

Práctica 1

Desarrollo e integración del software

Desarrollo Java y gestión de versiones con GIT y Github.

Enunciado de la práctica

El objetivo de esta práctica trabajar los principios básicos del lenguaje de programación Java y dar los primeros pasos en el uso de Git y la herramienta Github.

- La práctica se realizará en grupos de 4 personas.
- La práctica deberá subirse a una cuenta de Github alojada en Github Classroom asociada al grupo.
- No sólo se evaluará el resultado final sino todo el proceso hasta completarla.
- Es muy importante subir la carpeta del proyecto (incluyendo .git) a Canvas antes de la fecha límite. **No se evaluará ninguna práctica que no esté subida en Canvas, aunque el repositorio esté correctamente alojado en Github.**
- Se evaluará positivamente todo el contenido adicional a la asignatura contenido en la práctica, siempre que guarde relación y aporte valor al objetivo de esta.
- Es importante demostrar todo el conocimiento adquirido de control de versiones a la hora de gestionar el repositorio. **Gestión de ramas, merges, pull requests, etiquetas, recuperación de ficheros antiguos, etc...**

Enunciado

Dado un fichero csv con datos relativos a transacciones electrónicas, se pide crear una pequeña aplicación de Java en consola que permita leer el contenido del fichero y modelarlo mediante objetos Java. De esta forma será posible manejar la información y mostrarla en la consola del

Se pide realizar las siguientes tareas:

1. Clonar un repositorio disponible en la siguiente URL: <https://classroom.github.com/g/t22Ialt5>. Este repositorio contiene un fichero en formato csv denominado `electronic_card_transactions.csv`, además de las configuraciones básicas y mínimas para un proyecto Java. No dejéis de completar el fichero `README.md`.
2. Crear una clase o clases Java que permitan almacenar la información que se encuentra en el fichero csv. Debe contener todos los elementos necesarios para manipular la información del fichero csv clonado en el apartado anterior.
3. Leer el fichero csv e ir almacenando cada línea en su correspondiente objeto Java basado en la clase creada en el apartado anterior. Estos objetos deberán almacenarse en alguna estructura de array o listas, que permita recorrer todos los objetos almacenados.
4. Implementar un pequeño menú por consola que permita ejecutar las siguientes acciones:
 - a. Entrando un usuario una fecha en formato `mm/aaaa`, buscar en el fichero todas las operaciones de dicho período y devolver la suma del importe de las operaciones de este.
 - b. Mostrar todas las entradas con un importe superior a 1000\$ e inferiores a 10000\$ (campo `Data_Value`).
 - c. Guardar en un fichero llamado `salida.csv` los valores de las entradas cuyo campo `Series_title_2` es igual a "credit".
5. Las fechas deben mostrarse en formato español: `mm/aaaa`.

Toda la práctica debe estar correctamente almacenada en el repositorio de código que se os ha facilitado. Debéis seguir las pautas de trabajo con Git y Github que hemos visto en clase y crear las ramas que sean necesarias para gestionar el proyecto de la forma más profesional posible.

A continuación, disponéis de una tabla con la definición de los campos del fichero csv (aquellos con descripción "No aplica" no se deben tener en cuenta):

Nombre del campo	Tipo de datos	Descripción
Series_reference	String	Nombre de referencia de la serie de valores.
Period	Date	Mes en el que se agrupan las operaciones.
Data_value	Float	Valor de la transacción.
Supressed	Boolean	No definido.

Status	Character	Estado en el que se encuentra la transacción.
Units	String	Moneda.
Subject	String	Tipo de transacción.
Group	String	Definición del grupo al que pertenece la agrupación.
Series_title_1	String	Vencimiento de la transacción
Series_title_2	String	Sector en el que se ha realizado la transacción.
Series_title_3	String	No definido.
Series_title_4	String	No definido.
Series_title_5	String	No definido.

Memoria

Es necesario generar una memoria en la que recogeréis los distintos pasos que habéis realizado para resolver la práctica.

La memoria debe estar correctamente redactada, sin faltas de ortografía y con una estructura correcta (portada, índice, índice de figuras, tablas, numeración de las páginas, etc.) y contener puntos como la introducción y las conclusiones, entre otros.

