

Tema 5. Estructuras de control: selección

Duración: 3 sesiones

Índice:

1. **Introducción: la necesidad de usar estructuras de selección para programar**
2. **Estructuras de selección, o instrucciones condicionales, en Java**
 - Instrucción `if-else` simple y general (anidada y múltiple)
 - Operador ternario
 - Instrucción `switch`

- **Descarga** (desde la carpeta Tema 5 de PoliformaT) y **descomprime** los proyecto **BlueJ ejemplos – Tema 5** y **ejercicios – Tema 5**
- **Abre el proyecto ejemplos – Tema 5** y prepárate para usarlo

Introducción: la necesidad de estructuras de selección

- Hasta ahora hemos escrito métodos que se ajustan a un **flujo de control del programa secuencial**. Sin embargo, es necesario disponer de instrucciones que permitan alterar este flujo secuencial
- Las **estructuras de control** que permiten cambiar el flujo de ejecución de un programa son la **selección** y la **repetición** (iteración)
- En concreto, la **selección** nos permite **tomar decisiones** en función de las circunstancias (datos de entrada, resultados intermedios), **escogiendo una de entre varias alternativas**
- En este tema se estudian las **instrucciones Java** que permiten tomar decisiones, las denominadas instrucciones **de selección o condicionales**

Introducción: la necesidad de estructuras de selección

Ejemplo: validar, o filtrar, datos (sigue...)

BlueJ: pract4

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** PRECONDICIÓN:  $0 \leq h < 24$ ,  $0 \leq m < 60$ 
     *  crea el instante de tiempo h horas y m minutos */
    public TimeInstant(int h, int m) { hours = h; minutes = m; }
    ...
}
```

```
public class TestTimeInstant1 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.println("Lectura de un instante de tiempo");
        System.out.print(" -> Introduzca las horas (entre 0 y 23): ");
        int h = teclado.nextInt();
        System.out.print(" -> Introduzca los minutos (entre 0 y 59): ");
        int m = teclado.nextInt();
        TimeInstant iUsuario = new TimeInstant(h, m);
    }
}
```

¿Qué ocurre si los datos leídos de teclado NO son válidos?

Introducción: la necesidad de estructuras de selección

Ejemplo: validar, o filtrar, datos

BlueJ: pract4

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** PRECONDICIÓN:  $0 \leq h < 24$ ,  $0 \leq m < 60$ 
     * crea el instante de tiempo h horas y m minutos */
    public TimeInstant(int h, int m) { hours = h; minutes = m; }
    ...
}
```

```
public class TestTimeInstant1 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.println("Lectura de un instante de tiempo");
        System.out.print(" -> Introduzca las horas (entre 0 y 23): ");
        int h = teclado.nextInt();
        System.out.print(" -> Introduzca los minutos (entre 0 y 59): ");
        int m = teclado.nextInt();
        SI (h y m son válidos) ENTONCES TimeInstant iUsuario = new TimeInstant(h, m);
        SINO informar mediante un mensaje al usuario
    }
}
```

Necesitamos instrucciones que nos permitan “tomar decisiones”

Introducción: la necesidad de estructuras de selección

Ejemplo: mostrar el máximo de dos valores (sigue...)

BlueJ: pract4

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** devuelve un nº negativo si un instante es menor que otro,
     *  un nº positivo si es mayor y cero si son iguales */
    public int compareTo(TimeInstant otro) {
        return this.toMinutes() - otro.toMinutes();
    }
    ...
}
```

Se devuelve su diferencia porque son `int`

