Introducción a la Algoritmia

Equipo de profesores del curso

Métodos

Unidad 4

Semana 9

Métodos tipo void

Variables globales y variables locales

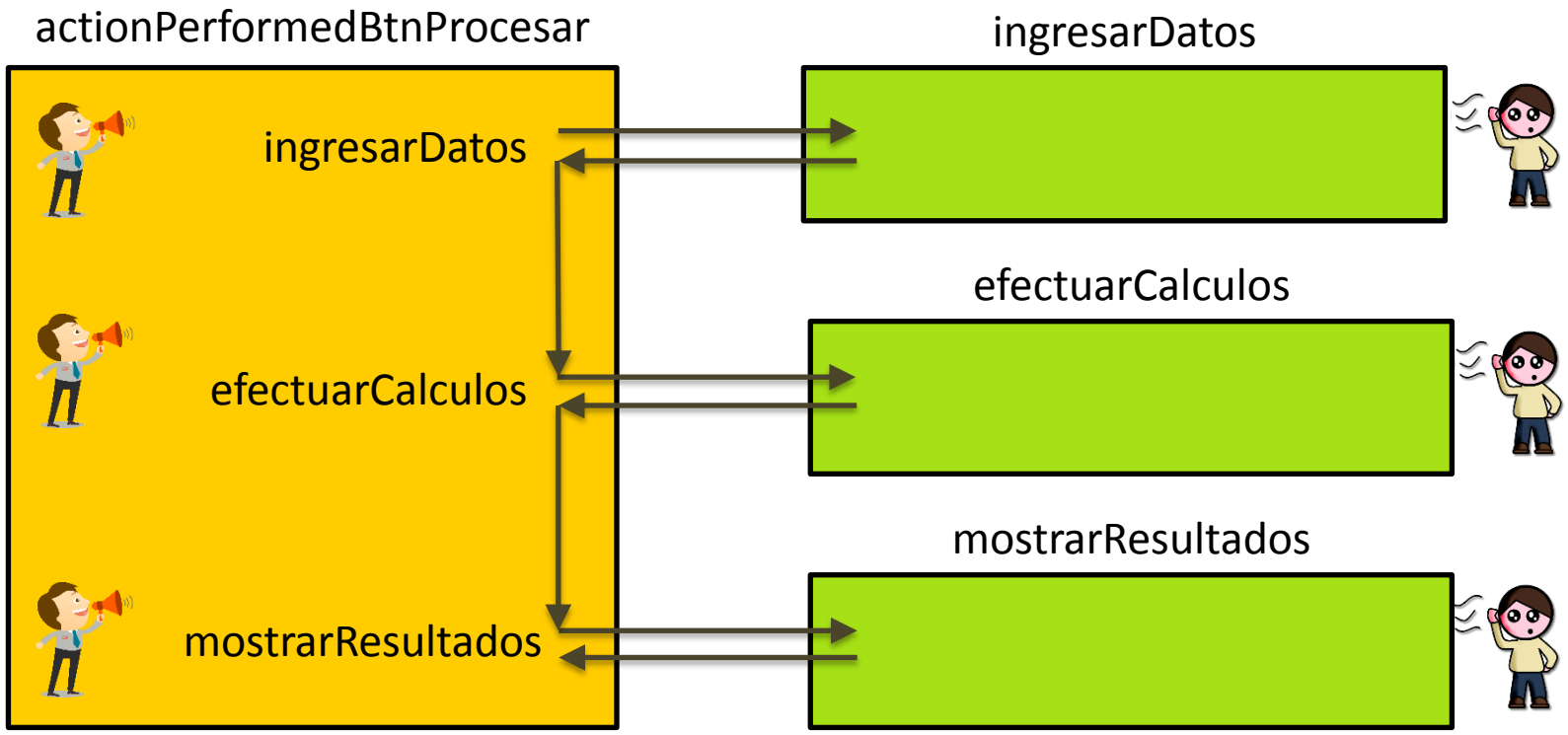
Concepto	Variabe Local	Variable Global
Declaración	En el interior de un método	En el exterior de todos los métodos
Creación	Al iniciar la ejecución del método	Al iniciar la ejecución del programa
Destrucción	Al finalizar la ejecución del método	Al finalizar la ejecución del programa
Alcance	Todo el método	Todo el programa
Inicialización	No se inicializa automáticamente	Se inicializa automáticamente

Inicialización automática de variables globales

Tipo	Valor inicial
int	0
double	0.0
String	null

Programación modular

- La programación modular es una metodología que consiste en construir un programa descomponiéndolo en **subprogramas** o **módulos** (denominados **métodos**) en el que cada método realiza una tarea específica dentro del programa.

