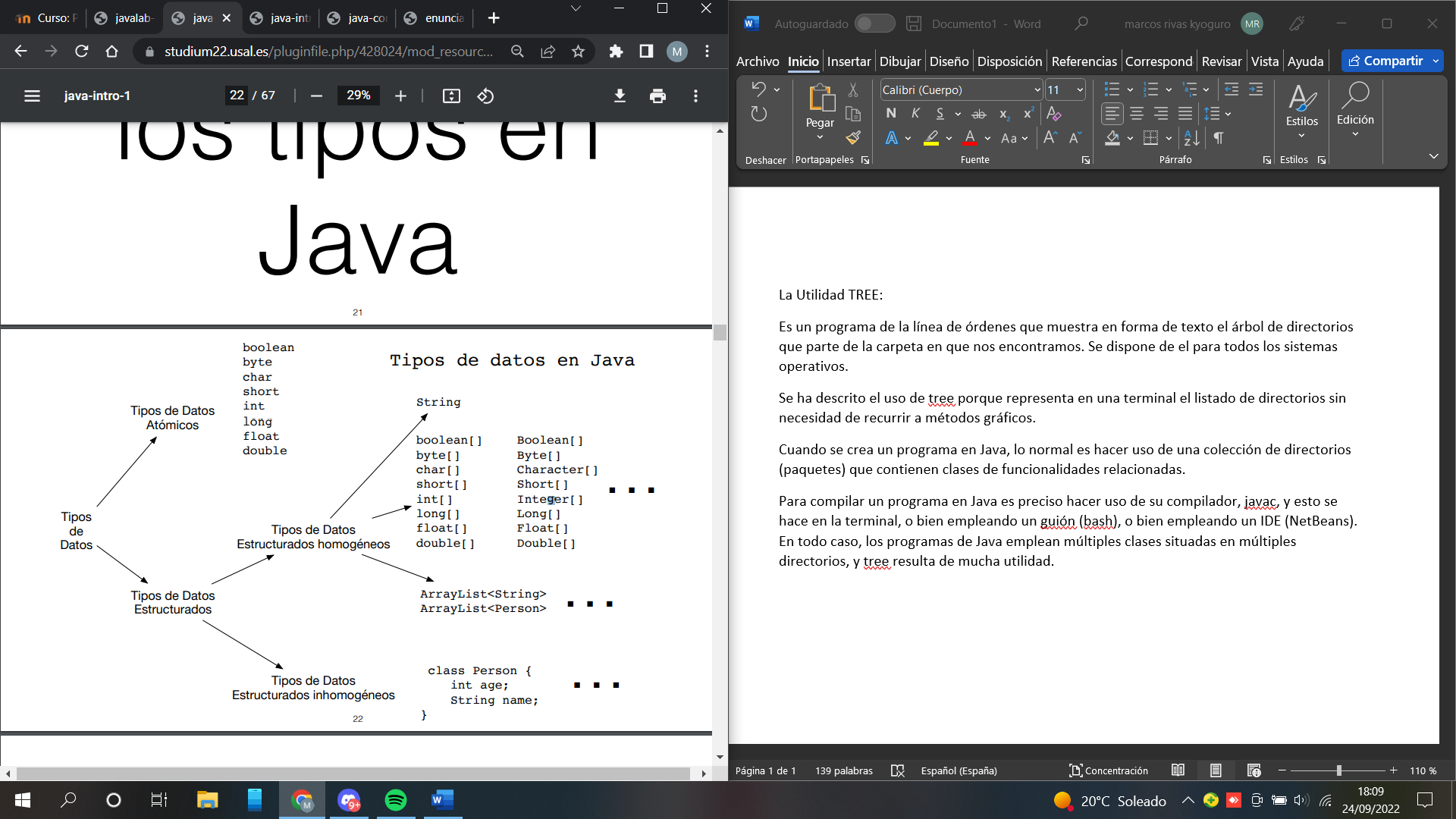
# LA UTILIDAD TREE:

Es un programa de la línea de órdenes que muestra en forma de texto el árbol de directorios que parte de la carpeta en que nos encontramos. Se dispone de él para todos los sistemas operativos.

