



**UNIVERSIDAD DE OVIEDO**

**ESCUELA POLITÉCNICA DE INGENIERÍA DE  
GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**“IMPLEMENTACIÓN Y MANTENIMIENTO DE LA  
HERRAMIENTA DE APOYO A LA DECISIÓN OPTAIN-SA PARA  
CONTROL DE TRÁFICO AÉREO”**



**ANDREA MENÉNDEZ GARCÍA  
JUNIO 2014**



**UNIVERSIDAD DE OVIEDO**  
**ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**“IMPLEMENTACIÓN Y MANTENIMIENTO DE LA  
HERRAMIENTA DE APOYO A LA DECISIÓN OPTAIN-SA PARA  
CONTROL DE TRÁFICO AÉREO”**

**DOCUMENTO N° I**

**MEMORIA**



**ANDREA MENÉNDEZ GARCÍA**  
**JUNIO 2014**

**LENGUAJES Y SISTEMAS  
INFORMÁTICOS**  
**TUTOR: PABLO JAVIER TUYA  
GONZÁLEZ**

## Tabla de contenido

Tabla de contenido .....	2
Índice de Ilustraciones.....	3
Índice de tablas.....	4
Resumen .....	5
Abstract .....	5
1. Introducción .....	6
2. Ámbito y alcance .....	7
2.1. Necesidad de la aplicación .....	7
2.2. Objetivos.....	7
2.3. Partes interesadas.....	7
2.4. Alcance .....	7
3. Terminología.....	8
3.1. Términos técnicos.....	8
3.2. Glosario .....	8
4. Requisitos técnicos.....	9
4.1. Servidor Web.....	9
4.2. Requisitos software .....	9
4.3. Entorno de desarrollo .....	10
4.3.1. Características .....	10
5. Concepto operacional.....	11
6. Documentos.....	14
7. Metodología usada .....	15
8. Trabajo realizado.....	17
9. Conclusiones .....	18
10. ANEXO .....	19
10.1. Tablas de velocidad .....	19
11. Bibliografía y referencias.....	20

## **Índice de Ilustraciones**

Ilustración 1: Remote System Details .....	10
Ilustración 2: Descripción del sistema .....	12
Ilustración 3: Concepto Operacional - TIELO (LORES) & TIETOL (TOLSO) & TIEKE (KENAS) .....	13
Ilustración 4: Fase de construcción .....	15

## **Índice de tablas**

Tabla 1: Ajuste de velocidad para separación de 5NM .....	19
Tabla 2: Ajuste de velocidad para separación de 10 NM .....	19

## Resumen

Este proyecto se enmarca dentro del proyecto SACTA (Sistema Automatizado de Control y Tráfico Aéreo), sistema de control de tráfico aéreo utilizado en España. Tiene como objetivo utilizarse como una herramienta de ayuda a la decisión para los controladores de tráfico aéreo, con vistas a su utilización en el entorno de Palma de Mallorca.

El objetivo de este proyecto es desarrollar un conjunto de funcionalidades para el procedimiento OPTAIN-SA (Optimized Profile Descent Approaches Implementing Windows-Speed Advisory) aplicado a la configuración oeste del aeródromo de Palma de Mallorca, para las llegadas occidentales que se integran en el fijo de aproximación inicial de Pollensa.

Esta herramienta de apoyo a la decisión permite que el controlador aplique el procedimiento OPTAIN-SA desde su posición de control. La herramienta presenta al controlador de tráfico aéreo recomendaciones de velocidad aplicables a una secuencia de aeronaves en aproximación. Con esto se consigue organizar adecuadamente las secuencias de llegadas y la separación de las aeronaves, así como conseguir una mejora de la capacidad y la seguridad.

## Abstract

This project is part of the SACTA project (Automated Air Traffic Control System) air traffic control system used in Spain. Its objective is to be used as a taking decisions aid tool by air traffic controllers, with a view to being implemented in Palma (Majorca) area.

The objective of this project is to develop a set of functionalities for the OPTAIN-SA (Optimized Profile Descent Approaches Implementing Windows-Speed Advisory) procedure applied to the configuration of the west side of Palma airport, for western arrivals that are integrated in the Pollensa initial approach fix.

This decision support tool enables the air traffic controller to apply OPTAIN-SA procedure from his control position. The tool presents to air traffic controller the air traffic speed recommendations applicable to a sequence of aircrafts in proximity. With it both the sequence of arrivals and aircrafts separation is organized properly and also the capacity and security improvement.

## 1. Introducción

Este proyecto surge con el objetivo de ayudar a optimizar el tráfico aéreo en las aproximaciones al aeropuerto de Palma de Mallorca mediante el cálculo de velocidades para las aeronaves que confluyen en el Fijo de Aproximación Inicial (IAF) de Pollensa, provenientes del mismo o distintos flujos de tráfico, garantizando una separación longitudinal mínima requerida entre cada pareja de aeronaves en la aproximación final al aeropuerto.

La aplicación que llevará a cabo este control del tráfico aéreo recibe el nombre de OPTAIN-SA (Optimized Profile Descent Approaches Implementing Windows-Speed Advisory).

La separación requerida en el IAF puede ser de 5NM ó 10NM dependiendo de las exigencias del tráfico y/o las necesidades de Control. En función de la separación requerida, la herramienta que implemente el procedimiento OPTAIN-SA, basándose en unas tablas de velocidad previamente definidas (Ver 10.1), propondrá al controlador los ajustes de velocidad necesarios que permitan facilitar las operaciones de descenso continuo.

Asimismo, conviene resaltar que las propuestas de velocidad proporcionadas por la herramienta que implemente el procedimiento OPTAIN-SA son de carácter meramente informativo. La separación correspondiente entre aeronaves y las velocidades deberán validarse por el controlador, comunicarse al piloto y finalmente verificarse en la pantalla radar de la Posición de Control.

## **2. Ámbito y alcance**

### **2.1. Necesidad de la aplicación**

Esta aplicación surge de la necesidad de obtener una herramienta de apoyo para las personas encargadas de controlar el tráfico aéreo, que les permita conocer las aeronaves próximas a un punto fijo y la velocidad que deben llevar, manteniendo en todo momento una distancia mínima con las aeronaves anterior y posterior.

### **2.2. Objetivos**

El objetivo del procedimiento OPTAIN-SA es ayudar al controlador de tráfico aéreo en el proceso de proveer instrucciones de ajuste de velocidad a las aeronaves en la secuencia de llegada al Fijo de Aproximación Inicial (IAF) de Pollensa, para el mismo o distintos flujos de tráfico, de forma que se asegure la separación longitudinal requerida.

La aplicación consiste en la implementación de una herramienta ATC (Air Traffic Control) que, a través de una Ventana OPTA-IN, muestre gráficamente la secuencia de vuelos de llegada al IAF del aeropuerto LEPA (Aeropuerto de Palma de Mallorca) desde los flujos de tráfico correspondientes, sobre una regla de distancia vertical, junto con el ajuste de velocidad dado por el procedimiento para las dos separaciones consideradas (5NM/10NM).

### **2.3. Partes interesadas**

Las partes interesadas cuya presencia es fundamental en diferentes etapas del proyecto se describen a continuación.

En primer lugar AENA (Aeropuertos Españoles y Navegación Aérea), se trata del proveedor principal de servicios de tránsito aéreo en España.

En segundo lugar está Indra sistemas que son los encargados de dar las especificaciones, se trata por tanto del cliente.

El encargado de llevar a cabo el desarrollo de la aplicación es Indra Software Labs.

Y por último aparece la figura del usuario de la aplicación que serán los controladores aéreos. La función que desempeñan los controladores aéreos es dirigir el tránsito de aeronaves en el espacio aéreo y en los aeropuertos garantizando seguridad, orden y rapidez.

Los controladores son los encargados de autorizar a los pilotos con instrucciones de información necesarias dentro del espacio aéreo bajo su jurisdicción con el objetivo de prevenir colisiones, principalmente entre aeronaves y obstáculos en el área de maniobras.

### **2.4. Alcance**

El alcance de este proyecto que se presenta como trabajo fin de Máster consiste en el diseño y la codificación en lenguaje Java de una parte de la funcionalidad descrita en los requisitos. Además se incluye el mantenimiento y correcciones necesarias debido a los defectos que se pudieran encontrar durante las pruebas llevadas a cabo por el cliente así como los cambios que estos soliciten.



### 3. Terminología

Se incluyen a continuación un conjunto de términos técnicos [IACATA] así como una serie de acrónimos a los que se hace referencia en el presente documento.

#### 3.1. Términos técnicos

- *Controlador de tráfico aéreo*: personas encargadas de coordinar los movimientos de cada una de las aeronaves desde su estacionamiento en el aeropuerto de origen, su despegue, su vuelo por el espacio aéreo, su aterrizaje y hasta su estacionamiento en el aeropuerto destino, manteniendo las distancias de seguridad entre ellas, evitando su vuelo por zonas de meteorología adversa y asegurando su flujo de tráfico que permita retrasos mínimos.
- *Control de tráfico aéreo*: servicio reglamentado cuya labor es distribuir el espacio Aéreo y separar Aeronaves que pretenden utilizarlo.
- *Torre de control*: Es una instalación elevada que desde su sala de control a través de un fanal se contempla visualmente el aeródromo y sus inmediaciones.
- *Kts*: Nudos, knots.
- *Aproximación*: Fase del vuelo donde se ejecuta un procedimiento de aproximación que alinea la aeronave con la pista de aterrizaje designada del aeropuerto destino.

#### 3.2. Glosario

- *AENA*: Aeropuertos Españoles y Navegación Aérea
- *ATC*: Air Traffic Control
- *BBDD*: Bases de Datos
- *CVS*: Concurrent Versions System
- *HCI*: Human Computer Interaction
- *IAF*: Initial Approach Fix
- *IDE*: Entorno de desarrollo integrado
- *LEPA*: Aeropuerto de Palma de Mallorca
- *MIDAS*: Método Indra para Desarrollo, Adaptación y Servicios
- *MIGP*: Método Indra para la Gestión de Proyectos
- *MMU*: Unidad de manejo de memoria
- *NM*: Nautical Miles
- *OPTAIN-SA*: Optimized Profile Descent Approaches Implementing Windows-Speed Advisory
- *RISC*: Reduced Instruction Set Computer
- *SACTA*: Sistema Automatizado de Control y Tráfico Aéreo
- *SINA*: Sistema de Información de Navegación Aérea
- *TMA*: Terminal Manoeuvre Area

## 4. Requisitos técnicos

El PC donde debe ejecutar el cliente está predefinido [*RDIO*] y tiene los siguientes requisitos técnicos:

- Windows XP SP2
- Internet Explorer 8
- 2GB RAM
- Resolución de pantalla 1280 x 800
- CPUx1 2GHz

### 4.1. *Servidor Web*

- Sun ULTRA 60 , 1 Procesador
- 512 Mb RAM
- Monitor 17", teclado y ratón
- 2 interfaces conexión LAN
- 1 Lector CD-ROM
- Disco Interno mínimo de 8GB

### 4.2. *Requisitos software*

- Sistema Operativo: Solaris 2.8
- Oracle 8.1.5
  - Servidor Web: Apache 1.3.6
  - Motor de ejecución de Servlets : Jserv 1.0
  - Motor de ejecución de páginas JSP: GNU-JSP 1.1
  - Máquina Virtual Java: 1.0

### 4.3. Entorno de desarrollo

La aplicación se desarrolla con la plataforma de desarrollo Eclipse, en su versión 3.7. Se trata de una plataforma de código abierto y multiplataforma. Es una completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones en Java como en este caso.

#### 4.3.1. Características

Las características principales de la plataforma Eclipse que son requeridas para el desarrollo de este proyecto son:

- El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no.
- Permite la posibilidad de trabajar desde terminal.
- Permite el acceso remoto mediante la instalación de un plug-in.

The screenshot shows the 'Remote System Details' window in Eclipse. It contains a table with the following data:

Resource	Parent profile	Remote syste...	Connection st...	Host name	Default User ID	Description
Local	ASCASTANO...	Local	Some subsystem...	LOCALHOST	amenendezg	
sina2	ASCASTANO...	SSH Only	Some subsystem...	192.168.21.87	sactaweb	

Ilustración 1: Remote System Details

## 5. Concepto operacional

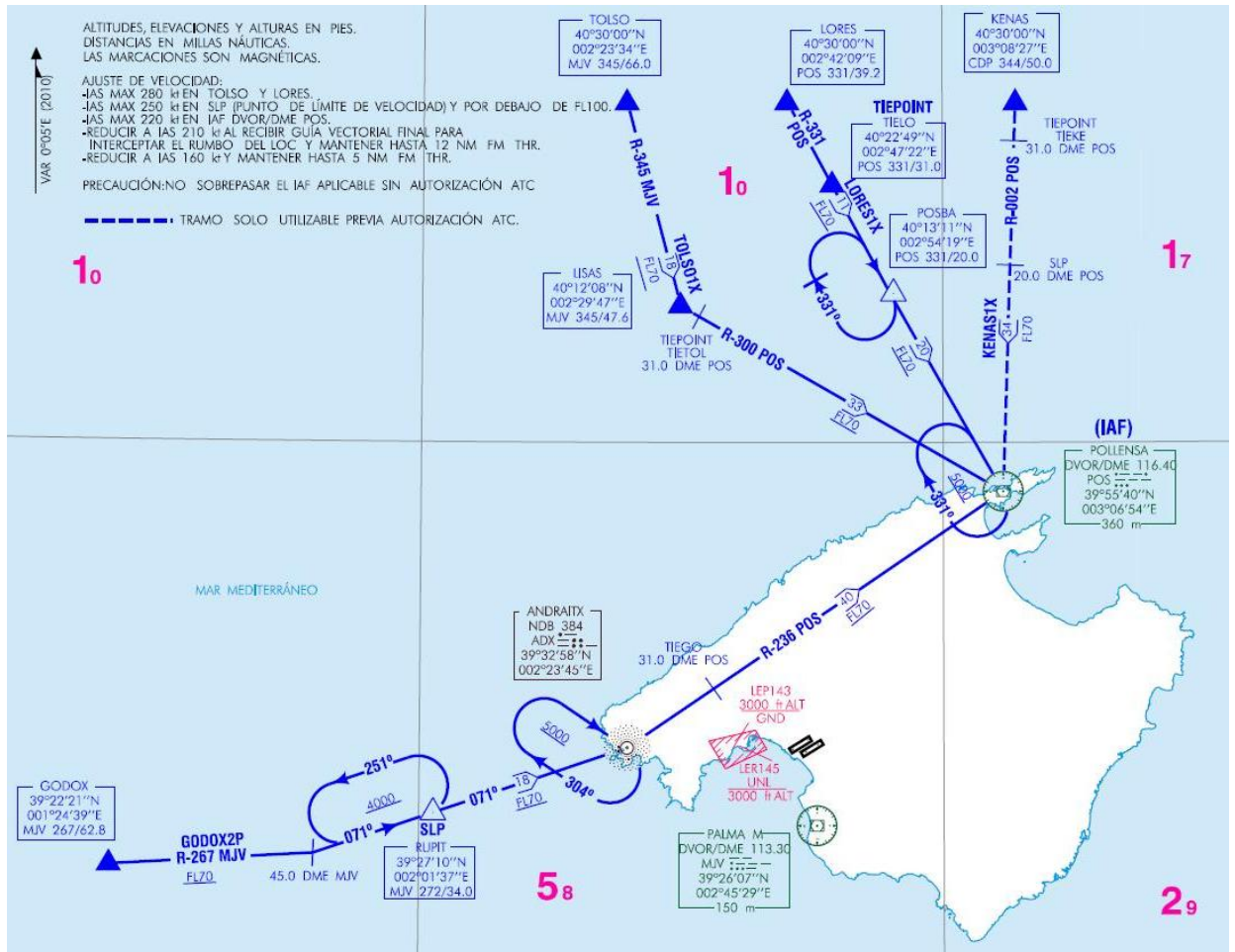
El concepto operacional [OPDAIW] del procedimiento OPTAIN-SA se basa en los siguientes parámetros:

- Velocidad de entrada al TMA (Terminal Manoeuvre Area). Las aeronaves serán transferidas al TMA de Palma de Mallorca a una velocidad de referencia de 280 Kts (nudos).
- Puntos de referencia, conocidos como “TIE”, equidistantes al IAF una distancia definida por configuración y situados en cada uno de los flujos de llegada. Éstos puntos permitirán al controlador evaluar la separación longitudinal con la que van a confluir dos aeronaves consecutivas en llegada al IAF y por tanto evaluar si es necesario realizar un ajuste de velocidad sobre una de ellas para evitar el conflicto. Típicamente se representarán como los puntos de corte de un arco, centrado en el IAF y a una distancia configurable de éste, con los distintos flujos de tráfico aéreo de llegada.
- Separación longitudinal de los vuelos en la secuencia de llegada sobre el IAF. Tal distancia será especificada por el controlador encargado del tráfico en dicho punto. Se considerarán como posibles valores 5NM ó 10NM.
- Velocidad de una aeronave de la secuencia de llegada y distancia de ésta a la aeronave anterior de la secuencia en el momento en que esta segunda cruce su punto TIE.
- Limitaciones de velocidad en el Speed Limit Point (250 kts) y en el IAF (220 kts), las cuales se mantienen igual que en el modo de operación actual.

De esta forma, dadas dos aeronaves que se aproximen al IAF por el mismo o distinto flujo, cuando la primera aeronave alcance una distancia dada hasta el IAF (representado por el paso de ésta por el punto TIE correspondiente, dado por el punto de cruce del arco circular centrado en el IAF con el flujo de llegada correspondiente, como puede verse en la **Ilustración 2**), se podrá ajustar la velocidad de la aeronave posterior en la secuencia de llegada, basándose en unas tablas previamente definidas (Ver 10.1), de forma que el controlador dispondrá de las velocidades recomendadas necesarias que faciliten las operaciones de descenso continuo para la segunda aeronave, dependiendo de la distancia de ésta a su punto TIE en tal instante y de la distancia configurada por el controlador (5NM/10NM) para la separación de los vuelos de la secuencia de llegada.



En la **Ilustración 3** se pueden ver los puntos TIE, uno para cada flujo de entrada y el punto IAF que es donde todos los flujos confluyen antes de comenzar la fase de aproximación.



**Ilustración 3: Concepto Operacional - TIELO (LORES) & TIETOL (TOLSO) & TIEKE (KENAS)**

## **6. Documentos**

La distribución de los distintos documentos que componen proyecto es la siguiente:

- I. Memoria
- II. Planificación y Presupuesto
- III. Documentos técnico-científicos

El primer documento es la memoria donde se describen los objetivos de la aplicación, el alcance y se indicará el trabajo realizado.

El segundo documento recoge la planificación detallada de las tareas seguidas para la elaboración de la aplicación y el presupuesto correspondiente.

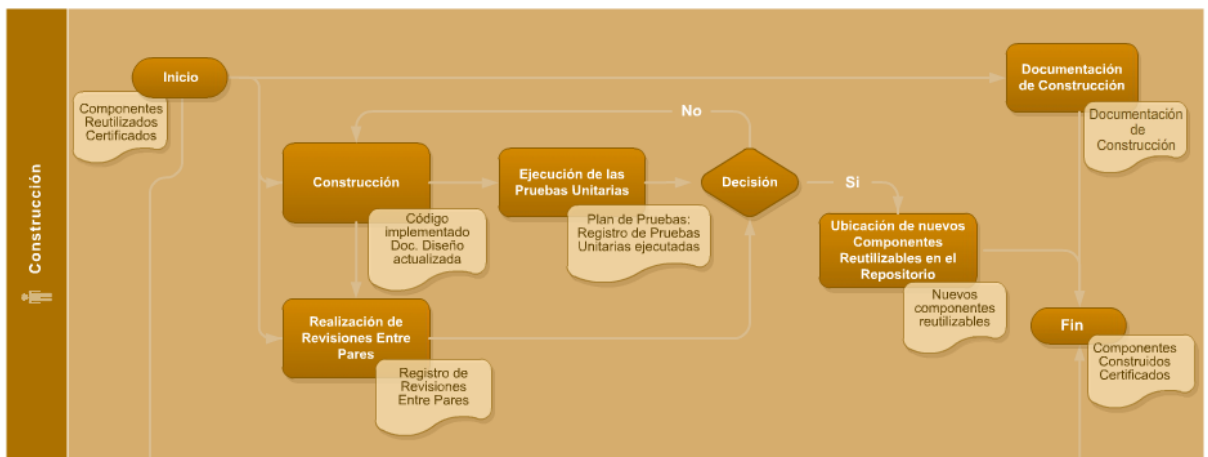
Por último, el documento con el nombre Documentos científico-técnicos incluirá las especificaciones de requisitos, diseño, los manuales necesarios y el código fuente que se ha considerado oportuno.

## 7. Metodología usada

Respecto a la metodología usada para este proyecto, se puede decir, que se toman diversas metodologías. Se aplican aquellas técnicas que parecen más adecuadas teniendo en cuenta las características del proyecto.

La metodología principal que se aplica en Indra se llama MIDAS [MIDAS] (Método Indra para Desarrollo, Adaptación y Servicios) que se utiliza de forma combinada con MIGP (Método Indra para la Gestión de Proyectos) que es la norma por la cual se gestionan los ciclos precontractual y contractual de los proyectos mediante los cuales se estructura el negocio de Indra.

Dentro de MIDAS, para la fase de construcción, se aplica MIDO. La etapa MIDO está centrada en la elaboración de software según la necesidad descrita, la solución propuesta y los acuerdos concertados. Se corresponde con el desarrollo/reutilización o adquisición/adaptación de componentes para la consecución de un sistema/producto.



**Ilustración 4: Fase de construcción**

En esta tarea de construcción se realizará la implementación del modelo físico de datos diseñado previamente (creando las tablas de bases de datos necesarias y otros elementos necesarios), así como los informes/formularios, interfaces de usuario HCI, pantallas, flujos de navegación, etc. diseñados en fases anteriores. En todo caso, esta implementación debe realizarse según los estándares definidos y en función de los requisitos del usuario. Si fuera necesario se actualizará la documentación de diseño de partida.

En cuanto a la construcción de las unidades software se debe:

- Realizar la codificación en función de la documentación recibida y acorde a los estándares fijados.
- Incluir comentarios explicativos en el código implementado
- Actualizar la documentación de la unidad de software (al menos, descripción general y notas del programador).
- Identificar código no portable, entendiendo como tal aquella estructura de programación (parte del código de la unidad software) no soportada por alguna plataforma, y documentarlo adecuadamente.



- Se deberá tener especial cuidado en la construcción de las Interfaces de los Componentes con el fin de que se garantice su integración y servicio en el sistema en conjunto.

Con todo esto se pretende que la documentación generada sea clara y sencilla de leer, de tal forma que sea lo más definida posible, para que pueda ser leída sin dificultades por la mayor parte del público en general.

## 8. Trabajo realizado

El trabajo realizado consiste en el diseño y la codificación en lenguaje Java de una parte de la funcionalidad descrita por los requisitos. Las funcionalidades implementadas han sido:

- Generación de informes.
- Panel de selección de velocidad propuesta.

En el “Documento técnico-científicos” se explica el objetivo de estas funcionalidades, su funcionamiento y la implementación de ambas.

Y por otro lado un conjunto de dieciocho cambios solicitados por el cliente, así como los problemas detectados. Estos cambios se describen en el documento “Documento técnico-científicos” en el que además se detalla cómo se ha llevado a cabo la resolución de los mismos.

## **9. Conclusiones**

La herramienta ha cumplido con los objetivos planteados. Actualmente OPTAIN-SA ha sido validado por los controladores y los resultados obtenidos han sido satisfactorios. Se han realizado varias aproximaciones controladas según el procedimiento OPTA-IN y con los datos proporcionados por la herramienta.

Los únicos ajustes necesarios en estas validaciones han sido geográficos para lo que ha sido necesario modificar estos elementos en la configuración de la base de datos.

En cuanto a las herramientas utilizadas para el desarrollo del proyecto se ha decidido trabajar con Eclipse por ser gratuito e integrar todo lo necesario para el desarrollo como es la posibilidad de edición remota y la posibilidad de trabajar desde terminal.

Por otro lado el lenguaje de programación Java se ha utilizado debido a que se trata la de la evolución de un sistema existente (SINA, Sistema de Información de Navegación Aérea) programado en este lenguaje.

## 10. ANEXO

### 10.1. Tablas de velocidad

A continuación se recogen las tablas de ajuste de velocidad [EDPO] propuestas por el algoritmo del procedimiento OPTAIN-SA. Dichas tablas fueron calculadas durante el desarrollo del proyecto OPTAIN y relacionan las distancias de una aeronave de la secuencia de llegada a su punto TIE, en el momento de paso de la aeronave anterior en la secuencia a su propio punto TIE, con las indicaciones de ajustes de velocidad propuestas para la aeronave posterior. Éstas han sido calculadas para puntos TIE situados a una distancia de 31NM del IAF.

$d^{\text{offset}}$ [N]	IAS [kts]
3.0 – 3.8	220
3.8 – 4.7	230
4.7 – 5.6	240
5.6 – 6.3	250
6.3 – 7.3	260
7.3 – 7.6	270
> 7.6	280

**Tabla 1: Ajuste de velocidad para separación de 5NM**

$d^{\text{offset}}$ [N]	IAS [kts]
9.4 – 10.6	220
10.6 – 12.0	230
12.0 – 13.3	240
13.3 – 14.2	250
14.2 – 15.2	260
15.2 – 16.0	270
> 16.0	280

**Tabla 2: Ajuste de velocidad para separación de 10 NM**

## 11. Bibliografía y referencias

Se cita a continuación la bibliografía más importante y que ha resultado más útil, entre toda la documentación consultada para el desarrollo de este documento.

Toda ella consultada, tanto para la documentación como para el desarrollo de la aplicación.

- *[OPDAIW]Optimized Profile Descent Approaches Implementing Windows (OPTAIN), Ineco 2012*
- *[IACTA]“Introducción al control de tráfico aéreo”, Indra, 2013*
- *[MIDAS]“Método Indra para Desarrollo, Adaptación y Servicios”, Documentación de metodología MIDAS consultable en la intranet de Indra, 2013*
- *[EDPO]Especificación del procedimiento OPTAIN-SA en el sistema SINA, Indra 2013*
- *[RDIO]Requisitos de implantación de OPTAIN-SA en el sistema SINA, Indra 2013*
- *<http://www.aena.es/>*



**UNIVERSIDAD DE OVIEDO**  
**ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**“IMPLEMENTACIÓN Y MANTENIMIENTO DE LA  
HERRAMIENTA DE APOYO A LA DECISIÓN OPTAIN-SA PARA  
CONTROL DE TRÁFICO AÉREO”**

**DOCUMENTO N° II**

**PLANIFICACIÓN Y PRESUPUESTO**



**ANDREA MENÉNDEZ GARCÍA**  
**JUNIO 2014**

**LENGUAJES Y SISTEMAS  
INFORMÁTICOS**  
**TUTOR: PABLO JAVIER TUYA  
GONZÁLEZ**

## Tabla de contenido

Tabla de contenido .....	2
Índice de Ilustraciones.....	3
Índice de Tablas.....	4
1. Planificación.....	5
1.1. Descripción de las tareas .....	5
1.2. Estimación de recursos necesarios .....	6
1.2.1. Recursos humanos.....	6
1.2.2. Recursos software .....	6
1.2.3. Recursos hardware .....	7
1.3. Planificación temporal.....	8
2. Presupuesto.....	9
2.1. Presupuesto detallado .....	9
2.1.1. Recursos humanos.....	9
2.1.2. Recursos software .....	10
2.1.3. Recursos hardware .....	11
2.2. Presupuesto total.....	12

## Índice de Ilustraciones

Ilustración 1: Planificación temporal .....	8
Ilustración 2: Planificación temporal .....	8



## Índice de Tablas

Tabla 1: Estimación de recursos humanos .....	6
Tabla 2: Estimación de recursos software.....	6
Tabla 3: Estimación de recursos hardware .....	7
Tabla 4: Estimación de recursos humanos .....	9
Tabla 5: Estimación de recursos software.....	10
Tabla 6: Estimación de recursos hardware .....	11
Tabla 7: Presupuesto total sin I.V.A. ....	12
Tabla 8: Presupuesto total .....	12

## 1. Planificación

Se recoge a continuación la planificación temporal del proyecto. En la planificación temporal se realizará una estimación de los recursos necesarios (software, hardware y humano) para su desarrollo. Además, se incluye el diagrama de Gantt que representa la secuencia temporal de actividades a completar a lo largo de la ejecución del proyecto.

### 1.1. Descripción de las tareas

- **Análisis de documentación:** Engloba aquellas tareas llevadas a cabo con el objetivo de repasar conceptos y conocer otros nuevos con los que se trabajan en el desarrollo de la aplicación.
  - Introducción al tráfico aéreo
  - Arquitectura
  - Servlets + JSP
  - Applets
  - Diseño
  - Requisitos
  
- **Análisis de solución actual SINA:** Se dedican varios días a analizar la solución de la que se parte para generar el nuevo servicio OPTAIN, el entorno de desarrollo utilizado, el entorno de pruebas y una breve revisión del código.
  - Entorno de desarrollo
  - Entorno de pruebas
  - Revisión de código
  
- **Construcción OPTAIN:** La parte importante del proyecto se encuentra en este apartado. Se trata por un lado del desarrollo de las dos funcionalidades, por otro lado la resolución de las diferentes anomalías que se presentan y por último la generación de la memoria que se lleva a cabo durante todo el desarrollo del proyecto.
  - Diálogo Cambio Velocidad
  - Diálogo Informes
  - Resolución de anomalías
  - Generación de memoria

## 1.2. Estimación de recursos necesarios

### 1.2.1. Recursos humanos

Para la estimación de las horas dedicadas, se ha considerado que se trabajará con una media de cinco horas útiles cada día en el desarrollo del proyecto.

Id Unidad	Descripción	Unidad de medición	Nº de unidades
HU01	Programador junior	Horas	410

Tabla 1: Estimación de recursos humanos

### 1.2.2. Recursos software

Id Unidad	Descripción	Unidad de medición	Nº de unidades
SW01	Windows 7 Enterprise SP1	Unidades	1
SW02	Microsoft Office 2007	Unidades	1
SW03	Eclipse Indigo	Unidades	1
SW04	JRE 1.6_0_37	Unidades	1
SW05	yEd Graph Editor	Unidades	1
SW06	Openproj-1.4	Unidades	1

Tabla 2: Estimación de recursos software

### 1.2.3. Recursos hardware

<b>Id Unidad</b>	<b>Descripción</b>	<b>Unidad de medición</b>	<b>Nº de unidades</b>
HW01	PC HP	Unidades	1

**Tabla 3: Estimación de recursos hardware**

### 1.3. Planificación temporal

Se detalla la planificación temporal del proyecto. Cada una de las tareas descritas en el apartado anterior han sido planificadas de forma individual.

	Nombre	Duración	Inicio	Terminado
1	<b>☒ Análisis de documentación</b>	9 days	<b>2/12/13 8:00</b>	<b>16/12/13 17:00</b>
2	Introducción al tráfico aéreo	2 days	2/12/13 8:00	3/12/13 17:00
3	Arquitectura	1 day	4/12/13 8:00	4/12/13 17:00
4	Servlets + JSP	1 day	5/12/13 8:00	5/12/13 17:00
5	Applets	1 day	10/12/13 8:00	10/12/13 17:00
6	Diseño	2 days	11/12/13 8:00	12/12/13 17:00
7	Requisitos	2 days	13/12/13 8:00	16/12/13 17:00
8	<b>☒ Análisis de Solución actual SINA</b>	12 days	<b>17/12/13 8:00</b>	<b>14/01/14 17:00</b>
9	Entorno de desarrollo	2 days	17/12/13 8:00	18/12/13 17:00
10	Entorno de pruebas	4 days	19/12/13 8:00	3/01/14 17:00
11	Revisión de código	6 days	7/01/14 8:00	14/01/14 17:00
12	<b>☒ Construcción OPTAIN</b>	40 days	<b>15/01/14 8:00</b>	<b>13/03/14 17:00</b>
13	Diálogo Cambio Velocidad	20 days	15/01/14 8:00	11/02/14 17:00
14	Diálogo Informes	20 days	12/02/14 8:00	13/03/14 17:00
15	Resolución de anomalías	48 days	20/01/14 8:00	28/03/14 17:00
16	Generación de memoria	73 days	17/12/13 8:00	11/04/14 17:00

Ilustración 1: Planificación temporal

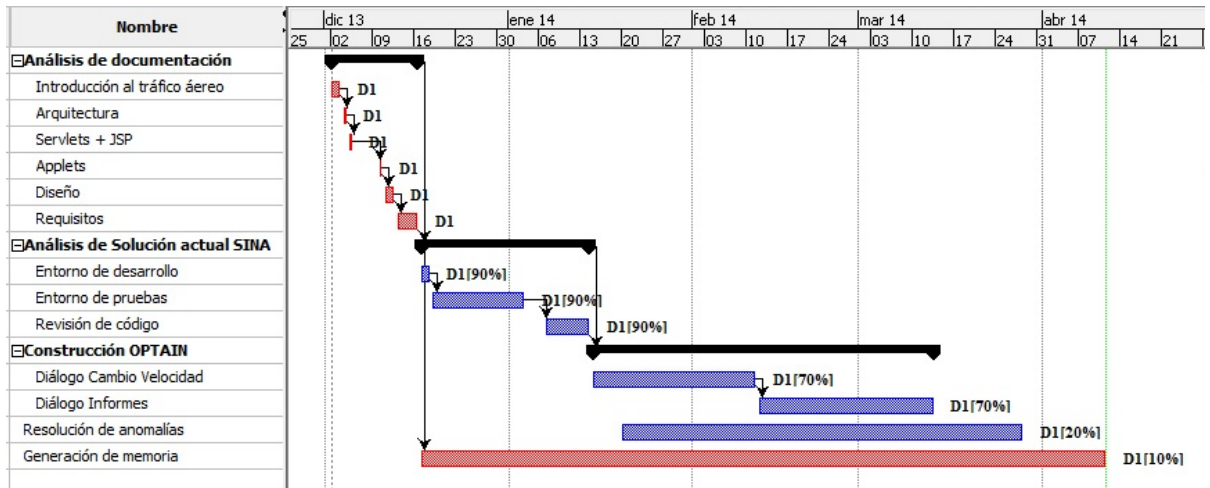


Ilustración 2: Planificación temporal

## 2. Presupuesto

En este documento se recoge el presupuesto detallado del proyecto. El coste que supone los recursos humanos, el software y el hardware necesario.

