Días Lluvia

```java
public class DiasLluvia {  2 usages
    private boolean[][] any = new boolean[12][31];  8 usages

    public boolean registroDia(int dia, int mes, boolean lluvia) {  14 usages
        if (dia < 1 || mes < 1 || mes > 12 || dia > diasEnMes(mes)) {
            System.err.println("NO SE HA PODIDO REGISTRAR EL DIA DESEADO: Fecha fuera de rango.");
            return false;
        }
        any[mes - 1][dia - 1] = lluvia;
        return true;
    }

    public boolean consultarDia(int dia, int mes) { return any[mes - 1][dia - 1]; }

    public int contarDiasLluviosos() {  3 usages
        int cont = 0;
        for (int i = 0; i < any.length; i++) {
            for (int j = 0; j < any[i].length; j++) {
                if (any[i][j])
                    cont++;
            }
        }
        return cont;
    }
```

```java
public int trimestreLluvioso() {  3 usages
    int[] trimestres = new int[4];
    for (int mes = 1; mes <= 12; mes++) {
        int n = (mes - 1) / 3;
        for (int dia = 1; dia <= diasEnMes(mes); dia++) {
            if (consultarDia(dia, mes)) {
                trimestres[n]++;
            }
        }
    }

    int maxTrim = 0;
    int maxDias = trimestres[0];
    for (int i = 1; i < trimestres.length; i++) {
        if (trimestres[i]>maxDias) {
            maxDias = trimestres[i];
            maxTrim = i;
        }
    }

    return maxTrim + 1;
}
```

```java
public int primerDiaLluvia() {  3 usages
    int contador=0;
    for (int i = 0; i < any.length; i++) {
        for (int j = 0; j < any[i].length; j++) {
            contador++;
            if (any[i][j]) {
                return contador;
            }
        }
    }
    return -1;
}


private static int diasEnMes(int mes) {  2 usages
    switch (mes) {
        case 2:
            return 28;
        case 4:
        case 6:
        case 9:
        case 11:
            return 30;
        default:
            return 31;
    }
}
```

Tests

```java
class DiasLluviaTest {
    private DiasLluvia DL = new DiasLluvia();  26 usages

    @Test
    void testRegistroDia() {
        assertEquals( expected: true, DL.registroDia( dia: 1,  mes: 8,  lluvia: true));
        assertEquals( expected: true, DL.registroDia( dia: 2,  mes: 8,  lluvia: false));
        assertNotEquals( unexpected: true, DL.registroDia( dia: 35,  mes: 8,  lluvia: true));
    }

    @Test
    void testConsultarDia() {
        DL.registroDia( dia: 2,  mes: 8,  lluvia: false);
        DL.registroDia( dia: 3, mes: 8,  lluvia: true);
        assertEquals( expected: true, DL.consultarDia( dia: 3, mes: 8));
        assertEquals( expected: false, DL.consultarDia( dia: 2, mes: 8));
        assertNotEquals( unexpected: false, DL.consultarDia( dia: 3, mes: 8));
    }
```

```java
    @Test
    void testContarDiasLluviosos() {
        DL.registroDia( dia: 1,  mes: 8,  lluvia: true);
        DL.registroDia( dia: 3, mes: 8,  lluvia: true);
        assertEquals( expected: 2, DL.contarDiasLluviosos());
        DL.registroDia( dia: 6, mes: 12,  lluvia: true);
        assertEquals( expected: 3, DL.contarDiasLluviosos());
        assertNotEquals( unexpected: 5, DL.contarDiasLluviosos());
    }

    @Test
    void testTrimestreLluvioso() {
        DL.registroDia( dia: 1,  mes: 8,  lluvia: true);
        DL.registroDia( dia: 3, mes: 8,  lluvia: true);
        assertEquals( expected: 3, DL.trimestreLluvioso());
        DL.registroDia( dia: 11, mes: 3,  lluvia: true);
        DL.registroDia( dia: 1, mes: 2,  lluvia: true);
        assertEquals( expected: 1, DL.trimestreLluvioso());
        assertNotEquals( unexpected: 4, DL.trimestreLluvioso());
    }

    @Test
    void testPrimerDiaLluvia() {
        DL.registroDia( dia: 1, mes: 2,  lluvia: true);
        assertEquals( expected: 32, DL.primerDiaLluvia());
        DL.registroDia( dia: 16, mes: 1,  lluvia: true);
        assertEquals( expected: 16, DL.primerDiaLluvia());
        assertNotEquals( unexpected: 135, DL.primerDiaLluvia());
    }
```