

**UNIVERSIDAD NACIONAL DE INGENIERÍA**



**Introducción a la Programación**

**Método de ordenamiento: Insertion sort**

**Nombre:** Steven Alejandro Álvarez Avendaño

**N.º Carnet:** 2024-1675U

**Docente:** Ing. Cristopher Larios

**Grupo:** 1M7-S

**Fecha:** domingo 05 de mayo de 2024

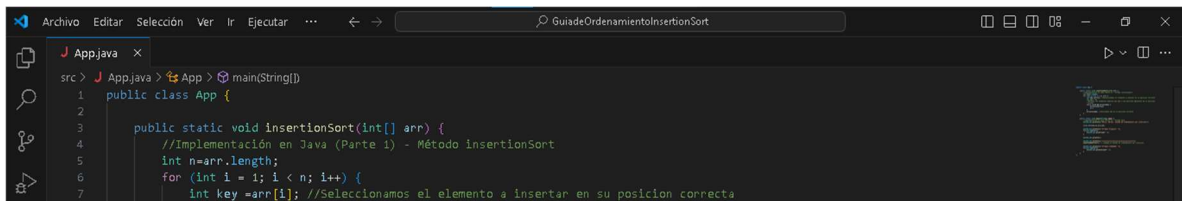
## Explique con sus propias palabras como puede ocupar este método de ordenamiento por inserción.

Este método de ordenamiento, es apropiado para usarse en arreglos con pocos valores, por lo tanto, un buen ejemplo para usar este método es el de ordenar en el sistema a los estudiantes de un colegio, ya sea por su nombre, edad o por otros atributos, ya que no es tan extenso como un inventario con cientos de productos, y así tener un mejor control y de una forma más eficiente.

## Elaboración de la Guía

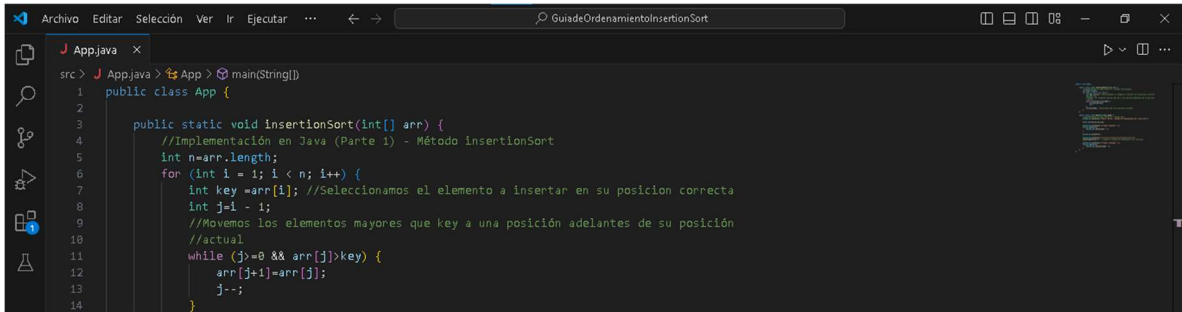
### Implementación en Java (Parte 1) - Método insertionSort

- Key representa el elemento que queremos insertar en la parte ordenada del arreglo.



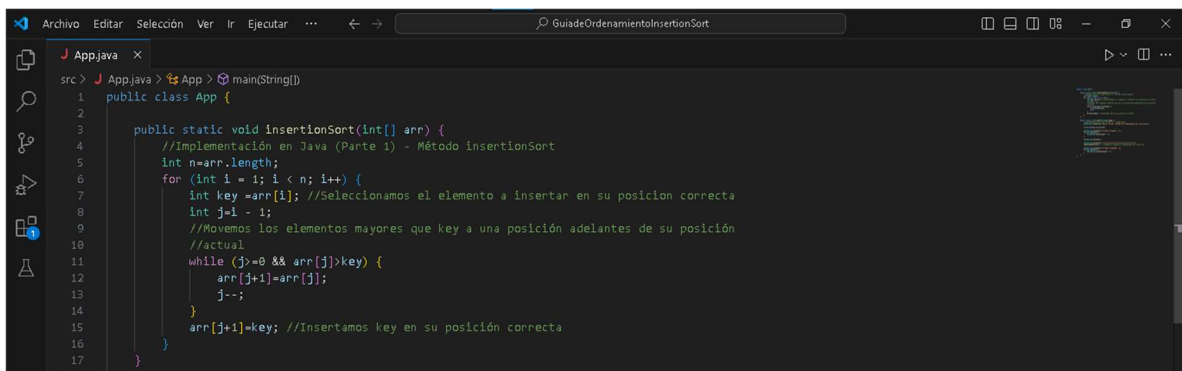
```
src > J App.java > App > main(String[])
1 public class App {
2
3     public static void insertionSort(int[] arr) {
4         //Implementación en Java (Parte 1) - Método insertionSort
5         int n=arr.length;
6         for (int i = 1; i < n; i++) {
7             int key =arr[i]; //Seleccionamos el elemento a insertar en su posición correcta
```

