

**UNIVERSIDAD DE OVIEDO**

**ESCUELA POLITÉCNICA DE INGENIERÍA DE  
GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**PROCEDIMIENTOS DE PRUEBAS EN DESARROLLO DE UNA  
FACTORIA DE SOFTWARE**



**RUBÉN VALDÉS SECO  
JULIO 2013**

## Índice de contenido

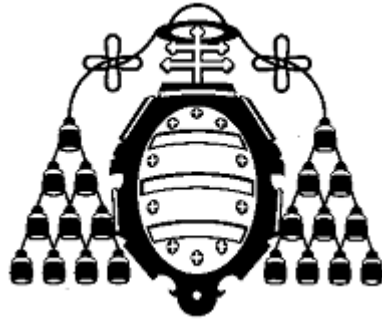
<b>1. MEMORIA</b> .....	<b>4</b>
1.1. DESCRIPCIÓN .....	4
1.2. OBJETIVOS .....	4
1.3. ALCANCE .....	4
1.4. CONCEPTOS .....	5
1.5. ESTRUCTURA DEL TFM .....	6
1.6. METODOLOGÍA DE TRABAJO .....	6
1.6.1. <i>Definir principios fundamentales de la guía de estilo</i> .....	6
1.6.2. <i>Seleccionar fuentes de información</i> .....	7
1.6.3. <i>Definir primeras tareas</i> .....	7
1.6.3.1. Identificar actividades de pruebas ya realizadas .....	7
1.6.3.2. Seleccionar ejemplos y derivar casos de prueba .....	8
1.6.3.3. Complementar información entre casos derivados y ya ejecutados .....	8
1.6.4. <i>Definir siguientes incrementos</i> .....	8
1.6.5. <i>Proceso de revisión y validación</i> .....	9
1.7. SOLUCIONES APORTADAS .....	11
1.8. CONCLUSIONES .....	12
1.9. BIBLIOGRAFÍA .....	12
<b>2. PLANIFICACIÓN Y PRESUPUESTO</b> .....	<b>14</b>
2.1. INTRODUCCIÓN .....	14
2.2. ALCANCE .....	14
2.3. PLANIFICACIÓN .....	14
2.3.1. <i>Planificación inicial</i> .....	14
2.3.2. <i>Seguimiento de la planificación</i> .....	16
2.3.2.1. Primer seguimiento .....	16
2.3.2.2. Segundo seguimiento .....	18
2.3.2.3. Tercer seguimiento .....	20
2.3.2.4. Cuarto seguimiento .....	22
2.3.2.5. Seguimiento final .....	24
2.4. PRESUPUESTO .....	26
<b>3. GUÍA DE PRUEBAS EN DESARROLLO</b> .....	<b>29</b>
3.1. INTRODUCCIÓN .....	29
3.2. ALCANCE .....	29

## Índice de figuras

Figura 1. Representación del esfuerzo de revisión realizado .....	10
Figura 2. Técnicas de prueba seleccionadas para la guía de pruebas en desarrollo de ISL Gijón .....	11
Figura 3. Planificación inicial .....	15
Figura 4. Seguimiento 1 - 01/04/2013 .....	17
Figura 5. Seguimiento 2 - 10/04/2013: nuevas tareas .....	19
Figura 6. Seguimiento 3 - 01/05/2013 .....	21
Figura 7. Seguimiento 3 - 04/06/2013: nuevas tareas a fin de prácticas en empresa .....	23
Figura 8. Seguimiento final - 05/07/2013 .....	25

## Índice de tablas

Tabla 1. Listado de entregables del TFM .....	6
Tabla 2. Descomposición de tareas del TFM a alto nivel .....	9
Tabla 3. Modelo del proceso de revisión adoptado .....	10
Tabla 4. Costes del proyecto (sin IVA) .....	26
Tabla 5. Presupuesto de ejecución por contrata .....	27



**UNIVERSIDAD DE OVIEDO**

**ESCUELA POLITÉCNICA DE INGENIERÍA DE  
GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**PROCEDIMIENTOS DE PRUEBAS EN DESARROLLO DE UNA  
FACTORIA DE SOFTWARE**

**DOCUMENTO N° 1**

**MEMORIA**



**RUBÉN VALDÉS SECO  
JULIO 2013**

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

# 1. MEMORIA

## 1.1.Descripción

La misión principal de una factoría de software consiste en el desarrollo de proyectos software de acuerdo a un conjunto de requisitos proporcionados por un cliente.

Entre las actividades realizadas en una factoría de software se encuentran las *pruebas en desarrollo*, que se refieren a las diseñadas y ejecutadas por el personal con rol de desarrollador, para obtener evidencias de que las funcionalidades implementadas cumplen con la especificación del sistema a construir.

Aunque habitualmente se asocia el concepto de pruebas en desarrollo con *pruebas unitarias*, en general se puede extender el concepto a cualquier tipo de pruebas a bajo nivel (a nivel de componentes individuales o de la integración de varios).

Este Trabajo de Fin de Máster (TFM) se enmarca en la categoría de estudio científico-técnico y en él se tratan algunos aspectos relacionados con las pruebas en desarrollo en la factoría *Indra Software Labs Gijón* (ISL Gijón), entidad en la que se realizó Prácticas en Empresa del Máster en Informática.

En esta factoría no existe en la actualidad un proceso de pruebas formal para este tipo de pruebas, siendo tanto el diseño como la ejecución de pruebas actividades cuyo resultado depende mucho de cada desarrollador particular (por ejemplo, de su formación previa o de la exhaustividad con que esté habituado a realizarlas).

La motivación principal de este TFM es evaluar las actividades que ya se realizan en las pruebas en desarrollo en ISL Gijón para proporcionar una primera guía que recoja un conjunto de prácticas y criterios que permitan dar cierto grado de uniformidad a las pruebas.

El presente documento corresponde a la memoria de este TFM, siendo su principal contenido el siguiente: la descripción de los objetivos y alcance del trabajo, la argumentación de la metodología de trabajo utilizada, las soluciones aportadas para cumplir los objetivos fijados y las conclusiones obtenidas a la finalización.

## 1.2. Objetivos

Este TFM tiene como objetivo principal la generación de una guía técnica compuesta por un conjunto de prácticas y criterios que asistan al personal con perfil de desarrollador en el diseño y ejecución de pruebas en desarrollo de una factoría de software.

La guía debe contribuir en la medida de lo posible a diseñar pruebas bajo una uniformidad de criterios, tanto teóricos (conceptos básicos de pruebas) como prácticos (aplicación de técnicas de prueba) en el tipo de aplicaciones que se desarrolla en la factoría ISL Gijón.

Un objetivo secundario consiste en presentar un conjunto de criterios y recomendaciones a tener en cuenta en la definición de una estrategia de pruebas basada en los riesgos asociados a las entregas al cliente, desde el punto de vista técnico (daño ocasionado ante la existencia de fallos).