```
public class TestTimeInstant2 {
    public static void main(String[] args) {
        ...
        TimeInstant iUsuario = new TimeInstant(h, m);
        TimeInstant iUTC = new TimeInstant();
        ...
        int res = iUsuario.compareTo(iUTC);
        ...
    }
}
```

¿Cómo mostrar por pantalla un mensaje que indique cuál de los 2 instantes es mayor (posterior) que el otro?

Introducción: la necesidad de estructuras de selección

Ejemplo: mostrar el máximo de dos valores

BlueJ: pract4

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** devuelve un nº negativo si un instante es menor que otro,
     *  un nº positivo si es mayor y cero si son iguales */
    public int compareTo(TimeInstant otro) {
        return this.toMinutes() - otro.toMinutes();
    }
    ...
}
```

```
public class TestTimeInstant2 {
    public static void main(String[] args) {
        ...
        TimeInstant iUsuario = new TimeInstant(h, m);
        TimeInstant iUTC = new TimeInstant();
        ...
        int res = iUsuario.compareTo(iUTC);
        SI (res > 0) ENTONCES System.out.println("El instante posterior es el de usuario");
        SINO System.out.println("El instante posterior, o igual al de usuario, es el UTC");
    }
}
```

Necesitamos instrucciones que nos permitan “tomar decisiones”

Instrucción condicional if-else simple (I)

```
if (condición) { i1; }
```

condición	true	false
	i1	;

```
if (condición) { i1; }  
else { i2; }
```

condición	true	false
	i1	i2

Ejemplo: obtener en max el mayor de 2 números X e y

– **Versión 1:**

```
if (x > y) { max = x; }  
else { max = y; }
```

– **Versión 2:**

```
if (x >= y) { max = x; }  
else { max = y; }
```

– **Versión 3:**

```
max = y; // hipótesis inicial: y es el mayor  
if (x > y) { max = x; }
```

¿Qué versión es la mejor?

Instrucción condicional if-else simple (II)

```
if (condición) { i1; }
```

condición	true	false
	i1	;

```
if (condición) { i1; }  
else { i2; }
```

condición	true	false
	i1	i2

Ejemplo: obtener en max el mayor de 2 números X e y

– **Versión 1:** 1 comparación

```
if (x > y) { max = x; }  
else { max = y; }
```

– **Versión 2:** 2 comparaciones

```
if (x >= y) { max = x; }  
else { max = y; }
```

– **Versión 3:** 1 comparación

```
max = y; // hipótesis inicial: y es el mayor  
if (x > y) { max = x; }
```

¿Qué versión es la mejor?

La que menos comparaciones haga

Instrucción condicional if-else simple



Ejemplo: La clase Bonometro

Diseña la clase Bonometro, que representa un título de transporte con el que se puede viajar en el metro

- Un bonometro TIENE UN nº de tiques disponibles, **10 por defecto (BONO_10)**
- Un bonometro se puede cargar bien con un nº de tiques dado o bien con **10**. También se puede consultar su nº de tiques disponibles y recargarlo
- El método **picar** “gasta” un tique de un bonometro, esto es...

SI QUEDAN, decrementa en 1 el nº de tiques disponibles del bonometro y devuelve el mensaje “**Bono con xxx tiques disponibles**”; **SINO** devuelve “**Bono agotado. Recargalo YA!**”

Análisis por casos

BlueJ: ejemplos - Terma 5

```
public class BonoMetro {  
    public static final int BONO_10 = 10; private int numeroDeTiques;  
    public BonoMetro(int n) { this.numeroDeTiques = n; }  
    public BonoMetro() { this(BONO_10); }  
    public int getNumeroDeTiques() { return numeroDeTiques; }  
    public void recargar(int cantidad) { numeroDeTiques += cantidad; }  
    public String toString() { return "Bono con " + numeroDeTiques + " tiques"; }  
    public String picar() {  
        if (numeroDeTiques > 0) {  
            numeroDeTiques--;  
            return this.toString();  
        }  
        else { return "Bono agotado. Recargalo YA!"; }  
    }  
}
```

Bloque de
instrucciones

Instrucción condicional if-else simple

Ejemplo: validar, o filtrar, datos

BlueJ: pract4

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** PRECONDICIÓN:  $0 \leq h < 24$ ,  $0 \leq m < 60$ 
     *  crea el instante de tiempo h horas y m minutos */
    public TimeInstant(int h, int m) { hours = h; minutes = m; }
    ...
}
```

