

Ejercicio inicial

El objetivo de este ejercicio es que programes una pieza de software básico para manipular un archivo de texto. Este software deberá realizar algunas operaciones sobre el archivo.

Para realizar el ejercicio, toma en cuenta lo siguiente:

Debes contar con una cuenta de Git.

El tiempo estimado para realizar el proyecto es de 1 hora. El tiempo para llevar a cabo la instalación de las herramientas necesarias es adicional.

Tu eliges cómo trabajar.

No se restringe la consulta de información.

Deberás enviar el proyecto al correo academy-mx@datiobd.com antes del jueves 31 de enero antes de las 11:00 am.

Debes contar con estas herramientas:

IntelliJ IDE

Java 8.x 1u 161

Scala 2.11

Maven 3.3 o superior

Git

Editor de texto plano (por ejemplo Vim, Nano, Pico, SublimeText, Atom, Emacs)

Instrucciones

Fase 1. Manipulación de archivos mediante comandos Linux

Vía consola, descargar de internet el siguiente archivo: <http://samplecsvs.s3.amazonaws.com/Sacramento realestate transactions.csv>

Si aplica, descomprimirlo.

Contar el número de líneas del archivo.

Imprimir las primeras y últimas 10 líneas del archivo.

Hacer un filtrado por la palabra Multi-Family.

Crear un shell script que realice los pasos anteriores.

Fase 2. Preparación del proyecto

Crear un proyecto Maven usando Java 8 o Scala 2.11.

Importar el proyecto en un IDE (IntelliJ).

Inicializar el repositorio Git.

Hacer el clon del proyecto.

Crear un archivo de texto que contenga tu nombre.

Hacer el primer commit y después hacer push.

Copiar el shell generado anteriormente a la carpeta 'resources' del proyecto Maven.

Sincronizar los cambios hacia el repositorio remoto.

Fase 3. Programación

Realizar un programa que lea el archivo descargado, especificando la ruta como argumento del programa. El programa debe tokenizar el archivo e imprimir las líneas que contengan la palabra Multi-Family, así como imprimir la cantidad de ocurrencias de esta palabra.

Crear una rama usando Git para desarrollar el ejercicio.

Implementar un test unitario (opcional). Se recomienda usar JUnit.
Incluir loggers para validar comportamiento del programa (recomendado).
Validación de argumentos de entrada (opcional).
Imprimir al final del programa el tiempo que duró la ejecución.
Añadir al índice de Git los archivos modificados y hacer un commit de los mismos.
Generar el artefacto (JAR) asociado al proyecto.
Fase 4. Entregable Envío y retroalimentación
<p>Generar un tar.gz y entregarlo al entrevistador o bien, enviar el proyecto por email a academy-mx@datiobd.com</p> <p>En un documento, responde:</p> <p>¿Cómo realizaste el ejercicio?</p> <p>¿Qué dificultades enfrentaste y cómo las resolviste?</p>