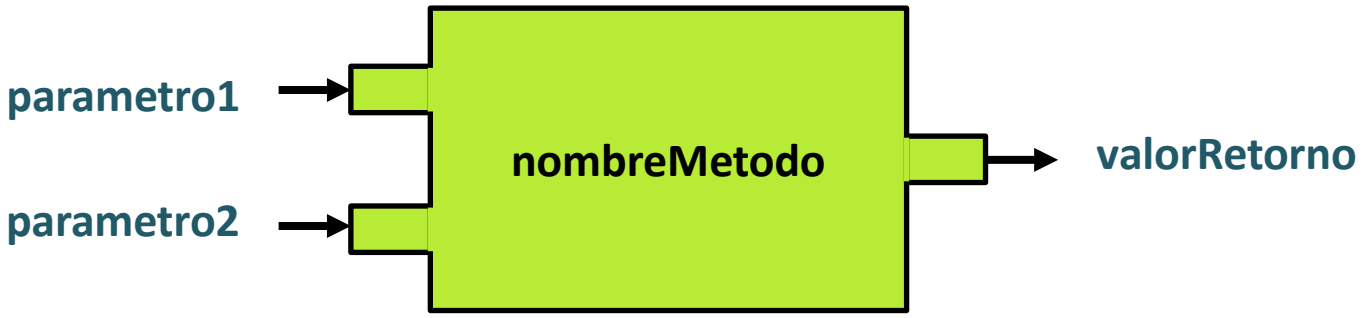


Programación modular

- Las ventajas de la programación modular son:
  - Facilidad de mantenimiento
  - Posibilidad de reutilizar módulos
  - Reducción de la complejidad

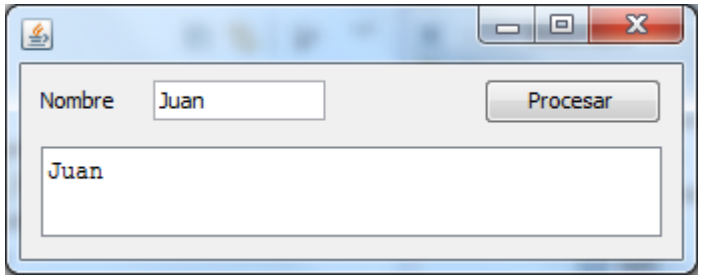
Método

- Un método es un módulo de programa que realiza una tarea específica para lo cual puede recibir datos de entrada a través de variables de entrada denominadas **parámetros** y retornar un único **valor de retorno**.



Análisis de un caso

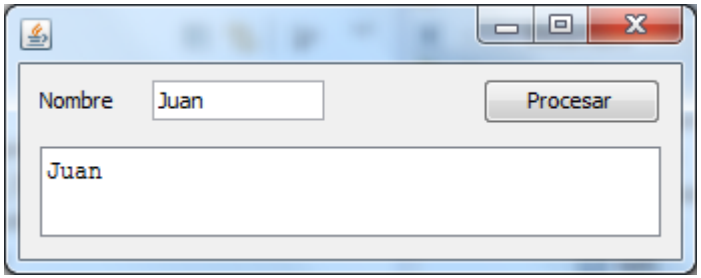
```
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    String nom;
    nom = txtNombre.getText();
    txtS.setText(nom);
}
```



Análisis de un caso

```
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
>   String nom;
    nom = txtNombre.getText();
    txtS.setText(nom);
}
```