```
public class TestTimeInstant1 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.println("Lectura de un instante de tiempo");
        System.out.print(" -> Introduzca las horas (entre 0 y 23): ");
        int h = teclado.nextInt();
        System.out.print(" -> Introduzca los minutos (entre 0 y 59): ");
        int m = teclado.nextInt();

        if ((h >= 0 && h <= 23) && (m >= 0 && m <= 59)) {
            TimeInstant iUsuario = new TimeInstant(h, m);
        }
        else { System.out.println("Datos no validos. Bye!"); }
    }
}
```

Instrucción condicional if-else general: Anidada

La instrucción asociada a la condición del **if** o del **else** es, a su vez, una instrucción condicional: diferentes niveles de anidamiento

```
if (B1) {  
    if (B2) { i1; }  
    else { i2; }  
}
```

B1 \ B2	true	false
	i1	i2
false	;	;

```
if (B1) {  
    if (B2) { i1; }  
    else { i2; }  
} else { i3; }
```

B1 \ B2	true	false
	i1	i2
false	i3	i3

```
if (B1) {  
    if (B2) { i1; }  
    else { i2; }  
} else {  
    if (B3) { i3; }  
    else { i4; }  
}
```

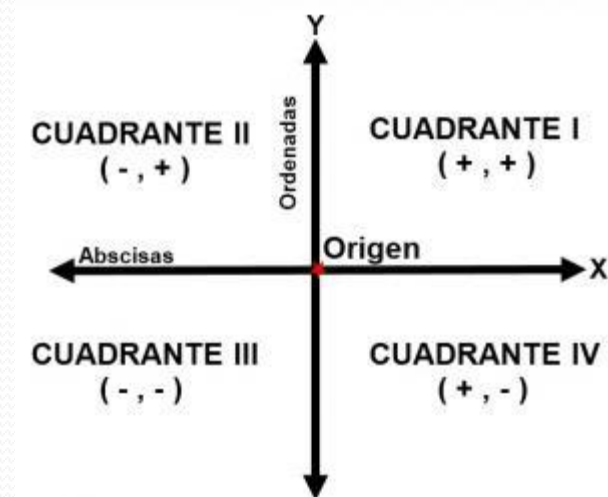
B1	B2	B3	Instrucción a ejecutar
true	true	-	i1
true	false	-	i2
false	-	true	i3
false	-	false	i4

Instrucción condicional if-else general: Anidada

Ejemplo: posición de un Punto en el Plano (sigue...)

Escribe una clase programa que lea dos valores reales desde el teclado y muestre por pantalla un mensaje indicando si el par leído representa a un punto que está en ...

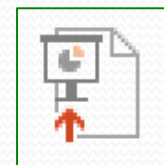
- El origen de coordenadas
- El eje de ordenadas
- El eje de abscisas
- Alguno de los cuatro cuadrantes



PASO 1: análisis por casos para (x, y)

- Si $x \neq 0$
 - Si $y == 0$, el punto está en el eje de abscisas
 - Si $y \neq 0$, el punto está en alguno de los 4 cuadrantes
- Sino, si $x == 0$
 - Si $y == 0$, el punto está en el origen de coordenadas (0, 0)
 - Si $y \neq 0$, el punto está en el eje de ordenadas

TRUCO para obtener este análisis correcto y eficiente “a la primera”...



Instrucción condicional if-else general: Anidada

Ejemplo: posición de un Punto en el Plano



BlueJ: ejemplos - Terma 5

PASO 2: implementación en Java

```
...
public class CompruebaCoordenadas {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in).useLocale(Locale.US);
        double x = teclado.nextDouble();
        double y = teclado.nextDouble();

