

Clase registro donde están los atributos en privado para uso posterior con getter y setters en la clase main.

```
class Registro {  
    2 usages  
    private String dni;  
    2 usages  
    private String nombre;  
    2 usages  
    private int edad;  
    2 usages  
    private String puesto;  
    2 usages  
    private double salario;  
  
    2 usages new *  
    public Registro(String dni, String nombre, int edad, String puesto, double salario) {  
        this.dni = dni;  
        this.nombre = nombre;  
        this.edad = edad;  
        this.puesto = puesto;  
        this.salario = salario;  
    }  
  
    1 usage new *  
    public String getDni() {  
        return dni;  
    }  
  
    1 usage new *  
    public String getNombre() {
```

Clase main:

Primero se prueba si existe el fichero y se introducen los registros

```
public static void main(String[] args) {  
    //CheckFileExistence evita  
    checkFileExistence();  
    introducirRegistros();  
    Scanner scanner = new Scanner(System.in);  
    int choice;  
    //Do-while para que siempre salga el menú hasta que el usuario introduzca 4  
    do {  
        System.out.println("Menú:");  
        System.out.println("1. Visualizar registros");  
        System.out.println("2. Visualizar sueldo mayor");  
        System.out.println("3. Visualizar sueldo medio");  
        System.out.println("4. Exit");  
        System.out.print("Ingrese su elección: ");  
        choice = scanner.nextInt();  
        scanner.nextLine(); // Limpiar el buffer de entrada
```

Se hace un do-while del menú para que el usuario no salga hasta que pulse 4.

```
do {  
    System.out.println("Menú:");  
    System.out.println("1. Visualizar registros");  
    System.out.println("2. Visualizar sueldo mayor");  
    System.out.println("3. Visualizar sueldo medio");  
    System.out.println("4. Exit");  
    System.out.print("Ingrese su elección: ");  
    choice = scanner.nextInt();  
    scanner.nextLine(); // Limpia el buffer de entrada  
  
    switch (choice) {  
        case 1:  
            // introducirRegistros();  
            visualizarRegistros();  
            System.out.println("nada");  
            break;  
        case 2:  
            // visualizarRegistros();  
            visualizarSueldoMayorYMedio();  
            break;  
        case 3:  
            sueldoMedio();  
            break;  
        case 4:  
            System.out.println("Saliendo del programa. ¡Hasta luego!");  
            break;  
        default:  
            System.out.println("Opción no válida. Por favor, ingrese una opción válida.");  
            break;  
    }  
}
```

Método para evitar el fallo de si el fichero no existe

```
/**
 * Verifica si existe el fichero
 */
1 usage new *
private static void checkFileExistence() {
    File file = new File(XML_FILE_PATH);

    if (!file.exists()) {
        try {
            file.createNewFile();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

/**
```

Método para introducir 4 registros

```
/**
 * Introducir registros
 */
1 usage new *
private static void introducirRegistros() {
    List<Registro> registros = new ArrayList<>();
    Scanner scanner = new Scanner(System.in);
    //For para obligar que se introduzcan 4 registros.
    for (int i = 0; i < 4; i++) {
        System.out.println("Registro #" + (i + 1));
        System.out.print("DNI: ");
        String dni = scanner.nextLine();
        System.out.print("Nombre: ");
        String nombre = scanner.nextLine();
        System.out.print("Edad: ");
        int edad = scanner.nextInt();
        scanner.nextLine(); // Limpiar el buffer de entrada
        System.out.print("Puesto: ");
        String puesto = scanner.nextLine();
        System.out.print("Salario: ");
        double salario = scanner.nextDouble();
        scanner.nextLine(); // Limpiar el buffer de entrada

        registros.add(new Registro(dni, nombre, edad, puesto, salario));
    }

    // Guardar registros en el archivo XML
    escribirRegistrosEnXML(registros);
}
```

