DOMÒTICA D'UNA CASA
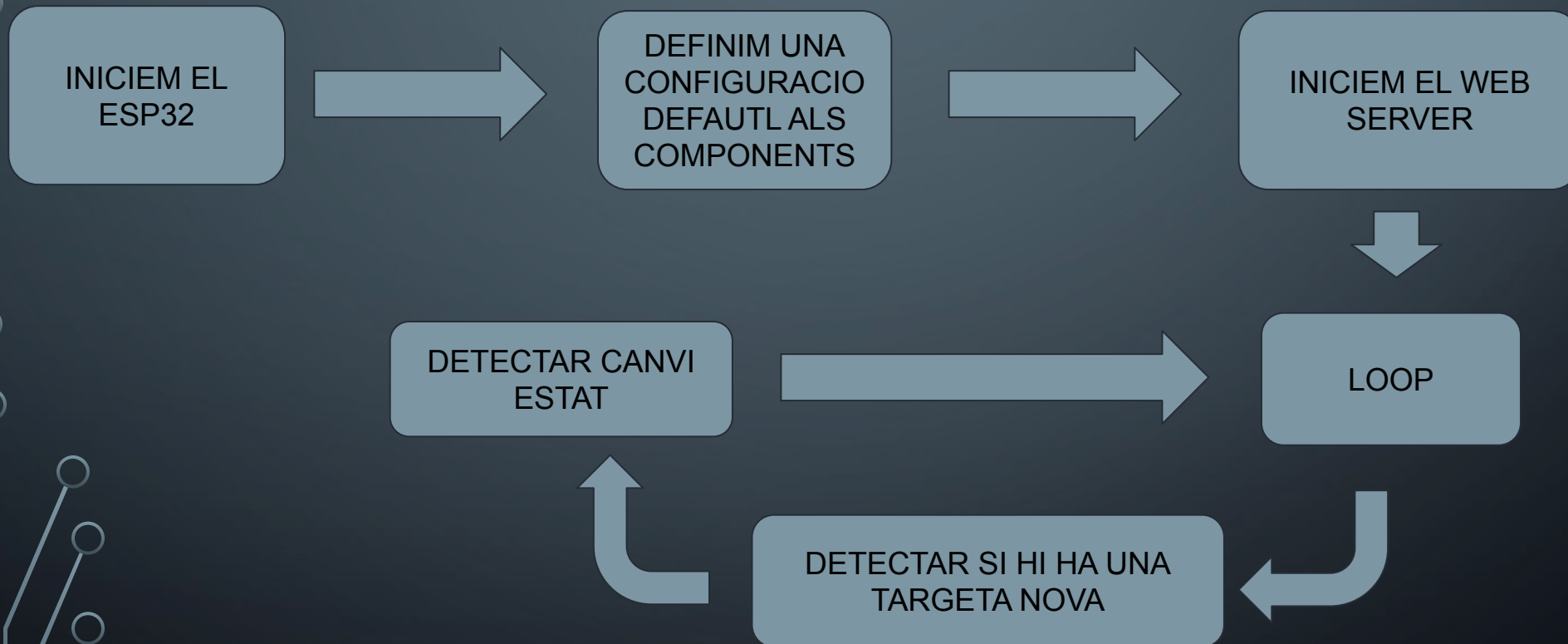
PROJECTE DE PROCESSADORS