



## UNIVERSIDAD AUTÓNOMA DE CAMPECHE

**NOMBRE DEL PROFESOR:** Ing. Héctor Manuel Quej Cosgaya**NOMBRE DE LA PRÁCTICA:** Estructuras de decisión**PRÁCTICA NÚM.** [ 5 ]

<b>LABORATORIO:</b>	Centro de Ingeniería Computacional
<b>MATERIA:</b>	Lenguaje de Programación I
<b>UNIDAD:</b>	Subcompetencia III
<b>TIEMPO:</b>	2 horas

**OBJETIVO:**

Comprender el efecto que las diferentes estructuras de decisión tienen sobre el flujo de ejecución de un programa.

**MARCO TEÓRICO:**

Las estructuras de decisión proveen a los programas computacionales una forma de decidir un camino a seguir cuando surge una determinada situación. Dado que los programas deben estar listos para funcionar sin importar los datos que reciban, es fundamental aprender las diferentes maneras que tenemos de controlar el flujo de ejecución de un programa, para así poder manipular nuestros datos de una manera correcta y eficiente. En la práctica, se procede a codificar pequeños ejemplos de estructuras de decisión, y posteriormente, se utiliza una herramienta de apoyo para visualizar el flujo de ejecución que sigue el programa.

**LISTA DE MATERIALES:**

- Java SDK
- Entorno de Desarrollo Integrado NetBeans 7.0 o superior.

**EQUIPO DE LABORATORIO:**

- Computadora Personal

**DESARROLLO DE LA PRÁCTICA:****Preparativos: Configuración de un proyecto en NetBeans**

Lo primero que tenemos que hacer es crear un proyecto para trabajar. Si nunca has trabajado con un Entorno de Desarrollo Integrado (IDE) antes, continúa leyendo. Si ya tienes experiencia trabajando con NetBeans, puedes ir directamente al paso 8.

1. Abre el Entorno de Desarrollo Integrado (IDE) **NetBeans**. Normalmente, hay un acceso directo en el Escritorio para él.



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

2. Crea un nuevo **Proyecto** haciendo clic en el botón “**Nuevo Proyecto...**” de la barra de herramientas estándar de **NetBeans**, o presionando la combinación de teclas Ctrl + Mayus + N.
3. Selecciona la categoría **Java**, y tipo de proyecto **Aplicación Java** en el cuadro de diálogo **Nuevo Proyecto**. Haz clic en el botón “**Siguiente**”.

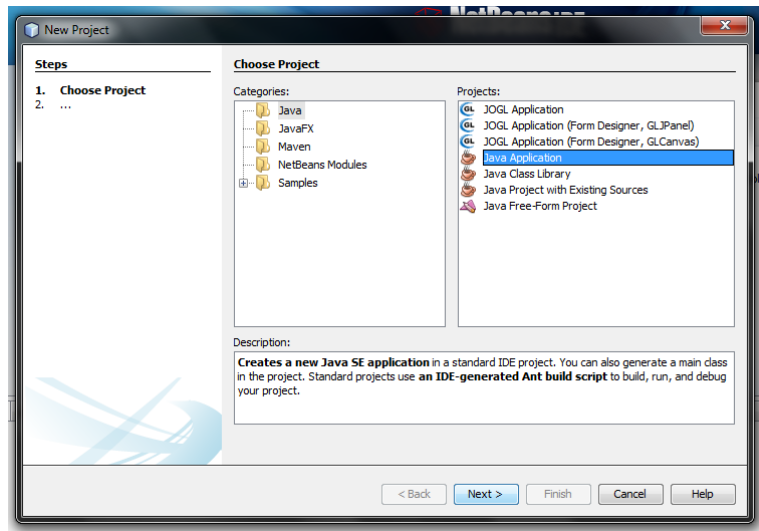


Fig. 1 – Selección del tipo de proyecto

4. Escribe un nombre para tu proyecto en el cuadro de texto “**Nombre del proyecto**”. Deja la ubicación y la carpeta del proyecto en su ubicación por default, y **desactiva** la casilla de verificación que dice “**Crear clase principal**”. Haz clic en el botón finalizar.