## 1.3. Alcance

El alcance fijado es el relativo al proceso de desarrollo actual en la factoría *Indra Software Labs Gijón*, para el cual se personaliza la guía técnica producto de este TFM.

En la actualidad, la mayor parte de la facturación de ISL Gijón corresponde al desarrollo de proyectos relativos a sistemas ATM (*Air Traffic Management*) en cuyo desarrollo participa la compañía *Indra*.

El desarrollo de estos sistemas se distribuye entre diversos *Software Lab* (denominación que reciben las factorías de software de la compañía Indra).

En ISL Gijón, se lleva a cabo el desarrollo de distintos proyectos que afectan a un conjunto de subsistemas específicos de esos sistemas ATM: *Plan de Vuelo*, *Vigilancia* y *Posición*. Los nombres de estos subsistemas dan nombre a las áreas principales de desarrollo de ISL Gijón.

Para realizar el trabajo se toma como referencia el proyecto EPPI (*Estudio Preliminar del Proceso de Pruebas en Indra Software Labs*), ejecutado en el ejercicio 2012 en el marco de la Cátedra Indra de colaboración con la Universidad de Oviedo [EPPI].

Una vez identificadas las tareas de pruebas que ya se realizaban en el proceso de desarrollo actual, se fijaron los aspectos concretos que debería cubrir la guía, para particularizarla y adaptarla a las necesidades específicas de ISL Gijón.

En concreto, se priorizaron los siguientes aspectos:

- La guía debía cubrir las pruebas relativas a los desarrollos realizados en las 3 áreas principales de la factoría: Plan de Vuelo (PV), Vigilancia (VIG) y Posición (POS).
- Introducir al desarrollador en conceptos relativos a técnicas de pruebas y su aplicación práctica.
- Definir los criterios comunes necesarios para definir una estrategia de pruebas basada en riesgos.
- El carácter general de la guía debía ser eminentemente práctico, es decir, las explicaciones debían estar dirigidas por ejemplos.

Para cumplir el último aspecto, los responsables de la factoría seleccionaron 2 proyectos reales aún en fase de desarrollo:

- *SESAR 12.03.02 DISEÑO FUNCIONAL DE ALERTAS “CONFORMANCE MONITORING (CMON)” y “CONFLICTING ATC CLEARANCES (ATC)”*, relativo a la detección de alertas en la superficie de un aeródromo y su presentación gráfica al controlador de torre.
- *SESAR 12.05.02* cuyo objetivo es la presentación de las alertas anteriores en HMI (*Human Machine Interface*).

## 1.4. Conceptos

En este apartado se recogen los términos básicos cuya explicación es necesaria para comprender el dominio del proyecto:

- **Pruebas en desarrollo:** dada una funcionalidad implementada por un desarrollador, se refiere a las pruebas que este realiza para verificar que el código construido cumple la especificación. Se trata por tanto de pruebas que involucran uno o varios componentes de un sistema.
- **Objetivo de prueba:** se refiere a lo que va ser probado, como por ejemplo un requisito, un componente, una funcionalidad específica de un componente o un determinado riesgo de un producto.
- **Situación de prueba:** se refiere a lo que debería ser probado respecto a un objetivo de prueba, como por ejemplo un valor concreto, un rango de valores, una combinación de valores o una transición de un estado a otro.
- **Caso de prueba:** es la forma de ejercitar situaciones de prueba y tiene la siguiente estructura:
  - *Situación inicial:* es la situación de partida en la que tiene que estar el sistema y su entorno de ejecución.
  - Uno o varios *pasos de prueba:* cada uno tiene un conjunto de valores de entrada y una descripción de las acciones que es necesario realizar para procesarlos. Los pasos de prueba se ejecutan secuencialmente y cada uno tiene una salida esperada que podría ser la entrada del siguiente paso de prueba.

Cada caso de prueba puede ejercitar una o varias situaciones de prueba o varias combinaciones de situaciones de prueba.

- **Test basis:** cualquier tipo de información que se considere relevante para el diseño de pruebas, como por ejemplo especificaciones de requisitos, documentos de diseño funcional, documentos que describen la arquitectura del sistema a probar o manuales de usuario.
- **Modelo:** una forma de representar la especificación que facilita la derivación de situaciones de prueba, como por ejemplo diagramas de transición de estados, tablas de decisión o condiciones complejas.
- **Técnica de diseño de pruebas:** es un procedimiento estándar para identificar situaciones de prueba de forma sistemática.

- **Riesgo:** probabilidad de que ocurra un fallo en relación al daño esperado si este se produce. Se puede definir mediante la fórmula

$$\text{Riesgo} = \text{Probabilidad de fallo} \times \text{Daño}$$

donde:

- Probabilidad de fallo = Probabilidad de presencia de defectos x Frecuencia de uso
- Daño: un daño directo podría involucrar pérdidas de ingresos, daños a terceros, daños físicos o en el medio ambiente (en especial cuando se trata de sistemas empotrados) o sobrecostes por correcciones.

Un daño indirecto podría involucrar daño en la imagen corporativa o pérdida de confianza de los clientes entre otros.

- **Intensidad de pruebas:** nivel de detalle con el que se decide probar un determinado objetivo de prueba, atendiendo al nivel de riesgo identificado para el mismo (normalmente a mayor riesgo mayor intensidad de pruebas). Involucra la aplicación de una o varias técnicas de prueba.
- **Estrategia de pruebas basada en riesgos:** documento que recoge el desglose de tareas de desarrollo para un proyecto, donde cada tarea tiene asociado una determinada intensidad de pruebas determinada en función de los riesgos identificados para ella. Es la base para la planificación de las tareas, priorizándolas de acuerdo a su riesgo asociado (las de mayor riesgo primero).
- **Regresión:** se experimenta cuando la calidad de un sistema se deteriora como consecuencia de haber introducido cambios.
- **Pruebas de regresión:** su objetivo es verificar que tras la implementación de un cambio el resto de partes del sistema que no han sido modificadas funcionan correctamente.

## 1.5. Estructura del TFM

Este TFM está compuesto por los siguientes entregables:

Entregable	Descripción
Documento 1	Memoria
Documento 2	Planificación y presupuesto
Documento 3	Guía de pruebas en desarrollo en Indra Software Labs Gijón

Tabla 1. Listado de entregables del TFM

## 1.6. Metodología de trabajo

En el momento de definir la estructura de tareas de este TFM existía cierto grado de incertidumbre respecto a los aspectos a los que se debía dar más prioridad.

Debido a esto se decidió utilizar el siguiente enfoque incremental:

- Un primer incremento, cuyo objetivo principal era obtener resultados rápidos que permitiesen evaluar aquellos aspectos a los que se debía dar más prioridad y sobre los que se debía centrar el trabajo.
- Los posteriores incrementos, definidos en base al análisis de los resultados del primer incremento.

En las siguientes secciones se detallan las tareas realizadas en los distintos incrementos.