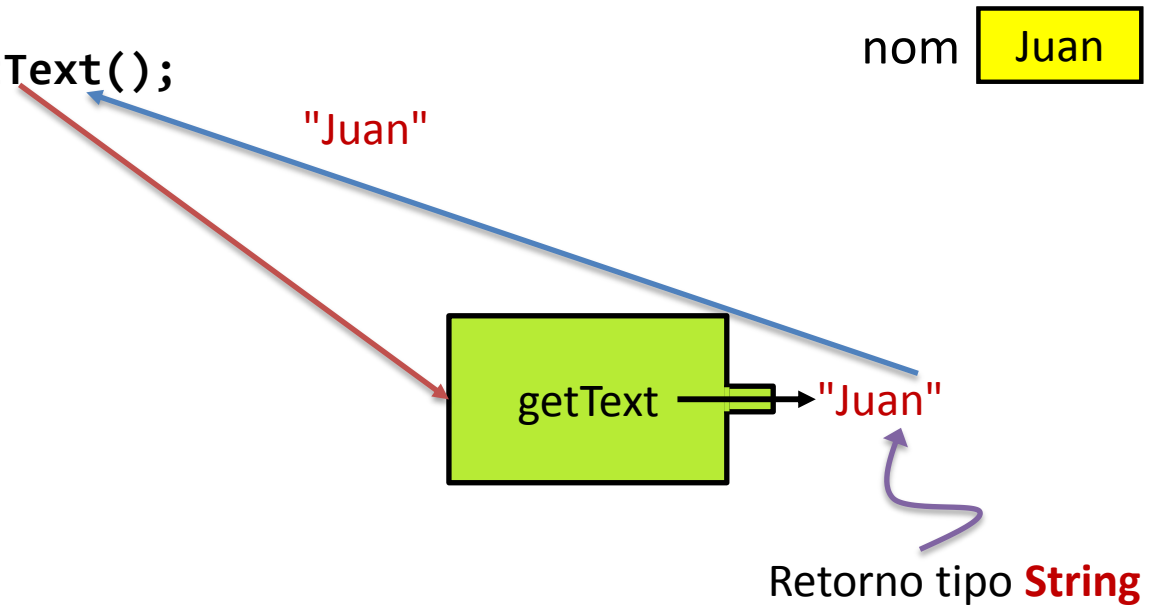
nom

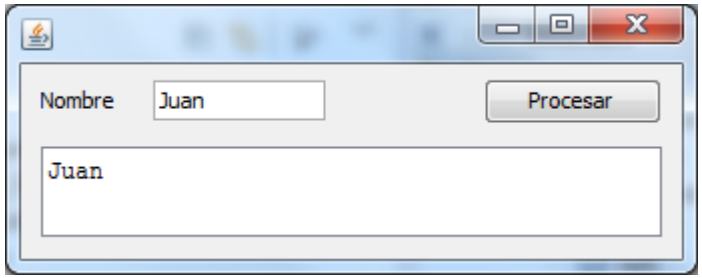


CIBERTEC

Análisis de un caso

```
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
>   String nom;
>   nom = txtNombre.getText();
    txtS.setText(nom);
}
```



