

Global Solution: Future at Work

WorkGuard Suite - AICSS



Integrantes:

- **Gabriel Tonelli - RM: 564705**
- **Vinícius Adrian - RM: 564962**
- **Luiz Henrique Poss - RM: 562177**

1. Introdução e Alinhamento ao Tema

Título: WorkGuard Suite: Inteligência e Privacidade no Futuro do Trabalho

Tema Central: Tecnologias Inteligentes para o Futuro do Trabalho (Segurança e Bem-Estar)

Contexto: O WorkGuard Suite é um sistema IoT de borda (ESP32) que resolve o dilema entre a necessidade de monitoramento da **Segurança Física (NR-12/NR-17)** e a **Privacidade de Dados (LGPD)** do colaborador. Ele garante um ambiente de trabalho seguro e ético, reagindo a riscos e anonimizando informações pessoais em tempo real.

2. Arquitetura e Componentes (Motivação)

A aplicação utiliza um conjunto de sensores e atuadores alinhados diretamente às Normas Regulamentadoras (NRs):

Componente	Pino	Função na Aplicação	Justificativa
Ultrassônico (Sensor)	5 / 18	Mede distância de áreas críticas (máquinas).	Segurança / NR-12. Detecção de risco de acidente por aproximação.
DHT22 (Sensor)	32	Coleta Temperatura e Umidade.	Bem-estar / NR-17. Alerta sobre desconforto ou risco de insalubridade.
LDR (Sensor)	34	Mede Luminosidade.	Ergonomia Visual. Previne fadiga ocular.
Potenciômetro	26	Simula Sensor de Gás/CO2.	Saúde Ambiental. Monitora a qualidade do ar interno (foco na produtividade).

LEDs R/A/V (Atuador)	16 / 17 / 27	Semáforo de Risco.	Alertas Multinível / Gestão à Vista.
LED AZUL (Inovação)	4	Indicador de Privacidade Ativa.	Inovação / LGPD. Alerta visual de que a anonimização está ligada.

3. Requisitos Técnicos e Demonstração

Critério	Implementação no Código e Pitch
Namespace	Cumprido com o namespace <code>WorkGuardSystem</code> (ou <code>MeuSistema</code>). Ele agrupa a coleta (<code>atualizarSensores</code>) e a lógica (<code>checarAlertas</code>).
Anonimização (Hash) & Botão	O <code>PINO_BOTAO</code> alterna o <code>modoAdmin</code> . No Modo Usuário (LGPD) , a função <code>anonimizar()</code> converte dados (Temp, CO2, Luz) em Hash Hexadecimal (ex: <code>TMP-2AFX</code>) no Dashboard Serial e LCD.
Alertas Multinível	Implementado no <code>switch case</code> da função <code>atualizarAtuadores()</code> . Possui 3 níveis: Seguro (Verde) , Atenção (Amarelo) e Perigo (Vermelho + Buzzer) .
Dashboard no Monitor Serial	O sistema exibe uma tabela textual formatada (`
Inovação	Privacidade por Design (LED Azul). O LED Azul (Pino 4) é ativado unicamente para indicar o modo de anonimização, oferecendo um feedback físico de <i>Compliance</i> .