### 1.6.1. Definir principios fundamentales de la guía de estilo

El objetivo de esta tarea es definir un conjunto de criterios que describan el propósito general de la guía de pruebas en desarrollo:

1. El lenguaje utilizado debe ser el adecuado atendiendo a la audiencia principal, en este caso personas con perfil de desarrollador de software.
2. Todos los conceptos presentados y desarrollados deben estar sustentados por ejemplos.
3. Los ejemplos incluidos en la guía deben tener un nivel de detalle adecuado para que la audiencia comprenda adecuadamente el esfuerzo que requiere la aplicación de los conceptos presentados.
4. En la medida de lo posible se deben presentar ejemplos reales extraídos de las fuentes de información utilizadas para realizar el trabajo. En aquellos casos donde no sea posible utilizar un ejemplo real, se presentarán ejemplos sintéticos.

5. La guía debe ser auto-contenida, es decir, todos los conceptos y definiciones presentados deben estar incluidos en ella sin necesidad de consultar fuentes adicionales.

### 1.6.2. Seleccionar fuentes de información

El criterio para seleccionar las principales fuentes de información para realizar el análisis inicial se basó en establecer un conjunto de requisitos deseables:

- Documentación
  - Seleccionar un proyecto real cuyo ámbito abarcara las 3 áreas principales de ISL Gijón: Plan de Vuelo, Vigilancia y Posición.
  - El proyecto debía estar aún en fase de desarrollo en el momento de comenzar el TFM, ya que de esta forma sería posible establecer una comparación entre el diseño de pruebas asistido por técnicas y el diseño manual realizado en el proyecto hasta el momento, para obtener un feedback relevante.
- Entrevistas
  - Desarrolladores del proyecto seleccionado como referencia.
  - Stakeholders con visión más global del proceso de desarrollo, con perfil de gestor de proyecto.

Teniendo en cuenta todos los requisitos anteriores se seleccionó un conjunto de proyectos que cumplía todos los requisitos, en concreto el proyecto con la denominación *SESAR 12.03.02 DISEÑO FUNCIONAL DE ALERTAS “CONFORMANCE MONITORING (CMON)”* y *“CONFLICTING ATC CLEARANCES (ATC)”*, junto con el proyecto asociado *SESAR 12.05.02* [SESAR12.03.02-12.05.02].

El proyecto SESAR 12.03.02 consiste básicamente en la implementación de requisitos relativos a la *detección* de alertas de tipo CMON y ATC, que afectan a los móviles que circulan por la superficie de un aeródromo, cubriendo las áreas de Plan de Vuelo y Vigilancia.

El proyecto SESAR 12.05.02 especifica los requisitos de *presentación* de las alertas de tipo CMON y ATC en el interfaz gráfico que usan los controladores de torre, cubriendo el área de Posición.

### 1.6.3. Definir primeras tareas

El propósito de las primeras tareas es determinar lo antes posible el conjunto de tareas necesario para alcanzar los objetivos fijados, priorizando aquellos aspectos que permitan aportar valor a las pruebas en desarrollo de ISL Gijón.

En los siguientes apartados se detallan estas primeras tareas, que sirven como inmersión en el proceso de desarrollo actual de ISL Gijón y en el conocimiento funcional mínimo necesario sobre los proyectos de referencia para evaluar la viabilidad de aplicar conceptos teóricos de diseño de pruebas.

Los stakeholders consideraron importante realizar una labor de análisis independiente para los desarrollos en el área de Posición, debido a las características particulares de los desarrollos en esa área. Por este motivo las siguientes tareas se realizaron de forma independiente tanto para el proyecto SESAR 12.03.02 (PV y VIG) como para SESAR 12.05.02 (POS).

#### 1.6.3.1. Identificar actividades de pruebas ya realizadas

Se utilizaron 2 fuentes de información para realizar esta tarea:

1. el proyecto EPPI [EPPI].
2. el feedback obtenido por medio de entrevistas con los desarrolladores involucrados en los proyectos SESAR 12.03.02 y SESAR 12.05.02.

Uno de los objetivos del proyecto EPPI era obtener una instantánea de las actividades relacionadas con pruebas que un desarrollador realizaría normalmente en las áreas de Plan de Vuelo, Vigilancia y Posición.

Dicha instantánea tenía como finalidad servir para establecer una comparación con las actividades de pruebas que se realizan en otros procesos de pruebas estructurados más formales utilizados como referencia, TMAP en este caso concreto [TMAP][TPI].

El proceso de desarrollo en ISL Gijón ha sufrido algunos cambios a nivel estructural desde la finalización del proyecto EPPI, de modo que para validar esta fuente de información se cotejaron las conclusiones extraídas del proyecto EPPI con la información recopilada por medio de entrevistas con un desarrollador involucrado en el proyecto SESAR 12.03.02 por un lado y por otro con un desarrollador del proyecto SESAR 12.05.02.

Se pudo concluir que, en general, las actividades (procedurales) realizadas eran prácticamente las mismas en ambos casos. En particular se observó que el número y calidad de las pruebas diseñadas dependía bastante de cada desarrollador particular (formación previa, conocimiento funcional y/o experiencia acumulada).

Del mismo modo que en la factoría ya existe una guía de estilo de desarrollo cuya función es que el software construido se atenga a un conjunto de prácticas (estructura del código, nombrado,...), se decidió, en conjunto con los stakeholders, que elaborar un documento similar que sirviese como referencia de cara al diseño de pruebas podría aportar valor, dando uniformidad a las pruebas.

El medio más adecuado para conseguir este fin era presentar al desarrollador un conjunto de técnicas de prueba adecuado para el tipo de desarrollos que realizan y que ayudasen a diseñar pruebas de forma sistemática.

#### **1.6.3.2. Seleccionar ejemplos y derivar casos de prueba**

Una vez identificada la línea principal de trabajo, las siguientes tareas estarían enfocadas a refinarla, identificando aspectos relevantes con vistas a particularizar la guía técnica para el tipo de desarrollos que se realizan en ISL Gijón.

Para ello se decidió extraer un conjunto pequeño pero representativo de ejemplos a partir de la especificación de los proyectos de referencia [SESAR12.03.02-12.05.02] y aplicarles un conjunto de técnicas para derivar situaciones de prueba y casos de prueba.

El propósito de hacer esto era obtener de forma rápida una primera selección de técnicas de prueba candidatas a ser incluidas en la guía técnica.

#### **1.6.3.3. Complementar información entre casos derivados y ya ejecutados**

Dado que los proyectos de referencia estaban aún en fase de desarrollo en el momento de comenzar el TFM, era posible comparar algunas de las pruebas que los desarrolladores habían ejecutado con las primeras pruebas diseñadas candidatas a incluir como ejemplo en la guía.

El objetivo de establecer esta comparación era doble:

- por un lado identificar si había aspectos concretos que los desarrolladores cubrían con sus pruebas y que no se podrían reflejar en las pruebas diseñadas para la guía, de cara a esa labor de personalización identificada como necesaria.
- por otro lado que el desarrollador realizase una primera validación de las técnicas de prueba utilizadas, evaluando si:
  - la ejecución de los casos de prueba diseñados presentado en la guía era factible.
  - podría utilizar las técnicas propuestas en su trabajo diario.

