



Programa de gestión de
una Base Aérea Militar

AirBase

Parte 4

SonarInk

Constantino Lapaz Giménez

Juan Mauricio González Arenas

Rafael Antonio Vázquez Flores

DAM Sector

Índice

Punto de partida	1
Trello	2
Errores del SonarInk	3
Error Escogido por Mauricio	5
Explicación	6
Resultado:	6
Error Escogido por Constantino	7
Explicación	7
Resultado:	8
Error Escogido por FaLy	9
Resumen y nota final.	12
Resultado final de SonarInk	13

Punto de partida

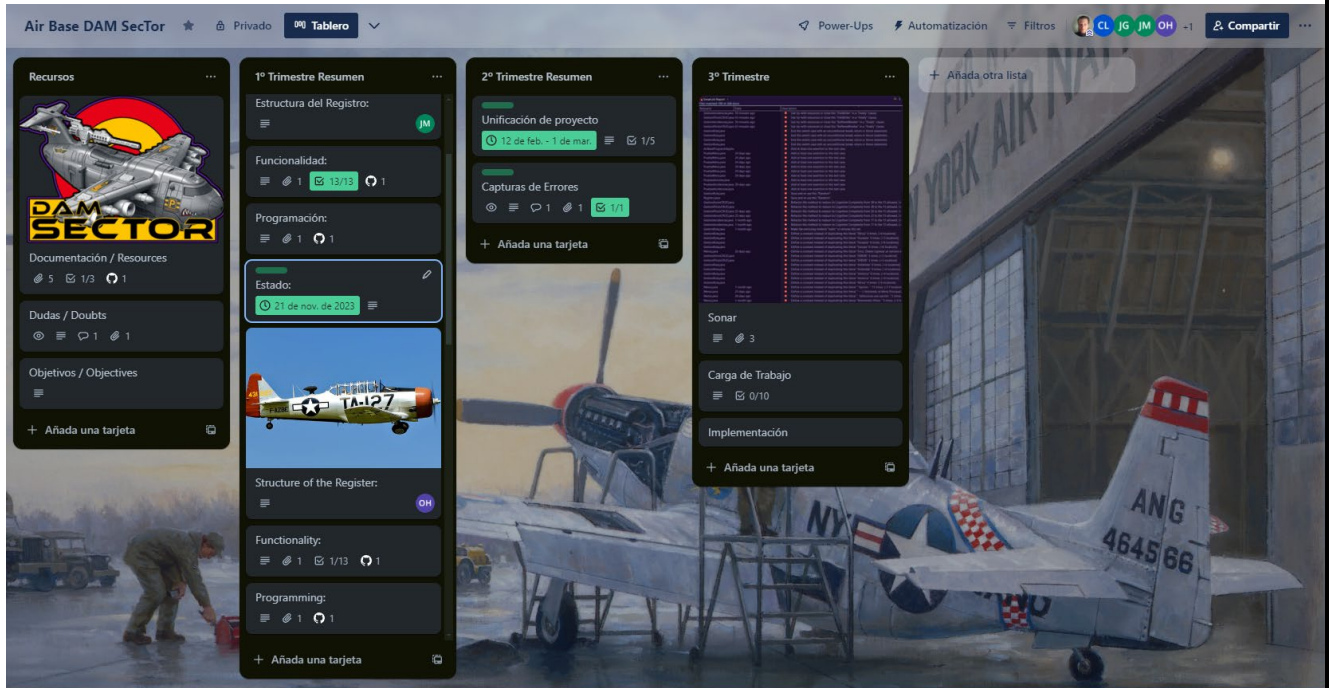
Debéis instalar la extensión SonarInt para Eclipse y obtener un informe de todo el proyecto de los errores más destacados y comunes que os muestra Sonar.

Podéis mostrar los errores de cada clase, o de todo el proyecto, mostrar un listado de errores y cuantas veces os aparece. Por otro lado, debéis buscar los errores más complejos (que no sea el `System.out.println`) y mostrarme un ejemplo del error y cómo se solucionaría con una refactorización.