4. Código-Fonte

```
/*
 * GLOBAL SOLUTION 2025.2 - WORKGUARD SUITE
 * TEMA: Segurança e Bem-estar no Trabalho (WorkGuard)
 * ---
 * O que cada pino faz:
 * [SENSORES]
 * - Ultrassônico: 5 (Trig) e 18 (Echo) -> Mede distância da máquina
 * - DHT22: 32 -> Mede Temperatura
 * - LDR: 34 -> Mede Luz
 * - Potenciômetro: 26 -> Simula Gás CO2
 *
 * [ATUADORES]
 * - LEDs: Vermelho(16), Amarelo(17), Verde(27) -> Semáforo de Risco
 * - LED AZUL: 4 -> Aviso de Privacidade (LGPD)
 * - Buzzer: 19 -> Alarme
 *
 * [CONTROLE]
 * - Botão: 35 -> Troca modo Admin/User
 * - LCD: 21 (SDA) e 22 (SCL)
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

// --- 1. DEFINIÇÕES DOS PINOS ---
#define PINO_TRIG 5
#define PINO_ECHO 18
#define PINO_DHT 32
#define TIPO_DHT DHT22
#define PINO_LDR 34
#define PINO_POT_CO2 26

// Atuadores
#define PINO_LED_VERMELHO 16
#define PINO_LED_AMARELO 17
#define PINO_LED_VERDE 27
#define PINO_BUZZER 19
#define PINO_LED_AZUL 4
```

```
// Entrada
#define PINO_BOTAO 35

// LCD
#define LCD_COLUNAS 20
#define LCD_LINHAS 4
#define LCD_ENDERECHO 0x27

// --- 2. OBJETOS ---
LiquidCrystal_I2C lcd(LCD_ENDERECHO, LCD_COLUNAS, LCD_LINHAS);
DHT dht(PINO_DHT, TIPO_DHT);

// --- 3. VARIÁVEIS GLOBAIS ---
bool modoAdmin = false;
bool estadoBotaoAntigo = HIGH;
unsigned long ultimoTempoClique = 0;
unsigned long intervaloMinimoClique = 250;

// --- 4. NAMESPACE ---
namespace WorkGuardSystem {

    float temperatura = 0.0;
    float umidade = 0.0;
    int luminosidade = 0;
    int distancia = 0;
    int nivelCO2 = 0;

    int nivelAlerta = 0;

    // Função de Anonimização (Hash Simples)
    String anonimizar(float valor, String prefixo) {
        long calc = (long)(valor * 55);
        String hash = String(calc, HEX);
        hash.toUpperCase();
        return prefixo + "-" + hash + "X";
    }

    // Função para ler o Ultrassônico
    int lerDistancia() {
```

```

digitalWrite(PINO_TRIG, LOW);
delayMicroseconds(2);
digitalWrite(PINO_TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(PINO_TRIG, LOW);
long duracao = pulseIn(PINO_ECHO, HIGH, 30000);
int cm = duracao * 0.034 / 2;
if (cm == 0 || cm > 400) return 400;
return cm;
}

// Função que lê todos os sensores
void atualizarSensores() {
    temperatura = dht.readTemperature();
    umidade = dht.readHumidity();
    luminosidade = map(analogRead(PINO_LDR), 0, 4095, 0, 100);
    distancia = lerDistancia();
    nivelCO2 = map(analogRead(PINO_POT_CO2), 0, 4095, 0, 100);

    if (isnan(temperatura)) temperatura = 0.0;
}

// Função que decide o nível de risco
void processarRiscos() {
    // REGRA 1: Risco Físico Imediato (NR-12)
    if (distancia < 30) {
        nivelAlerta = 2; // PERIGO
        return;
    }

    // REGRA 2: Risco à Saúde (NR-15 / NR-17)
    if (nivelCO2 > 70 || temperatura > 32.0) {
        nivelAlerta = 2; // PERIGO
    }
    // REGRA 3: Desconforto / Atenção (LDR INCLUÍDO)
    else if (nivelCO2 > 40 || temperatura > 28.0 || luminosidade < 20)
    {
        nivelAlerta = 1; // ATENÇÃO
    }
    // REGRA 4: Tudo OK
}

```

```

        else {
            nivelAlerta = 0; // SEGURO
        }
    }
} // Fim do Namespace

// --- 5. FUNÇÕES DE ATUALIZAÇÃO (Hardware) ---

void atualizarAtuadores() {
    digitalWrite(PINO_LED_VERDE, LOW);
    digitalWrite(PINO_LED_AMARELO, LOW);
    digitalWrite(PINO_LED_VERMELHO, LOW);
    digitalWrite(PINO_BUZZER, LOW);

    // Controle do LED AZUL (Privacidade)
    if (modoAdmin == false) {
        digitalWrite(PINO_LED_AZUL, HIGH);
    } else {
        digitalWrite(PINO_LED_AZUL, LOW);
    }

    // Semáforo de Segurança
    switch (WorkGuardSystem::nivelAlerta) {
        case 0:
            digitalWrite(PINO_LED_VERDE, HIGH);
            break;
        case 1:
            digitalWrite(PINO_LED_AMARELO, HIGH);
            break;
        case 2:
            digitalWrite(PINO_LED_VERMELHO, HIGH);
            if (millis() % 500 < 250) {
                digitalWrite(PINO_BUZZER, HIGH);
            }
            break;
    }
}

void atualizarLCD() {