No debéis introducir el código fuente en esta memoria, puesto que dispondremos de él para poder verlo. Si hay algún fragmento de código que se considere importante para el desarrollo de vuestra solución, sí podéis añadirlo y explicar y comentar el porqué es importante ese fragmento.

Debéis explicar cómo habéis afrontado la solución de la práctica, cómo se ha dividido el trabajo y cómo se ha gestionado el equipo para el uso de Git.

Podéis adjuntar imágenes sobre la gestión del repositorio, que pueden incluir desde capturas de la consola en la que deseáis mostrar algún comando o comportamiento particular o relevante que os haya surgido; a capturas del árbol de ramas generado por git. Siempre como complemento y no como contenido principal.

Entregables

Los entregables que debéis comprimir y subir a la tarea de Canvas son los siguientes:

- Carpeta del proyecto, incluyendo la carpeta .git.
- La aplicación Java con la estructura de carpetas y ficheros descriptores del proyecto que permita ejecución de la misma desde el IDE para comprobar su correcto funcionamiento.
- Los ficheros csv generados mediante vuestra aplicación de consola.
- La memoria de la práctica en formato **.PDF**.

El nombre del fichero comprimido debe ser “grupo[X]_P1”. Se admiten formatos .rar y .zip.

La fecha tope de entrega de esta práctica es el día 4 de octubre a las 23:59.

Rúbrica de evaluación

La siguiente rúbrica será para evaluar la práctica anteriormente descrita. El desarrollo/código y la memoria tienen el mismo peso. 50% del total de la práctica.

Git

DIMENSIÓN	VALORACIÓN			
	0 puntos			10 puntos
Utiliza las funcionalidades de Git de forma correcta	No existe control de versiones	Control de versiones excesivamente sencillo	Utiliza las funcionalidades justas de forma correcta.	Utiliza todas las funcionalidades estudiadas de forma correcta.
El repositorio está ordenado, es claro, y tiene las configuraciones básicas. Readme, .gitignore, etc...	Repositorio con configuraciones por defecto	Configuraciones sencillas y no completas.		Todas las configuraciones del repositorio realizadas correctamente.
Resuelve problemas de cierta complejidad.	Gestión de versiones básica	Resuelve algún conflicto o situación compleja en el repositorio	Soluciona conflictos de cierta complejidad, aunque de formas rudimentarias.	Soluciona correcta y limpiamente soluciones complejas en el repositorio.

Java

DIMENSIÓN	VALORACIÓN			
	0 puntos			10 puntos
Sigue las especificaciones del enunciado para la creación de la aplicación Java.	No lee el fichero ni define una clase capaz de almacenar la información contenida en este. Los objetos no son almacenados en una estructura de listas o arrays, como se especifica. No implementa las acciones solicitadas en la aplicación por consola. El formato de fechas	Lee el fichero, pero no es capaz de almacenarlo correctamente en las clases java. La clase Java que almacena la información tiene deficiencias y no coinciden los tipos con los especificados en el enunciado. Resuelve las acciones de la aplicación en consola de forma deficiente o errónea. Las fechas	Lee el fichero y es capaz de almacenar la información leída del mismo en una clase java correctamente formada, aunque con algunas deficiencias. Los tipos de datos son adecuados, aunque pueden no concordar con la especificación. Resuelve la aplicación solicitada, aunque no muestra correctamente toda	Lee el fichero y es capaz de almacenar la información leída en una clase correctamente formada y definida. Los tipos de datos concuerdan perfectamente. Resuelve la aplicación solicitada mostrando toda la información pedida y generando el fichero de salida con el contenido correcto. Las fechas están

