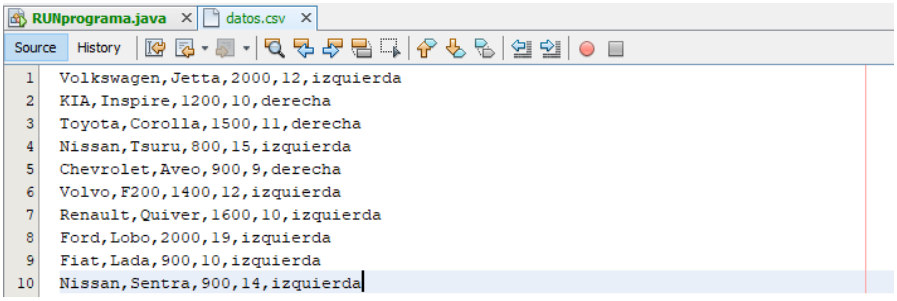
Se creó un archivo csv con los siguientes contenidos:

Donde los elementos de cada renglón están en el formato:

Marca, Modelo, Precio, gasolina, direccion



Se creo un programa con una clase Automóvil que tiene los siguientes campos:

```
public class Automovil {
    private String marca;
    private String modelo;
    private String precio;
    private int gasolina;
```

El constructor de esa clase los inicializa:

```
    Automovil(String marca, String modelo, String precio, int gas) {
        this.marca = marca;
        this.modelo = modelo;
        this.precio = precio;
        encender(gas);
    }
```

La gasolina solo se puede inicializar mediante el método encender() que lanza una excepción si no es mayor que 1.

Al ejecutar el programa, inicializa 10 automóviles leyendo los datos para inicializarlo desde el archivo. Los primeros 4 elementos separados por coma de cada renglón son los datos necesarios para la inicialización.

Después realiza los métodos de cada carro. La salida es la siguiente:

```
Marca: Volkswagen
Modelo: Jetta
precio: $2000
Volteando a la izquierda

Marca: KIA
Modelo: Inspire
precio: $1200
Volteando a la derecha

Marca: Toyota
Modelo: Corolla
precio: $1500
Volteando a la derecha

Marca: Nissan
Modelo: Tsuru
precio: $800
Volteando a la izquierda

Marca: Chevrolet
Modelo: Aveo
precio: $900
Volteando a la derecha

Marca: Volvo
Modelo: F200
precio: $1400
Volteando a la izquierda

Marca: Renault
Modelo: Quiver
precio: $1600
Volteando a la izquierda


Marca: Ford
Modelo: Lobo
precio: $2000
Volteando a la izquierda

Marca: Fiat
Modelo: Lada
precio: $900
Volteando a la izquierda

Marca: Nissan
Modelo: Sentra
precio: $900
Volteando a la izquierda
BUILD SUCCESSFUL (total time: 1 second)
```

Parte 2: Anotaciones.

Se creo la anotación personalizada @DocumentacionMiFuncion con la anotación @Documented.

```java
@Documented
public @interface DocumentacionMiFuncion {
    String entradas() default "N/A";
    String retorno() default "N/A";
    String precauciones() default "N/A";
}
```

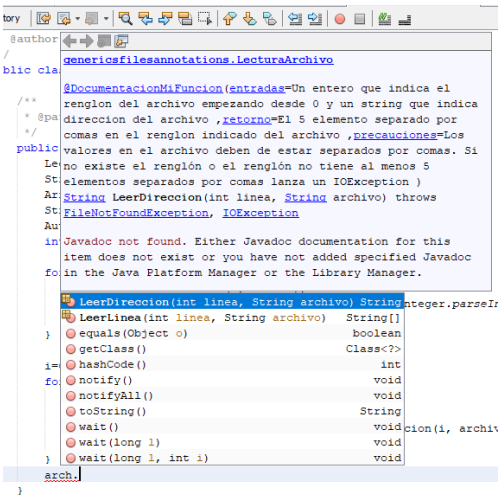
Se uso está anotación en las funciones usadas para leer el archivo:

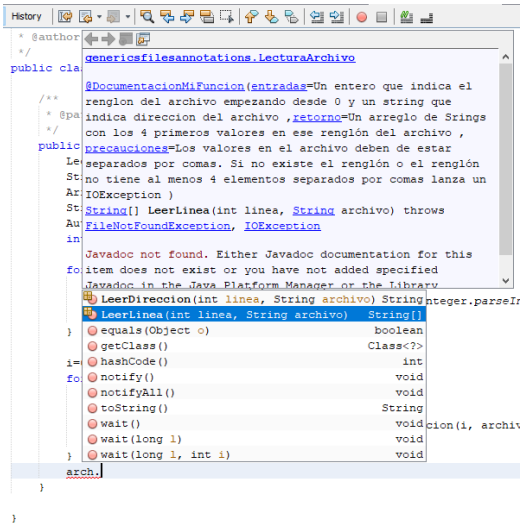
```java
package genericsfilesannotations;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

/**
 *
 * @author Funxb
 */
public class LecturaArchivo {
    @DocumentacionMiFuncion(
            entradas = "Un entero que indica el renglon del archivo empezando desde 0 y un string que indica direccion del archivo",
            retorno = "Un arreglo de Srings con los 4 primeros valores en ese renglón del archivo",
            precauciones = "Los valores en el archivo deben de estar separados por comas. "
                    + "Si no existe el renglón o el renglón no tiene al menos 4 elementos separados por comas lanza un IOException"
    )
    String[] LeerLinea(int linea,String archivo) throws FileNotFoundException, IOException {...45 lines }

    @DocumentacionMiFuncion(
            entradas = "Un entero que indica el renglon del archivo empezando desde 0 y un string que indica direccion del archivo",
            retorno = "El 5 elemento separado por comas en el renglon indicado del archivo",
            precauciones = "Los valores en el archivo deben de estar separados por comas. "
                    + "Si no existe el renglón o el renglón no tiene al menos 5 elementos separados por comas lanza un IOException"
    )
    String LeerDireccion(int linea, String archivo) throws FileNotFoundException, IOException {...40 lines }
```

Ahora cuando se usan aparecen los datos el el IDE:





Además, se creó una clase llamada CarroViejo que es heredada de clase Automóvil. Se marcó esta clase como deprecada con @Deprecated y dentro tiene un método sobreescrito anotado con @Override

```
@Deprecated
public class AutomovilViejo extends Automovil{

    public AutomovilViejo(String marca, String modelo, String precio, int gas) {...3 lines }

    @Override
    public String movimiento(String direccion) {...9 lines }

}
```

Si hago un objeto de tipo carro viejo el IDE me muestra lo siguiente:

```
38                  System.out.println(car
39                  i++;
40              }
41              AutomovilViejo ca;
42          }
43
```