Esta tarea no se habría podido realizar si el proyecto de referencia seleccionado estuviese en fase de mantenimiento, ya que el tipo de pruebas realizado en ese caso tienen un enfoque algo distinto al de las pruebas en desarrollo.

#### **1.6.4. Definir siguientes incrementos**

En base al feedback obtenido con las primeras tareas se definió la estructura general que debería tener la guía de pruebas en desarrollo.

Dicha estructura sería la siguiente:

- Contenido orientado al desarrollador:
  - Conceptos generales de diseño de pruebas
  - Técnicas de prueba



- Nivel 1: introducir al desarrollador en conceptos generales de diseño de pruebas, presentando un conjunto de técnicas básicas mediante ejemplos sencillos.
    - Nivel 2: desarrollar las técnicas anteriores sobre ejemplos más completos, presentando algunas particularidades y/o excepciones que pueden suceder al aplicar determinadas técnicas.
    - Nivel 3: demostrar mediante ejemplos cómo se pueden combinar las técnicas presentadas para diseñar pruebas más completas.
- Contenido orientado al responsable/gestor de proyecto
  - Estrategia de pruebas
    - Identificar riesgos asociados a las tareas de desarrollo.
    - Clasificar riesgos.
    - Asignar intensidad de pruebas en función del riesgo.
    - Proponer una selección de técnicas de prueba a utilizar en función de la intensidad fijada.

La estructura anterior se utilizó como referencia para determinar los incrementos 2 y 3:

Incremento	Tareas	Resultados
1	Seleccionar un pequeño conjunto de ejemplos de los proyectos SESAR 12.03.02 y SESAR 12.05.02 para centrar resto del trabajo a realizar.	Empezar a cubrir los niveles 1 y 2 de técnicas de prueba.
2	Extender y completar el primer conjunto de ejemplos seleccionado para el proyecto SESAR 12.03.02 (PV y VIG).	Cubrir niveles 1, 2 y 3 de técnicas.
	Proseguir estudio del proyecto SESAR 12.05.02 (POS).	Completar niveles 1, 2 y 3 con detalles específicos de pruebas en POS.
	Proponer criterios para la definición de una estrategia basada en riesgos.	Cubrir la estrategia de pruebas.
3	Si los resultados del segundo incremento no fuesen suficientemente representativos de todo el proceso de desarrollo, extender el análisis a otros proyectos.	Seguir completando niveles 1, 2 y 3 en caso de ser necesario.

**Tabla 2. Descomposición de tareas del TFM a alto nivel**

El detalle de la descomposición de tareas de cada incremento se puede consultar en el documento 2: *Planificación y presupuesto*.

### 1.6.5. Proceso de revisión y validación

Una vez definida la descomposición de tareas necesaria para obtener el producto final de este estudio, se fijó un conjunto de requisitos de validación de los resultados obtenidos en cada incremento:

- *Compresibilidad*: comprobar que los conceptos presentados son entendibles por las distintas audiencias (desarrolladores y responsables/gerentes de proyecto).
- *Veracidad*: que no existan afirmaciones o asunciones erróneas en los ejemplos desarrollados a partir de especificaciones reales extraídas de los proyectos de referencia.
- *Aplicación práctica*: que las prácticas y criterios presentados en la guía técnica sean suficientemente abordables como para ser utilizados en el trabajo diario.

Para este propósito se constituyó un equipo con la siguiente organización:

- Equipo revisor de ISL Gijón
  - 2 gerentes de proyecto
  - 1 desarrollador
- Tutor académico

El proceso de revisión seguido fue el siguiente:

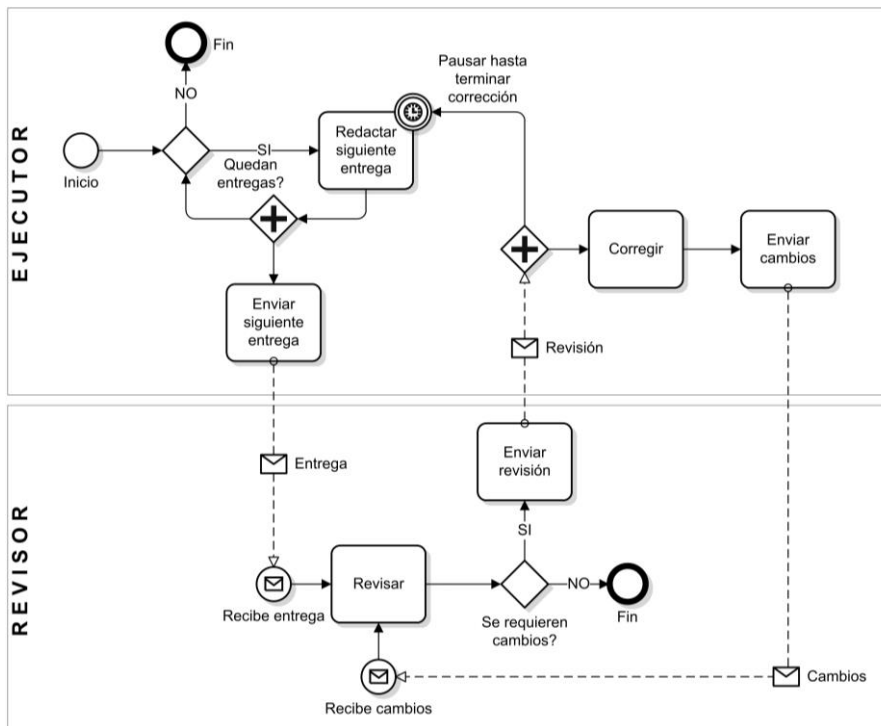


Tabla 3. Modelo del proceso de revisión adoptado

Al término del incremento 2 se verificó que las conclusiones extraídas eran suficientemente representativas del proceso de pruebas en desarrollo como para aportar una primera guía de pruebas, de modo que se consideró que no era necesario analizar otros proyectos.(es decir, se descartó proseguir con el tercer incremento).

