4). Aquí se expondrán de manera corta los métodos que se consideran importantes de cada una de las clases si es necesario.

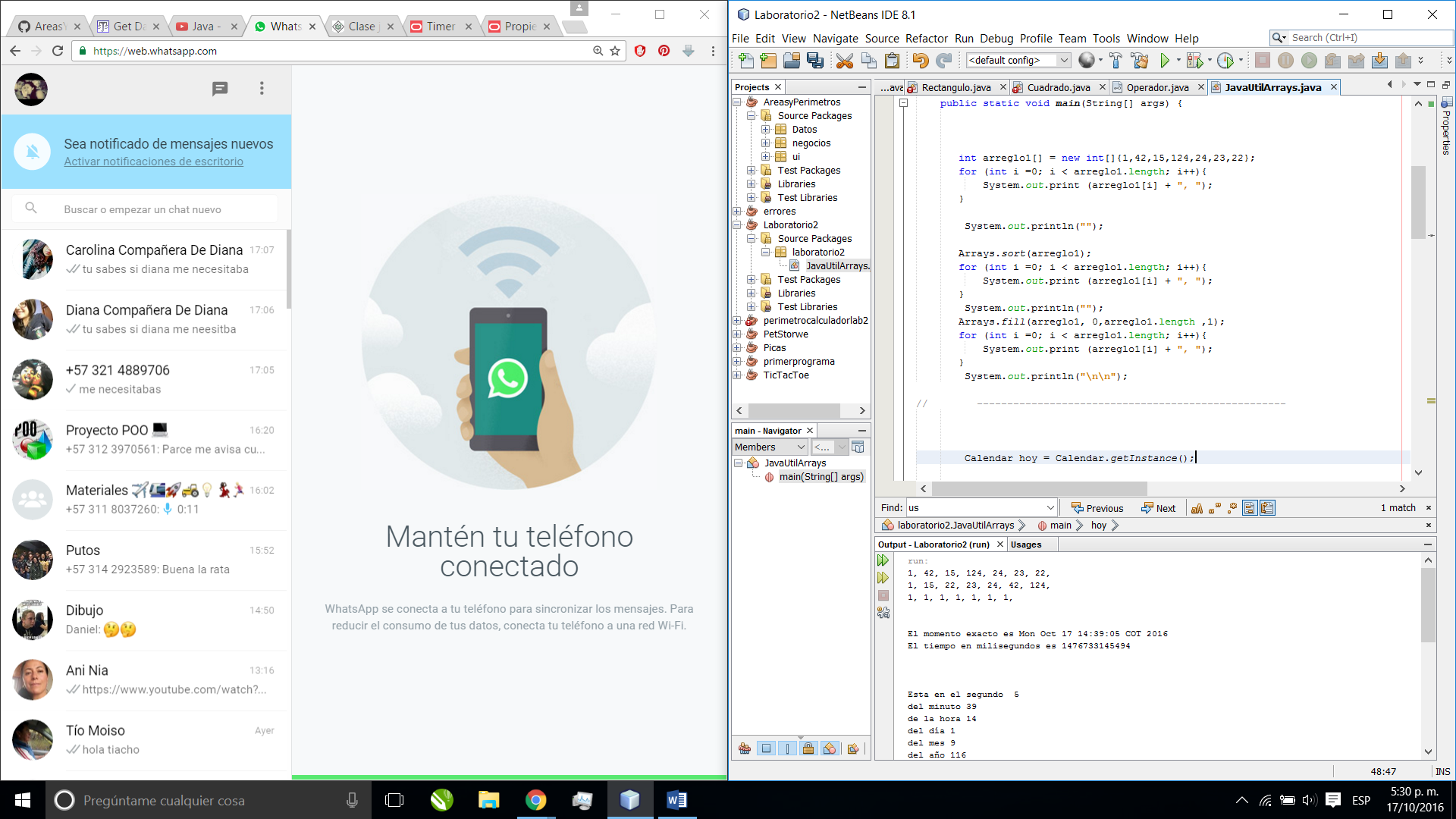
**java.util.Arrays;**

Es una clase que proporciona una serie de variados métodos que permiten manipular de manera adecuada los arreglos como, por ejemplo.

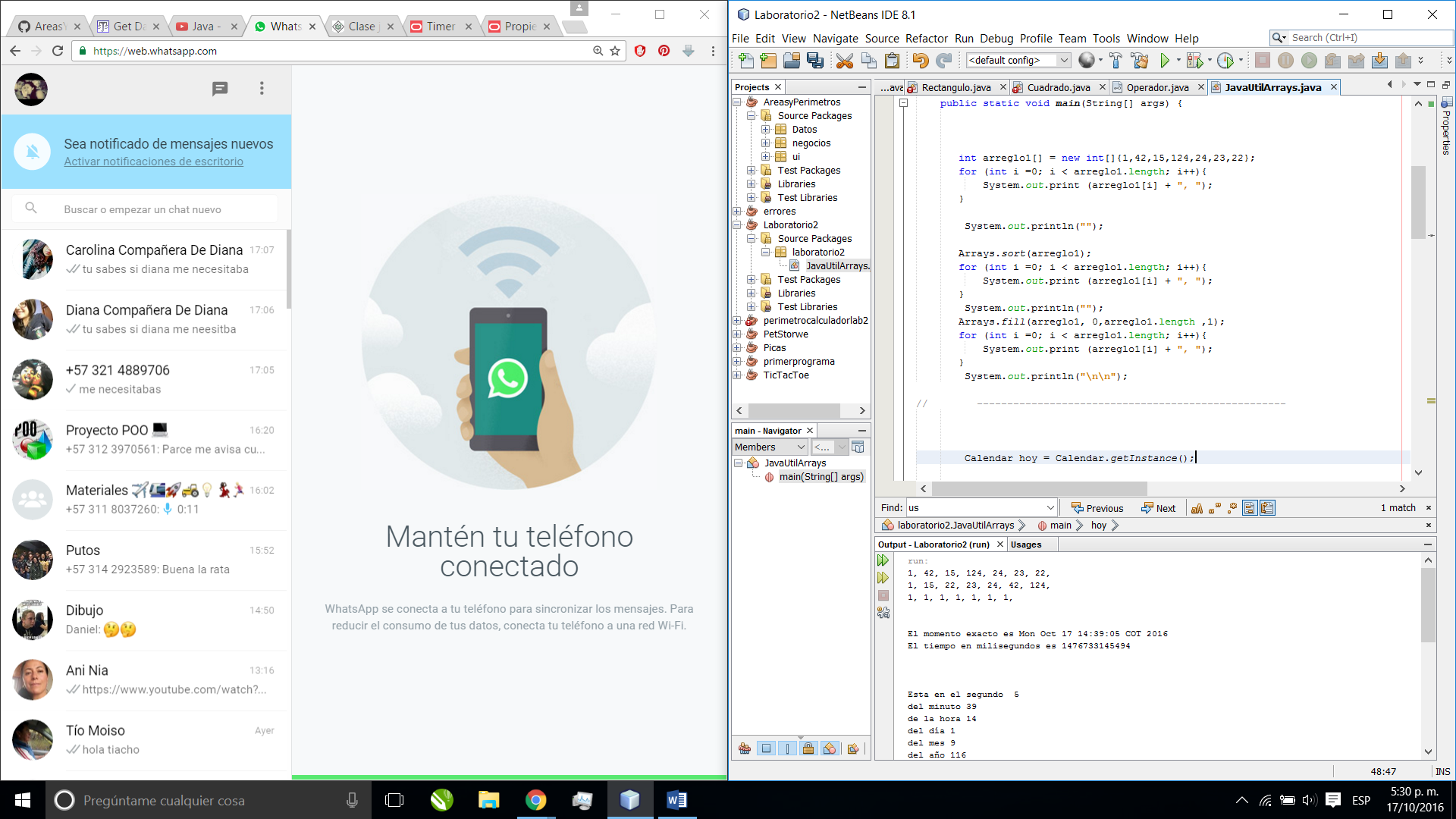
* sort: ordena los elementos del arreglo de menor a mayor
* hashCode: devuelve un hashCode dependiendo de los elementos del arreglo.
* fill: llena los elementos definidos del arreglo con un elemento predeterminado del mismo tipo.
* equals: permite comparar entre dos valores pertenecientes al arreglo si son iguales, retornando un booleano con el valor de esta igualdad.
* deepEquals: permite comparar si dos arreglos son profundamente iguales.
* copyOf: copea una matriz especificada truncando o rellenando con ceros si es necesario para que la matriz copia tenga una misma longitud.
* copyOfRange: copea un rango especifico de un arreglo en otro.
* binarySearch: permite la búsqueda en el arreglo por medio del algoritmo binario.
* asList: devuelve una lista del arreglo.

