

Ejecución paso a paso de una aplicación en Java con Netbeans

Aquí tienes un ejemplo básico en Java para realizar la depuración paso a paso usando el depurador de NetBeans. Este código simple calcula la suma de los primeros \((n \) números enteros, y lo utilizaremos para observar cómo se ejecutan cada línea y se modifican las variables en cada paso.

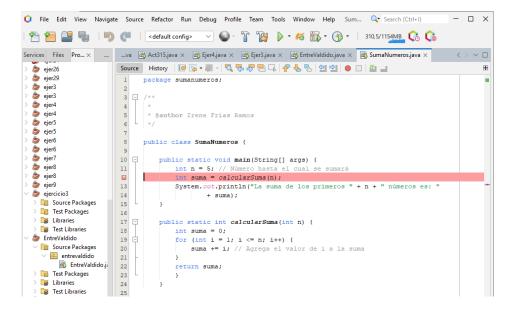
```
### Código de Ejemplo
Crea un nuevo proyecto en NetBeans, y luego agrega la clase `SumaNumeros` con el
siguiente código:
```java
public class SumaNumeros {
 public static void main(String[] args) {
 int n = 5; // Número hasta el cual se sumará
 int suma = calcularSuma(n);
 System.out.println("La suma de los primeros " + n + " números es: " + suma);
 public static int calcularSuma(int n) {
 int suma = 0;
 for (int i = 1; i \le n; i++) {
 suma += i; // Agrega el valor de i a la suma
 }
 return suma;
 }
}
```



#### ### Pasos para Depurar en NetBeans

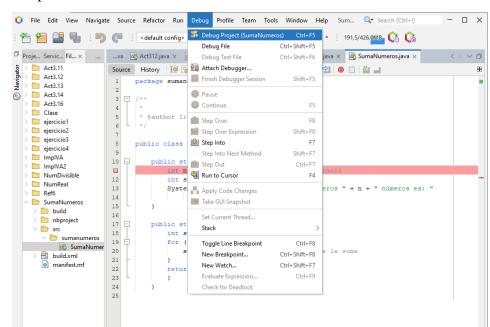
### 1. \*\*Agregar un Punto de Interrupción\*\*:

- Coloca un punto de interrupción haciendo clic en el margen izquierdo de la línea donde deseas detener la ejecución (por ejemplo, en la línea `int suma = calcularSuma(n);` en el método `main`).



## 2. \*\*Ejecutar en Modo Depuración\*\*:

- Haz clic derecho en la clase
`SumaNumeros` y selecciona
`Depurar archivo` o presiona
\*\*Ctrl+F5\*\*. El programa comenzará a ejecutarse hasta el punto de interrupción (por la parte de menú por la que llegamos).





- 3. \*\*Paso a Paso\*\*:
  - Usa las opciones del depurador:
    - \*\*Step Over (Paso por encima)\*\*: Ejecuta la línea actual y avanza a la siguiente línea en el mismo nivel de método. → F8

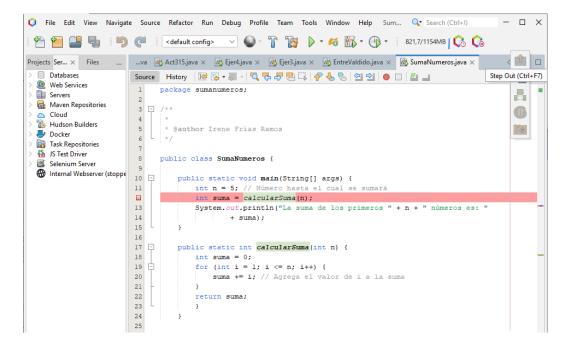
```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Sum... 🗣 Search (Ctrl+I)
! 🏲 🚰 🔡 🖣 : 🌓 🍘 : | <default config> 🗸 📦 - 🚏 👸 🕨 - 🍇 🚮 - 🐧 - : 🔲 🕕 👂 🛍 🐿 💺 : :
 ⟨ > ∨ □ Variables × Step Over (F8)
...va 🚳 Ejer4.java × 🍇 Ejer3.java × 🚳 EntreValdido.java × 🚳 SumaNumeros.java ×
Debugging
 Source History | 🔀 👼 • 🔻 • 🔼 🞝 🖶 📮 | 💎 😓 🥦 | 🕮 💇 | ● 🖂 | 💯 🚆
 H
 package sumanumeros;
 <Ente
 Static
 3 - /**
 *
٩
 > 🔷 args String[] #55(I...
 * * @author Irene Frías Ramos
 2
 ... 5
... 15
 n int
 🔷 suma int
public class SumaNumeros {
 10 public static void main(String[] args) {
.
 int n = 5; // Número hasta el cual se sumará
int suma = calcularSuma(n);
System.out.println("La suma de los primeros " + n + " números es: "
Files
 ➪
+ suma):
₽
 17 📮
 public static int calcularSuma(int n) {
 int suma = 0;
 suma += 1; 1 <= n; i++) {
 suma += i; // Agrega el valor de i a la suma
}</pre>
 22
23
 return suma;
 25
```

- \*\*Step Into (Paso dentro)\*\*: Entra en el método llamado en la línea actual para ver su ejecución interna. → F7

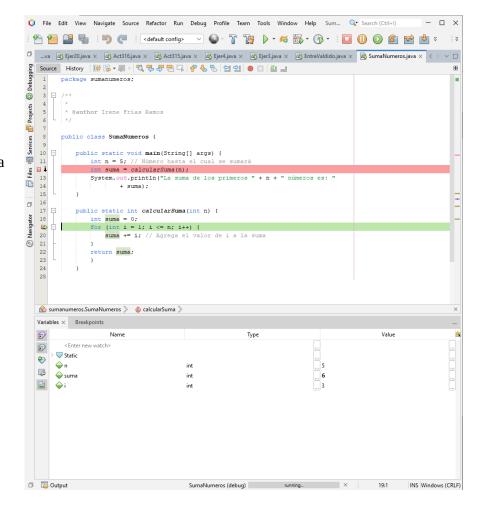
```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Sum... 🗣 Search (Ctrl+1)
: 🍄 🚰 📳 : 🍏 🍘 : 🐧 🚳 : 🗸 default config> 🔻 🚳 " 👸 🍃 * 🍪 🚯 * 🕦 * : 🔲 🕕 🔘 🚳 📸 🖢 *
🗇 ...va 🚳 Ejer4.java × 🚳 Ejer3.java × 🚳 EntreValdido.java × 🚳 SumaNumeros.java ×
 <> ∨ □
a
 4
 Name Type
 Value 🛕
 <Ente
 2 3 - /**
 Static
 ⊕
٩
 n int
Projects
 2
 * @author Irene Frías Ramos
*/
 ā
 public class SumaNumeros {
₩ Services
 10 public static void main(String[] args) {
 int n = 5; // Número hasta el cual se sumará
int suma = calcularSuma(n);
 ■ ↓
Files
 System.out.println("La suma de los primeros " + n + " números es: "
14
 + suma);
ð
 public static int calcularSuma(int n) {
 for (int i = 1; i <= n; i++) {
 suma += i; // Agrega el valor de i a la suma</pre>
 22
 return suma:
 23
```



- \*\*Step Out (Salir)\*\*: Sale del método actual y vuelve al método que lo llamó. → Ctrl. + F7.



- 4. \*\*Ver Variables\*\*:
- Abre la ventana `Variables` para ver los valores de las variables en cada paso. Observa cómo cambia `i` y `suma` en el método `calcularSuma`.





#### 5. \*\*Finalizar Depuración\*\*:

- Una vez que hayas observado la ejecución, puedes detener la depuración haciendo clic en el botón `Stop` o dejando que el programa finalice naturalmente.