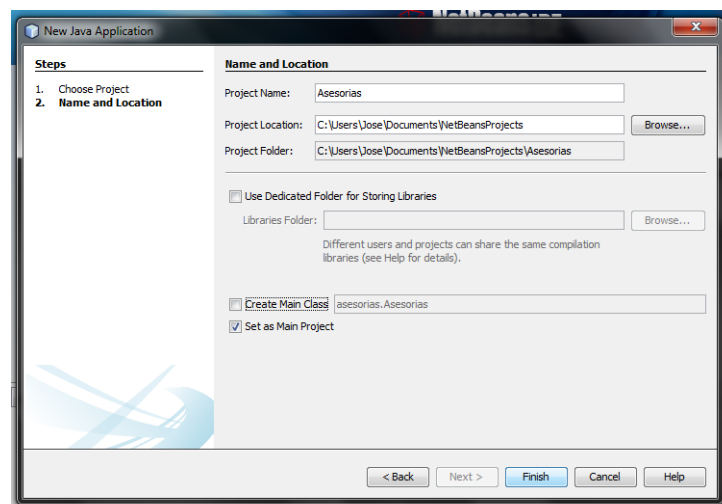


Fig. 2 – Configuración del proyecto



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

5. Crea un nuevo archivo dentro de tu proyecto recién creado haciendo clic en el botón **“Nuevo Archivo...”** de la barra de herramientas estándar de **NetBeans**, o presionando la combinación de teclas **Ctrl + N**.
6. Selecciona como categoría **Java**, y tipo de archivo **Clase Java** en el cuadro de diálogo **Nuevo Archivo**. Haz clic en el botón **“Siguiete”**.

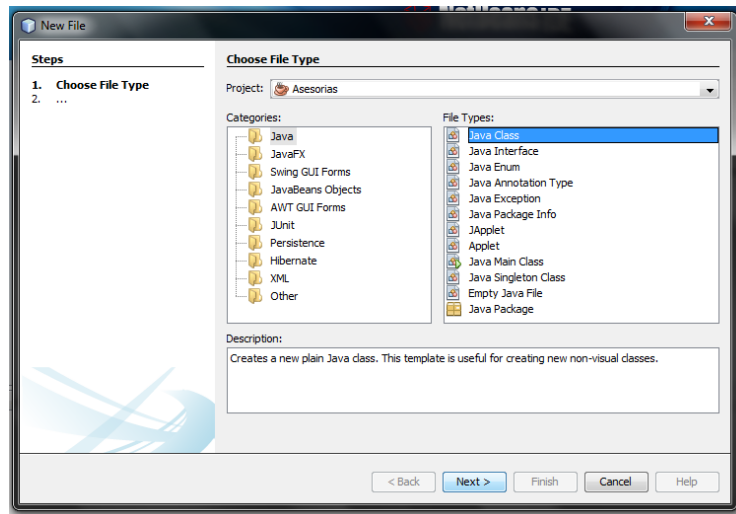


Figura 3 – Selección del tipo de archivo

7. Nombra a tu archivo **“Practica5”** en el cuadro de texto **“Nombre de Clase”**. Haz clic en el botón **“Finalizar”**.

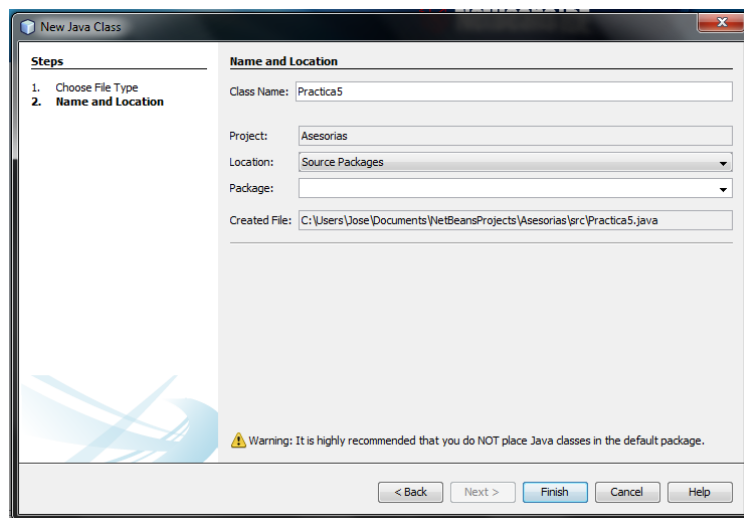


Figura 4 – Configuración de la clase **“Practica5”**

Eso es todo lo que necesitamos configurar por ahora. Aparecerá en el IDE un editor de texto con la declaración de nuestra clase **“Practica5”**.