Ejemplo:

* Código



* Consola

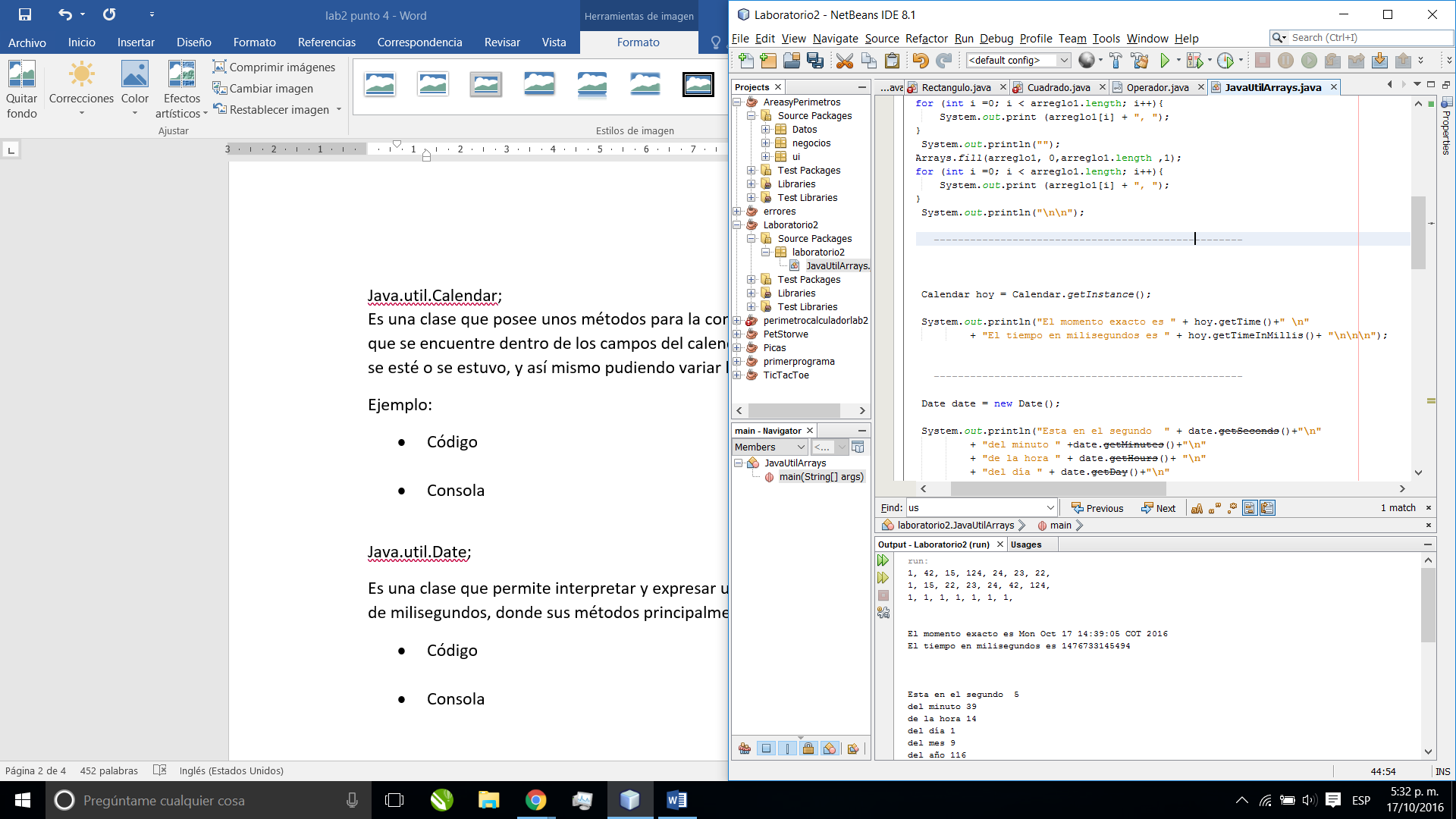


**java.util.Calendar;**

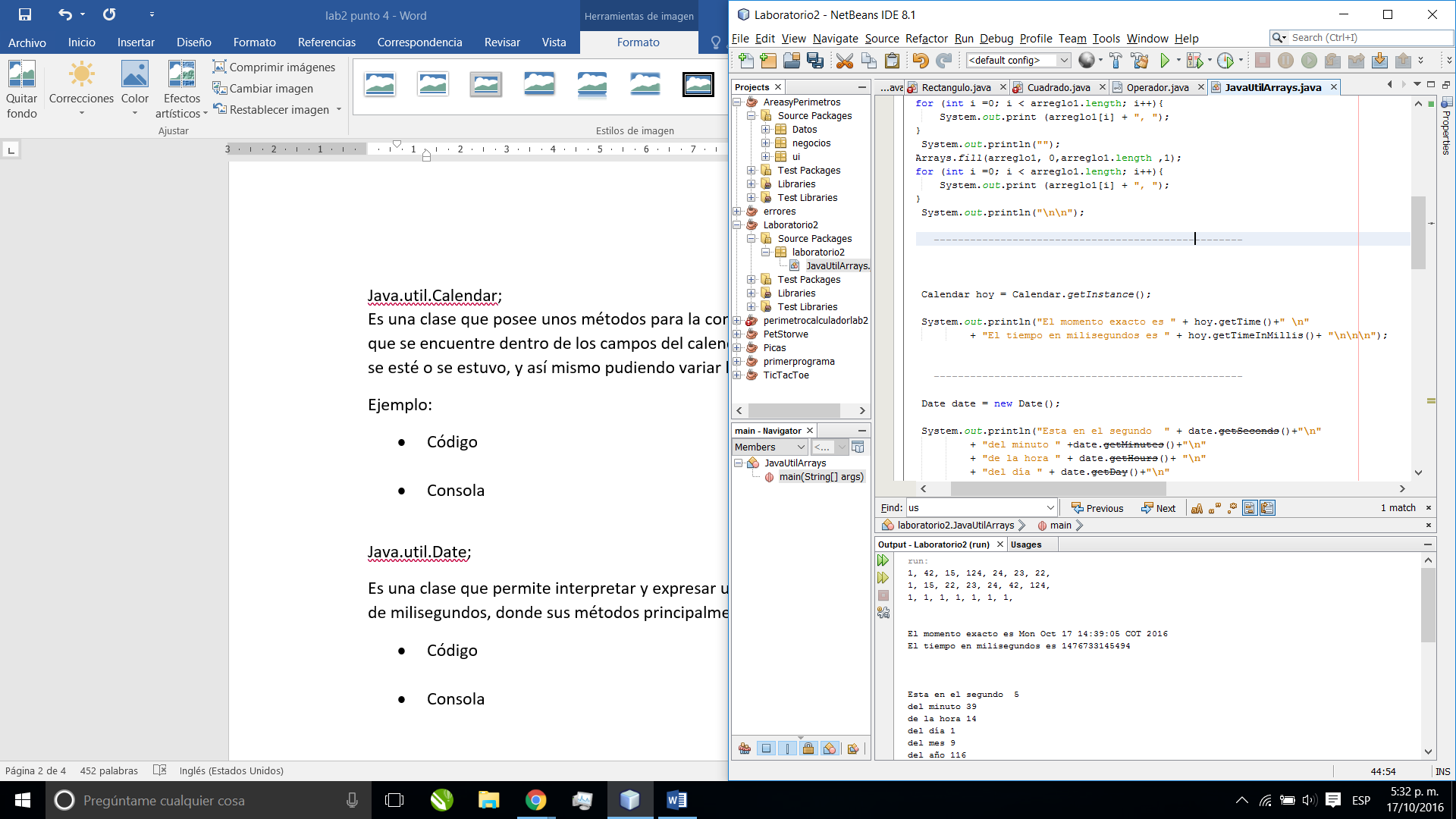
Es una clase que posee unos métodos para la conversión de un instante definido en el tiempo a un que se encuentre dentro de los campos del calendario, como por ejemplo obtener la del día en que se esté o se estuvo, y así mismo pudiendo variar los periodos de tiempo a analizar.