El coste estimativo que se presenta a continuación no incluye el IVA. que se tiene en cuenta al final del documento.

### 2.1. *Presupuesto detallado*

#### 2.1.1. Recursos humanos

<b>Id Unidad</b>	<b>Categoría</b>	<b>Número de personas</b>	<b>Unidad de medición</b>	<b>Nº de unidades (horas)</b>	<b>Precio unitario (€)</b>	<b>Importe total (€)</b>
HU01	Programador junior	1	Horas	410	20,00	8.200
Subtotal recursos humanos:						8.200,00

**Tabla 4: Estimación de recursos humanos**

### 2.1.2. Recursos software

<b>Id Unidad</b>	<b>Descripción</b>	<b>Unidad de medición</b>	<b>Nº de unidades</b>	<b>Precio unitario (€)</b>	<b>Importe total (€)</b>
SW01	Windows 7 Enterprise SP1 <sup>1</sup>	Unidades	1	26,00	26,00
SW02	Microsoft Office 2007 <sup>2</sup>	Unidades	1	68,00	68,00
SW03	Eclipse Indigo	Unidades	1	-	-
SW04	JRE 1.6_0_37	Unidades	1	-	-
SW05	yEd Graph Editor	Unidades	1	-	-
SW06	Openproj-1.4	Unidades	1	-	-
Subtotal software:					94,00

**Tabla 5: Estimación de recursos software**

<sup>1</sup> Amortización de la licencia durante el desarrollo del proyecto

<sup>2</sup> Amortización de la licencia durante el desarrollo del proyecto

### 2.1.3. Recursos hardware

<b>Id Unidad</b>	<b>Descripción</b>	<b>Unidad de medición</b>	<b>Nº de unidades</b>	<b>Precio unitario (€)</b>	<b>Importe total (€)</b>
HW01	PC HP <sup>3</sup>	Unidades	1	350,00	350,00
Subtotal hardware:					350,00

**Tabla 6: Estimación de recursos hardware**

---

<sup>3</sup> Amortización del PC durante la ejecución del proyecto



## 2.2. Presupuesto total

El presupuesto sin IVA que supone los recursos humanos, el software y el hardware necesario resultaría el siguiente:

<b>Id Unidad</b>	<b>Descripción</b>	<b>Precio (€)</b>
TOT01	Subtotal recursos humanos	8.200,00
TOT02	Subtotal software	94,00
TOT03	Subtotal hardware	350,00
Total:		8.644,00

**Tabla 7: Presupuesto total sin I.V.A.**

Por último añadimos los costes generales asociados a una persona, un porcentaje estimativo de beneficio industrial y el impuesto sobre el valor añadido para obtener el presupuesto total.

<b>Concepto</b>	<b>Euros</b>
Presupuesto de ejecución de material	8.644,00
Costes generales asociados a la persona	270,00
Beneficio industrial (35%)	3.119,90
I.V.A. (21%)	2.527,12
Presupuesto total con I.V.A.	14.561,02

**Tabla 8: Presupuesto total**

El presupuesto asciende a un total de 14.561,02 € (CATORCE MIL QUINIENTOS SESENTA Y UNO CON DOS CÉNTIMOS), que incluye IVA.

Gijón, Junio 2014



**UNIVERSIDAD DE OVIEDO**  
**ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**“IMPLEMENTACIÓN Y MANTENIMIENTO DE LA  
HERRAMIENTA DE APOYO A LA DECISIÓN OPTAIN-SA PARA  
CONTROL DE TRÁFICO AÉREO”**

**DOCUMENTO N° III**

**DOCUMENTOS TÉCNICO-CIENTÍFICOS**



**ANDREA MENÉNDEZ GARCÍA**  
**JUNIO 2014**

**LENGUAJES Y SISTEMAS  
INFORMÁTICOS**  
**TUTOR: PABLO JAVIER TUYA  
GONZÁLEZ**

## Tabla de contenidos

Tabla de contenidos.....	2
Índice de ilustraciones.....	4
Índice de tablas.....	6
1. Establecimiento del alcance del sistema.....	7
1.1. Objetivos.....	7
1.2. Usuarios.....	7
2. Terminología.....	8
2.1. Términos técnicos.....	8
2.2. Glosario.....	9
3. Especificación del sistema.....	10
3.1. Descripción general de la solución a implementar.....	10
3.2. Descripción técnica de SINA.....	13
3.2.1. Vista general.....	13
3.2.2. Páginas estáticas.....	14
3.2.3. Módulo de acceso a base de datos.....	14
3.2.4. Servlets.....	15
3.2.5. Applets.....	15
3.3. Arquitectura del servidor SINA.....	18
4. Requerimientos funcionales del procedimiento OPTAIN-SA.....	19
4.1. Selección de vuelos candidatos OPTAIN-SA.....	19
4.2. Ventana OPTA-IN.....	20
4.3. Tabular secuenciador.....	24
4.4. Velocidad propuesta por el procedimiento OPTAIN-SA.....	25
4.5. Implementación del procedimiento OPTAIN-SA a partir de SINA.....	26
4.6. Matriz de trazabilidad de requisitos.....	32
4.7. Corrección de problemas.....	33
5. Diseño del sistema.....	39
5.1. Escenarios.....	39
5.1.1. Descripción de los escenarios.....	40
5.2. Secuencia de acciones.....	42
5.3. Arquitectura de OPTAIN-SA.....	44
5.4. Modelo de datos del sistema.....	45
5.5. Clases involucradas en el sistema.....	50
6. Construcción.....	55
6.1. Preparación del entorno de generación y construcción.....	55

6.1.1.	Herramientas .....	55
6.1.2.	Equipo de trabajo .....	56
7.	Manual de usuario .....	57
7.1.	Iniciando una sesión .....	57
7.2.	Finalizando una sesión .....	60
7.2.1.	Funcionalidades de la herramienta .....	61
7.2.2.	Ventana OPTA-IN .....	61
7.2.3.	Activación de un procedimiento OPTA-IN: .....	64
7.2.4.	Selección de velocidad para aplicar un procedimiento: .....	66
7.2.5.	Cancelación de un procedimiento OPTA-IN .....	67
7.2.6.	Ventanas secundarias de 5 y 10 NM .....	68
7.2.7.	Ojeo .....	70
7.2.8.	Excluir e Incluir vuelos .....	70
7.2.9.	Generar Informes .....	72
8.	ANEXO .....	76
8.1.	Tablas de velocidad .....	76
8.2.	Integración con el sistema existente .....	77
8.2.1.	OptainDialogoVelocidad .....	77
8.2.2.	OptainDialogoInforme .....	78
8.3.	Implementación .....	79
8.3.1.	Funciones selección de velocidad .....	79
8.3.2.	Código fuente selección de velocidad .....	80
8.3.3.	Funciones generación de informes .....	83
9.	Bibliografía y referencias .....	100

## Índice de ilustraciones

Ilustración 1: Ventana de login de SINA .....	11
Ilustración 2: HMI del servicio SINA de implementación de OPTAIN-SA .....	12
Ilustración 3: Arquitectura del servidor SINA.....	18
Ilustración 4: Área geográfica OPTAIN-SA.....	20
Ilustración 5: Ventana de selección de velocidad.....	23
Ilustración 6: Tabular secuenciador 5NM.....	25
Ilustración 7: Tabular secuenciador 10NM.....	25
Ilustración 8: Ventana OPTAIN.....	27
Ilustración 9: Tabular de secuencia de llegada .....	28
Ilustración 10: Ventana Informe .....	29
Ilustración 11: Panel de control de la ventana principal del servicio SINA de implementación de OPTAIN.....	30
Ilustración 12: Ventana principal del servicio SINA de implementación de OPTAIN .....	30
Ilustración 13: Escenarios .....	39
Ilustración 14: Secuencia de acciones.....	42
Ilustración 15: Arquitectura de OPTAIN-SA .....	44
Ilustración 16: Modelo de datos del sistema.....	46
Ilustración 17: Modelo de datos del sistema.....	47
Ilustración 18: Modelo de datos del sistema.....	48
Ilustración 19: Modelo de datos del sistema.....	49
Ilustración 20: OptainClientApplet.....	50
Ilustración 21: OptainCuerpoSecuenciador .....	51
Ilustración 22: OptainTituloSecuenciador .....	51
Ilustración 23: OptainPistaSecuenciador .....	51
Ilustración 24: OptainDialogoVelocidad .....	52
Ilustración 25: OptainDialogoInforme.....	52
Ilustración 26: TablaResultados.....	53
Ilustración 27: NumericTextField.....	53
Ilustración 28: CabeceraResultados .....	53
Ilustración 29: OptainSecuenciador.....	54
Ilustración 30: Icono de la aplicación .....	57
Ilustración 31: Ventana de acceso a la herramienta.....	57
Ilustración 32: Botón “Pulse para entrar” .....	57
Ilustración 33: Ventana de login .....	58
Ilustración 34: Botón “Enviar”.....	58
Ilustración 35: Ventana de alerta de seguridad.....	58
Ilustración 36: Detalle de la ventana de alerta.....	59
Ilustración 37: Ventana principal de OPTA-IN .....	59
Ilustración 38: Botón para cerrar la aplicación OPTA-IN.....	60
Ilustración 39: Combinación de teclas para cerrar la aplicación OPTA-IN.....	60
Ilustración 40: Ventana de regla vertical OPTA-IN .....	61
Ilustración 41: Detalle del TIE Point .....	62
Ilustración 42: Detalle del tabular .....	62
Ilustración 43: Ejemplos de tabulares de velocidad.....	63
Ilustración 44: Ejemplos de tabulares de velocidad.....	63
Ilustración 45: Ejemplo de vuelos en cada uno de los estados .....	63
Ilustración 46: Pareja de vuelos candidatos .....	64

Ilustración 47: Pareja de vuelos aplicando el procedimiento OPTA-IN .....	65
Ilustración 48: Detalle de las velocidades resaltadas y atenuadas .....	65
Ilustración 49: Ventana de selección de velocidad .....	66
Ilustración 50: Actualización del estado de los vuelos al excluir uno de ellos .....	67
Ilustración 51: Actualización del estado de los vuelos al incluir un vuelo que estaba excluido .....	68
Ilustración 52: Botones de “5NM” y “10NM” .....	68
Ilustración 53: Ventana secundaria de 5NM.....	69
Ilustración 54: Ventana secundaria de 10NM.....	69
Ilustración 55: Botón para cerrar las ventanas de 5NM y 10NM .....	70
Ilustración 56: Botón “Ojeo” seleccionado.....	70
Ilustración 57: Botón “Ojeo” sin seleccionar.....	70
Ilustración 58: Tabular de vuelo seleccionado.....	71
Ilustración 59: Botón para excluir/incluir vuelos.....	71
Ilustración 60: Ejemplo de actualización de las velocidades propuestas .....	71
Ilustración 61: Botón “Informe” .....	72
Ilustración 62: Ventana de Informe.....	72
Ilustración 63: Generar informe con filtro por fecha .....	73
Ilustración 64: Generar informe con filtro por TIE Point .....	73
Ilustración 65: Generar informe con filtro por distancia en el IAF .....	73
Ilustración 66: Botón “Limpiar” .....	74
Ilustración 67: Ejemplo de ventana informe con búsqueda con filtro por fecha .....	74
Ilustración 68: Botón “Exportar” .....	74
Ilustración 69: Fichero Excel de informe generado .....	75
Ilustración 70: Botón “Cancelar” para cerrar la ventana de informe.....	75
Ilustración 71: Botón para cerrar la ventana de informe .....	75

## Índice de tablas

Tabla 1: Características equipo de trabajo .....	56
Tabla 2: Ajuste de velocidad para separación de 5NM .....	76
Tabla 3: Ajuste de velocidad para separación de 10 NM .....	76

## **1. Establecimiento del alcance del sistema**

### **1.1. *Objetivos***

El objetivo del procedimiento OPTAIN-SA es ayudar al controlador de tráfico aéreo en el proceso de proporcionar instrucciones de ajuste de velocidad a los pilotos de las aeronaves en la secuencia de llegada al Fijo de Aproximación Inicial (IAF), para el mismo o distintos flujos de tráfico, de forma que se garantice la separación longitudinal requerida entre las aeronaves.

### **1.2. *Usuarios***

Sólo habrá un rol para los usuarios finales y es el de controlador de tráfico aéreo. Su función es la de distribuir el espacio aéreo y separar aeronaves que pretenden utilizarlo. Estos controladores estarán situados en los Centros de Operaciones y Torres de Aeródromos y estarán comunicados con los pilotos.

Los controladores se apoyarán en la información suministrada por la aplicación OPTAIN-SA (entre otras) que interpretarán y gestionarán para tomar las decisiones oportunas y transmitirán a los pilotos para llevarlas a cabo.

La mayor prioridad es garantizar la Seguridad y proporcionar Orden y Rapidez al tráfico aéreo.



## 2. Terminología

Se incluyen a continuación un conjunto de términos técnicos [IACTA] así como una serie de acrónimos a los que se hace referencia en el presente documento.

### 2.1. Términos técnicos

- *Plan de vuelo*: Información de un vuelo ordenada en campos con todos los datos necesarios que definen el vuelo.
- *Pista radar*: Una pista de vuelo o traza proporciona en cada momento, la posición, la velocidad sobre el suelo y la velocidad vertical de una aeronave, de la manera más fiable y exacta posible, minimizando los errores de la detección radar.
- *Fases de vuelo*: una aeronave comercial, desde su salida hasta su llegada, sigue las siguientes fases de vuelo.
  - *Prevuelo*: Esta fase se inicia en tierra del aeródromo origen, desde su estacionamiento hasta la pista de despegue, incluyendo los chequeos previos de la aeronave y de la confirmación de su plan de vuelo, su puesta en marcha, en algunos casos un retroceso remolcado y rodaje hasta la pista de despegue.
  - *Despegue*: La aeronave acelera su carrera en pista hasta que adquiere una velocidad que le permite la sustentación necesaria para ascender.
  - *Salida*: Tras el despegue, la aeronave sigue ascendiendo hasta una altitud de crucero siguiendo un procedimiento o ruta de salida que finaliza en un punto de conexión con una aerovía principal.
  - *En ruta*: Alcanzada al aerovía inicial, el vuelo continúa la ruta establecida, cruzando varios espacios aéreos hasta las cercanías del aeropuerto destino.
  - *Llegada*: Desde el último punto de su ruta, el vuelo inicia un descenso a través de un procedimiento o ruta de llegada hasta alcanzar las proximidades del aeropuerto destino.
  - *Aproximación*: Donde se ejecuta un procedimiento de aproximación que alinea la aeronave con la pista de aterrizaje designada del aeropuerto destino.
  - *Aterrizaje*: La aeronave desciende suavemente, toma tierra sobre la pista, decelera dejando la pista y rueda por calles hasta su aparcamiento en la terminal del aeródromo.
- Dentro del espacio controlado de un FIR (Flight Information Region), existen a su vez diferentes zonas de control específicas como:
  - *ATZ*: Zona de Tráfico de Aeródromo es la zona de responsabilidad de la Torre de Control (TWR). Es un espacio aéreo pequeño de forma cilíndrica, que arranca desde el suelo y de altura dependiente de la visibilidad y que está centrado en un punto llamado ARP (Airport Reference Point), y cuyo radio suele ser de 5 NM.
  - *CTR*: Zona de Control de Aeródromo es una zona más grande que el ATZ y que engloba una o varias zonas ATZ cercanas y hasta una altitud de transición del vuelo, bajo la jurisdicción del personal dedicado al control de aproximación y que puede hacerse cargo de las funciones de TWR.

- *TMA*: Área de control terminal, es un área más grande que el CTR que lo rodea y cubre desde la altitud de transición hasta el límite vertical del FIR (región de información del vuelo), bajo responsabilidad del control de aproximación. Es el espacio aéreo donde confluyen las aerovías próximas a uno o más aeropuertos y se enlaza la fase de vuelo en ruta con la de aproximación y viceversa.

## **2.2. Glosario**

- *AENA*: Aeropuertos Españoles y Navegación Aérea
- *AIP*: Aeronautical Information Publications
- *ARP*: Airport Reference Point
- *ATC*: Air Traffic Control
- *ATZ*: Airport Traffic Zone
- *CTR*: Aerodrome Control Zone
- *DISPI*: DIstribución de PIstas Radar
- *FIR*: Flight Information Region
- *GIPV*: Gestión de Información de Plan de Vuelo
- *HMI*: Human Machine Interface
- *IAF*: Initial Approach Fix
- *IAS*: Indicated Air Speed
- *IBM*: International Business Machines
- *JDT*: Java Development Toolkit
- *JSP*: Java Server Pages
- *LEPA*: Aeropuerto de Palma de Mallorca
- *NM*: Millas Náuticas
- *OPTAIN-SA*: Optimized Profile Descent Approaches Implementing Windows-Speed Advisory
- *SACTA*: Sistema Automático de Control de Tráfico Aéreo
- *SDP*: Surveillance Data Processing
- *SESAR*: Single European Sky ATM Research
- *SINA*: Sistema de Información de Navegación Aérea
- *SIS*: Mensaje del Sistema
- *TCPV*: Tratamiento Central de Plan de Vuelo
- *TMA*: Terminal Manoeuvre Area
- *TWR*: ToWeR, Torre de Control

### 3. Especificación del sistema

Este apartado contiene la especificación de implementación del procedimiento OPTAIN-SA. En primer lugar se describe la solución general a implementar y a continuación se hace una descripción técnica más en detalle de la aplicación SINA (Sistema de Información de Navegación Aérea).

#### **3.1. Descripción general de la solución a implementar**

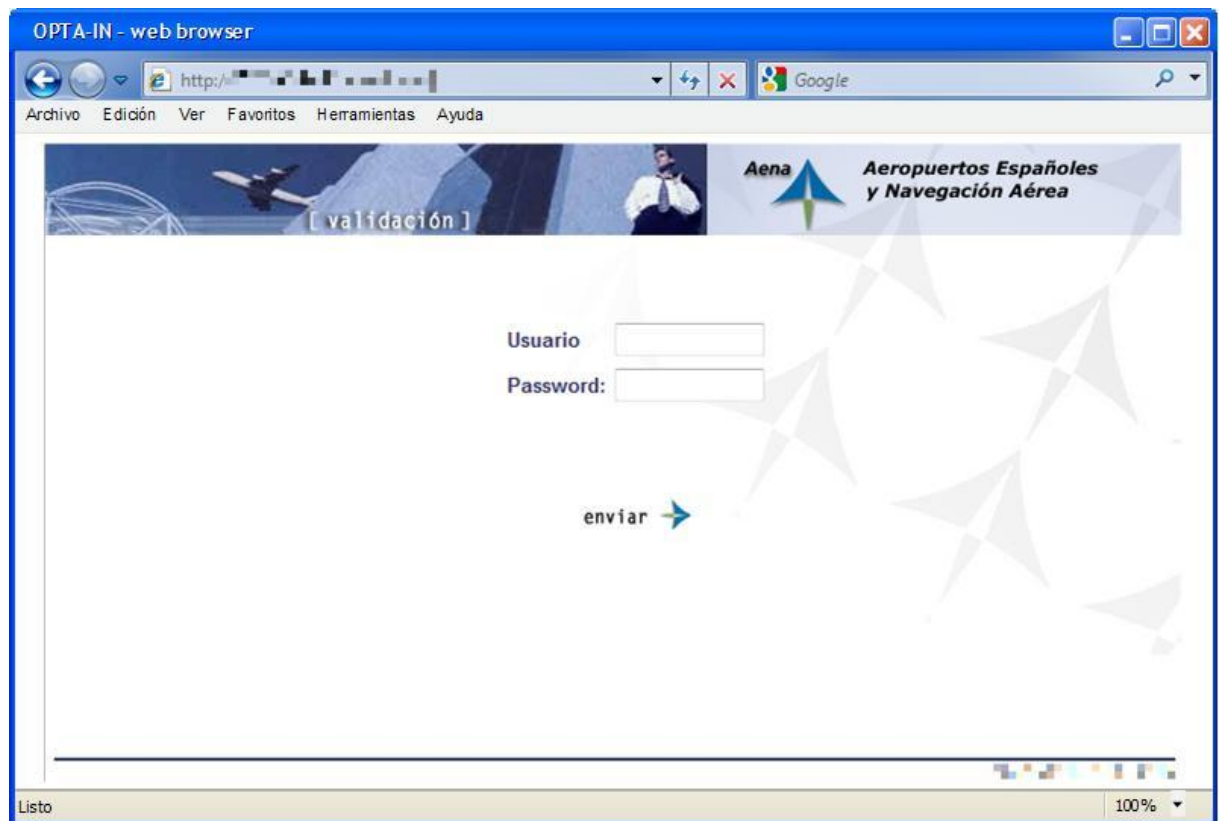
Se presenta una propuesta de implementación de la funcionalidad del procedimiento OPTAIN-SA dentro la aplicación SINA como un nuevo servicio para los usuarios de la misma.

El objetivo de la aplicación SINA es el de ofrecer a sus usuarios una serie de servicios con funciones de explotación de procesamiento de datos de vigilancia (SDP, Surveillance Data Processing) y de datos de Plan de Vuelo Central (TCPV, Tratamiento Central de Plan de Vuelo) para aquellos vuelos pertenecientes a las compañías aéreas y aeropuertos de destino configurados previamente para cada uno de los usuarios de acceso.

Para ello, la aplicación SINA se conecta al sistema SACTA (Sistema Automático de Control de Tráfico Aéreo) a través de los siguientes servidores como un cliente externo, y de los que obtiene información en tiempo real de pistas y planes de vuelo.

- *Servidor SIS (Mensaje del Sistema)*. Permite la captura de pistas radar en tiempo real sobre una zona geográfica previamente definida en los alrededores de un aeropuerto.
- *Servidor GIPV (Gestión de Información de Plan de Vuelo)*. Permite la obtención de Planes de Vuelo Central cuyo aeródromo de destino conste entre los previamente configurados, y de los cuales se reciben actualizaciones a través de eventos de modificación, terminación y cancelación.

Un servidor web conectado al sistema SACTA estará disponible para ser utilizado por los ordenadores clientes conectados a una Intranet de AENA (Aeropuertos Españoles y Navegación Aérea) por medio de un navegador de Internet. El acceso sólo se permite a los usuarios que utilicen una combinación de usuario autorizado y contraseña, de acuerdo al servicio SINA al que se pretenda acceder.



**Ilustración 1: Ventana de login de SINA**

Los servicios actualmente definidos para la aplicación SINA se exponen a continuación:

- Servicio de presentación gráfica de pistas radar. Representa gráficamente sobre un mapa geográfico las pistas de vigilancia en el área de interés definida en torno al aeropuerto configurado, que complementa con tabulares con información de Plan de Vuelo asociada a las pistas presentadas.
- Servicio secuenciador de llegadas por pista. Presenta información de aproximación de vuelos a un aeropuerto dado, para cada una de las configuraciones de llegada consideradas.
- Servicio de presentación de información de Plan de Vuelo. Permite al usuario realizar consultas de búsqueda de Planes de Vuelo de acuerdo a una serie de parámetros de filtrado con el fin de obtener información detallada de los mismos.

La solución propuesta pasa por la especificación de una nueva herramienta ATC (Air Traffic Control) que a través una de Ventana OPTA-IN mostrará gráficamente la secuencia de vuelos de llegada al IAF del aeropuerto LEPA (Aeropuerto de Palma de Mallorca) desde los flujos de tráfico correspondientes, sobre una regla de distancia vertical, junto con el ajuste de velocidad dado por el procedimiento para las dos separaciones consideradas (5NM/10NM).

Asimismo, se podrá acceder a su vez a ventanas auxiliares con información de los vuelos de dicha secuencia en formato tabular, particularizada la velocidad propuesta por el procedimiento para una sola de las dos separaciones.

El nuevo servicio SINA también estará provisto de una ventana de búsqueda de vuelos que hayan aplicado la velocidad dada por el procedimiento OPTAIN-SA, y para la que se definirán distintas opciones de filtrado para configurar la búsqueda. Asimismo, será posible exportar los resultados representados en dicha ventana a un fichero en formato Excel.

La siguiente figura muestra el HMI (Human Machine Interface) completo del nuevo servicio SINA que implementará la funcionalidad del procedimiento OPTAIN-SA.

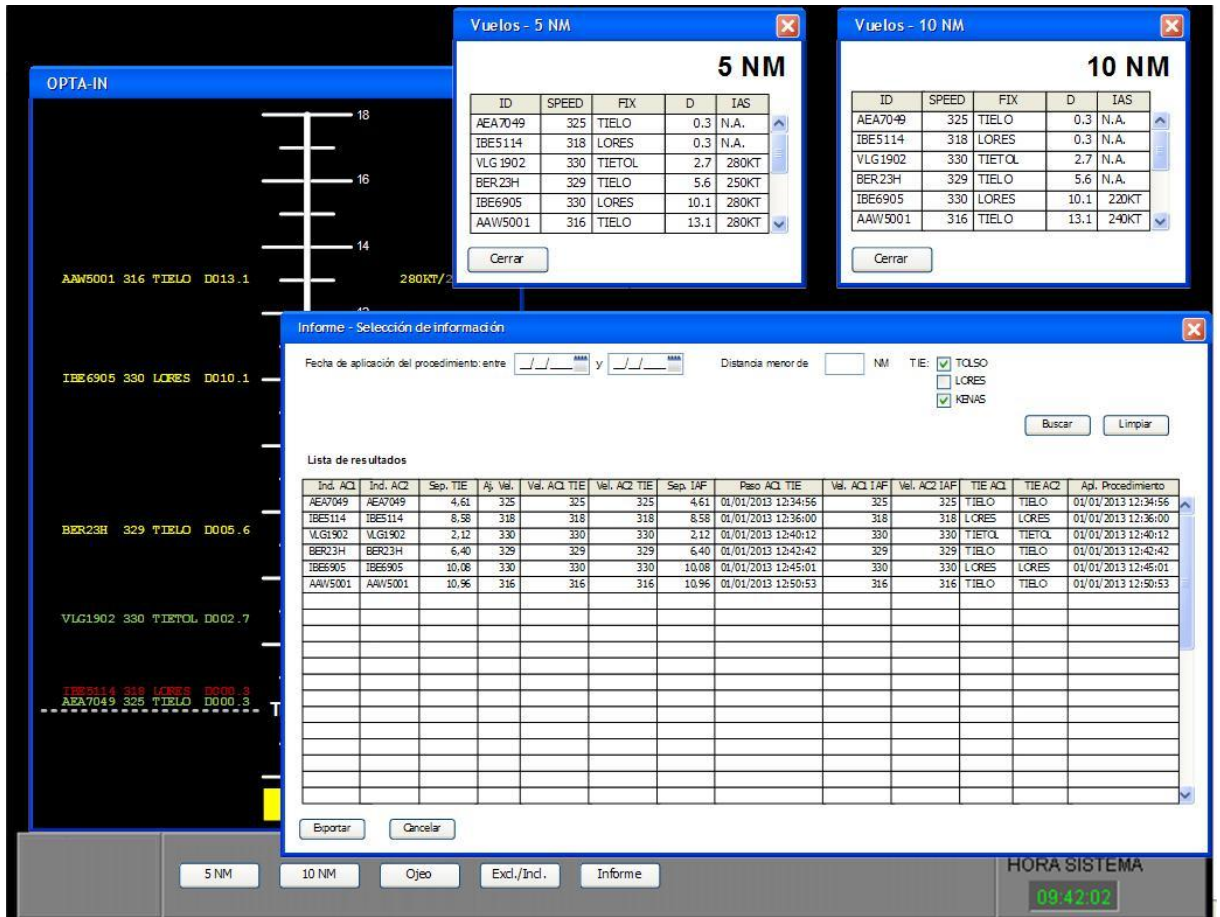


Ilustración 2: HMI del servicio SINA de implementación de OPTAIN-SA

### **3.2. Descripción técnica de SINA**

Se presenta a continuación una descripción técnica del Sistema de Información de Navegación Aérea (SINA) [DST].

El sistema SINA permite observar a través de un navegador Web (HTTP), información sobre las pistas radar obtenidas a través del sistema DISPI (Distribución de Pistas Radar), así como sobre los planes de vuelo obtenidos a través del sistema GIPV (Gestión de Información de Plan de Vuelo).

La presentación de esta información se puede realizar de diferentes formas, siendo posibles en la actualidad:

- *Consulta de vuelos*: presenta información limitada sobre un plan de vuelo, accediendo sólo a tablas con información GIPV.
- *Seguimiento de un vuelo*: se presenta gráficamente sobre mapas geopolíticos la evolución estimada de un vuelo a partir de su plan de vuelo.
- *Aviso de llegada de un vuelo*: en el que después de ver información sobre un plan de vuelo, se puede solicitar que se envíe un mensaje (SMS o correo electrónico) con un tiempo de antelación sobre la llegada del vuelo.
- *Información detallada del plan de vuelo*: presenta toda la información de un plan de vuelo, de forma alfanumérica o gráfica. En algunos momentos se puede solicitar información sobre la pista radar asociada al plan de vuelo.
- *Lista de llegadas para un aeródromo*: se presenta en forma de tabular, información a partir de los planes de vuelo sobre las próximas llegadas a un aeródromo determinado.
- *Tráfico aéreo en el entorno de un aeródromo*: se presenta información suministrada por DISPI de las pistas radar en el entorno de un aeródromo y que cumplan con unos filtros. También se tiene acceso a información sobre planes de vuelo a través de 2 tabulares (vuelos previstos y vuelos terminados).
- *Secuencia de llegadas a un aeródromo*: presenta información no geográfica a partir de los datos DISPI. También se tiene acceso a información sobre planes de vuelo a través de dos tabulares (vuelos previstos y vuelos terminados).

#### **3.2.1. Vista general**

En general, la aplicación SINA se divide en varias partes o módulos diferenciados. Estos módulos son:

- Páginas HTML estáticas.
- Clases o módulo para interactuar con la base de datos. Este módulo está duplicado, de tal forma que es factible el funcionamiento independiente de ambas partes.
- Servlets que tramitan las peticiones de los clientes.
- Applets que visualizan la información en los clientes.

Otra peculiaridad de SINA a tener en cuenta, es la diferenciación de los llamados servicios anónimos respecto a los restringidos, ya que en los primeros el diseño buscaba la no utilización de recursos para los distintos usuarios (ni siquiera sesiones en el servidor) de forma que no hubiera peligro de falta de recursos al crecer el número de usuarios. Por el contrario, los restringidos hacen bastante uso de sesiones y memoria en el servidor.

En general, cuando el cliente entra en la dirección correspondiente a la aplicación, recibe páginas estáticas (como la de bienvenida), para después navegar a través de servlets que redirigen a páginas HTML (aunque en la parte anónima esta redirección es menos rígida, y por tanto menos definida la separación vista-código).

En las páginas que exista un applet Java, estos se comunican con el servidor a través de servlets que utilizan las clases de acceso a datos y clases auxiliares.

### **3.2.2. Páginas estáticas**

En realidad la inmensa mayoría de estas páginas tienen código (son páginas JSP, Java Server Pages). Estas páginas son compiladas por el motor de JSP a servlets la primera vez que se ejecutan. Dichos servlets con cada llamada producen el HTML final. Incluyen la página de bienvenida, la de identificación, las de búsqueda y otras muchas.

### **3.2.3. Módulo de acceso a base de datos**

Este módulo está duplicado en su mayoría, por lo que en realidad existen dos pools de conexiones con la base de datos y similarmente el resto de la estructura.

La base de datos Oracle es alimentada por los clientes DISPI y GIPV, y utilizada por estas clases para obtener la información requerida. Estas clases acceden a varias tablas de dicha base de datos, como la tabla PISTAS\_DISPI y la tabla PVS. En la primera se encuentran los datos dinámicos de las pistas, mientras que en la segunda hay información sobre los planes de vuelo.

También se accede a las tablas necesarias para poder dibujar los mapas como son las tablas ELEMENTOS\_DIBUJO y PUNTOS\_MAPA, y a las necesarias para la autenticación del usuario (USUARIOS Y CATEGORIAS). Además, este módulo también incluye diversas clases de apoyo.

Las principales clases que componen este módulo son:

#### **jdbconnectionpool**

Clase que crea y administra un pool de conexiones con la base de datos, de forma que cada conexión se reutilice sin necesidad de crearla y destruirla con cada consulta a la base de datos. De esta forma se consigue optimizar el tiempo de acceso, ya que la operación de crear una conexión es lenta, además de evitarse la creación de excesivas conexiones que pudieran causar un overhead del sistema.

#### **SACtadbdriverimpl**

Esta clase es la implementación del interface SACTADBDriver, y contiene toda la información específica para el acceso a la BD, de forma que sus métodos devuelven información contenida en ésta. Por ejemplo, en esta clase están contenidas las sentencias SQL que se utilizan.

## **sactadbmanager**

Esta clase hace de interface entre los servlets y el acceso a la BD, siendo también la que contiene el hilo que periódicamente renueva información general (como las listas de pistas y tabulares en una de las partes o los aeródromos y compañías para búsqueda anónima en la otra).

### **3.2.4. Servlets**

Los servlets son clases Java un tanto especiales, de forma que manejan una petición HTTP realizada por un cliente. La mayoría de los servlets que componen SINA se pueden clasificar como servlets de servidor de applet o servlets de redirección a JSP.

Los primeros serían los que heredan de SACTAHttpChannelServlet (que define la base para la comunicación applet-servlet), y su principal función es la de responder a las distintas peticiones que le llegan desde el applet. Esa respuesta habitualmente consiste en un objeto serializado.