## UNIVERSIDAD AUTÓNOMA DE CAMPECHE

Primera parte: **Sentencia if**

8. **Borra** todos los comentarios añadidos por el IDE, de manera que la declaración de la clase "public class Practica5{" sea la línea número 1.
9. **Añade** el siguiente fragmento de código dentro de la clase "Practica5":

```
public static void main(String[] args) {  
    // Primera parte: Sentencia if  
    boolean condicion = true;  
    System.out.println("Entrando al if");  
    if (condicion) {  
        System.out.println("Estamos dentro del if");  
    }  
    System.out.println("Estamos fuera del if");  
}
```

Asegúrate que la declaración del método main sea la línea número 2, y que copies el código respetando los saltos de línea especificados arriba (ahora verás el porqué).

10. Del lado izquierdo, en la pequeña barra que lleva la cuenta de los números de línea, haz clic en las líneas **5, 6, 7 y 9** para marcarlas como **breakpoints**. Observa que el número de línea es sustituido por un pequeño rectángulo rojo. Tu código debe lucir como el siguiente:

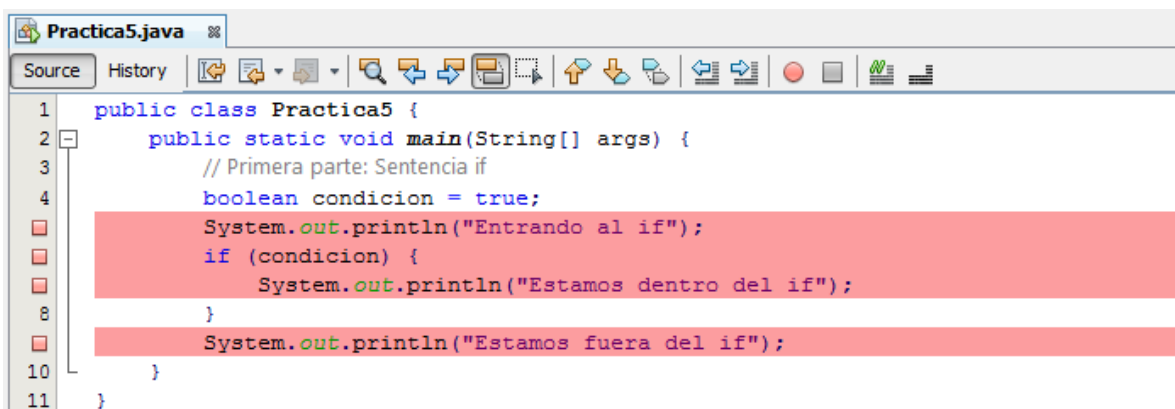


Fig. 5 – Creación de breakpoints en NetBeans

11. En el menú **Depuración**, selecciona la opción **Depurar Archivo**, o presiona la combinación de teclas **Ctrl + Mayus + F5** para ejecutar el programa en modo **Depuración**. Aparecerá la barra de herramientas **Depuración**.
12. Observa que el color de la línea 5 ha cambiado de rojo a **verde**. El programa ha empezado a ejecutarse, pero se encontró con el **breakpoint** de la línea 5, por lo que pausa su ejecución. La línea 5 se muestra verde para indicar que es la línea en la que se encuentra el **contador** del programa, y que está lista para ejecutarse.



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

```
1 public class Practica5 {
2     public static void main(String[] args) {
3         // Primera parte: Sentencia if
4         boolean condicion = true;
5         System.out.println("Entrando al if");
6         if (condicion) {
7             System.out.println("Estamos dentro del if");
8         }
9         System.out.println("Estamos fuera del if");
10    }
11 }
```

Fig. 6 – La sentencia 5 lista para ejecutarse, y la barra de herramientas Depuración