- El bucle While desplaza los elementos mayores que key una posición adelante para hacer espacio para key.



```
8             int key =arr[i]; //Seleccionamos el elemento a insertar en su posición correcta
9             int j=i - 1;
10            //Movemos los elementos mayores que key a una posición adelante de su posición
11            //actual
12            while (j>=0 && arr[j]>key) {
13                arr[j+1]=arr[j];
14                j--;
```

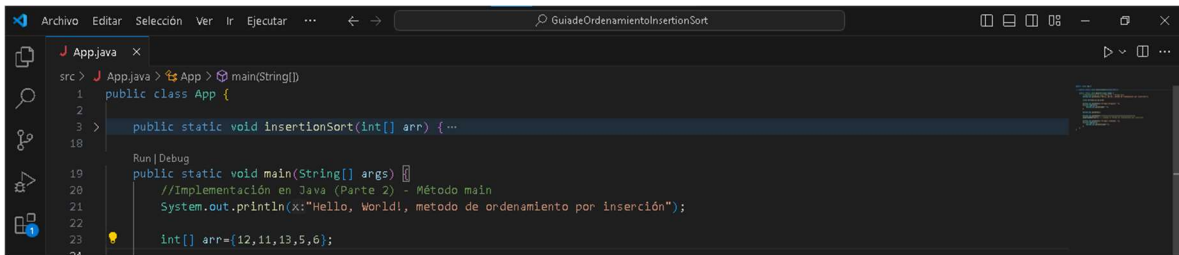
- Finalmente, key se inserta en la posición correcta j + 1.



```
15                arr[j+1]=key; //Insertamos key en su posición correcta
16            }
17        }
18    }
```

## Implementación en Java (Parte 2) - Método main

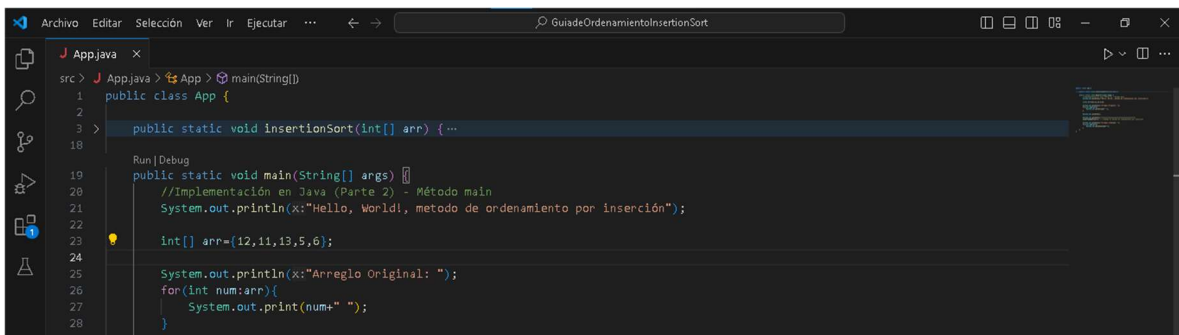
- Se crea un arreglo desordenado.



The screenshot shows an IDE window titled 'GuiadeOrdenamientoInsertionSort'. The code is in a file named 'App.java' and contains the following lines:

```
src> J App.java > App > main(String[])
1 public class App {
2
3 > public static void insertionSort(int[] arr) { ...
18
Run | Debug
19 public static void main(String[] args) {}
20 //Implementación en Java (Parte 2) - Método main
21 System.out.println(x:"Hello, World!, metodo de ordenamiento por inserción");
22
23 int[] arr={12,11,13,5,6};
24
```

