**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA**

FACULTAD DE INGENIERÍA MINAS, GEOLOGÍA Y CIVIL

ESCUELA DE FORMACION PROFESIONAL DE INGENIERÍA DE SISTEMAS

**INFORME DE LABORATORIO 04**

**MATRICES**

09 octubre 2019

Jueves 2 - 5 pm

ASIGNATURA: IS142 Programación Orientada a Objetos

DOCENTE: M.sC. Ing. Fredy Barrientos

ALUMNO: Ramos Conga, Christian Yordi

AYACUCHO-PERÚ

2019

**LABORATORIO 005**

**1. Objetivos**

**1.1. Objetivo General**

Poner en práctica el concepto de Matrices en Java.

**1.2. Objetivos Específicos**

 Declarar, instanciar, inicializar, asignar y leer datos de una matriz.

 Iterar por medio de un ciclo for anidado los elementos de una matriz.

**2. Requisitos**

Este laboratorio requiere de conocimientos previos de manejo de variables, cadenas,

operadores, estructuras de control y arreglos de una dimensión en Java.

**3. Conceptos relacionados**

**3.1. Matrices en Java**

Otra estructura de datos en Java que podemos utilizar es una matriz. A diferencia de

un arreglo de una dimensión, una matriz podríamos verla como dos arreglos, un

arreglo maneja los renglones y otro arreglo las columnas, y al juntarlos obtenemos una

matriz. Sin embargo, veremos que existen muchas cosas similares con un arreglo, y

una vez entendido el concepto de arreglo es más sencillo entender el concepto de

matriz.

Por ejemplo, podemos definir una matriz de 4 renglones por 7 columnas, de tipo

enteros, sin embargo, puede ser de cualquier tipo que definamos.

Podemos recuperar el largo de los renglones con el código nombreArreglo.length y

podemos obtener el largo de las columnas escribiendo nombreArreglo[0].length, es

decir, que con cualquier renglón válido seleccionado podemos obtener el largo de las

columnas. Esto nos va a servir posteriormente para iterar por medio de un ciclo for

anidado cada uno de los elementos de la matriz.

De igual manera que un arreglo, no todos los elementos de una matriz deben contener

valores. Los valores que no tengan un valor asignado tendrán el valor por default

según el tipo de datos definido para la matriz.

**3.2. Declaración de una matriz**

A continuación, veamos la sintaxis para declarar una matriz:

tipo [][] nombreArreglo ó tipo nombreArreglo [][];

**3.3. Instanciar matrices**

A continuación, veremos cómo inicializar una matriz, ya que hasta el momento con

sólo declarar una variable de tipo matriz la JVM no sabe cuan largo y ancho es la

matriz, para ello debemos inicializarlo, veamos como:

nombreArreglo = new tipo[renglones][columnas];

**3.4. Inicializar los elementos de una matriz**

A continuación, veremos cómo inicializar los elementos de una matriz:

nombreArreglo[índice\_renglon][índice\_columna] = valor;

**3.5. Extraer elementos de una matriz**

A continuación, veremos cómo extraer elementos de una matriz:

variableReceptora = nombreArreglo[índice\_renglon][índice\_columna];

**4. Desarrollo del laboratorio**

**4.1. Matriz en Java**

Vamos a crear el proyecto de nombre Matrices. Los pasos para crear un proyecto ya lo

conocemos:

