

Instituto Tecnológico y de Estudios  
Superiores de Occidente – ITESO



**ITESO**

Universidad Jesuita  
de Guadalajara

Materia: Programación Orientada a Objetos

Maestro: Hugo Iván Piza Dávila

TAREA 1

Sesión: 4

Fecha: 29 de agosto el 2025

Temas: Condicionales  
Control de Flujo

Autor: Alor Santiago Oscar Alberto

## PROBLEMA 1

### Descripción:

Determina el signo zodiacal de una persona a partir de su fecha de nacimiento.

### Código fuente:

Despliega dos cuadros de diálogo de entrada de datos. La primera entrada solicita "Mes de nacimiento" mientras que el segundo solicita el "día de nacimiento":

```
1 String mes = JOptionPane.showInputDialog("Dame tu mes de nacimiento");
2 String diaNacimiento = JOptionPane.showInputDialog("Dame tu día de nacimiento");
```

Se crea un nuevo String "mesZodiacal" para el mes el (Se pudo sobrescribir el primero realmente) en donde se modifica el contenido, primero crean una cadena nueva que elimina espacio y después un nuevo objeto que ahora es todo minúscula. después se crear un int "diaNac" el cual guardará la conversión de String a Int de "díaNacimiento".

```
1 String mesZodiacal = mes.trim().toLowerCase();
2 int diaNac = Integer.parseInt(diaNacimiento);
```

después se ejecuta el código principal el cual es la asignación de un string "signoZodiacal" que guardará el signo dependiendo de la condición. Para las condiciones solo se tomo en cuenta que la primera mitad del mes corresponde a un signo Zodiacal mientras que para la otra mitad es diferente, el siguiente mes inicia su primera mitad con el ultimo Signo Zodiacal del mes anterior y la otra mitad cambia y así sucesivamente.

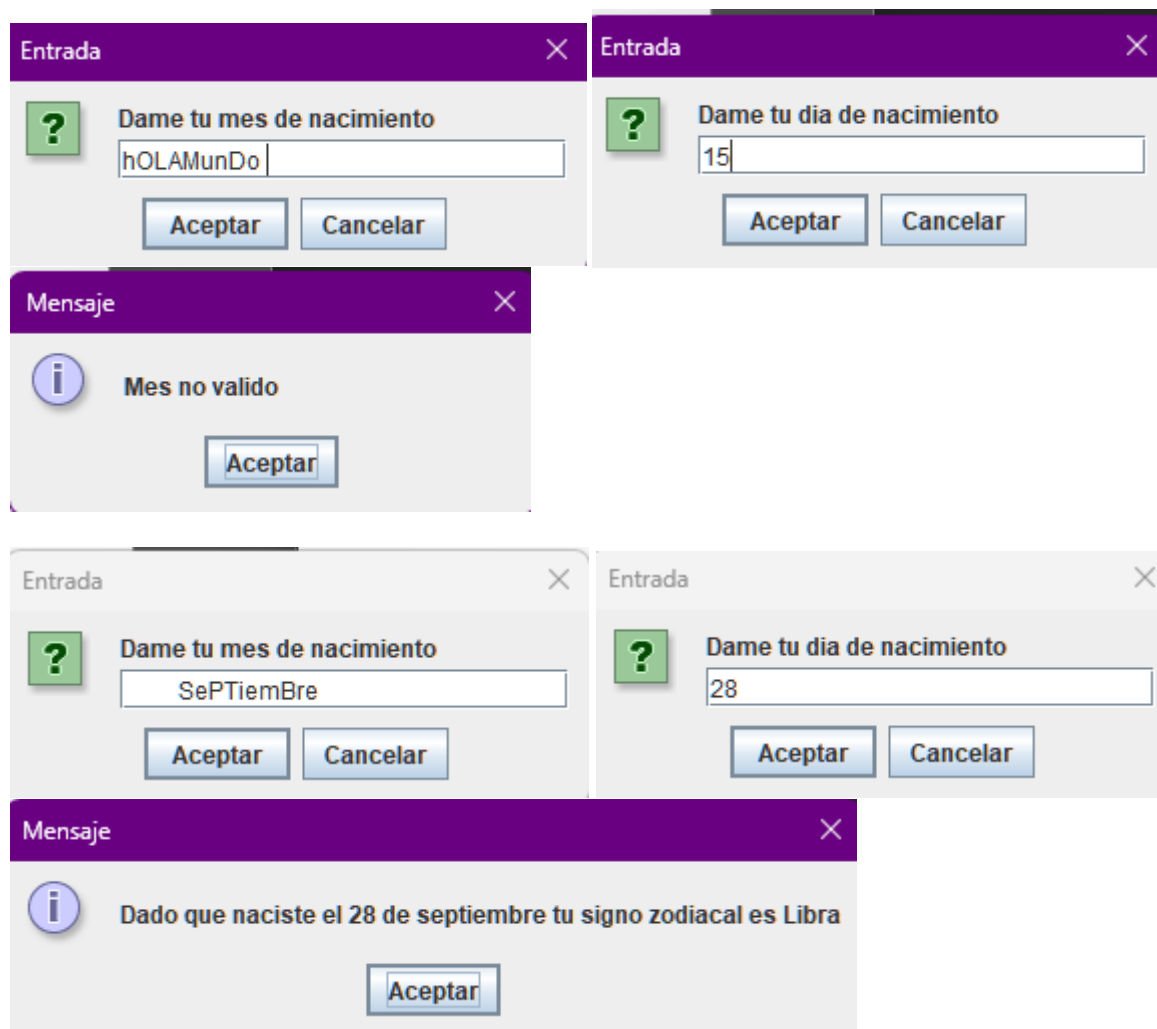
Nótese que el switch evalúa el contenido de mesZodiacal que fue modificado anteriormente:

```
1 String signoZodiacal = switch (mesZodiacal){
2     case "enero" -> (diaNac < 20) ? "Capricornio" : "Acuario";
3     case "febrero" -> (diaNac < 19) ? "Acuario" : "Piscis";
4     case "marzo" -> (diaNac < 20) ? "Piscis" : "Aries";
5     case "abril" -> (diaNac < 20) ? "Aries" : "Tauro";
6     case "mayo" -> (diaNac < 21) ? "Tauro" : "Geminis";
7     case "junio" -> (diaNac < 21) ? "Geminis" : "Cancer";
8     case "julio" -> (diaNac < 23) ? "Cancer" : "Leo";
9     case "agosto" -> (diaNac < 23) ? "Leo" : "Virgo";
10    case "septiembre" -> (diaNac < 23) ? "Virgo" : "Libra";
11    case "octubre" -> (diaNac < 23) ? "Libra" : "Escorpio";
12    case "noviembre" -> (diaNac < 22) ? "Escorpio" : "Sagitario";
13    case "diciembre" -> (diaNac < 22) ? "Sagitario" :
14    "Capricornio";
15    default -> "Mes no valido";
};
```