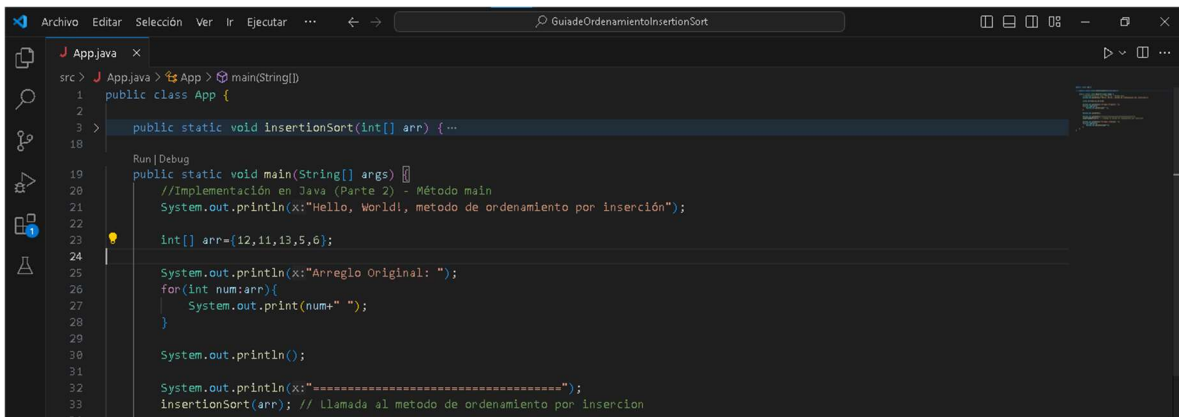
- Se imprime este arreglo antes de ordenarlo



The screenshot shows the same IDE window with additional code added to the main method:

```
src> J App.java > App > main(String[])
1 public class App {
2
3 > public static void insertionSort(int[] arr) { ...
18
Run | Debug
19 public static void main(String[] args) {}
20 //Implementación en Java (Parte 2) - Método main
21 System.out.println(x:"Hello, World!, metodo de ordenamiento por inserción");
22
23 int[] arr={12,11,13,5,6};
24
25 System.out.println(x:"Arreglo Original: ");
26 for(int num:arr){
27     System.out.print(num+" ");
28 }
29
```

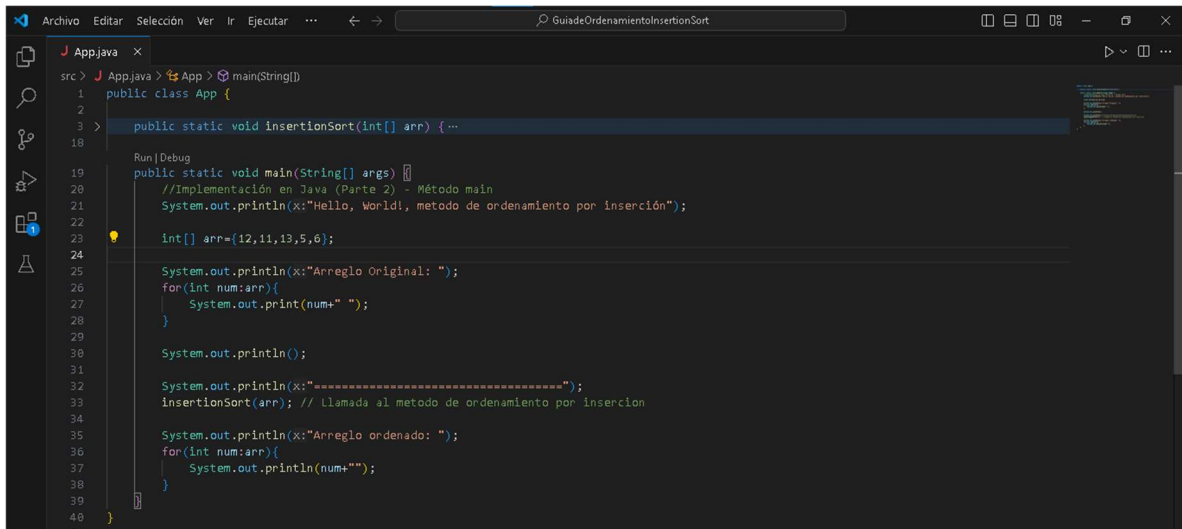
- Llamamos al método insertionSort para ordenar el arreglo.



The screenshot shows the same IDE window with the call to the insertionSort method added to the main method:

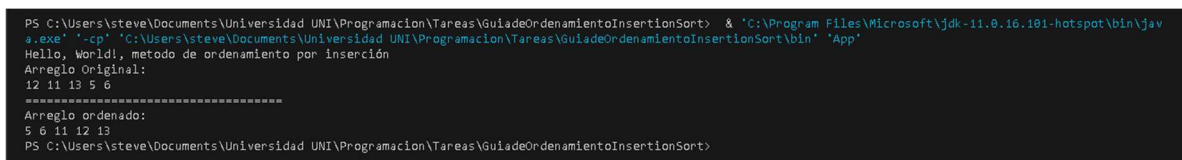
```
src> J App.java > App > main(String[])
1 public class App {
2
3 > public static void insertionSort(int[] arr) { ...
18
Run | Debug
19 public static void main(String[] args) {}
20 //Implementación en Java (Parte 2) - Método main
21 System.out.println(x:"Hello, World!, metodo de ordenamiento por inserción");
22
23 int[] arr={12,11,13,5,6};
24
25 System.out.println(x:"Arreglo Original: ");
26 for(int num:arr){
27     System.out.print(num+" ");
28 }
29
30 System.out.println();
31
32 System.out.println(x:"=====");
33 insertionSort(arr); // llamada al metodo de ordenamiento por insercion
34
```