Para cada uno de los integrantes del proyecto debéis decirme un error y una solución.

Trello

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



Errores del SonarInk


SonarLint Report		
Filter matched 100 of 208 items		
Resource	Date	Description
GestionIncidencias.java	50 minutes ago	Use try-with-resources or close this "FileWriter" in a "finally" clause.
GestionPilotoCRUD.java	53 minutes ago	Use try-with-resources or close this "FileWriter" in a "finally" clause.
GestionIncidencias.java	50 minutes ago	Use try-with-resources or close this "BufferedReader" in a "finally" clause.
GestionPilotoCRUD.java	53 minutes ago	Use try-with-resources or close this "BufferedReader" in a "finally" clause.
GestionRuta.java		End this switch case with an unconditional break, return or throw statement.
GestionRuta.java		End this switch case with an unconditional break, return or throw statement.
GestionRuta.java		End this switch case with an unconditional break, return or throw statement.
GestionRuta.java		End this switch case with an unconditional break, return or throw statement.
AirBaseProgramliApplic		Add at least one assertion to this test case.
PruebaMenu.java	24 days ago	Add at least one assertion to this test case.
PruebaMenu.java	24 days ago	Add at least one assertion to this test case.
PruebaMenu.java	24 days ago	Add at least one assertion to this test case.
PruebaMenu.java	24 days ago	Add at least one assertion to this test case.
PruebaMenu.java	24 days ago	Add at least one assertion to this test case.
PruebaMenu.java	24 days ago	Add at least one assertion to this test case.
PruebasAviones.java		Add at least one assertion to this test case.
PruebasIncidencias.java	29 days ago	Add at least one assertion to this test case.
PruebasIncidencias.java		Add at least one assertion to this test case.
GestionRuta.java		Save and re-use this "Random".
Registro.java		Save and re-use this "Random".
GestionAvionCRUD.java		Refactor this method to reduce its Cognitive Complexity from 38 to the 15 allowed. [+]
GestionPilotoCRUD.java		Refactor this method to reduce its Cognitive Complexity from 38 to the 15 allowed. [+]
GestionPilotoCRUD.java	25 days ago	Refactor this method to reduce its Cognitive Complexity from 26 to the 15 allowed. [+]
GestionAvionCRUD.java	25 days ago	Refactor this method to reduce its Cognitive Complexity from 22 to the 15 allowed. [+]
GestionIncidencias.java	1 month ago	Refactor this method to reduce its Cognitive Complexity from 17 to the 15 allowed. [+]
GestionIncidencias.java	1 month ago	Refactor this method to reduce its Cognitive Complexity from 17 to the 15 allowed. [+]
GestionRuta.java	1 month ago	Make the enclosing method "static" or remove this set.
GestionRuta.java		Define a constant instead of duplicating this literal "África" 6 times. [+6 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "Oceania" 5 times. [+5 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "Oceania" 6 times. [+6 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "Europa" 6 times. [+6 locations]
Menus.java	25 days ago	Define a constant instead of duplicating this literal "Error: Debes ingresar un número e
GestionAvionCRUD.java		Define a constant instead of duplicating this literal "ERROR" 3 times. [+3 locations]
GestionPilotoCRUD.java		Define a constant instead of duplicating this literal "ERROR" 3 times. [+3 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "Antártida" 6 times. [+6 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "Antártida" 6 times. [+6 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "América" 6 times. [+6 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "América" 6 times. [+6 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "America" 6 times. [+6 locations]
GestionRuta.java		Define a constant instead of duplicating this literal "África" 6 times. [+6 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " Opción : " 11 times. [+11 location]
Menus.java	25 days ago	Define a constant instead of duplicating this literal " ---{ Volviendo al Menú Principal.}
Menus.java	26 days ago	Define a constant instead of duplicating this literal " Seleccione una opción: " 5 times.
Menus.java	1 month ago	Define a constant instead of duplicating this literal "Bienvenido Piloto " 5 times. [+5 lo

[illegible]

File Name	Last Commit	Issue Description
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionRuta.java		Remove this useless assignment to local variable "horas".
GestionIncidencias.java	1 month ago	Remove this useless assignment to local variable "descrip".
GestionIncidencias.java		Remove this useless assignment to local variable "descrip".
GestionAvionCRUD.java		Remove this unused method parameter "sc".
GestionPilotoCRUD.java		Remove this unused method parameter "sc".
GestionRuta.java	25 days ago	Remove this unused method parameter "h".
Menus.java	26 days ago	Remove this unused method parameter "entidad".

