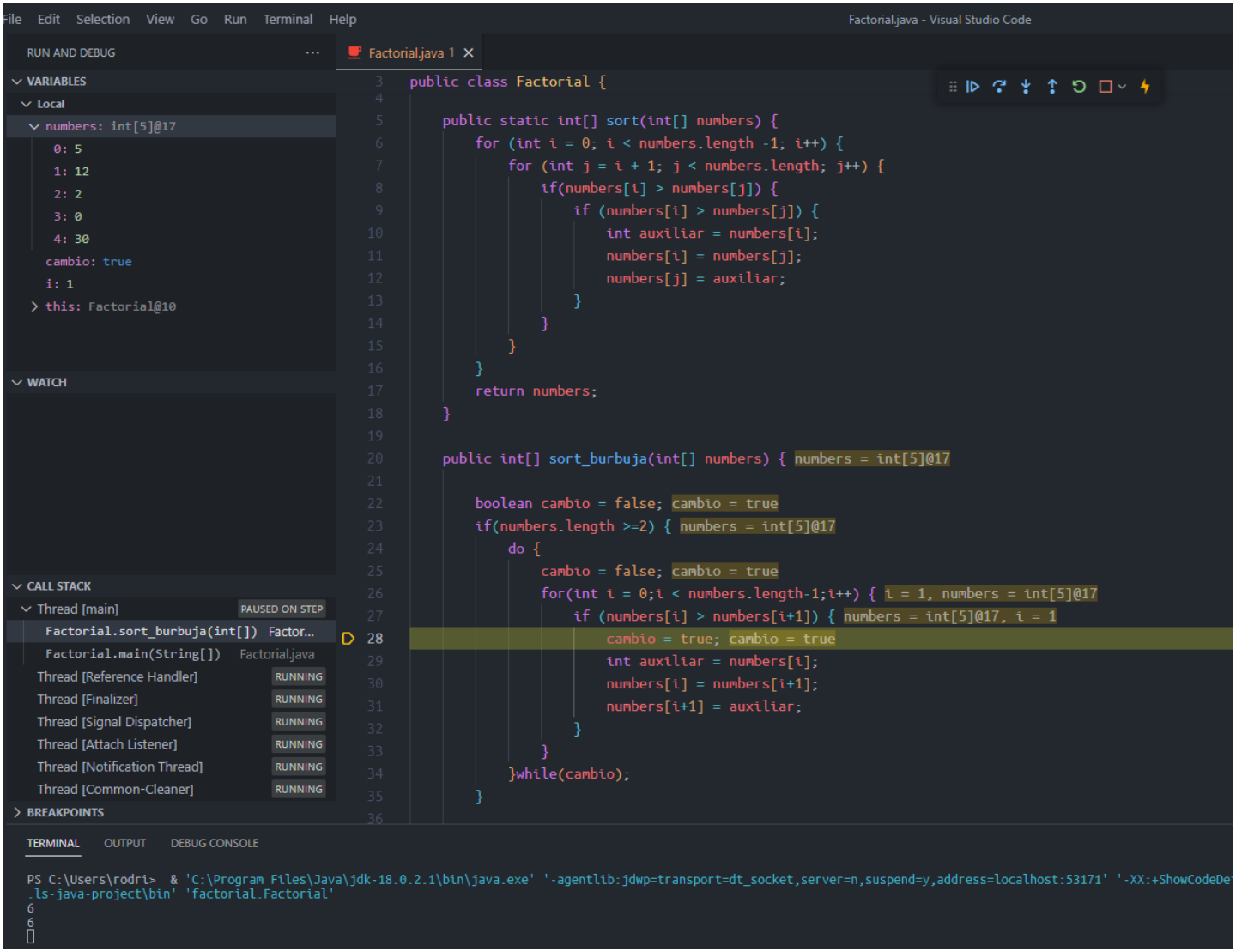
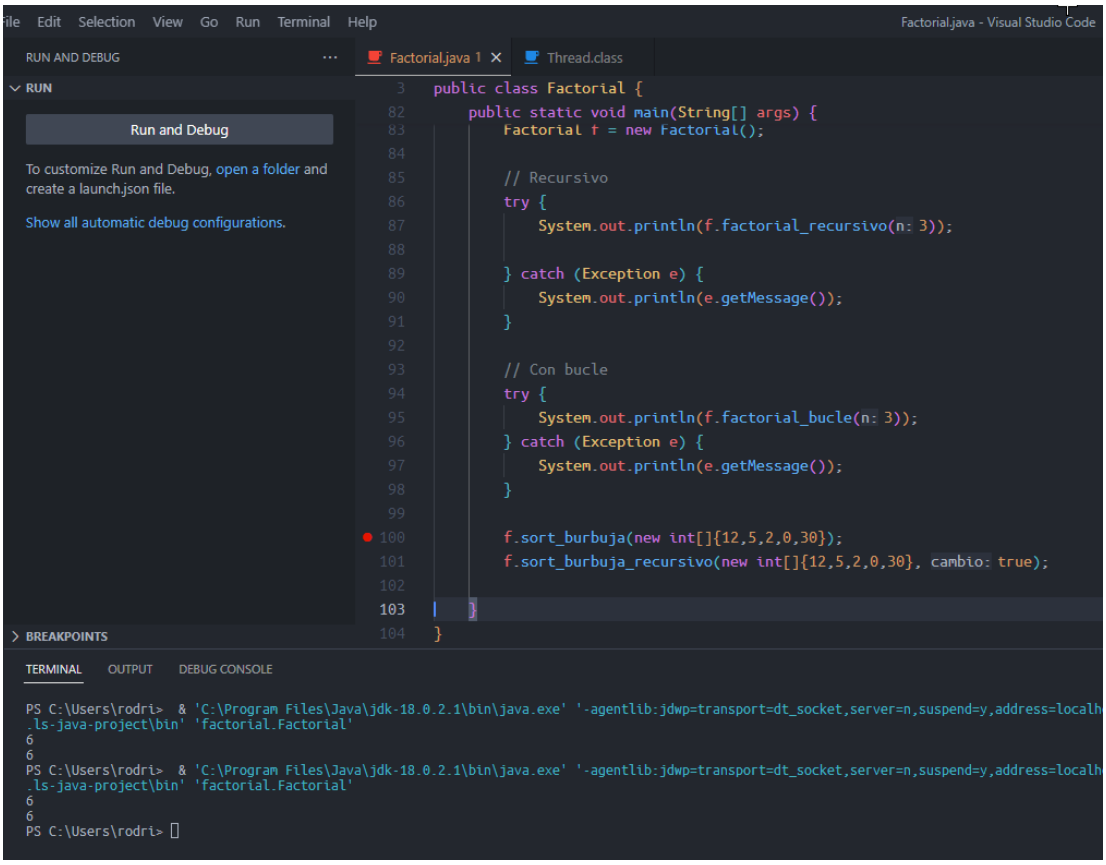