**5. Assignment**

**La Tarea Académica N° 04 (TA04) consiste en crear un proyecto de nombre**

**TA04\_Nombre\_ApellidoPaterno, y el paquete debe llamarse matrices. A continuación,**

**agregar una clase main para cada ejercicio, por ejemplo, para el ejercicio 1, la clase**

**main debe llamarse Ejercicio001.**

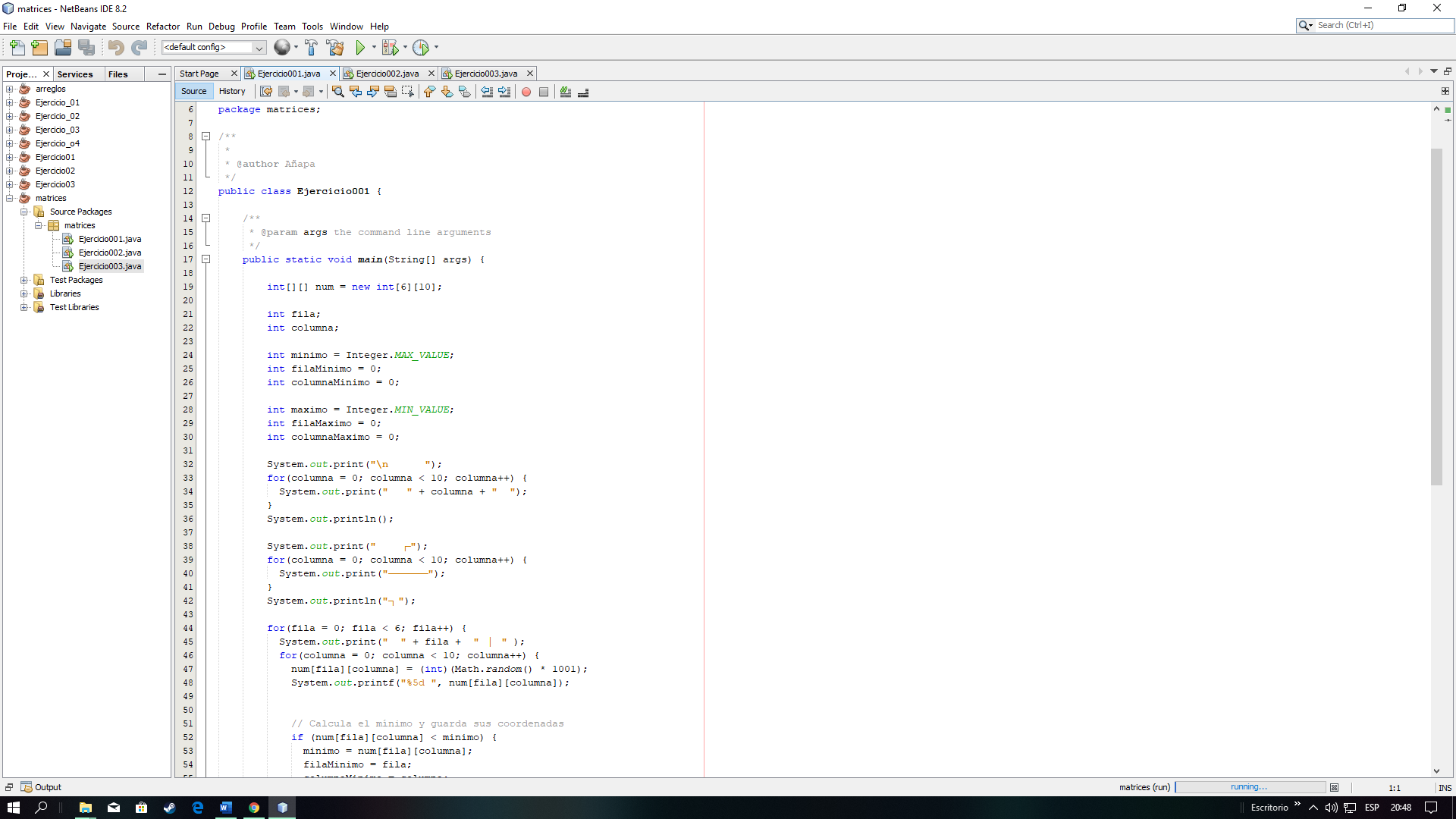
La TA04 está compuesta por los siguientes ejercicios:

