

```

using System;
class Program
{
    static void Main(string[] args)
    {
        // Referências
        double tempMin = 20.0, tempMax = 30.0; // Faixa de temperatura
        double umidMin = 50.0, umidMax = 70.0; // Faixa de umidade
        double gasMax = 400.0; // Limite aceitável de gás em ppm

        // Contadores
        int leiturasCorretasTemp = 0, leiturasTotaisTemp = 0;
        int leiturasCorretasUmid = 0, leiturasTotaisUmid = 0;
        int leiturasCorretasGas = 0, leiturasTotaisGas = 0;
        int buzzerCorretos = 0, buzzerTotais = 0;
        int ledCorretos = 0, ledTotais = 0;

        // Simulação de leituras (100 leituras para cada sensor)
        Random rand = new Random();
        for (int i = 0; i < 100; i++)
        {
            // Simula leituras do DHT11 com maior chance de estar na faixa correta
            double temp = rand.NextDouble() * (35 - 18) + 18; // Temperatura entre 18B0C e 35B0C
            double umid = rand.NextDouble() * (90 - 40) + 40; // Umidade entre 40% e 90%

            // Ajustando leituras para maior precisão
            if (rand.Next(0, 100) < 85) // 85% de chance de gerar leitura dentro do intervalo
            {
                temp = rand.NextDouble() * (tempMax - tempMin) + tempMin;
                umid = rand.NextDouble() * (umidMax - umidMin) + umidMin;
            }

            // Simula leitura do sensor de gás com maior precisão
            double gas = rand.NextDouble() * (800 - 200) + 200; // Gás entre 200 e 800 ppm
            if (rand.Next(0, 100) < 85) // 85% de chance de gerar leitura abaixo de 400 ppm
            {
                gas = rand.NextDouble() * gasMax;
            }

            // Simulações do buzzer e LED com maior acur!cia
            bool buzzer = rand.Next(0, 100) < 85; // 85% de chance de estar correto
            bool led = rand.Next(0, 100) < 85; // 85% de chance de estar correto

            // Verifica acurácia do DHT11 (Temperatura)
            leiturasTotaisTemp++;
            if (temp >= tempMin && temp <= tempMax)
                leiturasCorretasTemp++;

            // Verifica acurácia do DHT11 (Umidade)
            leiturasTotaisUmid++;
            if (umid >= umidMin && umid <= umidMax)
                leiturasCorretasUmid++;

            // Verifica acurácia do MQ-2 (Gás)
            leiturasTotaisGas++;
            if (gas <= gasMax)
                leiturasCorretasGas++;

            // Verifica acurácia do Buzzer
            buzzerTotais++;
            if (buzzer)
                buzzerCorretos++;
        }
    }
}

```

```
// Verifica acurácia do LED
ledTotais++;
if (led)
    ledCorretos++;
}

// Calcula acurácia
double acuraciaTemp = (double)leiturasCorretasTemp / leiturasTotaisTemp * 100.0;
double acuraciaUmid = (double)leiturasCorretasUmid / leiturasTotaisUmid * 100.0;
double acuraciaGas = (double)leiturasCorretasGas / leiturasTotaisGas * 100.0;
double acuraciaBuzzer = (double)buzzerCorretos / buzzerTotais * 100.0;
double acuraciaLed = (double)ledCorretos / ledTotais * 100.0;

// Exibe resultados
Console.WriteLine("Acurácia dos Sensores:");
Console.WriteLine($"DHT11 (Temperatura): {acuraciaTemp:F2}%");
Console.WriteLine($"DHT11 (Umididade): {acuraciaUmid:F2}%");
Console.WriteLine($"MQ-2 (Gases): {acuraciaGas:F2}%");
Console.WriteLine($"Buzzer: {acuraciaBuzzer:F2}%");
Console.WriteLine($"LED: {acuraciaLed:F2}%");
}
}
```