

EJERCICIO TEÓRICO DE TESTING PARA PROBLEMA 3

Método principal para testear:

```
public static Recomendacion recomendar(Persona p, Tiempo t, Map<TipoActividad, Actividad> actividades) throws NingunaActividadException, AforoSuperadoException {
    List<Actividad> recomendaciones = new ArrayList<>();
    Recomendacion r = new Recomendacion(recomendaciones);
    if(p.puedeRealizarActividadFisica()){
        if(t.getTemperatura() < 0 && t.getHumedad() < 15 && t.hayPrecipitacion()){
            Actividad a = actividades.get(TipoActividad.QUEDARSE_EN_CASA);
            r.addRecomendacion(a);
        }
        if(t.getTemperatura() < 0 && t.getHumedad() < 15 && !(t.hayPrecipitacion())){
            Actividad a = actividades.get(TipoActividad.ESQUI);
            comprobarAforo(a);
            r.addRecomendacion(a);
        }
        if(t.getTemperatura() ≥ 0 && t.getTemperatura() ≤ 15 && !t.hayPrecipitacionAgua()){
            Actividad a = actividades.get(TipoActividad.SENDERISMO);
            Actividad b = actividades.get(TipoActividad.ESCALADA);
            comprobarAforo(a);
            r.addRecomendacion(a);
            comprobarAforo(b);
            r.addRecomendacion(b);
        }
        if(t.getTemperatura() ≥ 15 && t.getTemperatura() ≤ 25 && !t.hayPrecipitacionAgua() && !t.isNublado() && t.getHumedad() ≤ 60){
            Actividad a = actividades.get(TipoActividad.CATALOGO_PRIMAVERA_VERANO_OTONO);
            comprobarAforo(a);
            r.addRecomendacion(a);
        }
        if(t.getTemperatura() ≥ 25 && t.getTemperatura() ≤ 35 && !t.hayPrecipitacionAgua()){
            Actividad a = actividades.get(TipoActividad.GASTRONOMICO);
            Actividad b = actividades.get(TipoActividad.CULTURAL);
            comprobarAforo(a);
            r.addRecomendacion(a);
            comprobarAforo(b);
            r.addRecomendacion(b);
        }
        if(t.getTemperatura() > 35 && !t.hayPrecipitacionAgua()){
            Actividad a = actividades.get(TipoActividad.PLAYA);
            Actividad b = actividades.get(TipoActividad.PISCINA);
            r.addRecomendacion(a);
            comprobarAforo(b);
            r.addRecomendacion(b);
        }
    } else {
        throw new NingunaActividadException(message: "La persona no puede realizar actividad física.");
    }
    return r;
}
```

Métodos relacionados a tener en cuenta:

```
public static void comprobarAforo(Actividad a) throws AforoSuperadoException {
    if (a.isAforoSuperado()) {
        throw new AforoSuperadoException("Aforo de la actividad " + a.getTipo() + " superado.");
    }
}

public boolean puedeRealizarActividadFisica() {
    return facultadesFisicas && !sintomasEnfermedades;
}

public boolean hayPrecipitacion() {
    return precipitacionAgua || precipitacionNieve;
}
```

Identificación de los parámetros de entrada del método:

Persona:

facultadesFisicas y sintomasEnfermedades

Actividad:

aforoSuperado

Tiempo:

temperatura, humedad, precipitacionNieve, precipitacionAgua, nublado

Identificación de valores de prueba para los parámetros

Parámetro	Clases de Equivalencia	Valores de Prueba
FacultadesFisicas: bool	True, False	Clases de equivalencia: {True, False}
sintomasEnfermedads: bool	True, False	Clases de equivalencia: {True, False}
aforoSuperado: bool	True, False	Clases de equivalencia: {True, False}
temperatura: int	($-\infty$, 0], [0, 15], [15,25], [25, 35], (30, ∞)	Clases de equivalencia: {-2, 10, 20, 27, 50} Valores Límite: {-1, 0, 1, 14, 15, 16, 24, 25, 26, 29, 30, 31, 34, 35, 36} Conjetura de errores: { $2 \cdot 10^7$, "e", "", " "}
humedad: int	($-\infty$, 0], [0, 15], [16, 60], [61, ∞)	Clases de equivalencia: {-2, 10, 20, 70} Valores Límite: {-1, 0, 1, 14, 15, 16, 17, 59, 60, 61, 62} Conjetura de errores: { $2 \cdot 10^7$, "e", "", " "}
precipitacionNieve: bool	True, False	Clases de equivalencia: {True, False}
precipitacionAgua: bool	True, False	Clases de equivalencia: {True, False}
Nublado: bool	True, False	Clases de equivalencia: {True, False}

Máximo número de casos de prueba

En primer lugar, para calcular el número máximo posible de casos de pruebas que se podrían generar a partir de los valores de pruebas, deberemos multiplicar la cantidad de valores de prueba de cada parámetro, de tal forma que tenemos:

facultadesFisicas: 2 valores de prueba

sintomasEnfermedades: 2 valores de prueba

aforoSuperado: 2 valores de prueba

temperatura: 24 valores de prueba

humedad: 19 valores de prueba

precipitacionNieve: 2 valores de prueba

precipitacionAgua: 2 valores de prueba

nublado: 2 valores de prueba

Por lo que:

$$2^6 * 24 * 19 = \mathbf{29.184} \text{ casos de prueba como máximo}$$

Casos de prueba para cumplir “Each-Use”