En la siguiente gráfica se representa la relación de esfuerzo de revisión a lo largo las distintas entregas intermedias:

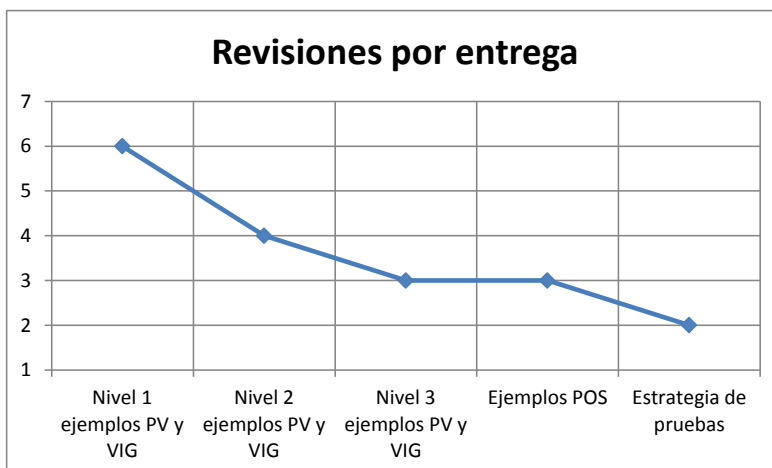


Figura 1. Representación del esfuerzo de revisión realizado

Se puede observar que se realizó un mayor esfuerzo en las primeras tareas, debido principalmente a la dificultad de encontrar el lenguaje adecuado para presentar los conceptos desarrollados a personas sin formación específica en pruebas y la corrección de errores de interpretación de conceptos de negocio.

## 1.7. Soluciones aportadas

La principal aportación consiste en la elaboración de una guía técnica en la que se recoge un conjunto de prácticas y criterios que permitan dotar de uniformidad a las pruebas en desarrollo en *Indra Software Labs Gijón*, orientado a dos tipos de audiencias distintas:

- *Desarrollador*: guiarle en la aplicación de técnicas de diseño de pruebas
- *Responsable/gerente de proyecto*: proporcionarle un marco de referencia para la definición de estrategias de pruebas basadas en los riesgos de las entregas al cliente

Queda fuera del alcance de este TFM incluir mediciones que permitan justificar que, como resultado de la aplicación de las prácticas y criterios incluidos en la guía de pruebas propuesta, se produzca una mejora real en la rentabilidad respecto a las pruebas realizadas en el proceso de desarrollo actual de ISL Gijón.

Sin embargo, existen diversos trabajos que respaldan, en base a la experimentación, que:

- diseñar pruebas en base a riesgos permite obtener un feedback valioso lo antes posible de cara a mantener actualizada la planificación.
- dotar de estructura al proceso de pruebas constituye una base necesaria para que sea posible recolectar métricas representativas de progreso y grado de calidad cubierto, facilitando el seguimiento y control del proceso de pruebas.
- la utilización y combinación de técnicas de prueba, aplicadas adecuadamente, permiten en general aumentar la probabilidad de detectar defectos importantes y que en algunos casos resultan difíciles de detectar si no se utilizan técnicas.

Los primeros 2 factores citados se tratan con menor grado de detalle en la guía de estilo elaborada, sin embargo el tercero se trata en mayor profundidad, dado que se trataba del aspecto más valorado por los stakeholders de ISL Gijón que participaron en la definición de objetivos de este trabajo.

En el contexto de todas las técnicas (formales) de prueba conocidas, el conjunto de técnicas seleccionado es el siguiente (se han resaltado en color verde):

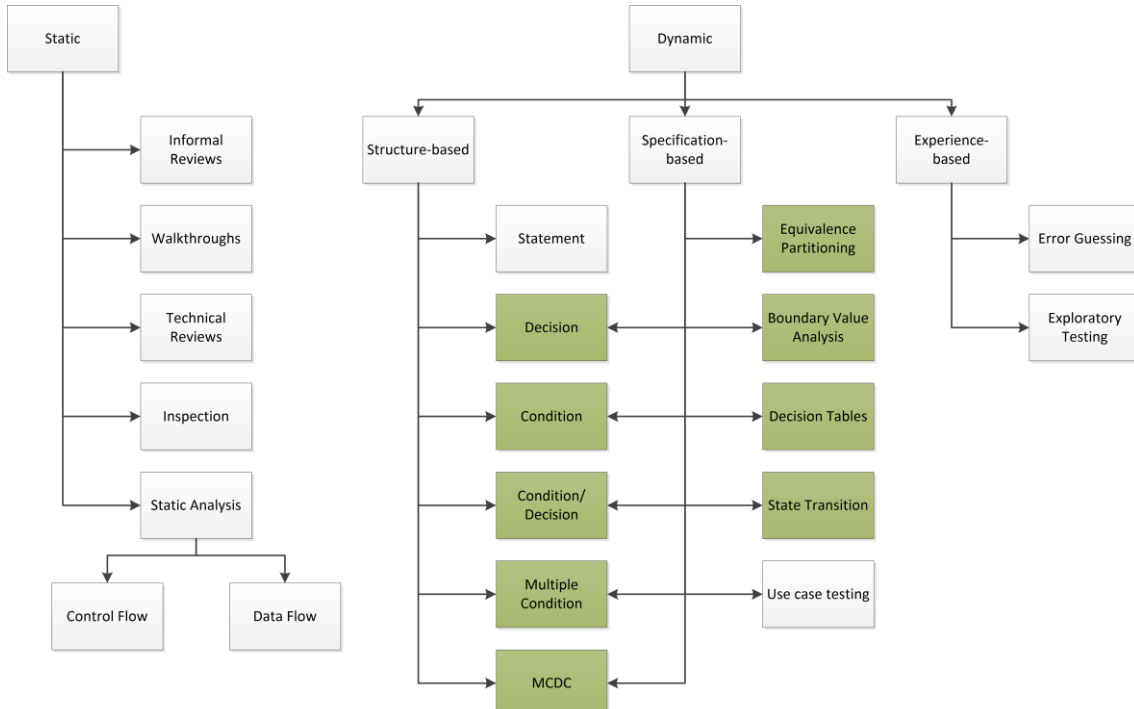


Figura 2. Técnicas de prueba seleccionadas para la guía de pruebas en desarrollo de ISL Gijón

Se considera que esta selección de técnicas de prueba es suficiente para asistir al desarrollador en el diseño de pruebas en desarrollo que cubran aspectos funcionales.

Cubrir otros tipos de pruebas cuyo objetivo sea ejercitar aspectos no funcionales (como por ejemplo pruebas de rendimiento o de robustez) podría ser abordado en futuros TFM que continuasen con este trabajo.

## 1.8. Conclusiones

Las conclusiones extraídas de este trabajo se pueden clasificar en dos grupos: por una parte las *lecciones aprendidas* como resultado del propio trabajo y por otra las *aportaciones personales* que ha supuesto la realización del trabajo.

Las principales lecciones aprendidas son:

- Conocimiento de los detalles de un proceso de desarrollo de una factoría de software (estructura, organización, ciclo de vida y seguimiento/control del progreso).
- Redactar un documento técnico que introduce muchos conceptos nuevos para una audiencia específica exige estructurar muy bien el documento y redactarlo poniéndose constantemente en el lugar de la audiencia que va leerlo.
- Realizar pruebas en desarrollo en una factoría de software es una tarea exigente:
  - Es habitual que el desarrollador se vea sometido a cierta incertidumbre:
    - Para realizar su trabajo tiene como entrada especificaciones no siempre completas y estables.
    - A veces la especificación puede describir funcionalidades de forma ambigua, lo que exige verificar la información con terceros o con el cliente.
  - Aunque el desarrollador posea mucho conocimiento funcional del sistema bajo prueba, muchos aspectos no obvios podrían quedar sin probar si no se identifican de forma sistemática.
- Las técnicas de prueba suponen el medio más adecuado para identificar de forma sistemática lo que se debe probar de un sistema, aunque exigen aprendizaje y cambiar la mentalidad “constructiva” que habitualmente tiene el desarrollador.
- La estrategia de pruebas debe reflejar los 2 principios sobre los que se fundamentará el trabajo de pruebas de cada desarrollador: “probar primero lo más crítico” y “probar con mayor intensidad lo más crítico”.