Error Escogido por Mauricio

GestionPilotoCRUD.java

 Refactor this method to reduce its Cognitive Complexity from 35 to the 15 allowed. [+15 locations]

```

public String elegirCarga(boolean carga, Scanner sc, String[][] t) {
    /*En esta funcion entra como parametro un booleano, Scanner sc y una matriz t.
    boolean carga:
        TRUE-->Provisiones-->ARMAMENTO MUNICION EXPLOSIVOS
        FALSE-->Suministro-->TROPAS VEHICULOS GASOLINA
    String [][] t:
        Se recorre la matriz para verificar que la carga es correcta, si no es asi se pide de nuevo
    Devuelve el tipo de carga que va a transportar
    */
    boolean verificar=true;
    String texto;
    +1
    1 do {
        +2 (incl 1 for nesting)
        2 if (carga ==true) {
            System.out.println(" ¿Que tipo de provision va a transportar?");
            System.out.println("      [Armamento, Munición, Explosivos]");
            texto=sc.next();
            String c=texto;
            +3 (incl 2 for nesting)
            3 for (int i = 0; i<t.length;i++) {
                +4 (incl 3 for nesting)
                4 for(int j =0;j<t[i].length;j++) {
                    +5 (incl 4 for nesting) | +1
                    5 if(c.equals("Armamento") 6 ||c.equals("Municion")||c.equals("Explosivos")) {
                        verificar=true;
                    }
                    +1
                    7 else {
                        verificar = false;
                    }
                }
            }
            +3 (incl 2 for nesting)
            8 if(verificar ==false) {
                escribirError("Error en elegir tipo de provision");
                System.out.println("ERROR");
            }
            +1
            9 else {
                escribirError("Tipo de provision elegido correctamente");
            }
        }
        +1
        10 else {
            System.out.println(" ¿Que tipo de suministro va a transportar?");
            System.out.println("      [Tropas, Vehiculos, Gasolina]");
            texto=sc.next();
            String c=texto;
            +3 (incl 2 for nesting)
            11 for (int i = 0; i<t.length;i++) {
                +4 (incl 3 for nesting)
                12 for(int j =0;j<t[i].length;j++) {
                    +5 (incl 4 for nesting) | +1
                    13 if(c.equals("Tropas") 14 ||c.equals("Vehiculos")||c.equals("Gasolina")) {
                        verificar=true;
                    }
                    +1
                    15 else {
                        verificar = false;
                    }
                }
            }
            +3 (incl 2 for nesting)
            16 if(verificar ==false) {
                escribirError("Error en elegir tipo de o suministro");
                System.out.println("ERROR");
            }
            +1
            17 else {
                escribirError("Tipo de y suministro elegido correctamente");
            }
        }
    }
    while(verificar==false);
    return texto;
}

```

7-4-2024

Explicación

- Eliminación de innecesaria igualación a 'true' de la variable booleana verificar antes del bucle 'do-while'.
- Simplificamos la entrada de texto del usuario para verificar la carga. Hacemos primero el 'if'.
- Uso de un solo método **escribirError()** para mostrar mensajes de error.
- Refactorizar el bucle anidado, es decir, usar 'for each' que recorre mejor la matriz bidimensional.
- Dependiendo de la carga igualamos el texto a lo que debería ser y lo devolvemos con la variable verificar.