Carga los registros que tiene el xml

```
private static List<Registro> cargarRegistrosDesdeXML() {  
    List<Registro> registros = new ArrayList<>();  
  
    try {  
        File xmlFile = new File(XML_FILE_PATH);  
        DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();  
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();  
        Document doc = docBuilder.parse(xmlFile);  
  
        doc.getDocumentElement().normalize();  
  
        NodeList nodeList = doc.getElementsByTagName("registro");  
  
        for (int i = 0; i < nodeList.getLength(); i++) {  
            Node node = nodeList.item(i);  
  
            if (node.getNodeType() == Node.ELEMENT_NODE) {  
                Element element = (Element) node;  
                String dni = element.getElementsByTagName("dni").item(0).getTextContent();  
                String nombre = element.getElementsByTagName("nombre").item(0).getTextContent();  
                int edad = Integer.parseInt(element.getElementsByTagName("edad").item(0).getTextContent());  
                String puesto = element.getElementsByTagName("puesto").item(0).getTextContent();  
                double salario = Double.parseDouble(element.getElementsByTagName("salario").item(0).getTextContent());  
  
                registros.add(new Registro(dni, nombre, edad, puesto, salario));  
            }  
        }  
    }  
}
```

Escriben los registros en xml

```
private static void escribirRegistrosEnXML(List<Registro> nuevosRegistros) {  
    // Cargar registros existentes desde el archivo XML  
    List<Registro> registrosExistentes = cargarRegistrosDesdeXML();  
  
    // Agregar nuevos registros a la lista existente  
    registrosExistentes.addAll(nuevosRegistros);  
    try {  
        DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();  
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();  
  
        // Elemento raíz  
        Document doc = docBuilder.newDocument();  
        Element rootElement = doc.createElement( tagName: "registros");  
        doc.appendChild(rootElement);  
  
        // Agregar cada registro como un elemento al XML  
        for (Registro registro : registrosExistentes) {  
            Element registroElement = doc.createElement( tagName: "registro");  
  
            Element dniElement = doc.createElement( tagName: "dni");  
            dniElement.appendChild(doc.createTextNode(registro.getDni()));  
            registroElement.appendChild(dniElement);  
  
            Element nombreElement = doc.createElement( tagName: "nombre");  
            nombreElement.appendChild(doc.createTextNode(registro.getNombre()));  
            registroElement.appendChild(nombreElement);  
  
            Element edadElement = doc.createElement( tagName: "edad");
```

Transformer sirve para el xml se guarde con estructura de xml porque sino lo hace todo en una línea.

```
// Escribir el contenido en el archivo XML
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
transformer.setOutputProperty(OutputKeys.INDENT, value: "yes");
transformer.setOutputProperty(name: "{http://xml.apache.org/xslt}indent-amount", value: "2");

DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(new File(XML_FILE_PATH));
transformer.transform(source, result);

System.out.println("Registros almacenados en " + XML_FILE_PATH);

} catch (ParserConfigurationException | TransformerException e) {
    System.out.println("Error por: "+e);
}
}
```

Visualiza los registros del xml

```
/**
 * Visualizar registros: Sirve para mostrar todos los registros que tiene el xml
 */
1 usage new *
private static void visualizarRegistros() {
    try {
        File xmlFile = new File(XML_FILE_PATH);
        DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        NodeList nodeList = doc.getElementsByTagName("registro");

        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);

            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) node;
                System.out.println("DNI: " + element.getElementsByTagName("dni").item(0).getTextContent());
                System.out.println("Nombre: " + element.getElementsByTagName("nombre").item(0).getTextContent());
                System.out.println("Edad: " + element.getElementsByTagName("edad").item(0).getTextContent());
                System.out.println("Puesto: " + element.getElementsByTagName("puesto").item(0).getTextContent());
                System.out.println("Salario: " + element.getElementsByTagName("salario").item(0).getTextContent());
                System.out.println("-----");
            }
        }
    }
}
```


Método para visualizar el sueldo medio

```
1 usage: new *
private static void sueldoMedio(){
    try {
        File xmlFile = new File(XML_FILE_PATH);
        DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        NodeList nodeList = doc.getElementsByTagName("registro");
        double sueldoMayor = 0;
        double sueldoTotal = 0;

        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);

            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) node;
                double salario = Double.parseDouble(element.getElementsByTagName("salario").item(0).getTextContent());
                sueldoTotal += salario;

                if (salario > sueldoMayor) {
                    sueldoMayor = salario;
                }
            }
        }

        double sueldoMedio = sueldoTotal / nodeList.getLength();

        // System.out.println("Sueldo Mayor: " + sueldoMayor);
        System.out.println("Sueldo Medio: " + sueldoMedio);
    } catch (ParserConfigurationException | SAXException | IOException e) {
        e.printStackTrace();
    }
}
```