

# Informe de Clases de la Tarea para AD02.

*El proyecto contiene dos clases Main, por lo que para su ejecución debe hacerse por separado para cada ejercicio.*

El proyecto consta de tres paquetes:

- **com.cyphrestudios.ad02.apartado01** : contiene la clase main para la ejecución del ejercicio 1.
- **com.cyphrestudios.ad02.apartado02** : contiene la clase main para la ejecución del ejercicio 2.
- **com.cyphrestudios.ad02.model** : contiene las clases que implementan todas las funcionalidades, a través de sus métodos, requeridas en la tarea.

## Ejercicio 1 – Breve Informe

Clase Empleado
----------------

Modela la entidad Empleado y proporciona métodos para escribir y leer registros de empleados en archivos mediante la técnica de acceso aleatorio (RandomAccessFile). Cada instancia de Empleado representa un registro con atributos como código de empleado, nombre, apellidos, dirección, población, código postal, salario y antigüedad.

Facilita la manipulación de registros de empleados en archivos, especialmente diseñada para el acceso aleatorio y la gestión de cadenas con longitud fija.

### Atributos:

- codEmp: Código de empleado (int).
- nombre: Nombre del empleado (String).
- apellidos: Apellidos del empleado (String).
- direccion: Dirección del empleado (String).
- poblacion: Población del empleado (String).
- cp: Código postal del empleado (int).
- salario: Salario del empleado (float).
- antigüedad: Antigüedad del empleado (int).

### Constante:

- TAMANIO\_REGISTRO: Tamaño total de un registro de empleado en bytes.

### Métodos Principales:

- escribirEmpleado: Escribe un registro de empleado en el archivo.
- leerEmpleado: Lee un registro de empleado desde el archivo en una posición específica.
- escribirCadena: Método auxiliar para escribir cadenas en el archivo con longitud fija.
- leerCadena: Método auxiliar para leer cadenas del archivo.
- También incluye los métodos get de acceso a sus atributos y los set para establecer los valores de cada instancia de la clase, así como el método toString.

#### Clase GestionEmpleados:

Facilita la creación y manipulación de empleados, así como su escritura en un archivo de acceso aleatorio, ofreciendo métodos para su gestión.

##### Funcionalidades Principales:

- **Creación de Empleados:**  
Genera una lista de empleados predefinidos con sus respectivos detalles, como código, nombre, dirección, salario, etc.
- **Escritura de Empleados en Archivo:**  
Escribe la información de los empleados en un archivo de acceso aleatorio.

##### Métodos Principales:

- `crearEmpleados`: Crea y devuelve una lista de empleados predefinidos.
- `escribirEmpleadosEnArchivo`: Escribe la información de los empleados en un archivo de acceso aleatorio.

#### Clase AlmacenarEmpleados:

Se encarga de escribir datos de empleados almacenados en un archivo de acceso aleatorio en un formato XML. Convirtiendo los datos de empleados almacenados en un archivo de acceso aleatorio en un archivo XML estructurado.

##### Funcionalidades Principales:

- **Generación de Archivo XML:**  
Utiliza la API DOM para crear un documento XML que refleje la información de los empleados. Cada empleado se representa como un conjunto de nodos XML con detalles como código, nombre, dirección, salario, etc.
- **Escritura en Archivo XML:**  
Los nodos XML generados se escriben en un archivo XML designado en el sistema de archivos.

##### Uso y Resultado:

Llamando al método `escribirXML` con el nombre del archivo de datos de empleados (DAT) como parámetro, se generará un archivo XML (EMPLEADOS.XML) en el directorio raíz del proyecto que contiene la información estructurada de los empleados.

#### Clase Main LaunchApp01:

Punto de entrada de la aplicación y realiza dos operaciones fundamentales: crea empleados y los escribe en un archivo (EMPLEADOS.DAT). Luego, utiliza la clase `AlmacenarEmpleados` para convertir esos datos en un archivo XML (EMPLEADOS.XML).

La clase maneja excepciones relacionadas con la manipulación de archivos y utiliza las clases `Empleado`, `GestionEmpleados`, y `AlmacenarEmpleados` para llevar a cabo estas operaciones en el contexto de una aplicación de gestión de empleados.

## Apartado 2 – Breve Informe

### Clase XMLDOM - Resumen Breve

Implementa la funcionalidad para visualizar todas las etiquetas del archivo XML "libros.xml" utilizando la técnica DOM. Se apoya en la API DOM para analizar la estructura del documento XML y mostrar los detalles de cada libro.

#### Funcionalidades Principales:

- **Lectura del Archivo XML:**  
Utiliza la API DOM para crear un árbol de objetos representando la estructura del documento XML.
- **Obtención y Visualización de Detalles:**  
Itera sobre la lista de nodos "libro" del documento XML.  
Accede a los elementos específicos como título, autores, editorial, año y precio para cada libro.
- **Impresión de Información:**  
Muestra los detalles de cada libro en un formato específico.

#### Método Principal:

- `printXMLFileDOM`: Lee el archivo XML "libros.xml" y muestra los detalles de cada libro utilizando la técnica DOM.

### Clase XMLSAX

Implementa un analizador SAX para procesar el archivo XML "libros.xml". Utiliza la tecnología SAX para manejar eventos como la apertura y cierre de etiquetas, y la lectura de texto entre ellas. Esta clase presenta métodos para imprimir información específica sobre los libros contenidos en el archivo XML.

#### Funcionalidades Principales:

- **Métodos:**  
Utiliza la API SAX para procesar eventos durante el análisis del archivo XML.
- **Almacenamiento de Información:**  
Mantiene variables para almacenar información relevante de cada libro, como título, autores, editorial, año y precio.
- **Impresión de Información:**  
Imprime la información de cada libro en un formato específico cuando se cierra la etiqueta "libro".

#### Método de Uso:

- `printXMLFileSAX`: Inicia el proceso de análisis del archivo XML "libros.xml", utilizando la instancia actual de XMLSAX como manejador de eventos SAX.

### Clase Main LaunchApp02

Implementa la funcionalidad para visualizar todas las etiquetas del archivo XML "libros.xml" utilizando las técnicas DOM y SAX. Se compone de dos bloques principales para cada técnica, instanciando las clases **XMLDOM** y **XMLSAX** respectivamente.