File Edit Selection View Go Run Terminal Help
Factorial.java - Visual Studio Code
RUN AND DEBUG
Factorial.java 1
VARIABLES
Local
numbers: int[5]@17
0: 5
1: 12
2: 2
3: 0
4: 30
cambio: true
i: 1
this: Factorial@10
WATCH
CALL STACK
Thread [main] PAUSED ON STEP
Factorial.sort_burbuja(int[]) Factor...
Factorial.main(String[]) Factorial.java
Thread [Reference Handler] RUNNING
Thread [Finalizer] RUNNING
Thread [Signal Dispatcher] RUNNING
Thread [Attach Listener] RUNNING
Thread [Notification Thread] RUNNING
Thread [Common-Cleaner] RUNNING
BREAKPOINTS
TERMINAL OUTPUT DEBUG CONSOLE
PS C:\Users\rodri> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:53171' '-XX:+ShowCodeDe
.ls-java-project\bin' 'factorial.Factorial'
6
6


File Edit Selection View Go Run Terminal Help
Factorial.java - Visual Studio Code
RUN AND DEBUG
Factorial.java 1
Thread.class
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
Show all automatic debug configurations.
BREAKPOINTS
TERMINAL OUTPUT DEBUG CONSOLE
PS C:\Users\rodri> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localh
.ls-java-project\bin' 'factorial.Factorial'
6
6
PS C:\Users\rodri> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localh
.ls-java-project\bin' 'factorial.Factorial'
6
6
PS C:\Users\rodri>

La idea detrás del algoritmo de ordenamiento de burbuja es comparar elementos adyacentes y, si están en el orden incorrecto, intercambiarlos. Este proceso se repite varias veces hasta que no se realicen más intercambios.