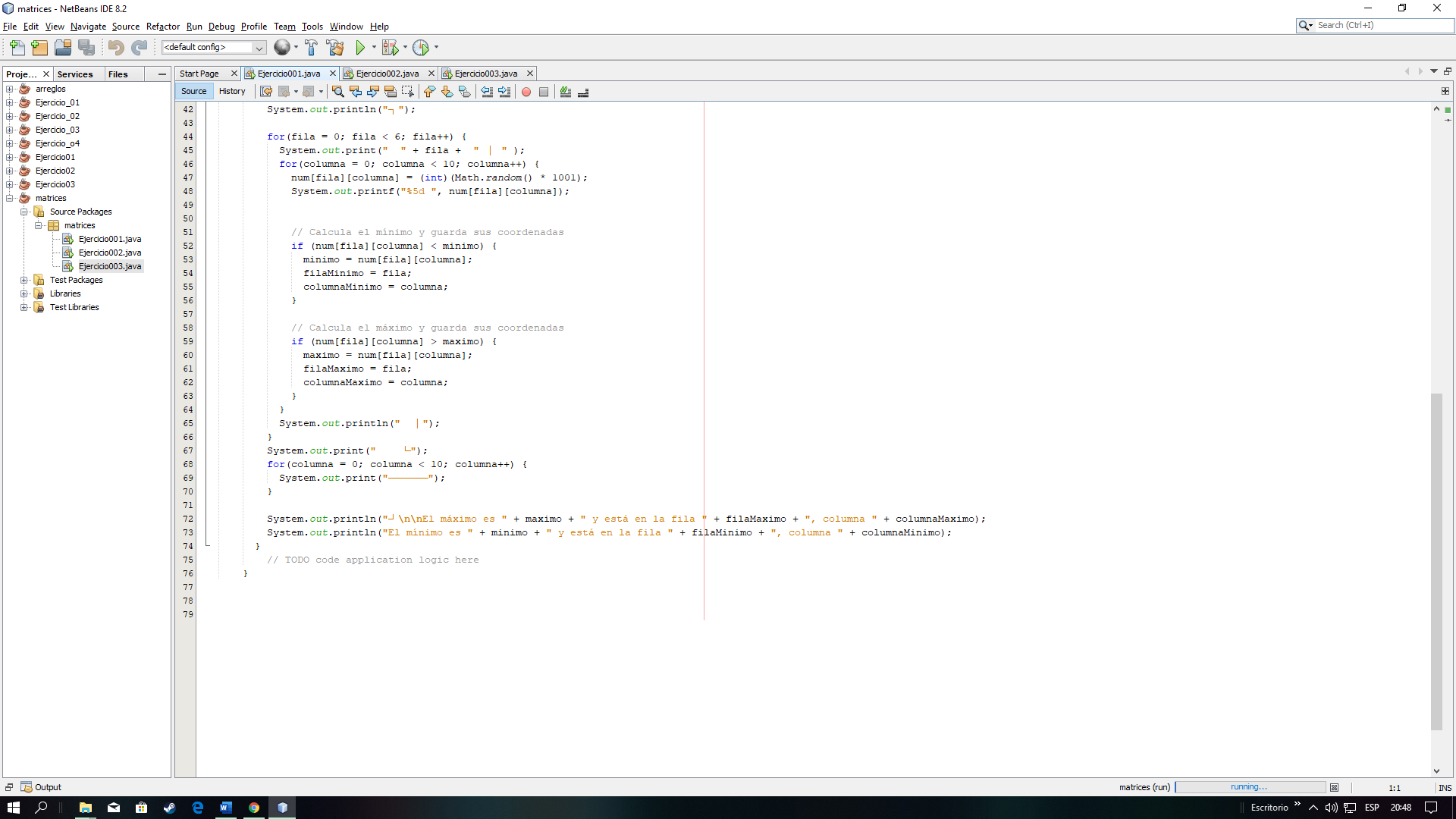
1. Realiza un programa que rellene un array de 6 filas por 10 columnas con