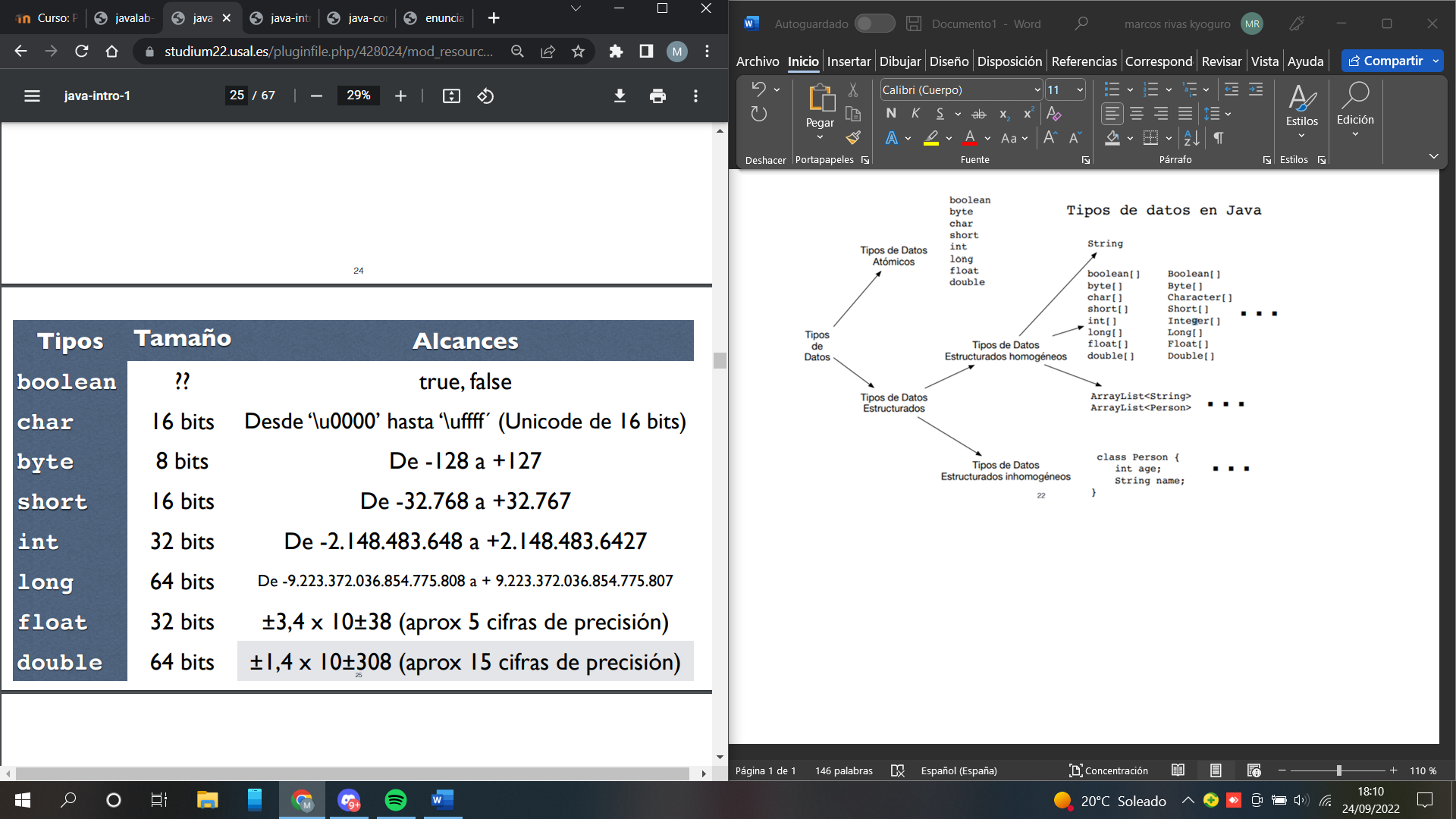
Se ha descrito el uso de tree porque representa en una terminal el listado de directorios sin necesidad de recurrir a métodos gráficos. Cuando se crea un programa en Java, lo normal es hacer uso de una colección de directorios (paquetes) que contienen clases de funcionalidades relacionadas.