así mismo el valor por default es "Mes no valido", en la última parte del código se evalúa si el contenido de "signoZodiacal" es igual a esta cadena de caracteres, en caso de que si, se envía dicho mensaje al usuario, de lo contrario, se lanza un nuevo mensaje con todos los valores del usuario, así como el resultado:

```
1 // Conditional
2 if (signoZodiacal.equals("Mes no valido")) {
3     JOptionPane.showMessageDialog(null, signoZodiacal);
4 } else {
5     JOptionPane.showMessageDialog(null, String.format("Dado que naciste el %d de %s
6     tu signo zodiacal es %s", diaNac, mesZodiacal, signoZodiacal));
7 }
```

### Ejecución:



Entrada

?

Dame tu mes de nacimiento

DICIEMBRE

Aceptar Cancelar

Entrada

?

Dame tu día de nacimiento

23

Aceptar Cancelar

Mensaje

i

Dado que naciste el 23 de diciembre tu signo zodiacal es Capricornio

Aceptar

## PROBLEMA 2

### Descripción:

Encuentra el número primo más grande en un rango [A, B] dado por el usuario.

### Código fuente:

Se despliegan dos cuadros de diálogos, el primero solicita un numero que será "menor" en el rango y el segundo será el numero "mayor" del mismo rango.

```
1 String minNumero = JOptionPane.showInputDialog("Escribe el numero menor");
2 String maxNumero = JOptionPane.showInputDialog("Escribe el numero mayor");
```

Despues de ello se "parsea" cada "String" para se guardado dentro de un "int" que será usado para la evaluación.

```
1 int minNum = Integer.parseInt(minNumero);
2 int maxNum = Integer.parseInt(maxNumero);
```

Después de ello se declaran dos variables. Una es "control" que usada dentro del ciclo "while" para confirmar si dentro de las divisiones posibles de un número, hubo alguna operación que dividió el numero siendo evaluado y quedó en cero, si ello sucede "control" se queda con el valor de "0" y se rompe el ciclo. El otro valor es "result" el cual se encarga de almacenar el valor del último primo encontrado dependiendo de lo que dé el resultado del "while" al finalizar (véase más adelante).

```
1 int control = 0;
2 int result = 0;
```

La lógica del "for" solamente itera entre cada número del rango y en cada iteración reinicia el valor de "divisor". El "while" por otro lado siempre se ejecuta al menos una vez y tiene dos condiciones, si el numero evaluado al ser dividido entre el "divisor" es igual a 0 alguna vez, significa que no es primo, nótese que se empieza en 2 y termina en cuando divisor es igual a la raíz cuadrada del numero evaluado en el "for".