Ejemplo:

* Código



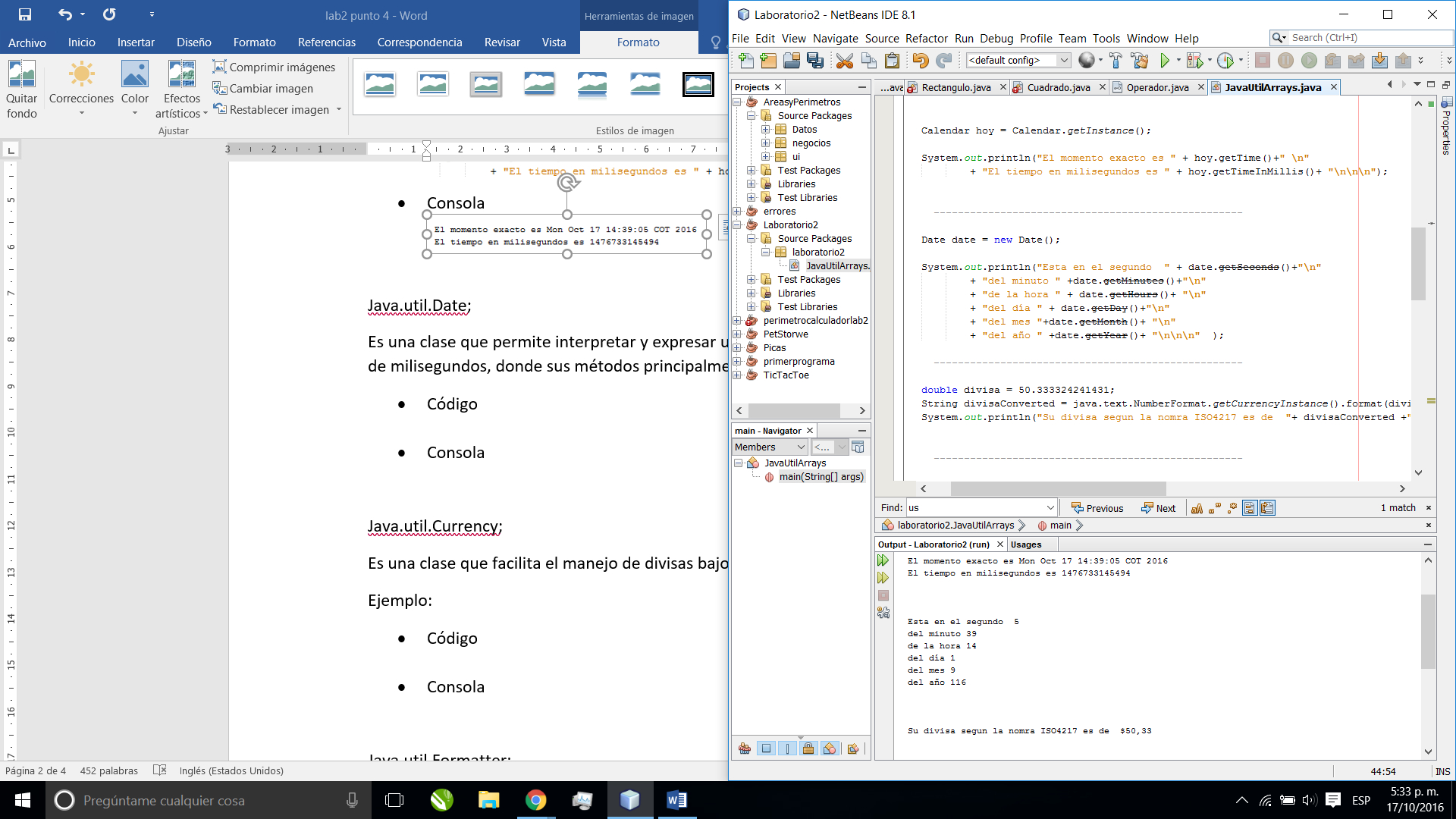
* Consola



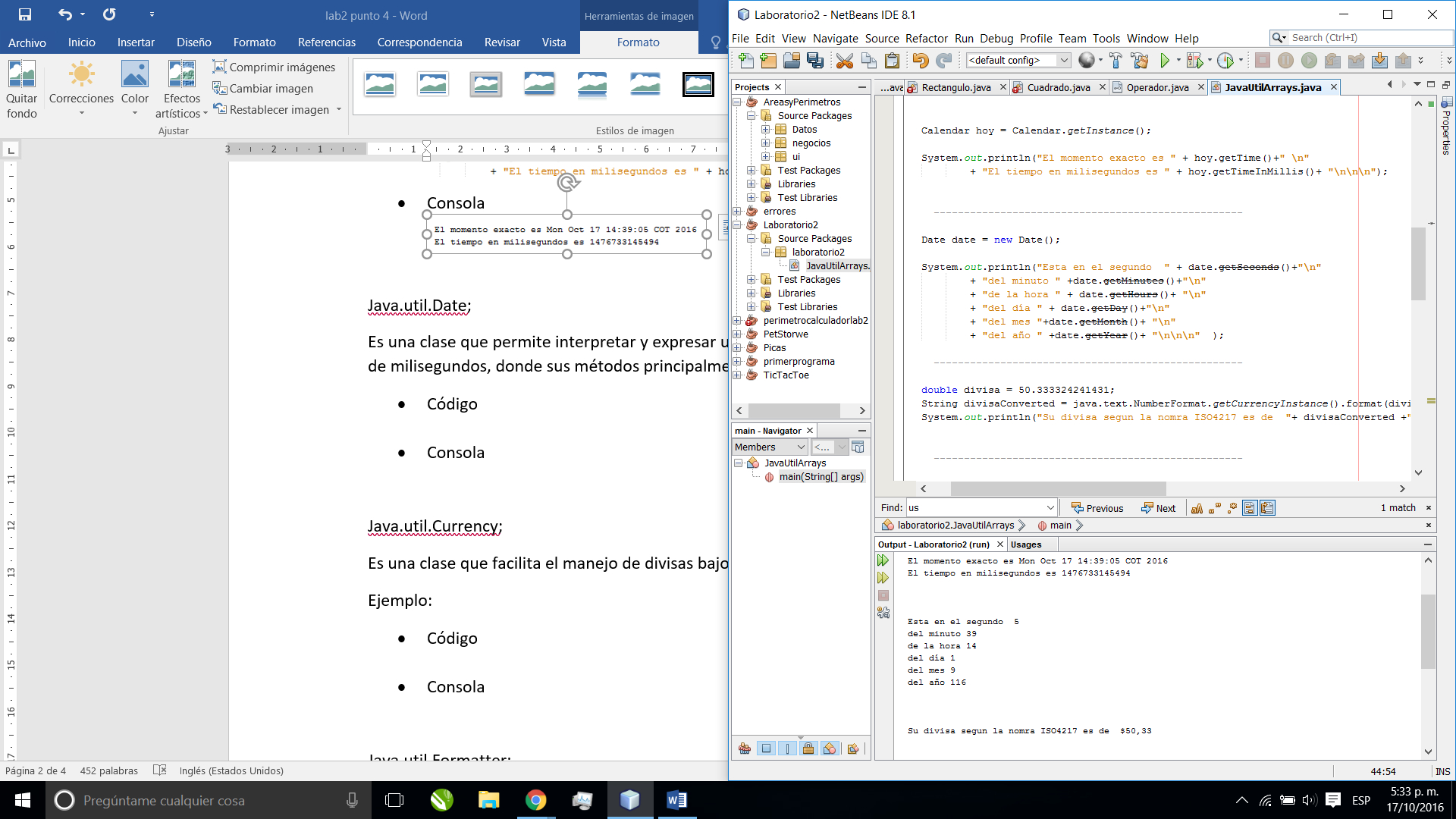
**java.util.Date;**

Es una clase que permite interpretar y expresar un instante de tiempo especifico con una precisión de milisegundos, donde sus métodos principalmente admiten accederlos y modificarlos. Ejemplo:

