

```
public double CalcularDiferencaConsumo(double consumoAtual, double consumoAnterior)
{
    return consumoAtual - consumoAnterior;
}
```

```
public void AtualizarLblComparacao()
{
    double atual = 350.0;
    double anterior = 300.0;

    double diferenca = CalcularDiferencaConsumo(atual, anterior);

    Lbl_Comparacao.Text = "Diferença de consumo: " + diferenca + " kWh";
}
```

### **Interface:**

```
this.panel4.BackColor = System.Drawing.Color.FromArgb(37, 42, 64);
this.panel4.Controls.Add(this.label4);
this.panel4.Controls.Add(this.Lbl_TerminarDepois);
// ... outros componentes
```

```
private void Btn_Dashboard_Click(object sender, EventArgs e)
{
    Pnl_Nav.Height = Btn_Dashboard.Height;
    Pnl_Nav.Top = Btn_Dashboard.Top;
    Btn_Dashboard.BackColor = Color.FromArgb(46, 51, 73);
}
```

### **Função para calcular o consumo:**

```
public double CalcularConsumo(double leituraAtual, double leituraAnterior)
{
    return leituraAtual - leituraAnterior;
}
```

### **Procedimento para atualizar o Label com o consumo:**

```
public void AtualizarLabelConsumo(double consumo)
{
    Lbl_Comparacao.Text = "Consumo atual: " + consumo.ToString("0.00") + " kWh";
}
```

### **Chamada dos métodos no Load ou Click:**

```
private void Frm_Dashboard_Load(object sender, EventArgs e)
{
    double leituraAtual = 1500; // Exemplo
    double leituraAnterior = 1300; // Exemplo

    double consumo = CalcularConsumo(leituraAtual, leituraAnterior);
    AtualizarLabelConsumo(consumo);
}
```

```
}
```

## **Cálculo de Fatorial, Verificação de Número Primo e Soma de Números Pares até N:**

```
// Procedimento para exibir o resultado na tela
public void MostrarResultado(string mensagem)
{
    MessageBox.Show(mensagem, "Resultado");
}
```

```
// Função para calcular fatorial
public int CalcularFatorial(int n)
{
    int resultado = 1;
    for (int i = 1; i <= n; i++)
    {
        resultado *= i;
    }
    return resultado;
}
```

```
// Função para verificar se um número é primo
public bool EhPrimo(int n)
{
    if (n < 2)
        return false;

    for (int i = 2; i <= Math.Sqrt(n); i++)
    {
        if (n % i == 0)
            return false;
    }
    return true;
}
```

```
// Função para somar números pares até N
public int SomarPares(int n)
{
    int soma = 0;
    for (int i = 2; i <= n; i += 2)
    {
        soma += i;
    }
    return soma;
}
```

```
// Evento do botão (exemplo)
```

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    int numero = int.Parse(txtNumero.Text);

    int fatorial = CalcularFatorial(numero);
    MostrarResultado("Fatorial: " + fatorial.ToString());

    bool primo = EhPrimo(numero);
    MostrarResultado("É primo? " + (primo ? "Sim" : "Não")); // Usar if comum, sem ternário,
    se preferir.

    int somaPares = SomarPares(numero);
    MostrarResultado("Soma dos pares até " + numero + ": " + somaPares);
}
```