13. En la barra de herramientas **Depuración**, haz clic en el botón **Continuar**, o presiona la tecla **F5**. La línea 5 se ejecuta, imprimiendo el mensaje hacia la salida. El programa se encuentra el **breakpoint** de la línea 6, por lo que pausa su ejecución de nuevo.
14. La línea 6 es la condición de la sentencia **if**. Dado que definimos a **condicion** como **true**, veremos que el flujo entra en el bloque del **if**. Presiona el botón **Continuar** para comprobarlo.

```
1 public class Practica5 {
2     public static void main(String[] args) {
3         // Primera parte: Sentencia if
4         boolean condicion = true;
5         System.out.println("Entrando al if");
6         if (condicion) {
7             System.out.println("Estamos dentro del if");
8         }
9         System.out.println("Estamos fuera del if");
10    }
11 }
```

Output | Javadoc | Variables

Asesorias (debug) x Debugger Console x

debug:  
Entrando al if

Fig. 7 – El flujo del programa entra al bloque if

15. Presiona el botón **Continuar** una vez más. La sentencia de la línea 7 se ejecuta, imprimiendo su mensaje en pantalla. El control del programa se detiene de nuevo



## UNIVERSIDAD AUTÓNOMA DE CAMPECHE

en el **breakpoint** de la línea 9.

16. Presiona el botón **Continuar** una última vez. La sentencia de la línea 9 se ejecuta, imprimiendo su mensaje en pantalla. El contador del programa alcanza el final del código, y termina de ejecutarse.

Hemos visto ahora el flujo del programa en acción, viendo el camino que sigue su ejecución. Ahora veremos que sucede cuando la condición de un bloque **if** no se cumple.

17. Cambia el valor de la variable `condicion` por `false`. Ejecuta el programa de nuevo en modo **Depuración** y observa lo que sucede con el flujo del programa.

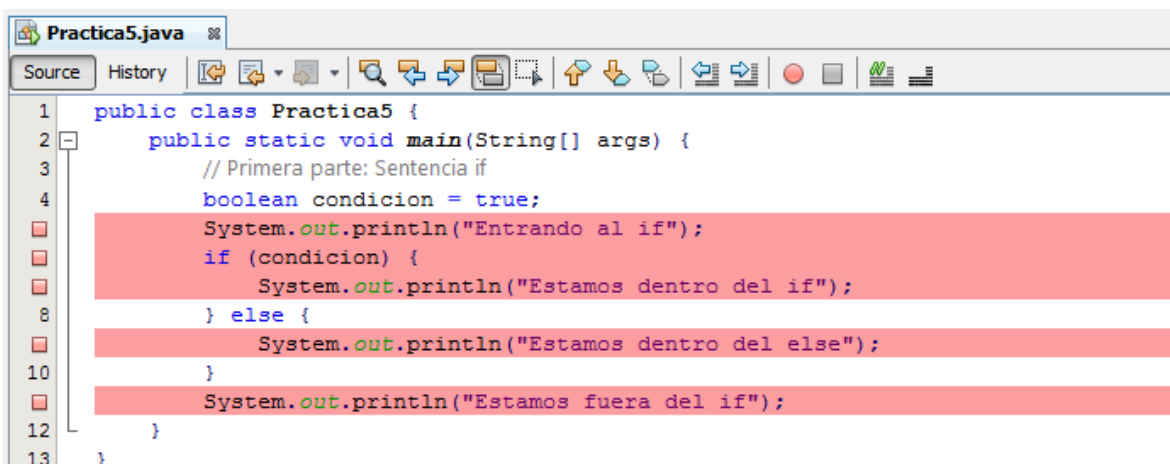
Dado que la condición ahora es `false`, el contador del programa ignora completamente la instrucción que se encuentra dentro del bloque **if**.

Segunda parte: **Sentencia if – else**.

18. Añade el siguiente fragmento de código al final de la línea 8.

```
else {  
    System.out.println("Estamos dentro del else");  
}
```

19. Añade un **breakpoint** en la nueva línea 9. Debes tener **breakpoints** en las líneas 5, 6, 7, 9 y 11. Tu código debe lucir como el siguiente:



```
1 public class Practica5 {  
2     public static void main(String[] args) {  
3         // Primera parte: Sentencia if  
4         boolean condicion = true;  
5         System.out.println("Entrando al if");  
6         if (condicion) {  
7             System.out.println("Estamos dentro del if");  
8         } else {  
9             System.out.println("Estamos dentro del else");  
10        }  
11        System.out.println("Estamos fuera del if");  
12    }  
13 }
```

Fig. 8 – Breakpoints para la sentencia if – else

20. Ejecuta el programa de nuevo en modo **Depuración**. Observa el flujo del programa para ver el camino que sigue.
21. Borra todos los **breakpoints** que creaste hasta el momento antes de pasar a la siguiente sección haciendo clic sobre ellos.



