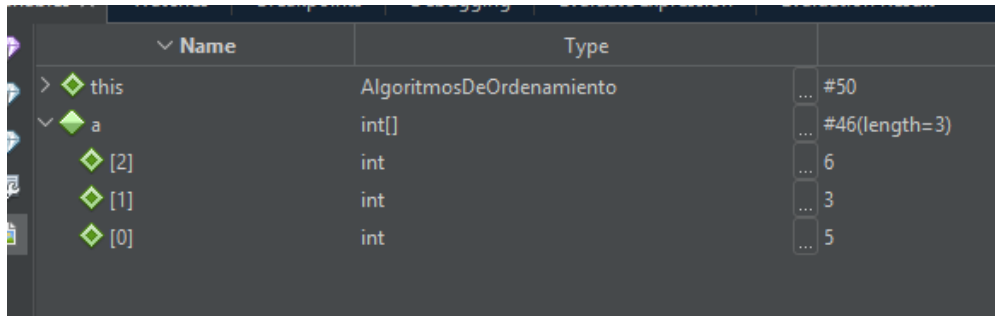
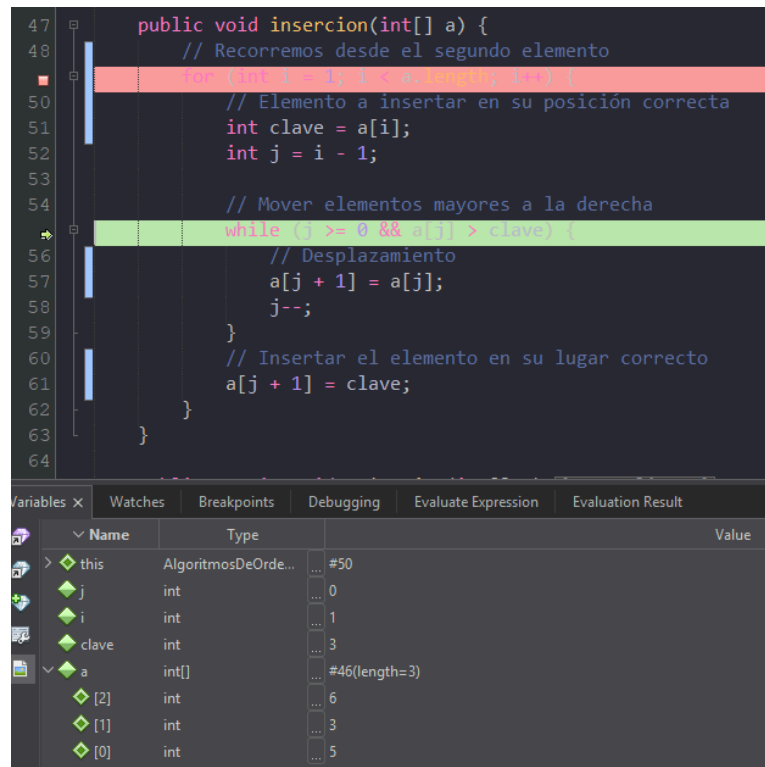


## REPORTE DE DEPURACIÓN:

1. Se declara el arreglo y es mandado al metodo de insercion:



2. Se definen las variables 'i' del for, 'clave' con valor del índice del arreglo según el valor de i y 'j' con el valor de i-1:



3. Se verifica el while y se mueven los elementos mayores a la derecha, 'j' pasa a valer -1 y 'clave' pasa a valer 3:

```
// Mover elementos mayores a la derecha
while (j >= 0 && a[j] > clave) {
    // Desplazamiento
    a[j + 1] = a[j];
    j--;
}
// Insertar el elemento en su lugar correcto
a[j + 1] = clave;
}
```

Name	Type	Value
this	AlgoritmosDeOrde...	#50
j	int	-1
i	int	1
clave	int	3
a	int[]	#46(length=3)
[2]	int	6
[1]	int	5
[0]	int	5

4. Se mueven los valores mayores a la derecha y se mueve el elemento a su lugar correcto y vuelve a iniciar el ciclo for.