C P	facultade sFisicas	sintomasEnf ermedades	aforoSu perado	tempe ratura	hum edad	precipitac ionNieve	precipitac ionAgua	nubl ado
1	True	False	True	-2	-2	False	True	False
2	False	True	False	10	10	True	False	True
3	True	False	True	20	20	False	True	False
4	False	True	False	27	70	True	False	True
5	True	False	True	50	-1	False	True	False
6	False	True	False	-1	0	True	False	True
7	True	False	True	0	1	False	True	False
8	False	True	False	1	14	True	False	True

9	True	False	True	14	15	False	True	False
10	False	True	False	15	16	True	False	True
11	True	False	True	16	17	False	True	False
12	False	True	False	24	59	True	False	True
13	True	False	True	25	60	False	True	False
14	False	True	False	26	61	True	False	True
15	True	False	True	29	62	False	True	False
16	False	True	False	30	$2 \cdot 10^7$	True	False	True
17	True	False	True	31	"e"	False	True	False
18	False	True	False	34	" "	True	False	True
19	True	False	True	35	" "	False	True	False
20	False	True	False	36	-2	True	False	True
21	True	False	True	$2 \cdot 10^7$	10	False	True	False
22	False	True	False	"e"	20	True	False	True
23	True	False	True	" "	27	False	True	False
24	False	True	False	" "	50	True	False	True

Conjunto de pruebas para alcanzar pairwise

facultadesFisicas	sintomasEnfermedades	aforoSuperado	temperatura
humedad	precipitacionNieve	precipitacionAgua	nublado
True	True	False	36
60	False	True	True

True	False	True	50	-2	True	False	False
False	False	False	20000000	EMPTY	True	True	False
False	True	True	15	10	False	False	True
False	True	True	26	-1	True	True	True
True	True	False	31	62	False	False	False
True	False	True	27	SPACE	False	True	True
False	True	False	14	-2	False	True	True
False	False	True	25	EURO	False	False	True
True	False	False	-1	10	True	True	False
True	False	True	35	1	True	True	False
False	True	False	27	20	True	False	False
False	True	False	EURO	1	False	False	True
False	False	True	31	15	True	True	True
False	False	False	SPACE	61	False	True	False
True	False	False	15	16	True	True	False
True	True	True	31	EMPTY	False	False	True
True	True	False	24	15	False	False	False
False	False	True	1	60	True	False	False
True	True	True	20000000	60	False	False	True

True	False	True	EMPTY20		False	True	True
True	True	True	15	1	False	True	True
False	False	True	24	20000000	True	True	True
False	True	True	31	16	False	False	True
False	True	False	25	SPACE	True	False	False
True	False	False	26	16	False	False	False
True	True	False	27	1	False	False	False
True	True	False	29	0	True	True	False
False	True	False	35	17	False	False	True
True	False	True	14	20000000	True	False	False
False	False	True	27	-2	True	False	False
False	False	True	26	70	True	True	False
False	True	True	-1	SPACE	False	False	True
True	True	False	-2	20000000	False	False	False
True	True	True	SPACE10		True	False	True
False	True	True	34	15	True	False	False
False	False	True	30	15	True	False	False
True	False	True	EURO	20	True	True	False
False	True	False	27	16	False	True	False

False	False	True	10	62	True	True	True
True	False	True	25	20	False	True	False
True	True	True	14	61	True	False	True
True	False	True	0	60	True	True	True
False	True	False	EMPTY	EMPTY	True	False	False
True	False	True	EURO	14	True	False	False
False	True	False	0	1	False	False	False
True	True	False	25	61	True	True	False
False	False	True	29	SPACE	False	False	True
False	False	False	36	14	False	False	False
True	True	False	14	62	False	False	False
False	False	True	EURO	16	False	True	False
False	False	True	16	60	True	False	True
False	False	True	15	20000000	False	True	False
False	True	True	25	14	True	True	True
True	True	False	20000000	14	True	False	True
True	False	True	31	59	False	False	True
True	False	False	20000000	20	True	False	True
True	True	True	20000000	SPACE	True	False	False

True	False	True	31	20000000	True	False	True
False	True	False	20000000	20000000	True	True	False
False	True	True	29	14	True	True	True
True	True	False	16	SPACE	False	True	False
True	True	False	20000000	70	False	False	True
False	True	False	EMPTY	59	True	True	False
False	False	False	0	EMPTY	True	True	True
False	False	True	29	59	True	False	True
True	True	True	31	20	True	True	False
True	False	False	34	61	False	True	True
True	True	False	30	70	False	True	True
True	False	True	25	60	True	False	True
False	True	False	50	10	False	True	True
True	True	True	36	20	True	True	False
False	False	False	10	0	False	False	True
True	True	False	10	10	True	False	False
True	False	False	14	14	True	False	False
False	False	False	20	20000000	False	True	True
False	False	False	30	60	False	False	True