DIGITALS
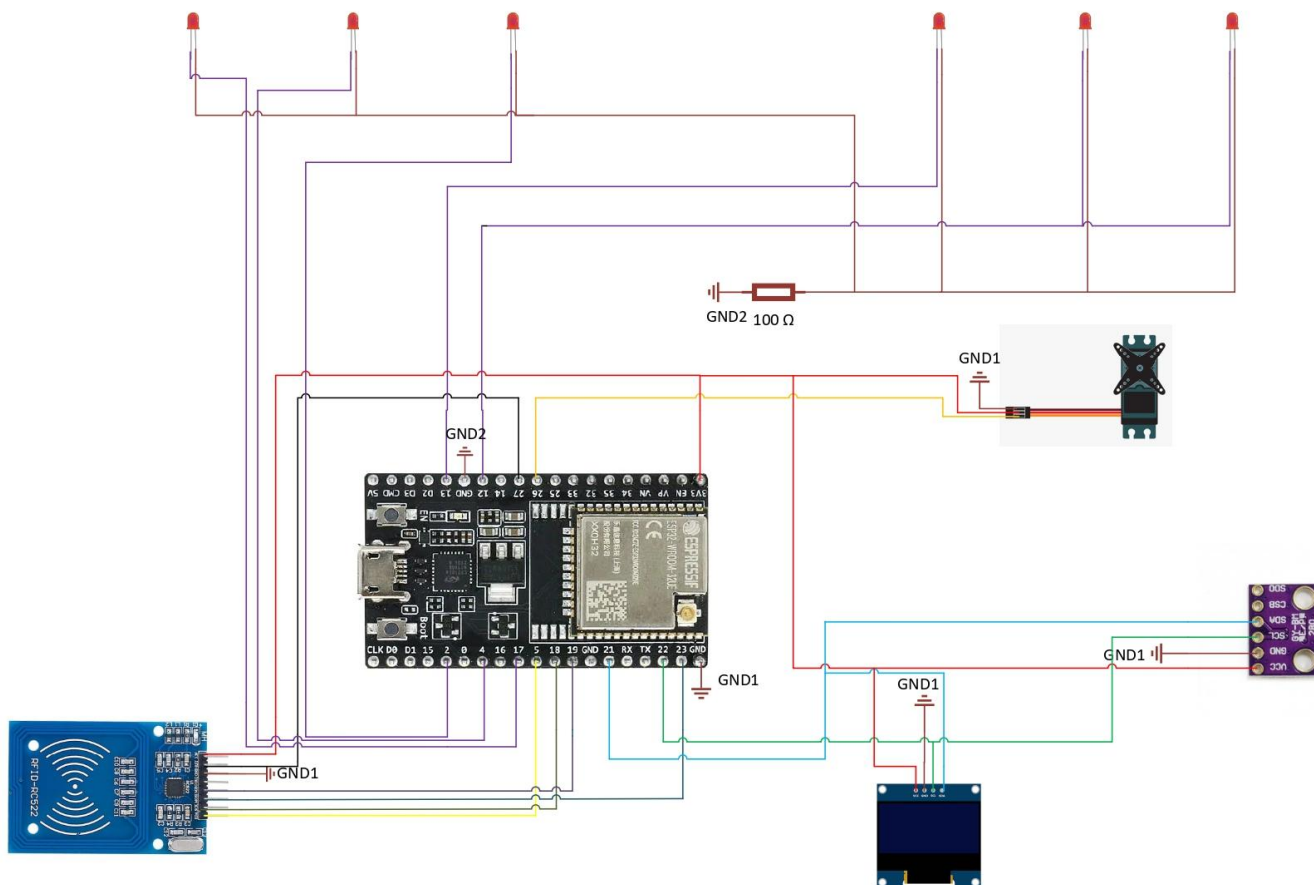
Pol Raich
Víctor Pàllas