Los segundos son los que heredan de SACTAFrameworkServlet que incluye métodos generales de redirección y acceso a la sesión. Estos servlets tienen asociada una página JSP a través del fichero de configuración (sactaservlets.properties), que es a la que se dirige si no se le indica al servlet a través del parámetro TASK alguna otra tarea a realizar.

Existen otros servlets que no se incluyen en ninguna de las categorías anteriores, ya que aunque podrían clasificarse como servlets de redirección a JSP, al no manejar la sesión (son completamente anónimos) no heredan SACTAFrameworkServlet.

### **3.2.5. Applets**

Los applets son clases Java que se ejecutan en el cliente. La mayoría de los applets de SINA realizan una comunicación con el servidor, salvo el applet de seguimiento, que una vez cargado ya no se comunica con el servidor y un pequeño applet que se dedica a cargar clases asíncronamente para disminuir la carga de clases posterior. Por supuesto, muchos de ellos además de dicha comunicación realizan una representación de información.

#### **3.2.5.1. Entrada**

La entrada se realiza a través de la JSP index.jsp (a la que se llega a través de una redirección JavaScript desde index.htm del servidor Web). Tras mostrar la imagen de bienvenida se pasa directamente a login.jsp desde la que se puede seleccionar los servicios anónimos o introducir un usuario y contraseña.

#### **3.2.5.2. Seguimiento de vuelo**

Al pulsar sobre el enlace se pasa a filtrajeAn.jsp en la que se dan varias opciones de búsqueda (las opciones dadas dependen del servicio seleccionado). Para el caso del seguimiento de vuelo se muestran todas las opciones de búsqueda que son:

- *Búsqueda por compañía y número de vuelo*, que lleva a planes\_vueloAn1.jsp, en la que se muestran los resultados de la búsqueda (lanzada desde la misma página).



- *Búsqueda por origen*, destino y horario, que llevará a planes\_vueloAn2.jsp, en la que también se muestra el listado de vuelo que cumplen los criterios.
- *Búsqueda aleatoria*, que lleva a planes\_vueloAn3.jsp.

Desde estas tres páginas automáticamente si la búsqueda ha llevado a un único resultado, o tras seleccionar un vuelo nos lleva a la parte específica de cada servicio anónimo.

En el caso del seguimiento de vuelo, nos lleva a seguimiento.jsp, página que nos muestra el applet `sacta.ftrack.MapClientApplet`, que únicamente se comunica con el servidor (a través de `sacta.servlet.MapServerServlet`) para cargar las clases, las imágenes y los datos del plan de vuelo.

### **3.2.5.3. Consulta de vuelos**

En este servicio no se tiene acceso a la búsqueda aleatoria. Tras obtener el listado de vuelos como en el apartado anterior, no hay ninguna otra opción posterior.

### **3.2.5.4. Aviso de llegada**

Este servicio también cuenta con los tres tipos de búsqueda, ya sea mediante la redirección automática si sólo hay un vuelo o al seleccionar uno del listado se llega a `aviso.jsp`, en la que se pueden introducir los criterios del aviso y a través de la cual se agrega el aviso a la base de datos.

### **3.2.5.5. Validación**

Para el resto de servicios se requiere la validación del usuario, que se realiza a través de `sacta.servlet.LoginServlet`, que según el nivel de autorización del usuario redirigirá a diferentes recursos.

### **3.2.5.6. Información de un vuelo**

Para este servicio y para el de lista de llegadas el servlet de autenticación nos redirige a `principal2.jsp`. Desde esta página, si seleccionamos este servicio pasamos a `filtraje.jsp` a través de `sacta.servlet.PVFilterServlet.java`. Tras seleccionar los criterios que precisemos, se pasa a `planes_vuelo.jsp` a través de `sacta.servlet.PVListServlet`.

En ella se muestra la lista de vuelo, que comprueba cambios en el servidor a través del applet `sacta.applet.PVListClientApplet` que se comunica con el servlet `sacta.servlet.PVListServerServlet`.

Desde esa lista de vuelos se puede seleccionar ver el mapa gráfico, que nos lleva a `mapdesktop.jsp` donde se muestra el applet `sacta.applet.MapClientApplet`, en contacto con el servidor a través de `sacta.servlet.MapServerServlet`.

Y la otra opción es ver la información de un plan de vuelo del listado, que nos lleva a `pv.jsp` a través de `sacta.servlet.PVServlet`. En esta página se muestra el applet `sacta.applet.PVAlfaClientApplet` que se comunica con el servidor a través del servlet `sacta.servlet.PVAlfaServerServlet`.

Si se seleccionan puntos y sectores se pasa a PVSectorPointAlfaP.jsp en la que se muestra el applet `sacta.applet.PVPointAlfaClientApplet`, que se comunica con el servidor a través del servlet `sacta.servlet.PVPointAlfaServerServlet`. Este applet muestra los puntos de ruta del plan de vuelo.

Si se seleccionan sectores se pasa a PVSectorPointAlfaS.jsp en la que se muestra el applet `sacta.applet.PVSectorAlfaClientApplet`, que se comunica con el servidor a través del servlet `sacta.servlet.PVSectorAlfaServerServlet`. Este applet muestra los sectores atravesados por el plan de vuelo.

Si se selecciona información gráfica se pasa a PVSectorPointGraficPS.jsp en la que se muestra el applet `sacta.applet.ProfileClientApplet`, que se comunica con el servidor a través del servlet `sacta.servlet.ProfileServerServlet`. Este applet muestra un perfil del plan de vuelo con información sobre los puntos y sectores.

Si se selecciona mapa de vuelo, se pasa a `bigmap.jsp` que muestra el applet `sacta.applet.MapClientApplet` que se comunica con el servidor vía el servlet `sacta.servlet.MapServerServlet`.

### **3.2.5.7. Lista de llegadas**

Al seleccionar este servicio se pasa a la página `llegadas.jsp` a través del servlet `sacta.servlet.LlegadasServlet`. En esta página se muestra el applet `sacta.applet.ListaApplet`, que se comunica con el servidor a través de `sacta.servlet.MapServerServlet`.

### **3.2.5.8. Tráfico aéreo sobre un aeródromo**

Al identificarse correctamente como un usuario con acceso a esta parte, se pasa a `indexprod.jsp` que abre a pantalla completa la página `mapa.jsp` que redirige a distintas páginas (por ejemplo `mapa800x600.jsp`), en la que se muestra el applet `prod.applet.MaClientApplet` con un tamaño adecuado para la resolución del cliente. Este applet se comunica con el servidor a través de `prod.servlet.MapServerServlet`.

### **3.2.5.9. Secuencia de llegadas a un aeródromo**

Al identificarse correctamente como un usuario con acceso a esta parte, se pasa a `indexsec.jsp` que abre a pantalla completa la página `mapasec.jsp` que redirige a distintas páginas (por ejemplo `mapasec800x600.jsp`), en la que se muestra el applet `prod.applet.SecuenciaApplet` con un tamaño adecuado para la resolución del cliente. Este applet se comunica con el servidor a través de `prod.servlet.MapServerServlet`.

### 3.3. *Arquitectura del servidor SINA*

La Arquitectura del Servidor SINA está compuesta por una máquina SUN ULTRA 60 conectada al Tándem del SIS y al Tándem del GIPV mediante el LAN de Usuarios del Sistema SACTA.

Mientras que la distribución externa de pistas SACTA se realiza por el Centro de Control, la Distribución de Planes de Vuelo se realiza a nivel de Tratamiento Central.

Los clientes PC's SINA están conectados a la Red Corporativa de AENA (RECOA) y entre las dos Redes (usuarios SACTA y RECOA) existe un Fire-Wall que previamente configurado sólo deja pasar tráfico de Red por puertos específicos a máquinas específicas.

En la actualidad se dispone de un Servidor SINA en cada uno de los centros de control de España, estos son, Madrid, Barcelona, Palma de Mallorca, Sevilla y Gran Canaria.

Del SIS se obtiene la información de vigilancia, es decir, la información radar, y del GIPV se obtienen los planes de vuelo.

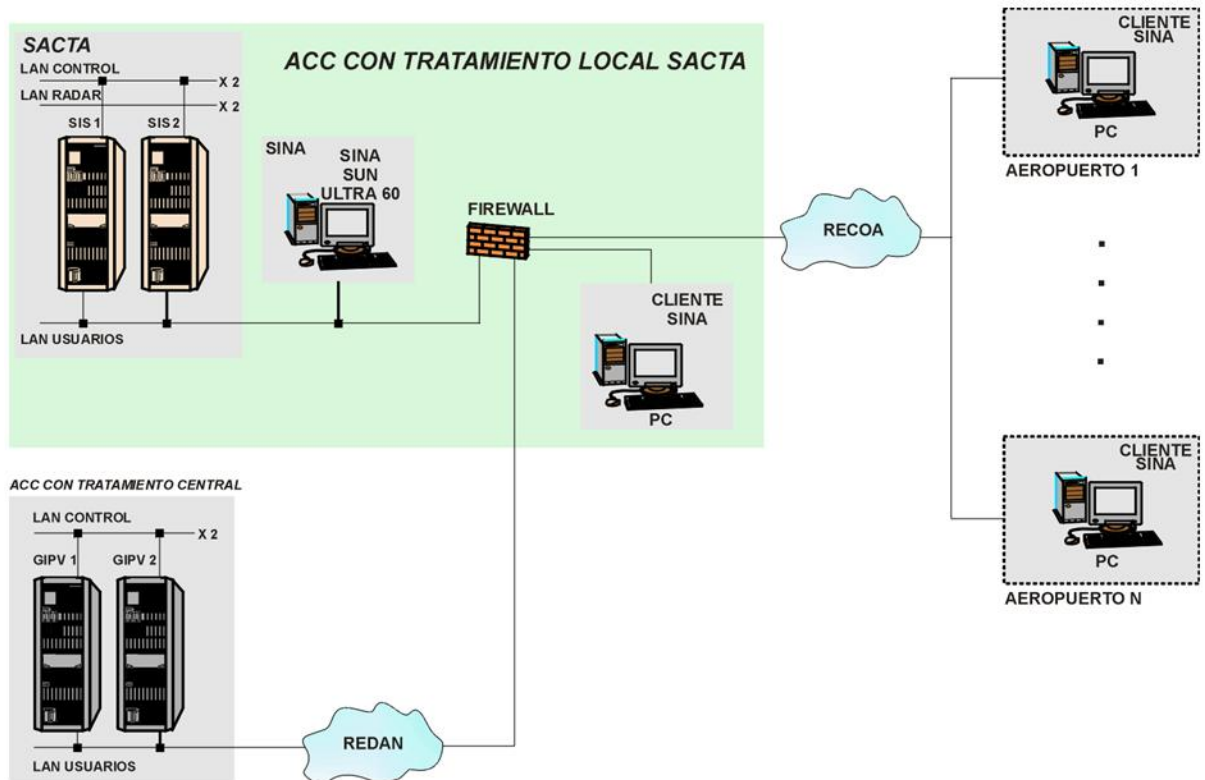


Ilustración 3: Arquitectura del servidor SINA

## 4. Requerimientos funcionales del procedimiento OPTAIN-SA

Los requisitos hacen referencia a una serie de requisitos que define el programa europeo SESAR (Single European Sky ATM Research) [OPDAIW] para una herramienta ATC OPTA-IN.

De ahí su doble numeración en algunos casos donde REQ-X hace referencia a la numeración actual del presente documento y REQ-OPTA-INX que hace referencia al requisito X del programa europeo.

SESAR es un proyecto conjunto de la comunidad de transporte aéreo europeo cuyo objetivo es la implantación de una red ATM europea de altas prestaciones. Este proyecto nació de la necesidad de crear una visión integrada y común sobre la evolución del sistema de gestión del tránsito aéreo, con el objeto de acomodar, a través de la implantación de nuevos procedimientos y tecnologías, el incremento de demanda previsto para los próximos años.

Se describen los treinta y seis requisitos funcionales del procedimiento OPTAIN-SA [EDPO] y a continuación se describen los dieciocho requisitos a partir de los problemas detectados tras la primera entrega de la aplicación.

### 4.1. Selección de vuelos candidatos OPTAIN-SA

**REQ-1 (REQ-OPTA-IN7)**. Se define el área geográfica de selección de vuelos candidatos de OPTAIN-SA como una corona circular comprendida entre dos radios, con centro en el IAF y ángulos inicial y final definidos.

- El ángulo inicial se establecerá en el radial 280°
- El ángulo final se establecerá en el radial 10°

El radio menor de la corona circular será de 31 NM (la distancia entre el IAF y los puntos TIE) y el radio mayor será de 49 NM (31+18).

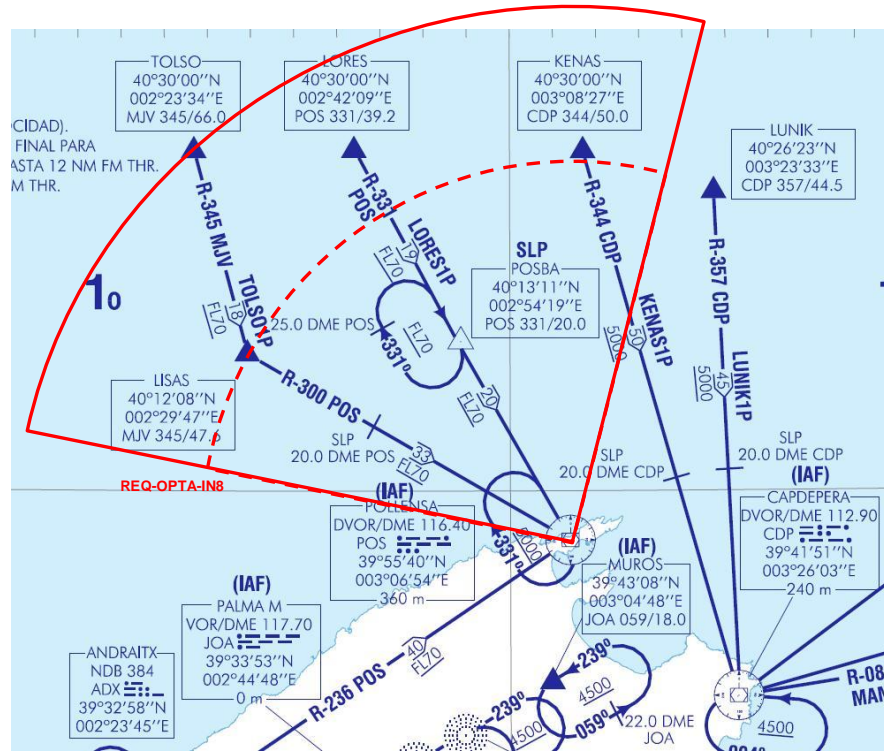


Ilustración 4: Área geográfica OPTAIN-SA

**REQ-2 (REQ-OPTA-IN1, REQ-OPTA-IN8).** La función de selección de vuelos de OPTAIN-SA determinará los posibles candidatos a ser secuenciados de forma automática en función de los siguientes criterios:

- La pista multirradar asociada al vuelo se encuentre dentro de la corona circular definida por **REQ-1**.
- El vuelo pasa por el IAF de Pollensa (POS) y su destino es el aeropuerto de Palma de Mallorca (LEPA).

**REQ-3 (REQ-OPTA-IN2).** Si un vuelo no cumple los criterios de candidato a OPTAIN-SA definidos en el requisito **REQ-2** será automáticamente descartado de la secuencia de vuelos candidatos, y no se mostrará en la ventana OPTA-IN.

#### 4.2. Ventana OPTA-IN

**REQ-4 (REQ-OPTA-IN10).** La secuencia de vuelos de llegada al IAF definida por **REQ-2** se presentará gráficamente en una ventana con una regla de distancia vertical con escala de 1 NM. En dicha ventana se representará, sin hacer distinción del flujo de procedencia, la distancia de cada aeronave de la secuencia de llegada a su punto TIE y al IAF. La posición de cada vuelo sobre la regla vertical será fiel, en principio, a la distancia del mismo respecto al punto TIE y al IAF, salvo que el excesivo número de vuelos dentro de un intervalo de la regla provoque limitaciones de espacio, en cuyo caso se representarán los vuelos apilados sin reflejar la distancia real.

**REQ-5 (REQ-OPTA-IN11, REQ-OPTA-IN22).** En la ventana secuenciadora OPTA-IN se mostrarán los vuelos hasta el momento en el que dichas aeronaves sobrepasen el punto TIE en una distancia de 2 NM. El rango de análisis tendrá como límite superior 18NM desde el punto

TIE, de forma que no se considerarán las aeronaves que se encuentren a una mayor distancia del punto TIE.

**REQ-6 (REQ-OPTA-IN11).** La ventana OPTA-IN presentará una línea horizontal discontinua correspondiente al punto TIE (indicada mediante el rotulo “TIE-Point” en el eje de ordenadas) y en la parte inferior de la ventana se mostrará el punto vertical del IAF, donde convergen los flujos de tráfico analizados, sobre la regla de distancia.

**REQ-7 (REQ-OPTA-IN14).** La ventana OPTA-IN mostrará, en el lado izquierdo de la regla de distancia, a la altura correspondiente a cada vuelo, los siguientes campos para los vuelos de la secuencia de llegada al IAF representados:

- Indicativo
- Velocidad (información radar)
- Punto TIE correspondiente
- Distancia a su punto TIE

**REQ-8 (REQ-OPTA-IN21).** El indicativo del vuelo en la ventana OPTA-IN deberá resaltarse (indicación a través de parpadeo) en el momento en el que dicho vuelo se encuentre a una distancia de 1 NM de su punto TIE correspondiente. Dicha indicación permanecerá activa hasta el momento en el que se encuentre a una distancia mayor de 1NM desde el sobrevuelo del punto TIE correspondiente (en este momento es cuando se espera que el controlador tenga que transmitir el ajuste de velocidad a la aeronave). Como puntualización a **REQ-5**, los vuelos representados en la ventana OPTA-IN que se encuentren a una distancia entre 1 NM y 2 NM tras sobrepasar el punto TIE, no mostrarán dicho resalte.

**REQ-8-B:** El requisito anterior (**REQ-8**) se aplicará sólo a los vuelos candidatos no excluidos, y siempre que estén emparejados. Es decir, no deberá resaltarse el indicativo de los vuelos excluidos (ya sea de forma manual o automática) ni de los que no tengan detrás una aeronave a menos de 16NM (aeronaves separadas por más de 16NM no se considerarán pareja, ya que con esa separación no hace falta modificar la velocidad para conseguir una separación de 5 ó 10 NM).

**REQ-9 (REQ-OPTA-IN15).** El procedimiento OPTAIN-SA asociará a la aeronave posterior de la secuencia de llegada al IAF una propuesta de velocidad para cumplir con las 2 separaciones configurables (5NM/10NM). De esta forma, la ventana OPTA-IN mostrará, en el lado derecho de la regla de distancia, a la altura correspondiente a cada vuelo, las velocidades propuestas por el procedimiento OPTAIN-SA para cada una de las dos separaciones configurables (5NM/10NM) inicialmente, y posteriormente mostrará las seleccionadas por el usuario, según se define en **REQ-15** y **REQ-16**.

**REQ-10 (REQ-OPTA-IN3).** La ventana OPTA-IN permitirá la exclusión o inclusión de forma manual de vuelos de la secuencia inicial dada por la función de selección automática definida en **REQ-2**, mediante el botón 'EXCL/INCL', situado en la parte inferior derecha de la misma. El procedimiento, tanto para la inclusión como para la exclusión de un vuelo, será el siguiente:

- Selección del vuelo con el botón izquierdo del ratón en la ventana OPTA-IN.
- Hacer “click” en el botón “EXCL/INCL”.

**REQ-11 (REQ-OPTA-IN4).** Los vuelos en los cuales las velocidades propuestas para ambas separaciones sea ‘N.A.’ se considerarán excluidos automáticamente.

**REQ-12 (REQ-OPTA-IN4).** Los vuelos excluidos seguirán apareciendo en la ventana OPTA-IN, con un color rojo si la función OJEO está activa.

**REQ-12-B (REQ-OPTA-IN4).** Las velocidades de los vuelos excluidos, ya sea manual o automáticamente, seguirán siendo actualizadas en cada refresco de la ventana, hasta el momento en el que la aeronave precedente llega a su punto TIE.

**REQ-13 (REQ-OPTA-IN5).** La ventana OPTA-IN contará con un botón “OJEO” que permitirá ocultar o visualizar (con las características expuestas en **REQ-10** y **REQ-12**) los vuelos excluidos.

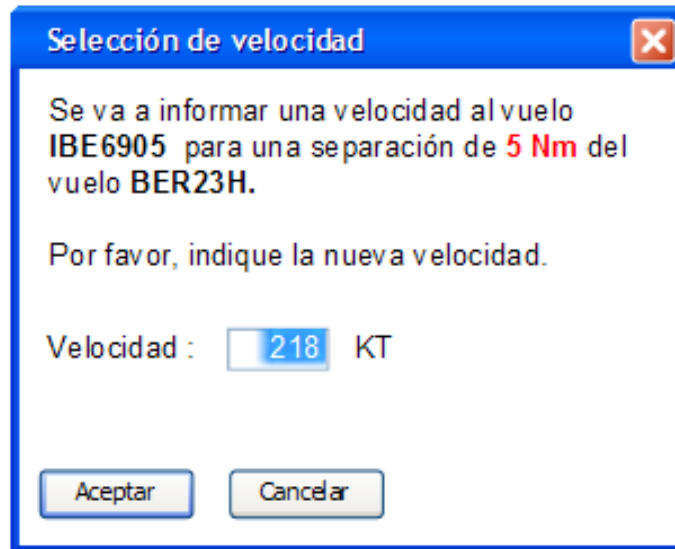
**REQ-14 (REQ-OPTA-IN6).** Las velocidades propuestas por el procedimiento OPTAIN-SA para garantizar la separación entre aeronaves consecutivas en la secuencia de llegada se actualizarán tras las acciones manuales de exclusión/inclusión de vuelos de la secuencia, siempre que dichas velocidades no hayan sido modificadas manualmente.

*NOTA:* Se considera que una velocidad es modificada manualmente cuando el controlador cambia el valor propuesto por la herramienta, por otro que considera más adecuado. En el caso de que el controlador acepte la velocidad propuesta por la herramienta, esta no se considerará una modificación.

**REQ-14-B.** El requisito anterior (**REQ-14**) se aplicará también a las aeronaves excluidas de forma automática.

**REQ-15 (REQ-OPTA-IN23).** La ventana OPTA-IN permitirá al controlador escoger, mediante selección con el ratón en la velocidad propuesta para el vuelo (click con el botón izquierdo sobre el dato), la alternativa de ajuste de velocidad que se va a indicar al vuelo. De esta forma, se resaltarán la velocidad seleccionada (para secuenciar a 5NM o 10NM), y a su vez, se atenuará la indicación no utilizada. Los vuelos candidatos a ajustar su velocidad a través del procedimiento OPTA-IN se representarán en amarillo en la ventana OPTA-IN.

**REQ-16 (REQ-OPTA-IN23, REQ-OPTA-IN26).** Al seleccionar una velocidad se mostrará un diálogo en el que se sugerirá, para la velocidad escogida, el último valor informado, mostrándose la primera vez el sugerido por el sistema, tanto para una separación de 5NM como de 10NM. Asimismo, se indicará en esta ventana de selección de velocidad el vuelo del que se está separando.



**Ilustración 5: Ventana de selección de velocidad**

**REQ-17 (REQ-OPTA-IN24).** La ventana OPTA-IN indicará en color verde aquellos pares de vuelos que estén ejecutando el procedimiento OPTA-IN. Tras la selección de la velocidad en una aeronave dada, se producirá el cambio automático a color verde para dicha aeronave y su anterior en la secuencia de llegada, diferenciándose así del resto de vuelos.

**REQ-18 (REQ-OPTA-IN25).** Una vez se ha seleccionado la indicación de velocidad que se le va a dar a un vuelo para proporcionar una secuencia de llegada a 5NM/10NM, en los vuelos posteriores de forma automática quedará resaltada la velocidad indicada para la misma separación (5NM/10NM) hasta que no se seleccione otra opción. Es decir, pese al resalte automático en los vuelos posteriores, será posible cambiar en éstos la velocidad indicada de acuerdo a la otra opción de distancia de separación.

**REQ-19 (REQ-OPTA-IN19, REQ-OPTA-IN20).** En la ventana OPTA-IN habrá dos botones para configurar la distancia de separación: '5NM' y '10NM'. Cada botón abrirá una ventana auxiliar de formato tabular con la secuencia de llegada de los vuelos candidatos OPTAIN-SA para cumplir con la separación, respectivamente, de 5 NM y 10 NM.

**REQ-20 (REQ-OPTA-IN27, REQ-OPTA-IN28).** Cada vuelo al que se ajuste la velocidad por medio de OPTAIN-SA (ya sea la indicación automática o la selección manual introducida por el usuario) y llegue al IAF incrementará un contador interno, que se almacenará en un dispositivo externo (disco o base de datos) y se mostrará posteriormente en el informe. Asimismo, se guardará la siguiente información para cada par de vuelos que participen en un procedimiento OPTAIN.

- Indicativo AC1
- Indicativo AC2
- Distancia de separación en el punto TIE entre AC1 y AC2, cuando AC1 llega al punto TIE
- Ajuste de velocidad activo, cuando AC1 llega al punto TIE
- Velocidad de AC1 en punto TIE
- Velocidad de AC2, cuando AC1 llega al punto TIE



- Distancia de separación AC1-AC2, cuando AC1 llegue al IAF
- Instante de paso de AC1 por punto TIE
- Velocidad de AC1 en el IAF
- Velocidad de AC2 en el IAF
- Punto TIE de AC1
- Punto TIE de AC2
- Día (fecha y hora) de aplicación del procedimiento OPTAIN-SA

La notación AC1 se refiere a la aeronave anterior de la secuencia, mientras que AC2 va referido a la aeronave que aplica la indicación de velocidad.

**REQ-20-B.** También se almacenará un contador interno por cada pareja de vuelos candidata a hacer un procedimiento OPTAIN, se haya realizado el procedimiento o no.

*NOTA:* Se considera una pareja de vuelos candidata a todo par de vuelos separados por una distancia comprendida entre las 3NM y las 16NM.

### **4.3. Tabular secuenciador**

**REQ-21 (REQ-OPTA-IN13).** El criterio de ordenación de los vuelos en el tabular del secuenciador vendrá dado por su distancia al punto TIE correspondiente, de forma que serán ordenados en el tabular por orden creciente de distancia.

**REQ-22 (REQ-OPTA-IN14).** El tabulador presentará para cada vuelo de la secuencia de llegada los siguientes campos:

- Indicativo
- Velocidad (información radar)
- Punto TIE correspondiente
- Distancia a su punto TIE

**REQ-23 (REQ-OPTA-IN15).** El tabulador presentará las velocidades indicadas (IAS, Indicated Air Speed) propuestas por el procedimiento OPTAIN-SA para conseguir una separación de 5NM/10NM, de acuerdo con la distancia configurada en el tabular, en el IAF con la aeronave anterior en la secuencia.

**REQ-24 (REQ-OPTA-IN21).** El indicativo del vuelo en el tabular secuenciador deberá resaltarse (indicación a través de parpadeo) en el momento en el que dicho vuelo se encuentre a una distancia de 1 NM de su punto TIE correspondiente. Dicha indicación permanecerá activa hasta el momento en el que se encuentre a una distancia mayor de 1NM a partir del sobrevuelo del punto TIE correspondiente. Como puntualización a **REQ-5**, los vuelos representados en la ventana OPTA-IN que se encuentren a una distancia entre 1 NM y 2 NM tras sobrepasar el punto TIE, no mostrarán dicho resalte.

**Vuelos - 5 NM**

**5 NM**

ID	SPEED	FIX	D	IAS
AEA7049	325	TIELO	0.3	N.A.
IBE5114	318	LORES	0.3	N.A.
VLG1902	330	TIETOL	2.7	280KT
BER23H	329	TIELO	5.6	250KT
IBE6905	330	LORES	10.1	280KT
AAW5001	316	TIELO	13.1	280KT

Cerrar

Ilustración 6: Tabular secuenciador 5NM

**Vuelos - 10 NM**

**10 NM**

ID	SPEED	FIX	D	IAS
AEA7049	325	TIELO	0.3	N.A.
IBE5114	318	LORES	0.3	N.A.
VLG1902	330	TIETOL	2.7	N.A.
BER.23H	329	TIELO	5.6	N.A.
IBE6905	330	LORES	10.1	220KT
AAW5001	316	TIELO	13.1	240KT

Cerrar

Ilustración 7: Tabular secuenciador 10NM

#### 4.4. *Velocidad propuesta por el procedimiento OPTAIN-SA*

Dada una aeronave de la secuencia de llegada al IAF, el procedimiento OPTAIN-SA podrá proponer una velocidad de ajuste para la aeronave posterior en la secuencia, que permitirá cumplir con el objetivo de separación configurada entre aeronaves, de acuerdo a los siguientes requisitos. Esto se aplicará para todas las parejas de aeronaves candidatas:

**REQ-25 (REQ-OPTA-IN16).** Las indicaciones de ajuste de velocidad para la aeronave posterior se calcularán en función de unas tablas predefinidas que relacionarán la distancia de esta segunda aeronave a su punto TIE, o lo que es lo mismo, la distancia entre ambas aeronaves en el momento en que la aeronave precedente atravesase su propio punto TIE, con

las correspondientes velocidades propuestas para cumplir con el criterio de separación, que dependerá de la configuración de 5NM/10NM escogida. Las aeronaves excluidas no se tendrán en cuenta para estos cálculos. Las tablas se adjuntan en el Anexo de este documento (Ver 8.1). Estas velocidades se actualizarán en cada refresco de la ventana, hasta el momento en el que la aeronave precedente llega a su punto TIE.

**REQ-26 (REQ-OPTA-IN17).** Si la distancia de la aeronave posterior a su punto TIE es mayor a la establecida en las tablas de ajuste de velocidad del Anexo (aeronaves muy separadas) o la aeronave anterior es la primera en la secuencia de llegada al IAF, se proporcionará una indicación de velocidad de ‘280Kts’ por defecto para 5NM/10NM.

**REQ-27 (REQ-OPTA-IN18).** Si la distancia entre 2 aeronaves no permite secuenciarlas apropiadamente a 5NM/10NM de separación en el IAF mediante un ajuste de velocidad (ej. la distancia al punto TIE de la aeronave posterior es lo suficientemente reducida), se mostrará el campo de velocidad propuesta como ‘N.A.’

#### **4.5. Implementación del procedimiento OPTAIN-SA a partir de SINA**

La integración del procedimiento OPTAIN-SA dentro de un nuevo servicio de la aplicación SINA cumplirá con los requisitos funcionales detallados en las secciones anteriores. Adicionalmente, se definen los siguientes requisitos de implementación:

**REQ-28.** Se definirá un nuevo servicio en SINA que, a partir de la información almacenada en la Base de Datos operacional y con la misma conexión TCP/IP actualmente mantenida con los servidores SACTA, implemente el procedimiento OPTAIN-SA sin interferir con los servicios actuales de la aplicación SINA.

**REQ-29.** El resto de servicios de SINA quedarán inhabilitados para OPTAIN-SA, a excepción del login y administración.

**REQ-30 (REQ-OPTA-IN29).** El acceso al nuevo servicio en SINA que implemente el procedimiento OPTAIN-SA sólo será accesible a los usuarios que utilicen una combinación de usuario autorizado y contraseña válidos.

**REQ-31.** El nuevo servicio de la aplicación SINA que implemente el procedimiento OPTAIN-SA se servirá de los datos operativos de pistas en tiempo real y planes de vuelo, recibidos a través de un canal autorizado externo de comunicaciones de datos.

**REQ-32.** La actualización de datos operativos de pistas y planes de vuelo estará ligada a la frecuencia de refresco actual de la aplicación SINA, es decir, con un periodo máximo de 10 segundos. Asimismo, las indicaciones dadas por **REQ-25** a los vuelos candidatos a ajustar su velocidad a través del procedimiento OPTA-IN se actualizarán de acuerdo al refresco de los datos operativos de pistas en tiempo real y planes de vuelo recibidos por la herramienta.

**REQ-33.** El HMI de la ventana del servicio SINA que implementará la funcionalidad OPTAIN-SA constará de los siguientes elementos:

- Ventana OPTA-IN. Cumplirá con los requisitos **REQ-4, REQ-5, REQ-6, REQ-7, REQ-8, REQ-9, REQ-11, REQ-12, REQ-14, REQ-15, REQ-16** y **REQ-17** especificados en el presente documento. Tendrá la apariencia mostrada en la siguiente figura.