False	False	False	-1	60	True	False	True
True	False	False	0	20	True	False	True
True	True	False	16	59	True	False	True
False	False	False	0	15	True	False	True
True	True	False	34	59	True	False	True
False	True	False	-1	15	False	False	True
True	True	True	25	62	True	True	True
False	True	True	36	10	True	False	False
False	True	True	31	61	False	False	True
True	True	True	30	62	True	True	True
False	True	True	10	-2	False	False	True
True	True	False	14	EURO	True	True	False
True	True	False	1	-2	False	True	True
False	False	True	14	15	False	False	False
True	False	True	36	17	True	True	False
True	True	True	20	1	True	False	False
False	True	False	SPACE	SPACE	True	False	True
True	True	False	36	61	True	True	False
False	False	True	30	20	False	False	True

False	False	False	EMPTYSPACE	False	True	True
True	True	True	10	1	True	True
True	True	False	14	17	False	True
False	True	True	EURO	59	False	False
True	True	True	1	SPACE	True	True
True	False	True	-2	-1	True	False
True	False	True	35	SPACE	False	True
False	True	False	26	20	True	True
True	True	False	34	20	False	True
True	False	False	24	20	True	False
False	False	True	EMPTY1	True	False	False
False	True	True	16	70	False	True
True	True	False	16	20	False	True
False	False	True	24	60	True	False
False	True	True	20	15	False	True
True	False	True	EMPTY14	True	False	False
True	False	True	20	60	True	True
False	True	True	-2	SPACE	False	True
False	False	True	34	60	True	True

True	False	True	0	0	False	False	False
True	True	True	EMPTY-2		False	True	True
False	True	False	16	61	True	True	False
False	True	False	SPACEEMPTY	True	True	False	
True	True	True	16	EMPTY	True	False	False
False	True	False	29	17	False	False	True
False	False	False	-1	16	True	False	True
True	True	True	EURO	60	True	False	False
True	False	True	24	1	False	True	False
False	True	True	0	70	True	True	False
True	True	True	30	16	True	True	True
False	False	False	14	-1	False	True	True
True	False	False	26	60	False	True	True
True	False	True	-2	60	True	False	False
True	True	False	-1	62	True	False	True
True	False	True	25	20000000	True	False	False
False	True	False	SPACE	62	False	True	False
False	True	True	30	59	False	True	False
False	False	False	34	EURO	False	False	False

True	False	True	20	SPACE	False	False	False
False	True	False	20	-1	True	True	False
True	True	False	16	62	False	True	True
False	True	True	SPACE	20	False	True	False
True	False	False	50	1	False	False	True
True	True	False	26	10	True	False	True
True	False	True	29	62	True	True	True
True	False	True	1	15	True	True	True
False	True	True	-2	10	False	True	False
False	False	True	24	14	True	False	True
False	True	True	26	14	False	True	True
False	True	True	29	15	False	False	True
False	False	True	50	SPACE	False	True	False
False	False	False	29	20000000	True	True	True
True	False	False	26	15	False	True	False
False	True	True	34	1	True	False	True
True	True	True	34	10	False	True	False
True	False	True	36	0	False	False	True
False	True	False	27	60	True	True	False

True	False	False	20	61	False	False	False
False	True	False	34	-1	False	False	True
True	True	True	30	61	False	True	False
False	False	False	16	10	True	True	False
False	True	False	-1	-1	False	True	True
True	False	True	50	61	False	True	False
False	False	False	EMPTY20000000		False	True	False
True	False	False	10	20000000	True	False	True
True	False	True	31	0	False	True	False
False	False	False	25	0	True	False	False
True	False	True	25	EMPTYTrue	True	True	
True	False	False	35	59	False	True	True
False	True	True	14	SPACEFalse	False	False	
False	True	True	29	EMPTYTrue	False	False	
True	False	False	36	59	True	True	True
True	True	True	27	17	True	False	True
False	True	True	SPACE1		True	True	False
False	False	True	20	16	True	True	True
False	True	False	-1	EMPTYFalse	True	False	

False	False	False	50	EURO	True	False	False
True	False	False	-1	20000000	True	False	True
False	True	True	50	20	False	True	False
False	True	False	20	EURO	False	True	True
True	False	False	14	1	True	False	False
False	True	True	SPACE70		True	True	False
False	True	False	36	-1	True	False	False
False	False	False	20000000	16	True	True	True
False	False	False	14	0	True	True	True
False	True	False	27	62	True	False	True
False	False	False	1	-1	False	False	True
True	True	False	27	70	False	True	False
False	False	True	20000000	15	False	False	True
False	False	True	SPACE-1		False	True	False
True	False	True	EMPTYEURO	True	False	True	
True	True	False	29	20	True	True	True
False	True	True	29	16	False	False	False
False	False	True	16	16	True	True	False
True	False	True	20	-2	True	False	True

True	True	False	10	SPACE	False	False	True
False	False	True	EMPTY	10	False	False	False
True	False	False	-2	61	True	False	False
False	False	False	EURO	EMPTY	True	False	False
False	False	False	31	70	True	True	True
False	False	False	24	SPACE	False	False	True
True	True	False	20000000	10	False	False	True
True	False	True	SPACE-2		True	False	False
True	False	True	EMPTY-1		False	False	False
True	False	False	-2	20	False	False	True
False	True	False	30	10	False	True	False
True	False	False	14	59	False	False	False
True	True	True	1	20	False	True	True
True	True	True	31	17	True	True	False
True	False	False	20	0	True	True	False
False	False	True	50	62	True	False	True
False	True	False	25	15	False	True	True
True	False	False	-1	59	False	False	False
False	True	False	1	70	False	False	True