        System.out.print("El punto esta en ");
        if (x != 0) {
            if (y != 0) { System.out.println("uno de los 4 cuadrantes"); }
            else { System.out.println("el Eje X"); }
        }
        else { // x == 0
            if (y != 0) { System.out.println("el Eje Y"); }
            else { System.out.println("el origen de coordenadas"); }
        }
    }
}
```

Instrucción condicional if-else

Ejercicio nº 3 del libro de la asignatura



BlueJ: ejercicios - Tema 5

En una tienda de electrodomésticos se aplican distintos descuentos en función del **total** de compras realizadas.

- Si **total** < 500€, no hay descuento, o se paga el total (**res = total**;))
- Si $500\text{€} \leq \text{total} \leq 2000\text{€}$, hay un 30% de descuento, o se paga el 70% (**total * 0.7**)
- Si **total** > 2000€, hay un 50% de descuento, o se paga el 50% (**total * 0.5**)

Escribe en la clase **Condicionales** un método que, dado un **total**, devuelva la **cantidad a pagar** tras aplicar el descuento correspondiente. Se debe resolver el problema con una única instrucción condicional (anidada) que empiece por...

- **if** (**total** >= 500)
- **if** (**total** <= 2000)

PISTA: transcribe a Java esta “Tabla de Casos”, donde los resultados que NO se pueden dar, como **total** < 500 AND 30% descuento, se han representado con un recuadro azulón

descuento	0	30%	50%
total €			
< 500	total		
[500, 2000]		total * 0.7	
> 2000			total * 0.5

Instrucción condicional if-else

Ejercicio nº 4 del libro de la asignatura



BlueJ: ejercicios - Tema 5

Reescribe en la clase Condicionales el método ejercicio4 (que figura a continuación) para que su resultado sea **exactamente** el mismo **pero usando una condicional más sencilla, con estructura** `if (x < 0) ... else ...`

```
public static String ejercicio4(int x, char c) {  
    String res = "";  
    if (x < 0 && c == 'x') { res = "Caso 1"; }  
    else if (x < 0 && c != 'x') { res = "Caso 2"; }  
    else if (x >= 0 && c == 'y') { res = "Caso 3"; }  
    else if (x >= 0 && c != 'y') { res = "Caso 4"; }  
    return res;  
}
```

PISTA: completa y transcribe a Java esta “Tabla de Casos”

c \ x				
< 0		Caso 2		
				Caso 4

Instrucción condicional if-else

Ejercicios 11 y 22 del libro de la asignatura



BlueJ: ejercicios - Tema 5

Escribe en la clase Condicionales...

- Un método estático que, dados tres valores enteros *a*, *b* y *c*, implemente distintas soluciones del siguiente análisis por casos; si quieres, puedes hacer uso de operadores cortocircuitados.
 - Si $a > b$ entonces *res* = true
 - Si $a < b$ entonces *res* = false
 - Si $a = b$ y $a > c$ entonces *res* = true
 - Si $a = b$ y $a < c$ entonces *res* = false
 - Si $a = b$ y $a = c$ entonces *res* = false
- Un método estático que dados dos números enteros, *num1* y *num2*, devuelva true si su producto es positivo o nulo y false en caso contrario (producto negativo).

Para establecer el resultado **NO debes calcular el producto**

PISTA: para cada ejercicio, construye y transcribe a Java su “Tabla de Casos”

Instrucción condicional if-else general: **Múltiple**

La instrucción asociada a la condición del **else** es, a su vez, una instrucción condicional

```
if (B1) { i1; }  
else if (B2) { i2; }  
else if (B3) { i3; }  
else { i4; }
```

B1	B2	B3	Instrucción a ejecutar
true	-	-	i1
false	true	-	i2
false	false	true	i3
false	false	false	i4

Instrucción condicional if-else general: Múltiple

Ejemplo: resultado de comparar dos valores

BlueJ: pract4