Resultado:

```
boolean verificar;
String texto;
do {
    verificar = true;
    if (carga) {
        System.out.println(" ¿Qué tipo de provisiones va a transportar?");
        System.out.println("      [Armamento, Munición, Explosivos]");
    } else {
        System.out.println(" ¿Qué tipo de suministro va a transportar?");
        System.out.println("      [Tropas, Vehículos, Gasolina]");
    }
    texto = sc.next();
    for (String[] strings : t) {
        for (String s : strings) {
            if (carga) {
                if (!texto.equals("Armamento") && !texto.equals("Munición") && !texto.equals("Explosivos")) {
                    verificar = false;
                }
            } else {
                if (!texto.equals("Tropas") && !texto.equals("Vehículos") && !texto.equals("Gasolina")) {
                    verificar = false;
                }
            }
        }
    }
} while (!verificar);
return texto;
}
```


Error Escogido por Constantino

```
GestionIncidencias.java 1 month ago Refactor this method to reduce its Cognitive Complexity from 17 to the 15 allowed. [+12 locations]

public void incidenciaAvion(ArrayList<Avion> aviones) {
    boolean ok = false;
    int seleccion = 0;
    String numIncidencia = "";
    String codAv = "";
    String descrip = "";
    String op = "";
    System.out.println("Introduce el numero de incidencia");
    numIncidencia = dato.next();
    while(ok==false) {
        System.out.println("Introduce el código del avión");
        codAv = dato.next();
        for(int i = 0;i<aviones.size() && ok == false;i++) {
            if(codAv.equals(aviones.get(i).getCodigo())) {
                seleccion = i;
                ok = true;
            } else {ok = false;}
        }
        if(ok == false) {
            System.out.println("Ningún avión encontrado");
        }
    }
    System.out.println("Avión seleccionado: " +aviones.get(seleccion).getNombre());
    System.out.println("Introduzca una descripción de la incidencia");
    descrip = dato.nextLine();
    descrip = dato.nextLine();
    while (ok == true) {
        System.out.println("¿Apto para el servicio?");
        op = dato.next();
        if(op.charAt(0) == 'n' || op.charAt(0) == 'N') {
            aviones.get(seleccion).setOperativo(false);
            ok = false;
        } else if(op.charAt(0) == 's' || op.charAt(0) == 'S') {
            aviones.get(seleccion).setOperativo(true);
            ok = false;
        } else {
            System.out.println("Error, respuesta no valida");
            ok = true;
        }
    }
    incidencia.add(new Incidencias(numIncidencia,codAv,descrip,op));
    actualizarArchivoCSV();
}
```

Explicación

- Se ha utilizado un objeto Scanner para leer la entrada del usuario.
- Se ha utilizado una variable **avionSeleccionado** para almacenar el avión seleccionado por el usuario.
- Se ha utilizado un bucle **while** para buscar el avión por código.
- Se ha agregado un mensaje de error para cuando no se encuentra el avión.

- Se ha utilizado un bucle do-while para validar la entrada del usuario para la aptitud para el servicio.
- Se ha utilizado el operador ternario para asignar el valor de aptoParaVuelo a la cadena "S" o "N" según el valor booleano.
- Se ha agregado el código para crear la nueva instancia de Incidencias y agregarlo a la lista incidencias.

Resultado:

```
public void incidenciaAvion(ArrayList<Avion> aviones) {
    try (Scanner datos = new Scanner(System.in)) {
        System.out.print("Introduce el número de incidencia: ");
        String numeroIncidencia = datos.nextLine();

        Avion avionSeleccionado = null;
        while (avionSeleccionado == null) {
            System.out.print("Introduce el código del avión: ");
            String codigoAvion = datos.nextLine();

            for (int i = 0; i < aviones.size(); i++) {
                if (codigoAvion.equals(aviones.get(i).getCodigo())) {
                    avionSeleccionado = aviones.get(i);
                    break;
                }
            }

            if (avionSeleccionado == null) {
                System.out.println("Ningún avión encontrado.");
            }
        }

        System.out.println("Avión seleccionado: " + avionSeleccionado.getNombre());

        System.out.print("Introduzca una descripción de la incidencia: ");
        String descripcion = datos.nextLine();

        boolean aptoParaVuelo;
        do {
            System.out.print("¿Apto para el servicio? (S/N): ");
            String opcion = datos.nextLine().toUpperCase();

            if (opcion.equals("S")) {
                aptoParaVuelo = true;
                break;
            } else if (opcion.equals("N")) {
                aptoParaVuelo = false;
                break;
            } else {
                System.out.println("Error, respuesta no válida.");
            }
        } while (true);

        avionSeleccionado.setOperativo(aptosParaVuelo);

        Incidencias.add(new Incidencias(numeroIncidencia, avionSeleccionado.getCodigo(), descripcion, aptoParaVuelo ? "S" : "N"));
    }

    actualizarArchivoCSV();
}
```