True	False	False	-1	70	False	False	True
True	False	False	24	61	False	False	False
False	False	True	EURO	62	False	False	True
True	True	False	35	20	True	True	False
False	True	True	EURO	SPACE	True	False	False
False	False	False	25	59	False	False	False
True	True	True	16	-1	False	True	False
False	True	False	35	16	True	False	False
False	True	True	EURO	-1	True	True	False
True	True	True	30	0	True	False	True
True	False	True	50	15	True	False	True
False	False	True	20	14	False	True	True
False	False	False	25	-1	False	True	True
True	True	False	20	10	True	True	False
False	True	True	-1	17	True	True	False
False	False	False	-2	EMPTY	True	False	False
False	True	True	35	70	True	False	False
False	False	False	20	59	False	True	True
False	True	True	15	17	False	True	True

True	True	False	27	59	True	False	False
True	False	True	24	62	False	False	True
False	True	True	16	1	True	True	False
True	True	False	-2	16	False	True	False
False	True	True	35	20000000	True	True	True
True	True	True	20000000	59	False	True	False
True	True	True	-1	EURO	False	False	False
True	True	True	15	0	True	False	False
False	True	False	15	EMPTY	True	True	False
True	False	False	1	14	False	True	True
False	False	False	29	EURO	True	True	False
True	True	False	30	EURO	True	False	True
False	True	True	0	-1	False	True	False
True	False	True	34	SPACE	True	False	False
True	True	True	26	62	True	False	False
True	False	True	30	-2	True	True	False
False	True	True	20	62	False	False	False
True	False	True	10	EURO	False	False	False
False	True	False	SPACE	14	False	True	False

True	True	True	24	EURO	False	False	False
False	True	False	0	10	True	True	True
True	False	False	20	70	False	True	False
True	False	True	-2	15	True	False	False
False	True	True	29	1	False	False	False
True	True	True	34	0	True	False	True
False	False	True	29	10	False	True	False
True	False	True	-2	EURO	True	True	True
True	False	True	SPACE0		False	True	False
False	False	True	26	SPACE	False	False	True
False	False	False	30	17	False	True	True
False	False	False	EURO	EURO	False	False	False
True	False	False	16	EURO	False	True	True
False	False	True	36	EURO	False	False	True
True	False	False	-2	1	True	True	False
False	True	False	0	EURO	True	True	False
True	False	True	29	70	False	False	False
True	True	False	10	-1	True	True	False
True	True	True	29	60	False	False	True

True	False	True	36	EMPTY	True	False	True
True	True	True	20	17	False	False	False
True	True	False	15	59	True	False	True
False	True	True	25	17	True	True	False
True	True	False	34	-2	False	False	False
True	True	True	27	10	False	False	True
False	True	True	EMPTY	62	False	True	False
False	False	True	1	17	False	True	True
True	False	True	26	20000000	True	False	False
True	True	True	10	59	False	True	True
True	False	True	20000000	0	False	True	False
True	True	False	1	20000000	True	False	False
True	False	True	EURO	17	True	False	True
False	False	True	1	EMPTY	False	True	True
True	False	True	35	10	True	True	False
False	True	False	36	SPACE	True	True	False
True	False	True	36	15	True	True	False
False	False	False	31	EURO	False	True	False
False	False	False	26	17	False	True	True

False	True	False	EMPTY15		False	True	False
False	False	True	10	14	True	True	True
True	False	False	15	62	False	False	True
True	True	True	30	14	False	True	False
True	True	False	31	SPACE	True	False	False
True	True	True	30	1	False	False	False
True	False	False	50	14	False	False	False
True	True	True	27	EMPTY	True	False	True
True	True	False	SPACE20000000		True	False	False
True	True	False	SPACEEURO	True	True	False	
False	True	True	16	17	False	True	False
True	True	False	EURO	10	False	True	False
False	False	True	29	61	True	True	False
True	True	True	EURO	20000000	True	False	False
True	False	False	10	15	True	False	False
True	False	False	0	62	True	True	True
False	True	True	25	-2	True	True	True
False	True	True	10	17	False	True	False
False	True	False	0	16	False	False	False

True	True	True	EMPTY16		False	False	False	
False	True	False	1		EURO	True	False	False
False	True	False	EMPTY61		True	False	True	
False	False	True	15		EURO	True	False	True
False	True	False	0	61		True	True	True
False	False	True	EMPTY70		False	False	True	
False	True	False	10		EMPTY	False	False	True
False	False	True	31	60		False	True	True
True	True	False	SPACE17		False	False	False	
True	True	True	20		EMPTY	True	False	False
False	True	True	-1	0		True	True	False
True	True	False	14	10		False	True	False
False	False	True	24	70		True	True	True
True	True	True	10	16		True	False	True
False	False	True	25	1		False	True	True
False	True	False	26	1		False	False	True
False	False	False	-1	61		False	True	True
True	False	True	EMPTY17		True	True	True	
True	False	False	36	20000000		True	True	False