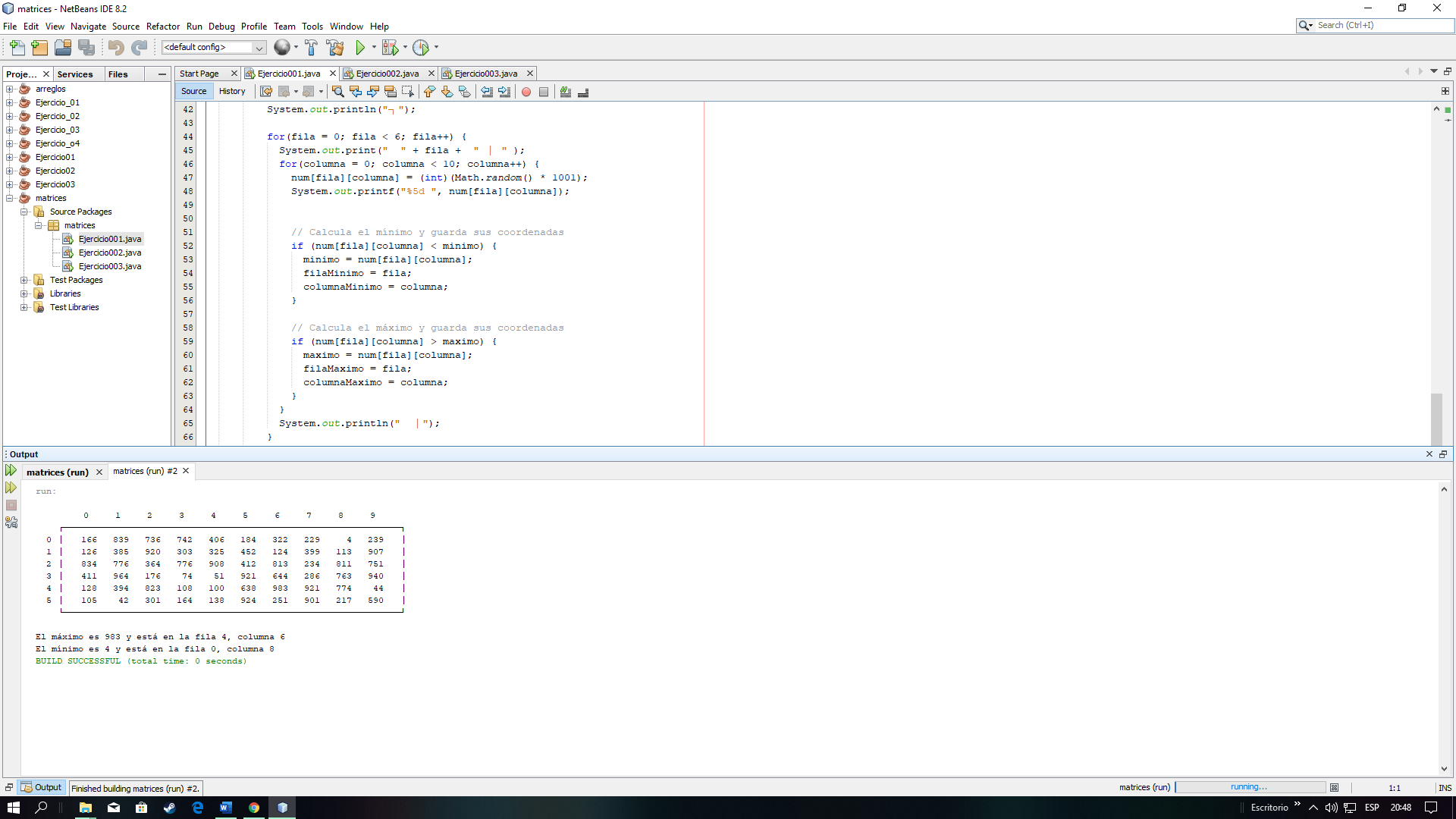
números enteros positivos comprendidos entre 0 y 1000 (ambos incluidos). A

continuación, el programa deberá dar la posición tanto del máximo como del

mínimo.





