Programa de gestión de una Base Aérea Militar

**AirBase**

Parte 3

Ficheros y Log

Constantino Lapaz Giménez

Juan Mauricio González Arenas

Rafael Antonio Vázquez Flores

DAM Sector

Diagrama

Descripción generada automáticamente

Índice

[Punto de partida 1](#_Toc163422091)

[Trello 2](#_Toc163422092)

[Estructura del Proyecto 3](#_Toc163422093)

[Gestión Piloto 3](#_Toc163422094)

[Modificaciones 5](#_Toc163422095)

[Para el resto de las clases 6](#_Toc163422096)

[Gestión Ruta 6](#_Toc163422097)

[Gestión Incidencias 8](#_Toc163422098)

[Fichero .log 9](#_Toc163422099)

[Resumen 11](#_Toc163422100)

[Nota final 11](#_Toc163422101)

# Punto de partida

La tarea consiste en añadir ficheros y uso de cadenas de texto al proyecto:

* Uso de ficheros para leer datos: Los datos deben estar estructurados en formato csv, esto es, cada campo separado de otro campo por “;” y en cada línea un registro. Se valorará positivamente el uso de distintas funciones de cadenas de textos (split, indexOf, contains, substring, sprit, toLowerCase, toUpperCase, etc.)
* Uso de ficheros para escribir un log de toda vuestra aplicación: Cada opción de menú, cada método, cada excepción controlada, deben escribir un registro en un archivo .log. Para ello el programa cuando se ejecuta debe crear una carpeta logs dentro de vuestro proyecto, borrando la carpeta en caso de que exista (carpeta y archivos). Una vez creada la carpeta se creará un archivo para los logs con extensión .log en el que se irán incluyendo los registros de log. Se valorará positivamente la creación de varios archivos por día, por tipo de registro de log, cuando se supere un tamaño, etc...
* Se debe crear un archivo de datos para cada uno de los módulos realizados en la aplicación de forma que toda la información de la aplicación esté en los archivos y no directamente en el código.

# Trello

Hemos actualizado Trello para esta nueva entrega realizando así las modificaciones pertinentes para este trimestre.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# Estructura del Proyecto

Texto

Descripción generada automáticamenteGeneramos diferentes ficheros en cada uno de los métodos concretos, en este caso son:

## Texto Descripción generada automáticamenteGestión Piloto

Texto

Descripción generada automáticamenteCreamos la variable ***rutaArchivo*** y la declaramos como publica ya que serán varias clases la que accederán a ella. Cada clase modificara la ruta para su conveniencia, pero todas serán el mismo Directorio “ficheros”

Generamos el archivo mediante esta función, capturamos una excepción para luego (más tarde lo detallare) integrar ese error en un archivo log.

Texto

Descripción generada automáticamente

Escribimos en el ArrayList con el formato del fichero indicando que se salte la primera línea que usaremos como cabecera.

Después indicamos que cada columna está separada por “,” y a que pertenece cada ***campo***

Para terminar, mostramos lo escrito e indicamos que lo hemos leído correctamente, cualquier error lo capturamos con el catch y lo volcamos como mensaje en el archivo log.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Para Actualizar el archivo, solo tenemos que escribir en los nuevos campos que queremos añadir directamente, para eso generamos en un **for** los GET que tenga nuestra Clase

Nuevamente volvemos a capturar la excepción para el archivo log.

# Modificaciones

Texto

Descripción generada automáticamente

Como podemos ver algunas funciones antiguas, ya no son necesarias por lo que modificamos y actualizarlos para que ahora se haga todo desde el fichero y no desde el ***arrayList***

Texto

Descripción generada automáticamente

Ahora debemos llamar en el resto de funciones a nuestro método ***actualizarArchivoCSV()*** de esta manera siempre que generemos un dato clave para nuestro fichero, estará agregado correctamente.

# Para el resto de las clases

Hemos usado el mismo método y funcionalidad cambiando las partes claves que involucran dichas clases.

Texto

Descripción generada automáticamenteImagen que contiene Texto

Descripción generada automáticamenteGestión Ruta

Texto

Descripción generada automáticamente

## Texto Descripción generada automáticamenteTexto Descripción generada automáticamenteTexto Descripción generada automáticamenteGestión Incidencias

Captura de pantalla de computadora

Descripción generada automáticamente

# Fichero .log

Texto

Descripción generada automáticamenteSon fichero en los que podemos ver características o propiedades de nuestro programa, nosotros hemos generado un .log donde vemos los errores que saltan en nuestro programa.

Texto

Descripción generada automáticamente

Como se puede ver muy bien en el ejemplo, cada error añade su línea de código al fichero especificándonos donde se cometió el error, la fecha de cuando sucedió y a que hora agregando minutos segundos y milésimas de segundo.

# Resumen

Gracias a las Buenas Practicas no nos fue muy complicado añadir y modificar esta parte.

# Nota final

Si bien los ***arrayList*** nos permiten gestionar los datos de nuestro programa, estos no han de ser inicializado si queremos trabajar con ellos, sin posibilidad de guardarlos de ninguna manera. Solventamos con los ficheros este gran problema, pudiendo acceder a ellos una vez el programa cierre su ejecución