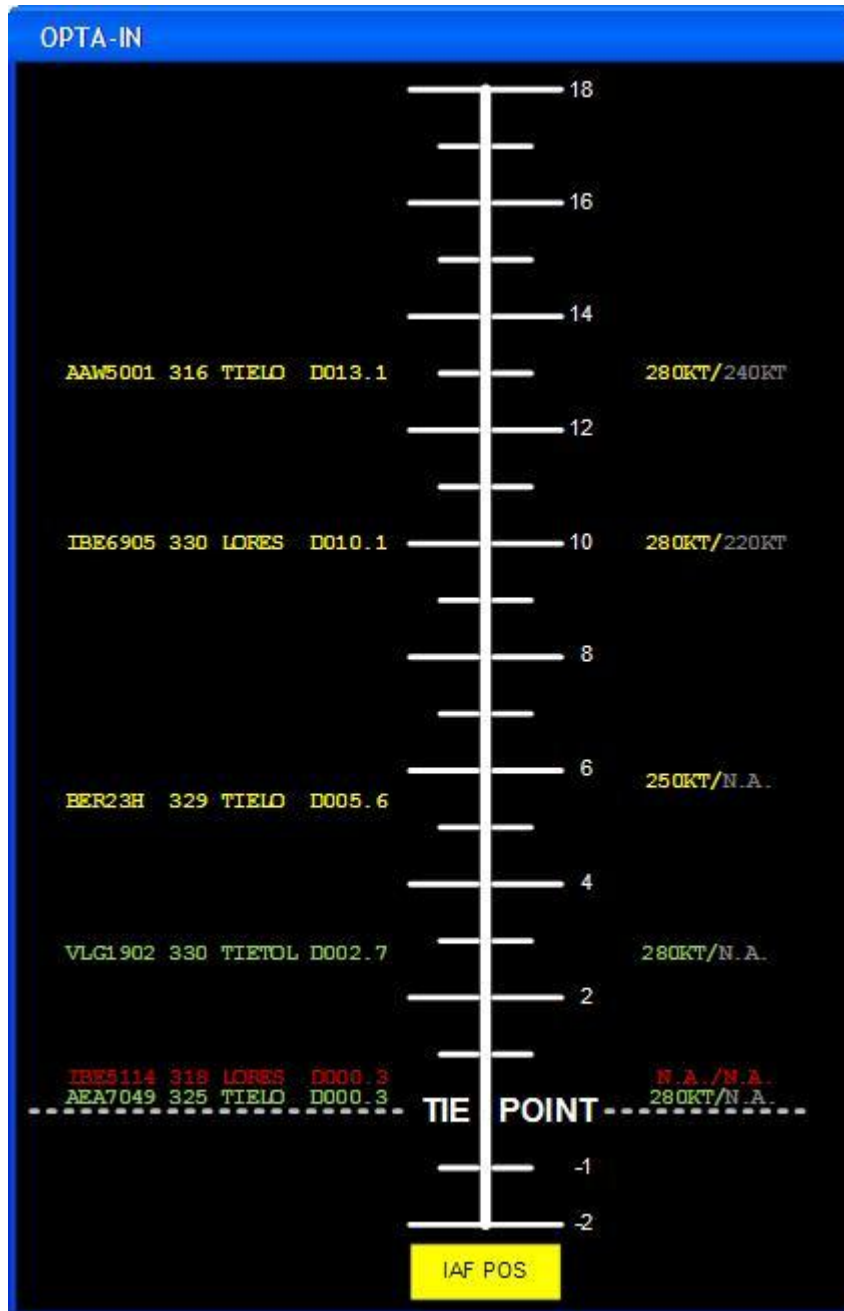


Ilustración 8: Ventana OPTAIN

- Tabular de la secuencia de llegada de vuelos al IAF. Cumplirá con los requisitos **REQ-21**, **REQ-22**, **REQ-23** y **REQ-24** especificados en el presente documento. Tendrá la apariencia mostrada en la siguiente figura.

ID	SPEED	FIX	D	IAS
AEA7049	325	TIELO	0.3	N.A.
IBE5114	318	LORES	0.3	N.A.
VLG1902	330	TIETOL	2.7	280KT
BER23H	329	TIELO	5.6	250KT
IBE6905	330	LORES	10.1	280KT
AAW5001	316	TIELO	13.1	280KT

**Ilustración 9: Tabular de secuencia de llegada**

- Ventana “Informe”. La información guardada de los vuelos que apliquen el procedimiento OPTAIN-SA que se especifica en **REQ-20** y **REQ-20-B** podrá ser representada en la ventana “Informe”, a la que se accederá desde el botón “Informe” del panel de control. En dicha ventana se podrán definir búsquedas de vuelos atendiendo a los siguientes parámetros de búsqueda:
  - Fecha de aplicación del procedimiento OPTAIN-SA. Permitirá la definición de fecha de inicio y fecha de fin del instante de aplicación del procedimiento.
  - Distancia de separación en el IAF. Se definirá introduciendo una distancia máxima de separación, expresada en NM.
  - Puntos TIE de cruce. A través del correspondiente checkbox se podrán incluir o excluir en la búsqueda los vuelos que pasen por los puntos TOLSO, LORES y KENAS.

Esta información se guardará siempre que alguna sesión esté abierta, independientemente del usuario. La información de todos los vuelos recibidos en el servidor OPTAIN durante el periodo de tiempo en el que ningún usuario está conectado no debe ser almacenada.

Con el fin de poder realizar un estudio posterior, esta información estará disponible en todo momento, por lo que no debe ser borrada de forma automática en ningún caso.

Por defecto, cuando el usuario no introduzca valores de configuración para cualquiera de los tres anteriores parámetros de búsqueda, se considerarán todos los valores posibles guardados para tales parámetros.

El botón “Buscar” iniciará la búsqueda de vuelos para los parámetros configurados, mientras que el botón “Limpiar” borrará de la ventana “Informe” el contenido de la última lista de resultados mostrada.

Será posible guardar en formato Excel la información representada en la ventana “Informe” a través del botón “Exportar”. El fichero guardado constará de dos partes: “Cabecera” y “Datos”.

- *Cabecera*: contendrá los parámetros de configuración de búsqueda introducidos por el usuario.
- *Datos*: contendrá el valor de los campos especificados en **REQ-20** para los vuelos resultantes de la búsqueda configurada.

También se mostrará en el informe la comparación del número total de parejas candidatas a aplicar el procedimiento OPTAIN (**REQ-20-B**) y las que realmente aplicaron el procedimiento. También puede ser aplicado un filtro por fecha, en ese caso el número de parejas se restringirá al periodo comprendido entre las fechas indicadas.

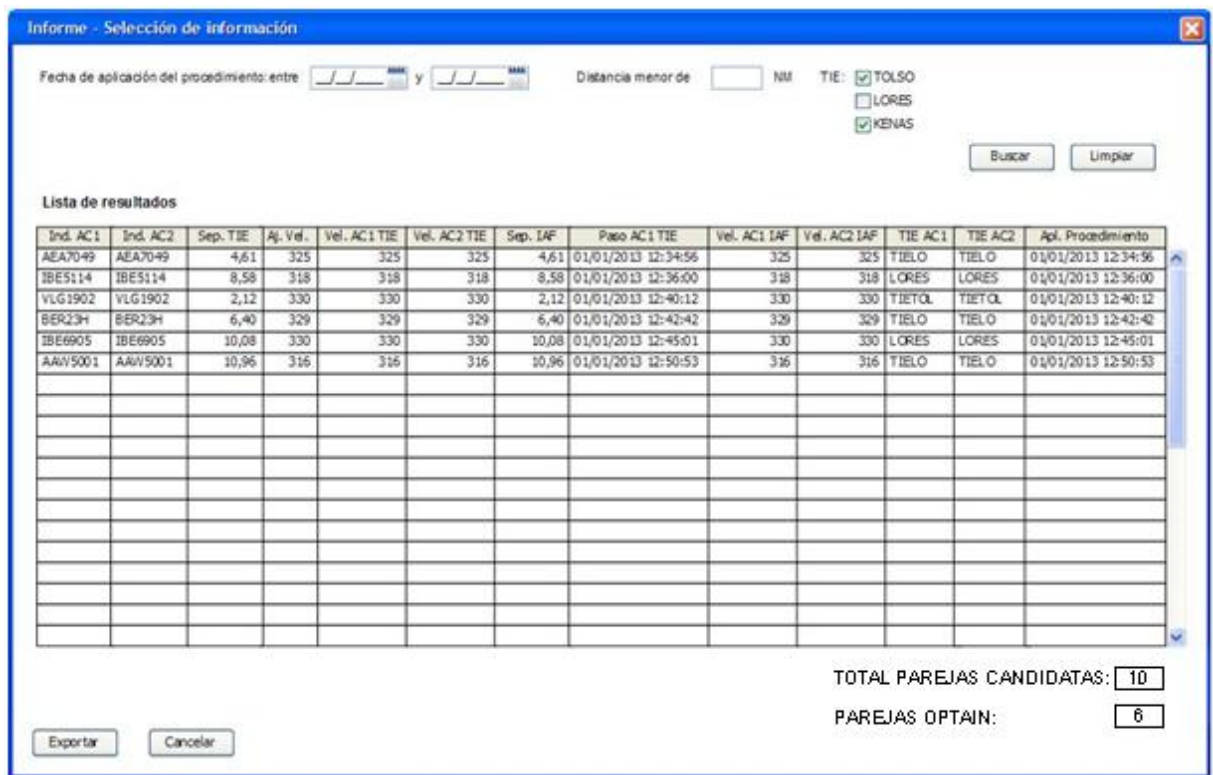


Ilustración 10: Ventana Informe

- Panel de control. Contará con los siguientes elementos:
  - Botones de aplicación en la Ventana OPTA-IN
    - ‘5NM’, ‘10NM’. Permitirán la apertura del respectivo Tabular Secuenciador, de acuerdo al requisito **REQ-18**.
    - ‘OJEO’. Permitirá ocultar o visualizar los vuelos excluidos de forma manual de la secuencia inicial, como se especifica en **REQ-13**.
    - ‘EXCL/INCL’. Permitirá la exclusión o inclusión de forma manual de vuelos de la secuencia inicial dada por la función de selección, de acuerdo al requisito **REQ-10**.

- ‘Informe’. Permitirá acceder a la ventana ‘Informe’ para efectuar búsquedas de vuelos que apliquen el procedimiento OPTAIN-SA.
- Hora del sistema



**Ilustración 11: Panel de control de la ventana principal del servicio SINA de implementación de OPTAIN**



**Ilustración 12: Ventana principal del servicio SINA de implementación de OPTAIN**

**REQ-34.** No será posible modificar o configurar el sistema operacional SACTA a través de la herramienta de implementación del procedimiento OPTAIN-SA.

**REQ-35.** No será posible conectar la herramienta de implementación del procedimiento OPTAIN-SA al sistema operacional SACTA.

**REQ-36.** El nuevo servicio se valdrá de la configuración off-line de los siguientes parámetros:

- Datos AIP (Aeronautical Information Publications) necesarios, tales como OPTA legs de los flujos de tráfico por los fijos a tratar, y la lista de fijos en sí misma. Asimismo, se identificará en la configuración el IAF sobre el que se aplica el procedimiento.
- Mapas locales ad-hoc para el escenario contemplado (ej. arco de corte con los puntos TIE de los flujos de tráfico considerados).
- Tablas de velocidad propuesta por el procedimiento OPTAIN-SA para la separación objetivo en el IAF.
- Parámetros de configuración del área geográfica de selección de vuelos candidatos:
  - Distancia del arco circular al IAF (radio menor de la corona circular). Por defecto su valor será de 31NM.
  - Radio mayor de la corona circular. Por defecto su valor será de 49NM.
  - El ángulo inicial. Por defecto su valor será de 280°.
  - El ángulo final. Por defecto su valor será de 10°.
  - Coordenadas geodésicas del IAF sobre el que se aplica el procedimiento.





## 4.7. Corrección de problemas

Además de los requisitos anteriores se incluyen los requisitos derivados de los problemas encontrados por el usuario numerados de la siguiente forma: REQ-X donde X es el número que identifica el problema continuando con la numeración de los requisitos anteriores y su resolución se identifica con RESOL-X donde X representa el número del problema que se resuelve.

**REQ-37.** Parpadeo de cabeceras de tabla en el tabular de 10NM.

**RESOL-37.** Se comprobó que estaban utilizando una versión diferente del JRE por lo que hubo que actualizar la versión que estaban utilizando hasta el momento.

**REQ-38.** Vuelo con velocidad propuesta y posteriormente excluido es reseteado a 280 en velocidad y no vuelve a proponer velocidad al incluirlo.

**RESOL-38.** Cuando un vuelo se excluía se consideraba que había sido modificado manualmente por lo que su velocidad no se debería de actualizar automáticamente, pero esto no debería de ocurrir porque su velocidad no ha sido modificada.

```
if ((velocidadActual != null) && (!velocidadActual.equals(""))) {  
  
    if (separacion.equals("5 NM")) {  
        if (pistaAccion.velocidadPropuesta5NM != Integer.parseInt(velocidadActual)) {  
            pistaAccion.modificadaManual = true;  
        }  
        pistaAccion.velocidadPropuesta5NM = Integer.parseInt(velocidadActual);  
    }  
    else {  
        if (pistaAccion.velocidadPropuesta10NM != Integer.parseInt(velocidadActual)) {  
            pistaAccion.modificadaManual = true;  
        }  
        pistaAccion.velocidadPropuesta10NM = Integer.parseInt(velocidadActual);  
    }  
}
```

**REQ-39.** Scroll del report no permite visualizar todas las parejas de vuelos que deberían mostrarse ya que al desplazarlo a la posición máxima de visualización faltan vuelos por mostrar.

**RESOL-39.** El tamaño del panel crecía menos con cada inserción en el listado que lo que ocupaba la inserción, por lo que el listado no permitía ver todo su contenido. Se estableció un tamaño fijo para cada fila y con cada inserción en la lista el panel aumentaba el tamaño de una fila.

**REQ-40.** Un vuelo excluido automáticamente se muestra en color amarillo, en lugar de color rojo.

**RESOL-40.** No se había contemplado esa situación en cuanto al cambio de color cuando se producía una exclusión automáticamente, para lo que fue necesario únicamente llevar a cabo ese cambio a color rojo del vuelo afectado.

**REQ-41.** Hay una zona en el fondo negro de la aplicación (franja horizontal que hay justo por debajo de la parte superior, de 1cm de grosor, más o menos) en la que se puede desplegar un menú con el botón derecho del ratón.

**RESOL-41.** Esa franja horizontal quedaba fuera de lo que abarcaba la aplicación y mostraba un menú de contexto. Se resolvió añadiendo en la siguiente línea el código `oncontextmenu="return false;"`

```
<body bgcolor="black" oncontextmenu="return false;" onLoad="displayMessage()">
```

**REQ-42.** Tres vuelos seguidos, el primero 280/280 los otros dos NA NA, si al último le pongo una velocidad lo pone en verde y al segundo también verde con NA NA.

**RESOL-42.** La solución fue no permitir modificar la velocidad ni mostrar la ventana donde cambiarla para aquellos vuelos excluidos o en primera posición en la secuencia.

**REQ-43.** Si para dos vuelos consecutivos (para el segundo pone NA NA y está excluido automáticamente) excluimos el primero, para este segundo no actualiza velocidades y sigue mostrando NA NA.

**RESOL-43.** Fue necesario añadir una condición que tuviera en cuenta si el vuelo fue excluido automáticamente y fue actualizado con una velocidad mayor que 0, y en ese caso, el vuelo pasa a estar incluido en la secuencia.

```
//Si la pista esta excluida automaticamente y ha sido actualizada con una velocidad
//mayor que 0 para una distancia de 5NM entonces, se incluye
if (pistaOptainExt.excluidaAutomaticamente && pistaOptainExt.velocidadPropuesta5NM > 0) {
    pistaOptainExt.excluidaAutomaticamente = false;
    pistaOptainExt.excluida = false;
}
```

**REQ-44.** Tres vuelos AC1, AC2, AC3 (AC1 es el primero). Si acepto la velocidad propuesta para el AC3, pone en verde vuelo AC3 y AC2 como aplicando OPTAIN. Si excluyo el vuelo AC2, deja en verde el vuelo AC3, el AC2 en rojo y el AC1 en amarillo, pero el AC3 debería pasar a amarillo ya que se ha cancelado el procedimiento OPTAIN al excluir el vuelo AC2.

**RESOL-44.** Se añadió el siguiente código para tener en cuenta si al excluir un vuelo eso afecta al vuelo anterior o al siguiente en la secuencia, y actuar en consecuencia.

```
public void excluyePista(OptainPistaSecuenciador pista) {
    // Excluimos una pista que esta afectando a otra posterior
    // hay eliminar los calculos en la pista posterior
    if (pista.indicativoPistaPost != null) {
        OptainPistaSecuenciador pistaPost = getOptainPistaSecuenciador(pista.indicativoPistaPost);
        // Obtenemos la pista posterior en el TIE
        // La pista posterior deja de ejecutar optain a no ser que sea a su vez la anterior de otra pareja
        if (pistaPost != null && pistaPost.ejecutaOptain && !pistaPost.esParejaOptain) {
            pistaPost.setEjecutandoOptain(false);
        }
        else if (pistaPost != null && pistaPost.ejecutaOptain) {
            pistaPost.ejecutaOptain = false;
        }
    }
    // Recorrer toda la lista de pistas y ver si el indicativo de la pista que estamos
    // excluyendo (pista.dispiBean.INDICATIVO)
    // aparece como indicativoPistaPost en alguna de esas pistas
    if (pista.dispiBean.INDICATIVO != null) {
        OptainPistaSecuenciador pistaAnt = getOptainPistaAnteriorSecuenciador(pista.dispiBean.INDICATIVO);
        // Obtenemos la pista anterior en el TIE
        // La pista anterior deja de ejecutar optain a no ser que sea a su vez anterior de otra pareja
        if (pistaAnt != null && pista.ejecutaOptain && !pistaAnt.ejecutaOptain) { //XXX AQUI
            pistaAnt.setEjecutandoOptain(false);
        }
        else if (pistaAnt != null && pista.ejecutaOptain) {
            pistaAnt.esParejaOptain = false;
        }
    }
    pista.setEjecutandoOptain(false);
}
```

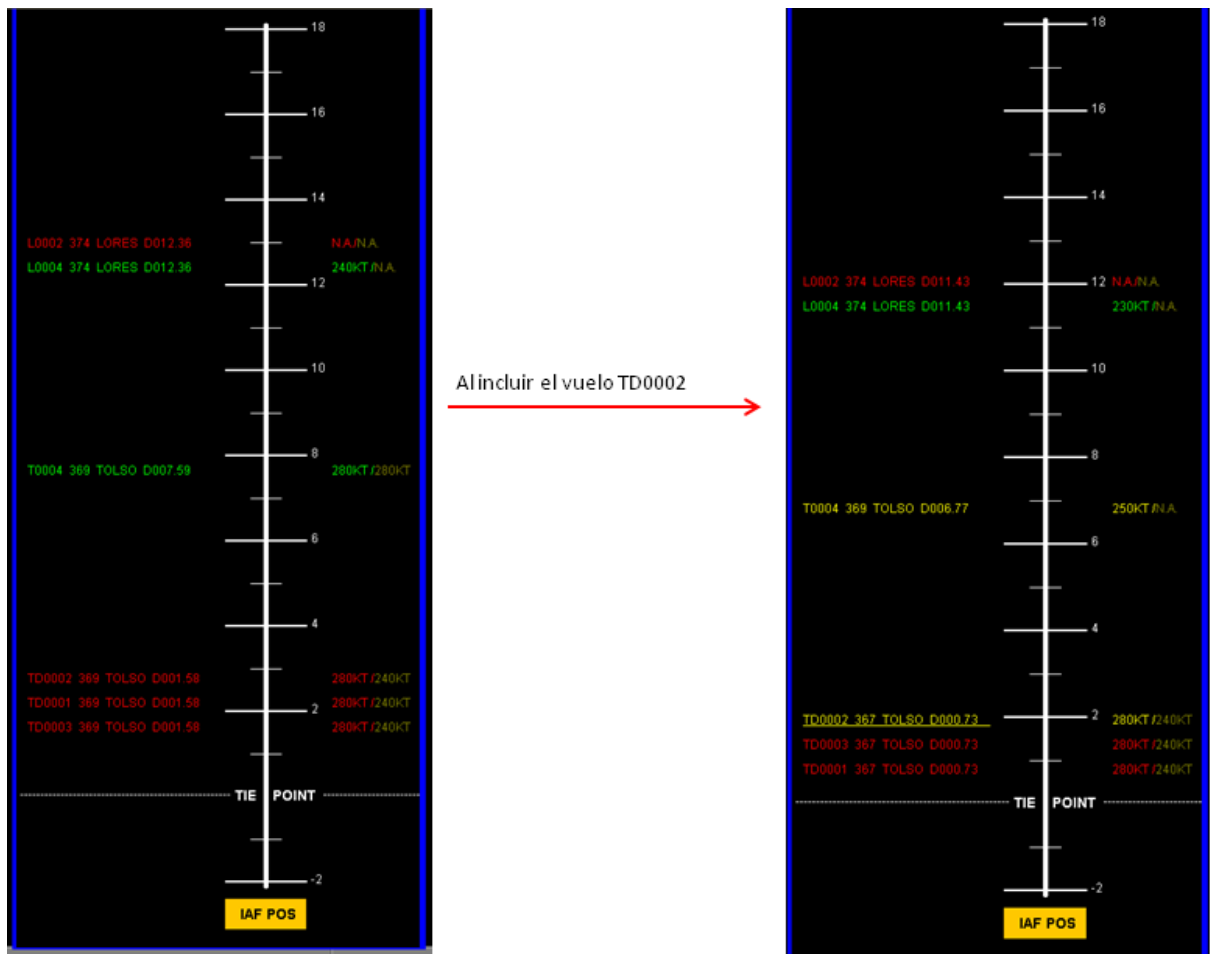
**REQ-45.** Si a partir de la situación anterior, se quiere aceptar la velocidad propuesta para AC3 o bien introducir manualmente una distinta, tampoco pone en verde el AC1, solo pone en verde el AC3, cuando deberían estar en verde ambos ya que están aplicando OPTAIN.

**RESOL-45.** Se consideraba únicamente como pista anterior la pista que se encontraba delante independientemente de su estado, para resolver esto fue necesario identificar la pista anterior incluida para que, en el caso propuesto, ejecutara con ella el procedimiento.

```
if (pistaAccion == null) {
    pistaAnterior = pista;
    if (!pista.excluida) {
        pistaAnteriorIncluida = pista;
    }
}
```

**REQ-46.** Hay ocasiones en las que para una pareja de vuelos que están aplicando procedimiento OPTAIN (ambos en verde), se queda uno solo de los dos vuelos en color verde (el segundo vuelo) al hacer una acción posterior sobre otro vuelo no involucrado, cosa incorrecta y debiendo pasar en todo caso ambos vuelos a amarillo. Ese vuelo en verde se comporta a lo largo del tiempo inapropiadamente, llegando a desaparecer unos instantes, y luego volviendo a aparecer, incluso cambiando a color verde cuando esta sobre el TIE point.

**RESOL-46.** Tras varios intentos se consiguió reproducir el problema que era el siguiente, cuando se incluía un vuelo excluido manualmente y alteraba la velocidad de uno de los vuelos que estaban aplicando el procedimiento, ese vuelo con el cambio de velocidad dejaba de aplicar el procedimiento.



Para resolver esta situación fue necesario incluir el siguiente código:

```

if((pistaAnterior.indicativoPistaPost != null) && !pistaAnterior.indicativoPistaPost.equals
(pistaOptainExt.dispiBean.INDICATIVO)){
    // Cambio de pistaPosterior. Eliminar la ejecucion optain
    OptainPistaSecuenciador pistaPost =
    getOptainPistaSecuenciador(pistaAnterior.indicativoPistaPost);
    if (pistaPost != null && pistaPost.ejecutaOptain && pistaAnterior.esParejaOptain) {
        if (!pistaPost.esParejaOptain) {
            pistaPost.setEjecutandoOptain(false);
        }
        if (!pistaAnterior.ejecutaOptain) {
            pistaAnterior.setEjecutandoOptain(false);
        }
    }
}
else {
    pistaAnterior.indicativoPistaPost = pistaOptainExt.dispiBean.INDICATIVO;
}

```

Y comprobar si la siguiente pista es la primera o la segunda de la pareja que aplica el procedimiento.

**REQ-47.** Cuando tres seguidos ejecutan OPTAIN al excluir el ultimo pasa a amarillo también el segundo.

**RESOL-47.** Cómo se puede ver en el código adjunto a la **RESOL-44** antes de pasar un vuelo a amarillo se comprueba que éste no esté ejecutando el procedimiento con otro vuelo.

**REQ-48.** Problemas al mezclar OPTAIN con excluir automáticamente.

**RESOL-48.** Estos problemas quedaron resueltos con la **RESOL-46**.

**REQ-49.** Ventana informe: columnas AC1 y AC2 al revés.

**RESOL-49.** El código estaba bien pero no en el orden correcto, sólo fue necesario modificar el orden.

**REQ-50.** Al abrir la ventana de Informe, se muestra un símbolo de “warning” al lado de la esquina superior derecha (por fuera de la ventana). También aparece este símbolo “warning” al abrir las ventanas de 5NM y 10 NM, cosa que con la versión anterior no pasaba.

**RESOL-50.** Siempre se veía el símbolo al lado de la ventana para corregir esta incidencia fue necesario modificar *java.policy* añadiendo la siguiente instrucción.

```
grant {  
    permission java.io.FilePermission "<<ALL FILES>>", "read";  
    permission java.io.FilePermission "<<ALL FILES>>", "write";  
    permission java.awt.AWTPermission "showWindowWithoutWarningBanner";  
};
```

**REQ-51.** Los vuelos que no están haciendo un procedimiento OPTA-IN (están en amarillo o rojo), no deberían parpadear al llegar al TIE Point.

**RESOL-51.** Todos los vuelos sin excepción parpadeaban al llegar al TIE Point cuando esto no debería ocurrir por lo que fue necesario añadir la restricción de que esto solo se produjera cuando ese vuelo estaba aplicando un procedimiento OPTA-IN.

**REQ-52.** Si al primer vuelo de la secuencia le aceptamos la velocidad propuesta de 280, pone en verde este vuelo como aplicando OPTAIN sin tener una pareja con la que aplicar el procedimiento.

**RESOL-52.** No se contemplaba si el vuelo que iba a aplicar el procedimiento OPTAIN tenía un vuelo anterior incluido con el que aplicar el procedimiento, por lo que hubo que tenerlo en cuenta y antes de mostrar la ventana para modificar la velocidad y pasar a ejecutar OPTAIN se comprueba si realmente puede llevar a cabo el procedimiento con algún vuelo de la secuencia.

**REQ-53.** No se actualizan las velocidades de los vuelos para los que se ha aceptado la velocidad propuesta por la aplicación.

**RESOL-53.** Se consideraba que la velocidad del vuelo había sido modificada y en ese caso la velocidad no se tiene que actualizar, para solucionarlo se añadió una nueva restricción que contemplara si la velocidad actual de la aeronave es diferente a la propuesta y en ese caso sí que fue modificada o por el contrario la velocidad actual es la misma que la propuesta y en ese caso la velocidad no ha sido modificada y deberá actualizarse cuando sea oportuno.

```
if (pistaAccion.velocidadPropuesta5NM != Integer.parseInt(velocidadActual))
{
    pistaAccion.modificadaManual = true;
}
```

**REQ-54.** Cuando el segundo vuelo de una de las parejas que está aplicando procedimiento OPTAIN está entre -1 y 1NM del TIE point parpadea. Este segundo vuelo ya no debería parpadear, solo el paso del primer vuelo de la pareja debería realizar el parpadeo.

**RESOL-54.** Fue necesario añadir una nueva restricción que comprobara si era el primero de la pareja que estaba aplicando OPTAIN y en ese caso parpadear o si por el contrario era el segundo de la pareja y en ese caso no debería parpadear.

```
OptainPistaSecuenciador pista =
tab.applet.secl.getOptainPistaSecuenciador(indicativo);
    if (Math.abs(dDistancia) < 1 && pista != null && pista.esParejaOptain) {
        if (segundos % 2 == 0) {
            g.setColor(Color.black);
        }
        else {
            g.setColor(Color.white);
        }
    }
    else {
        g.setColor(Color.black);
    }
```

La variable esParejaOptain devuelve true cuando se trata del primer vuelo de la pareja que aplica el procedimiento.

## 5. Diseño del sistema

### 5.1. Escenarios

A continuación se muestra un esquema de los escenarios del nuevo servicio de SINA que implementará el procedimiento OPTAIN-SA. Puesto que los requisitos están muy definidos se identifican los distintos escenarios y se explica cada uno de ellos en detalle.

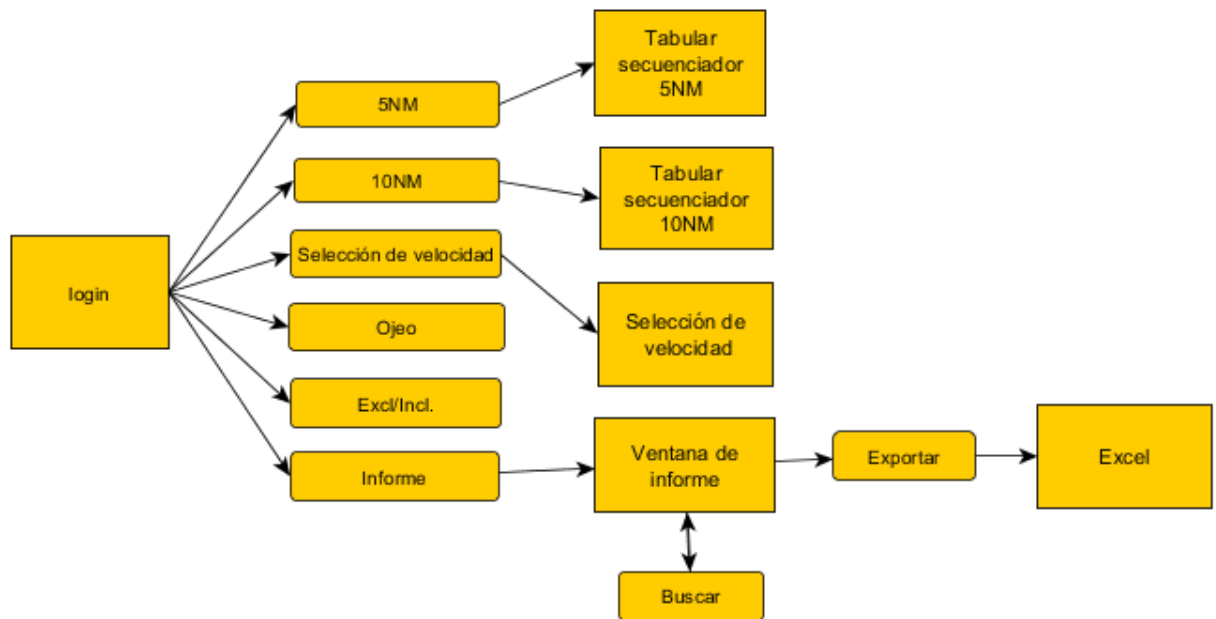


Ilustración 13: Escenarios



### 5.1.1. Descripción de los escenarios

- **“Tabular secuenciador 5NM”:** Un usuario desea abrir la ventana que muestra las parejas de vuelos candidatas para lograr una separación en su IAF de 5NM, para ello el usuario debe hacer click con el botón izquierdo del ratón sobre el botón “5NM” que se puede encontrar en la barra de botones inferior.
- **“Tabular secuenciador 10NM”:** Un usuario desea abrir la ventana que muestra las parejas de vuelos candidatas para lograr una separación en su IAF de 10NM, para ello el usuario debe hacer click con el botón izquierdo del ratón sobre el botón “10NM” que se puede encontrar en la barra de botones inferior.
- **“Selección de velocidad”:** Un usuario desea modificar la velocidad de vuelo de una aeronave. Para ello:
  1. El usuario debe pulsar sobre la velocidad del vuelo que desea modificar haciendo click sobre la velocidad con el botón izquierdo del ratón
  2. El sistema muestra automáticamente una ventana con un campo que muestra la velocidad actual de la aeronave
  3. El usuario debe escribir la nueva velocidad y pulsar el botón “Aceptar” con el botón izquierdo del ratón si desea guardar la nueva velocidad o por el contrario pulsar el botón “Cancelar” con el botón izquierdo del ratón para cerrar sin guardar.
  4. El sistema muestra los tabulares de los dos vuelos implicados en el procedimiento en color verde.
- **“Ojeo”:** Un usuario desea ocultar o visualizar los vuelos excluidos (automática o manualmente) de la secuencia, para ello, el usuario debe hacer click con el botón izquierdo del ratón sobre el botón “Ojeo” que se puede encontrar en la barra de botones inferior.

Si el botón “Ojeo” está activado, eso es, cuando las letras están en color rojo, el sistema muestra los vuelos excluidos, es decir, los tabulares de color rojo, en la secuencia de la ventana OPTA-IN. Por el contrario, cuando el botón no está activado, esos vuelos no se muestran.

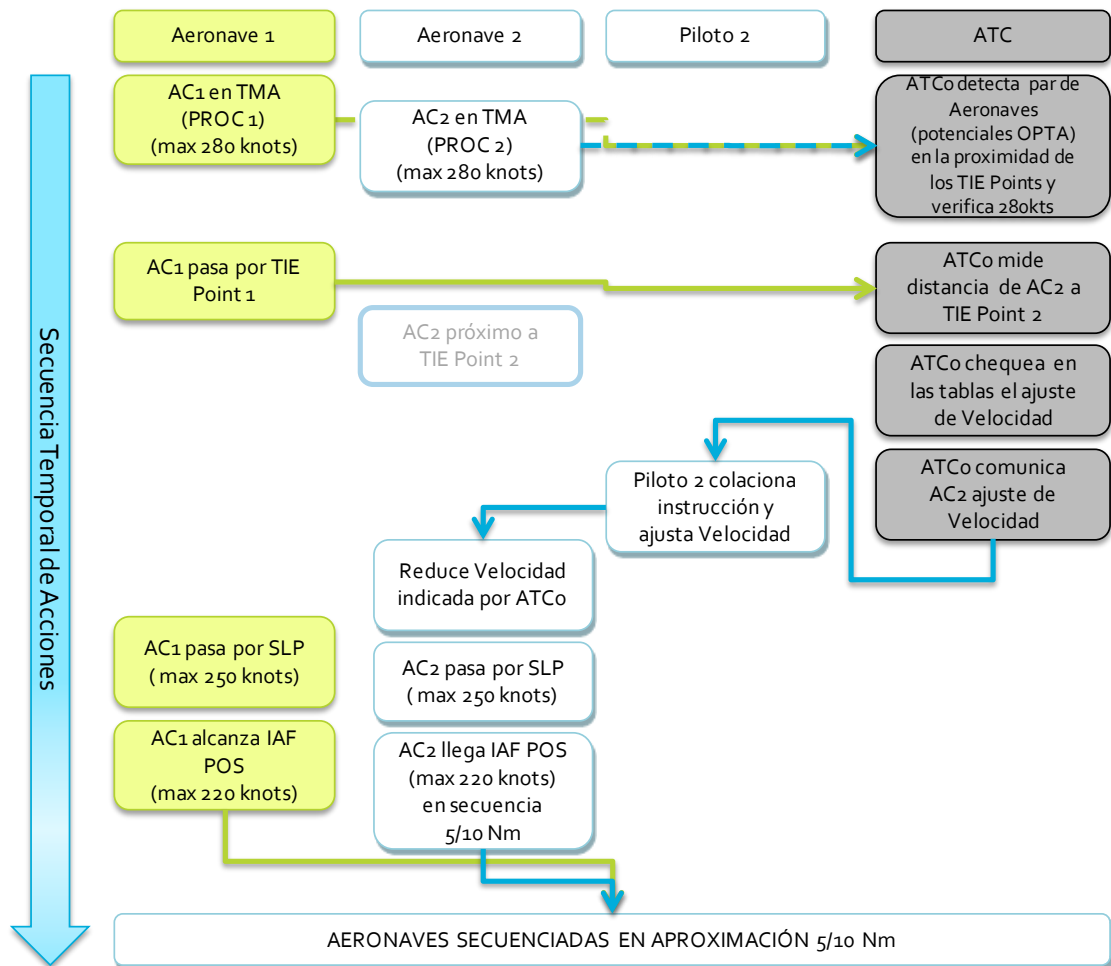
- **“Excl/Incl.”:** Un usuario desea excluir o incluir un vuelo manualmente. Para ello:
  1. El usuario debe seleccionar el vuelo que quiere excluir o incluir, haciendo click con el botón izquierdo del ratón sobre el vuelo. En ese momento, el vuelo seleccionado aparecerá subrayado.
  2. Una vez seleccionado debe pulsar el botón “Excl/Incl.” que se encuentra en la barra de botones inferior

Si el vuelo estaba excluido el vuelo se incluye y el sistema lo muestra en color amarillo y si por el contrario el vuelo estaba incluido el vuelo pasa a estar excluido y el sistema lo muestra en color rojo.