	no es correcto o no están tratadas. No existe un fichero de salida con la información pedida.	están bien tratadas, pero no se muestran correctamente.	la información solicitada. Las fechas están bien tratadas y se muestran correctamente. El contenido del fichero de salida no es completamente correcto.	perfectamente tratadas.
Hace un uso adecuado de las herramientas de Java vistas en clase y aplica los conocimientos aprendidos correctamente.	No define las clases ni usa correctamente las herramientas vistas en clase para la generación de esta. Los métodos para la lectura del fichero no son los adecuados. No usa las herramientas correctas para la generación del fichero de salida ni las librerías adecuadas para la gestión de las fechas. El menú no se genera usando métodos que permitan ampliar la información de forma correcta.	Define las clases java en base a las directrices vistas en clase, pero no están completas. Los métodos para la lectura del fichero no se corresponden con los vistos en clase, pero funcionan de manera parcial. El código generado para la aplicación de consola es poco cuidado y no se ciñe a las buenas prácticas disponibles en el aula virtual. No se gestionan las fechas correctamente al no trabajar con los métodos de la librería adecuados.	Se definen las clases y los métodos acorde a lo estudiado durante el curso. Los métodos de lectura son correctos, aunque no se correspondan directamente con lo visto en clase. El código se ciñe en gran medida a las buenas prácticas de Java disponibles en el aula virtual, aunque no las cumple completamente. Las librerías usadas para la gestión de fechas son adecuadas, aunque el resultado no es el esperado	Todas las clases y métodos se definen acorde a lo visto en clase durante el curso. Se siguen las buenas prácticas de código disponibles en el curso virtual y se usan las librerías y paquetes adecuados para la lectura y escritura de los ficheros solicitados. El menú se genera usando métodos que permiten la reutilización de código y la ampliación de la información de manera correcta. La gestión de fechas es correcta.

Memoria

DIMENSIÓN	VALORACIÓN			
	0 puntos			10 puntos
Redacta adecuadamente la memoria del proyecto	La memoria contiene numerosas faltas de ortografía o gramaticales, con una expresión poco formal, que dificulta su entendimiento	La memoria no expresa con un lenguaje propio del ámbito de conocimiento los conceptos clave	La memoria expresa con lenguaje propio del ámbito de conocimiento los conceptos clave, pero contiene algunos errores ortográficos y gramaticales	La memoria está libre de errores ortográficos o gramaticales y expresa con lenguaje propio del ámbito de conocimiento los conceptos
Presenta una memoria con un formato adecuado	El formato está muy poco cuidado (tamaños	El formato está suficientemente cuidado	El formato está suficientemente cuidado	El formato está cuidado (con numeración de

	inadecuados, no justifica párrafos, no encabezados ni pies de página) y no incluye todas las secciones básicas de un trabajo académico (portada, índice, desarrollo, conclusiones y bibliografía)	(numeración de páginas, tamaño de letras y justificación de párrafos), pero no incluye todas las secciones básicas de un trabajo académico	(numeración de páginas, tamaño de letras y justificación de párrafos), incluye todas las secciones básicas de un trabajo académico (portada, índice, desarrollo, conclusiones y bibliografía) pero no expresa ni usa adecuadamente la bibliografía	páginas, encabezados, tamaños adecuados de letra, justificación de párrafos y pies de figuras), incluye todas las secciones básicas de un trabajo académico (portada, índice, desarrollo, conclusiones y bibliografía) y la bibliografía está adecuadamente expresada y referenciada en el texto
Detalle del contenido de la memoria	El contenido de la memoria es el visto en clase	La memoria contiene el contenido visto en clase, complementado con diagramas UML, y otras secciones investigadas por el alumno.	La memoria contiene toda la información anterior además de ventajas/desventajas de los elementos estudiados y/o otras secciones relevantes	La memoria contiene todo lo anterior, además de conclusiones por cada sección estudiadas y probadas.
Presenta una memoria cuyos contenidos son correctos y coherentes con el proyecto desarrollado	Los diagramas de flujo y las explicaciones no son coherentes con el código.	Existen incoherencias graves entre el código y el contenido de la memoria.	Existen incoherencias entre el código y el contenido de la memoria.	El código y la memoria están perfectamente alineados.
Introduce, explica y concluye adecuadamente y con rigor académico	La introducción no permite contextualizar adecuadamente el trabajo, no está bien explicado el diseño y desarrollo del proyecto y las conclusiones no son relevantes	La introducción no permite contextualizar adecuadamente el trabajo y las conclusiones no son relevantes, pero está bien explicado el diseño y desarrollo del proyecto	La introducción y las explicaciones son relevantes, y están bien argumentadas y ajustadas al proyecto y al ámbito de conocimiento, pero no hay conclusiones o no son relevantes	La introducción, las explicaciones y las conclusiones son relevantes, y están bien argumentadas y ajustadas al proyecto y al ámbito de conocimiento