

## Tema

Introducción a la Programación con JAVA,  
Elementos Condicionales.

## Tiempo estimado

Se estima que el estudiante requerirá **20 minutos** para crear la aplicación guiada y **25 minutos** para agregar los elementos adicionales.

## Objetivo

Construir un programa en **Netbeans** que nos permita aplicar un formato de ingreso de visitantes a un parque de diversiones, según el último dígito de la cédula, utilizando estructuras condicionales. Por políticas internas, no pueden ingresar personas menores de 18 años.

## Información previa

A pesar de que las políticas de ingreso al parque pueden cambiar según sea la situación de salud del país, por el momento definiremos las siguientes reglas de ingreso para cada día de la semana. Como se indicó anteriormente, no podrán ingresar personas menores de 18 años. Además, tome en cuenta que el parque no abre sábados ni domingos.

Número de Día	Último dígito de la cédula
Lunes	Cédulas terminadas en 0 y 1
Martes	Cédulas terminadas en 2 y 3
Miércoles	Cédulas terminadas en 4 y 5
Jueves	Cédulas terminadas en 6 y 7
Viernes	Cédulas terminadas en 8 y 9

Se solicitará al usuario el último dígito de la cédula y el programa deberá indicarle el día en que **NO** podrá visitar el parque.

Para esto escribiremos gran parte de la lógica de nuestro programa en un método que llamaremos “revisarRestriccion()”.

## Pasos a seguir

En todo proceso de programación, nuestro primer paso será asegurarnos que tenemos claro el objetivo de lo que vamos a programar, si este objetivo no está claro,

consecuentemente el resultado de nuestra programación no va a ser el correcto.

Recordemos que contamos con dos instrucciones para poder tomar algún tipo de decisiones, cada una de estas instrucciones tiene sus características particulares y su forma de utilizarse, por lo que el ejercicio que desarrollaremos es una alternativa, pero existen muchas maneras de desarrollarlo, de acuerdo con otras consideraciones y requerimientos adicionales.

Lo primero que realizaremos es la creación de un nuevo proyecto (como lo hemos hecho durante las clases), al final del proceso contaremos con nuestro proyecto con la estructura básica del mismo.

```
package com.estudiante.lab03;

public class Lab03 {
    public static void main(String[] args) {
    }
}
```

*Si bien tenemos clara la importancia de tener suficiente y correctamente documentados nuestros programas, en estos laboratorios los omitiremos por temas de claridad y espacio.*

Solicitaremos en el método principal el último dígito del número de cédula. Este valor lo enviaremos como parámetro a otro método, este otro método identificará según lo establecido el día en que el visitante no podrá ingresar al parque y retornará la respuesta al método principal para ser mostrada.

Por lo tanto, actualizaremos el método principal con lo siguiente.

```
package com.estudiante.lab03;
import javax.swing.JOptionPane;
public class Lab03 {
    public static void main(String[] args) {
        String mensajeRetorno;
        int ultimoDigitoCedula;
        ultimoDigitoCedula=Integer.parseInt(JOptionPane.
            showInputDialog(null,"Último dígito de la cédula:"));
        mensajeRetorno=revisarRestriccion(ultimoDigitoCedula);
        JOptionPane.showMessageDialog(null,mensajeRetorno);
    }
}
```

Como se puede observar en el código, para el momento de escribir el código, el método “revisarRestriccion()” aun no existe dentro del programa, es por esto que Netbeans no lo reconoce y lo muestra como un error, pero esto se corregirá en el momento de crearlo.

Si analizamos cada una de esas líneas podemos identificar los siguiente.

```
import javax.swing.JOptionPane;
```

Nos permite importar la funcionalidad que ofrece el objeto “swing” llamado “JOptionPane”, este objeto lo utilizaremos tanto para capturar información como para mostrarla.

```
String mensajeRetorno;  
int ultimoDigitoCedula;
```

Contamos con la declaración de 2 variables que correspondientemente nos servirán para almacenar el mensaje que nos retornará nuestro nuevo método y posteriormente mostrar; la otra variable nos permitirá almacenar el valor ingresado por el usuario, el cuál enviaremos como parámetro al método.

```
ultimoDigitoCedula=Integer.parseInt(JOptionPane.  
showInputDialog(null,"Último dígito de la cédula:"));
```