En cuanto a las aportaciones personales, destacar principalmente que el trabajo ha servido para:

- Refinar las capacidades de auto-organización y planificación del trabajo.
- Mejorar la auto-disciplina, en el sentido de llevar un seguimiento del trabajo propio.
- Poner en práctica la formación académica recibida, tomando conciencia de la dificultad que siempre supone hacer un ejercicio de abstracción y posterior aplicación de conceptos a un caso real.

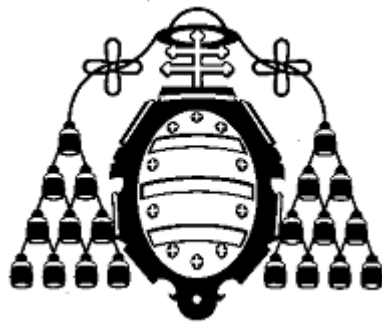
## 1.9. Bibliografía

[EPPI] Tuya, J, de la Riva, C., Valdés, R., *Estudio preliminar del proceso de pruebas en Indra Software Labs Gijón*, Edición 1.0, Julio 2012

[SESAR12.03.02-12.05.02] *Diseño funcional de alertas “Conformance Monitoring (cmon)” y “Conflicting ATC Clearances”*, SESAR 12.03.02 - Enhanced Surface Safety Nets, Prototipo Fase 2

[TMAP] Koomen, T., van der Aalst, L., Broekman, B., Vroon M., *TMap Next for result-driven testing*, 2006

[TPI] Koomen, T., Pol, M., *Test Process Improvement: A Step-by-step Guide to Structured Testing*, 1999



**UNIVERSIDAD DE OVIEDO**

**ESCUELA POLITÉCNICA DE INGENIERÍA DE  
GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**PROCEDIMIENTOS DE PRUEBAS EN DESARROLLO DE UNA  
FACTORIA DE SOFTWARE**

**DOCUMENTO Nº 2**

**PLANIFICACIÓN Y PRESUPUESTO**



**RUBÉN VALDÉS SECO  
JULIO 2013**

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

## 2. PLANIFICACIÓN Y PRESUPUESTO

### 2.1. Introducción

En este documento se presenta la estructura de tareas, planificación temporal y seguimiento del trabajo realizado para redactar la guía técnica de pruebas en desarrollo de ISL Gijón, así como el presupuesto asociado.

### 2.2. Alcance

El presente TFM está vinculado a unas prácticas en empresa, de manera que el alcance fijado para la planificación incluye las tareas realizadas y tuteladas en la empresa, así como las que se han realizado una vez terminada la estancia en la empresa y que forman parte del trabajo individual del proyectante.

### 2.3. Planificación

Al comienzo del proyecto, existía cierto grado de incertidumbre que dificultaba definir con precisión todas las tareas que serían necesarias para satisfacer los objetivos fijados.

Debido a esto, se decidió planificar el trabajo de forma incremental:

- Un primer incremento, compuesto por una serie de tareas destinadas a seleccionar un pequeño conjunto de ejemplos de los proyectos SESAR 12.03.02 y SESAR 12.05.02 para centrar resto del trabajo a realizar.
- Un segundo incremento, cuya estructura se definiría a partir del feedback obtenido del primer incremento y que debería reflejar el resto del trabajo necesario.
- Un tercer incremento, que se realizaría únicamente en caso de observar que las conclusiones extraídas a partir de los proyectos SESAR 12.03.02 y SESAR 12.05.02 no son suficientemente representativas del tipo de desarrollos que se llevan a cabo en ISL Gijón. En ese caso el contenido de la guía técnica se completaría analizando otros proyectos.

Para llevar el control y el seguimiento del trabajo se utilizaron los siguientes criterios:

- Para cada tarea, el progreso del trabajo realizado se imputa en horas empleadas (trabajo real) y se actualiza el número de horas pendientes para finalizarla (trabajo restante).
- Para comparar el progreso con el trabajo programado:
  - Se estableció una nueva línea base con periodicidad mensual.
  - Se estableció una nueva línea base cada vez que se aprobó la inserción o eliminación de tareas en la EDT.

#### 2.3.1. Planificación inicial

La planificación inicial tiene fecha de comienzo el 1 de Marzo de 2013 y se estima su finalización a fecha 5 de Abril de 2013, lo que supone una duración aproximada de 1 mes (21 días laborables).

El periodo de tiempo comprendido entre el 1 de Marzo y el 4 de Junio de 2013 corresponderá al trabajo realizado en ISL Gijón como parte de las prácticas en empresa.

En este intervalo se dispone de un único recurso humano, *Becario*, para el que se ha establecido un horario laboral en el que se trabaja 5 días a la semana, un total de 25 horas.

Los períodos comprendidos entre el 25 de Marzo al 29 de Marzo se han establecido como no laborables en ISL Gijón, además de los días contemplados como no laborables en el calendario laboral 2013 en España.

El segundo incremento de trabajo se refleja sin duración dado que inicialmente no se conoce su estructura de tareas y por tanto no se puede realizar una estimación.

El tercer incremento se incluye en la planificación, aunque es opcional y se definiría en un momento posterior, siempre que se diesen las condiciones necesarias indicadas anteriormente.

En la Figura 3 se muestra la planificación inicial:

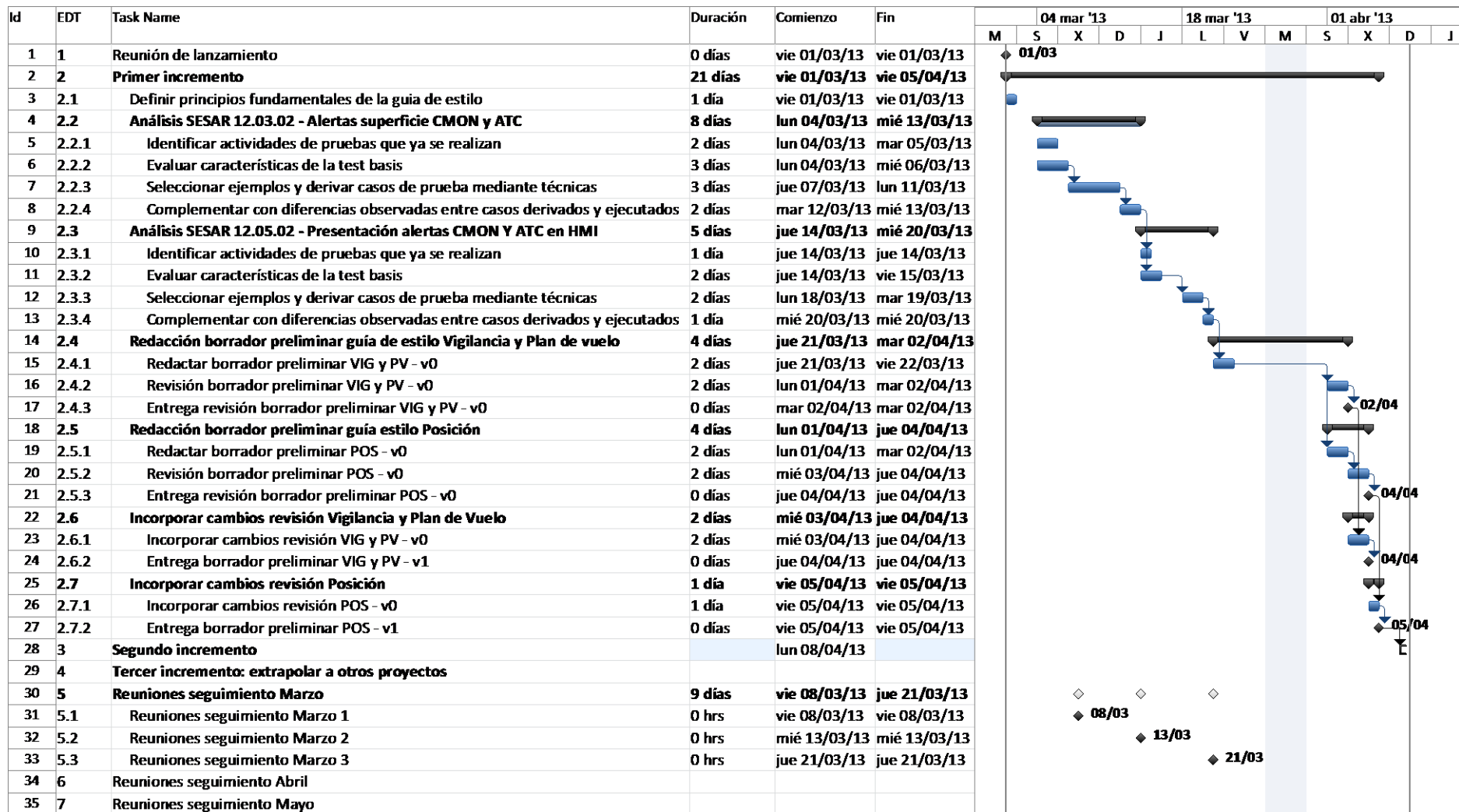


Figura 3. Planificación inicial

### **2.3.2. Seguimiento de la planificación**

En este apartado se recoge la información de seguimiento del proyecto en las fechas predeterminadas para llevar a cabo el control del progreso realizado.

#### **2.3.2.1. Primer seguimiento**

En la Figura 4 se presenta el gantt de seguimiento correspondiente, respecto a la línea base de la planificación inicial.

En el seguimiento se refleja que se subestimó el esfuerzo necesario para completar algunas de las tareas iniciales y debido a ello se comenzó con retraso las tareas posteriores.



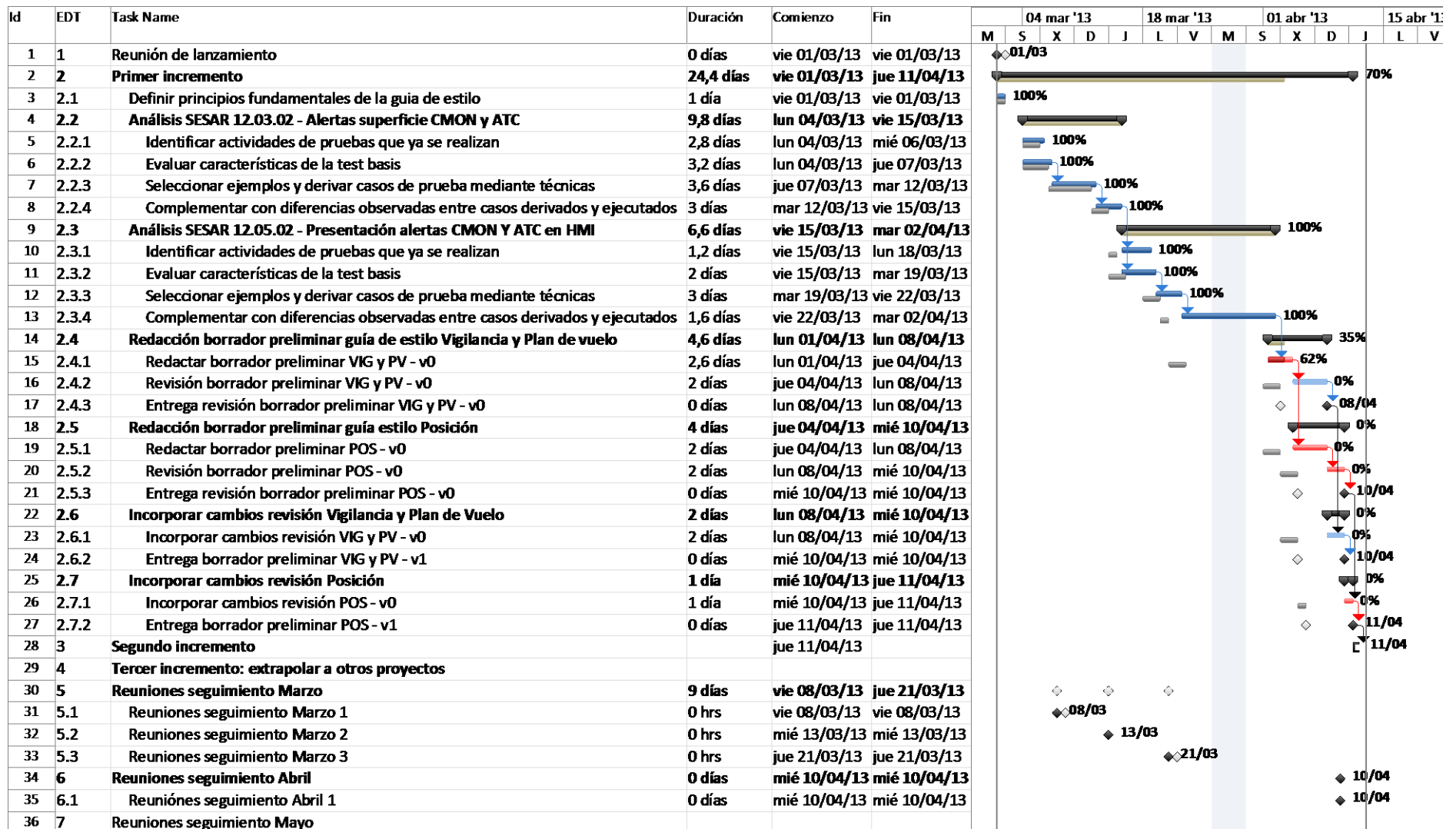


Figura 4. Seguimiento 1 - 01/04/2013

### 2.3.2.2. Segundo seguimiento

En la Figura 5 se presenta una instantánea del progreso a fecha 10 de Abril de 2013. Se corresponde con una reunión de seguimiento en donde, disponiendo ya de suficiente feedback de las tareas iniciales, se define la estructura de tareas del segundo incremento.