* Código



* Consola

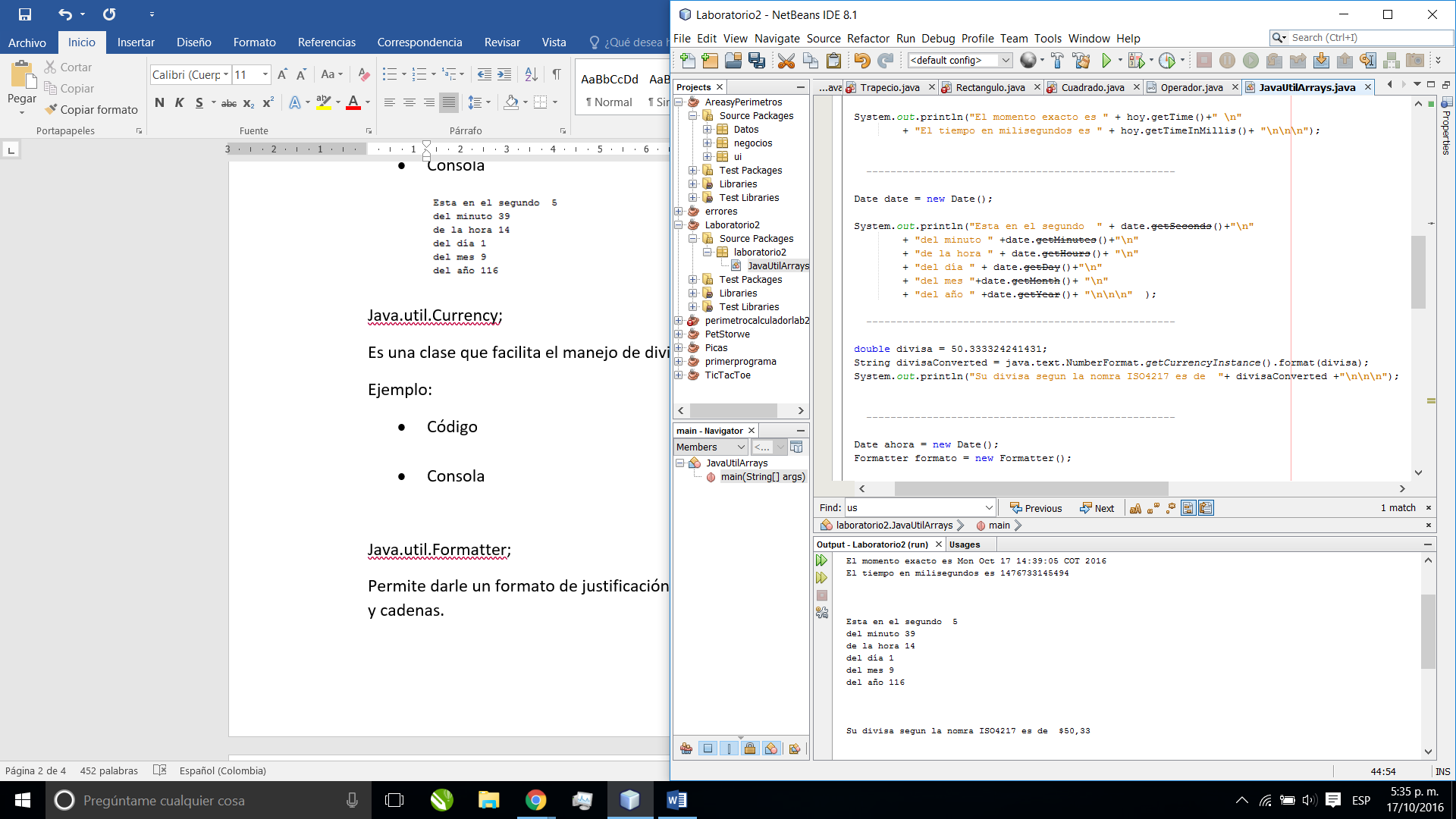


**java.util.Currency;**

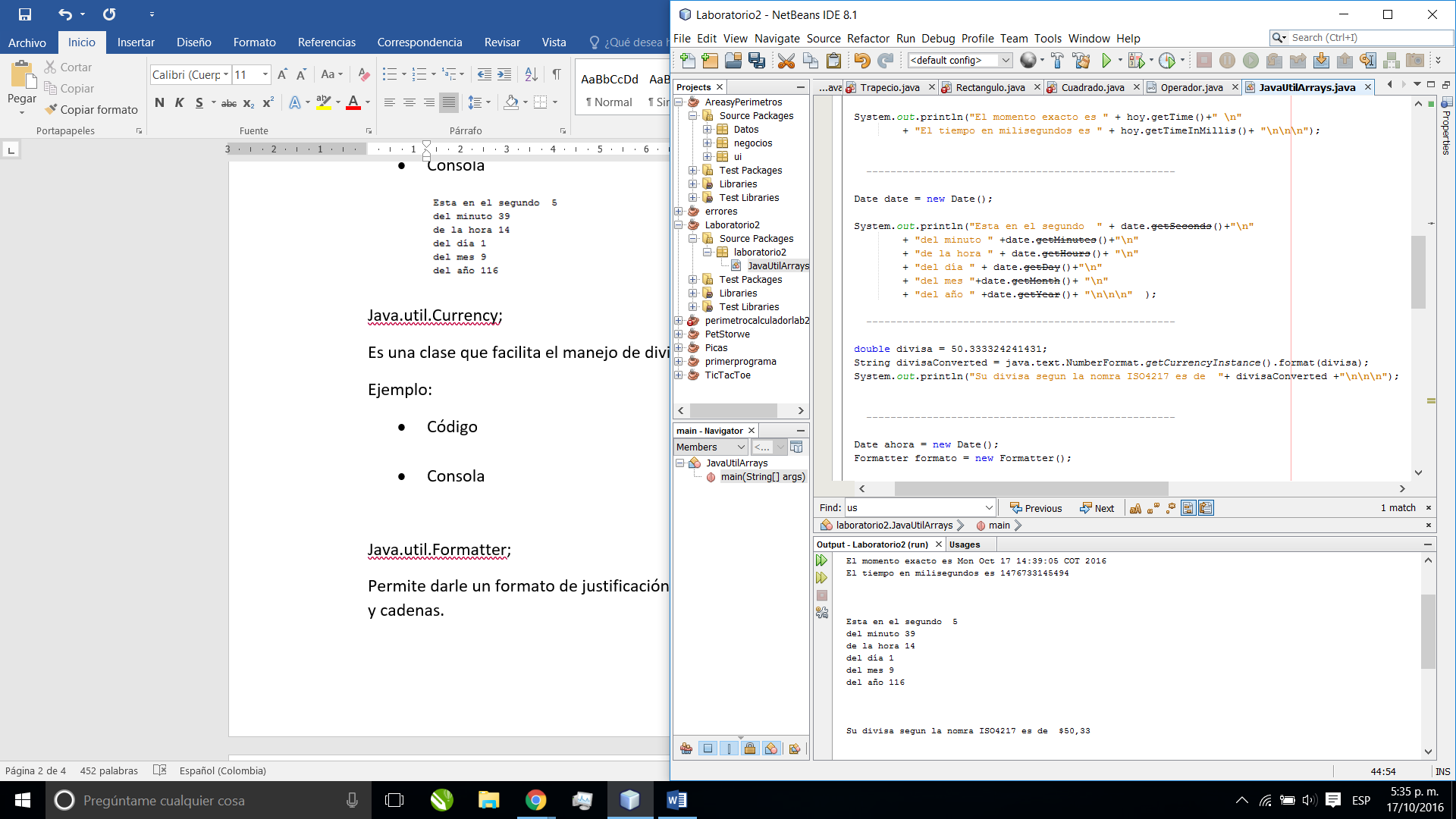
Es una clase que facilita el manejo de divisas bajo la norma ISO4217.