Error Escogido por FaLy

Refactorizar la Clase GestionRutas para que tenga menos errores.

```
public static int horasVuelo(String a, String b, int h ) {
    /*Esta funcion va a calcular aproximadamente la duracion de
    la ruta dependiendo del continente
    origen y el continente del destino.*/
    String origen = a;
    String destino = b;
    int horas=0;
    switch(origen) {
    case "Europa":
        //Europa--->Africa = [6,10]
        //Europa--->Asia = [4,12]
        //Europa--->Oceania = [10,14]
        //Europa--->America = [8,13]
        //Europa--->Antartida = [5,9]
        switch(destino) {
        case "Africa", "África":
            horas=generarRandom(6,10);
            break;
        case "Asia":
            horas=generarRandom(4,12);
            break;
        case "Oceania", "Oceanía":
            horas=generarRandom(10,14);
            break;
        case "America", "América":
            horas=generarRandom(8,13);
            break;
        case "Antartida", "Antártida":
            horas=generarRandom(5,9);
            break;
        default:
            horas = 4;
        }
        break;
    case "Africa", "África":
        //Africa--->Europa = [6,10]
        //Africa--->Asia = [3,11]
        //Africa--->Oceania = [7,12]
        //Africa--->America = [6,13]
        //Africa--->Antartida = [8,12]
        switch(destino) {
        case "Europa":
            horas=generarRandom(6,10);
            break;
        case "Asia":
            horas=generarRandom(3,11);
            break;
        case "Oceania", "Oceanía":
            horas=generarRandom(7,12);
            break;
        case "America", "América":
            horas=generarRandom(6,13);
            break;
        case "Antartida", "Antártida":
```

```

        horas=generarRandom(8,12);
        break;
    default:
        horas = 4;
    }
break;
case "Asia":
    //Asia--->Africa = [3,11]
    //Asia--->Europa = [4,12]
    //Asia--->Oceania = [2,8]
    //Asia--->America = [3,9]
    //Asia--->Antartida = [4,8]
    switch(destino) {
    case "Africa", "África":
        horas=generarRandom(3,11);
        break;
    case "Europa":
        horas=generarRandom(4,12);
        break;
    case "Oceania":
        horas=generarRandom(2,8);
        break;
    case "America", "América":
        horas=generarRandom(3,9);
        break;
    case "Antartida", "Antártida":
        horas=generarRandom(4,8);
        break;
    default:
        horas = 4;
    }
    break;
case "Oceania", "Oceanía":
    //Oceania--->Africa = [7,12]
    //Oceania--->Asia = [2,8]
    //Oceania--->Europa = [10,14]
    //Oceania--->America = [4,10]
    //Oceania--->Antartida = [6,12]
    switch(destino) {
    case "Africa", "África":
        horas=generarRandom(7,12);
        break;
    case "Asia":
        horas=generarRandom(2,8);
        break;
    case "Europa":
        horas=generarRandom(10,14);
        break;
    case "America", "América":
        horas=generarRandom(4,10);
        break;
    case "Antartida", "Antártida":
        horas=generarRandom(6,12);
        break;
    default:
        horas = 4;
    }
    break;
case "America", "América":
    //America--->Africa = [6,13]