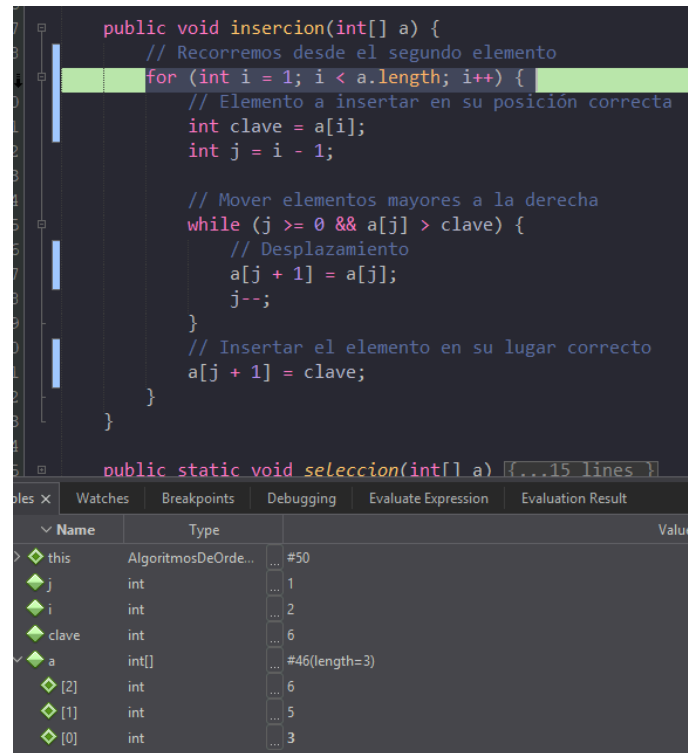
```
for (int i = 1; i < a.length; i++) {
    // Elemento a insertar en su posición correcta
    int clave = a[i];
    int j = i - 1;

    // Mover elementos mayores a la derecha
    while (j >= 0 && a[j] > clave) {
        // Desplazamiento
        a[j + 1] = a[j];
        j--;
    }
    // Insertar el elemento en su lugar correcto
    a[j + 1] = clave;
}

public static void seleccion(int[] a) {...15 lines }
```

Name	Type	Value
this	AlgoritmosDeOrde...	#50
i	int	1
a	int[]	#46(length=3)
[2]	int	6
[1]	int	5
[0]	int	3

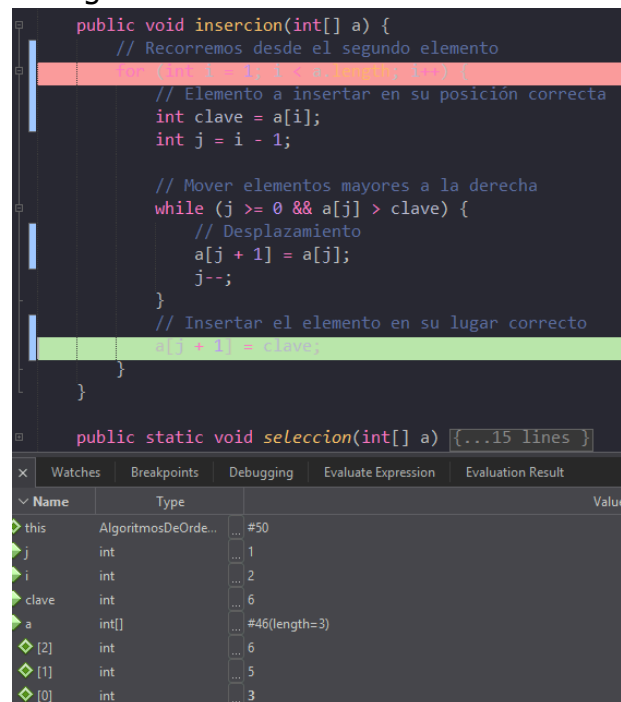
5. Vuelve a crearse las variables 'i', 'clave' y 'j', solo que 'clave' ahora vale 6 y 'i' ahora vale 2:



```
public void insercion(int[] a) {  
    // Recorremos desde el segundo elemento  
    for (int i = 1; i < a.length; i++) {  
        // Elemento a insertar en su posición correcta  
        int clave = a[i];  
        int j = i - 1;  
  
        // Mover elementos mayores a la derecha  
        while (j >= 0 && a[j] > clave) {  
            // Desplazamiento  
            a[j + 1] = a[j];  
            j--;  
        }  
        // Insertar el elemento en su lugar correcto  
        a[j + 1] = clave;  
    }  
}  
  
public static void seleccion(int[] a) { ...15 lines }
```

Name	Type	Value
this	AlgoritmosDeOrde...	#50
j	int	1
i	int	2
clave	int	6
a	int[]	#46(length=3)
a[2]	int	6
a[1]	int	5
a[0]	int	3