## UNIVERSIDAD AUTÓNOMA DE CAMPECHE

Tercera parte: **Sentencia if – else if**

22. Añade el siguiente fragmento de código después de la línea 11:

```
// Tercera parte: Sentencia if - else if
int calificacion = 85;
if (calificacion > 90) {
    System.out.println("A");
} else if (calificacion > 80) {
    System.out.println("B");
} else if (calificacion > 70) {
    System.out.println("C");
} else if (calificacion > 60) {
    System.out.println("D");
} else {
    System.out.println("F");
}
System.out.println("Fuera del if - else if");
```

23. Añade **breakpoints** en las líneas **14 – 21**, y en la línea **23**. Tu código debe lucir como el siguiente:

```
Practica5.java
Source History
11 System.out.println("Estamos fuera del if");
12 // Tercera parte: Sentencia if - else if
13 int calificacion = 85;
14 if(calificacion > 90){
15     System.out.println("A");
16 } else if (calificacion > 80){
17     System.out.println("B");
18 } else if (calificacion > 70){
19     System.out.println("C");
20 } else if (calificacion > 60){
21     System.out.println("D");
22 } else {
23     System.out.println("F");
24 }
25 System.out.println("Fuera del if - else if");
26 }
27 }
```

Fig. 9 – La sentencia if – else if

24. Ejecuta el programa de nuevo en modo **Depuración**. Observa el camino que sigue el flujo del programa.
25. Cambia el valor de calificacion por 50. Observa los cambios en el camino que sigue el flujo del programa.
26. Borra todos los **breakpoints** que añadiste en esta sección antes de pasar a la



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

siguiente parte.

Cuarta parte: **Sentencia switch**

27. Añade el siguiente fragmento de código después de la línea 25:

```
// Cuarta parte: Sentencia switch
int dia = 5;
switch (dia) {
    case 1 : System.out.println("Lunes");
        break;
    case 2 : System.out.println("Martes");
        break;
    case 3 : System.out.println("Miércoles");
        break;
    case 4 : System.out.println("Jueves");
        break;
    case 5 : System.out.println("Viernes");
        break;
    case 6 : System.out.println("Sábado");
        break;
    case 7 : System.out.println("Domingo");
        break;
    default : System.out.println("Día inválido");
}
System.out.println("Fuera del switch");
```

28. Añade **breakpoints** en las líneas **29, 31, 33, 35, 37, 39, 41, 43** y **45**. Tu código debe lucir como el siguiente:

```
Practica5.java
Source History
25 System.out.println("Fuera del if - else if");
26 // Cuarta parte: Sentencia switch
27 int dia = 5;
28 switch(dia){
29     case 1 : System.out.println("Lunes");
30         break;
31     case 2 : System.out.println("Martes");
32         break;
33     case 3 : System.out.println("Miércoles");
34         break;
35     case 4 : System.out.println("Jueves");
36         break;
37     case 5 : System.out.println("Viernes");
38         break;
39     case 6 : System.out.println("Sábado");
40         break;
41     case 7 : System.out.println("Domingo");
42         break;
43     default : System.out.println("Día inválido");
44 }
45 System.out.println("Fuera del switch");
46 }
47 }
```

Fig. 10 – La sentencia switch





**UNIVERSIDAD AUTÓNOMA DE CAMPECHE**

29. Ejecuta el programa en modo **Depuración**. Observa el camino que sigue el flujo del programa.
30. Cambia el valor de día por 9. Observa el cambio en el camino del flujo del programa.

**Fin de la práctica.**

**RETROALIMENTACIÓN:**

- Escribe un programa simple que pida 3 números enteros al usuario. Utiliza las estructuras de decisión para mostrar luego el mayor y el menor de ellos.

**RECOMENDACIONES ADICIONALES:**

- Investiga más acerca del proceso de depuración y sus beneficios,

**BIBLIOGRAFÍA:**

- Dean, J. S., & Dean, R. H. (2009). Introducción a la programación con Java. México: Mc Graw Hill.
- Roberts, Simon; Heller Philip y Ernest, Michael (1999). The Complete Java 2 Certification Study Guide. Alameda, California: SYBEX.