

Entrega un archivo PDF con las siguientes capturas de pantalla y comentarios:

- Código fuente principal (clase **Empleado** y clase de gestión).

Main

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

/**
 * Recoge la informacion de un archivo txt para deserializarlo a una
 * lista de empleados
 * Envia la informacion de la lista de empleados a un archivo json
 * Al producir cualquier error se guardara en un log
 *
 * <p>Ejemplo de uso:
 * <pre>en una empresa <code>
 * Empleados em = new Empleados();
 * List<Empeados> listaEmpleados = listarEmpleadosTXT(emplados);
 * guardarJson(listaEmpleados, empleadoJSON);
 * </code></pre>
 *
 * @author Ignacio Toledano Caneo
 * @version 1.0
 * @since 1.0
 * @see Empleado
 * @see AlbumData
 */
public class Main {
```

```

public static void main(String[] args) throws Exception {

    // archivos
    String empleadosTXT = "src/empleados.txt";
    String empleadoJSON = "src/empleados.json";
    String empleadoLOG = "src/empleados.log";

    System.out.println("Aquí tienes el contenido de el archivos
empleados.txt: ");
    List<Empleado> listaEmpleados = listarTXTEmpleados(empleadosTXT);
    leerLista(listaEmpleados);

    System.out.println("Ahora los empleados se guardaran en un
archivos JSON llamdo " + empleadoJSON);
    guardarJson(listaEmpleados, empleadoJSON);

    escribirLogDeError("error inesperado ", empleadoLOG);
}

// Esta funcion se le pasa solo la ruta del archivo txt para devolver
una lista de empleados.
public static List<Empleado> listarTXTEmpleados(String rutaString) {
    List<Empleado> returnEmpleados = new ArrayList<Empleado>();

    try (BufferedReader rd = new BufferedReader(new
FileReader(rutaString))) {
        String aux;
        Empleado empleadoAUX = new Empleado();
        //Evitamos que coja la primera linea ya que no tiene
contenido
        boolean notFirst = false;

        while ((aux = rd.readLine()) != null) {
            String[] lista = aux.split(";");
            if (notFirst){
                empleadoAUX.setNombre(lista[0]);
                empleadoAUX.setEdad(Integer.parseInt(lista[1]));
            }
        }
    }
}

```

```

        //aquí por si quieres forzar un error para el log
        // empleadoAUX.setEdad(lista[1]);

        empleadoAUX.setDepartamento(lista[2]);
        returnEmpleados.add(empleadoAUX);
    }
    notFirst = true;
}

} catch (Exception e) {
    String error = (" No se ha podido leer el archivo. ");
    escribirLogDeError(error, "src/empleados.log");
}
return returnEmpleados;
}

//leemos la lista de empleados con un foreach y luego la imprimimos
para mostrársela a la persona
public static void leerLista(List<Empleado> empleados) {
    for (Empleado empleado : empleados) {
        System.out.println(empleado.toString());
    }
}

// guardar la lista de empleados recogida del txt para serializarla
en un json
public static void guardarJson(List<Empleado> empleados, String
archivo){
    Gson gson = new GsonBuilder().setPrettyPrinting().create();
    try {
        FileWriter w = new FileWriter(archivo);
        //le damos formato json para que a la hora de guardarse se
guarde con formato json
        w.write(gson.toJson(empleados));
        w.close();
        System.out.println("Se ha guardado correctamente en el
archivo json los empleados. ");
        // int num = 0/0;

```

```

    } catch (Exception e) {
        String error = " Ha fallado la subida de archivos a json. ";
        escribirLogDeError(error, "src/empleados.log");
    }
}

// Archivamos todos los errores que pueda dar el programa en un
archivo log
public static void escribirLogDeError(String error, String archivo){
    try {
        FileWriter w = new FileWriter(archivo, true);
        Date fecha = new Date();
        String errorMod = fecha + ". Ha ocurrido el error: " + error
+ " en el dia. " ;
        w.write(errorMod);
        w.close();
    } catch (Exception e) {
        // En caso de que de un error tambien el escribir en el log
se mostrara un error en terminal
        System.err.println("Ha ocurrido un error tan grave que no se
guardara en el LOG");
    }
}
}

```

Empleado:

```

public class Empleado {

    private String nombre;
    private int edad;
    private String departamento;
    public Empleado(String nombre, int edad, String departamento) {
        this.nombre = nombre;
        this.edad = edad;
        this.departamento = departamento;
    }
}

```