```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Sum...
 The line is a second of the li
 ...va 🚯 Ejer20.java × 🚯 Act316.java × 🚳 Act315.java × 🚳 Ejer4.java × 🚳 Ejer3.java × 🚳 EntreValdido, i va × 🚳 SumaNumeros.java ×
 (C) Debugging
 package sumanumeros;
 3 🖵 /**
 Projects
 * @author Irene Frías Ramos
*/
 public class SumaNumeros {
⊞ Services
 10 📮
 public static void main(String[] args) {
 int n = 5; // Número hasta el cual se sumará
int suma = calcularSuma(n);
 11
 ■ ↓
Files
 System.out.println("La suma de los primeros " + n + " números es: '
 + suma);
ð
 public static int calcularSuma(int n) {
 17 📮
 int suma = 0;
 for (int i = 1; i <= n; i++) {
 suma += i; // Agrega el valor de i a la suma
(8)
 return suma;
 23
 24
```

Este ejemplo te permitirá ver cómo se modifica la variable `suma` a medida que se suman los números del 1 al 5, paso a paso, en el depurador de NetBeans.

```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Sum... 🗣 Search (Ctrl+1)
: 😷 🚰 🛂 🖣 : 🍏 🏴 : < default config> 🗸 🚳 🚏 🎉 Þ 🕫 🚮 🔻 🕩 🖰 : 🔲 🕕 🚳 🔯 🕹 🌣
🗇 ...va 🚳 Act316.java × 🚳 Act315.java × 🚳 Ejer4.java × 🚳 Ejer4.java × 🚳 EntreValdido.java × 🚳 SumaNumeros.java × selfsampler × < >
package sumanume C:\Users\Irene\Documents\NetBeansProjects\Act3.15\src\act3\pkg15\Act315.java
 1 package sumanume C\User\\rene\Dc
2
3 \(-\frac{1}{2} \) /**
4 | *
5 | * \(\text{8 author Irene Frias Ramos} \) /*
7
 public class SumaNumeros {

public static void main(String[] args) {
⊞ Services
 | jumic static void main(String[] args) {
| int n = 5; // Número hasta el cual se sumará
| int suma = calcularSuma(n);
| System.out.println("La suma de los primeros " + n + " números es: " |
| + suma);
| }
Files
 o
 💰 sumanumeros.SumaNumeros 🔪 🍈 calcularSuma 🔊
 Variables × Breakpoints
 Name
 Туре
 <Enter new watch>
 > 🛡 Static
 @
 □ Suma
 int
```