True	False	False	24	17	True	True	True
False	False	True	EMPTY0		True	False	False
False	False	True	20000000		EURO	False	False
False	True	False	1	0	False	False	False
False	False	False	15	15	True	True	True
True	True	True	1	16	True	False	False
True	True	True	36	1	True	False	True
False	False	True	24	10	False	True	False
True	False	False	30	-1	True	True	True
True	True	True	31	1	False	False	False
True	True	False	15	61	True	False	False
True	True	False	26	61	True	True	False
True	True	True	15	-2	True	False	False
True	False	False	15	20	False	False	True
False	True	False	0	SPACETRUE	True	True	True
True	True	True	29	-1	False	False	False
False	True	False	26	-2	False	True	True
True	True	False	EURO	61	False	True	True
False	False	True	36	70	False	False	True

True	False	False	26	EURO	True	True	False
True	True	True	16	15	False	False	True
False	False	True	34	EMPTY	True	True	True
False	True	True	35	EMPTY	True	False	True
False	True	False	34	70	False	True	True
False	False	False	EURO	15	True	False	True
True	True	False	1	1	False	False	False
True	True	False	0	17	False	True	False
False	True	False	20000000	62	False	False	True
True	True	False	24	59	False	False	False
True	True	False	20	20	True	False	False
False	True	False	35	14	False	False	True
True	False	False	35	-1	False	False	False
False	True	True	34	14	True	True	False
True	True	False	35	15	False	True	False
False	False	False	27	14	False	False	False
False	True	True	14	16	True	False	False
False	False	True	35	-2	False	False	False
False	True	False	-2	70	True	True	False

False	False	False	16	0	False	False	True
True	False	False	27	15	True	False	False
False	False	False	-1	1	True	False	True
False	False	False	20000000	-2	False	False	True
False	False	True	-1	14	False	True	True
True	False	False	20000000	1	True	True	False
False	False	False	0	-2	True	True	False
False	False	True	0	20000000	False	True	True
True	True	False	30	SPACE	False	False	False
True	True	False	10	61	True	False	True
False	True	True	14	20	False	False	False
True	True	True	35	EURO	False	False	True
False	False	False	16	14	True	True	False
True	False	False	31	-1	False	False	True
False	False	False	20000000	-1	False	True	True
True	False	False	-2	62	True	False	False
False	False	False	26	EMPTY	False	False	True
True	False	False	EURO	0	False	False	False
True	False	False	15	-1	False	False	True

True	True	True	27	-1	True	False	True
True	False	False	27	61	True	True	False
True	False	True	15	70	False	False	True
True	True	True	27	20000000	False	False	True
False	True	False	10	20	True	True	True
True	False	True	-2	59	True	False	True
True	False	True	25	10	True	True	True
True	True	True	27	EURO	True	False	True
True	True	True	31	14	False	False	True
True	False	False	35	60	False	True	True
True	True	True	1	10	True	False	False
False	True	False	SPACE59		False	False	False
False	True	False	31	-2	False	False	True
True	False	True	30	EMPTY	True	False	True
False	True	True	50	70	False	True	True
True	False	False	35	62	True	False	True
True	True	True	36	62	True	True	False
True	True	True	16	-2	True	False	False
False	True	False	26	0	False	False	True

False	True	False	SPACE16		True	True	False
False	True	False	36	-2	False	False	True
True	True	False	0	14	True	True	False
False	False	True	-2	17	False	True	True
True	True	True	34	62	False	False	False
True	True	True	34	20000000	True	False	True
False	True	True	EURO	70	False	False	True
False	False	True	0	59	False	True	True
True	False	True	16	20000000	True	False	False
False	False	False	-2	-2	True	True	True
False	True	True	-1	20	False	True	False
False	True	True	20000000	17	False	False	True
True	True	True	SPACE15		False	False	False
False	True	True	14	EMPTY	True	False	True
False	True	False	34	16	False	False	True
True	False	True	20000000	61	False	False	False
False	True	True	36	16	False	False	False
False	True	True	24	16	False	True	False
True	True	True	31	10	False	True	True

True	False	True	14	60	False	True	False
False	True	False	-2	0	False	False	False
True	False	True	15	SPACE	False	False	False
False	True	False	27	0	False	False	True
False	True	True	1	61	True	True	True
False	False	True	1	62	False	True	False
True	True	True	50	17	True	False	True
False	True	True	50	16	True	True	False
False	False	True	50	-1	False	True	True
True	True	True	30	20000000	True	True	False
False	True	True	SPACE	60	False	True	False
False	False	True	50	0	True	False	True
False	True	True	EMPTY	60	False	False	False
False	True	False	1	59	False	True	True
True	True	False	26	59	True	False	False
False	False	False	EURO	-2	True	False	False
False	True	False	24	-1	True	True	True
False	False	True	10	60	False	False	False
True	False	False	25	70	False	True	False

True	False	False	50	EMPTY	True	False	False
False	True	False	-1	-2	True	True	True
False	True	True	24	-2	False	True	False
False	True	False	34	17	False	False	False
True	True	True	24	0	True	False	False
False	False	False	50	59	True	False	False
False	True	False	15	14	True	False	True
True	False	False	14	70	False	False	False
True	True	False	35	61	True	False	False
False	False	False	24	EMPTY	False	True	False
True	False	False	15	60	True	True	True
True	True	True	-2	14	True	True	True
True	False	True	25	16	False	False	True
True	False	True	50	20000000	True	True	False
False	False	True	29	-2	False	True	True
False	True	True	50	60	True	True	True
False	False	True	10	70	False	True	False
False	True	False	35	0	True	False	False

