

TAREAS Ampliación FICHEROS

Tarea 1: Gestor de Contactos

Objetivo: Desarrollar una aplicación de gestión de contactos utilizando ficheros de texto.

Pasos:

1. Crea un nuevo proyecto Java en IntelliJ IDEA llamado "GestorContactos".
2. Inicializa un repositorio Git local: VCS > Enable Version Control Integration > Git.
3. Crea una clase Contacto con atributos: nombre, teléfono, email.
4. Implementa una clase GestorContactos con métodos para:
 - Añadir contacto
 - Listar contactos
 - Buscar contacto
 - Eliminar contacto
5. Utiliza FileWriter y FileReader para escribir y leer contactos en un archivo de texto.
6. Implementa un menú en consola para interactuar con el usuario.
7. Usa try-with-resources para manejar los recursos de archivo.
8. Realiza commits frecuentes con mensajes descriptivos.
9. Crea un repositorio en GitHub y haz push de tu código.

Entregable: URL del repositorio GitHub con el código fuente y un README.md con instrucciones de uso.

Tarea 2: Conversor CSV a XML

Objetivo: Crear una aplicación que convierta datos de estudiantes de formato CSV a XML.

Pasos:

1. Crea un nuevo proyecto Java en IntelliJ IDEA llamado "ConversorCSVaXML".
2. Inicializa un repositorio Git local.
3. Crea un archivo CSV de ejemplo con datos de estudiantes (ID, Nombre, Apellido, Edad, Curso).

4. Implementa una clase `ConversorCSVaXML` que:
 - Lea el archivo CSV usando `CSVReader` (puedes usar una librería externa como `OpenCSV`).
 - Genere un archivo XML estructurado con los mismos datos.
 - Utilice `javax.xml.parsers.DocumentBuilder` para crear el documento XML.
5. Añade validación para asegurar que el CSV tiene el formato correcto.
6. Implementa logging utilizando SLF4J con Logback.
7. Crea pruebas unitarias con JUnit para las funciones principales.
8. Utiliza Maven o Gradle para gestionar las dependencias del proyecto.
9. Realiza commits frecuentes y significativos.
10. Crea un repositorio en GitHub y sube tu código.

Entregable: URL del repositorio GitHub, archivo CSV de ejemplo, y el XML generado.

Tarea 3: Analizador de Logs

Objetivo: Crear un programa que analice un archivo de log y genere estadísticas.

Pasos:

1. Crea un nuevo proyecto Java en IntelliJ IDEA llamado "AnalizadorLogs".
2. Inicializa un repositorio Git local.
3. Crea un archivo de log de ejemplo con entradas que incluyan: timestamp, nivel de log (INFO, WARNING, ERROR), y mensaje.
4. Implementa una clase `AnalizadorLog` que:
 - Lea el archivo de log línea por línea usando `BufferedReader`.
 - Cuente las ocurrencias de cada nivel de log.
 - Identifique las 5 mensajes de error más comunes.
5. Genera un informe en un nuevo archivo de texto con las estadísticas.
6. Utiliza streams de Java 8+ para procesar los datos cuando sea posible.
7. Implementa manejo de excepciones adecuado.
8. Realiza commits para cada funcionalidad importante.
9. Crea un repositorio en GitHub y sube tu código.

Entregable: URL del repositorio GitHub, archivo de log de ejemplo, y el informe generado.

Instrucciones generales para todas las tareas

1. Utiliza la integración de Git en IntelliJ IDEA para todas las operaciones de control de versiones.
2. Crea ramas separadas para características principales y fusiónalas a través de Pull Requests en GitHub.
3. Configura un archivo .gitignore apropiado para proyectos Java e IntelliJ IDEA.
4. Documenta tu código con comentarios JavaDoc.
5. Incluye un README.md detallado en cada repositorio con:
 - **Descripción del proyecto**
 - **Instrucciones de configuración y ejecución**
 - **Ejemplos de uso**
 - **Requisitos del sistema**
6. Utiliza las herramientas de refactorización de IntelliJ IDEA para mantener un código limpio y bien estructurado.



I. E. S. POLITÉCNICO H. LANZ

GRANADA