Asi mismo, si el resultado es diferente de 0 entonces el divisor aumenta (Estando condicionado por el "while") y seguirá aumentando siempre y cuando no haya resultados que den 0 (Si hay no es primo) o cuando se alcance la raíz cuadrada (Es primo).

```
1 for(int i = minNum; i <= maxNum; i++) {
2     int divisor = 2;
3     do {
4         if( i%(double)divisor == 0 ) {
5             control = 0;
6             break;
7         }
8         divisor++;
9         control = 1;
```

```

10         }while (divisor <= (double)Math.sqrt(i));
11         // Acaba While
12         if (control == 1) result = i;
13     }

```

Aquí sucede lo siguiente:

1. Si la división del numero siendo evaluado por la iteración NUNCA da cero, entonces es primo y el valor se guarda en "result", el cual ahora tiene el numero primo. Cambiará solo si encuentra otro dentro del rango (El mayor).
2. Si la división alguna vez da cero, "control" es 0, se termina el programa y aquí suceden dos cosas también:
  - a. Dado que "control" finaliza en cero, "result" queda en 0 como se declaró originalmente lo que significa que no es primo porque alguna vez se dividió correctamente.
  - b. Si la división nunca quedó en 0, entonces "control" es 1 siempre y la condicional fuera del "while" asigna el valor de la iteración a "result".

La explicación de arriba queda mas clara con la última parte del código en el cual avisa que no hay número primos si "result" se mantuvo en 0 o bien, avisa del último primo guardado si alguna vez algún número del rango cumplió con que ninguna división dio cero:

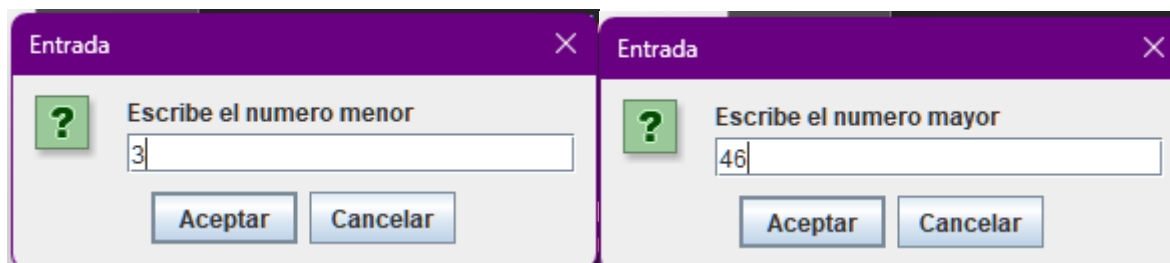
```

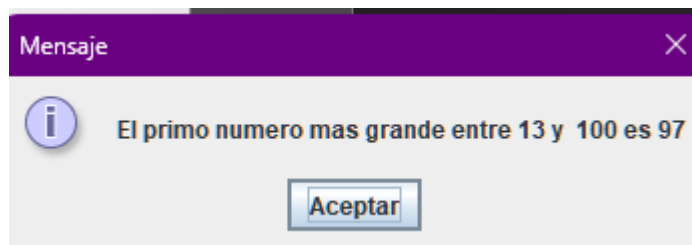
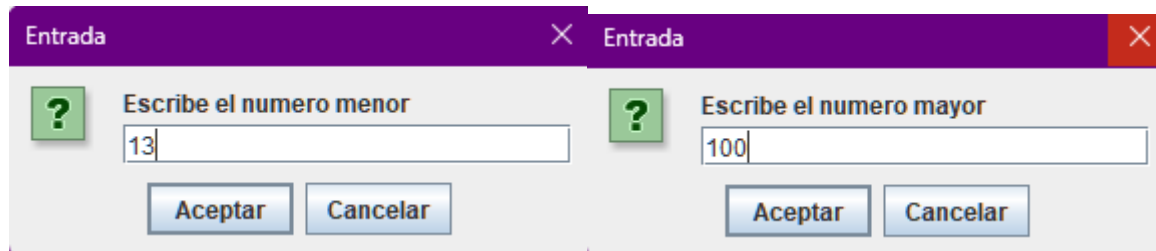
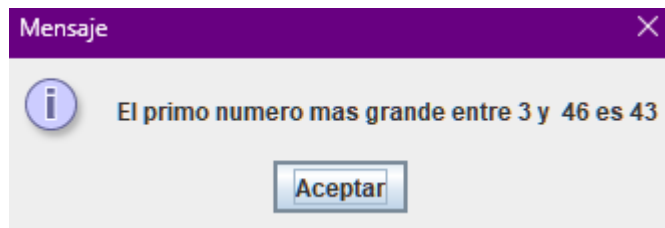
1  if(result == 0) {
2      JOptionPane.showMessageDialog(null, "No hay numero primos en tu rango");
3  } else {
4      JOptionPane.showMessageDialog(null, String.format("El primo numero mas grande
5  entre %d y %d es %d", minNum, maxNum, result));
6  }