Casos de prueba para alcanzar cobertura de decisiones

Para alcanzar cobertura de decisiones, se debe evaluar cada decisión a verdadero y falso al menos una vez.

```
// Primera decisión:  
if(p.puedeRealizarActividadFisica()){  
  
public boolean puedeRealizarActividadFisica() {  
    return facultadesFisicas && !sintomasEnfermedades;  
}
```

p = facultadesFisicas

q = sintomasEnfermedades

p	q	$p \wedge \neg q$
1	1	0
1	0	1
0	1	0
0	0	0

Evaluación	facultadesFisicas	sintomasEnfermedades
True	True	False
False	True	True

```
// Segunda decisión:  
if(t.getTemperatura() < 0 && t.getHumedad() < 15 && t.hayPrecipitacion()){  
  
public boolean hayPrecipitacion() {  
    return precipitacionAgua || precipitacionNieve;  
}
```

p = temperatura < 0

q = humedad < 15

r = precipitacionAgua

s = precipitacionNieve

p	q	r	s	rvs	$p \wedge q \wedge (rvs)$
1	1	1	1	1	1
1	1	1	0	1	1
1	1	0	1	1	1
1	1	0	0	0	0
1	0	1	1	1	0
1	0	1	0	1	0
1	0	0	1	1	0
1	0	0	0	0	0
0	1	1	1	1	0
0	1	1	0	1	0
0	1	0	1	1	0
0	1	0	0	0	0
0	0	1	1	1	0
0	0	1	0	1	0
0	0	0	1	1	0
0	0	0	0	0	0

Evaluación	temperatura	humedad	precipitacionAgua	precipitacionNieve
True	-3	10	False	True
False	10	20	False	False

// Tercera decisión:

```
if(t.getTemperatura() < 0 && t.getHumedad() < 15 && !(t.hayPrecipitacion())){
```

p = temperatura < 0

q = humedad < 15

r = precipitacionAgua

s = precipitacionNieve

p	q	r	s	rvs	$p \wedge q \wedge \neg(r \vee s)$
1	1	1	1	1	0
1	1	1	0	1	0
1	1	0	1	1	0
1	1	0	0	0	1
1	0	1	1	1	0
1	0	1	0	1	0
1	0	0	1	1	0
1	0	0	0	0	0
0	1	1	1	1	0
0	1	1	0	1	0
0	1	0	1	1	0
0	1	0	0	0	0
0	0	1	1	1	0
0	0	1	0	1	0
0	0	0	1	1	0
0	0	0	0	0	0

Evaluación	temperatura	humedad	precipitacionAgua	precipitacionNieve
True	-3	10	False	False
False	-3	10	True	True

```
// Cuarta decisión:
if(t.getTemperatura() >= 0 && t.getTemperatura() <= 15 && !t.hayPrecipitacionAgua()){
```

p = temperatura >= 0

q = temperatura <= 15

r = precipitacionAgua

p	q	r	$p \wedge q \wedge \neg r$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Evaluación	temperatura	precipitacionAgua
True	10	False
False	10	True

```
// Quinta decisión:
if(t.getTemperatura() >= 15 && t.getTemperatura() <= 25 && !t.hayPrecipitacionAgua()
&& !t.isNublado() && t.getHumedad() <= 60){
```

p = temperatura ≥ 15

q = temperatura ≤ 25

r = precipitacionAgua

s = nublado

t = humedad ≤ 60

p	q	r	s	t	$p \wedge q \wedge \neg r \wedge \neg s \wedge t$
1	1	1	1	1	0
1	1	1	1	0	0
1	1	1	0	1	0
1	1	1	0	0	0
1	1	0	1	1	0
1	1	0	1	0	0
1	1	0	0	1	1
1	1	0	0	0	0
1	0	1	1	1	0
1	0	1	1	0	0
1	0	1	0	1	0
1	0	1	0	0	0
1	0	0	1	1	0
1	0	0	1	0	0
1	0	0	0	1	0
1	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	0	0

Evaluación	temperatura	precipitacionAgua	nublado	humedad
True	20	False	False	40
False	-3	False	True	15

```
// Sexta decisión:
if(t.getTemperatura() >= 25 && t.getTemperatura() <= 35 && !t.hayPrecipitacionAgua()){
    p = temperatura >= 25
    q = temperatura <= 35
    r = precipitacionAgua
```

p	q	r	$p \wedge q \wedge \neg r$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Evaluación	temperatura	precipitacionAgua

True	30	False
False	10	True

```
// Séptima decisión:  
if(t.getTemperatura() > 30 && !t.hayPrecipitacionAgua){
```

p = temperatura > 30

q = precipitacionAgua

p	q	$p \wedge \neg q$
1	1	0
1	0	1
0	1	0
0	0	0

Evaluación	temperatura	precipitacionAgua
True	35	False
False	10	True

Para todas las condiciones de:

```
comprobarAforo();
```

Se deberá evaluar a True y a False la variable aforoSuperado de la actividad determinada que se esté comprobando, como se hace siempre igual, para evitar redundancias no se escribirán todos los casos de prueba repetidos.