Diagrama de blocs

ESQUEMA DE PINS

RFID522	
Vcc	3,3v
Rst	G27
GnD	GND1
MISO	G19
MOSI	G23
SCK	G18
SDA	G5

SSD1306	
Vcc	3,3v
GnD	GND1
SCL	G22
SDA	G21

SERVO SG90	
Vcc	3,3v
GND	GND1
SENYAL	G26

BMP280	
Vcc	3,3v
GND	GND1
SCL	G22
SDA	G21

CODI

HTML

```
String SendHTML(uint8_t LedSectorAStat, uint8_t LedSectorBStat, String targeta, float temperatura, uint8_t EstadoPuerta) {
  // Cabecera de todas las paginas WEB
  String ptr = "<!DOCTYPE html> <html>\n";

  // <meta> viewport. Para que la pagina se vea correctamente en cualquier dispositivo
  ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
  ptr += "<title>Control LED</title>\n";

/*
 * El estilo de la pagina WEB, tipo de letra, tamaño, colores,
 * El estilos de los botones (las clases en CSS) y la definicion de como van a cambiar dependiendo de como
 * cambien los estado de los LEDs, color fondo etc
 */
  ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto;}\n";
  ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3 {color: #444444;margin-bottom: 50px;}\n";
  ptr += "div#left{float: left; margin-left: 200px;}\n";
  ptr += "div#leftl{width: 300px; float: left; margin-left: 75px; margin-top: 20px;}\n";
  ptr += "div#right{float: right; margin-right: 200px;}\n";
  ptr += "div#rightr{width: 300px; float: right; margin-right: 75px; margin-top: 20px;}\n";
  ptr += "div#continuacio{clear: both;}\n";
  ptr += ".button {width: 50px;text-align: center;background-color: #3498db;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 18px;cursor: pointer;
  ptr += ".button-on {background-color: #3498db;}\n";
  ptr += ".button-on:active {background-color: #2980b9;}\n";
  ptr += ".button-off {background-color: #34495e;}\n";
  ptr += ".button-off:active {background-color: #2c3e50;}\n";
  ptr += "h1{text-align: center;}\n";
  ptr += "h3{text-align: center;}\n";
  ptr += "h5{font-size: 14px;color: #888;}\n";
  ptr += "h4{font-size: 16px; color #888; margin: 5px;}\n";
  ptr += "fieldset{text-align:center; color: black; border: solid 2px black;}\n";
  ptr += "p{float: left; margin-left: 50px; margin-right: 85px}\n";
  ptr += "</style>\n";
  ptr += "</head>\n";
  ptr += "<body>\n";
  /*
   * Encabezados de la pagina
```

```cpp
  ptr += "<div id=\"header\">\n";
  ptr += "<h1 align=\"center\">CASA</h1>\n";
  ptr += "<p><img src='https://lh3.googleusercontent.com/pw/AJFCJaVRWDJ4j-uNPBZANBcSTnUMzGjy8XL_0FHzj2-MYRR84BwW8pwHITB4-7dNBoj447y7HJxkfEmZ4VCqEKb2bVHnsCc-FmXq3ihvwbLNWfplw9C
  ptr += "<p><img src='https://lh3.googleusercontent.com/pw/AJFCJaXsi81TvsGtEWy91GqAYGXRzy7nhLSrEDwMfvfo2EfViuPNYj07y47e9fD5hOp2I09Xgywa6hnO4c1IUopY26w2-hSUFWVOmw_cW8WXm2WDnKV
  ptr += "</div>\n";
/*
 * Aqui esta la inteligencia del servidor WEB con ESP32, dependiento de los parametros de la funcion SendHTML
 * modificara la vista de la pagina WEB,  llamaran a las clases "button-on y button-off" que cambian como
 * se muestran los datos en la pagina WEB
 */
if (targeta == "ab540d41")
{
  ptr += "<fieldset><fieldset><h4>Luces</h4></fieldset>\n";

  ptr += "<div id=\"left\">\n";
  ptr += "<h5>Led's Sector A</h5>\n";
  if (LedSectorAStat)
  {
    ptr += "<a class=\"button button-on\" href=\"/LedSectorAoff\">ON</a>";
  }
  else
  {
    ptr += "<a class=\"button button-off\" href=\"/LedSectorAon\">OFF</a>";
  }
  ptr += "</div>\n";

  ptr += "<div id=\"right\">\n";
  ptr += "<h5>Led's Sector B</h5>\n";
  if (LedSectorBStat)
  {
    ptr += "<a class=\"button button-on\" href=\"/LedSectorBoff\">ON</a>";
  }
  else
  {
    ptr += "<a class=\"button button-off\" href=\"/LedSectorBon\">OFF</a>";
  }

  ptr += "<div id=\"leftl\">";
  ptr += "<fieldset><h4>Temperatura</h4></fieldset>\n";
  ptr += "<div id=\"leftl\">";
  ptr += "<a class=\"button button\" href=\"/Temp\">";
  ptr += temperatura;
  ptr += " C</a></div>\n";
  ptr += "</div>\n";

  ptr += "<div id=\"rightr\">\n";
  ptr += "<fieldset><h4>Puerta</h4></fieldset>\n";
  ptr += "<div id=\"leftl\">\n";
  if(EstadoPuerta)
  {
    ptr += "<a class=\"button button-on\" href=\"PuertaOff\">Abierta</a>";
  }
  else
  {
    ptr += "<a class=\"button button-off\" href=\"PuertaOn\">Cerrada</a>";
  }
  ptr +="</div>\n";

}
else
{
  ptr += "<fieldset><h4>Targeta no reconeguda</fieldset>";
}
  ptr += "</body>\n";
  ptr += "</html>\n";
  return ptr;
}
```