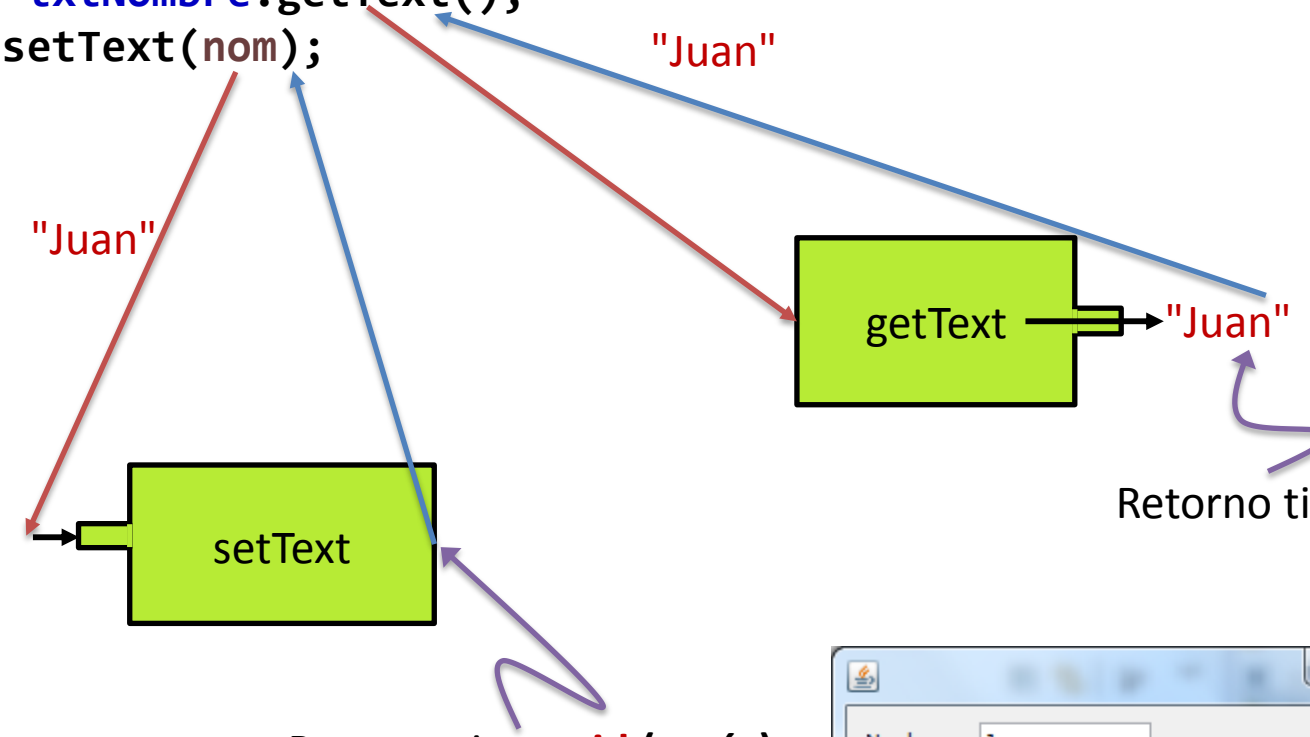


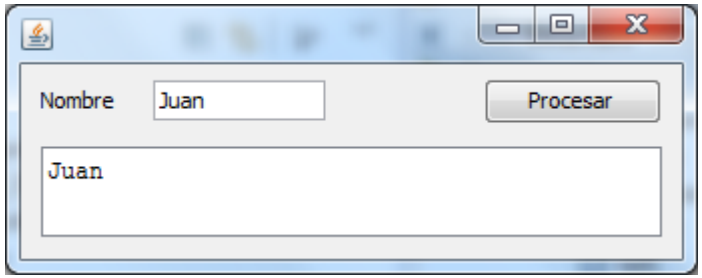
CIBERTEC

Análisis de un caso

```
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
>   String nom;
>   nom = txtNombre.getText();
>   txtS.setText(nom);
}
```

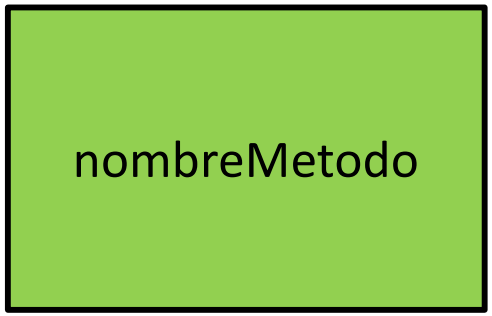
nom Juan





CIBERTEC

Métodos tipo void sin parámetros



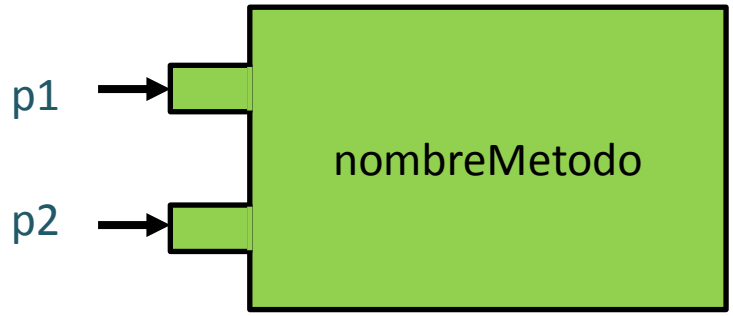
Definición

```
void nombreMetodo() {
  // Declaración de variables locales
  ...
  // Desarrollo del método
  ...
}
```

Invocación o llamada

```
nombreMetodo();
```

Métodos tipo void con parámetros



Definición

```
void nombreMetodo(tipo p1, tipo p2, ...) {
 // Declaración de variables locales
  ...
  // Desarrollo del método
  ...
}
```

Invocación o llamada

```
nombreMetodo(valor1, valor2, ...);
```