Para compilar un programa en Java es preciso hacer uso de su compilador, javac, y esto se hace en la terminal, o bien empleando un guión (bash), o bien empleando un IDE (NetBeans). En todo caso, los programas de Java emplean múltiples clases situadas en múltiples directorios, y tree resulta de mucha utilidad.

# VISTA GENERAL DE LOS TIPOS DE JAVA



# CUADRO DE TIPOS ATÓMICOS

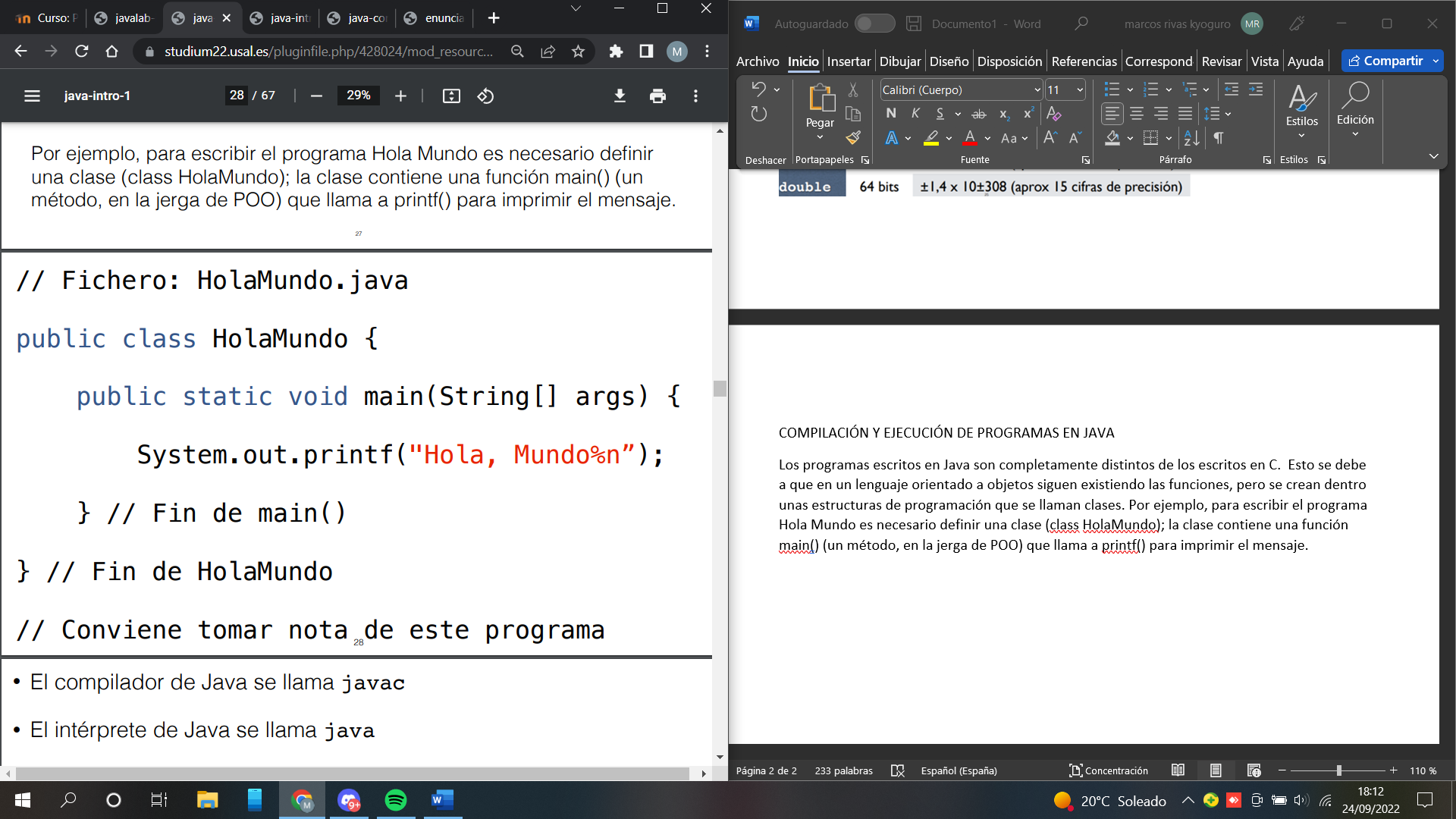


# COMPILACIÓN Y EJECUCIÓN DE PROGRAMAS EN JAVA

Los programas escritos en Java son completamente distintos de los escritos en C. Esto se debe a que, al ser un lenguaje orientado a objetos, siguen existiendo las funciones, pero se crean dentro de unas estructuras de programación que se llaman clases. Por ejemplo, para escribir el programa Hola Mundo es necesario definir una clase (class HolaMundo); la clase contiene una función main() (un método, en la jerga de POO) que llama a printf() para imprimir el mensaje.