- **“Informe”**: Un usuario desea ver los procedimientos OPTA-IN realizados hasta ese momento, para ello:
  1. El usuario debe pulsar sobre el botón “Informe” de la barra de botones inferior con el botón izquierdo del ratón.
  2. El usuario podrá filtrar la búsqueda por fecha, por los puntos de entrada al TMA (Terminal Manoeuvre Area) y por distancia una vez completados los filtros que se deseen el usuario pulsará el botón “Buscar” para que se muestren todas aquellas parejas que cumplan con los requisitos de filtrado.
  3. Una vez hecha la búsqueda de vuelos, la herramienta permite generar un fichero Excel con toda la información buscada, para ello es necesario pulsar el botón “Exportar”

## 5.2. Secuencia de acciones

Como punto de partida se tienen los puntos de entrada al TMA LORES, TOLSO y KENAS. El Control de Tráfico Aéreo realiza dentro de su actividad de vigilancia continua la detección de aeronaves que se aproximen a su punto TIE, en relación a la cual otras puedan requerir ajustes de velocidad.



**Ilustración 14: Secuencia de acciones**

Una vez que la aeronave 1 (AC1) alcanza el punto TIE 1 y la aeronave 2 (AC2) se encuentra próxima al punto TIE 2, el controlador de Tráfico Aéreo comprueba la distancia de AC2 al punto TIE 2.

Conocidas las condiciones de velocidad, altitud y la posición de las aeronaves, el controlador chequea el tabular proporcionado por la herramienta que implementa el procedimiento OPTAIN-SA, que le proporcionará soporte para secuenciar el tráfico en el punto de integración de llegadas con la separación adecuada. El procedimiento OPTAIN-SA sugiere una indicación de velocidad determinada, dependiendo de la separación deseada en el IAF y con referencia a la distancia que exista al segundo punto TIE, de forma que se comunique el ajuste a AC2.

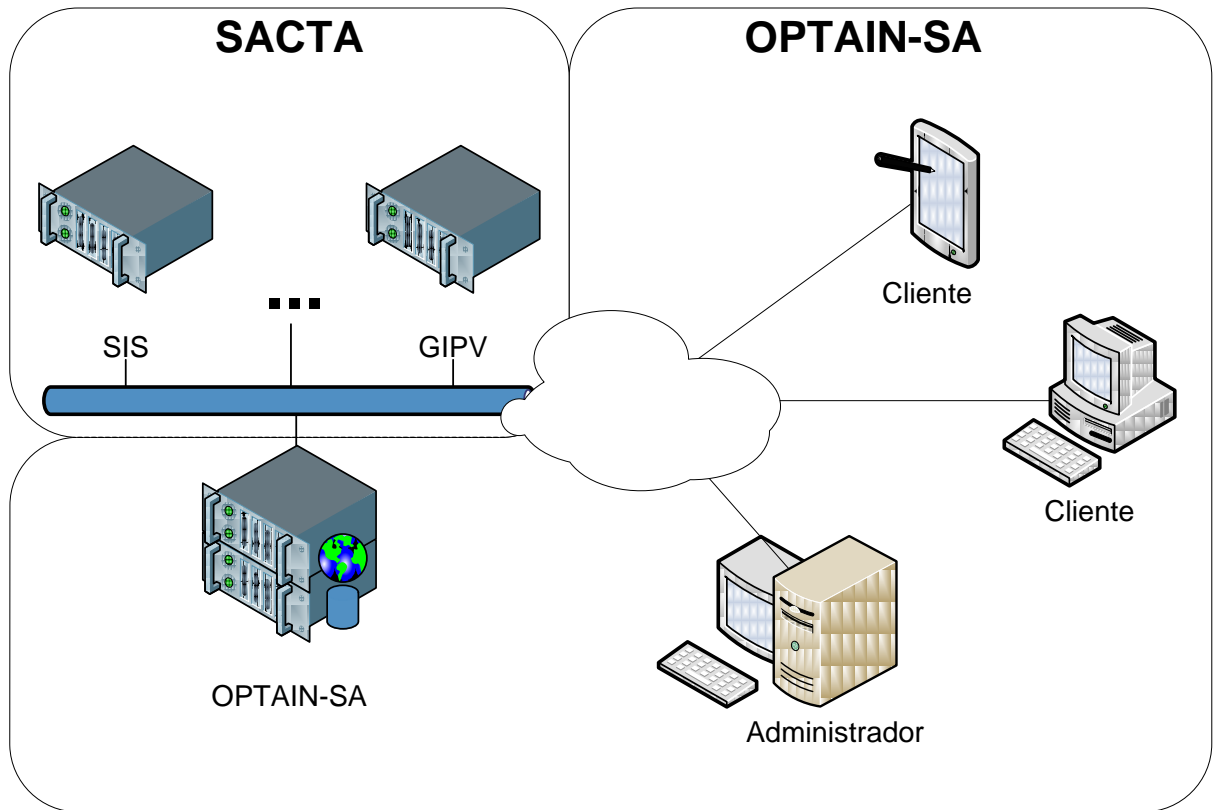
La acción que sigue en dicha secuencia corresponde a la confirmación de la instrucción para ajuste de velocidad por parte del piloto, tras lo cual la aeronave (siempre que le sea posible) seguirá la instrucción por la vía que menos tiempo de reacción le requiera.

AC1 continúa su procedimiento sin realizar ninguna acción sobre su maniobra, alcanzando un Speed Limit Point de 250kts (cumpliendo así la restricción de los procedimientos convencionales) y posteriormente cruza el punto POS, correspondiente al IAF. A su vez, la aeronave 2 (AC2), tras la corrección de la velocidad, también alcanzará su punto de límite de velocidad y POS.

Como resultado de la secuencia de acciones llevadas a cabo, se consigue el objetivo final de tener las aeronaves secuenciadas en el IAF a 5NM/10NM según sea necesario.

### 5.3. *Arquitectura de OPTAIN-SA*

La herramienta OPTAIN-SA obtiene información de los subsistemas SIS y GIPV de SACTA de forma periódica, almacenándose en su base de datos, considerándose como válidos retardos de hasta diez segundos entre el evento y la recepción en OPTAIN-SA. Dicha herramienta se compone de dos partes: servidor y cliente.



**Ilustración 15: Arquitectura de OPTAIN-SA**

Los clientes OPTAIN-SA requieren información sobre el servidor, al que se conectan a través de la red, haciendo solicitudes. Puede haber más de un cliente conectado.

El servidor de aplicaciones proporciona a los clientes los datos obtenidos de la base de datos.

#### **5.4. Modelo de datos del sistema**

El modelo de datos del sistema se trata de la base de datos de la aplicación SINA.

Por un lado se relaciona PUNTO, PV, SEGMENTO y SECTOR. Esto recoge la información dinámica del plan de vuelo.

- PUNTO: Son los puntos de la ruta
- PV: Es la información referente al Plan de Vuelo
- SEGMENTO: El espacio se divide en segmentos cada uno de ellos responsabilidad de un centro de control
- SECTOR: Son regiones del espacio

Hay que destacar también PISTA DISPI que se carga a través del SIS. Es una información dinámica de vigilancia.

El resto de las tablas son para llevar a cabo la configuración.

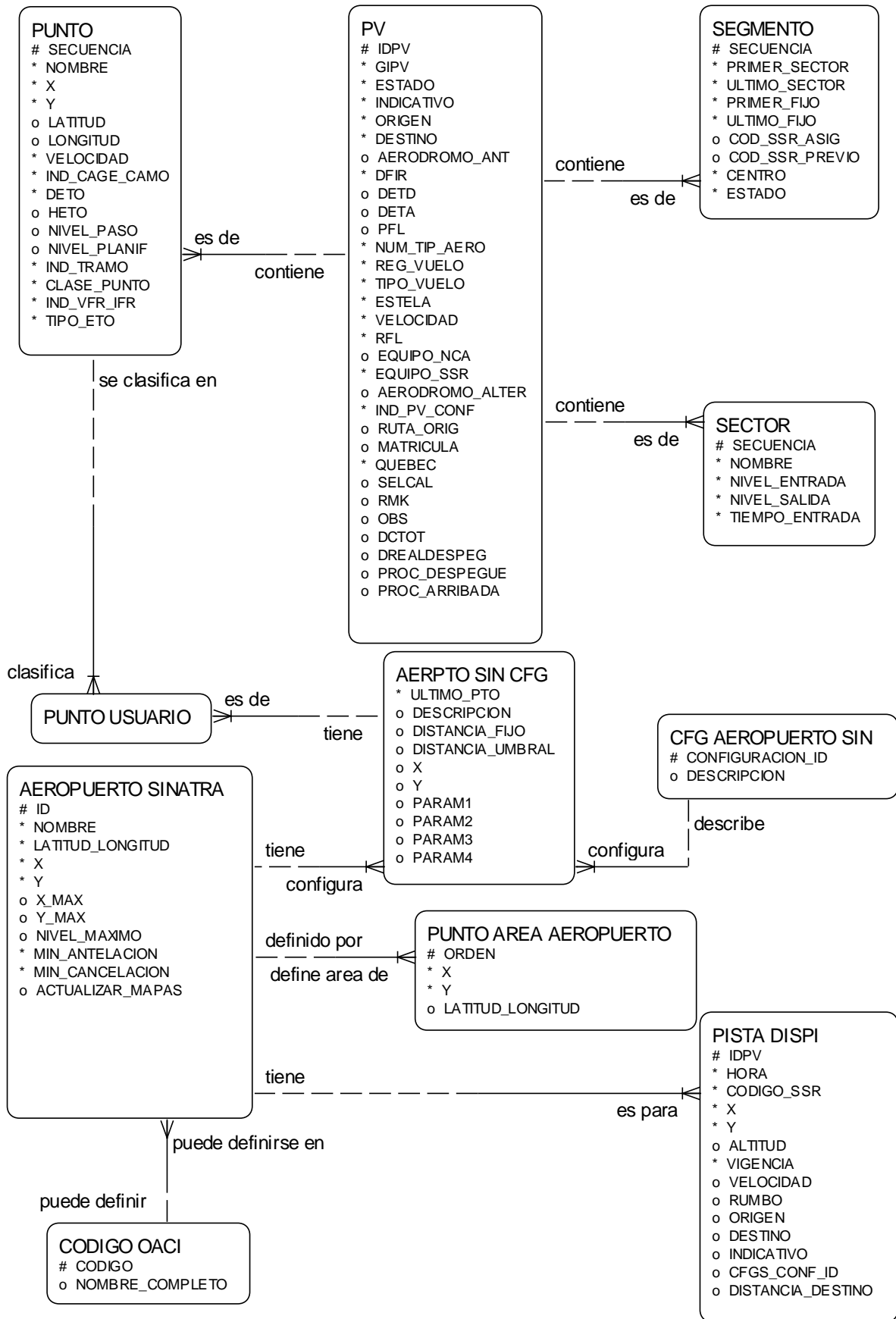


Ilustración 16: Modelo de datos del sistema

La Ilustración 17 recoge todas las tablas necesarias para los servicios de consultas. Por ejemplo consultas de planes de vuelo.

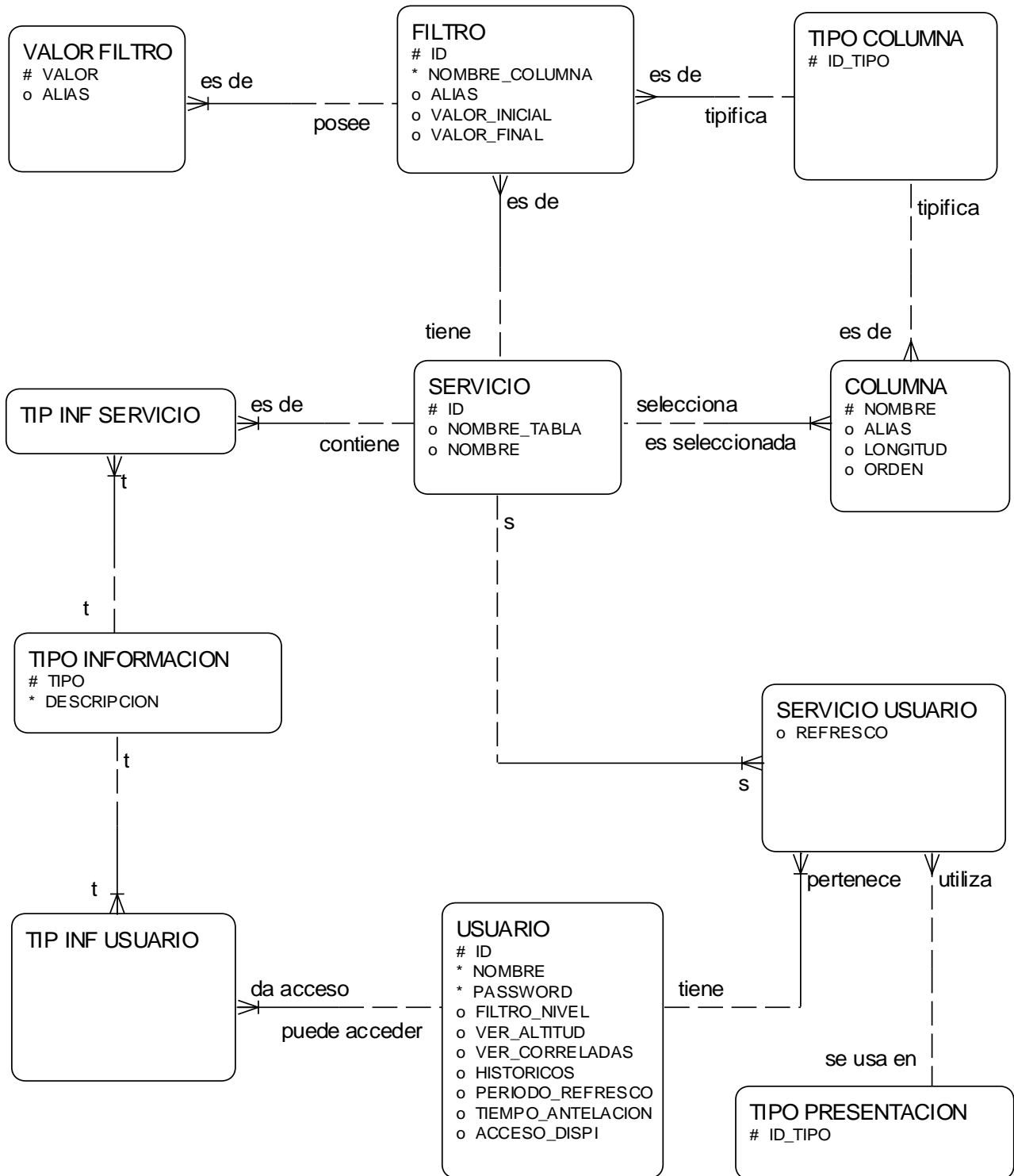


Ilustración 17: Modelo de datos del sistema



La Ilustración 18 incluye la parte de representación gráfica de SINA.

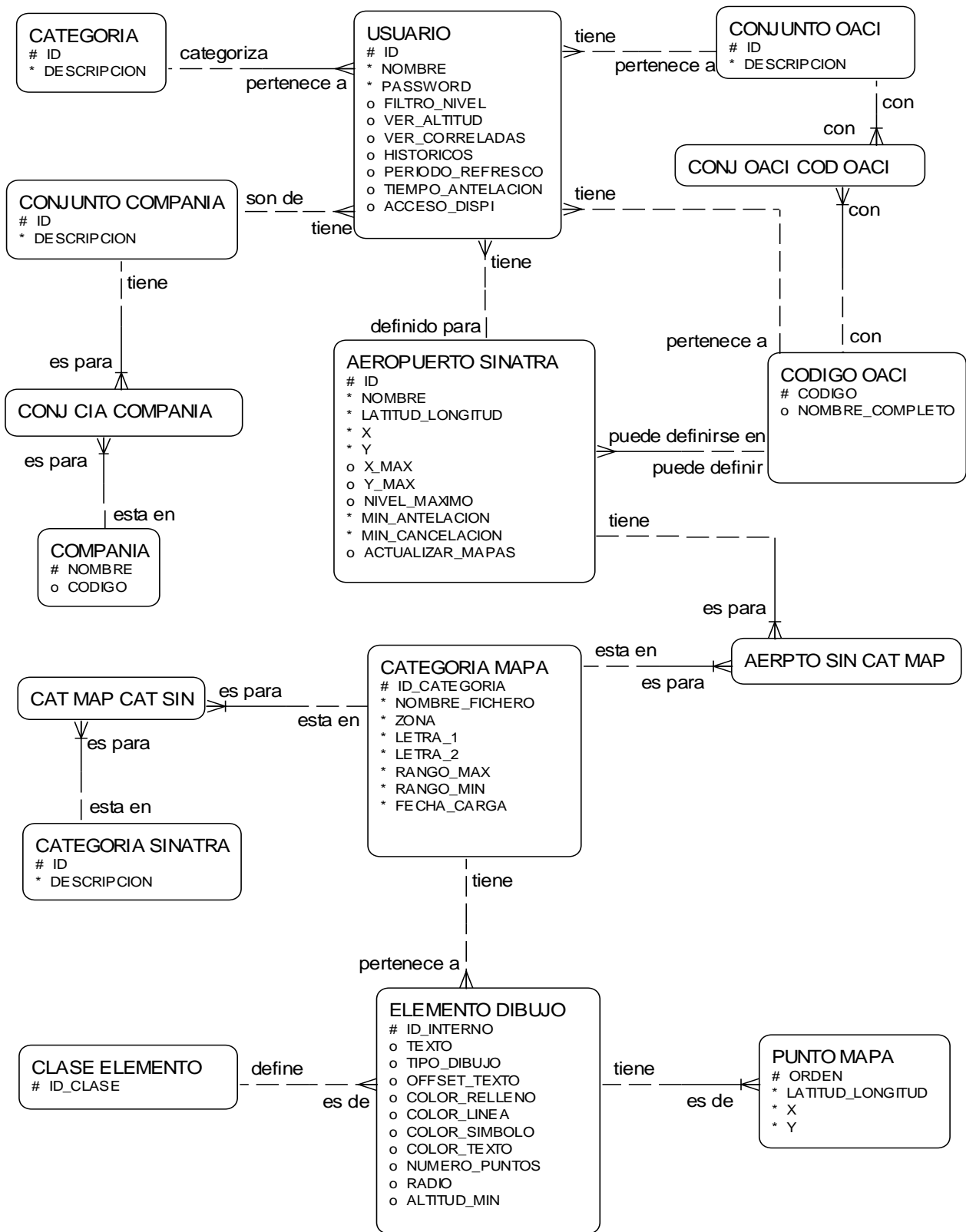


Ilustración 18: Modelo de datos del sistema

Por último se recogen todos los parámetros que se utilizan. Como por ejemplo AVISO que son los mensajes que se envían.

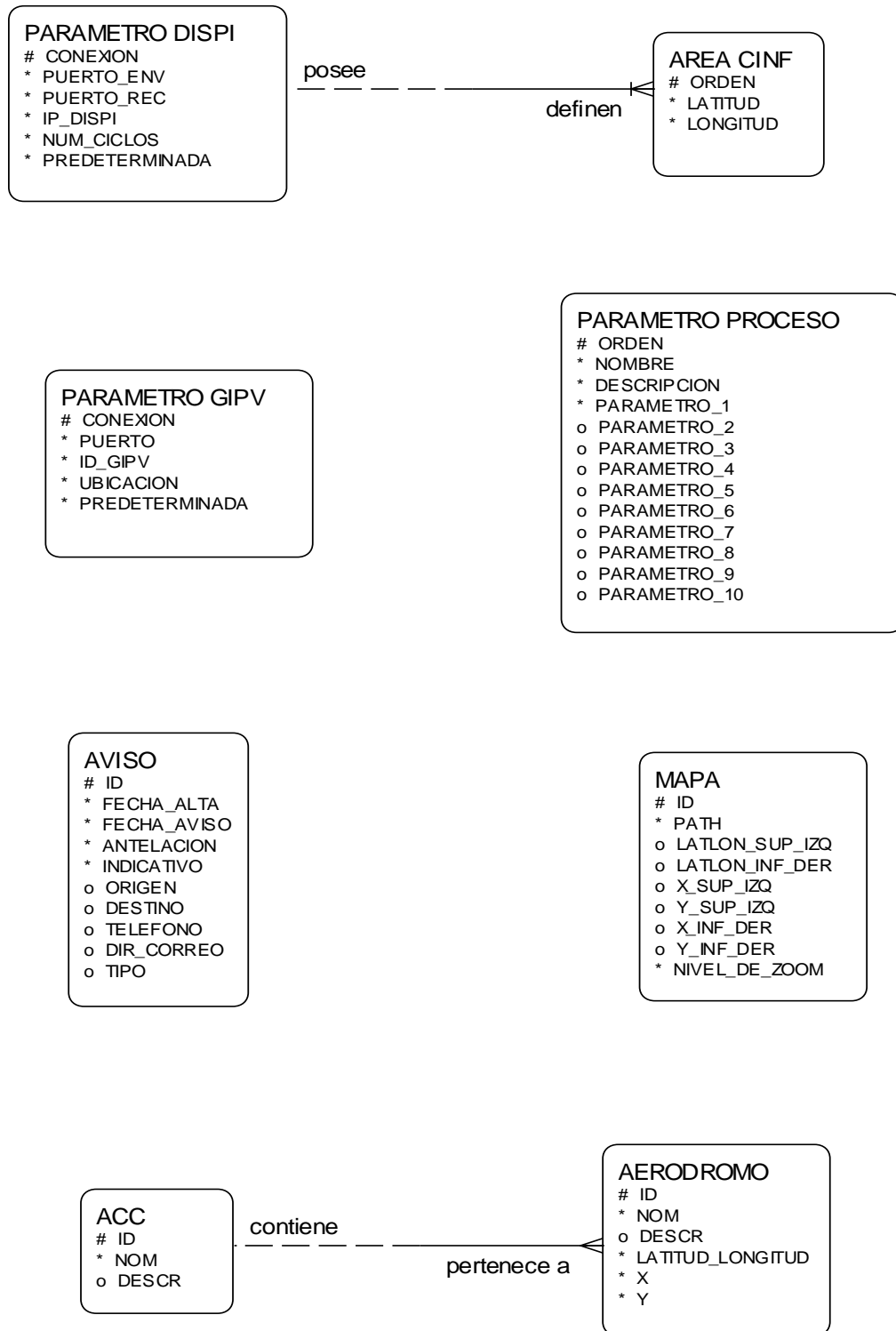


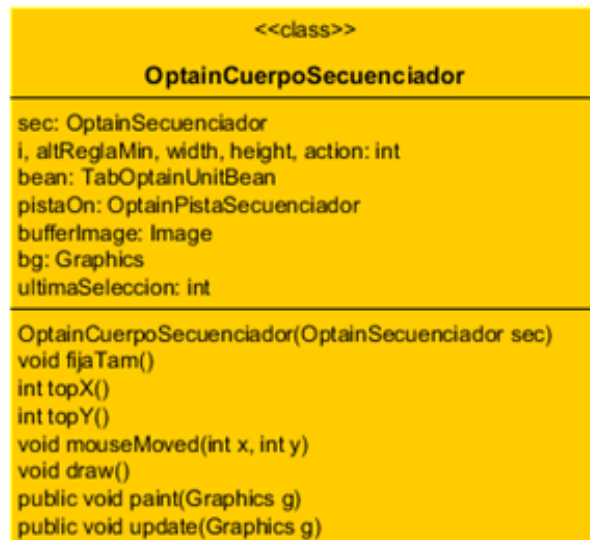
Ilustración 19: Modelo de datos del sistema

## 5.5. Clases involucradas en el sistema

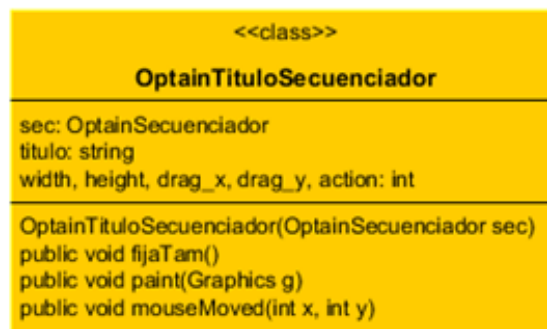
Con el objetivo de facilitar la comprensión del código desarrollado en este trabajo tanto para la resolución de los problemas encontrados como para el desarrollo de las dos funcionalidades se incluyen las clases principales del procedimiento OPTAIN especificando en primer lugar las variables y en segundo lugar sus funciones.

<pre>&lt;&lt;class&gt;&gt; OptainClientApplet</pre>	
<pre>httpCmdLoadMap, httpCmdLoadPistasList, httpCmdLoadTabsOptainList, httpCmdLoadAppletParameters, httpCmdLoadVelocidadesOptain, httpCmdGetTimestamp, httpCmdLoadInformeOptain, httpCmdAddPistaOptain, httpCmdAddPistaCandidata: string httpResSessionID, httpResClientID: string appResSessionID, appResClientID, appResPollDelay, appResServerURL: string sessionID, clientID, pollDelay, serverURL: string url: URL varg: HttpVarg request: HttpRequest dispvarg: HttpVarg dispiRequest: HttpRequest tabsOptainVarg: HttpVarg tabsOptainRequest: HttpRequest authVarg: HttpVarg authRequest: HttpRequest informeOptainVarg: HttpVarg informeOptainRequest: HttpRequest pistaOptainVarg: HttpVarg pistaOptainRequest: HttpRequest thread: Thread stopped, p true, updating: boolean gc: OptainClientAppletGC toolbar: OptainToolBar axis: OptainMapAxis map: OptainMap scheduler: OptainMapScheduler pistasContainer: OptainMapPistasContainer drawMapSchedulerOnly_flag, parametersLoaded, velocidadesLoaded: boolean parameters: AppleParametersBean drag_x, drag_y: int copy_backbuffer, action_on: boolean wPistaOculta, wRotaetiqueta: MenuItem tab1, tab2: OptainTabular tabList1, tabList2, velocidades: PList tabOver: Component sec1: OptainSecuenciador secsInitiated: boolean tabFont, secFont, drawFont: Font ojeo: boolean yClic, xClic: int</pre>	<pre>void newBackBuffer() void init() void start() void changeDrawFont(int tam) void changeTabFont(int tam) void validar() void validateStart() void stop() void destroy() String getAppletInfo() String[][] getParameterInfo() void setTabOver(Component component) boolean isTabOver(Component component) void setStopped(boolean stopped) Object loadPistasList(String aeropuerto) Object loadTabsOptainList(String aeropuerto) Object loadAppletParameters() Object loadVelocidadesOptain() Object loadMap(String aeropuerto) Object getTimestamp() Object loadInformeOptain(String fechaInicio, String fechaFin, String distancia, String valor1, String valor2, String valor3) Object addPistaOptain(String indicativoAC1, String indicativoAC2, String separacionTIE, String ajusteVel, String velAC1TIE, String velAC2TIE, String sepcionIAF, String pasoTIE, String velocidadAC1IAF, String velocidadAC2IAF, String tieAC1, String tieAC2, String dateTimeProc) Object addPistaCandidata(String indicativoAC1, String indicativoAC2, String dateTimeProc) void update() void update(MapBean map) void update(PList pistasList) void run() void mouseClicked(MouseEvent e) void mousePressed(MouseEvent e) void mouseReleased(MouseEvent e) void mouseEntered(MouseEvent e) void mouseExited(MouseEvent e) void mouseDragged(MouseEvent e) void mouseMoved(MouseEvent e) void update(Graphics g) void draw(Graphics g) void paint(Graphics g) void setPainted ( ) void waitPainted() void setUpdating(boolean updating) void waitUpdated()</pre>

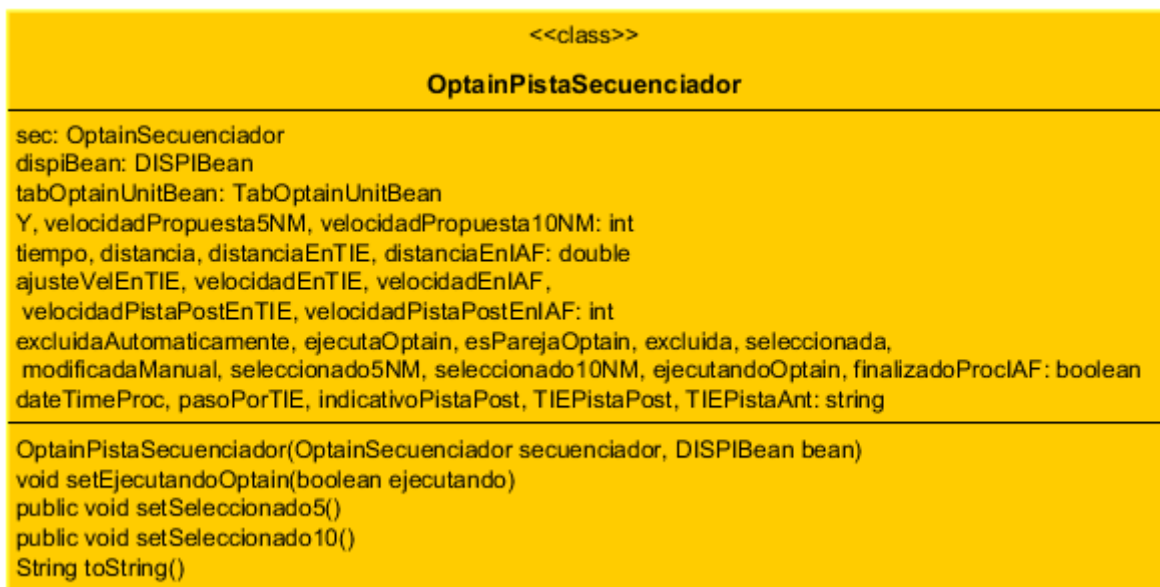
Ilustración 20: OptainClientApplet



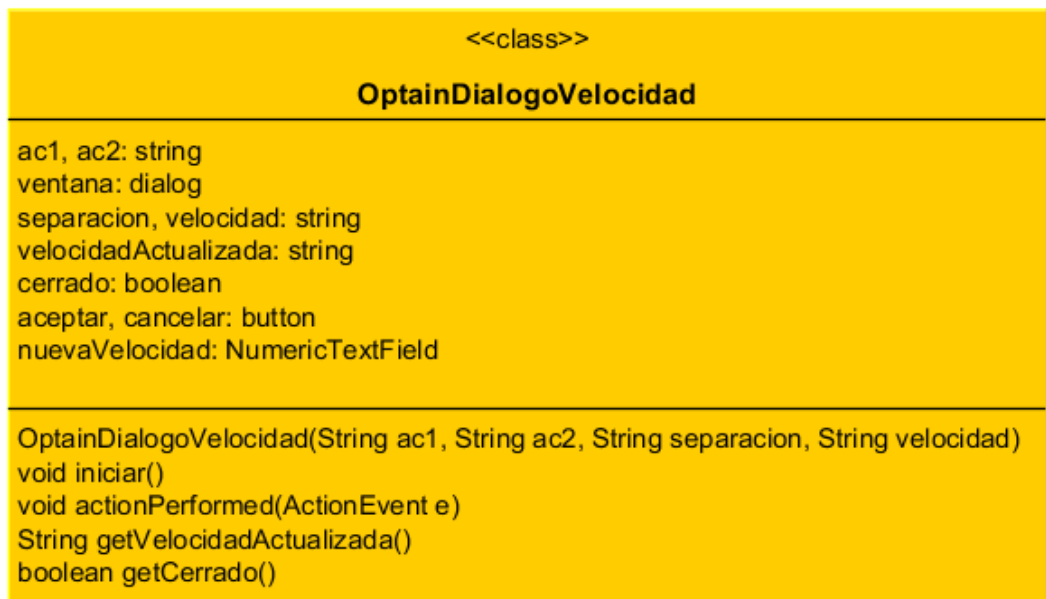
**Ilustración 21: OptainCuerpoSecuenciador**