```
public class TimeInstant {
    private int hours, minutes;
    ...
    /** devuelve un nº negativo si un instante es menor que otro,
     *  un nº positivo si es mayor y cero si son iguales */
    public int compareTo(TimeInstant otro) {
        return this.toMinutes() - otro.toMinutes();
    }
    ...
}
```

```
public class TestTimeInstant3 {
    public static void main(String[] args) {
        ...
        TimeInstant iUsuario = new TimeInstant(h, m);
        TimeInstant iUTC = new TimeInstant();
        ...
        int res = iUsuario.compareTo(iUTC);

        if (res > 0) { System.out.println("El instante posterior es el de usuario"); }
        else if (res < 0) { System.out.println("El instante posterior es el UTC"); }
        else { System.out.println("El instante posterior es el de usuario"); }
    }
}
```

Tienes un ejemplo similar en la clase TestIfs del proyecto ejemplos – Tema 5

Instrucción condicional if-else general: Múltiple

Ejemplo: comparar una comida con otra (sigue...)

Diseña en la clase Comida **un método compareTo** que indique si una comida es más sana que otra; es decir, si aporta menos calorías que la otra o, **a igualdad de calorías**, si el tiempo de preparación es menor



BlueJ: ejemplos - Terma 5

```
public class Comida {  
    private double calorías; // Una comida TIENE UN n° calorías  
    private int minutsPrep; // Una comida TIENE UNOS minutos de preparación  
    public Comida(double c, int m) { calorías = c; minutsPrep = m; }  
    ...  
    /** Devuelve...  
     * un int negativo si una comida es más sana que otra  
     * un int positivo si una comida es menos sana que otra  
     * el int cero si una comida es igual de sana que otra */  
    public int compareTo(Comida otra) {  
        // Primera versión correcta/aceptable (con if eficiente):  
        int res = minutsPrep - otra.minutsPrep; // ¿Hipótesis?  
        if (this.calorías < otra.calorías) { res = -1; }  
        else if (calorías > otra.calorías) { res = 1; }  
        return res;  
    }  
}
```

minutsPrep
es int!!

Instrucción condicional if-else general: Múltiple

Ejemplo: comparar una comida con otra

Diseña en la clase Comida **un método compareTo** que indique si una comida es más sana que otra; es decir, si aporta menos calorías que la otra o, **a igualdad de calorías**, si el tiempo de preparación es menor



BlueJ: ejemplos - Terma 5

```
public class Comida {  
    private double calorías; // Una comida TIENE UN n° calorías  
    private int minutsPrep; // Una comida TIENE UNOS minutos de preparación  
    public Comida(double c, int m) { calorías = c; minutsPrep = m; }  
    ...  
    /** Devuelve...  
     * un int negativo si una comida es más sana que otra  
     * un int positivo si una comida es menos sana que otra  
     * el int cero si una comida es igual de sana que otra */  
    public int compareTo(Comida otra) {  
        // Segunda versión, más “recomendable”:  
        int res = (int) Math.signum(calorías - otra.calorías);  
        if (res == 0) { res = minutsPrep - otra.minutsPrep; }  
        return res;  
    }  
}
```

Instrucción condicional if-else

Ejercicio CAP (nº 26 del libro de la asignatura)



Piedra, Papel o Tijera (clave CCDFG4ai)



IMPORTANTE: haz **antes**, **en papel**, un buen análisis por casos. Para ello, es imprescindible conocer este juego infantil: en una partida intervienen dos jugadores; cada uno de ellos puede decir, al mismo tiempo que el otro “Piedra” o “Papel” o “Tijeras”, de forma que:

- Si un jugador dice “Piedra” y el otro “Tijeras”, **gana** el jugador que haya elegido “Piedra”, ya que la piedra puede romper a las tijeras
- Si un jugador dice “Tijeras” y el otro elige “Papel”, **gana** el jugador que haya elegido “Tijeras”, ya que las tijeras cortan el papel
- Si un jugador dice “Papel” y el otro dice “Piedra”, **gana** el jugador que haya elegido “Papel”, ya que con el papel se puede envolver a la piedra
- Si ambos jugadores eligen lo mismo, **empatan**

PISTA: una primera versión de la “Tabla de Casos”

J2 \ J1	Piedra	Papel	Tijeras
Piedra	Empatan	Pierde J1	Gana J1
Papel	Gana J1	Empatan	Pierde J1
Tijeras	Pierde J1	Gana J1	Empatan

El operador ternario: ¿Qué es y por qué NO lo usaremos?

- **Definición:** es un **operador** especial que introduce Java (al igual que C y C++), con un uso **parecido** al de una instrucción condicional.

La forma general de una expresión en la que aparece este operador es ...

`exprbool ? expr1 : expr2`

donde ...

- `exprbool` es una expresión `boolean`
- `expr1` y `expr2` son expresiones del mismo tipo

- **Evaluación:** se evalúa `exprBool`; según el resultado sea `true` o `false`, el valor de **toda** la expresión es el valor de `expr1` o el de `expr2`, respectivamente

- **Ejemplos:**

1. Obtener en la variable `max` el mayor de 2 números enteros `x` e `y`