Esta línea se compone de diferentes instrucciones, utilizamos JOptionPane para capturar un valor, este valor es transformado en un número entero para poder ser almacenado dentro de la variable para tal propósito.

```
mensajeRetorno=revisarRestriccion(ultimoDigitoCedula);
```

En esta línea hacemos el llamado al nuevo método, le enviamos como parámetro entre los paréntesis el valor capturado y el valor de retorno lo almacenamos en la variable que creamos para tal fin.

```
JOptionPane.showMessageDialog(null, mensajeRetorno);
```

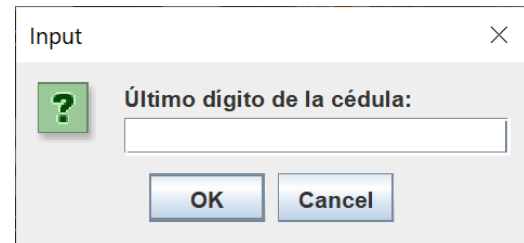
Finalmente, ese mensaje lo mostramos al usuario.

Con estos pasos tendríamos finalizado nuestro método principal, pero obviamente eso no es todo, debemos construir el método encargado de revisar realmente el tema de la restricción, de la siguiente manera.

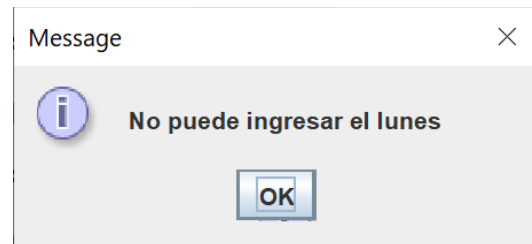
```
public static String revisarRestriccion(int valor){  
    if(valor==0 || valor==1){  
        return "No puede ingresar el lunes";  
    } else if(valor==2 || valor==3){  
        return "No puede ingresar el martes";  
    } else if(valor==4 || valor==5){  
        return "No puede ingresar el miércoles";  
    } else if(valor==6 || valor==7){  
        return "No puede ingresar el jueves";  
    } else if(valor==8 || valor==9){  
        return "No puede ingresar el viernes";  
    }  
}
```

En la declaración del método podemos observar cómo se define un parámetro llamado “valor”, el cuál evaluamos por medio de la construcción de los elementos condicionales, para dentro de cada uno de ellos retornar un mensaje si sus condiciones se cumplen.

Si procedemos a ejecutar nuestro programa podremos observar el siguiente funcionamiento.



Si digitamos un valor como “0” o “1”, obtendremos el siguiente resultado.



Si realizamos pruebas con otros valores podremos observar que se cumple lo establecido en el requerimiento inicial.

*Ahora bien, ¿esta manera en como se programó es la única?, ¿qué cambios recomendaría usted para el programa?*

## Pasos adicionales

Usted implementará algunos cambios en el programa porque ampliaremos el requerimiento, este quedará de la siguiente manera.

**Antes de consultar el último dígito del número de cédula, se preguntará si el visitante posee cédula.**

Lo que se espera ahora es que el programa indique no solamente el día en que no puede ingresar el visitante,

sino también que si no posee cédula no podrá ingresar definitivamente.

## Conclusiones

En este ejercicio, realizamos un análisis básico de cómo aplicar los condicionales en funcionalidades de la vida real, además pudimos identificar cómo podemos implementarlos de diferentes formas cada uno de ellos.

## Autoevaluación

Una vez realizado el laboratorio con los pasos adicionales y los ejercicios complementarios, realice una **autoevaluación objetiva** del nivel de comprensión de los temas expuestos. Seleccione una de las casillas.

Nivel de aprovechamiento	X
Es capaz de <b>crear</b> soluciones similares a partir de los conocimientos adquiridos	
Es capaz de <b>evaluar</b> soluciones similares y comprender su funcionamiento.	
Es capaz de <b>aplicar</b> los conocimientos adquiridos en soluciones similares	
<b>Comprende</b> los temas expuestos y su funcionamiento dentro de la solución.	
Aprendió y puede <b>recordar</b> los temas expuestos y su objetivo.	

Analice con su profesor los temas en que considera que aún tiene dudas, realice los ejercicios propuestos o utilice los recursos como las tutorías para reforzar dichos temas.