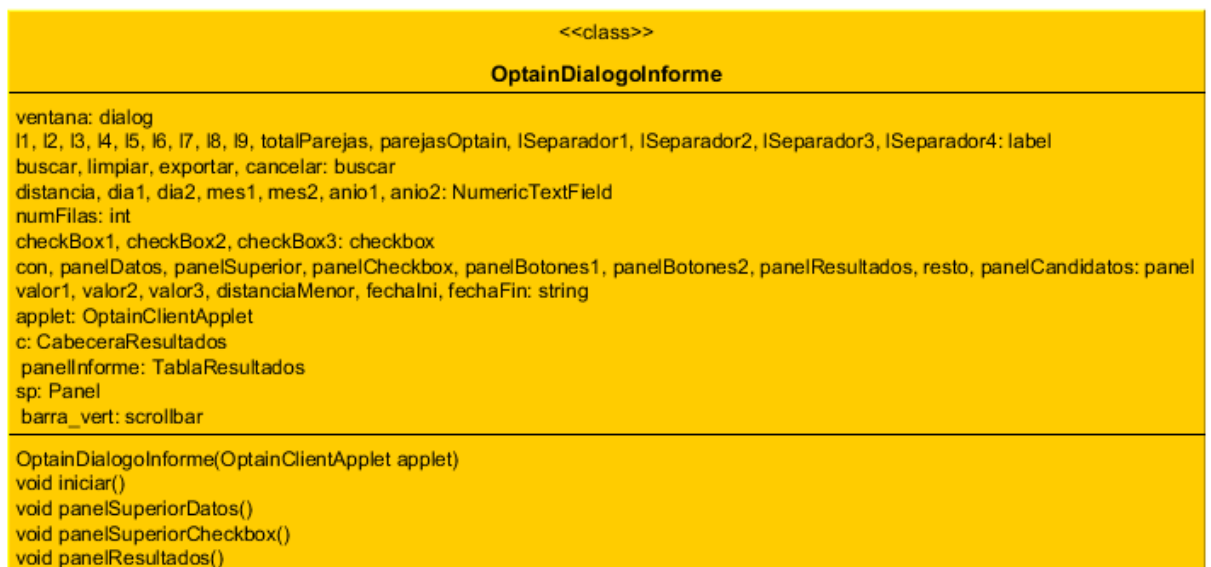
**Ilustración 22: OptainTituloSecuenciador**



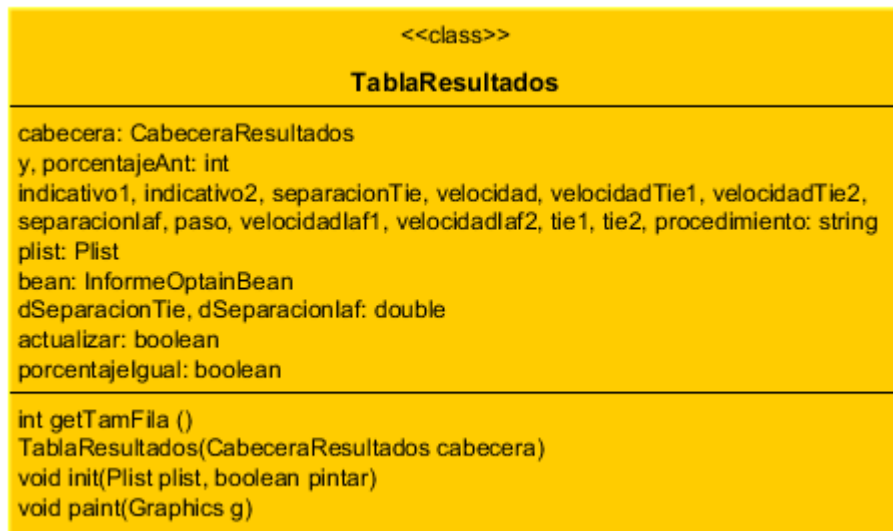
**Ilustración 23: OptainPistaSecuenciador**



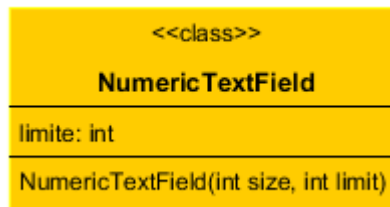
**Ilustración 24: OptainDialogoVelocidad**



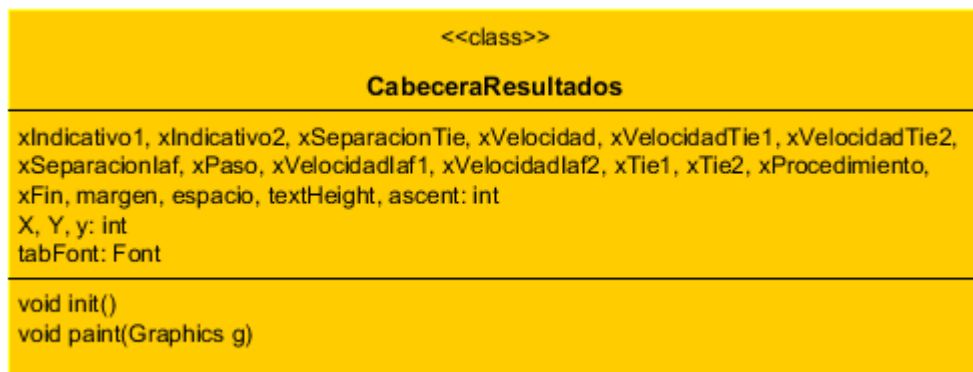
**Ilustración 25: OptainDialogoInforme**



**Ilustración 26: TablaResultados**



**Ilustración 27: NumericTextField**



**Ilustración 28: CabeceraResultados**

```

<<class>>
OptainSecuenciador

applet: OptainClientApplet
config: ConfigBean
titulo: String
X, Y, newX, newY, width, height: int
barraTitulo: OptainTituloSecuenciador
resto: Panel
cuerpo: OptainCuerpoSecuenciador
font: Font
ascent, descent, textHeight, Xmax: int
pistasOptain, pistasOptainExt, velocidadesOptain: Plist
pressed, dragged, cambiandoFont, exist: boolean

private doubleescala;

void excluyePista(OptainPistaSecuenciador pista)
void excluyeIncluyeSeleccionada()
OptainPistaSecuenciador getOptainPistaSecuenciador(final String indicativo)
OptainPistaSecuenciador getOptainPistaAnteriorSecuenciador(final String indicativo)
void actualizaPistasExt(OptainPistaSecuenciador optainPistaSecuenciador)
void limpiaPistasExt()
String getVelocidadPropuesta(String indicativo, int separacionRequerida)
OptainSecuenciador(OptainClientApplet applet, ConfigBean config, Plist velocidadesOptain, int x, int y, int height, boolean visible, boolean exist)
void init(boolean visible)
void setVars()
void cambiaFuente()
void fijaTam()
double getEscala()
void reset()
void addPista(DISPIBean bean)
void beginDrag()
void paint(Graphics g)
void paintLigero(Graphics g)
void mover(int dx, int dy)
void endDrag()
boolean filterPista(DISPIBean bean)
String formatTiempo(double tiempo)
void calculaVelocidadPropuesta(OptainPistaSecuenciador optainPista, double distancia)
void calculaVelocidadPropuesta5NM(OptainPistaSecuenciador optainPista, double distancia)
void calculaVelocidadPropuesta10NM(OptainPistaSecuenciador optainPista, double distancia)

```

Ilustración 29: OptainSecuenciador

## 6. Construcción

Los apartados que se desarrollan a continuación se centrarán en la descripción breve del proceso de preparación del entorno de construcción, centrándose en las necesidades del mismo.

Para conseguir la infraestructura necesaria para la generación del código de los componentes del sistema de información será necesario instalar y configurar todas las herramientas de desarrollo necesarias y todas las herramientas adicionales que permitan el funcionamiento de los prototipos.

### 6.1. *Preparación del entorno de generación y construcción*

El objetivo de esta actividad es garantizar la disponibilidad de un entorno sobre el cual poder construir con garantías el sistema de información. Los aspectos a tener en cuenta son entre otros:

- Instalación y configuración de las herramientas de desarrollo necesarias
- Preparación de los equipos de trabajo.
- Reserva de espacio de almacenamiento.

#### 6.1.1. Herramientas

La herramienta utilizada para el desarrollo de la aplicación es Eclipse, se trata de una plataforma de desarrollo open source basada en Java. Es un desarrollo de IBM (International Business Machines) cuyo código fuente fue puesto a disposición de los usuarios.

En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in). Hay plug-ins para el desarrollo de Java (JDT, Java Development Tools) así como el desarrollo en C/C++, COBOL, etc.

El lenguaje de programación empleado ha sido Java en la versión 1.0, se trata del primer lanzamiento del lenguaje Java en el año 1996. Entre otras cosas se diferencia de las siguientes versiones de Java porque es una versión sin swing. Swing es una biblioteca gráfica que incluye widgets para la interfaz gráfica de usuario tales como botones, tablas, cajas de texto, etc.

Otra herramienta que se ha utilizado en el desarrollo de la memoria para la realización de muchos de las ilustraciones es yEd Graph Editor se trata de una aplicación de escritorio gratuita para generar una amplia variedad de diagramas para expresar ideas complejas de forma sencilla.

Para realizar la planificación temporal se ha utilizado la herramienta OpenProj, un software libre de administración de proyectos. OpenProj controla todos los aspectos referentes a la gestión de proyectos, como la planificación, la gestión y asignación de recursos, la simulación de alternativas en procesos críticos, etc.



### 6.1.2. Equipo de trabajo

El equipo en el que se ha llevado a cabo el desarrollo de lo que se incluye en este trabajo dentro del desarrollo de la aplicación OPTA-IN presenta las siguientes características:

Sistema Operativo	Microsoft Windows 7 Enterprise
Versión	6.1.7601 Service Pack 1 Compilation 7601
Modelo del sistema	HP Compaq Elite 8300 CMT
Memoria física instalada (RAM)	4,00 GB
Memoria física total	3,95 GB
Memoria virtual total	7,89 GB

**Tabla 1: Características equipo de trabajo**

## 7. Manual de usuario

Se incluye a continuación un manual de usuario en detalle para comprender el funcionamiento de la aplicación.

### 7.1. *Iniciando una sesión*

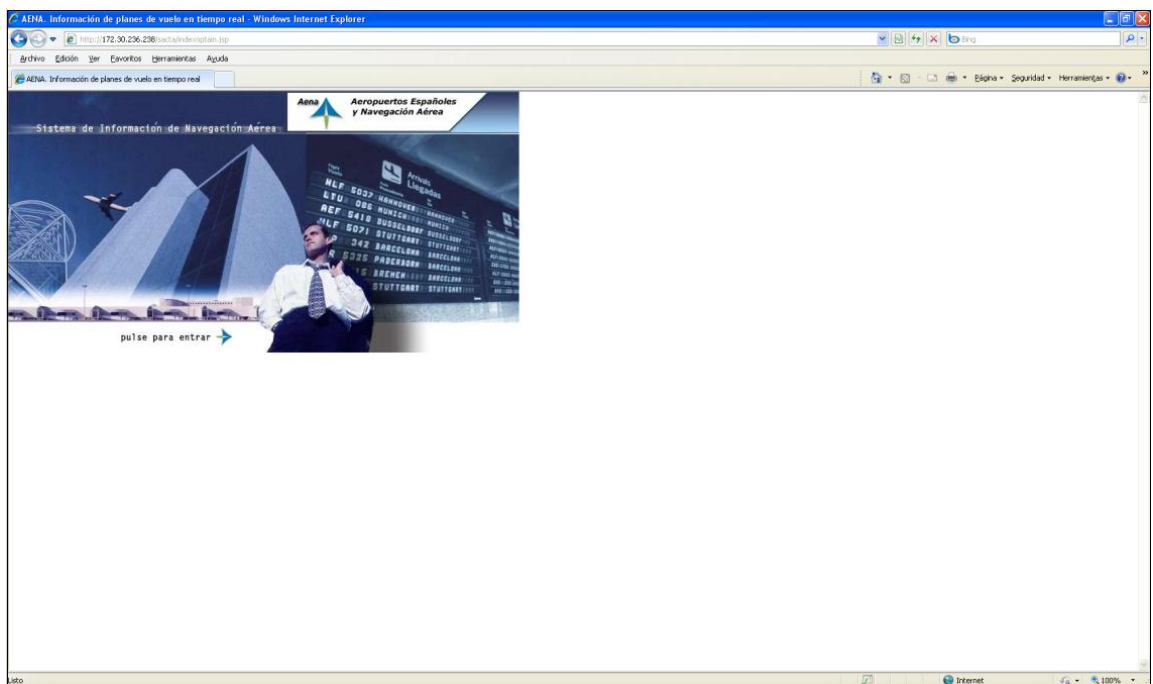
El procedimiento para arrancar la herramienta ATC de apoyo es el siguiente:

Encender el cliente y una vez haya arrancado el equipo, hacer doble click en el acceso directo OPTA-IN, ubicado en el escritorio de Windows, para acceder al navegador.



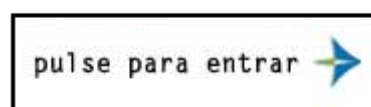
**Ilustración 30: Icono de la aplicación**

Y se abrirá un navegador, en el que se podrá ver lo siguiente:



**Ilustración 31: Ventana de acceso a la herramienta**

Para poder acceder a la herramienta, se debe pulsar en “Pulse para entrar”



**Ilustración 32: Botón “Pulse para entrar”**

Introducir Usuario y Contraseña en los campos correspondientes.

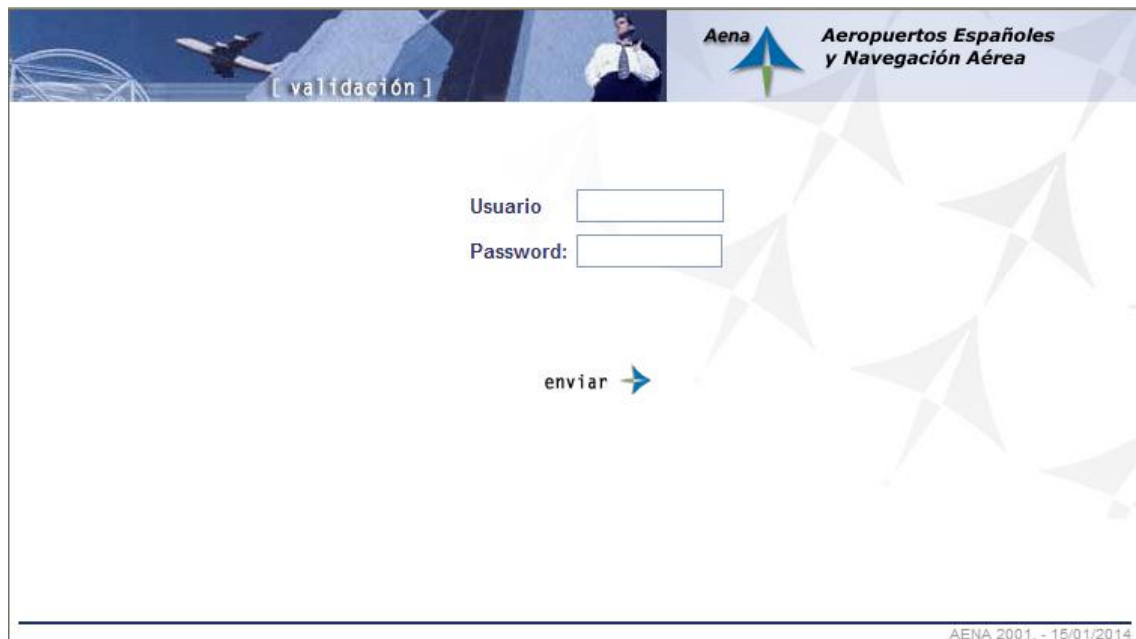


Ilustración 33: Ventana de login

Una vez introducidos los datos correctamente, hacer click en el botón “enviar”

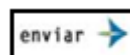


Ilustración 34: Botón “Enviar”

Según la configuración del cliente, puede salir una alerta de seguridad como la que se muestra a continuación:

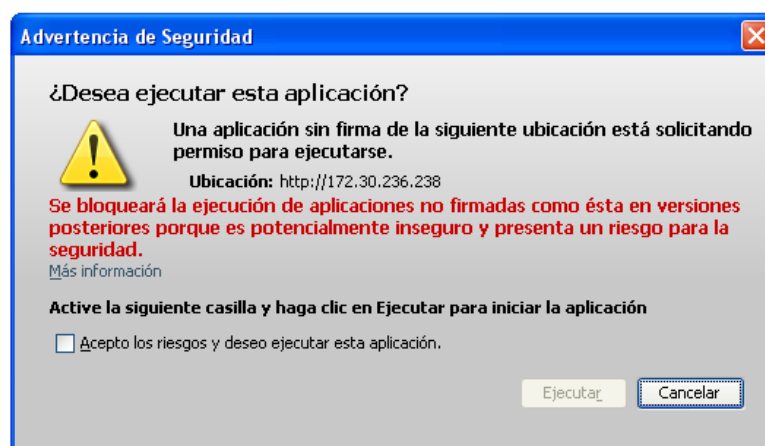
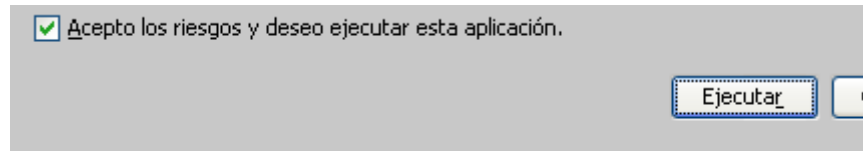


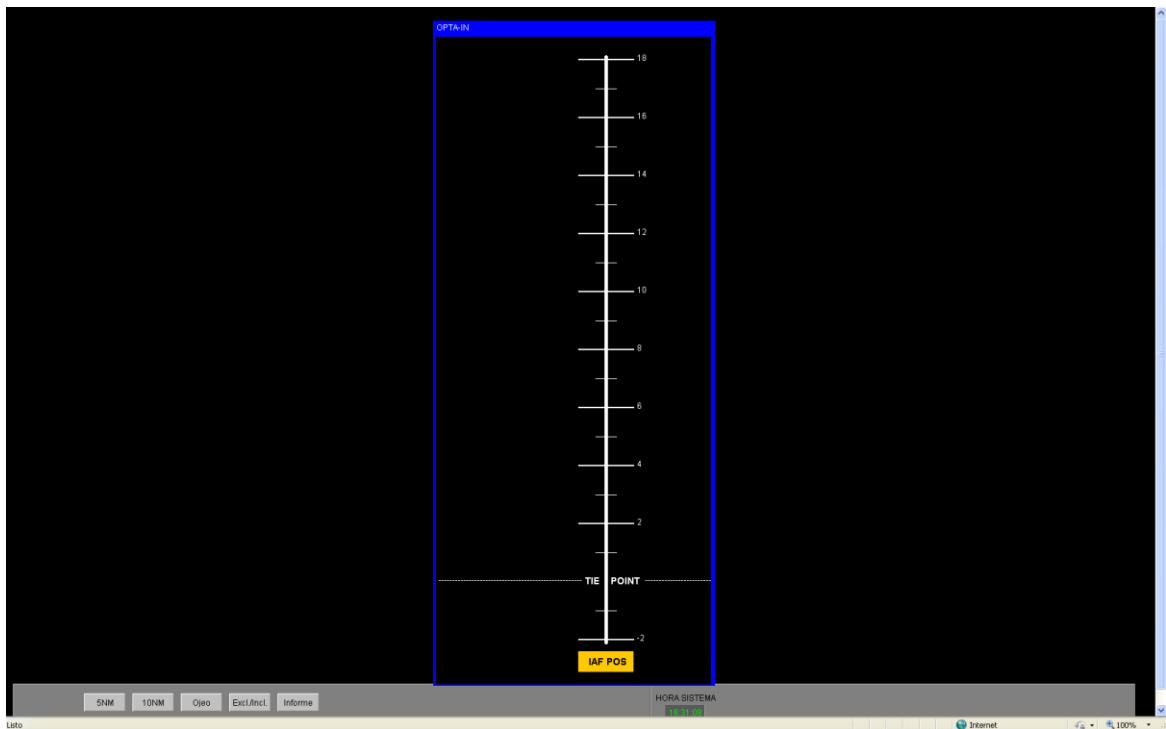
Ilustración 35: Ventana de alerta de seguridad

Si es así, habrá que marcar la casilla que aparece en la parte inferior de la ventana (“Acepto los riesgos y deseo ejecutar la aplicación”), y pulsar el botón “Ejecutar”



**Ilustración 36: Detalle de la ventana de alerta**

Una vez hecho esto, ya se tiene acceso a la ventana principal de la herramienta.



**Ilustración 37: Ventana principal de OPTA-IN**

En ella se podrá ver la ventana OPTA-IN, que mostrará gráficamente la secuencia de vuelos de llegada al IAF del aeropuerto LEPA desde los flujos de tráfico correspondientes, sobre una regla de distancia vertical, junto con el ajuste de velocidad dado por el procedimiento para las dos separaciones consideradas (5NM/10NM).

Asimismo, se podrá acceder a su vez a ventanas auxiliares con información de los vuelos de dicha secuencia en formato tabular, particularizada la velocidad propuesta por el procedimiento para una sola de las dos separaciones.

## **7.2. Finalizando una sesión**

El procedimiento para cerrar la herramienta es tan sencillo como cerrar la ventana del navegador web utilizado (Internet Explorer), como se hace habitualmente en los ordenadores con sistema operativo Windows.



**Ilustración 38: Botón para cerrar la aplicación OPTA-IN**

Otra opción, es usar la combinación de teclas “ALT+F4”.

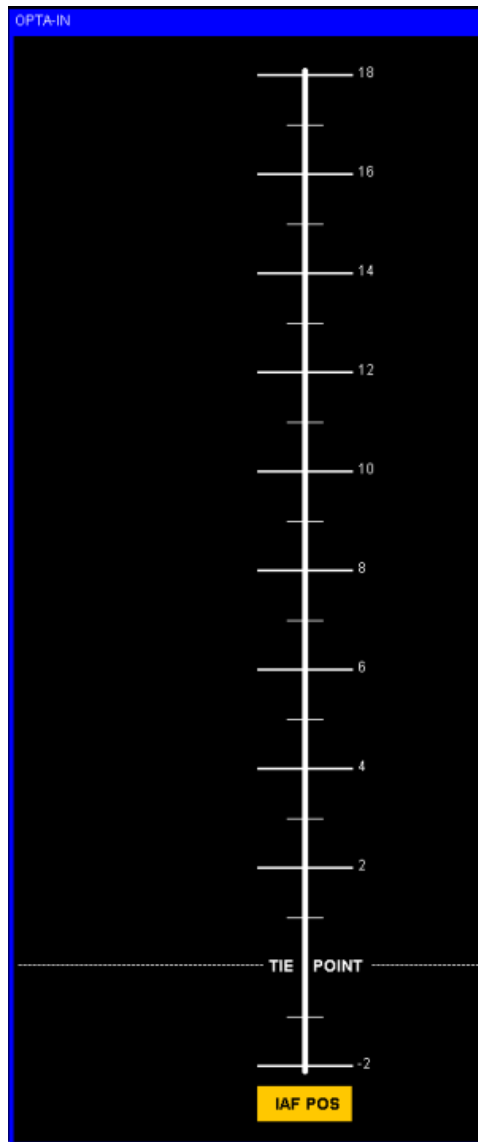


**Ilustración 39: Combinación de teclas para cerrar la aplicación OPTA-IN**

### **7.2.1. Funcionalidades de la herramienta**

#### **7.2.2. Ventana OPTA-IN**

La ventana OPTA-IN muestra gráficamente la secuencia de vuelos de llegada al IAF del aeropuerto LEPA, sobre una regla de distancia vertical, con escala de 1 NM, sin hacer distinción del flujo de procedencia.

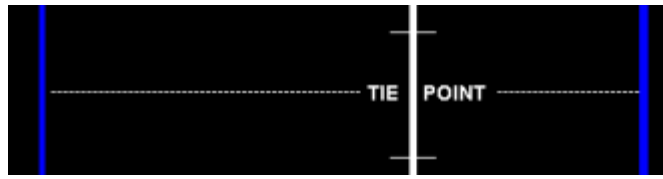


**Ilustración 40: Ventana de regla vertical OPTA-IN**

Además del aeropuerto de destino, los vuelos deben cumplir otros requisitos para ser mostrados en la ventana. Son los siguientes:

- El vuelo se encuentra dentro del área geográfica OPTA-IN definida (corona circular).
- Pasar por el IAF de POLLENSA (POS).

Los vuelos se muestran desde 18NM antes de llegar al TIE Point, hasta una vez sobrevolado dicho punto. Este estará marcado por una línea horizontal discontinua, como se muestra en el detalle de la siguiente imagen.

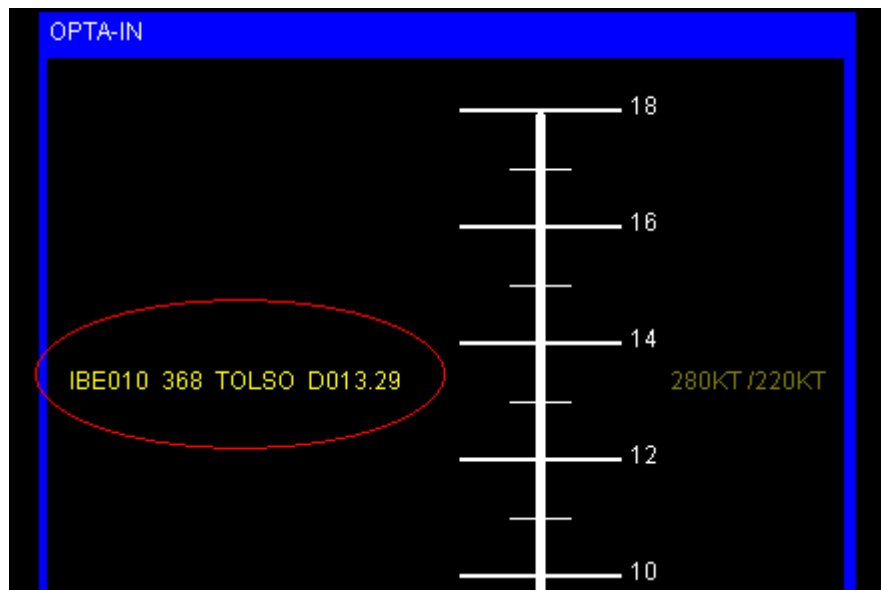


**Ilustración 41: Detalle del TIE Point**

En la parte inferior de la ventana también se muestra de forma simbólica el punto IAF, donde convergen los flujos de tráfico.

A la izquierda de la regla de distancia se muestra un tabular a la altura correspondiente de cada vuelo. La información mostrada en el tabular de cada vuelo será la siguiente:

- Identificador del vuelo
- Velocidad actual del vuelo
- Flujo de entrada del vuelo, que vendrá dado por el punto identificativo (TOLSO, LORES o KENAS)
- Distancia del vuelo al TIE Point.



**Ilustración 42: Detalle del tabular**

La posición del tabular será fiel, en principio, a la distancia del mismo respecto al punto TIE y al IAF, salvo que el excesivo número de vuelos dentro de un intervalo de la regla provoque limitaciones de espacio, en cuyo caso se representan los vuelos apilados sin reflejar la distancia real.

A la derecha de la regla de distancia se muestra, para cada vuelo, otro tabular con las velocidades propuestas para conseguir una separación de 5NM (la primera velocidad) y 10NM (la segunda) con la aeronave precedente. En el caso de que no haga falta ningún ajuste de velocidad para conseguir dichas separaciones, ya sea de 5 o 10 NM, la velocidad por defecto será de 280kt. Si por el contrario, debido a la escasa separación entre los vuelos, es imposible separarlos, la velocidad indicada será “N.A.”. En el caso de que no se pueda conseguir ninguna de las separaciones, estos vuelos se consideran excluidos de la secuencia. En el tabular de la secuencia aparece “N.A./N.A.”.



230KT /N.A.

**Ilustración 43: Ejemplos de tabulares de velocidad**



N.A./N.A.

**Ilustración 44: Ejemplos de tabulares de velocidad**

El color en el que se muestra la información de estos tabulares, depende del estado de los vuelos. Los posibles estados son los siguientes:

- *Candidato* (amarillo): son aquellos vuelos que pueden formar parte de un procedimiento OPTA-IN, ya que se encuentran a una distancia adecuada de su vuelo precedente en la secuencia de llegada al IAF.
- *En procedimiento* (verde): son aquellos vuelos que ya han sido seleccionados por el controlador para aplicar un procedimiento OPTA-IN.
- *Excluido* (rojo): son aquellos vuelos que no pueden aplicar un procedimiento OPTA-IN, ya sea porque el controlador así lo ha indicado manualmente, o bien porque se encuentran demasiado próximos a su vuelo precedente.



WHGT454 368 TOLSO D012.85

RSSS658 369 TOLSO D007.84

GRN003 367 TOLSO D012.26

**Ilustración 45: Ejemplo de vuelos en cada uno de los estados**

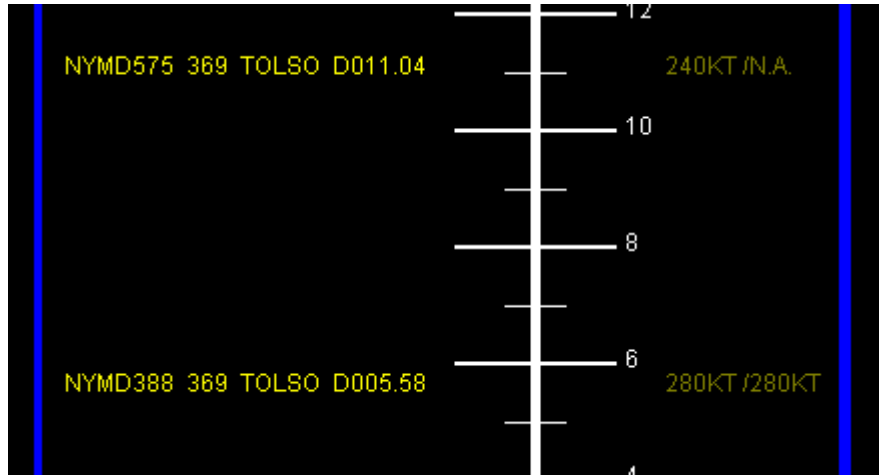
El primer tabular de la pareja que se encuentra aplicando el procedimiento (en verde) empieza a parpadear 1NM antes de llegar al TIE Point, y hasta 1NM una vez sobrevolado dicho punto, con el fin de avisar al controlador de que es el momento en el que debe comunicar a la aeronave posterior la velocidad que debe coger para conseguir la separación deseada (ya sea de 5 ó 10 NM).



### 7.2.3. Activación de un procedimiento OPTA-IN:

A continuación se detallan los pasos necesarios para indicar que una pareja de vuelos candidatos van a aplicar un procedimiento OPTA-IN.

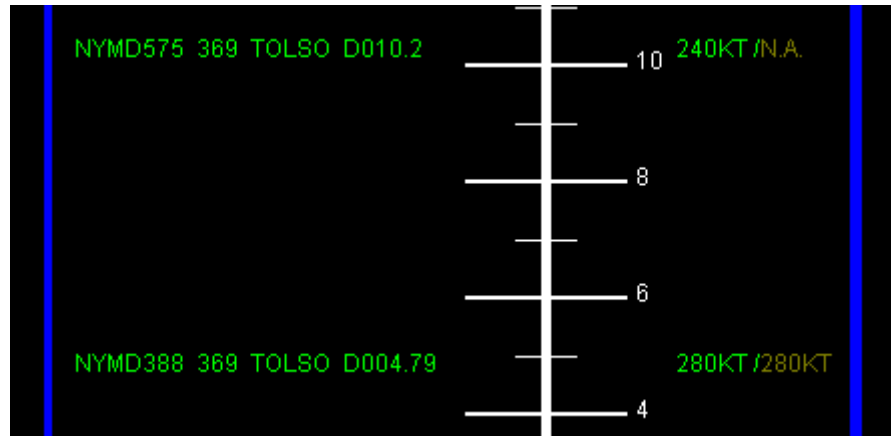
Lo primero es identificar una pareja de vuelos candidatos, que como ya se ha explicado, se representan en la ventana OPTA-IN con un tabular de color amarillo.



**Ilustración 46: Pareja de vuelos candidatos**

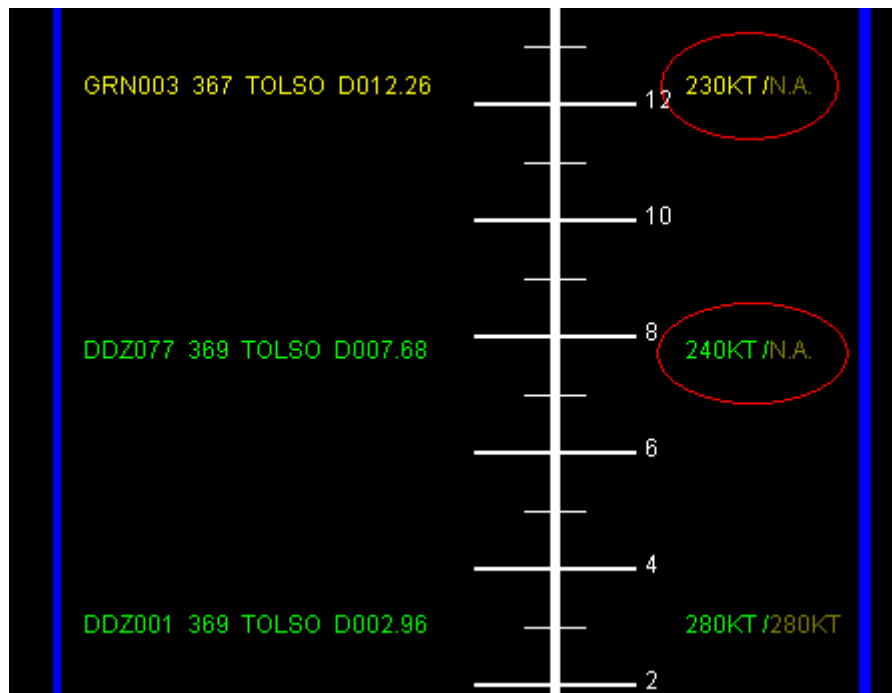
Una vez se tenga identificada la pareja de aeronaves que se quiere separar, se pasa a activar el procedimiento de la siguiente manera:

Sea AC1 la aeronave que va delante, y AC2 la que le precede. En la parte del tabular de AC2 que está a la derecha de la regla de distancia, como ya se explicó, se encontrarán dos velocidades, la primera es la que debería llevar AC2 para llegar al IAF con una separación de 5NM sobre AC1, y la segunda para que la separación sea de 10NM. Se selecciona la velocidad según la separación que se quiera conseguir en el IAF (en la siguiente sección se explicará con detalle cómo se selecciona la velocidad), y en ese momento, los tabulares de AC1 y AC2 pasan a color verde, indicando así que esos dos vuelos van a aplicar un procedimiento OPTA-IN. Además, en el tabular de AC2, se atenuará la velocidad que no ha sido seleccionada.