```
int x, y, max;
```

```
...
```

```
max = x > y ? x : y;
```

```
if (a > b) { max = a; }  
else { max = b; }
```

2. Obtener en la variable `cociente` el resultado de la división de 2 enteros `x` e `y`

```
int x, y, cociente;
```

```
...
```

```
cociente = (x != 0 && y != 0) ? x / y : 0;
```

```
if (x != 0 && y != 0) { cociente = x / y; }  
else { cociente = 0; }
```

Instrucción condicional switch

```
switch (expresion) {  
    case val1:  
        [SC1] [break;]  
    case val2:  
        [SC2] [break;]  
    . . . . .  
    . . . . .  
    case valn:  
        [SCn] [break;]  
    [default:  
        [SCn+1] [break;]]  
}
```

Requerimientos sintácticos:

- expresion es de tipo simple (**excepto** float y double) o un String (desde la versión 7 de Java)
- val1, val2, ..., valn son del mismo tipo o compatible con el de expresion
- SC1, SC2, ..., SCn+1 son secuencias de instrucciones

Modelo de ejecución:

1. Evaluar expresion
2. Comparar el resultado de 1. con cada uno de los valores de los case
3. Para el primer valor que coincida, ejecutar el código de ese case **y el código asociado a los case posteriores a él hasta que ...**
 - Bien se encuentre el primer break
 - Bien acabe el switch
4. Si ningún valor coincide, ejecutar las instrucciones asociadas a la etiqueta default (si existe) hasta el final del switch
5. Continuar con la instrucción que sigue al switch

Instrucción condicional switch

Modelo de ejecución: TestSwitch (ejercicios 7 y 8 Capítulo 7 del libro)



BlueJ: ejemplos - Terma 5

```
// ejercicio7(): qué se muestra por pantalla tras ejecutar el siguiente código ¿Por qué?  
switch (2) {  
    case 1:  
        System.out.println(1); break;  
    case 2:  
        System.out.println(2);  
    case 3:  
        System.out.println(3); break;  
    default:  
        System.out.println(4);
```

```
// ejercicio8(): qué se muestra por pantalla tras ejecutar el siguiente código ¿Por qué?  
int primOpcion = 1 // ¿Y si primOpcion vale 2?  
switch (primOpcion + 1) {  
    case 1:  
        System.out.print("Ensalada "); break;  
    case 2:  
        System.out.print("Paella "); break;  
    case 3:  
        System.out.print("Emperador ");  
    case 4:  
        System.out.print("Helado "); break;  
    default:  
        System.out.print("Buen provecho");
```


Instrucción condicional switch

Ejemplo: gestión de un menú



BlueJ: ejemplos - Terma 5