Ejemplo:

* Código



* Consola

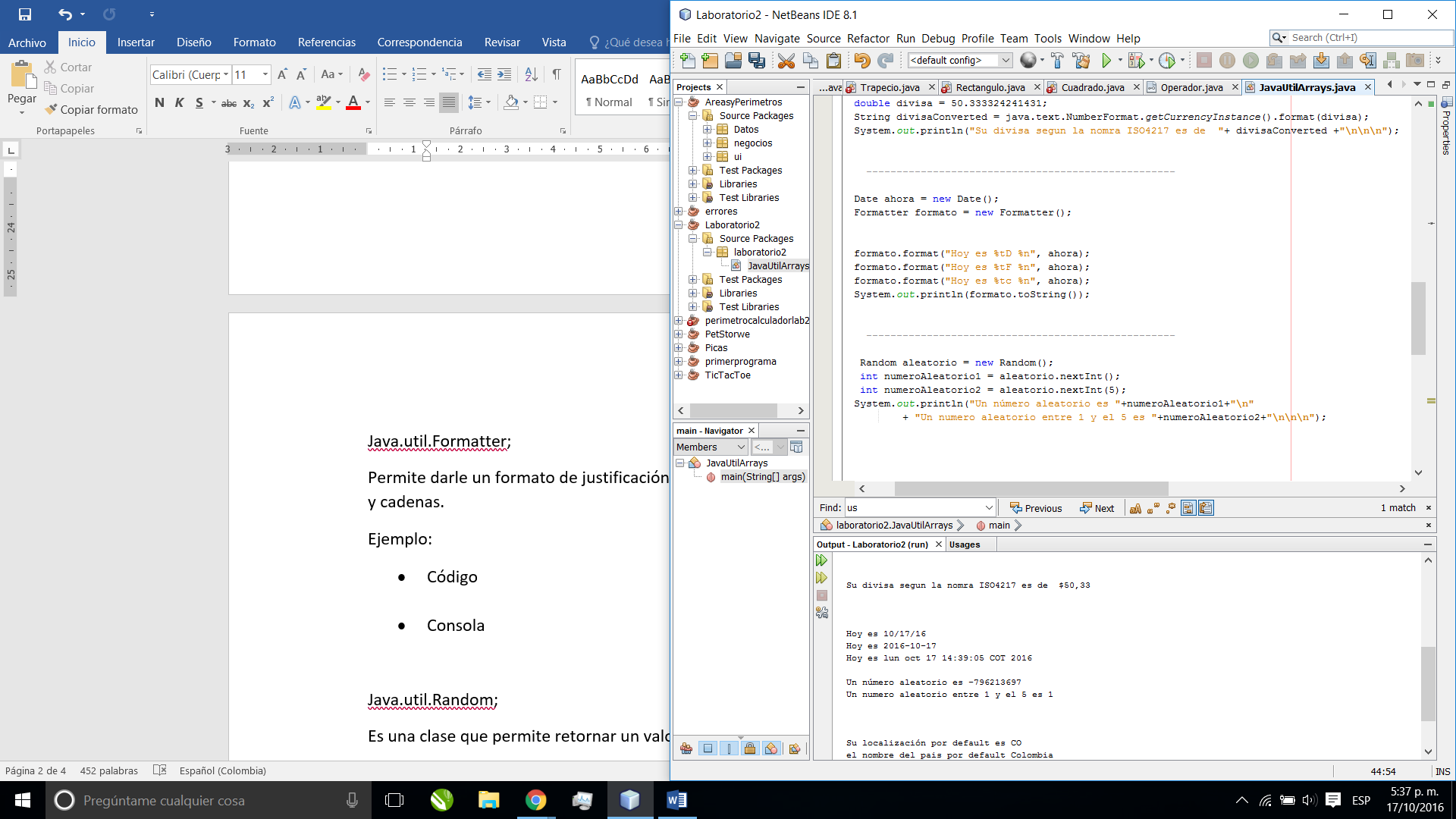


**java.util.Formatter;**

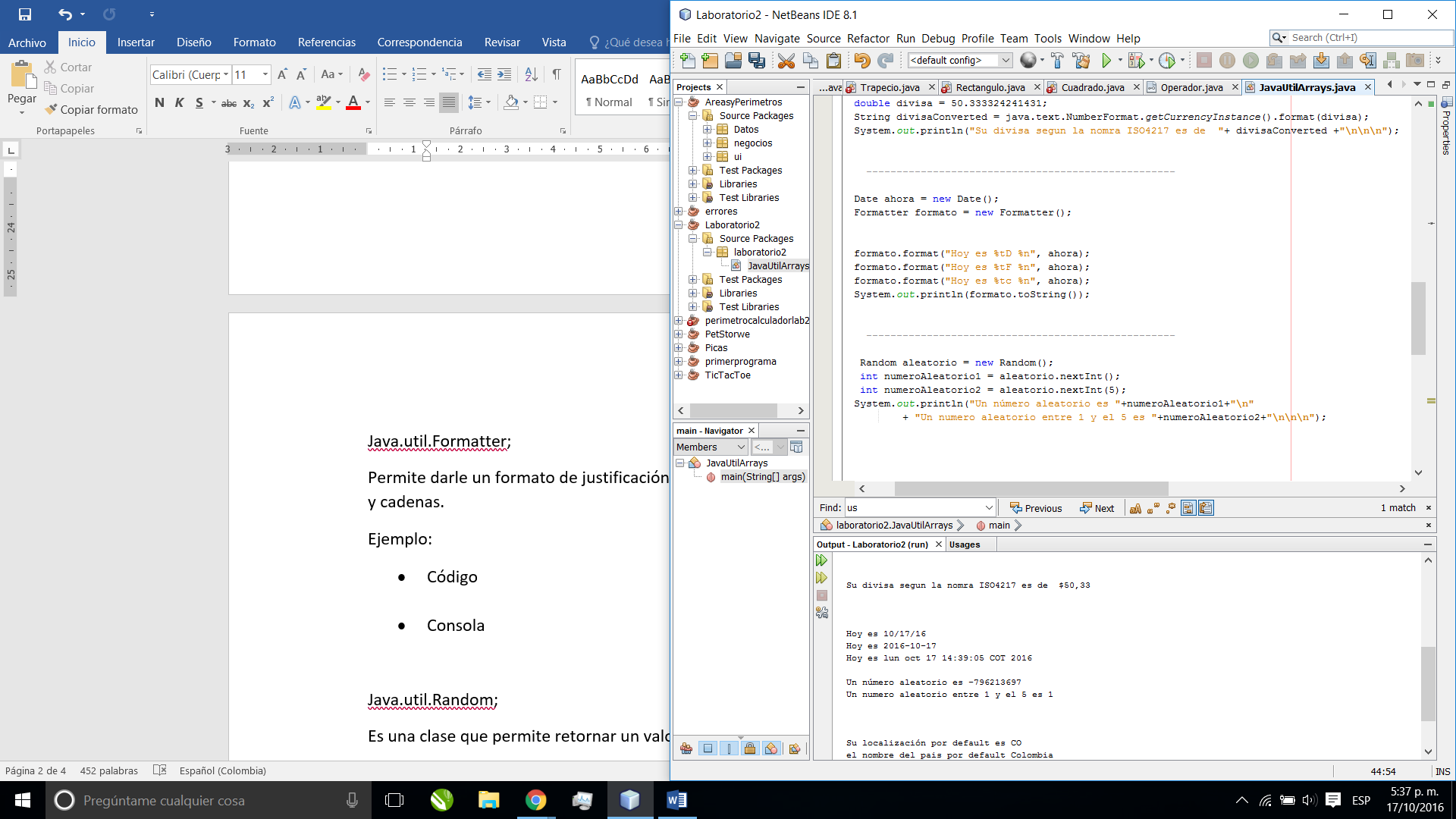
Permite darle un formato de justificación y alineación común para formatos numéricos, de tiempo, y cadenas.