**Ilustración 47: Pareja de vuelos aplicando el procedimiento OPTA-IN**

Una vez se ha seleccionado la indicación de velocidad que se le va a dar a un vuelo para proporcionar una secuencia a 5 ó 10 NM, en los vuelos posteriores de forma automática esta resaltada la velocidad indicada para la misma secuencia (5/10 NM) hasta que no se seleccione otra opción. Es decir, si se ha decidido separar AC1 y AC2 5NM, todos los vuelos que vengan por detrás, tendrán la velocidad indicada para una separación de 5NM resaltada, como se puede ver en la siguiente imagen. Pero eso no quiere decir que se tenga que seleccionar esa velocidad forzosamente para el resto de procedimientos que se quieran aplicar. En cualquier momento el controlador puede cambiar, y separar los vuelos 10NM, ya que la velocidad indicada para esa separación seguirá activa.



**Ilustración 48: Detalle de las velocidades resaltadas y atenuadas**

Para concluir, en el momento en el que AC1 esté a menos de 1NM de su TIE Point (momento en el que el tabular empezará a parpadear), el controlador se comunica con AC2 para indicar la velocidad que debe tomar (que es la seleccionada), para así conseguir la separación deseada en el IAF.

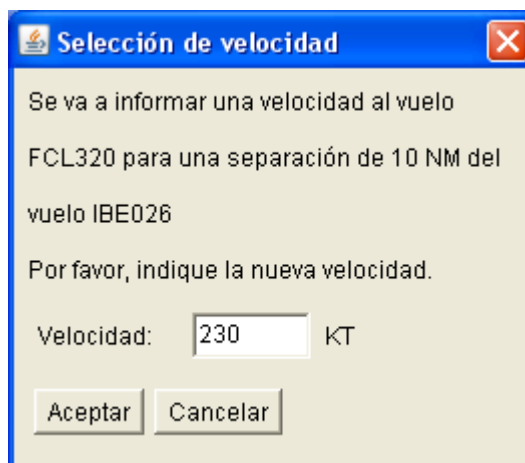
#### **7.2.4. Selección de velocidad para aplicar un procedimiento:**

Como se ha dicho en el apartado anterior, uno de los pasos fundamentales a la hora de aplicar el procedimiento, es la selección de la velocidad. Se detalla a continuación que se debe hacer para realizar dicha acción:

- El primer paso es decidir qué separación se desea conseguir entre las aeronaves, 5 ó 10 NM (si ambas opciones están disponibles).
- Una vez se tenga eso claro, se selecciona la velocidad con un clic de ratón sobre la misma, usando el botón izquierdo.
- En ese momento se abre una ventana, donde por defecto viene indicada la velocidad dada por las tablas de ajuste de velocidad. Si se quiere aplicar el procedimiento con el ajuste de esa velocidad, bastará con hacer clic en el botón “Aceptar”. Si por el contrario, el controlador estima que hay una velocidad mejor, y así lo quiere reflejar en el tabular, tendrá que modificar manualmente el valor indicado, y una vez modificado, hacer clic en “Aceptar”.

NOTA: Además de la velocidad seleccionada, en esta ventana se muestra más información: indicativo del vuelo seleccionado, separación seleccionada, indicativo del vuelo con el que se empareja en el procedimiento OPTA-IN.

- Para cerrar la ventana sin guardar ninguna velocidad, con lo que no se aplicaría el procedimiento, basta con hacer clic sobre el botón “Cancelar”.



**Ilustración 49: Ventana de selección de velocidad**

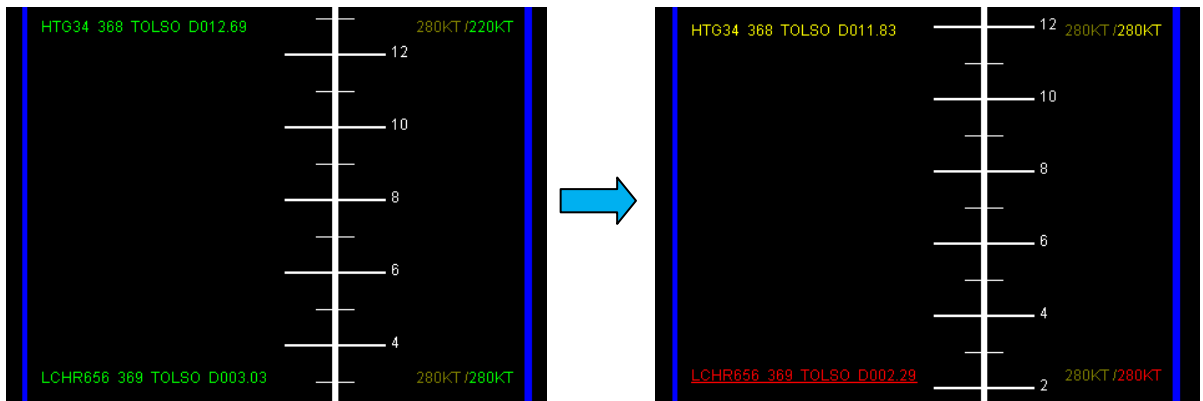
Después de este proceso, como ya se ha explicado en el apartado anterior, se va a ver que los tabulares de los dos vuelos implicados en el procedimiento pasan a color verde, y en el caso del segundo, se resalta la velocidad seleccionada en el tabular de la derecha.

**NOTA:** Una vez seleccionada una velocidad, y activado el procedimiento OPTA-IN, el controlador puede, si así lo desea, modificar manualmente la velocidad indicada. Para ello tendrá que repetir los pasos explicados anteriormente, pero sobre el vuelo que ya está en procedimiento (verde).

### 7.2.5. Cancelación de un procedimiento OPTA-IN

No existe una acción directa para poder cancelar un procedimiento que ya se ha activado, pero si existen formas de hacerlo indirectamente.

La forma más sencilla para cancelar un procedimiento es excluir uno de los vuelos implicados en dicho procedimiento. Al hacer eso, la herramienta pasará a mostrar el vuelo excluido en color rojo, y el otro que formaba parte del procedimiento, como vuelo candidato (en amarillo).



**Ilustración 50: Actualización del estado de los vuelos al excluir uno de ellos**

Ahora bien, dependiendo de la secuencia, puede haber otras formas de cancelar un procedimiento OPTA-IN. Supongamos que uno de los dos vuelos tiene delante otro lo suficientemente cerca como para no poder separarlo, el cual está excluido de forma manual. Entonces, si se vuelve a incluir ese vuelo de forma manual, automáticamente el que le precede pasa a estado excluido, ya que por proximidad no puede separarse, y el procedimiento OPTA-IN en el que iba a participar se cancela.

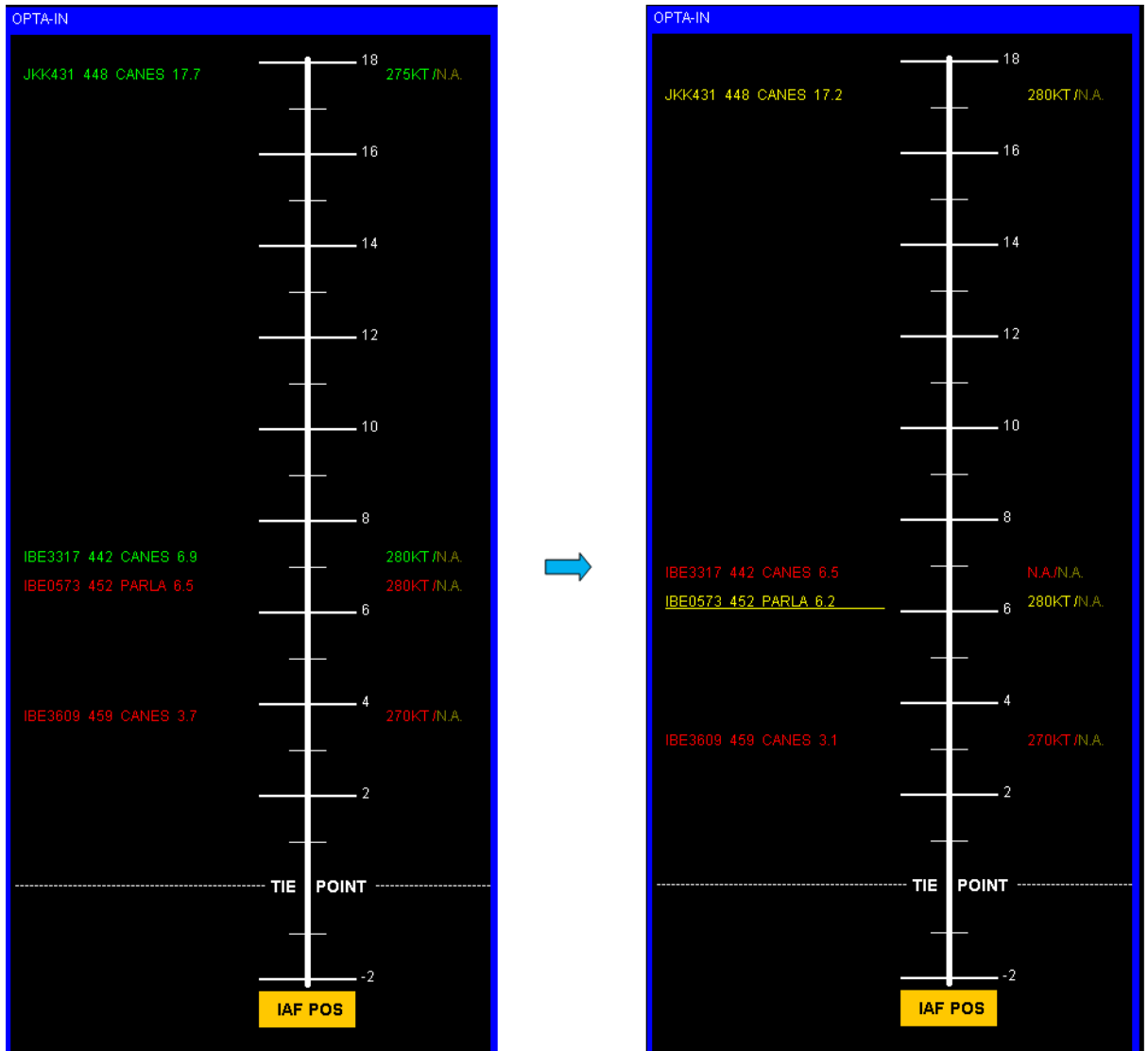


Ilustración 51: Actualización del estado de los vuelos al incluir un vuelo que estaba excluido

### 7.2.6. Ventanas secundarias de 5 y 10 NM

Además de la ventana OPTA-IN, que es la principal de la aplicación, la herramienta puede mostrar otras ventanas secundarias, entre ellas las ventanas de “5NM” y “10NM”.

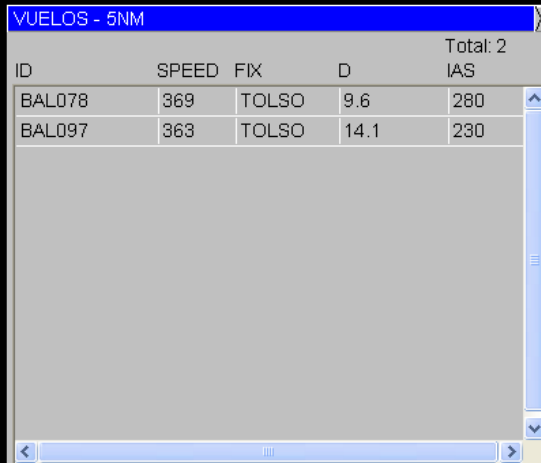
Estas ventanas se abren al hacer clic con el botón izquierdo del ratón sobre los botones respectivos que se pueden encontrar en la barra de botones inferior.



Ilustración 52: Botones de “5NM” y “10NM”

Estas ventanas muestran las parejas de vuelos candidatas para lograr una separación en el IAF (POS) de 5 ó 10 NM. Esta lista de vuelos estará ordenada por la distancia de los mismos a su TIE Point, al igual que en la ventana principal OPTA-IN, y muestran la misma información:

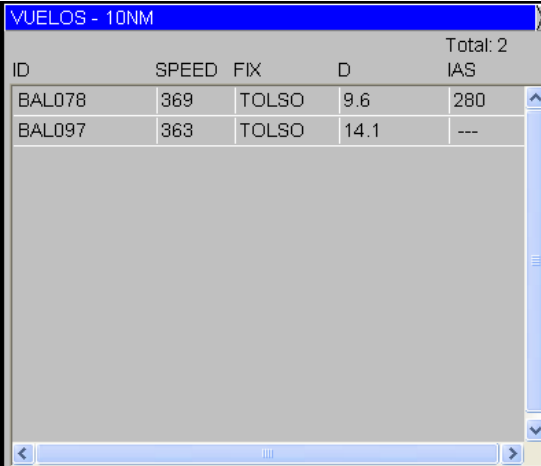
- Identificativo
- Velocidad
- TIE Point
- Distancia al TIE Point
- Ajuste de velocidad requerido



The screenshot shows a window titled "VUELOS - 5NM". It contains a table with the following data:

ID	SPEED	FIX	D	Total: 2 IAS
BAL078	369	TOLSO	9.6	280
BAL097	363	TOLSO	14.1	230

**Ilustración 53: Ventana secundaria de 5NM**



The screenshot shows a window titled "VUELOS - 10NM". It contains a table with the following data:

ID	SPEED	FIX	D	Total: 2 IAS
BAL078	369	TOLSO	9.6	280
BAL097	363	TOLSO	14.1	---

**Ilustración 54: Ventana secundaria de 10NM**

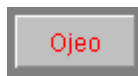
Para cerrar las ventanas de 5NM y 10NM se deben usar los botones de cerrar ubicados en la esquina superior derecha de las ventanas:



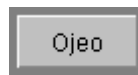
**Ilustración 55: Botón para cerrar las ventanas de 5NM y 10NM**

### 7.2.7. Ojeo

Mediante el botón “Ojeo” se pueden ocultar o visualizar los vuelos excluidos (automática o manualmente) de la secuencia.



**Ilustración 56: Botón “Ojeo” seleccionado**



**Ilustración 57: Botón “Ojeo” sin seleccionar**

Mientras el botón “Ojeo” esté activado, eso es cuando está iluminado (las letras en color rojo), se muestran los vuelos excluidos (tabulares de color rojo) en la secuencia de la ventana OPTA-IN. Por el contrario, cuando el botón no está activado, esos vuelos no se muestran.

Para activar o desactivar el botón “Ojeo”, basta con hacer clic con el botón izquierdo del ratón sobre dicho botón.

### 7.2.8. Excluir e Incluir vuelos

Como ya se ha comentado anteriormente, la aplicación puede excluir automáticamente un vuelo de la secuencia, cuando este no tiene la distancia mínima necesaria, con el vuelo precedente, para poder separarlos ni 5 ni 10 NM. Esos son los vuelos que aparecerán con el tabular en color rojo.

Pero la herramienta, permite además excluir vuelos de forma manual, para que el controlador así lo haga cuando lo crea conveniente.

Los pasos a seguir para excluir un vuelo, son los siguientes:

- Seleccionar el vuelo que se quiere excluir, para ello hay que hacer clic con el botón izquierdo del ratón sobre el tabular del vuelo. En ese momento, el tabular del vuelo seleccionado aparece subrayado.

BAL078 369 TOLSO D009.6

**Ilustración 58: Tabular de vuelo seleccionado**

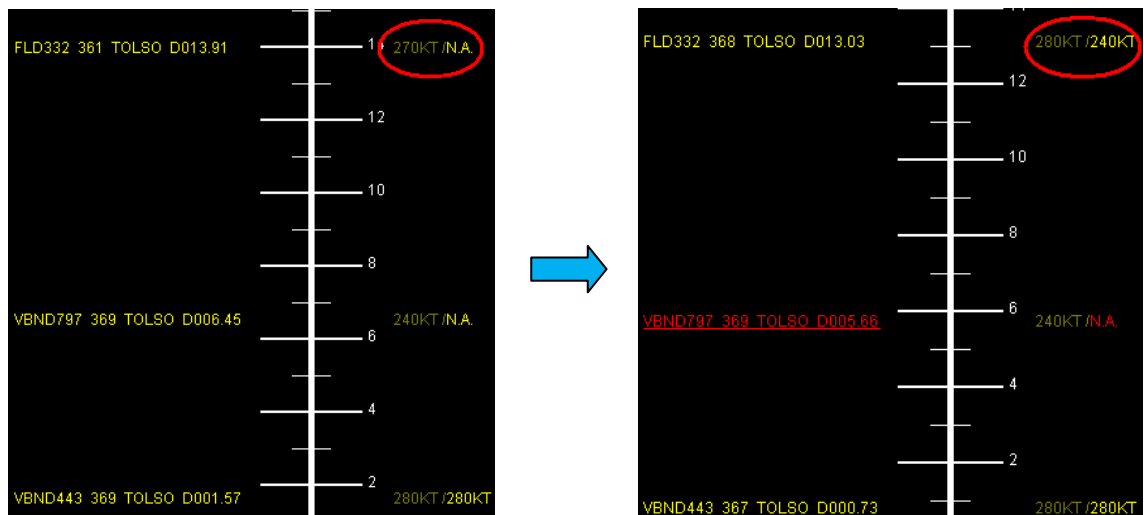
- Una vez esté el vuelo seleccionado, hay que hacer clic sobre el botón “Excl./Incl.”. Es entonces cuando el vuelo pasa a color rojo, indicando que está excluido.



**Ilustración 59: Botón para excluir/incluir vuelos**

NOTA: A hacer esto puede ocurrir que el vuelo desaparece. Eso es porque el botón “Ojeo” no está activado, con lo que no se mostrarán los vuelos excluidos. Activando el ojeo, se podrá comprobar que efectivamente el vuelo está excluido.

Cada vez que se excluye un vuelo de la secuencia, los demás vuelos mostrados en la ventana OPTA-IN, se actualizan, ya que las parejas de vuelos candidatos pueden haber cambiado.



**Ilustración 60: Ejemplo de actualización de las velocidades propuestas**

Del mismo modo que un controlador puede excluir un vuelo de forma manual, la herramienta permite incluir vuelos de forma manual, pero siempre y cuando esos vuelos hayan sido excluidos previamente de forma manual.

El proceso para incluir un vuelo de nuevo en la secuencia es la misma que para excluirlo, pero en este caso el vuelo seleccionado tiene que ser uno excluido (en color rojo). Para ello debe estar activo el botón “Ojeo”, ya que de lo contrario, no se verían estos vuelos en la ventana.



### 7.2.9. Generar Informes

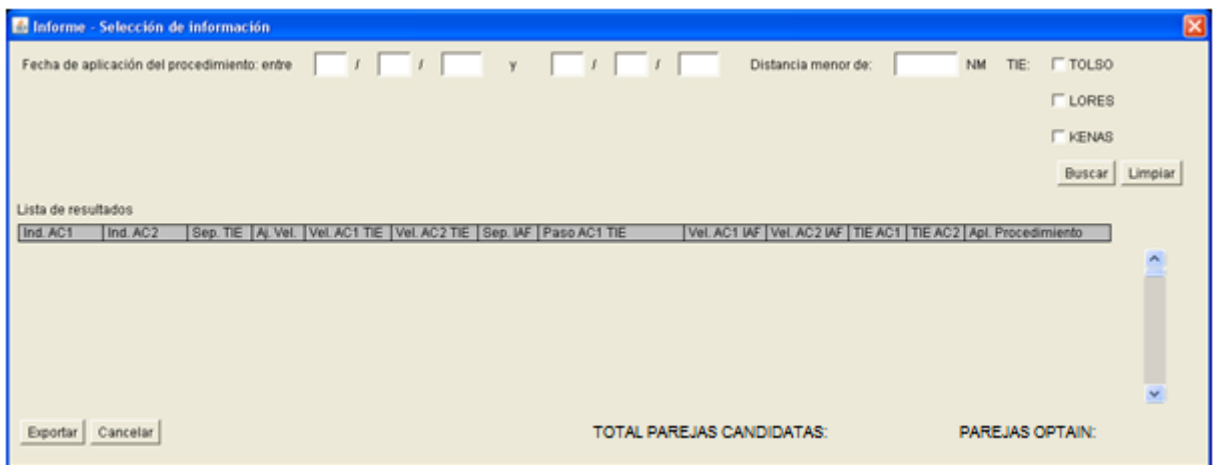
La herramienta permite visualizar los procedimientos OPTA-IN realizados hasta ese momento. También permite exportar esa información a un fichero Excel, que puede ser guardado donde el usuario desee, para su posterior consulta.

Para abrir la ventana informe, hay que pulsar sobre el botón “Informe” con el botón derecho del ratón.



**Ilustración 61: Botón “Informe”**

La apariencia de la ventana “Informe” es la siguiente:



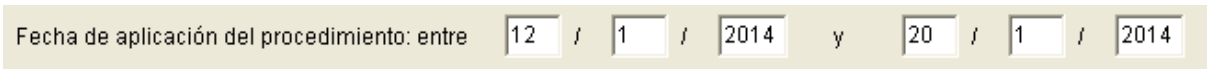
**Ilustración 62: Ventana de Informe**

La información de las parejas de aeronaves (AC1 y AC2) que se guarda en el informe es la siguiente:

- Identificador del AC1 (primera aeronave).
- Identificador del AC2 (segunda aeronave).
- Separación entre las aeronaves en el momento en la que AC1 pasa por su TIE Point.
- Ajuste de la velocidad que se le aplica a AC2 (cuando AC1 pasa por el TIE Point).
- Velocidad de AC1 en el TIE Point.
- Velocidad de AC2 cuando AC1 está en el TIE Point.
- Separación entre las dos aeronaves en el IAF.
- Fecha y hora en la que AC1 pasa por su TIE Point.
- Velocidad de AC1 en el IAF.
- Velocidad de AC2 en el IAF.
- Flujo de entrada de AC1 (TOLSO, LORES o KENAS).
- Flujo de entrada de AC2 (TOLSO, LORES o KENAS).
- Fecha y hora en la que se aplica el procedimiento.

Dentro de la ventana “Informe” se podrán realizar distintas acciones. A continuación vemos el detalle de cada una de ellas.

- *Filtro por fecha:* Si se quiere visualizar los procedimientos realizados en un intervalo de tiempo determinado, hay que seleccionar las fechas de inicio y fin del periodo en los campos correspondiente, y a continuación pulsar el botón “Buscar”, para que se muestren los vuelos en la ventana.



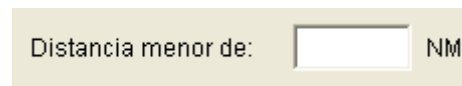
**Ilustración 63: Generar informe con filtro por fecha**

- *Filtro por los puntos de entrada al TMA:* También se puede hacer una búsqueda de procedimientos OPTA-IN aplicados, por los puntos de entrada de los vuelos implicados en los mismos. Los puntos de entrada, como ya se ha explicado con anterioridad, son LORES, TOLSO y KENAS. Si se quiere, por ejemplo, ver las parejas de vuelos, que han aplicado el procedimiento OPTA-IN, en las que al menos uno de ellos haya entrado en el TMA por LORES, hay que seleccionar ese punto, y luego pulsar el botón “Buscar”.



**Ilustración 64: Generar informe con filtro por TIE Point**

- *Filtro por distancia:* La aplicación permite hacer una búsqueda de las parejas que han aplicado el procedimiento OPTA-IN con una separación en el IAF menor a la indicada. Es decir, si se quiere obtener la lista de parejas que han aplicado el procedimiento OPTA-IN, cuya distancia de separación en el IAF es menor a un valor determinado, hay que rellenar el campo de “Distancia menor de” y pulsar el botón “Buscar”.



**Ilustración 65: Generar informe con filtro por distancia en el IAF**

- *Búsqueda general:* Si se quiere sacar el listado general de todos los procedimientos OPTA-IN aplicados hasta el momento, hay que pulsar el botón “Buscar” dejando en blanco todos los campos del filtro.

- *Limpiar la lista de resultados:* Una vez hecha una búsqueda de vuelos que han aplicado el procedimiento OPTA-IN, para borrar la información mostrada en la pantalla, y hacer otra búsqueda posteriormente, basta con pulsar el botón “Limpiar”.

Limpiar

**Ilustración 66: Botón “Limpiar”**

Informe - Selección de información

Fecha de aplicación del procedimiento: entre 12 / 1 / 2014 y 20 / 1 / 2014 Distancia menor de: NM TIE:  TOLSO  LORES  KENAS

Buscar Limpiar

Lista de resultados

Ind. AC1	Ind. AC2	Sep. TIE	Aj. Vel.	Vel. AC1 TIE	Vel. AC2 TIE	Sep. IAF	Paso AC1 TIE	Vel. AC1 IAF	Vel. AC2 IAF	TIE AC1	TIE AC2	Apl. Procedimiento
FRE2	FRE3	6.07	230	368	369	0.06	14/01/2014 13:11:19	368	373	TOLSO	TOLSO	14/01/2014 13:10:24
FRE3	GFD	14.96	250	369	369	3.33	14/01/2014 13:12:41	368	368	TOLSO	TOLSO	14/01/2014 13:10:10
GFD	GFD2	6.28	280	366	369	0.26	14/01/2014 13:15:25	368	368	TOLSO	TOLSO	14/01/2014 13:14:26
GFD6	GFD8	3.71	280	368	369	0.22	14/01/2014 13:24:18	368	369	TOLSO	TOLSO	14/01/2014 13:24:27
GFD8	GFD9	27.70	220	368	368	11.64	14/01/2014 13:27:13	368	366	TOLSO	TOLSO	14/01/2014 13:24:27
L0001	AA	9.15	250	375	369	0.71	14/01/2014 14:42:14	373	368	LORES	TOLSO	14/01/2014 14:41:44
T0002	T0003	8.48	230	365	369	0.85	15/01/2014 10:56:02	367	368	TOLSO	TOLSO	15/01/2014 10:55:37

Exportar Cancelar

TOTAL PAREJAS CANDIDATAS: 648 PAREJAS OPTAIN: 58

**Ilustración 67: Ejemplo de ventana informe con búsqueda con filtro por fecha**

Una vez hecha una búsqueda de vuelos, la herramienta permite generar un fichero Excel con toda esa información. Para ello hay que pulsar sobre el botón “Exportar”.

Exportar

**Ilustración 68: Botón “Exportar”**

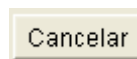
A continuación se abrirá la típica ventana del Windows, donde se podrá seleccionar el nombre del fichero Excel y la ubicación donde se quiere guardar. El fichero generado tendrá la siguiente apariencia:

Ind. AC1	Ind. AC2	Sep. TIE	Aj. Vel	Vel. AC1 TIE	Vel. AC2 TIE	Sep. IAF	Paso AC1 TIE	Vel. AC1 IAF	Vel. AC2 IAF	TIE AC1	TIE AC2	Apl. Procedimiento
HHH	HHH2	14.80	280	367	368	3.53	14/01/2014 10:54	367	368	TOLSO	TOLSO	14/01/2014 10:51
FRE	FRE2	4.16	280	368	369	0.11	14/01/2014 13:10	368	369	TOLSO	TOLSO	14/01/2014 13:10
FRE2	FRE3	6.07	230	368	369	0.06	14/01/2014 13:11	368	373	TOLSO	TOLSO	14/01/2014 13:10
FRE3	GFD	14.96	250	369	369	3.33	14/01/2014 13:12	368	368	TOLSO	TOLSO	14/01/2014 13:10
GFD	GFD2	6.28	280	366	369	0.26	14/01/2014 13:15	368	368	TOLSO	TOLSO	14/01/2014 13:14
GFD6	GFD8	3.71	280	368	369	0.22	14/01/2014 13:24	368	369	TOLSO	TOLSO	14/01/2014 13:24
GFD8	GFD9	27.70	220	368	368	11.64	14/01/2014 13:27	368	366	TOLSO	TOLSO	14/01/2014 13:24
L0001	AA	9.15	250	375	369	0.71	14/01/2014 14:42	373	368	LORES	TOLSO	14/01/2014 14:41
T0002	T0003	8.48	230	365	369	0.85	15/01/2014 10:56	367	368	TOLSO	TOLSO	15/01/2014 10:55
TD0003	L0002	4.10	280	365	373	1.77	15/01/2014 11:04	368	373	TOLSO	LORES	15/01/2014 11:02
PPPP7	PPPP8	20.77	250	367	372	6.76	15/01/2014 15:53	367	367	TOLSO	TOLSO	15/01/2014 15:50
PPPP8	PPPP9	4.70	280	366	369	0.10	15/01/2014 15:56	367	371	TOLSO	TOLSO	15/01/2014 15:54
JKKJK	JKKJK2	7.39	280	365	369	0.76	16/01/2014 14:06	368	368	TOLSO	TOLSO	16/01/2014 14:04
NYMD388	NYMD575	5.45	280	364	369	0.06	22/01/2014 10:01	368	374	TOLSO	TOLSO	22/01/2014 10:00
NYMD575	DDZ001	24.25	240	368	371	9.22	22/01/2014 10:04	368	366	TOLSO	TOLSO	22/01/2014 10:00
DDZ001	DDZ077	5.09	280	364	369	0.33	22/01/2014 10:07	368	369	TOLSO	TOLSO	22/01/2014 10:05
DDZ077	GRN003	4.64	240	364	368	0.04	22/01/2014 10:08	368	371	TOLSO	TOLSO	22/01/2014 10:05
T0002	T0003	5.76	220	368	369	0.07	22/01/2014 14:52	368	372	TOLSO	TOLSO	22/01/2014 14:50
L0001	L0003	8.46	280	375	374	4.39	22/01/2014 14:56	374	373	LORES	LORES	22/01/2014 14:55
L0003	T0004	16.31	280	375	367	4.64	22/01/2014 14:57	373	368	LORES	TOLSO	22/01/2014 14:55

**Ilustración 69: Fichero Excel de informe generado**

Estos ficheros Excel son los que se usarán posteriormente para el estudio de los procedimientos OPTA-IN.

Para cerrar la ventana de informe, se puede hacer el botón “Cancelar” de la ventana, o usar los botones de cerrar de las ventanas de Windows.



**Ilustración 70: Botón “Cancelar” para cerrar la ventana de informe**



**Ilustración 71: Botón para cerrar la ventana de informe**

## 8. ANEXO

### 8.1. Tablas de velocidad

A continuación se recogen las tablas de ajuste de velocidad *EDPO* propuestas por el algoritmo del procedimiento OPTAIN-SA. Dichas tablas fueron calculadas durante el desarrollo del proyecto OPTAIN y relacionan las distancias de una aeronave de la secuencia de llegada a su punto TIE, en el momento de paso de la aeronave anterior en la secuencia a su propio punto TIE, con las indicaciones de ajustes de velocidad propuestas para la aeronave posterior. Éstas han sido calculadas para puntos TIE situados a una distancia de 31NM del IAF.

$d^{\text{offset}}$ [N]	IAS [kts]
3.0 – 3.8	220
3.8 – 4.7	230
4.7 – 5.6	240
5.6 – 6.3	250
6.3 – 7.3	260
7.3 – 7.6	270
> 7.6	280

**Tabla 2: Ajuste de velocidad para separación de 5NM**

$d^{\text{offset}}$ [N]	IAS [kts]
9.4 – 10.6	220
10.6 – 12.0	230
12.0 – 13.3	240
13.3 – 14.2	250
14.2 – 15.2	260
15.2 – 16.0	270
> 16.0	280

**Tabla 3: Ajuste de velocidad para separación de 10 NM**

## 8.2. Integración con el sistema existente

En este apartado se describe como se integran las dos funcionalidades desarrolladas con el sistema existente.