```
import java.util.Scanner;
public class MenuPunto {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        PuntoR p = new PuntoR(1.5, 3.0);
        System.out.println(" MENÚ");
        System.out.println("1. Distancia al origen");
        System.out.println("2. Mover aleatorio");
        System.out.println("3. Datos del punto");
        System.out.print("\nElige una opcion: ");
        int opc = teclado.nextInt();
        switch (opc) {
            case 1:
                double d = p.distOrigen();
                System.out.printf("Distancia = %4.2f\n" + d);
                break;
            case 2:
                p.moverAleat();
                System.out.println("Punto desplazado " + p.toString());
                break;
            case 3:
                System.out.println("Datos del punto: " + p.toString());
                break;
            default:
                System.out.println("Opcion no valida");
        }
    }
}
```

Instrucción condicional switch

Ejemplo: comprobar la estación a la que pertenece un mes



Blue: ejemplos - Terma 5

```
// En la clase de utilidades MiUtil
public static String estacion(int numMes) {
    String estacion = "";
    switch (numMes) {
        case 1: case 2: case 12:
            estacion = "Invierno"; break;
        case 3: case 4: case 5:
            estacion = "Primavera"; break;
        case 6: case 7: case 8:
            estacion = "Verano"; break;
        case 9: case 10: case 11:
            estacion = "Otoño"; break;
        default:
            estacion = "Error"; break;
    }
    return estacion;
}
```

Análisis por casos:

- Primavera: marzo, abril y mayo (meses 3, 4 y 5)
- Verano: junio, julio y agosto (meses 6, 7 y 8)
- Otoño: septiembre, octubre y noviembre (meses 9, 10 y 11)
- Invierno: diciembre, enero y febrero (meses 12, 1 y 2)

```
if (numMes == 12 || numMes == 1 || numMes == 2) {
    estacion = "Invierno";
} else if (numMes >= 3 && numMes <= 5) {
    estacion = "Primavera";
} else if (numMes >= 6 && numMes <= 8) {
    estacion = "Verano";
} else if (numMes >= 9 && numMes <= 11) {
    estacion = "Otoño";
} else { estacion = "Error"; }
```

Instrucción condicional switch

Ejemplo: comprobar si una fecha es correcta

```
public boolean esCorrecta() {
    boolean correcta = false;
    if (anyo > 0 && (mes >= 1 && mes <= 12) && (dia >= 1 && dia <= 31)) {
        int diasDelMes;
        switch (mes) {
            case 2:
                if (esBisiesto()) { diasDelMes = 29; }
                else { diasDelMes = 28; }
                break;
            case 4:
            case 6:
            case 9:
            case 11:
                diasDelMes = 30;
                break;
            default:
                diasDelMes = 31; break;
        }
        if (this.dia <= diasDelMes) correcta = true;
    }
    return correcta;
}
```

```
if (mes == 2) {
    if (esBisiesto()) { diasDelMes = 29; }
    else { diasDelMes = 28; }
} else if (mes == 4 || mes == 6 || mes == 9 || mes == 11) {
    diasDelMes = 30;
} else { diasDelMes = 31; }
```

Instrucción condicional switch

Ejercicios 21 y 25 del Capítulo 7 del libro



BlueJ: ejercicios - Tema 5

- Escribe una clase programa que calcule la tarifa de una autoescuela teniendo en cuenta el tipo de carnet (A, B, C o D) y el número de prácticas realizadas

Tipo de carnet	Tarifa de les matrículas	Precio por práctica
A	150 €	15 €
B	325 €	21 €
C	520 €	36 €
D	610 €	50 €

- Escribe una clase programa que, tras leer de teclado dos operandos `double`, `num1` y `num2`, y un operador `op` de tipo `char`, muestre por pantalla el resultado de la operación `num1 op num2`

Ejercicios CAP



Fecha con `esFestivo` (clave CCDFJ4ai)



Nuevo `toString()` para la clase `Medicamento` (clave CCDFH4ai)