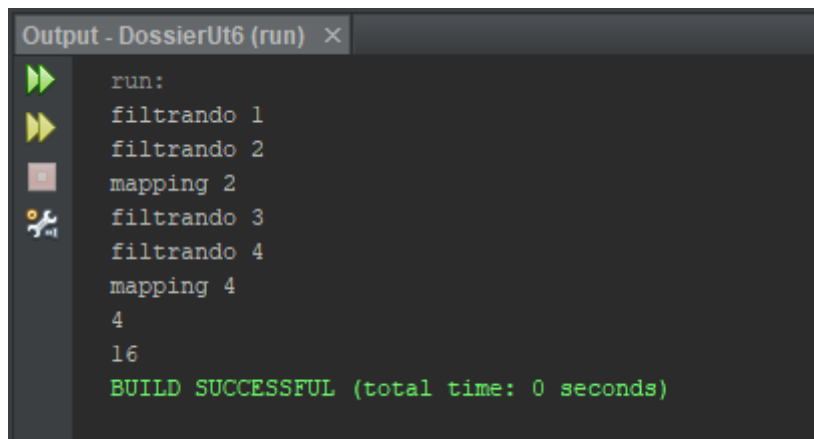


**Práctica 33:** Probar el código anterior. Agrega el código para mostrar lista que se obtuvo como resultado en pantalla Toma captura de pantalla del código y la ejecución

```
* @author Kevin Hernández García <kevinhg94@gmail.com>
*/
public class P33Main {

    public static void main(String[] args) {
        List<Integer> numeros = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8);
        List<Integer> lista = numeros.stream()
            .filter(n -> {
                System.out.println("filtrando " + n);
                return n % 2 == 0;
            })
            .map(n -> {
                System.out.println("mapping " + n);
                return n * n;
            })
            .limit(2)
            .collect(toList());

        lista.stream()
            .forEach(System.out::println);
    }
}
```



```
Output - DossierUt6 (run) ×
run:
filtrando 1
filtrando 2
mapping 2
filtrando 3
filtrando 4
mapping 4
4
16
BUILD SUCCESSFUL (total time: 0 seconds)
```

Con esta actividad podemos comprobar que los stream no se ejecutan hasta que se llega a la orden terminal, por eso no se muestran todos los valores “Filtrando n” y “mapping n”, antes de mostrarlo todo el stream puede ver el `limit(2)` y solo ejecutar las acciones necesarias para llevar a cabo la tarea, en este caso como lo que hace es buscar los números pares, solo con 4 iteraciones es suficiente.