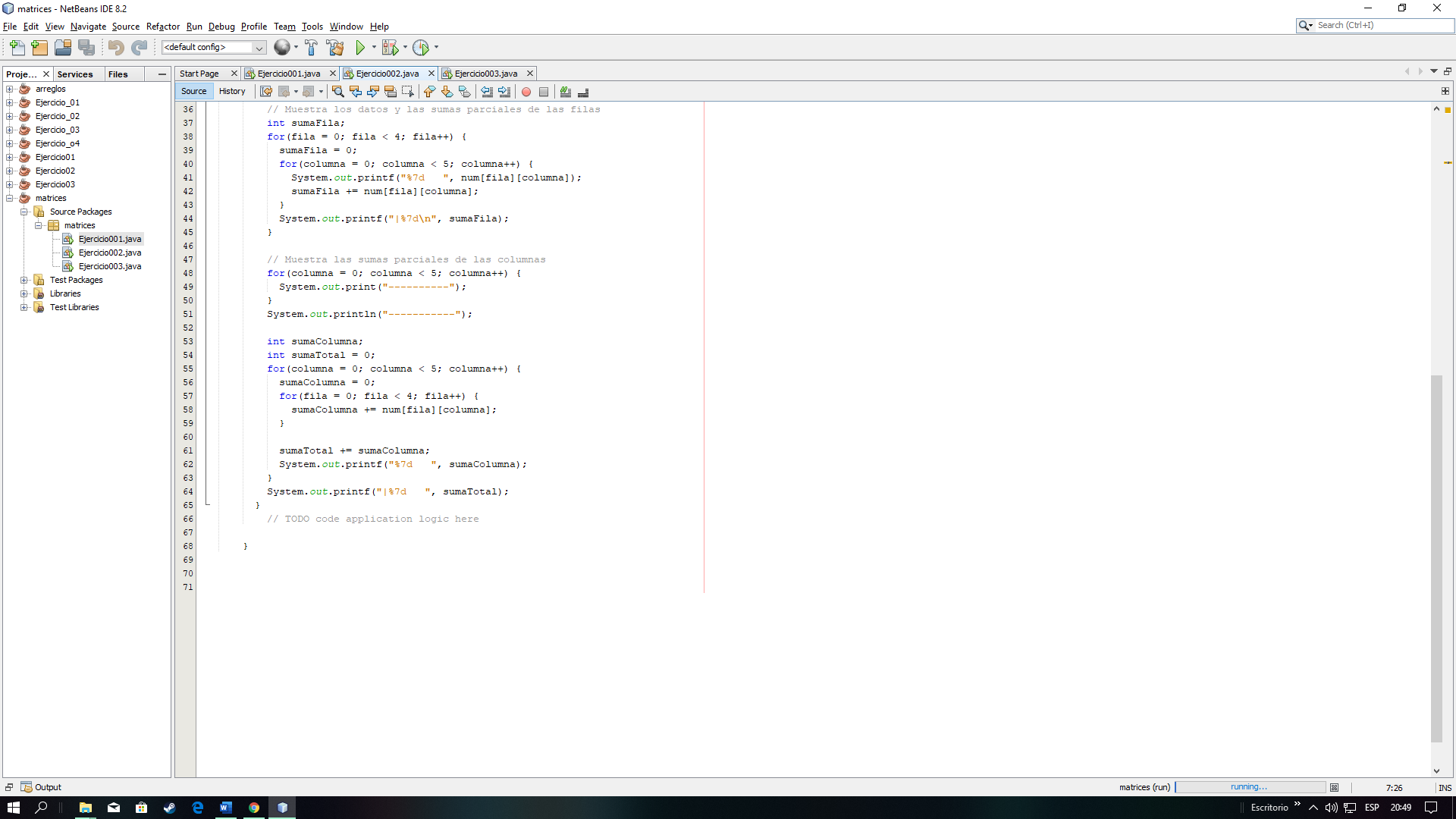
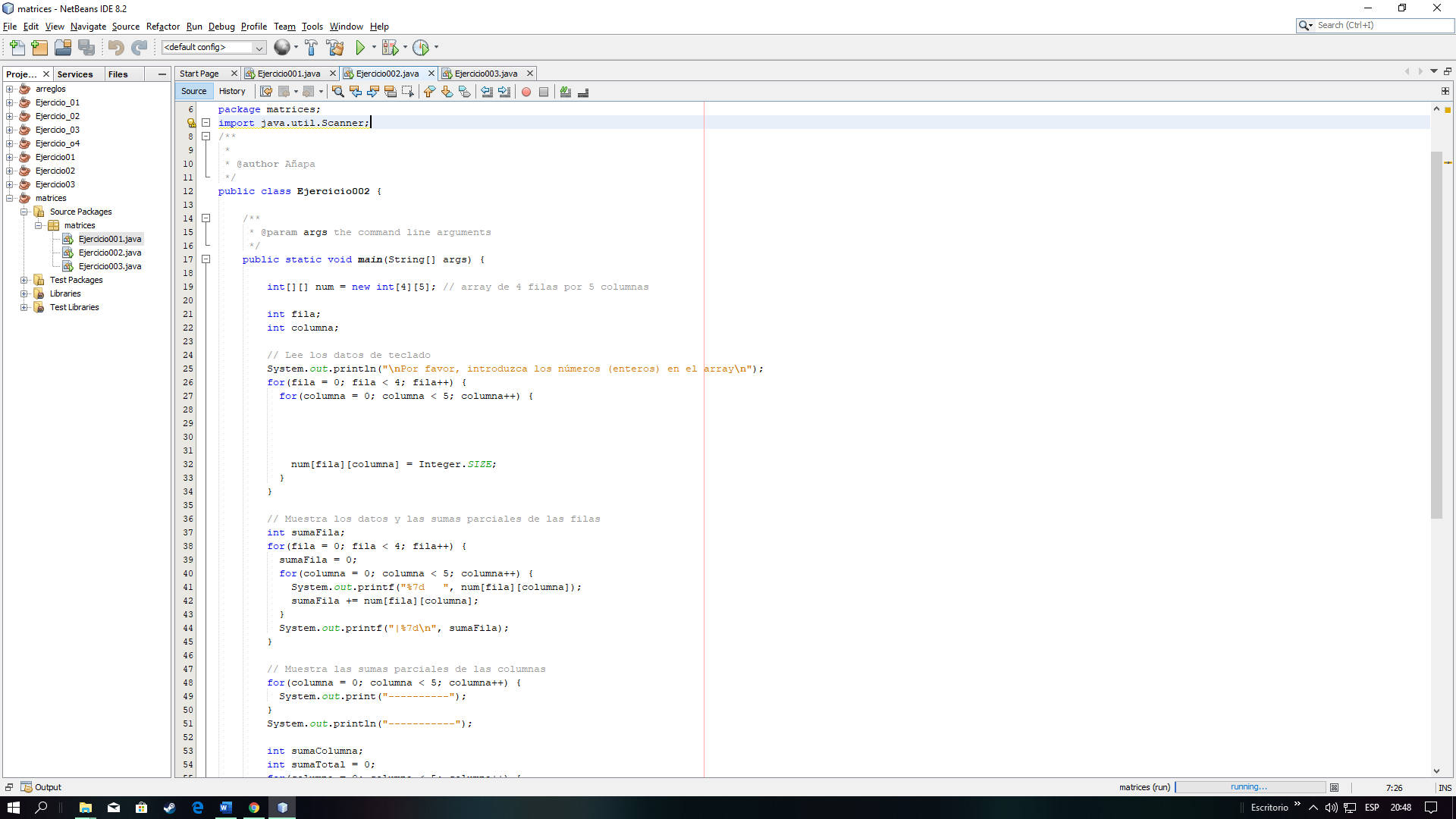