Para ejecutar un programa escrito en Java hay que utilizar su intérprete (java, sin final). Se escribe en la terminal java Programa, y la máquina virtual de Java carga el programa, lo interpreta y lo ejecuta. Para compilar un programa escrito en Java hay que utilizar su compilador (javac). Se escribe en la terminal javac Programa.java y entonces el compilador javac compila el programa a bytecode, generando un archivo llamado Programa.class

# HOLA MUNDO



* El compilador de Java se llama javac
* El intérprete de Java se llama java
* Para compilar un programa se escribe: javac Programa.java
* Para ejecutar un programa se escribe: java Programa
* Si el programa reside en un archivo .jar, para ejecutarlo se escribe: java -jar program.jar2

Para ver el programa “Hola, Mundo!” se crea un archivo llamado HolaMundo.java, empleando alguna de las herramientas de texto instaladas (gEdit (Linux), BBEdit (mac OS) o TextPad (Windows), nano).

Si ahora examinamos el contenido de la carpeta, ha aparecido un nuevo archivo llamado HolaMundo.class. Este es el programa interpretable, no es un programa ejecutable, Java es un intérprete y esto no es código nativo. Por eso no se llama HolaMundo.exe/.com/.app.

# E/S EMPLEANDO LA CLASE SYSTEM

La clase System, a través de sus atributos **err, out** e **in** (que son respectivamente dos PrintStream y un InputStream ofrece directamente un mecanismo de uso muy sencillo para leer y escribir en Java.

En Java hay dos formas predefinidas de lectura del teclado:

1. La entrada puede efectuarse a través de un ejemplar de Scanner, que se construye a partir de System.in
2. También a partir de un ejemplar de Console proporcionado por System.