Casos de prueba para alcanzar cobertura MCDC

Para alcanzar esta cobertura hay que intentar que cada condición sea dominante por sí misma, siempre que se pueda, para evaluar la decisión tanto a verdadera como a falsa.

```
// Primera decisión:  
if(p.puedeRealizarActividadFisica()){
```

```
public boolean puedeRealizarActividadFisica() {
    return facultadesFisicas && !sintomasEnfermedades;
}
```

p = facultadesFisicas

q = sintomasEnfermedades

p	q	$p \wedge \neg q$	Cond. Dominante
1	1	0	q
1	0	1	P, q
0	1	0	P, q
0	0	0	p

Caso 1: Condición dominante q para decisión false

Decisión Esperada	facultadesFisicas	sintomasEnfermedades
False	True	True

Caso 2: Condición dominante p para decisión false

Decisión Esperada	facultadesFisicas	sintomasEnfermedades
False	False	False

Caso 3: Condiciones dominantes p y q para decisión true

Decisión Esperada	facultadesFisicas	sintomasEnfermedades
True	True	False

```
// Segunda decisión:
if(t.getTemperatura() < 0 && t.getHumedad() < 15 && t.hayPrecipitacion()){

    public boolean hayPrecipitacion() {
        return precipitacionAgua || precipitacionNieve;
    }
}
```

p = temperatura < 0

q = humedad < 15

r = precipitacionAgua

s = precipitacionNieve

p	q	r	s	pΛqΛ(rVs)	Cond. Dominante
1	1	1	1	1	P, q, r, s
1	1	1	0	1	P, q, r
1	1	0	1	1	P, q, s
1	1	0	0	0	R, s
1	0	1	1	0	q
1	0	1	0	0	q
1	0	0	1	0	q
1	0	0	0	0	Q, r, s
0	1	1	1	0	p
0	1	1	0	0	p
0	1	0	1	0	p
0	1	0	0	0	P, r, s
0	0	1	1	0	P, q
0	0	1	0	0	P, q
0	0	0	1	0	P, q
0	0	0	0	0	P, q, r, s

Caso 1: Condición dominante p para decisión false

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
False	10	10	True	True

Caso 2: Condición dominante q para decisión false

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
False	-3	20	True	True

Caso 3: Condiciones dominantes r, s para decisión false

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve

False	-3	10	False	False
-------	----	----	-------	-------

Caso 4: Condiciones dominantes p, q, r para decisión true

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
True	-3	10	True	False

Caso 5: Condiciones dominantes p, q, s para decisión true

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
True	-3	10	False	True

// Tercera decisión:

```
if(t.getTemperatura() < 0 && t.getHumedad() < 15 && !(t.hayPrecipitacion())){
```

p = temperatura < 0

q = humedad < 15

r = precipitacionAgua

s = precipitacionNieve

p	q	r	s	pΛqΛ¬(rVs)	Cond Dominante
1	1	1	1	0	r, s
1	1	1	0	0	r
1	1	0	1	0	s
1	1	0	0	1	p, q, r, s
1	0	1	1	0	q, r, s
1	0	1	0	0	q, r
1	0	0	1	0	q, s
1	0	0	0	0	q
0	1	1	1	0	P, r, s
0	1	1	0	0	P, r
0	1	0	1	0	P, s
0	1	0	0	0	p
0	0	1	1	0	P, q, r, s

0	0	1	0	0	P, q, r
0	0	0	1	0	P, q, s
0	0	0	0	0	P, q

Caso 1: Condición dominante p para decisión false

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
False	10	10	False	False

Caso 2: Condición dominante q para decisión false

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
False	-3	20	False	False

Caso 3: Condición dominante r para decisión false

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
False	-3	10	True	False

Caso 4: Condición dominante s para decisión false

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
False	-3	10	False	True

Caso 5: Todas dominantes para decisión true

Decisión Esperada	temperatura	humedad	precipitacionAgua	precipitacionNieve
True	-3	10	False	False

```
// Cuarta decisión:
```

```
if(t.getTemperatura() >= 0 && t.getTemperatura() <= 15 && !t.hayPrecipitacionAgua()){
```

p = temperatura >= 0

$q = \text{temperatura} \leq 15$

$r = \text{precipitacionAgua}$

p	q	r	$p \wedge q \wedge \neg r$	Cond. Dominante
1	1	1	0	r
1	1	0	1	P, q, r
1	0	1	0	Q, r
1	0	0	0	q
0	1	1	0	P, r
0	1	0	0	p
0	0	1	0	P, q, r
0	0	0	0	P, q

Caso 1: Condición dominante p para decisión false

Decisión Esperada	temperatura	precipitacionAgua
False	-3	False

Caso 2: Condición dominante q para decisión false

Decisión Esperada	temperatura	precipitacionAgua
False	20	False

Caso 3: Condición dominante r para decisión false

Decisión Esperada	temperatura	precipitacionAgua
False	10	True

Caso 4: Todas dominantes para decisión true

Decisión Esperada	temperatura	precipitacionAgua
True	10	False

// Quinta decisión:

```
if(t.getTemperatura() >= 15 && t.getTemperatura() <= 25 && !t.hayPrecipitacionAgua()  
&& !t.isNublado() && t.getHumedad() <= 60){
```