```

```
// NENHUM LCD.CLEAR() AQUI

// LINHA 1: MODO
lcd.setCursor(0, 0);
if (modoAdmin) {
    lcd.print("ADMIN [DADOS REAIS]");
} else {
    lcd.print("LGPD [PROTEGIDO] ");
}

// LINHA 2: STATUS
lcd.setCursor(0, 1);
lcd.print("Status: ");
if (WorkGuardSystem::nivelAlerta == 2) {
    lcd.print("PERIGO! ");
} else if (WorkGuardSystem::nivelAlerta == 1) {
    lcd.print("ATENCAO ");
} else {
    lcd.print("SEGURO ");
}
lcd.print(" ");

// LINHA 3: TEMPERATURA E LUZ
lcd.setCursor(0, 2);
if (modoAdmin) {
    // Mostra Temp e Luz
    lcd.print("T:");
    lcd.print(WorkGuardSystem::temperatura, 0);
    lcd.print("C ");
    lcd.print("Luz:");
    lcd.print(WorkGuardSystem::luminosidade);
    lcd.print("%");
    lcd.print(" "); // Limpa o resto da linha
}
else {
    lcd.print("DADOS PROTEGIDOS "); // 20 chars
}

// LINHA 4: DISTÂNCIA
lcd.setCursor(0, 3);
```

```
if (modoAdmin) {
    // Mostra a distância
    lcd.print("Dist. Maq: ");
    lcd.print(WorkGuardSystem::distancia);
    lcd.print("cm");
    lcd.print("      "); // Limpa o resto da linha
}
else {
    // Esconde a distância (Anonimização Total)
    lcd.print("SEGURANCA ATIVA      "); // 20 chars
}
}

// --- SETUP ---
void setup() {
    Serial.begin(115200);

    pinMode(PINO_TRIG, OUTPUT);
    pinMode(PINO_ECHO, INPUT);
    pinMode(PINO_LED_VERMELHO, OUTPUT);
    pinMode(PINO_LED_AMARELO, OUTPUT);
    pinMode(PINO_LED_VERDE, OUTPUT);
    pinMode(PINO_LED_AZUL, OUTPUT);
    pinMode(PINO_BUZZER, OUTPUT);
    pinMode(PINO_BOTAO, INPUT_PULLUP);
    pinMode(PINO_LDR, INPUT);
    pinMode(PINO_POT_CO2, INPUT);

    dht.begin();
    lcd.init();
    lcd.backlight();

    lcd.setCursor(0,0);
    lcd.print("WorkGuard Suite");
    lcd.setCursor(0,1);
    lcd.print("Iniciando... ");
    delay(2000);

    lcd.clear();
```

```
}

// --- LOOP PRINCIPAL ---
void loop() {

    // --- 1. LÓGICA DE BOTÃO (FINAL) ---
    bool estadoBotaoAtual = digitalRead(PINO_BOTAO);

    if (estadoBotaoAtual == LOW && estadoBotaoAntigo == HIGH) {

        if (millis() - ultimoTempoClique > intervaloMinimoClique) {

            modoAdmin = !modoAdmin;
            ultimoTempoClique = millis();
        }
    }
    estadoBotaoAntigo = estadoBotaoAtual;

    // 2. Roda a Lógica
    WorkGuardSystem::atualizarSensores();
    WorkGuardSystem::processarRiscos();

    // 3. Atualiza o Hardware
    atualizarAtuadores();
    atualizarLCD();

    // 4. Imprime a Tabela no Serial

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

    Serial.println("=====");
    Serial.print(" --- PAINEL WORKGUARD --- | MODO: ");
    if(modoAdmin) Serial.println("ADMIN (MANUTENCAO)");
    else           Serial.println("USUARIO (PRIVACY)");

    Serial.println("=====");
}
```

```
    Serial.println("| SENSOR           | VALOR / HASH          | STATUS
| ") ;

Serial.println(" | ----- | ----- | -----
-- | ");

// Linha da Temperatura
Serial.print("| Temperatura      | ");
if(modoAdmin) {
    Serial.print(WorkGuardSystem::temperatura); Serial.print(" C
");
} else {

Serial.print(WorkGuardSystem::anonimizar(WorkGuardSystem::temperatura,
"TMP"));
    Serial.print("           ");
}
Serial.print(" | ");
if(WorkGuardSystem::temperatura > 28) Serial.println("ALERTA
|"); else Serial.println("OK           | ");

// Linha da Luminosidade
Serial.print("| Luminosidade     | ");
if(modoAdmin) {
    Serial.print(WorkGuardSystem::luminosidade); Serial.print(" %
");
} else {

Serial.print(WorkGuardSystem::anonimizar((float)WorkGuardSystem::luminosidade,
"LUZ"));
    Serial.print("           ");
}
Serial.print(" | ");
if(WorkGuardSystem::luminosidade < 20) Serial.println("BAIXA
|"); else Serial.println("ADEQUADA      | ");

// Linha da Distância (ANONIMIZADA)
Serial.print("| Dist. Maquina   | ");
if(modoAdmin) {
```

```

        Serial.print(WorkGuardSystem::distancia); Serial.print(" cm
");
    } else {

Serial.print(WorkGuardSystem::anonimizar((float)WorkGuardSystem::distancia
, "DST"));
    Serial.print("          ");
}
Serial.print(" | ");
if(WorkGuardSystem::distancia < 30) Serial.println("PERIGO!      | ");
else Serial.println("SEGURO      | ");

// Linha do CO2
Serial.print("| Qualidade Ar      | ");
if(modoAdmin) {
    Serial.print(WorkGuardSystem::nivelCO2); Serial.print(" % (CO2)
");
} else {

Serial.print(WorkGuardSystem::anonimizar(WorkGuardSystem::nivelCO2,
"GAS"));
    Serial.print("          ");
}
Serial.print(" | ");
if(WorkGuardSystem::nivelCO2 > 70) Serial.println("RUIM      | ");
else Serial.println("BOM      | ");

Serial.println("=====");
Serial.print("LED AZUL (LGPD): ");
if(!modoAdmin) Serial.println("LIGADO (DADOS PROTEGIDOS)");
else           Serial.println("DESLIGADO (MODO ADMIN)");

Serial.println("=====");
// Delay para estabilizar o LCD e o Botão.
delay(150);
}