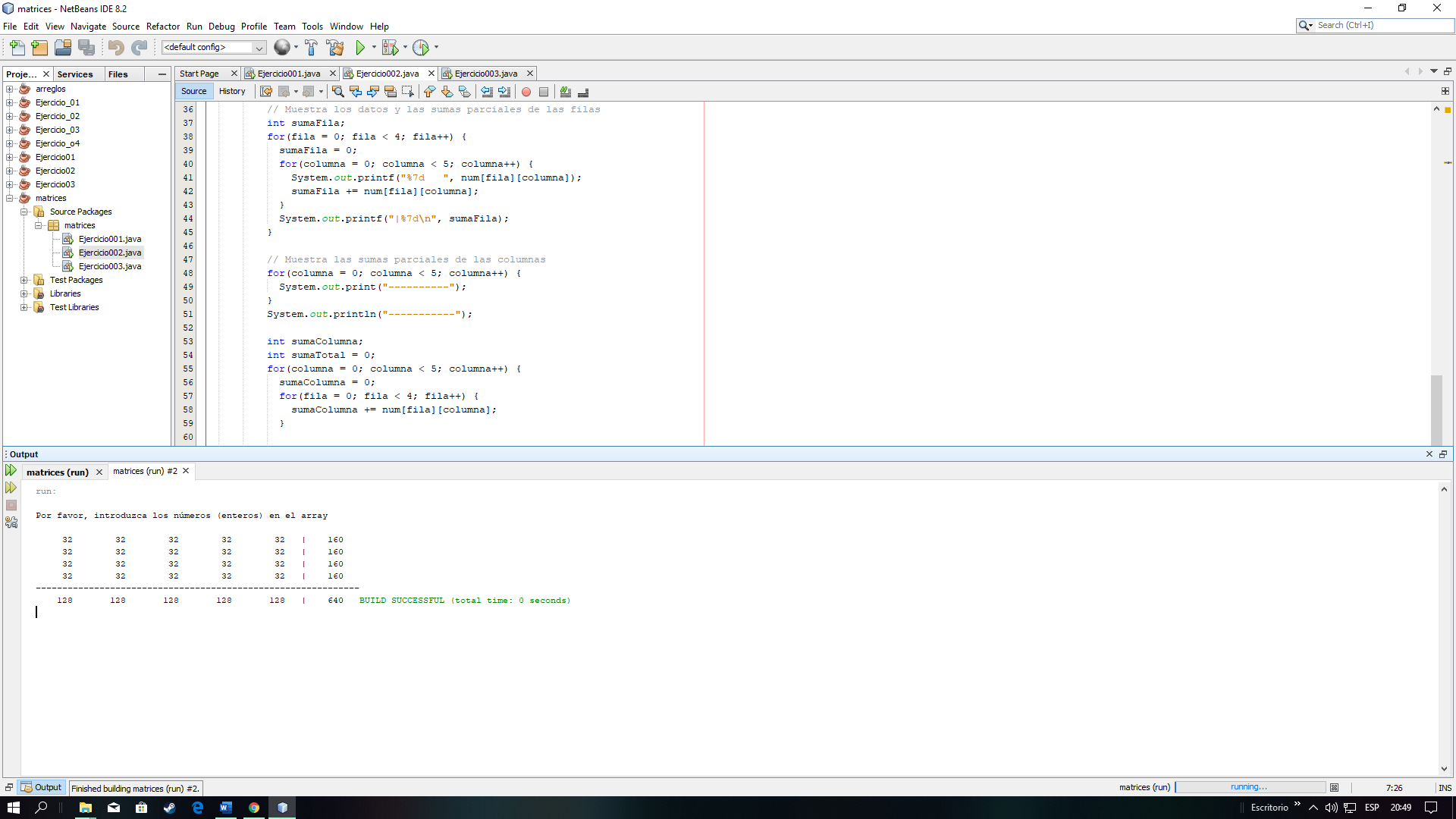