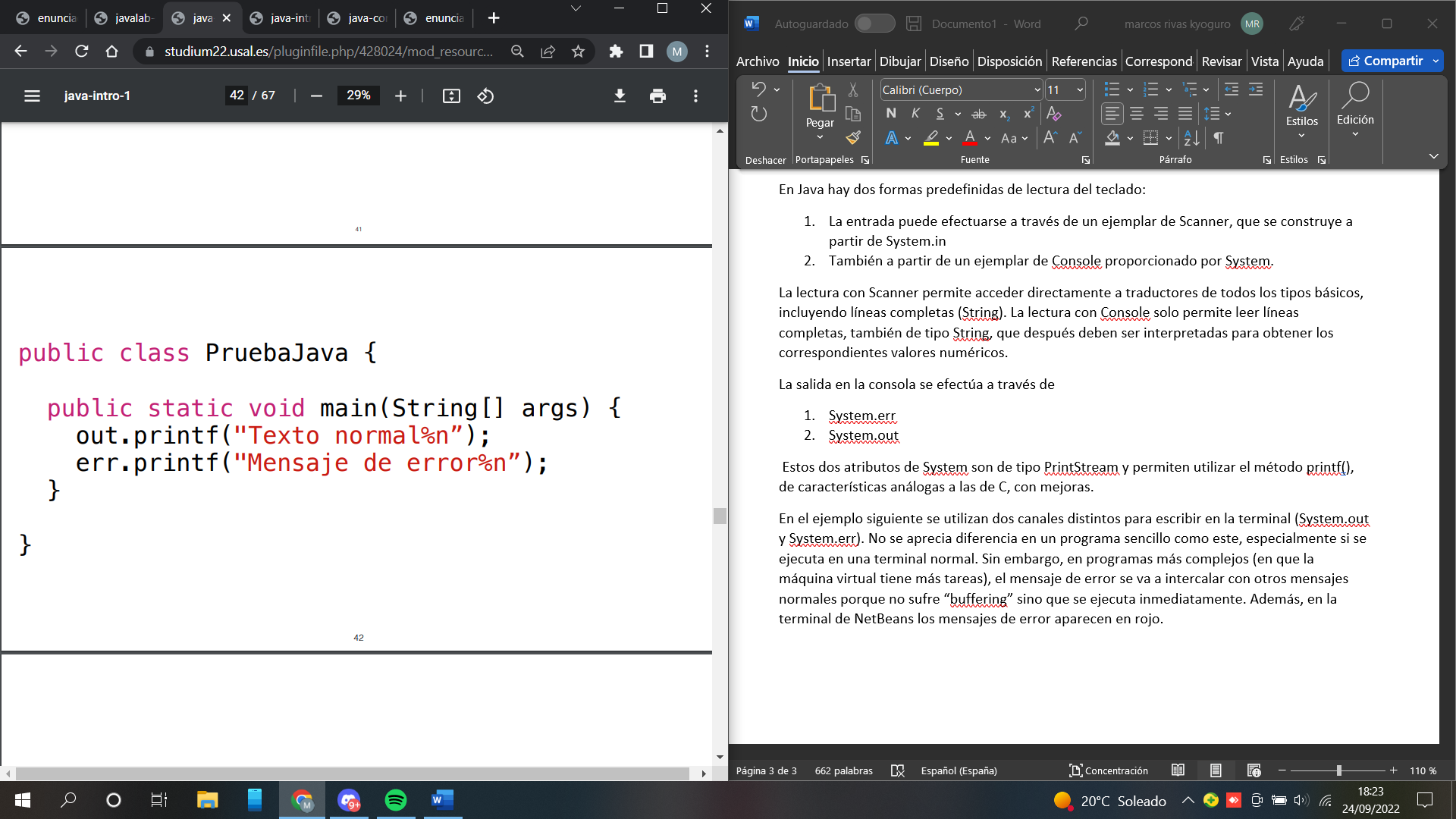
La lectura con Scanner permite acceder directamente a traductores de todos los tipos básicos, incluyendo líneas completas (String). La lectura con Console solo permite leer líneas completas, también de tipo String, que después deben ser interpretadas para obtener los correspondientes valores numéricos.

La salida en la consola se efectúa a través de

1. System.err
2. System.out

Estos dos atributos de System son de tipo PrintStream y permiten utilizar el método printf(), de características análogas a las de C, con mejoras.

En el ejemplo siguiente se utilizan dos canales distintos para escribir en la terminal (System.out y System.err). No se aprecia diferencia en un programa sencillo como este, especialmente si se ejecuta en una terminal normal. Sin embargo, en programas más complejos (en que la máquina virtual tiene más tareas), el mensaje de error se va a intercalar con otros mensajes normales porque no sufre “buffering” sino que se ejecuta inmediatamente. Además, en la terminal de NetBeans los mensajes de error aparecen en rojo.



El proceso de instalación de NetBeans, relativamente largo, no es necesario para compilar y ejecutar programas en Java. Realmente, basta usar un editor de texto como BBEdit (macOS) , Nano Linux/macOS/Windows), TextPad (Windows) para escribir el programa. De hecho, se puede obviar la instrucción package, que no tiene demasiado sentido puesto que se utiliza una sola clase. El programa podría quedar en la forma siguiente, en el archivo PruebaJava.java.

Una vez escrito este diminuto programa, se compila mediante instrucciones similares a las que se muestran a continuación. Se supone que una vez arrancada la terminal se ha hecho un cd a la carpeta que contiene PruebaJava.java