```

```

        //America--->Asia = [3,9]
        //America--->Oceania = [4,10]
        //America--->Europa = [8,13]
        //America--->Antartida = [2,12]
        switch(destino) {
        case "Africa", "África":
            horas=generarRandom(6,13);
            break;
        case "Asia":
            horas=generarRandom(3,9);
            break;
        case "Oceania", "Oceanía":
            horas=generarRandom(4,10);
            break;
        case "Europa":
            horas=generarRandom(8,13);
            break;
        case "Antartida", "Antártida":
            horas=generarRandom(2,12);
            break;
        default:
            horas =4;
        }
        break;
    case "Antartida", "Antártida":
        //Antartida--->Africa = [8,12]
        //Antartida--->Asia = [4,8]
        //Antartida--->Oceania = [6,12]
        //Antartida--->America = [2,12]
        //Antartida--->Europa = [5,9]
        switch(destino) {
        case "Africa", "África":
            horas=generarRandom(8,12);
            break;
        case "Asia":
            horas=generarRandom(4,8);
            break;
        case "Oceania", "Oceanía":
            horas=generarRandom(6,12);
            break;
        case "America", "América":
            horas=generarRandom(2,12);
            break;
        case "Europa":
            horas=generarRandom(5,9);
            break;
        default:
            horas = 4;
        }
        break;
    }

    return horas;
}

```

- La definición del método **horasRuta** se ha modificado para crear objetos Ruta y agregarlos al ArrayList.
- El método **horasVuelo** ahora recorre el ArrayList de rutas en lugar del Map.
- Se ha añadido una clase Ruta simple con getters y setters para los atributos origen, destino y **tiempoVuelo**.

```
public static int horasVuelo(String origen, String destino, int horas) {
    final int estandar = 4;
    ArrayList<Ruta> rutas = horasRuta();

    horas = estandar;

    for (Ruta ruta : rutas) {
        if (ruta.getPaisOrigen().equalsIgnoreCase(origen) && ruta.getPaisDestino().equalsIgnoreCase(destino)) {
            horas = ruta.getH();
            break;
        }
    }
    return horas;
}

private static ArrayList<Ruta> horasRuta() {
    ArrayList<Ruta> rutas = new ArrayList<>();
    try (BufferedReader br = new BufferedReader(new FileReader(tiempoVuelo))) {
        String linea;
        while ((linea = br.readLine()) != null) {
            String[] datos = linea.split(",");
            String origen = datos[0].toUpperCase();
            String destino = datos[1].toUpperCase();
            int tiempoVuelo = Integer.parseInt(datos[2]);

            Ruta ruta = new Ruta(origen, destino, tiempoVuelo);
            rutas.add(ruta);
        }
    } catch (IOException e) {
        System.err.println("Error al cargar datos de tiempos de vuelo: " + e.getMessage());
    }

    return rutas;
}
```

Resumen y nota final.

Las complejidades de modificar un archivo tan Grande son evidentes cuando corregir un error implica modificar casi todas las clases involucradas. Por eso creo que es mejor empezar con buenas practicas y con el sonarInk desde 1º hora del proyecto.