Ejemplo:

* Código



* Consola

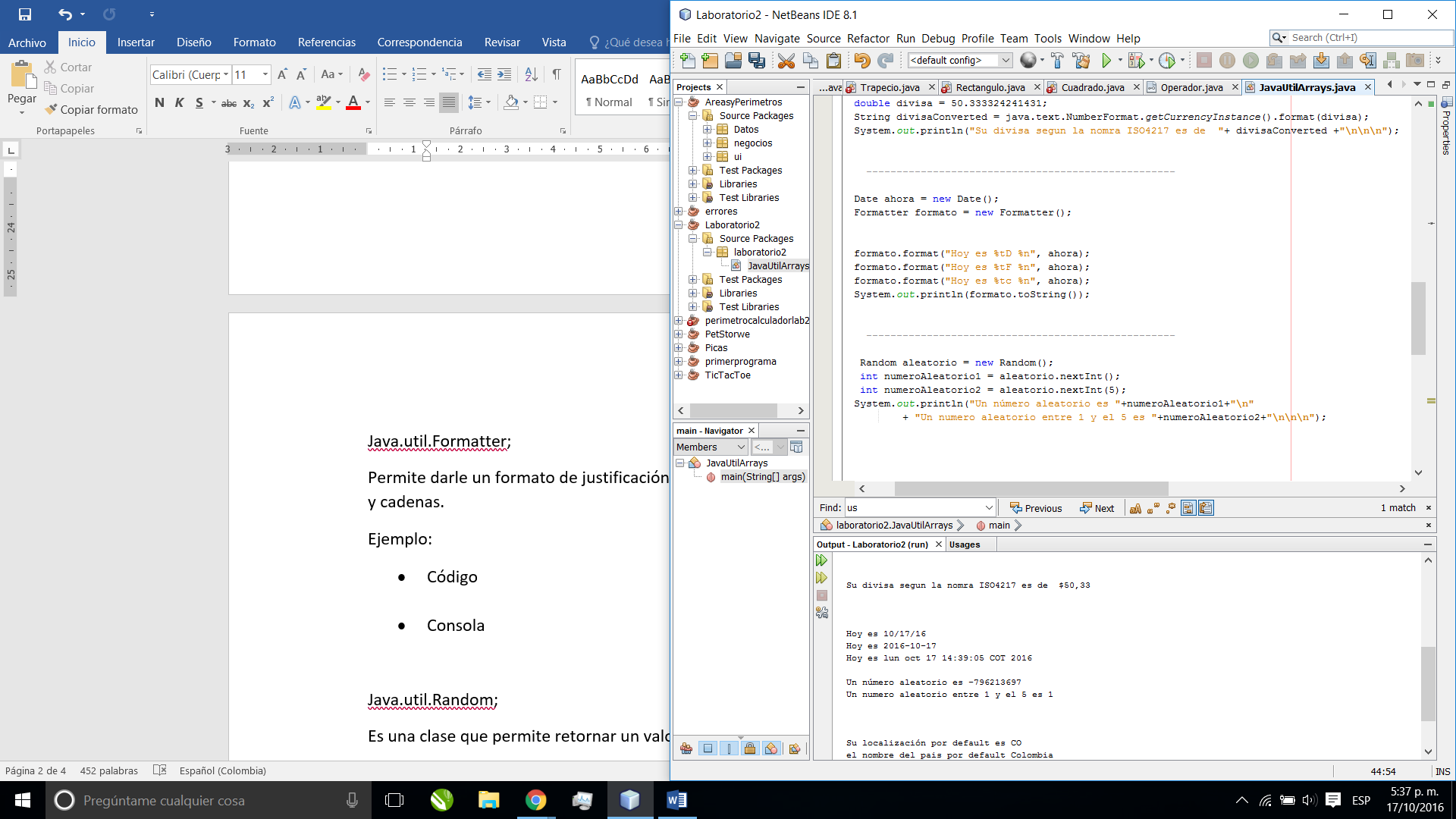


**java.util.Random;**

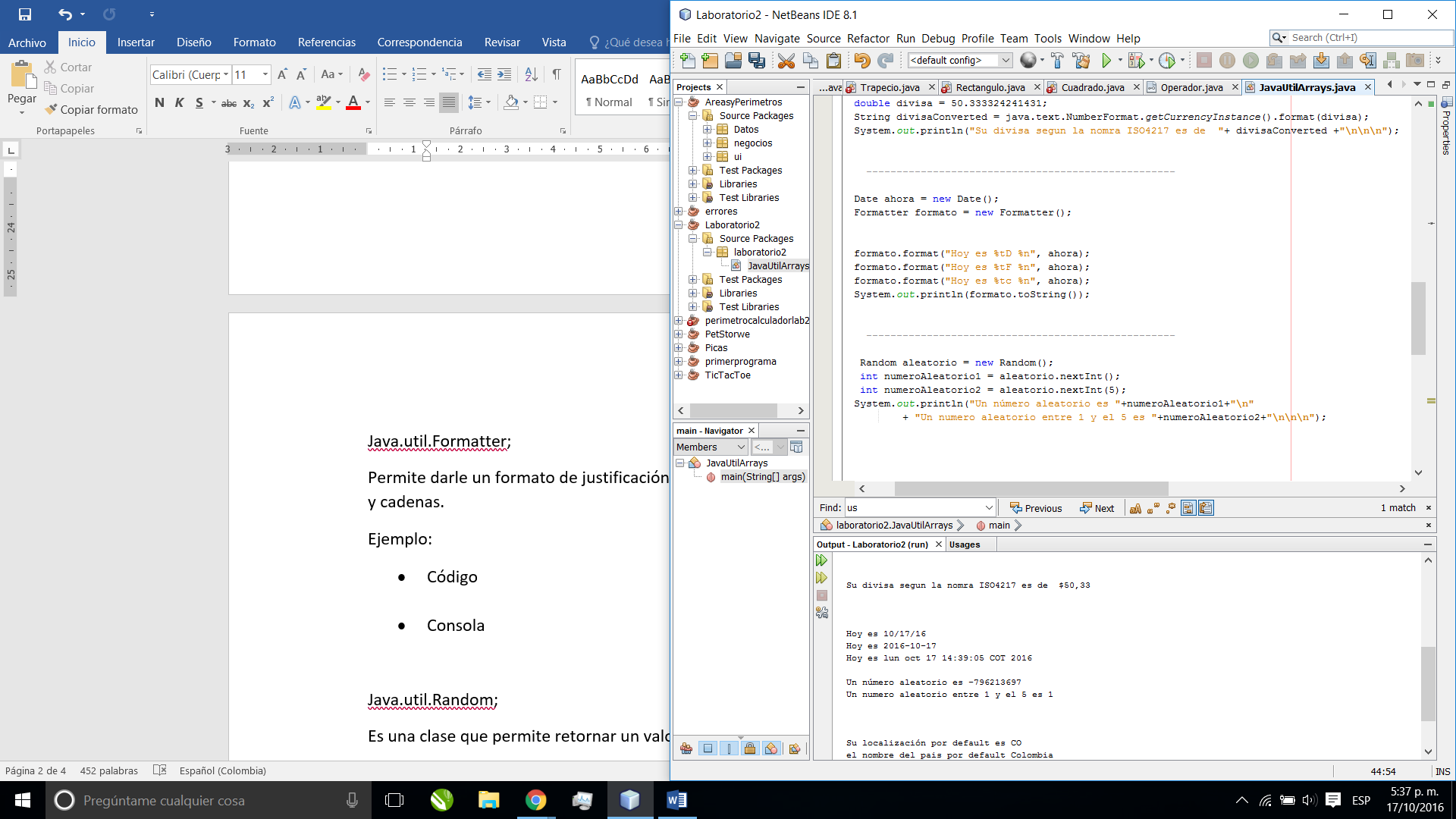
Es una clase que permite retornar un valor pseudoaleatorio de algún tipo de dato.