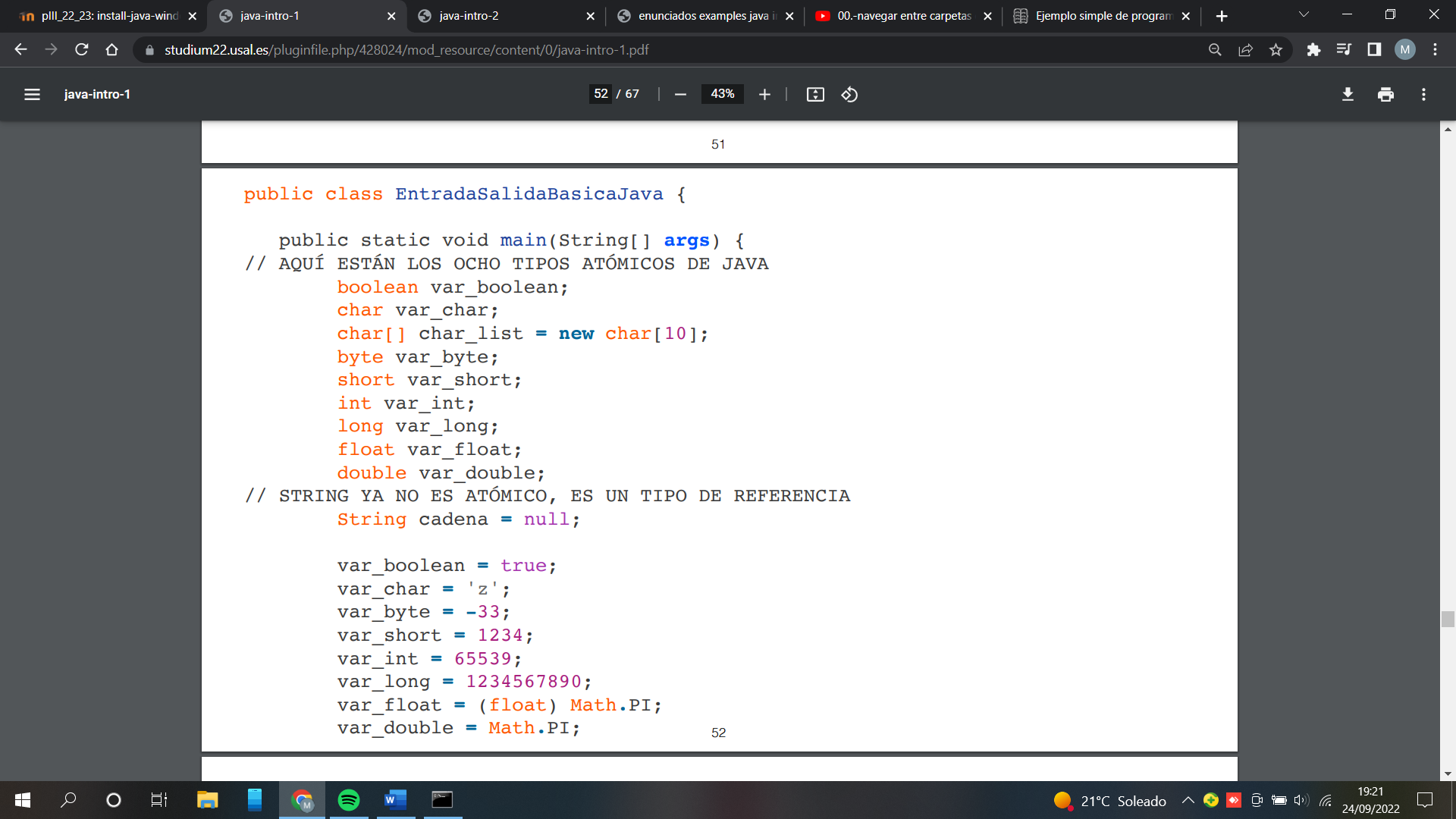
La terminal muestra los dos mensajes en el mismo color, la “terminal” de NetBeans es la que se encarga de colorear los resultados. Por supuesto, se puede conseguir el efecto de color en Bash anteponiendo códigos de color. (Esdia).