p = temperatura ≥ 15

q = temperatura ≤ 25

r = precipitacionAgua

s = nublado

t = humedad ≤ 60

p	q	r	s	t	$p \wedge q \wedge \neg r \wedge \neg s \wedge t$	Cond. Dominante
1	1	1	1	1	0	R, s
1	1	1	1	0	0	R, s, t
1	1	1	0	1	0	R
1	1	1	0	0	0	R, t
1	1	0	1	1	0	s
1	1	0	1	0	0	S, t
1	1	0	0	1	1	P, q, r, s, t
1	1	0	0	0	0	t
1	0	1	1	1	0	Q, r, s
1	0	1	1	0	0	Q, r, s, t
1	0	1	0	1	0	Q, r
1	0	1	0	0	0	Q, r, t
1	0	0	1	1	0	Q, s
1	0	0	1	0	0	Q, s, t
1	0	0	0	1	0	Q
1	0	0	0	0	0	Q, t
0	1	1	1	1	0	P, r, s
0	1	1	1	0	0	P, r, s, t
0	1	1	0	1	0	P, r
0	1	1	0	0	0	P, r, t
0	1	0	1	1	0	P, s
0	1	0	1	0	0	P, s, t
0	1	0	0	1	0	P
0	1	0	0	0	0	P, t
0	0	1	1	1	0	P, q, r, s
0	0	1	1	0	0	P, q, r, s, t
0	0	1	0	1	0	P, q, r
0	0	1	0	0	0	P, q, r, t
0	0	0	1	1	0	P, q, s
0	0	0	1	0	0	P, q, s, t
0	0	0	0	1	0	P, q
0	0	0	0	0	0	P, q, t

Caso 1: Condición dominante p para decisión false

Decisión Esperada	temperatura	precipitacionAgua	nublado	humedad
False	-3	False	False	True

Caso 2: Condición dominante q para decisión false

Decisión Esperada	temperatura	precipitacionAgua	nublado	humedad
False	35	False	False	True

Caso 3: Condición dominante r para decisión false

Decisión Esperada	temperatura	precipitacionAgua	nublado	humedad
False	20	True	False	True

Caso 4: Condición dominante s para decisión false

Decisión Esperada	temperatura	precipitacionAgua	nublado	humedad
False	20	False	True	True

Caso 5: Condición dominante t para decisión false

Decisión Esperada	temperatura	precipitacionAgua	nublado	humedad
False	20	False	False	False

Caso 6: Todas dominantes para decisión true

Decisión Esperada	temperatura	precipitacionAgua	nublado	humedad
False	20	False	False	True

```
// Sexta decisión:
```

```
if(t.getTemperatura() >= 25 && t.getTemperatura() <= 35 && !t.hayPrecipitacionAgua()){
```

```
p = temperatura >= 25
```

$q = \text{temperatura} \leq 35$

$r = \text{precipitacionAgua}$

p	q	r	$p \wedge q \wedge \neg r$	Cond. Dominante
1	1	1	0	r
1	1	0	1	P, q, r
1	0	1	0	Q, r
1	0	0	0	q
0	1	1	0	P, r
0	1	0	0	p
0	0	1	0	P, q, r
0	0	0	0	P, q

Caso 1: Condición dominante p para decisión false:

Decisión Esperada	temperatura	precipitacionAgua
False	10	False

Caso 2: Condición dominante q para decisión false:

Decisión Esperada	temperatura	precipitacionAgua
False	40	False

Caso 3: Condición dominante r para decisión false:

Decisión Esperada	temperatura	precipitacionAgua
False	30	True

Caso 4: Todas dominantes para decisión true:

Decisión Esperada	temperatura	precipitacionAgua
True	30	False

```
// Séptima decisión:  
if(t.getTemperatura() > 30 && !t.hayPrecipitacionAgua()){
```

$p = \text{temperatura} > 30$

$q = \text{precipitacionAgua}$

p	q	$p \wedge \neg q$	Cond. Dominante
1	1	0	q
1	0	1	P, q
0	1	0	P, q
0	0	0	P

Caso 1: Condición dominante p para decisión false:

Decisión Esperada	temperatura	precipitacionAgua
False	25	False

Caso 2: Condición dominante q para decisión false:

Decisión Esperada	temperatura	precipitacionAgua
False	35	True

Caso 3: Todas dominantes para decisión true:

Decisión Esperada	temperatura	precipitacionAgua
True	35	False

Coberturas alcanzadas en los casos anteriores

Como hemos podido observar, a la hora de calcular el número posible de combinaciones de valores de variables, el máximo número que podemos alcanzar es gigante, e inviable si tuviera que ser implementado; es por eso por lo que debemos definir otro alcance para conseguir otra cobertura distinta. Eso nos ha llevado a hacer casos de prueba para alcanzar cobertura each-use; la cual, si bien recorre todos los posibles valores al menos una vez, sigue sin ser la mejor cobertura pues seguimos sin atender a todas las

posibilidades que se nos pueden presentar en nuestro código. Por este motivo, tanto la cobertura de decisiones como la MCDC serían mejores opciones y más viables a la hora de ser implementadas que las dos anteriores.

Por otro lado, es evidente la gran influencia que tiene el diseño y la implementación del programa a la hora de crear casos de prueba para testearlo, ya que cuantas más decisiones y con más condiciones haya que tomar (esto depende de la implementación), más casos de prueba habrá para cubrir todo el código de forma correcta. Además, también influye la implementación ya que dependiendo de cómo se defina una variable o los rangos que pueda tomar dicha variable, esto hará que sea interesante probar más o menos valores del rango, llevando de igual manera a tener más casos de prueba para ella.