```cpp
void calibrador(bool calibrarLineas = false, bool sinConfirmar = false)
{
    int ctr = 0;
    Spectrum avgMic, avgLeft, avgRight;
    EstadoDelLimitador status;
    Calibracion cal;

    cal.leeCalibracion(1);
    printf("Line 1\n");
    cal.print();

    cal.leeCalibracion(2);
    printf("Line 2\n");
    cal.print();

    ...

    while (true)
    {
        sleep(1);

        status.actualiza(); //Si el sistema no está funcionando sale

        avgMic.sum(status.mic);
        avgLeft.sum(status.left);
        avgRight.sum(status.right);

        printf(" . ");
        ctr++;

        status.mic.show();
        status.left.showStereo(status.right);

        printf("Mic : %8.1f dBA\tLeft : %8.1f dBA\tRight : %8.1f\n",
            status.mic.globalAWeighted,
            status.left.globalAWeighted,
            status.right.globalAWeighted
        );

        if (ctr >= 10)
            break;
    }

    if (calibrarLineas)
    {
        puts("Lineas de entrada calibradas");

        ...

        avgMic.divide(ctr);
        avgLeft.divide(ctr);
        avgRight.divide(ctr);

        //Guardar calibracion izquierda
        cal.dBRef = avgMic.dB[Spectrum_1Khz_BandIndex];
        cal.ref = avgLeft.energy[Spectrum_1Khz_BandIndex];
        cal.ruido = 0;
        for (int i = 0; i < Spectrum_BandCount; i++)
            cal.equalization[i] = 0; //avgMic.dB[i]-avgLeft[i]
        cal.guardaCalibracion(1);

        cal.print();

        //Guardar calibracion derecha
        cal.dBRef = avgMic.dB[Spectrum_1Khz_BandIndex];
        cal.ref = avgRight.energy[Spectrum_1Khz_BandIndex];
        cal.ruido = 0;
        for (int i = 0; i < Spectrum_BandCount; i++)
            cal.equalization[i] = 0;
        cal.guardaCalibracion(2);

        cal.print();
    }
}
```