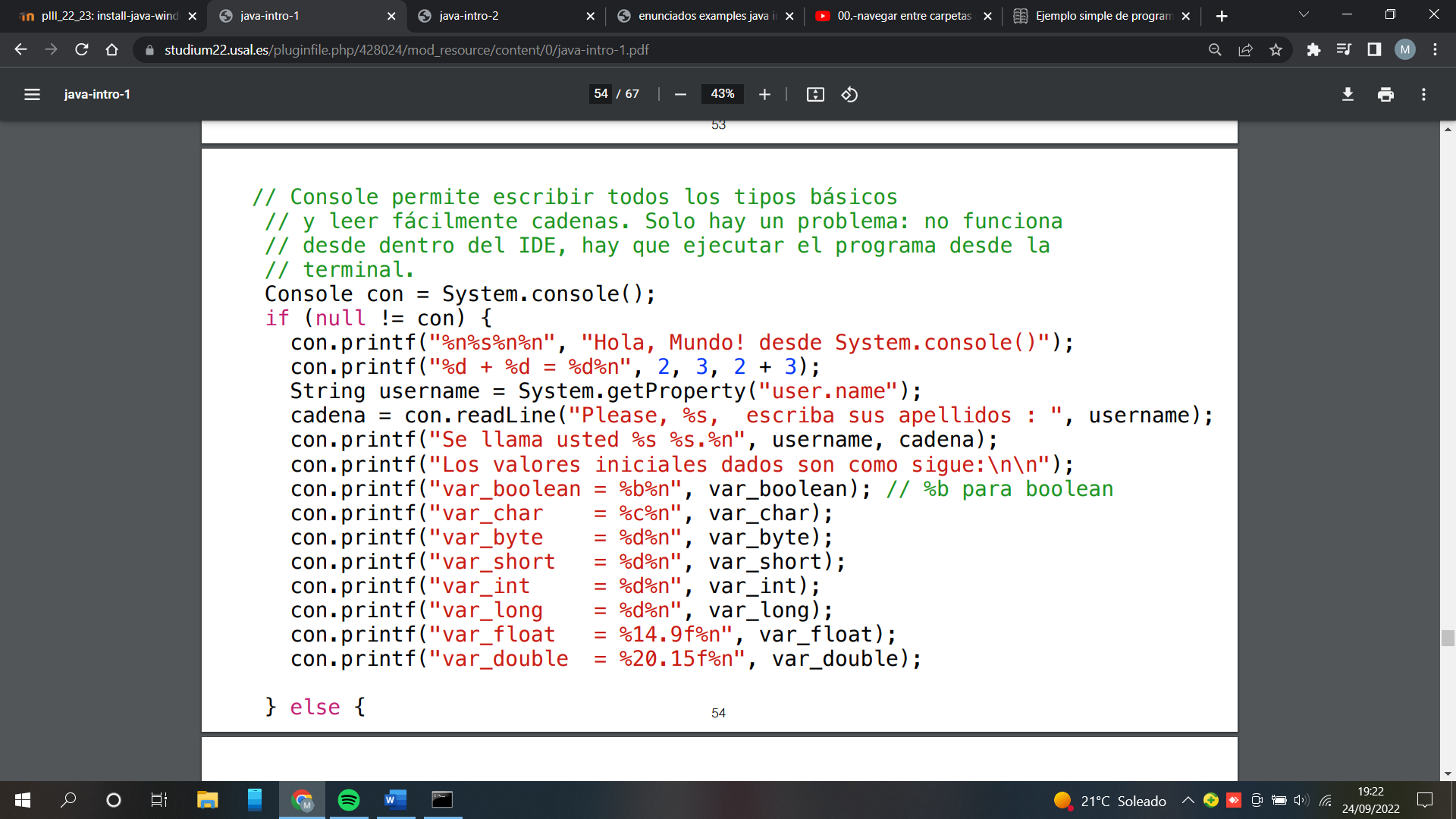
# EntradaSalidaBasicaJava

En este programa de declaran variables de distintos tipos y se muestran sus valores en la pantalla, empleando métodos de las clases Console y System. Se observa que Console es más limitado en lectura, porque su objetivo es ante todo la robustez. System por su parte, es más flexible, pero puede producir errores con relativa facilidad. Hay que capturarlos mediante try/catch.