1. Escribe un programa que pida 20 números enteros. Estos números se deben

introducir en un array de 4 filas por 5 columnas. El programa mostrará las

sumas parciales de filas y columnas igual que si de una hoja de cálculo se

tratara. La suma total debe aparecer en la esquina inferior derecha.

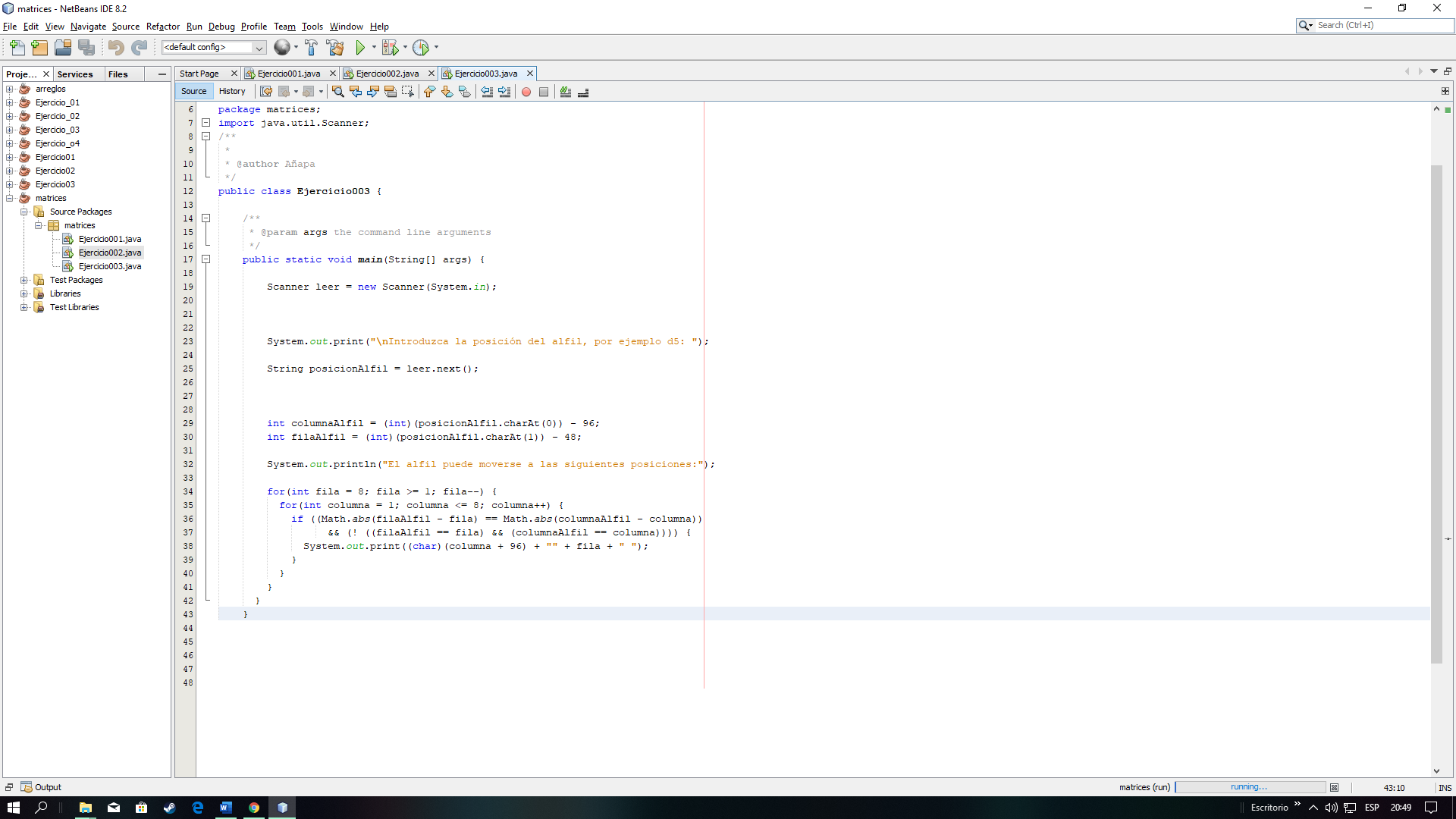
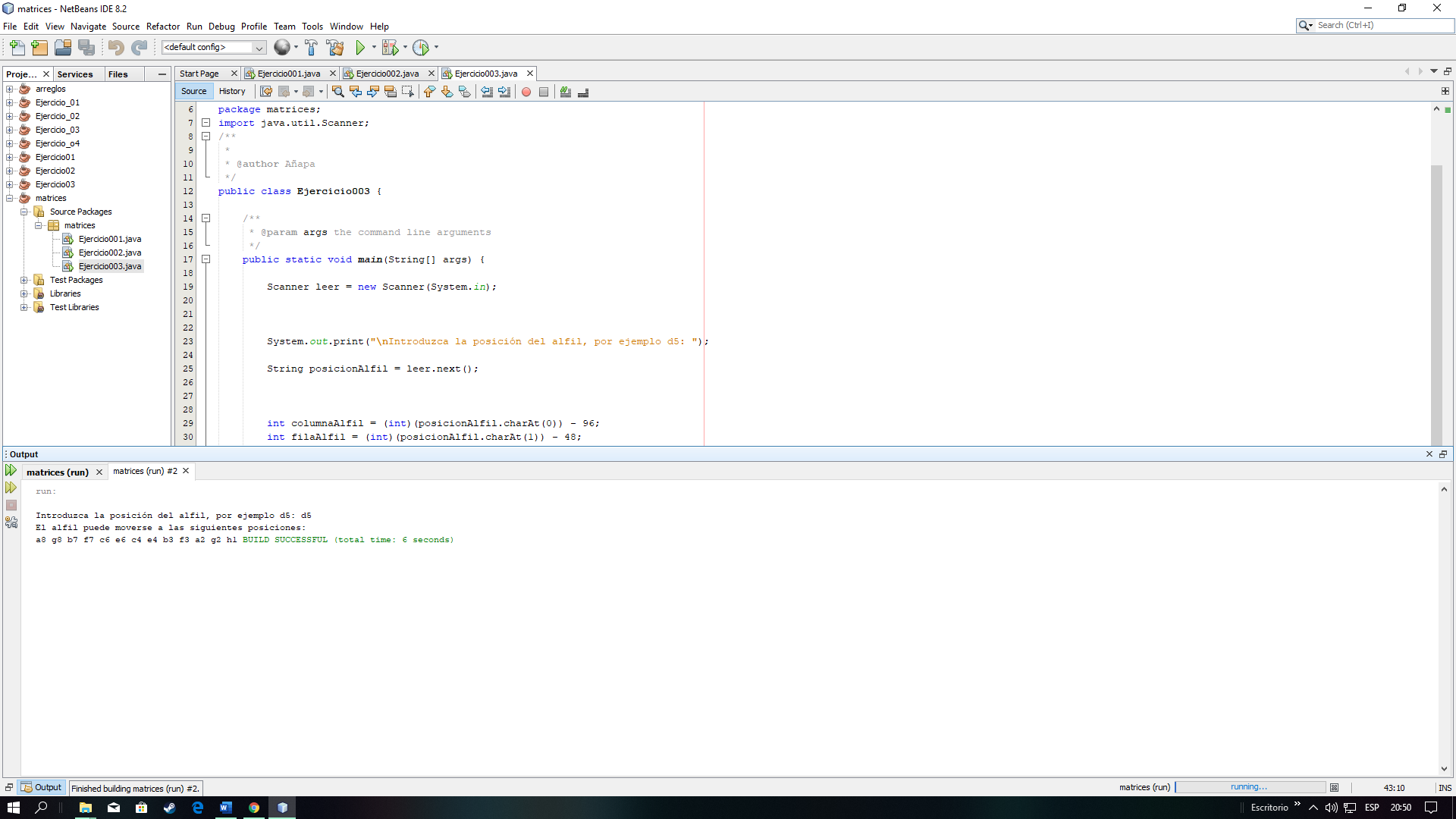
1. Escribe un programa que, dada una posición en un tablero de ajedrez, nos diga

a qué casillas podría saltar un alfil que se encuentra en esa posición. Como se

indica en la figura, el alfil se mueve siempre en diagonal. El tablero cuenta con

64 casillas. Las columnas se indican con las letras de la “a” a la “h” y las filas se

indican del 1 al 8.

Finalmente, debe subir el proyecto y el informe (.pdf) al repositorio remoto (Classroom

de GitHub).

Url para activar el repositorio remoto: https://classroom.github.com/a/dtddnILf