Resultado final de SonarInk

Filter matched 100 of 177 items		
Resource	Date	Description
GestionAvionCRUD.java		Refactor this method to reduce its Cognitive Complexity from 38 to the 15 allowed. [+15 locations]
GestionPilotoCRUD.java		Refactor this method to reduce its Cognitive Complexity from 35 to the 15 allowed. [+15 locations]
GestionPilotoCRUD.java	1 month ago	Refactor this method to reduce its Cognitive Complexity from 26 to the 15 allowed. [+16 locations]
GestionAvionCRUD.java	1 hour ago	Refactor this method to reduce its Cognitive Complexity from 22 to the 15 allowed. [+12 locations]
GestionIncidencias.java	1 month ago	Refactor this method to reduce its Cognitive Complexity from 17 to the 15 allowed. [+12 locations]
GestionRuta.java	1 month ago	Make the enclosing method "static" or remove this set.
Menus.java	1 month ago	Define a constant instead of duplicating this literal "Error: Debes ingresar un número entero." 5 times. [+5 locations]
GestionAvionCRUD.java		Define a constant instead of duplicating this literal "ERROR" 3 times. [+3 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " Opción : " 11 times. [+11 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " ---{ Volviendo al Menú Principal.}--- " 4 times. [+4 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " Seleccione una opción: " 5 times. [+5 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " Bienvenido Piloto " 5 times. [+5 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " Bienvenido Admin " 5 times. [+5 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " [6]. Salir Programa" 5 times. [+5 locations]
Menus.java	1 month ago	Define a constant instead of duplicating this literal " [3]. Incidencias " 6 times. [+6 locations]
GestionAvionCRUD.java	3 days ago	Use or store the value returned from "readLine" instead of throwing it away.
GestionAvionCRUD.java	3 days ago	Use or store the value returned from "readLine" instead of throwing it away.
GestionIncidencias.java	3 days ago	Use or store the value returned from "readLine" instead of throwing it away.
GestionIncidencias.java	5 days ago	Use or store the value returned from "readLine" instead of throwing it away.
GestionPilotoCRUD.java	3 days ago	Use or store the value returned from "readLine" instead of throwing it away.
GestionPilotoCRUD.java	5 days ago	Use or store the value returned from "readLine" instead of throwing it away.
GestionRuta.java	3 hours ago	Use or store the value returned from "readLine" instead of throwing it away.
GestionRuta.java	3 hours ago	Use or store the value returned from "readLine" instead of throwing it away.
Propiedad.java		Use "java.nio.file.Files#delete" here for better messages on error conditions.
Menus.java	1 month ago	Update this method so that its implementation is not identical to "menuIncidenc" on line 397. [+1 location]
GestionIncidencias.java	58 minutes ago	This block of commented-out lines of code should be removed. [+42 locations]
GestionRuta.java	3 hours ago	This block of commented-out lines of code should be removed. [+4 locations]
GestionAvionCRUD.java	3 days ago	This block of commented-out lines of code should be removed. [+3 locations]
GestionPilotoCRUD.java	3 days ago	This block of commented-out lines of code should be removed. [+3 locations]
GestionRuta.java	3 hours ago	This block of commented-out lines of code should be removed. [+3 locations]
GestionAvionCRUD.java	3 days ago	This block of commented-out lines of code should be removed. [+2 locations]
GestionIncidencias.java	3 days ago	This block of commented-out lines of code should be removed. [+2 locations]
GestionPilotoCRUD.java	3 days ago	This block of commented-out lines of code should be removed. [+2 locations]
PruebasPiloto.java	1 month ago	Swap these 2 arguments so they are in the correct order: expected value, actual value. [+1 location]
PruebasPiloto.java	1 month ago	Swap these 2 arguments so they are in the correct order: expected value, actual value. [+1 location]
GestionRuta.java	3 hours ago	Replace this String concatenation with Text block.
Menus.java	1 month ago	Replace this String concatenation with Text block.
Menus.java	1 month ago	Replace this String concatenation with Text block.
Menus.java	1 month ago	Replace this String concatenation with Text block.
GestionAvionCRUD.java	1 month ago	Rename this variable to not match a restricted identifier.
GestionPilotoCRUD.java	1 month ago	Rename this variable to not match a restricted identifier.
GestionRuta.java	3 hours ago	Rename "rutas" which hides the field declared at line 33.
GestionPilotoCRUD.java	5 days ago	Rename "pilotos" which hides the field declared at line 22.