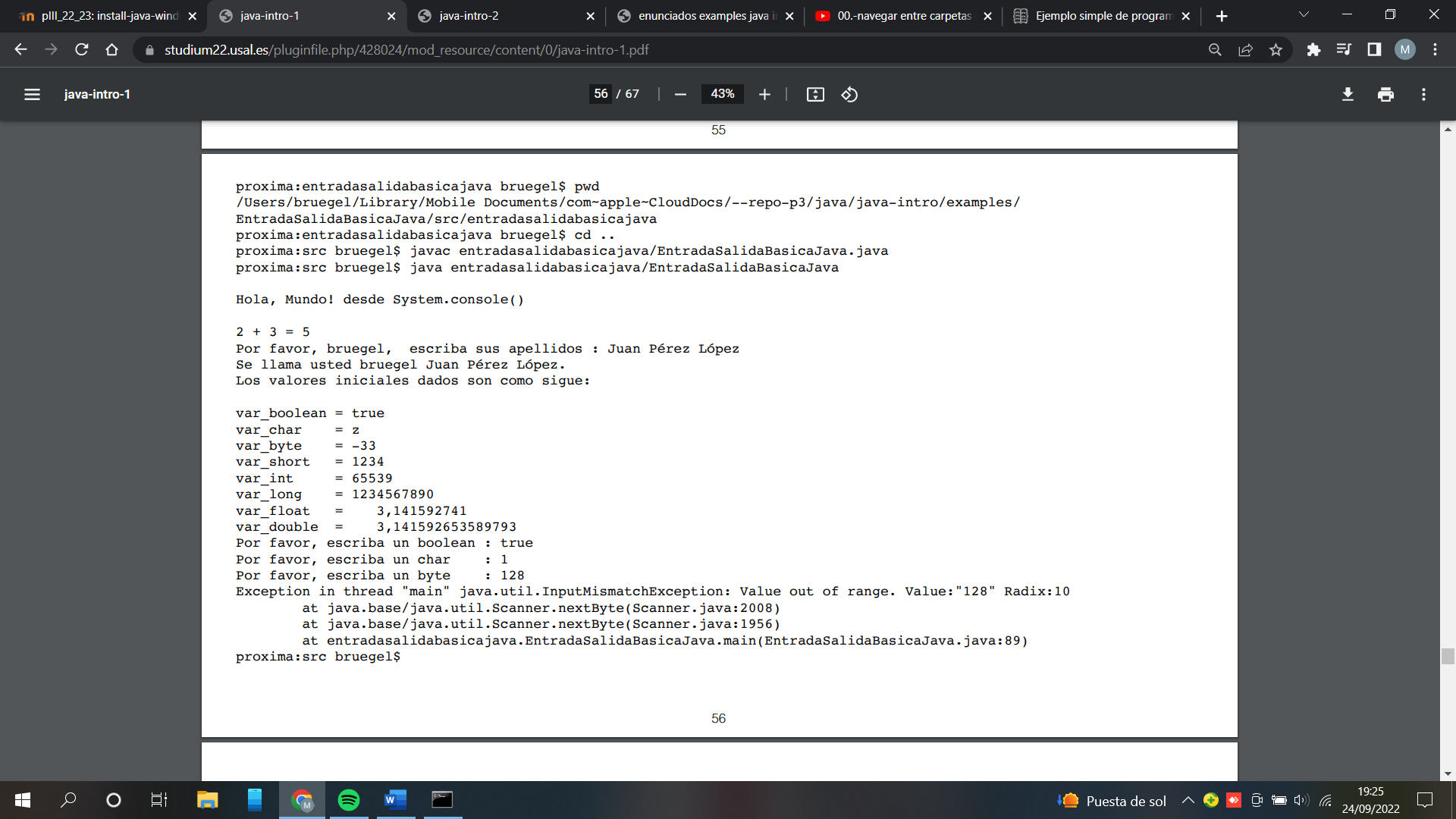
Console permite escribir todos los tipos básicos y leer fácilmente cadenas. Solo hay un problema: no funciona desde dentro del IDE, hay que ejecutar el programa desde la terminal

* Comenzamos por declarar las variables



* Ahora se escriben los valores de las variables en la terminal, empleando Console si está disponible

El resultado de ejecutar este programa en una terminal (Tools —> Open in Terminal, javac, java) es como sigue.



Java tiene comprobación de alcances, tanto para el rango de validez como para los índices válidos dentro de un array o una colección. Así que, se produce un error al introducir el valor 128 en un byte. Los bytes tienen un alcance de [-128, 127] y se ha detectado correctamente el error.

OJO!! Console no está disponible en NetBeans porque solo simula una terminal.

* Limitación de Console: solo ofrece el método readLine(), que lee siempre cadenas completas y descarta el retorno de carro final. Quizá no es una limitación sino una ventaja, porque es más seguro
* Ventaja de Console ofrece el método readPassword(), muy útil para leer contraseñas a través de la consola.