```

diagram.json

```
{  
    "version": 1,  
    "author": "Luizinho Poss",  
    "editor": "wokwi",  
    "parts": [  
        {  
            "type": "wokwi-breadboard-half",  
            "id": "bb1",  
            "top": -262.2,  
            "left": -170,  
            "rotate": 90,  
            "attrs": {}  
        },  
        { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -153.6,  
        "left": 292.84, "attrs": {} },  
        {  
            "type": "wokwi-hc-sr04",  
            "id": "ultrasonic1",  
            "top": -478.5,  
            "left": 274.3,  
            "attrs": { "distance": "127" }  
        },  
        {  
            "type": "wokwi-dht22",  
            "id": "dht1",  
            "top": -489.3,  
            "left": 196.2,  
            "attrs": { "humidity": "26", "temperature": "46.5" }  
        },  
        {  
            "type": "wokwi-lcd2004",  
            "id": "lcd1",  
            "top": -243.2,  
            "left": 677.6,  
            "attrs": { "pins": "i2c" }  
        },  
        {  
            "type": "wokwi-led",  
            "id": "led1",  
            "top": -243.2,  
            "left": 750.0,  
            "rotate": 90,  
            "attrs": {}  
        }  
    ]  
}
```

```
"id": "led1",
"top": 18,
"left": 45,
"rotate": 90,
"attrs": { "color": "red" }

} ,
{
  "type": "wokwi-led",
  "id": "led2",
  "top": 46.8,
  "left": 45,
  "rotate": 90,
  "attrs": { "color": "yellow" }

} ,
{
  "type": "wokwi-led",
  "id": "led3",
  "top": 75.6,
  "left": 45,
  "rotate": 90,
  "attrs": { "color": "limegreen" }

} ,
{
  "type": "wokwi-resistor",
  "id": "r1",
  "top": -169.6,
  "left": -87.15,
  "rotate": 90,
  "attrs": { "value": "330" }

} ,
{
  "type": "wokwi-resistor",
  "id": "r2",
  "top": -134.4,
  "left": -83,
  "rotate": 90,
  "attrs": { "value": "330" }

} ,
{
  "type": "wokwi-resistor",
```

```
"id": "r3",
"top": -97.8,
"left": -83,
"rotate": 90,
"attrs": { "value": "330" }

} ,
{
  "type": "wokwi-buzzer",
  "id": "bz1",
  "top": 12,
  "left": 145.8,
  "attrs": { "volume": "0.1" }

} ,
{
  "type": "wokwi-pushbutton",
  "id": "btn1",
  "top": -166.6,
  "left": 604.8,
  "attrs": { "color": "green" }

} ,
{
  "type": "wokwi-photoresistor-sensor",
  "id": "ldr1",
  "top": -467.2,
  "left": 452,
  "attrs": { }

} ,
{
  "type": "wokwi-slide-potentiometer",
  "id": "pot1",
  "top": -71.8,
  "left": 642.2,
  "attrs": { "travelLength": "80" }

} ,
{
  "type": "wokwi-led",