Resource	Date	Description
Menu.java	3 days ago	Rename "p" which hides the field declared at line 25.
Menu.java	30 days ago	Rename "opcion" which hides the field declared at line 16.
Menu.java	30 days ago	Rename "opcion" which hides the field declared at line 16.
Menu.java	30 days ago	Rename "opcion" which hides the field declared at line 16.
Menu.java	30 days ago	Rename "opcion" which hides the field declared at line 16.
Menu.java	1 month ago	Rename "opcion" which hides the field declared at line 16.
GestionAvionCRUD.java	3 days ago	Rename "aviones" which hides the field declared at line 24.
GestionAvionCRUD.java		Remove this useless assignment; "vuelo" already holds the assigned value along all execution paths.
GestionPilotoCRUD.java		Remove this useless assignment; "vuelo" already holds the assigned value along all execution paths.
GestionRuta.java	9 minutes ago	Remove this useless assignment; "ok" already holds the assigned value along all execution paths.
Propiedad.java		Remove this useless assignment to local variable "p".
GestionIncidencias.java		Remove this useless assignment to local variable "descrip".
GestionAvionCRUD.java		Remove this unused method parameter "sc".
GestionPilotoCRUD.java	3 days ago	Remove this unused method parameter "sc".
GestionRuta.java	3 hours ago	Remove this unused method parameter "rutas".
Menu.java	1 month ago	Remove this unused method parameter "entidad".
Menu.java	1 month ago	Remove this unused method parameter "args".
GestionRuta.java	1 month ago	Merge this if statement with the enclosing one. [+1 location]
GestionAvionCRUD.java		Either remove or fill this block of code.
GestionAvionCRUD.java	1 month ago	Change this condition so that it does not always evaluate to "true" [+2 locations]
GestionPilotoCRUD.java	3 days ago	Change this condition so that it does not always evaluate to "true" [+2 locations]
Registro.java		Change the visibility of this constructor to "protected". [+1 location]
Registro.java		Change the visibility of this constructor to "protected". [+1 location]
Avion.java		Add the "@Override" annotation above this method signature
Avion.java		Add the "@Override" annotation above this method signature
Avion.java		Add the "@Override" annotation above this method signature
Avion.java		Add the "@Override" annotation above this method signature
Avion.java		Add the "@Override" annotation above this method signature
Avion.java		Add the "@Override" annotation above this method signature
Piloto.java		Add the "@Override" annotation above this method signature
Piloto.java		Add the "@Override" annotation above this method signature
Piloto.java		Add the "@Override" annotation above this method signature
Piloto.java		Add the "@Override" annotation above this method signature
Piloto.java		Add the "@Override" annotation above this method signature
Constantes.java		Add a private constructor to hide the implicit public one.
GestionRuta.java	3 hours ago	The type of "rutas" should be an interface such as "List" rather than the implementation "ArrayList".
GestionRuta.java	1 month ago	The type of "rutas" should be an interface such as "List" rather than the implementation "ArrayList".
GestionRuta.java	1 month ago	The type of "rutas" should be an interface such as "List" rather than the implementation "ArrayList".
GestionRuta.java	1 month ago	The type of "rutas" should be an interface such as "List" rather than the implementation "ArrayList".
GestionIncidencias.java		The type of "pilotos" should be an interface such as "List" rather than the implementation "ArrayList".
GestionPilotoCRUD.java		The type of "pilotos" should be an interface such as "List" rather than the implementation "ArrayList".
GestionIncidencias.java	5 days ago	The type of "incidencia" should be an interface such as "List" rather than the implementation "ArrayList".
GestionAvionCRUD.java	1 month ago	The type of "aviones" should be an interface such as "List" rather than the implementation "ArrayList".

45 files of project AirBaseProgramI (at 08/04/2024 00:17). [Learn more](#)

GestionIncidencias.java	5 days ago	The type of "incidencia" should be an interface such as "List" rather than the implementation "ArrayList".
GestionAvionCRUD.java	1 month ago	The type of "aviones" should be an interface such as "List" rather than the implementation "ArrayList".
GestionIncidencias.java		The type of "aviones" should be an interface such as "List" rather than the implementation "ArrayList".
GestionAvionCRUD.java	3 days ago	The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionAvionCRUD.java	1 month ago	The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionIncidencias.java	5 days ago	The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionIncidencias.java		The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionPilotoCRUD.java	3 days ago	The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionPilotoCRUD.java	3 days ago	The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionRuta.java	3 hours ago	The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionRuta.java	1 month ago	The return type of this method should be an interface such as "List" rather than the implementation "ArrayList".
GestionAvionCRUD.java	1 hour ago	Replace this if-then-else statement by a single method invocation.
GestionPilotoCRUD.java	1 month ago	Replace this if-then-else statement by a single method invocation.
GestionAvionCRUD.java		Replace the type specification in this constructor call with the diamond operator ("<>").
GestionIncidencias.java		Replace the type specification in this constructor call with the diamond operator ("<>").