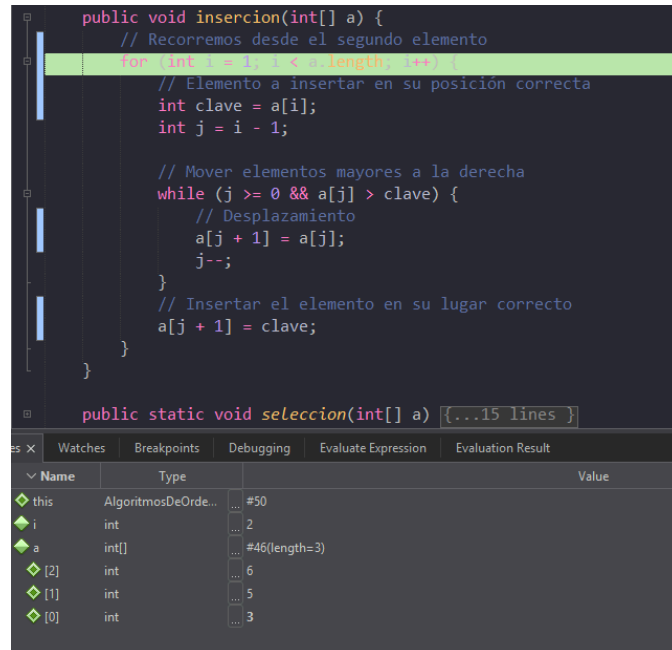
6. Dado que ya se deja de cumplir el while pasa directamente a cambiar el elemento en su lugar:



```
public void insercion(int[] a) {  
    // Recorremos desde el segundo elemento  
    for (int i = 1; i < a.length; i++) {  
        // Elemento a insertar en su posición correcta  
        int clave = a[i];  
        int j = i - 1;  
  
        // Mover elementos mayores a la derecha  
        while (j >= 0 && a[j] > clave) {  
            // Desplazamiento  
            a[j + 1] = a[j];  
            j--;  
        }  
        // Insertar el elemento en su lugar correcto  
        a[j + 1] = clave;  
    }  
}  
  
public static void seleccion(int[] a) { ...15 lines }
```

Name	Type	Value
this	AlgoritmosDeOrde...	#50
j	int	1
i	int	2
clave	int	6
a	int[]	#46(length=3)
a[2]	int	6
a[1]	int	5
a[0]	int	3

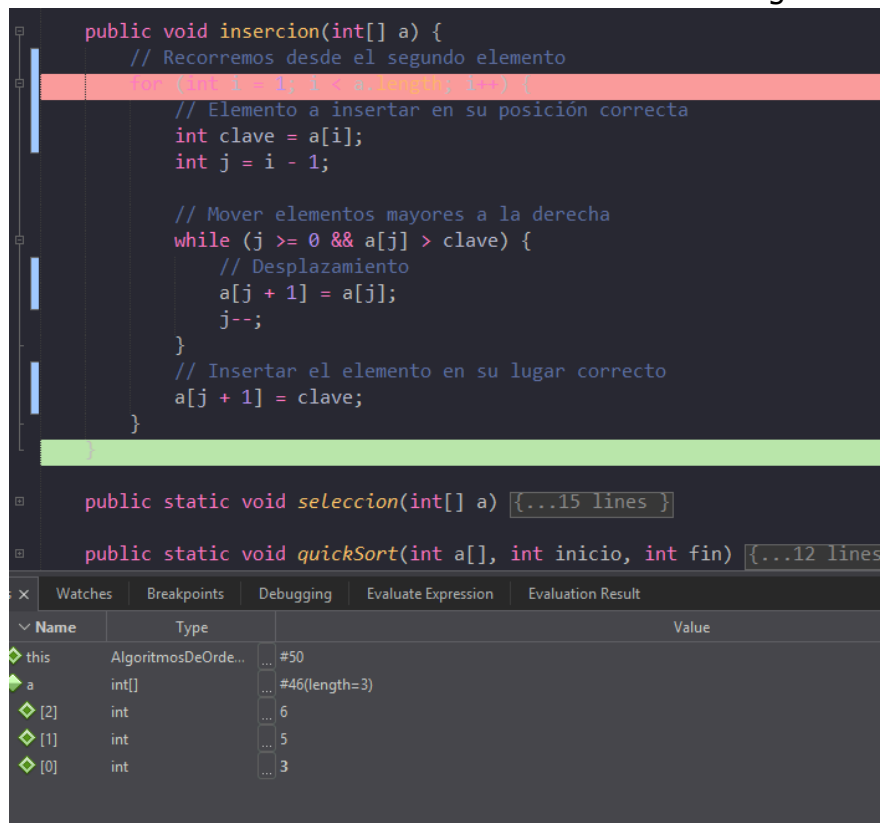
## 7. El ciclo for se termina:



```
public void insercion(int[] a) {  
    // Recorremos desde el segundo elemento  
    for (int i = 1; i < a.length; i++) {  
        // Elemento a insertar en su posición correcta  
        int clave = a[i];  
        int j = i - 1;  
  
        // Mover elementos mayores a la derecha  
        while (j >= 0 && a[j] > clave) {  
            // Desplazamiento  
            a[j + 1] = a[j];  
            j--;  
        }  
        // Insertar el elemento en su lugar correcto  
        a[j + 1] = clave;  
    }  
}  
  
public static void seleccion(int[] a) {...15 lines}
```