Ejemplo:

* Código



* Consola

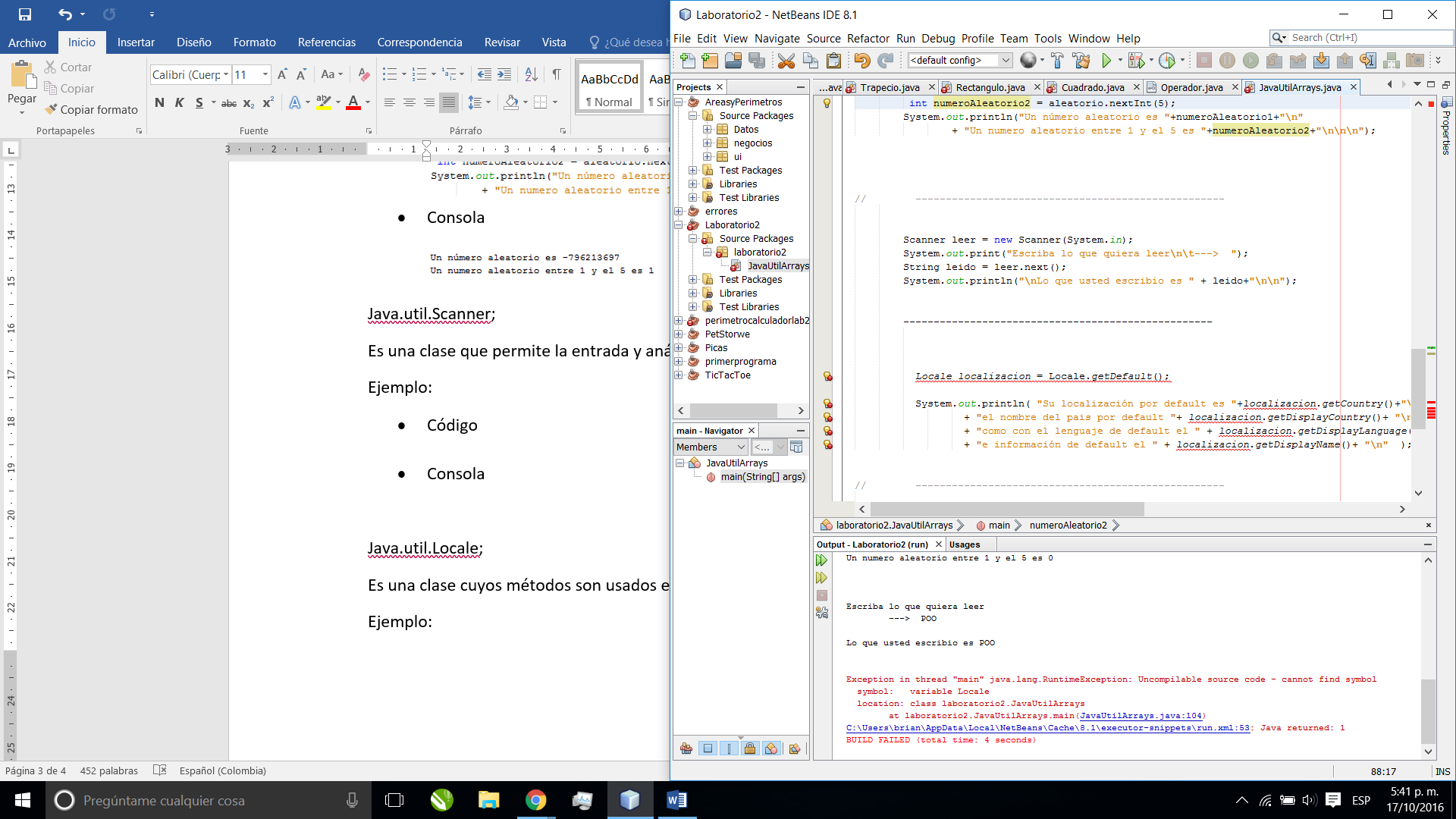


**java.util.Scanner;**

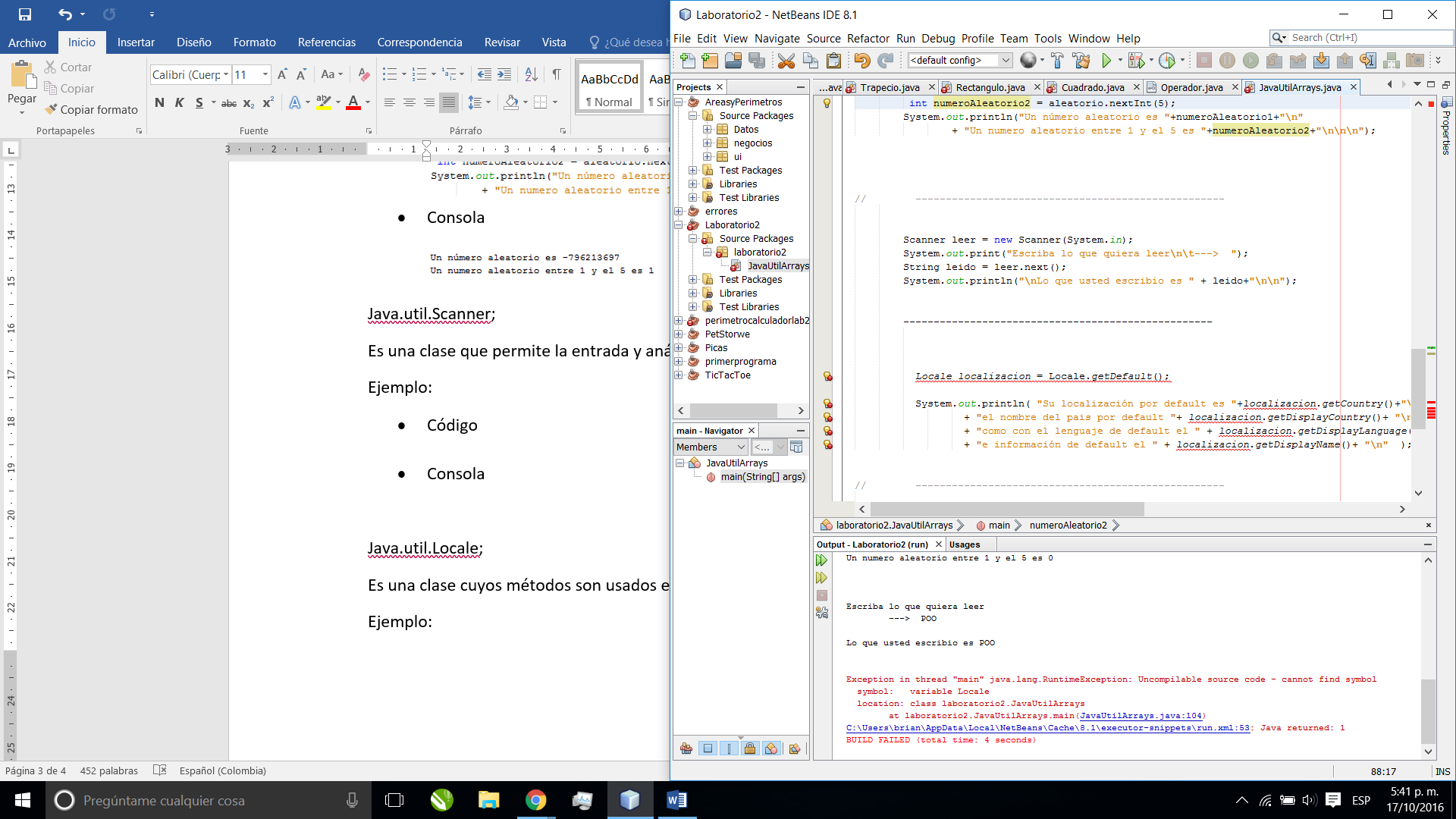
Es una clase que permite la entrada y análisis de datos a partir del sistema y desde archivos.

