely=0.6, anchor='center')
iniciar_sesion = ctk.CTkButton(pagina_registro, text="Guardar", width=250, height=50, corner_radius=10, command=procesar_registro, font=('Squada One', 22), bg_co
iniciar_sesion.place(relx=0.3, rely=0.75, anchor='center')

```

Botones y pantallas

```
126 # Cargar las imágenes de fondo
127 fondo1 = Image.open("INTERFAZ_BIENVENIDA.png").resize((1900, 1000), Image.LANCZOS)
128 fondo1_tk = ImageTk.PhotoImage(fondo1)
129
130 fondo2 = Image.open("INTERFAZ_INICIO_SESION.png").resize((1900, 1000), Image.LANCZOS)
131 fondo2_tk = ImageTk.PhotoImage(fondo2)
132
133 fondo3 = Image.open("INTERFAZ_REGISTRO_USUARIO.png").resize((1900, 1000), Image.LANCZOS)
134 fondo3_tk = ImageTk.PhotoImage(fondo3)
135
136 # Crear frames de las páginas con imagen de fondo
137 pagina_inicio = ctk.CTkFrame(ventana, width=1500, height=800, fg_color="transparent")
138 pagina_inicio = ctk.CTkFrame(ventana, width=1500, height=800, fg_color="transparent")
139 pagina_registro = ctk.CTkFrame(ventana, width=1500, height=800, fg_color="transparent")
140 pagina_login = ctk.CTkFrame(ventana, width=1500, height=800, fg_color="transparent")
141
142 pagina_datos = ctk.CTkFrame(ventana, width=1500, height=800, fg_color="transparent")
```

Guardado de datos

```
#pantalla de datos
datos1 = ctk.CTkEntry(pagina_datos, placeholder_text="Presencia A", width=250, height=50, corner_radius=15, font=('Squada One', 20), fg_color="white", bg_color="#
datos1.place(relx=0.2, rely=0.4, anchor='center')

datos2 = ctk.CTkEntry(pagina_datos, placeholder_text="Presencia B", width=250, height=50, corner_radius=15, font=('Squada One', 20), fg_color="white", bg_color="#
datos2.place(relx=0.4, rely=0.4, anchor='center')

datos3 = ctk.CTkEntry(pagina_datos, placeholder_text="Luminosidad 1", width=250, height=50, corner_radius=15, font=('Squada One', 20), fg_color="white", bg_color=
datos3.place(relx=0.2, rely=0.6, anchor='center')

datos4 = ctk.CTkEntry(pagina_datos, placeholder_text="Luminosidad 2", width=250, height=50, corner_radius=15, font=('Squada One', 20), fg_color="white", bg_color=
datos4.place(relx=0.4, rely=0.6, anchor='center')

boton_guardar = ctk.CTkButton(pagina_datos, text="GUARDAR", width=200, height=40, corner_radius=10, font=('Squada One', 30), command=lambda: [guardar_datos_csv()]
boton_guardar.place(relx=0.9, rely=0.9, anchor='center')
```

```
def guardar_datos_csv():
    """Recoge los datos de los campos y guarda en un archivo CSV."""
    data = {
        "Presencia A": datos1.get(),
        "Presencia B": datos2.get(),
        "Luminosidad 1": datos3.get(),
        "Luminosidad 2": datos4.get(),
    }
    df = pd.DataFrame([data])
    file_path = ctk.filedialog.asksaveasfilename(
        defaultextension=".csv",
        filetypes=[("Archivos CSV", "*.csv"), ("Todos los archivos", "*.*")],
        title="Guardar datos como"
    )
    if file_path:
        df.to_csv(file_path, index=False)
        print(f"Datos guardados en {file_path}")
```

JSON

```
ser = None
lectura_hilo = None

def inicializar_puerto_serial():
    global ser
    try:
        ser = serial.Serial('COM3', 9600, timeout=2)  # Cambia 'COM6' al puerto serial que estás utilizando
    except serial.SerialException as e:
        print("Error al abrir el puerto serial:", e)