Name	Type	Value
this	AlgoritmosDeOrde...	#50
i	int	2
a	int[]	#46(length=3)
a[2]	int	6
a[1]	int	5
a[0]	int	3

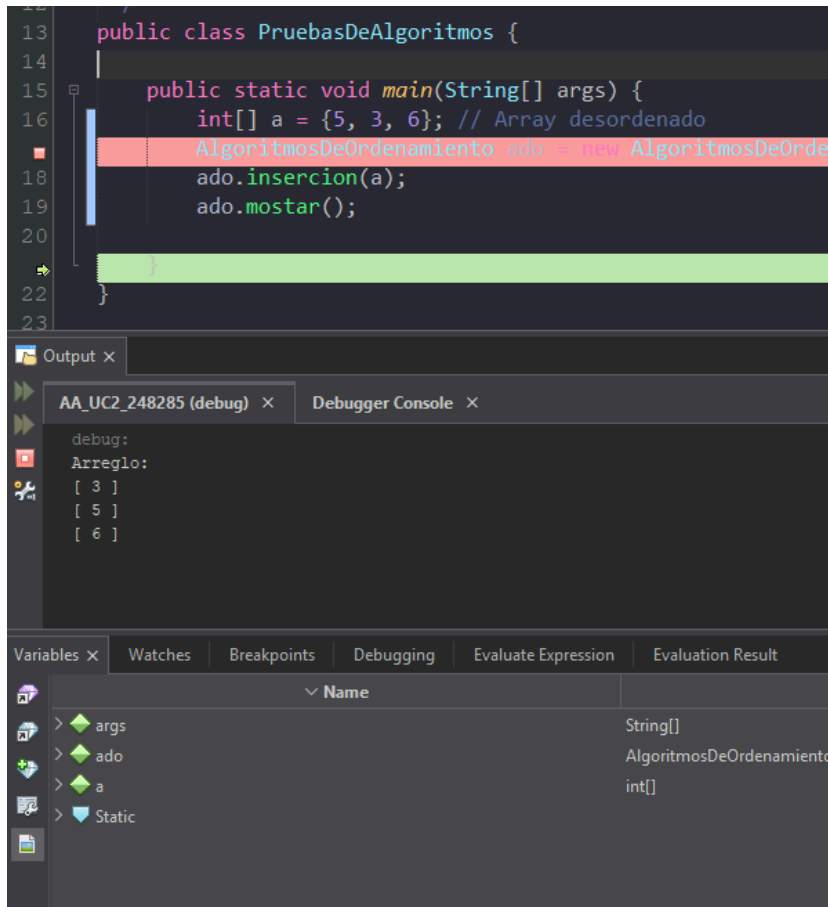
## 8. El método de inserción termina con los elementos de arreglo ordenados.



```
public void insercion(int[] a) {  
    // Recorremos desde el segundo elemento  
    for (int i = 1; i < a.length; i++) {  
        // Elemento a insertar en su posición correcta  
        int clave = a[i];  
        int j = i - 1;  
  
        // Mover elementos mayores a la derecha  
        while (j >= 0 && a[j] > clave) {  
            // Desplazamiento  
            a[j + 1] = a[j];  
            j--;  
        }  
        // Insertar el elemento en su lugar correcto  
        a[j + 1] = clave;  
    }  
}  
  
public static void seleccion(int[] a) {...15 lines}  
  
public static void quickSort(int a[], int inicio, int fin) {...12 lines}
```

Name	Type	Value
this	AlgoritmosDeOrde...	#50
a	int[]	#46(length=3)
a[2]	int	6
a[1]	int	5
a[0]	int	3

9. El sistema termina mostrando los elementos del arreglo totalmente ordenados.



```
13 public class PruebasDeAlgoritmos {
14
15     public static void main(String[] args) {
16         int[] a = {5, 3, 6}; // Array desordenado
17         AlgoritmosDeOrdenamiento ado = new AlgoritmosDeOrdenamiento();
18         ado.insercion(a);
19         ado.mostar();
20     }
21 }
22
23
```

Output x

AA\_UC2\_248285 (debug) x Debugger Console x

debug:

Arreglo:

[ 3 ]

[ 5 ]

[ 6 ]

Variables x Watches Breakpoints Debugging Evaluate Expression Evaluation Result

Name	
> args	String[]
> ado	AlgoritmosDeOrdenamiento
> a	int[]
> Static	