```
public Empleado() {  
    }  
    public String getNombre() {  
        return nombre;  
    }  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
    public int getEdad() {  
        return edad;  
    }  
    public void setEdad(int edad) {  
        this.edad = edad;  
    }  
    public String getDepartamento() {  
        return departamento;  
    }  
    public void setDepartamento(String departamento) {  
        this.departamento = departamento;  
    }  
    @Override  
    public String toString() {  
        return "Empleado [nombre=" + nombre + ", edad=" + edad + ",  
departamento=" + departamento + "];"  
    }  
}
```

- Ejecución por consola mostrando lectura del TXT y creación del JSON.

```

src > Main.java > Main > escribiLogDeError(String, String)
public class Main {
    public static void leerLista(List<Empleado> empleados) {
        // Leer la lista de empleados desde el archivo txt
        // ... (código omitido) ...
    }

    // guardar la lista de empleados recogida del txt para serializarla en un json
    public static void guardarJson(List<Empleado> empleados, String archivo){
        Gson gson = new GsonBuilder().setPrettyPrinting().create();
        try {
            FileWriter w = new FileWriter(archivo);
            // le damos formato json para que a la hora de guardarse se guarde con format.
            w.write(gson.toJson(empleados));
            w.close();
            System.out.println("Se ha guardado correctamente en el archivo json los em
            // int num = 0/0;
        } catch (Exception e) {
            String error = " Ha fallado la subida de archivos a json. ";
            escribiLogDeError(error, archivo:"src/empleados.log");
        }
    }

    // Archivamos todos los errores que pueda dar el programa en un archivo log
    public static void escribiLogDeError(String error, String archivo){
        try {
            FileWriter w = new FileWriter(archivo, append:true);
            Date fecha = new Date();
            String errorMod = fecha + ". Ha ocurrido el error: " + error + " en el día.
            w.write(errorMod);
            w.close();
        } catch (Exception e) {
            // En caso de que de un error tambien el escribir en el log se mostrara un e
            System.err.println("Ha ocurrido un error tan grave que no se guardara en e
        }
    }
}

```

```

src > empleados.json > ...
1 {
2   {
3     "nombre": "Carlos",
4     "edad": 40,
5     "departamento": "Sistemas"
6   },
7   {
8     "nombre": "Carlos",
9     "edad": 40,
10    "departamento": "Sistemas"
11  },
12  {
13    "nombre": "Carlos",
14    "edad": 40,
15    "departamento": "Sistemas"
16  }
17 }

```

```

TERMINAL
administrador@administrador-System-Product-Name:~/Documentos/CodigosDAMM/DAM2/EXAMENES/ACCESO A DATOS/TEMA1/P1/Examen1/ExamenTema1$ cd /home/administrador/Documentos/CodigosDAMM/DAM2/EXAMENES/ACCESO A DATOS/TEMA1/P1/Examen1/ExamenTema1
$ java -cp .:src:../lib/gson-2.10.2.jar Main
Aquí tienes el contenido de el archivos empleados.txt:
Empleado [nombre=Carlos, edad=40, departamento=Sistemas]
Ahora los empleados se guardaran en un archivos json llamdo src/empleados.json
Se ha guardado correctamente en el archivo json los empleados.
administrador@administrador-System-Product-Name:~/Documentos/CodigosDAMM/DAM2/EXAMENES/ACCESO A DATOS/TEMA1/P1/Examen1/ExamenTema1$

```

- Contenido de los archivos `empleados.json` y `errores.log` (fuerza algún error para que se genere).

```

src > empleados.log
1 Fri Oct 24 12:01:02 CEST 2025. Ha ocurrido el error: Ha fallado la subida de archivos a json.

```

```

src > empleados.json > ...
1 {
2   {
3     "nombre": "Carlos",
4     "edad": 40,
5     "departamento": "Sistemas"
6   },
7   {
8     "nombre": "Carlos",
9     "edad": 40,
10    "departamento": "Sistemas"
11  },
12  {
13    "nombre": "Carlos",
14    "edad": 40,
15    "departamento": "Sistemas"
16  }
17 }

```

- Un breve comentario (3–4 líneas) explicando qué parte te resultó más sencilla y cuál más difícil y por qué.

La parte que me ha resultado más difícil a sido la parte de las excepciones a la hora de enviarlas a el log debido a que no me he acordado como se pueden sacar para un string.