```

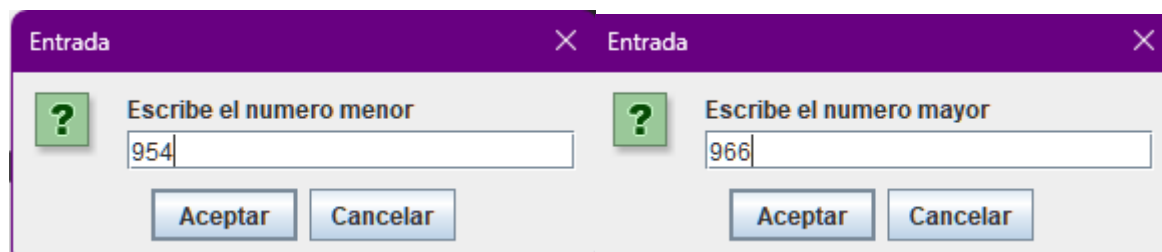
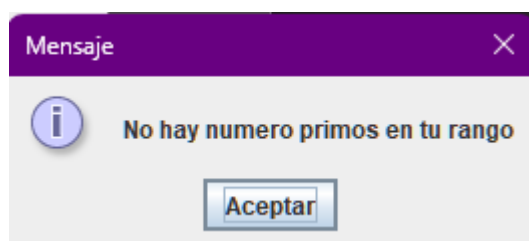
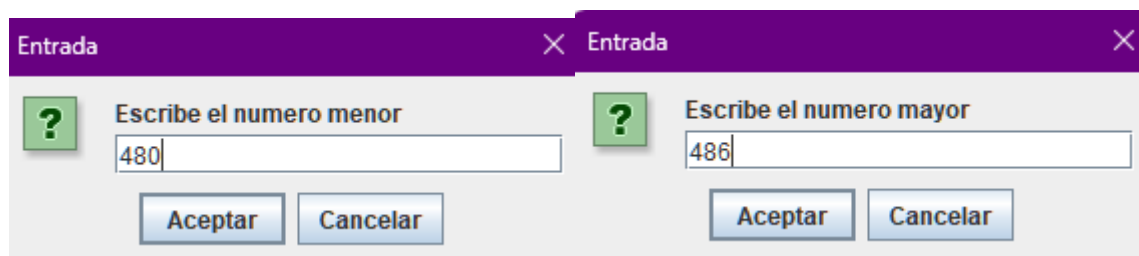
Ejecución:

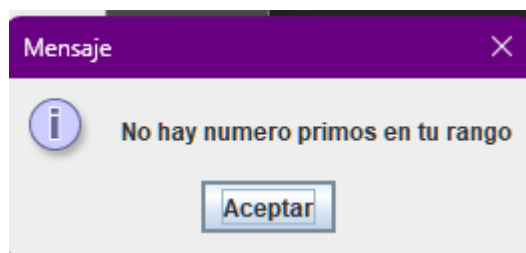
Para la ejecución se usaron números que abarquen el siguiente rango de primos, los primos dos deben dar como resultado 43 y 97 mientras que los otros dos están contenidos entre ["479+1" - "487-1"] y ["953+1" - "967-1"] no deben arrojar resultados:





NO PRIMOS ENCONTRADOS AHORA:





2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601
607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733
739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863
877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997		

Codigo Fuente:

[https://github.com/OsrKozuki/POO\\_ESI0170/blob/main/startingPoo/src/tareas/tareaUno.java](https://github.com/OsrKozuki/POO_ESI0170/blob/main/startingPoo/src/tareas/tareaUno.java)