### 8.2.1. OptainDialogoVelocidad

Por un lado se implementa la funcionalidad de selección de velocidad con la clase llamada class OptainDialogoVelocidad que es llamada desde la función *public void draw()* dentro de la clase OptainCuerpoSecuenciador:

```

sec.applet.yClic = -1;
if ((separacion != null) && (pistaAccion != null) &&
(pistaAnteriorIncluida != null) && (pistaAnteriorIncluida.dispiBean != null)) {
    OptainDialogoVelocidad dialogo = new OptainDialogoVelocidad
    (vuelo, pistaAnteriorIncluida.dispiBean.INDICATIVO, separacion, velocidad);

    String velocidadActual;
    dialogo.iniciar();
    try {
        while(!dialogo.getCerrado()) {
            //Thread.sleep(100);
        }

        velocidadActual = dialogo.getVelocidadActualizada();

        if ((velocidadActual != null) && (!velocidadActual.equals(""))) {

            if (separacion.equals("5 NM")) {
                if (pistaAccion.velocidadPropuesta5NM !=
                    Integer.parseInt(velocidadActual)) {
                    pistaAccion.modificadaManual = true;
                }
                pistaAccion.velocidadPropuesta5NM = Integer.parseInt(velocidadActual);
            }
            else {
                if (pistaAccion.velocidadPropuesta10NM !=
                    Integer.parseInt(velocidadActual)) {
                    pistaAccion.modificadaManual = true;
                }
                pistaAccion.velocidadPropuesta10NM = Integer.parseInt(velocidadActual);
            }
            pistaAccion.setEjecutandoOptain(true);
            //System.out.println("pistaAccion " + pistaAccion.dispiBean.INDICATIVO);
            pistaAccion.ejecutaOptain = true;
            pistaAccion.dateTimeProc = sec.applet.toolbar.dateTime;
            pistaAnteriorIncluida.setEjecutandoOptain(true);
            pistaAnteriorIncluida.esParejaOptain = true;
            pistaAnteriorIncluida.indicativoPistaPost = pistaAccion.dispiBean.INDICATIVO;
            // si la pista anterior tiene hora de ejecucion de optain, la mantenemos
            if (pistaAnteriorIncluida.dateTimeProc == null) {
                pistaAnteriorIncluida.dateTimeProc = pistaAccion.dateTimeProc;
            }
        }
    }
    /*else {
        System.out.println("velocidadActual null o vacia");
    }*/

    } catch (Exception excp) {
        System.out.println("ERROR OptainClientApplet draw:
        Exception caught, excp='" + excp + "'");
    }
}

```

### 8.2.2. OptainDialogoInforme

Y por otro lado se implementa la funcionalidad de generación de informes con la clase llamada class OptainDialogoInforme que se llama desde *public void mouseClicked* (*MouseEvent e*) en el siguiente switch dentro de la clase OptainClientApplet:

```
/* check */
switch (action) {
    case OptainToolbar.ACTION_TAB1_ON:
        setTabOver(tab1);
        tab1.setVisible(true);
        break;
    case OptainToolbar.ACTION_TAB2_ON:
        setTabOver(tab2);
        tab2.setVisible(true);
        break;
    case OptainToolbar.ACTION_TAB3_ON:
        ojeo = !ojeo;
        break;
    case OptainToolbar.ACTION_TAB4_ON:
        secl.excluyeIncluyeSeleccionada();
        break;
    case OptainToolbar.ACTION_TAB5_ON:
        OptainDialogoInforme informe = new OptainDialogoInforme(this);
        informe.iniciar();
        break;
    default:
        break;
}
```

### **8.3. Implementación**

#### **8.3.1. Funciones selección de velocidad**

Con el objetivo de conseguir una visión global de cada una de las funcionalidades desarrolladas se incluye una tabla que contiene las funciones que tiene cada clase y una breve descripción cada una de ellas.

La clase *OptainDialogoVelocidad* se compone de las siguientes funciones:

<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>OptainDialogoVelocidad</i>	Constructor
<i>iniciar()</i>	Inicializa panel donde se va a insertar la nueva velocidad
<i>actionPerformed(ActionEvent e)</i>	Maneja los eventos de los botones aceptar y cancelar
<i>getVelocidadActualizada()</i>	Obtiene la nueva velocidad
<i>getCerrado()</i>	Obtiene el valor del boolean cerrado



### 8.3.2. Código fuente selección de velocidad

Se incluye a continuación el código que implementa la funcionalidad completa de selección de velocidad.

```
class OptainDialogoVelocidad implements ActionListener{

    Dialog ventana;
    private String ac1;
    private String ac2;
    private String separacion;
    private String velocidad;
    private String velocidadActualizada;
    private boolean cerrado = false;

    Button aceptar, cancelar;
    NumericTextField nuevaVelocidad = null;

    public OptainDialogoVelocidad(String ac1, String ac2,
    String separacion, String velocidad){

        this.ac1=ac1;
        this.ac2=ac2;
        this.separacion=separacion;
        this.velocidad=velocidad;
        this.velocidadActualizada="";
    }

    /*
     * Funcion que dibuja el Frame con los datos.
     */

    public synchronized void iniciar(){

        Label l1, l2, l3, l4, l5, l6;

        Panel con = new Panel();
        Panel panelVelocidad = new Panel();
        Panel panelBotones = new Panel();

        ventana = new Dialog((Frame) null,
        "Selecci\u00f3n de velocidad", true);

        ventana.setSize(265, 230);
        ventana.setResizable(false); //para que no se pueda maximizar
        ventana.setLayout(new BorderLayout());
        con.setLayout(new FlowLayout(FlowLayout.LEFT));

        l1 = new Label("Se va a informar una velocidad al vuelo");
        l2 = new Label(ac1 + " para una separaci\u00f3n de " +
        separacion + " del");
        l3 = new Label("vuelo " +ac2);
        l4 = new Label("Por favor, indique la nueva velocidad.");

        //Panel horizontal donde esta la velocidad
        panelVelocidad.setLayout(new FlowLayout());
```

```
l5 = new Label("Velocidad: ");

nuevaVelocidad = new NumericTextField(5, 3);
nuevaVelocidad.setText(velocidad);

l6 = new Label("KT");

panelVelocidad.add(l5);
panelVelocidad.add(nuevaVelocidad);
panelVelocidad.add(l6);

//Panel horizontal donde estan los botones.
panelBotones.setLayout(new FlowLayout());
aceptar = new Button("Aceptar");
cancelar = new Button("Cancelar");

aceptar.addActionListener(this);
cancelar.addActionListener(this);

panelBotones.add(aceptar);
panelBotones.add(cancelar);

//Se insertan los elementos a los paneles.
con.add(l1);
con.add(l2);
con.add(l3);
con.add(l4);
con.add(panelVelocidad);
con.add(panelBotones);

//Para que se cierre el frame con la X
ventana.addWindowListener(new WindowAdapter(){
    public void windowClosing(WindowEvent we){
        cerrado = true;
        ventana.dispose();
    }
});

//Se insertan los paneles al frame y se hace visible
ventana.add(con);
ventana.setVisible(true);
}
```

```
    /*
    * Se manejan los eventos de los botones aceptar y cancelar.
    */
//XXX
    public synchronized void actionPerformed(ActionEvent e) {
        if(e.getSource()== aceptar){
            velocidadActualizada = nuevaVelocidad.getText();
            cerrado = true;
            ventana.dispose();
        }
        else{
            if(e.getSource()== cancelar){
                cerrado = true;
                ventana.dispose();
            }
        }
    }

//XXX
    public synchronized String getVelocidadActualizada(){
        return velocidadActualizada;
    }

//XXX
    public synchronized boolean getCerrado(){
        return cerrado;
    }
}
```

### 8.3.3. Funciones generación de informes

Para el desarrollo de la funcionalidad de generación de informes se desarrollaron tres clases diferentes con las siguientes funciones:

*OptainDialogoInforme:*

<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>OptainDialogoInforme(OptainClient applet)</i>	Constructor
<i>Iniciar()</i>	Inicializa el panel principal
<i>panelSuperiorDatos()</i> <i>panelSuperiorCheckbox()</i> <i>panelResultados()</i> <i>panelInferiorBotones()</i>	Inicializan las diferentes partes que formarán el panel principal
<i>actionPerformed(ActionEvent e)</i>	Maneja los eventos de los botones del panel principal
<i>obtenerDatos()</i>	Obtiene los datos insertados en el panel principal
<i>validarFecha(String fecha)</i>	Comprueba que la fecha es correcta

*TablaResultados:* Se ocupa de los datos que se muestran en la tabla.

<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>TablaResultados(CabeceraResultados cabecera)</i>	Constructor
<i>getTamFila()</i>	Obtiene el tamaño de cada fila de la tabla
<i>init(Plist plist, boolean pintar)</i>	Inicializa la tabla
<i>Paint(Graphics g)</i>	Dibuja los datos de la tabla

*CabeceraResultados*: Se ocupa de la cabecera de la tabla que contiene el resultado de la búsqueda.

<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>init()</i>	Inicializa la cabecera de la tabla
<i>Paint(Graphics g)</i>	Dibuja la cabecera de la tabla

Se incluye a continuación el código que implementa la funcionalidad de generación de informes.

```
class OptainDialogoInforme implements ActionListener, AdjustmentListener{

    Dialog ventana;
    Label l1, l2, l3, l4, l5, l6, l7, l8, l9, totalParejas, parejasOptain,
    lSeparador1, lSeparador2, lSeparador3, lSeparador4;
    Button buscar, limpiar, exportar, cancelar;
    NumericTextField distancia = null;
    NumericTextField dia1 = null;
    NumericTextField dia2 = null;
    NumericTextField mes1 = null;
    NumericTextField mes2 = null;
    NumericTextField anio1 = null;
    NumericTextField anio2 = null;

    int numFilas;

    Checkbox checkBox1, checkBox2, checkBox3;
    Panel con, panelDatos, panelSuperior, panelCheckbox, panelBotones1, panelBotones2,
    panelResultados, resto, panelCandidatos;
    String valor1, valor2, valor3, distanciaMenor, fechaIni, fechaFin;
    OptainClientApplet applet;
    CabeceraResultados c;
    TablaResultados panelInforme;
    Panel sp;
    Scrollbar barra_vert;

    public OptainDialogoInforme(OptainClientApplet applet){
        this.applet = applet;
        this.checkBox1 = new Checkbox("TOLSO");
        this.checkBox2 = new Checkbox("LORES");
        this.checkBox3 = new Checkbox("KENAS");
        this.valor1 = "";
        this.valor2 = "";
        this.valor3 = "";
        this.distanciaMenor = "";
        this.fechaIni = "";
        this.fechaFin = "";
    }
}
```

```
public synchronized void iniciar(){
    con = new Panel();

    ventana = new Dialog((Frame) null,
        "Informe - Selecci\u00f3n de informaci\u00f3n", true);
    ventana.setSize(1100, 420);
    ventana.setResizable(false);
    ventana.setLayout(new BorderLayout());

    con.setLayout(new FlowLayout(FlowLayout.LEFT));

    panelSuperiorDatos();
    panelResultados();
    panelInferiorBotones();

    //Se insertan los elementos al panel padre.
    con.add(panelSuperior);
    con.add(panelResultados);
    con.add(panelBotones2);

    //Para que se cierre el dialogo con la X.
    //Hay que ponerlo antes del show/setVisible
    ventana.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent we){
            ventana.dispose();
        }
    });

    //Se inserta el panel padre al dialog y se hace visible.
    ventana.add(con);
    ventana.setVisible(true);
}
```

```
private void panelSuperiorDatos()
{
    panelDatos = new Panel();
    panelSuperior = new Panel();

    //Panel horizontal superior
    panelDatos.setLayout(new FlowLayout());

    l1 = new Label("Fecha de aplicaci\u00f3n del procedimiento: entre ");
    dia1 = new NumericTextField(1, 2);
    lSeparador1 = new Label("/");
    mes1 = new NumericTextField(1, 2);
    lSeparador2 = new Label("/");
    anio1 = new NumericTextField(2, 4);
    l2 = new Label(" y ");
    dia2 = new NumericTextField(1, 2);
    lSeparador3 = new Label("/");
    mes2 = new NumericTextField(1, 2);
    lSeparador4 = new Label("/");
    anio2 = new NumericTextField(2, 4);

    l3 = new Label(" Distancia menor de: ");

    //Textfield para la distancia
    distancia = new NumericTextField(5, 2);

    l4 = new Label("NM");
    l5 = new Label("TIE: ");
    panelDatos.add(lSeparador1);
    panelDatos.add(mes1);
    panelDatos.add(lSeparador2);
    panelDatos.add(anio1);
    panelDatos.add(l2);
    panelDatos.add(dia2);
    panelDatos.add(lSeparador3);
    panelDatos.add(mes2);
    panelDatos.add(lSeparador4);
    panelDatos.add(anio2);
    panelDatos.add(l3);
    panelDatos.add(distancia);
    panelDatos.add(l4);
    panelDatos.add(l5);

    panelSuperiorCheckbox();

    panelSuperior.setLayout(new BorderLayout());
    panelSuperior.add(panelDatos, "West");
    panelSuperior.add(panelCheckbox, "East");
}
```



```
private void panelSuperiorCheckbox(){

    panelCheckbox = new Panel();
    panelBotones1 = new Panel();

    //Panel de los checkboxes
    panelCheckbox.setLayout(new GridLayout(4, 1));

    //Panel de los botones de la zona de los checkboxes
    panelBotones1.setLayout(new FlowLayout());

    buscar = new Button("Buscar");
    limpiar = new Button("Limpiar");

    buscar.setSize(2, 8);

    buscar.addActionListener(this);
    limpiar.addActionListener(this);

    panelBotones1.add(buscar);
    panelBotones1.add(limpiar);

    panelCheckbox.add(checkBox1);
    panelCheckbox.add(checkBox2);
    panelCheckbox.add(checkBox3);
    panelCheckbox.add(panelBotones1);
}

private void panelResultados(){

    resto = new Panel();
    c = new CabeceraResultados(); //es un panel
    barra_vert = new Scrollbar(Scrollbar.VERTICAL, 0, 1, 0, 100);
    panelResultados = new Panel();
    sp = new Panel();

    resto.setLayout(null);
    panelResultados.setLayout(new BorderLayout());
    sp.setLayout(null);

    l6 = new Label("Lista de resultados");

    c.init();
    c.setBounds(2, 0, 1030, 22);

    panelInforme = new TablaResultados(c); // es un panel
    panelInforme.setSize(1030, 10000);

    sp.setBounds(0, 25, 1030, 140);
    sp.add(panelInforme);

    barra_vert.setBounds(1030, 25, 20, 140);
    barra_vert.addAdjustmentListener(this);

    resto.setSize(1100, 170);
    resto.add(c);
    resto.add(sp);
    resto.add(barra_vert);

    panelResultados.add(l6, "North");
    panelResultados.add(resto, "Center");
}
```

```
public void adjustmentValueChanged (AdjustmentEvent e){
    int posicionesScroll = 0;
    int type = e.getAdjustmentType();

    posicionesScroll = barra_vert.getValue();

    switch(type) {
    case AdjustmentEvent.UNIT_INCREMENT:
        barra_vert.setValue(posicionesScroll + barra_vert.getUnitIncrement());
        break;

    case AdjustmentEvent.UNIT_DECREMENT:
        barra_vert.setValue(posicionesScroll - barra_vert.getUnitIncrement());
        break;

    case AdjustmentEvent.BLOCK_INCREMENT:
        barra_vert.setValue(posicionesScroll + barra_vert.getUnitIncrement());
        break;

    case AdjustmentEvent.BLOCK_DECREMENT:
        barra_vert.setValue(posicionesScroll - barra_vert.getUnitIncrement());
        break;
    }

    posicionesScroll = -posicionesScroll;
    panelInforme.setLocation(0, posicionesScroll);
}
```

```
/* Panel con los botones de la zona inferior*/
private void panelInferiorBotones(){
    Label l7, l8, l9;

    panelCandidatos = new Panel();
    panelCandidatos.setLayout(new FlowLayout());

    panelBotones2 = new Panel();
    panelBotones2.setLayout(new FlowLayout());

    exportar = new Button("Exportar");
    cancelar = new Button("Cancelar");
    l9 = new Label("
    " +
    ");

    l7 = new Label("TOTAL PAREJAS CANDIDATAS:");
    l8 = new Label("PAREJAS OPTAIN:");

    //INC3
    parejasOptain = new Label("
    ");
    totalParejas = new Label("
    ");

    exportar.addActionListener(this);
    cancelar.addActionListener(this);

    panelBotones2.add(exportar);
    panelBotones2.add(cancelar);
    panelBotones2.add(l9);

    panelCandidatos.add(l7);
    panelCandidatos.add(totalParejas);
    panelCandidatos.add(l8);
    panelCandidatos.add(parejasOptain);

    panelBotones2.add(panelCandidatos);
}
```

```
/*
 * Se manejan los eventos de los botones .
 */
public void actionPerformed(ActionEvent e) {
    if(e.getSource()== buscar){
        ventana.setCursor(new Cursor(Cursor.WAIT_CURSOR));
        if (!obtenerDatos()) {
            // Error en la consulta o datos invalidos. Limpiamos los resultados
            panelInforme.init(null, true);
            parejasOptain.setText("0");
            totalParejas.setText("0");
            panelInforme.repaint();
        }
        ventana.setCursor(Cursor.getDefaultCursor());
    }
    else {
        if(e.getSource()== limpiar){
            panelInforme.init(null, true);
            parejasOptain.setText("0");
            totalParejas.setText("0");
            panelInforme.repaint();
        }
        else {
            if(e.getSource()== exportar){
                String directorio;
                String fichero;

                FileDialog fd = new FileDialog((Frame) null, "Guardar informe",
                FileDialog.SAVE);
                fd.setFile("*.csv");
                fd.setVisible(true);
                directorio = fd.getDirectory();
                fichero = fd.getFile();
                if (fichero != null) {
                    if (!fichero.endsWith(".csv")) {
                        fichero = fichero + ".csv";
                    }
                }
                try {
                    BufferedWriter br = new BufferedWriter(new FileWriter
                    (directorio + fichero));
                    StringBuffer sb;
                    Plist pAux = panelInforme.plist;
                    InformeOptainBean bean;
                    br.write("sep=");
                    br.newLine();
                    br.write("PARAMETROS");
                    br.newLine();
                    br.write("Fecha Inicio: " + fechaIni + " Fecha Fin: " + fechaFin);
                    br.newLine();
                    br.write("Distancia: " + distanciaMenor);
                    br.newLine();
                    br.write("TIE: " + valor1 + " " + valor2 + " " + valor3);
                    br.newLine();
                    br.newLine();
                    br.write("Total parejas candidatas: " + totalParejas.getText());
                    br.newLine();
                    br.write("Total parejas OPTAIN: " + parejasOptain.getText());
                    br.newLine();
                    br.newLine();
                    br.write("Ind. AC1;Ind. AC2;Sep. TIE;Aj. Vel;Vel. AC1 TIE;Vel. AC2 TIE;Sep.
                    IAF;Paso AC1 TIE;Vel. AC1 IAF;Vel. AC2 IAF;TIE AC1;TIE AC2;Apl. Procedimiento");
                    br.newLine();
                }
            }
        }
    }
}
```

```
while (pAux != null) {
    if ((pAux.current() != null) && (pAux.current() instanceof
InformeOptainBean)) {
        bean = (InformeOptainBean) pAux.current();
        sb = new StringBuffer();
        sb.append(bean.INDICATIVO01); sb.append(";");
        sb.append(bean.INDICATIVO02); sb.append(";");
        sb.append(bean.SEPARACIONTIE); sb.append(";");
        sb.append(bean.VELOCIDAD); sb.append(";");
        sb.append(bean.VELOCIDADTIE1); sb.append(";");
        sb.append(bean.VELOCIDADTIE2); sb.append(";");
        sb.append(bean.SEPRACIONIAF); sb.append(";");
        sb.append(bean.PASO); sb.append(";");
        sb.append(bean.VELOCIDADIAF1); sb.append(";");
        sb.append(bean.VELOCIDADIAF2); sb.append(";");
        sb.append(bean.TIE1); sb.append(";");
        sb.append(bean.TIE2); sb.append(";");
        sb.append(bean.PROCEDIMIENTO);
        br.write(sb.toString());
        br.newLine();
        sb = null;
    }
    pAux = pAux.next();
}
br.close();
} catch (Exception excp) {
    System.out.println("ERROR OptainClientApplet actionPerformed:
Exception caught, excp='" + excp + "'");
}
}
}
}
}
```

```
public boolean obtenerDatos(){
    boolean fechasValidas = false;
    boolean result = false;

    fechaIni = "";
    fechaFin = "";

    distanciaMenor = distancia.getText();

    valor1 = checkBox1.getState() ? checkBox1.getLabel() : "";
    valor2 = checkBox2.getState() ? checkBox2.getLabel() : "";
    valor3 = checkBox3.getState() ? checkBox3.getLabel() : "";

    if (dia1.getText().equals("") && mes1.getText().equals("") &&
        anio1.getText().equals("") && dia2.getText().equals("") &&
        mes2.getText().equals("") && anio2.getText().equals("")) {
        fechasValidas = true;
    }
    else {
        try {
            String d1, m1, d2, m2;

            d1 = dia1.getText().trim();
            m1 = mes1.getText().trim();
            d2 = dia2.getText().trim();
            m2 = mes2.getText().trim();

            if(d1.length() < 2){
                d1 = "0" + d1;
            }

            if(m1.length() < 2){
                m1 = "0" + m1;
            }

            if(d2.length() < 2){
                d2 = "0" + d2;
            }

            if(m2.length() < 2){
                m2 = "0" + m2;
            }

            fechaIni = anio1.getText() + "/" + m1 + "/" + d1;
            fechaFin = anio2.getText() + "/" + m2 + "/" + d2;

            fechasValidas = validarFecha(fechaIni) && validarFecha(fechaFin);
        } catch(Exception excp) { fechasValidas = false; }
    }
}
```

```
if (fechasValidas){

    Object report;
    Plist plist;
    int numFilas = 0;

    while ((report = applet.loadInformeOptain(fechaIni, fechaFin,
        distanciaMenor, valor1, valor2, valor3)) == null){
        System.out.println("WARNING OptainDialogoInforme run:
        Broken pipe with remote Servlet while loadInformeOptain");
        try { Thread.currentThread().sleep(250); } catch(Exception excp) { }
        continue;
    }

    if(report instanceof Plist){
        plist = (Plist) report;
        numFilas = plist.size() - 1;
        // El ultimo registro es el numero de parejas candidatas
        sp.getBounds();

        //Cuantas filas entran por panel
        int FilasPorPanel = sp.getBounds().height/panelInforme.getTamFila();
        if(numFilas>=0 && numFilas>FilasPorPanel){
            barra_vert.setMaximum((numFilas - FilasPorPanel)
                * panelInforme.getTamFila());
        }

        else {
            barra_vert.setMaximum(0);
        }
        barra_vert.setValue(0);
        panelInforme.setLocation(0, 0);
        panelInforme.init(plist, true);
        panelInforme.repaint();
        result = true;

        //Muestra en informe el numero de parejas Optain
        String cadena = "";
        if (numFilas <= 0){
            parejasOptain.setText("0");
            if (numFilas < 0) {
                totalParejas.setText("0");
            }
            else {
                totalParejas.setText((String) plist.get(numFilas));
            }
        }
        else{
            cadena = String.valueOf(numFilas);
            parejasOptain.setText(cadena);
            totalParejas.setText((String) plist.get(numFilas));
        }
    }
}
return result;
}
```

```
public boolean validarFecha(String fecha){  
    SimpleDateFormat formato = new SimpleDateFormat("yyyy/MM/dd");  
    Date testDate = null;  
  
    if(fecha == null){  
        return false;  
    }  
  
    if(fecha.trim().length() != formato.toPattern().length()){  
        return false;  
    }  
  
    formato.setLenient(false);  
  
    try {  
        testDate = formato.parse(fecha.trim());  
    }catch (ParseException e){  
        return false;  
    }  
  
    if (!formato.format(testDate).equals(fecha.trim())) {  
        return false;  
    }  
  
    return true;  
}
```



```
class TablaResultados extends Panel{

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    CabeceraResultados cabecera;
    int y;
    String indicativo1, indicativo2, separacionTie,
    velocidad, velocidadTie1, velocidadTie2, separacionIaf,
    paso, velocidadIaf1, velocidadIaf2, tie1, tie2,
    procedimiento;
    Plist plist;
    InformeOptainBean bean;
    double dSeparacionTie = 0.0;
    double dSeparacionIaf = 0.0;
    boolean actualizar;
        int porcentajeAnt = 0;
    boolean porcentajeIgual = false;

    //Indico el tamaño fijo para cada fila
    public int getTamFila (){
        return 21;
    }

    public TablaResultados(CabeceraResultados cabecera){
        this.cabecera = cabecera;
        actualizar = false;
    }

    public void init(Plist plist, boolean pintar){
        if (this.plist != null) {
            this.plist.release();
        }
        this.plist = plist;
        this.actualizar = pintar;
    }
}
```

```

public void paint(Graphics g) {

    Plist pAux = plist;
    y = 0;

    if(actualizar){
        //Rellenar los datos
        while (pAux != null) {
            if (pAux.current() != null) {
                if (pAux.current() instanceof InformeOptainBean) {
                    bean = (InformeOptainBean) pAux.current();
                    indicativo1 = bean.INDICATIVO1;
                    indicativo2 = bean.INDICATIVO2;
                    separacionTie = bean.SEPARACIONTIE;
                    velocidad = bean.VELOCIDAD;
                    velocidadTie1 = bean.VELOCIDADTIE1;
                    velocidadTie2 = bean.VELOCIDADTIE2;
                    separacionIaf = bean.SEPRACIONIAF;
                    paso = bean.PASO;
                    velocidadIaf1 = bean.VELOCIDADIAF1;
                    velocidadIaf2 = bean.VELOCIDADIAF2;
                    tie1 = bean.TIE1;
                    tie2 = bean.TIE2;
                    procedimiento = bean.PROCEDIMIENTO;

                    int posY = y + 2 + cabecera.ascent;
                    g.drawString(indicativo1, cabecera.xIndicativo1 + 2 , posY);
                    g.drawString(indicativo2, cabecera.xIndicativo2 + 2, posY);
                    g.drawString(separacionTie, cabecera.xSeparacionTie + 2, posY);
                    g.drawString(velocidad, cabecera.xVelocidad + 2, posY);
                    g.drawString(velocidadTie1, cabecera.xVelocidadTie1 + 2, posY);
                    g.drawString(velocidadTie2, cabecera.xVelocidadTie2 + 2, posY);
                    g.drawString(separacionIaf, cabecera.xSeparacionIaf + 2, posY);
                    g.drawString(paso, cabecera.xPaso + 2, posY);
                    g.drawString(velocidadIaf1, cabecera.xVelocidadIaf1 + 2, posY);
                    g.drawString(velocidadIaf2, cabecera.xVelocidadIaf2 + 2, posY);
                    g.drawString(tie1, cabecera.xTie1 + 2, posY);
                    g.drawString(tie2, cabecera.xTie2 + 2, posY);
                    g.drawString(procedimiento, cabecera.xProcedimiento + 2, posY);

                    int posicionY = y + cabecera.textHeight + 3;
                    g.drawLine(cabecera.xIndicativo1 - 3, y + 1, cabecera.xIndicativo1 - 3, posicionY);
                    g.drawLine(cabecera.xIndicativo2 - 3, y + 1, cabecera.xIndicativo2 - 3, posicionY);
                    g.drawLine(cabecera.xSeparacionTie - 3, y + 1, cabecera.xSeparacionTie - 3, posicionY);
                    g.drawLine(cabecera.xVelocidad - 3, y + 1, cabecera.xVelocidad - 3, posicionY);
                    g.drawLine(cabecera.xVelocidadTie1 - 3, y + 1, cabecera.xVelocidadTie1 - 3, posicionY);
                    g.drawLine(cabecera.xVelocidadTie2 - 3, y + 1, cabecera.xVelocidadTie2 - 3, posicionY);
                    g.drawLine(cabecera.xSeparacionIaf - 3, y + 1, cabecera.xSeparacionIaf - 3, posicionY);
                    g.drawLine(cabecera.xPaso - 3, y + 1, cabecera.xPaso - 3, posicionY);
                    g.drawLine(cabecera.xVelocidadIaf1 - 3, y + 1, cabecera.xVelocidadIaf1 - 3, posicionY);
                    g.drawLine(cabecera.xVelocidadIaf2 - 3, y + 1, cabecera.xVelocidadIaf2 - 3, posicionY);
                    g.drawLine(cabecera.xTie1 - 3, y + 1, cabecera.xTie1 - 3, posicionY);
                    g.drawLine(cabecera.xTie2 - 3, y + 1, cabecera.xTie2 - 3, posicionY);
                    g.drawLine(cabecera.xProcedimiento - 3, y + 1, cabecera.xProcedimiento - 3, posicionY);
                    g.drawLine(cabecera.xFin - 3, y + 1, cabecera.xFin - 3, posicionY);
                    g.drawLine(1, y + cabecera.textHeight + 5, cabecera.xFin - 3, y + cabecera.textHeight + 5);
                    y = y + getTamFila();
                }
                pAux = pAux.next();
            }
        }
    }
}

```

```
class CabeceraResultados extends Panel{

    private static final long serialVersionUID = 1L;
    public int      xIndicativo1, xIndicativo2, xSeparacionTie,
xVelocidad, xVelocidadTie1, xVelocidadTie2,
xSeparacionIaf, xPaso, xVelocidadIaf1, xVelocidadIaf2,
xTie1, xTie2, xProcedimiento,
xFin, margen, espacio, textHeight, ascent;
    int      X, Y, y;
    Font tabFont;

    public void init(){
        X = 0;
        Y = 0;
        y = 2;
        margen = 4;
        espacio = 15;
        tabFont = new Font("Arial", Font.PLAIN, 12);

        setLocation(X, Y);
        setFont(tabFont);
        FontMetrics fontMetrics = getFontMetrics(tabFont);
        ascent = fontMetrics.getAscent();
        textHeight = ascent + fontMetrics.getDescent();

        xIndicativo1 = margen;
        xIndicativo2 = xIndicativo1 + fontMetrics.stringWidth("9CARACTER") + 1 + margen;
        xSeparacionTie = xIndicativo2 + fontMetrics.stringWidth("9CARACTER") + 1 + margen;
        xVelocidad = xSeparacionTie + fontMetrics.stringWidth("00.00") + espacio + 1 + (3 * margen);
        xVelocidadTie1 = xVelocidad + fontMetrics.stringWidth("4CAR") + espacio + 1;
        xVelocidadTie2 = xVelocidadTie1 + fontMetrics.stringWidth("4CAR") + espacio + 1 + (4 * margen) + espacio;
        xSeparacionIaf = xVelocidadTie2 + fontMetrics.stringWidth("4CAR") + (2 * espacio) + 1 + (4 * margen);
        xPaso = xSeparacionIaf + fontMetrics.stringWidth("00.00") + espacio + 1 + (2 * margen);
        xVelocidadIaf1 = xPaso + fontMetrics.stringWidth("9DD/MM/AAAA 00:00:00") + 1 + margen;
        xVelocidadIaf2 = xVelocidadIaf1 + fontMetrics.stringWidth("4CAR") + (2 * espacio) + 1 + (3 * margen);
        xTie1 = xVelocidadIaf2 + fontMetrics.stringWidth("4CAR") + (2 * espacio) + 1 + (3 *margen);
        xTie2 = xTie1 + fontMetrics.stringWidth("6CARAC") + 1 + margen;
        xProcedimiento = xTie2 + fontMetrics.stringWidth("6CARAC") + 1 + margen;
        xFin = xProcedimiento + fontMetrics.stringWidth("9DD/MM/AAAA 00:00:00") + 1 + margen;
    }
}
```

```
public void paint(Graphics g){
    g.setColor(Color.lightGray);
    g.fillRect(1, y, xFin - 3, y + textHeight);

    g.setColor(Color.black);
    g.drawLine(1, y, xFin - 3, y); //línea horizontal
    int asc = y + ascent;
    g.drawString("Ind. AC1", xIndicativo1 + 2, asc);
    g.drawString("Ind. AC2", xIndicativo2 + 2, asc);
    g.drawString("Sep. TIE", xSeparacionTie + 2, asc);
    g.drawString("Aj. Vel.", xVelocidad + 2, asc);
    g.drawString("Vel. AC1 TIE", xVelocidadTie1 + 2, asc);
    g.drawString("Vel. AC2 TIE", xVelocidadTie2 + 2, asc);
    g.drawString("Sep. IAF", xSeparacionIaf + 2, asc);
    g.drawString("Paso AC1 TIE", xPaso + 2, asc);
    g.drawString("Vel. AC1 IAF", xVelocidadIaf1 + 2, asc);
    g.drawString("Vel. AC2 IAF", xVelocidadIaf2 + 2, asc);
    g.drawString("TIE AC1", xTie1 + 2, asc);
    g.drawString("TIE AC2", xTie2 + 2, asc);
    g.drawString("Apl. Procedimiento", xProcedimiento + 2, asc);

    int desc = y + textHeight;
    g.drawLine(xIndicativo1 - 3, y + 1, xIndicativo1 - 3, desc);
    g.drawLine(xIndicativo2 - 3, y + 1, xIndicativo2 - 3, desc);
    g.drawLine(xSeparacionTie - 3, y + 1, xSeparacionTie - 3, desc);
    g.drawLine(xVelocidad - 3, y + 1, xVelocidad - 3, desc);
    g.drawLine(xVelocidadTie1 - 3, y + 1, xVelocidadTie1 - 3, desc);
    g.drawLine(xVelocidadTie2 - 3, y + 1, xVelocidadTie2 - 3, desc);
    g.drawLine(xSeparacionIaf - 3, y + 1, xSeparacionIaf - 3, desc);
    g.drawLine(xPaso - 3, y + 1, xPaso - 3, desc);
    g.drawLine(xVelocidadIaf1 - 3, y + 1, xVelocidadIaf1 - 3, desc);
    g.drawLine(xVelocidadIaf2 - 3, y + 1, xVelocidadIaf2 - 3, desc);
    g.drawLine(xTie1 - 3, y + 1, xTie1 - 3, desc);
    g.drawLine(xTie2 - 3, y + 1, xTie2 - 3, desc);
    g.drawLine(xProcedimiento - 3, y + 1, xProcedimiento - 3, desc);
    g.drawLine(xFin - 3, y + 1, xFin - 3, desc);
    g.drawLine(1, y + textHeight, xFin - 3, desc); //línea horizontal
}
}
```

## 9. Bibliografía y referencias

Se cita a continuación la bibliografía más importante y que ha resultado más útil, entre toda la documentación consultada para el desarrollo de este documento.

Toda ella consultada durante toda la realización del proyecto, tanto para la documentación como para el desarrollo de la aplicación.

- *[EDPO]Especificación del procedimiento OPTAIN-SA en el sistema SINA, Indra 2013*
- *[OPDAIW]Optimized Profile Descent Approaches Implementing Windows (OPTAIN), Ineco 2012*
- *[DST]Sina\_DescripcionTecnica, Indra Sistemas 2002*
- *[IACTA]“Introducción al control de tráfico aéreo”, Indra, 2013*
- *<http://www.aena.es/>*