CODI

C++

```cpp
void handle_PuertaOn() {
  EstadoPuerta = HIGH;
  server.send(200, "text/html", SendHTML(LedSectorAStat, LedSectorBStat, targeta, temperatura, true)); // 3
}

void handle_PuertaOff() {
  EstadoPuerta = LOW;
  server.send(200, "text/html", SendHTML(LedSectorAStat, LedSectorBStat, targeta, temperatura, false)); // 3
}

void handle_Temp(){
  if(bmp280.takeForcedMeasurement()){
    temperatura=bmp280.readTemperature();
  }
  server.send(200, "text/html", SendHTML(LedSectorAStat, LedSectorBStat, targeta, temperatura, EstadoPuerta)); // 3
}

void handle_OnConnect() {
  LEDA2Estado = LOW; // 1
  LEDA3Estado = LOW;
  LEDB1Estado = LOW;
  LEDB2Estado = LOW;
  LEDB3Estado = LOW;
  LedSectorAStat = LOW;
  LedSectorBStat = LOW;
  if(bmp280.takeForcedMeasurement()){
    temperatura = bmp280.readTemperature();
  }
  server.send(200, "text/html", SendHTML(LedSectorAStat, LedSectorBStat, targeta, temperatura, EstadoPuerta)); // 3
}

void handle_LedSectorAon() {
  LEDA2Estado = HIGH; // 1
  LEDA3Estado = HIGH;
  LedSectorAStat = HIGH;
  if(bmp280.takeForcedMeasurement()){
```

```cpp
void loop() {
  server.handleClient();

  // Revisamos si hay nuevas tarjetas  presentes
  if ( mfrc522.PICC_IsNewCardPresent())
        {
      //Seleccionamos una tarjeta
          if ( mfrc522.PICC_ReadCardSerial())
          {
        String tar;
        tar = "";
        targeta = "";
            // Enviamos serialemente su UID
            for (byte i = 0; i < mfrc522.uid.size; i++) {
                  tar += String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
                  tar += String(mfrc522.uid.uidByte[i], HEX);

            }
        Serial.println();
        for (int i = 0; i < tar.length(); i++)
        {
          if(tar[i] == ' '){}
          else {
            targeta += tar[i];
          }
        }
        if(bmp280.takeForcedMeasurement()){
            temperatura = bmp280.readTemperature();
        }
        // Terminamos la lectura de la tarjeta  actual
        mfrc522.PICC_HaltA();
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.setCursor(0,0);

        display.print("\nUser: ");
```

```cpp
void handle_LedSectorAoff() {
  LEDA2Estado = LOW; // 1
  LEDA3Estado = LOW;
  LedSectorAStat = LOW;
  if(bmp280.takeForcedMeasurement()){
    temperatura = bmp280.readTemperature();
  }
  server.send(200, "text/html", SendHTML(false, LedSectorBStat, targeta, temperatura, EstadoPuerta));
}

void handle_LedSectorBon() {
  LEDB1Estado = HIGH;
  LEDB2Estado = HIGH;
  LEDB3Estado = HIGH;
  LedSectorBStat = HIGH;
  if(bmp280.takeForcedMeasurement()){
    temperatura = bmp280.readTemperature();
  }
  server.send(200, "text/html", SendHTML(LedSectorAStat, true, targeta, temperatura, EstadoPuerta));
}

void handle_LedSectorBoff() {
  LEDB1Estado = LOW;
  LEDB2Estado = LOW;
  LEDB3Estado = LOW;
  LedSectorBStat = LOW;
  if(bmp280.takeForcedMeasurement()){
    temperatura = bmp280.readTemperature();
  }
  server.send(200, "text/html", SendHTML(LedSectorAStat, false, targeta, temperatura, EstadoPuerta));
}

void handle_NotFound() {
  server.send(404, "text/plain", "La pagina no existe");
}
```

```
                    if(targeta == "ab540d41"){
                      display.println("Pol");
                      for(angle; angle < 180; angle++){
                        servo.write(angle);
                      }
                      EstadoPuerta = HIGH;
                    }
                    else {
                      display.println("Unknown");
                    }
                    server.send(200, "text/html", SendHTML(LedSectorAStat, LedSectorBStat, targeta, temperatura, EstadoPuerta));
                    display.print("\nID: ");
                    display.println(targeta);

                    display.display();
                    display.clearDisplay();
                  }
                }

if(EstadoPuerta) {
  for(angle; angle < 180; angle++){
    servo.write(angle);
  }
}
else{
  for(angle; angle > 60; angle--){
    servo.write(angle);
  }
}

if (LedSectorAStat)
{
  digitalWrite(LEDA2pin, HIGH);
  digitalWrite(LEDA3pin, HIGH);
}
```

```
else
{
  digitalWrite(LEDA2pin, LOW);
  digitalWrite(LEDA3pin, LOW);
}

if (LedSectorBStat)
{
  digitalWrite(LEDB1pin, HIGH);
  digitalWrite(LEDB2pin, HIGH);
  digitalWrite(LEDB3pin, HIGH);
}
else
{
  digitalWrite(LEDB1pin, LOW);
  digitalWrite(LEDB2pin, LOW);
  digitalWrite(LEDB3pin, LOW);
}
}
```