def leer_datos_serial():
    global ser
    while True:
        if ser and ser.is_open:
            try:
                data = ser.readline().decode().strip()
                if data:
                    try:
                        json_data = json.loads(data)
                        actualizar_entradas(json_data)
                    except json.JSONDecodeError:
                        print("Error al decodificar JSON:", data)
            except serial.SerialException as e:
                print("Error en la lectura del puerto serial:", e)
        else:
            time.sleep(2)
```

JSON

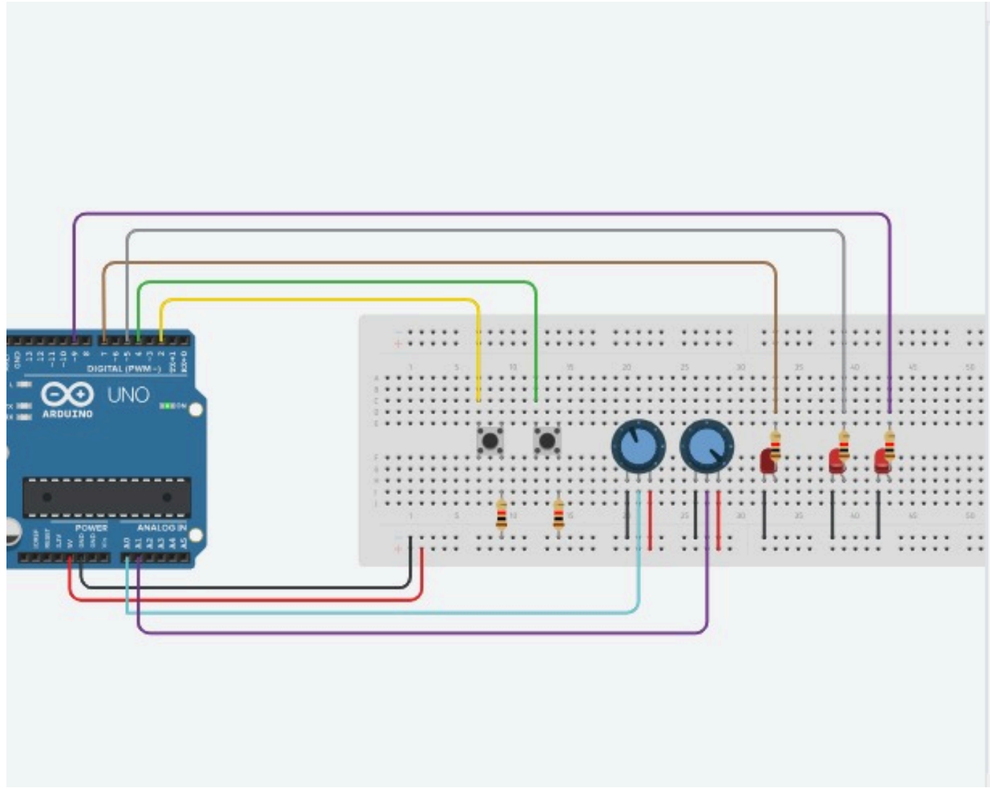
```python
def actualizar_entradas(data):
    """Actualiza las entradas de la interfaz con los datos recibidos del Arduino."""
    if 'datos1' in data:
        datos1.delete(0, ctk.END)
        datos1.insert(0, str(data['datos1']))
    if 'datos2' in data:
        datos2.delete(0, ctk.END)
        datos2.insert(0, str(data['datos2']))
    if 'datos3' in data:
        datos3.delete(0, ctk.END)
        datos3.insert(0, str(data['datos3']))
    if 'datos4' in data:
        datos4.delete(0, ctk.END)
        datos4.insert(0, str(data['datos4']))
```

```python
# Ejecuta la función para leer datos serial en un hilo aparte después de cargar la interfaz
def iniciar_lectura_serial():
    global lectura_hilo
    inicializar_puerto_serial()
    lectura_hilo = threading.Thread(target=leer_datos_serial, daemon=True)
    lectura_hilo.start()
```

Arduino

```
#include <ArduinoJson.h>
#include <SoftwareSerial.h>

#define finalcarA 2
#define finalcarB 4
#define poten1 A0
#define poten2 A1
#define led1 7
#define led2 5
#define led3 9

int presenciaA;
int presenciaB;
int lumin1;
int lumin2;

void setup() {
  pinMode(finalcarA, INPUT_PULLUP);
  pinMode(finalcarB, INPUT_PULLUP);
  pinMode(poten1, INPUT);
  pinMode(poten2, INPUT);
  pinMode(led1, OUTPUT);
  digitalWrite(led1, LOW);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);

  Serial.begin(9600);
}

void loop() {
  // Lectura de sensores de presencia
  presenciaA = digitalRead(finalcarA);
  presenciaB = digitalRead(finalcarB);

  if (presenciaA == LOW || presenciaB == LOW) {
    digitalWrite(led1, HIGH);
    delay(2000);
    digitalWrite(led1, LOW);
  } else {
    digitalWrite(led1, LOW);
  }

  // Lectura de potenciómetros y ajuste de luminosidad
  int medida1 = analogRead(poten1);
  int medida2 = analogRead(poten2);
  lumin1 = map(medida1, 8, 1015, 255, 0);
  lumin2 = map(medida2, 8, 1015, 255, 0);

  analogWrite(led2, lumin1);
  analogWrite(led3, lumin2);

  // Creación del objeto JSON
  StaticJsonDocument<200> jsonDoc;
  jsonDoc["datos1"] = presenciaA;
  jsonDoc["datos2"] = presenciaB;
  jsonDoc["datos3"] = lumin1;
  jsonDoc["datos4"] = lumin2;

  // Convertir el JSON a string y enviarlo por Serial
  String output;
  serializeJson(jsonDoc, output);
  Serial.println(output);

  delay(500); // Espera para evitar lecturas excesivas
}
```

Arduino




```
#define finalcarA 2
#define finalcarB 4
#define poten1 A0
#define poten2 A1
#define led1 7
#define led2 5
#define led3 9

int presenciaA;
int presenciaB;
int lumin1;
int lumin2;

void setup()
{
  pinMode(finalcarA, INPUT_PULLUP);
  pinMode(finalcarB, INPUT_PULLUP);
  pinMode(poten1, INPUT);
  pinMode(poten2, INPUT);
  pinMode(led1, OUTPUT);
  digitalWrite(led1, LOW);
  pinMode(led2, OUTPUT);
  digitalWrite(led2, LOW);
  pinMode(led3, OUTPUT);
  digitalWrite(led3, LOW);
  Serial.begin(9600);
}

void loop()
```

¡MUCHAS GRACIAS!

ESPERAMOS HAYA SIDO
DE SU AGRADO