Ejemplo:

* Código



* Consola

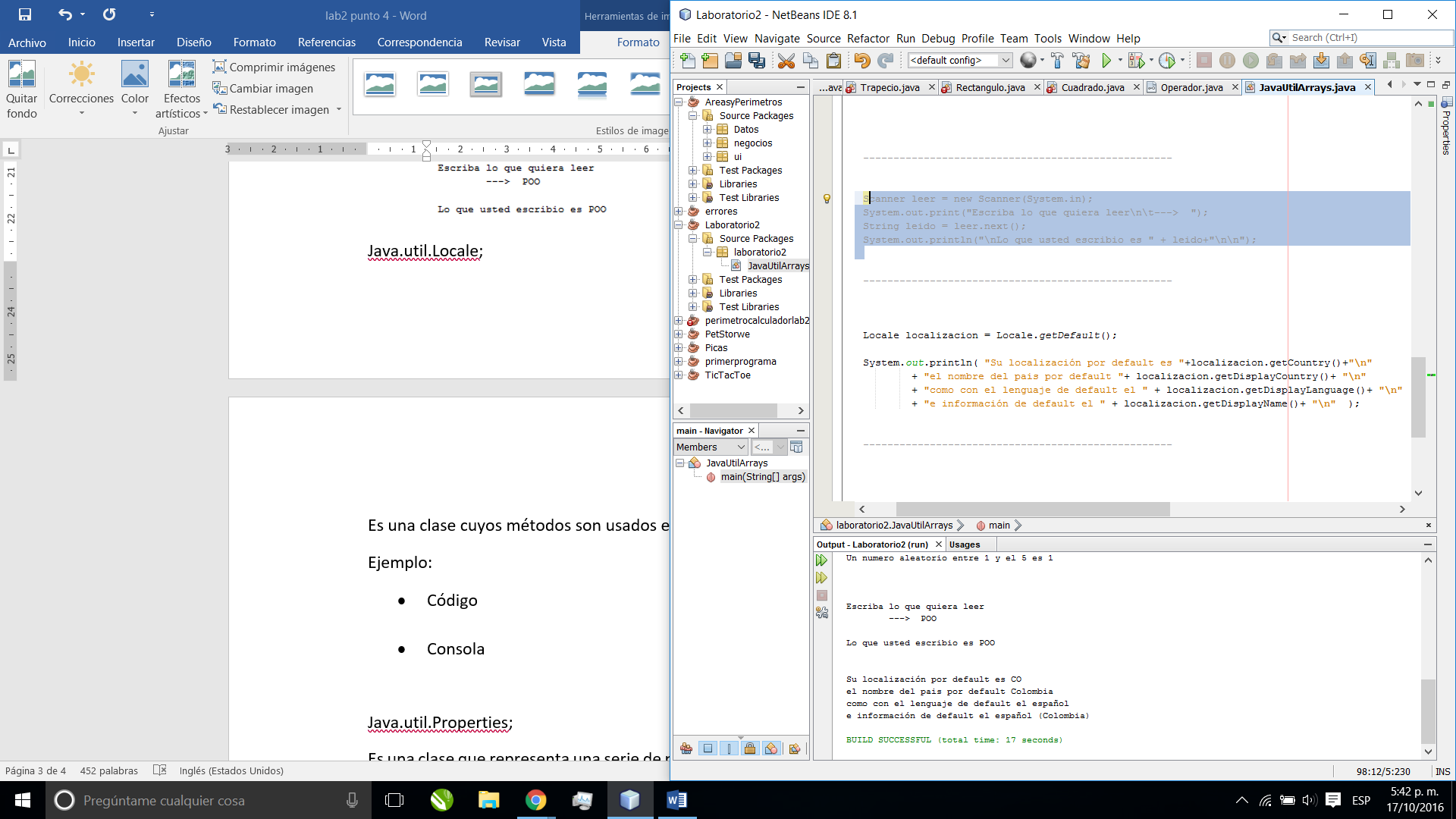


**java.util.Locale;**

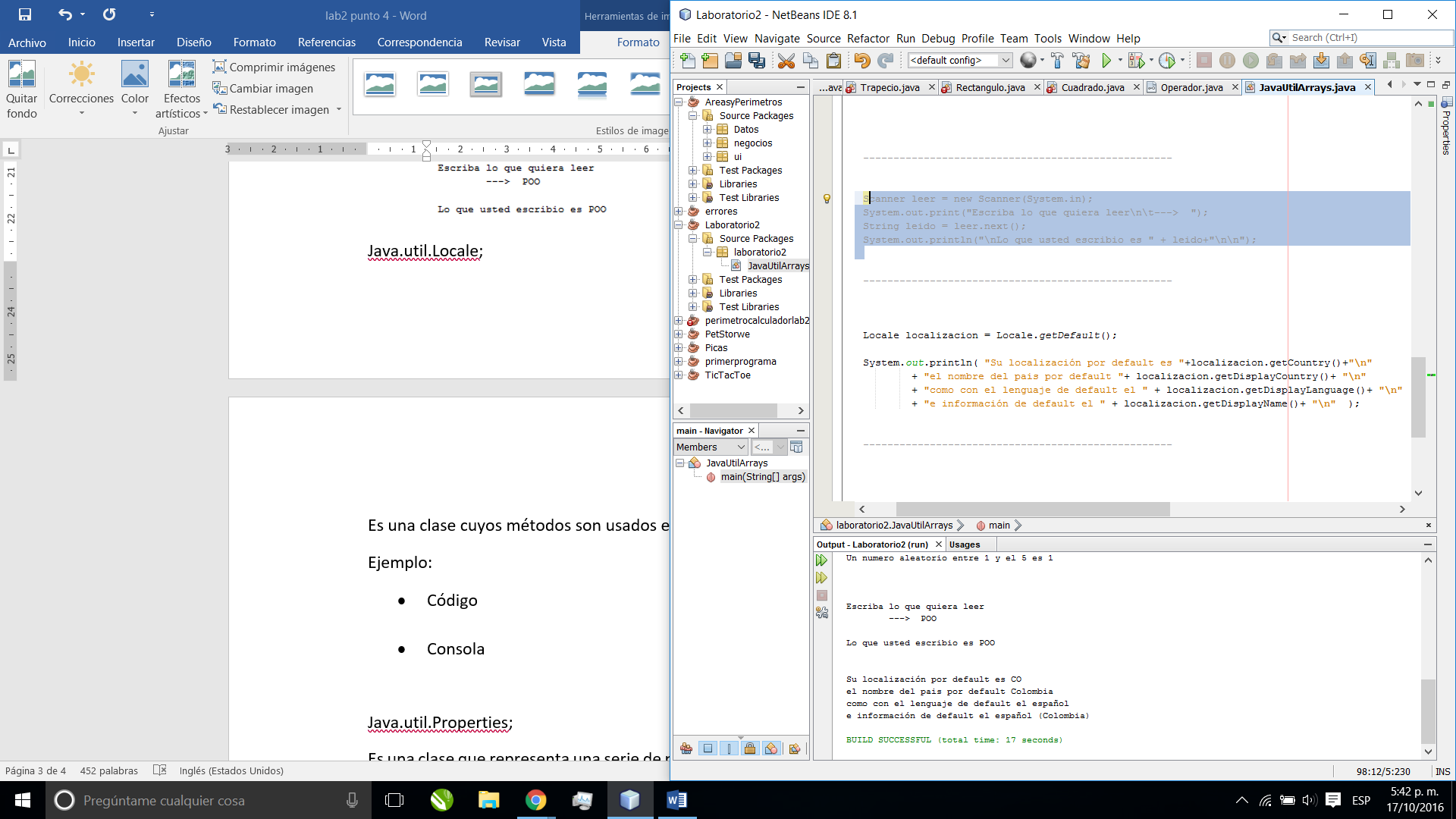
Es una clase cuyos métodos son usados en objetos que sean sensibles a la ubicación y o idioma.

Ejemplo:

* Código



* Consola



**java.util.Properties;**

Es una clase que representa una serie de propiedades que pueden ser por default o que los métodos pueden modificarlas, estas pueden ser guardadas o subidas desde un stream, tal que cada llave y su correspondiente valor en la propiedad es una cadena.

**java.util.Timer;**

Esta clase permite la programación de tareas que mediante un proceso en el fondo logran ser ejecutadas en lapsos de tiempos definidos por medio de sus métodos.