

DOCUMENTACIÓN TAREA PROGRAMADA - EIFFEL
JSON

Jose Pablo Munoz - 2019061904

1. Dar las instrucciones para ejecutar el programa

El programa cuenta con una clase principal llamada ACTION_LISTENER que se encarga de leer las operaciones ingresadas por el usuario. El programa ya viene preparado para correr al ejecutarse, por lo que la clase APPLICATION contiene una instancia de la clase ACTION_LISTENER, y en su make la crea y llama a la funcion run, la cual inicia el ciclo encargado de leer las operaciones. En la interfaz de la terminal se le pide al usuario que ingrese alguna operación. Estas deben ingresarse tal y como se solicitaba en las instrucciones del proyecto, excepto por los casos aclarados en el siguiente punto. El programa es capaz de notificar al usuario si ingreso una operación incorrectamente. Cuando se completa una operación el programa tambien notifica al usuario sobre el resultado y solicita la siguiente operación.

2. Corridas de prueba

Para cada uno de las pruebas siguientes, indicar si se obtuvo o no el resultado correcto. En caso de implementación parcial, se puede mostrar la ejecución de aquellos aspectos del programa que funcionen correctamente. Los archivos usados son *Equipos.csv* e *Integrantes.csv* que les fue dado. El archivo *Productos.csv* se incluye con este documento.

Pruebas	Comentario sobre resultado obtenido
>> load eq Equipos.csv	Éxito
>> save eq eq.json	Éxito
>> select eq eq2 alojamiento = Edificio A1	Éxito, unicamente es necesario que los valores (los de la derecha del =) que sean de tipo string se ingresen entre comillas ("") Ej. alojamiento = "Edificio A1"
>> savecsv eq2 eq.csv	Éxito
>> project eq eq3 equipo pais	Éxito
>> save eq3 eq3.json	Éxito
>> load int Integrantes.csv	Éxito
>> select int int2 primeraVez = true	Éxito, se cambio el nombre del atributo a: primera
>> project int2 int3 nombre apellidos fechaNac	Éxito
>> save int3 int3.json	Éxito
>> savecsv int3 int3.csv	Éxito

>> load prod Productos.csv	Éxito
>> select prod prod2 linea = Ships	Éxito, Ships debe ir entre comillas
>> project prod2 prod3 codigo nombre precio	Éxito
>> save prod3 prod3.json	Éxito
>> save prod3 prod3.csv	Éxito, la operación se cambio por: savecsv
>> exit	Éxito

Entregue con la documentación los archivos generados: **eq.json**, **eq.csv**, **eq3.json**, **int3.json**, **int3.csv**, **prod3.json**, **prod3.csv**.

3. Comentarios finales (estado del programa)

Indicar el estado final en que quedó el programa, problemas encontrados y limitaciones adicionales.

El programa fue realizado con éxito y cumple con la totalidad de las funcionalidades requeridas. Es decir, el programa es capaz de realizar todas las operaciones correctamente y cumplir con todas las pruebas. Algunas limitaciones encontradas son: Primero, que no hay suficiente seguridad para dar estabilidad al programa en caso de que no se encuentre un archivo de lectura para cargar datos. Segundo, no se logro imprimir correctamente las tildes en la interfaz de la terminal, sin embargo, si son mostradas correctamente en los archivos de texto. Por ultimo, el programa trabaja unicamente con dos digitos despues del punto decimal. Por ejemplo, el programa si es capaz de almacenar numeros con muchos digitos decimales, pero a la hora de guardarlos en archivos o hacer comparaciones con "select" solo se utilizan los primeros dos digitos. Cabe resaltar que las comparaciones entre numeros decimales deben hacerse con coma en vez de punto.