  "id": "sirenLed",
  "top": -176.4,
  "left": 560.6,
  "attrs": { "color": "blue" }
```

```

} ,
{
  "type": "wokwi-resistor",
  "id": "r_siren",
  "top": -70.5,
  "left": -53.4,
  "rotate": 90,
  "attrs": { "value": "330" }
}
],
"connections": [
[ "esp:TX", "$serialMonitor:RX", "", [] ],
[ "esp:RX", "$serialMonitor:TX", "", [] ],
[ "esp:5V", "bb1:tp.1", "red", [ "v86.4", "h-412.65", "v-220.8", "h0",
"v-202.3" ] ],
[ "esp:GND.1", "bb1:tn.1", "black", [ "h0" ] ],
[ "bb1:tp.1", "bb1:bp.1", "red", [ "v-3.2" ] ],
[ "bb1:tn.1", "bb1:bn.1", "black", [ "v-3.2" ] ],
[ "ultrasonic1:VCC", "bb1:tp.2", "red", [ "v0" ] ],
[ "ultrasonic1:GND", "bb1:tn.2", "black", [ "v0" ] ],
[ "ultrasonic1:TRIG", "esp:5", "green", [ "v0" ] ],
[ "ultrasonic1:ECHO", "esp:18", "green", [ "v0" ] ],
[ "dht1:VCC", "bb1:tp.3", "red", [ "v0" ] ],
[ "dht1:GND", "bb1:tn.3", "black", [ "v0" ] ],
[ "dht1:SDA", "esp:32", "green", [ "v0" ] ],
[ "lcd1:SDA", "esp:21", "blue", [ "v0" ] ],
[ "lcd1:SCL", "esp:22", "blue", [ "v0" ] ],
[ "lcd1:VCC", "bb1:bp.25", "red", [ "h0" ] ],
[ "lcd1:GND", "bb1:bn.25", "black", [ "h0" ] ],
[ "r1:1", "led1:A", "red", [ "v0" ] ],
[ "r1:2", "esp:16", "green", [ "v0" ] ],
[ "led1:C", "bb1:bn.10", "black", [ "h0" ] ],
[ "r2:1", "led2:A", "red", [ "v0" ] ],
[ "r2:2", "esp:17", "green", [ "v0" ] ],
[ "led2:C", "bb1:bn.11", "black", [ "h0" ] ],
[ "r3:1", "led3:A", "red", [ "v0" ] ],
[ "r3:2", "esp:27", "green", [ "v0" ] ],
[ "led3:C", "bb1:bn.12", "black", [ "h0" ] ],
[ "bz1:2", "esp:19", "red", [ "h210.8", "v-153.6" ] ],
[ "bz1:1", "bb1:bn.13", "black", [ "v28.8", "h-174.4" ] ],

```

```
[ "btn1:1.r", "esp:35", "green", [ "v0" ] ],
[ "btn1:2.r", "bb1:bn.14", "black", [ "v29", "h-28.6", "v258.8",
"h-741" ] ],
[ "ldr1:VCC", "bb1:bp.4", "red", [ "v240", "h-796.8", "v316.8",
"h65.6" ] ],
[ "ldr1:GND", "bb1:bn.4", "black", [ "v162.8", "h-796.8", "v413.6",
"h65.6" ] ],
[ "ldr1:AO", "esp:34", "green", [ "v133.7", "h-278.55" ] ],
[ "pot1:VCC", "bb1:bp.2", "red", [ "v172.8", "h-739.2", "v-219.9" ] ],
[ "pot1:GND", "bb1:bn.2", "black", [ "v163.2", "h-1220.6", "v-413.5" ] ],
[ "pot1:SIG", "esp:26", "green", [ "v0" ] ],
[ "sirenLed:C", "bb1:bn.7", "black", [ "h0" ] ],
[ "sirenLed:A", "r_siren:1", "blue", [ "v268.8", "h-643.2", "v-228.9" ] ],
[ "r_siren:2", "esp:4", "orange", [ "v-163.2", "h-201.2" ] ],
[ "r1:1", "bb1:10t.c", "", [ "$bb" ] ],
[ "r1:2", "bb1:10b.g", "", [ "$bb" ] ],
],
"dependencies": {}  
}
```

Link para o pitch: <https://youtu.be/QYpTq7KOebo>