- Imprimimos el arreglo ordenado.



```
src > J App.java > App > main(String[])
1 public class App {
2
3 > public static void insertionSort(int[] arr) { ...
18
19 Run | Debug
19 public static void main(String[] args) {
20 //Implementación en Java (Parte 2) - Método main
21 System.out.println(x:"Hello, World!, metodo de ordenamiento por inserción");
22
23 int[] arr={12,11,13,5,6};
24
25 System.out.println(x:"Arreglo Original: ");
26 for(int num:arr){
27     System.out.print(num+" ");
28 }
29
30 System.out.println();
31
32 System.out.println(x:"=====");
33 insertionSort(arr); // Llamada al metodo de ordenamiento por insercion
34
35 System.out.println(x:"Arreglo ordenado: ");
36 for(int num:arr){
37     System.out.println(num+"");
38 }
39
40 }
41 }
```

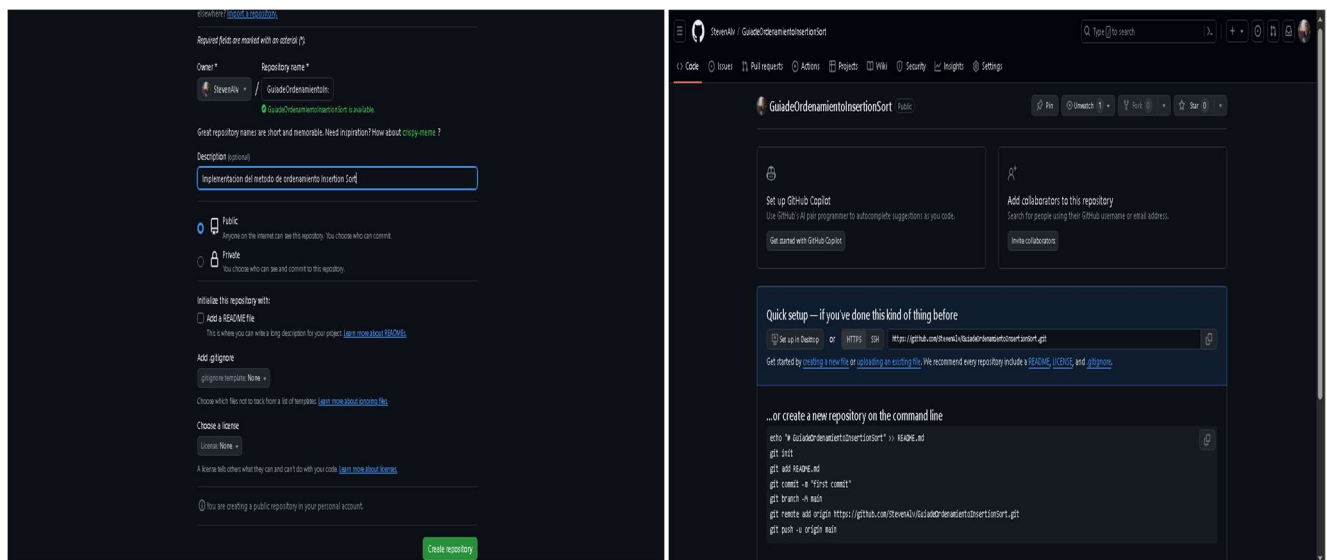
- Ejecución del Programa



```
PS C:\Users\steve\Documents\Universidad UNI\Programacion\Tareas\Guia de Ordenamiento Insertion Sort> & 'C:\Program Files\Microsoft\jdk-11.0.16-hotspot\bin\java.exe' -cp "C:\Users\steve\Documents\Universidad UNI\Programacion\Tareas\Guia de Ordenamiento Insertion Sort\bin" 'App'
Hello, World!, metodo de ordenamiento por inserción
Arreglo Original:
12 11 13 5 6
=====
Arreglo ordenado:
5 6 11 12 13
PS C:\Users\steve\Documents\Universidad UNI\Programacion\Tareas\Guia de Ordenamiento Insertion Sort>
```

## Creación y Subida al Repositorio en GitHub

- Creación del repositorio en GitHub



- Subir el proyecto por Git

[illegible]