Debido a la inserción de nuevas tareas en la planificación, se crea también una nueva línea base en la fecha anterior.

En concreto se define el *core* de la estructura de contenidos que debería tener la guía técnica de pruebas orientada al desarrollador: 3 niveles que irán introduciendo gradualmente al desarrollador en la utilización de técnicas de prueba, desde los conceptos básicos al uso más avanzado.

En el primer incremento se había realizado un análisis rápido de los proyectos SESAR 12.03.02 y SESAR 12.05.02. Este se trata de un punto de inflexión en donde se decide priorizar el trabajo futuro, de manera que la redacción de los 3 niveles se realizará utilizando como referencia el proyecto SESAR 12.03.02 (VIG y PV) y posteriormente se completará con detalles específicos de SESAR 12.05.02 (POS).

Estimar con precisión el esfuerzo total necesario para completar estas nuevas tareas presentaba serias dificultades, ya que el trabajo real dependería del número de ciclos de revisión y cambios asociado a cada nivel (cada nivel implicaría varios ciclos hasta obtener un producto estable) que fuesen finalmente necesarios.

Debido a lo anterior, el esfuerzo que aparece reflejado se corresponde con el estimado para el primer ciclo de revisión y cambios de cada nivel, siendo el planteamiento adoptado el de ir imputando el trabajo realizado y actualizando el restante en cada nuevo ciclo posterior.



### *2.3.2.3. Tercer seguimiento*

En la Figura 6 se presenta una instantánea del progreso a fecha 1 de Mayo de 2013, respecto a la línea base guardada el 10 de Abril de 2013, que es cuando se define la estructura del segundo incremento.

El seguimiento del trabajo realizado refleja principalmente que, aunque inicialmente se había contemplado redactar de forma secuencial los 3 niveles relativos a las técnicas de prueba, en la realidad se decidió dar más prioridad a obtener versiones estables de los niveles 1 y 2, retrasando en consecuencia el trabajo sobre el nivel 3 y para completar con ejemplos de POS.

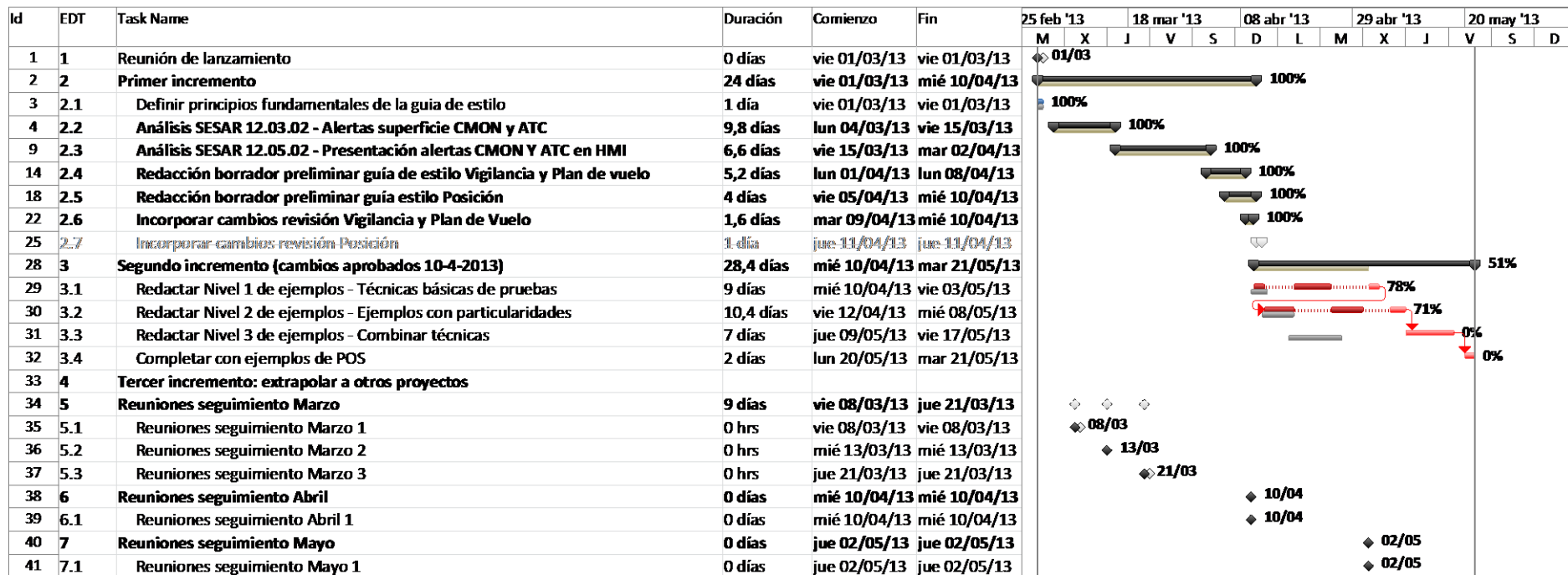


Figura 6. Seguimiento 3 - 01/05/2013

#### 2.3.2.4. Cuarto seguimiento

En la Figura 7 se presenta una instantánea del progreso a 4 de Junio de 2013, fecha en la que finaliza la parte del trabajo realizada durante las prácticas en empresa y comienza la parte del trabajo imputable al TFM como asignatura del Máster.

En este momento se aprueban las tareas restantes del trabajo, que consistiría en redactar un conjunto de criterios para elaborar estrategias de pruebas basadas en riesgos y se decide crear una nueva línea de base.

La ejecución de la tarea denominada *Tercer incremento: extrapolar a otros proyectos* estaba condicionada a que los resultados obtenidos de en el segundo incremento no fueran suficiente representativos. Dado que este no fue el caso, se decidió eliminarla de la planificación.



#### **2.3.2.5. Seguimiento final**

En la Figura 8 se presenta una instantánea del progreso a 5 de Julio de 2013, fecha real en la que finaliza el TFM, resultando una duración total de 85 días.

Durante el periodo de tiempo comprendido entre el 5 de Junio y el 5 de Julio de 2013 (trabajo imputado a trabajo individual no tutelado en la empresa) se asumió un horario laboral equivalente al que se tenía en el periodo de prácticas en empresa (25 horas semanales).

Al término de las tareas relativas a generar el contenido de la guía de estilo, se actualizó la planificación incluyendo tareas relativas a la redacción del resto de entregables del TFM. Este conjunto de tareas se refleja en la planificación como *Resto tareas TFM*.



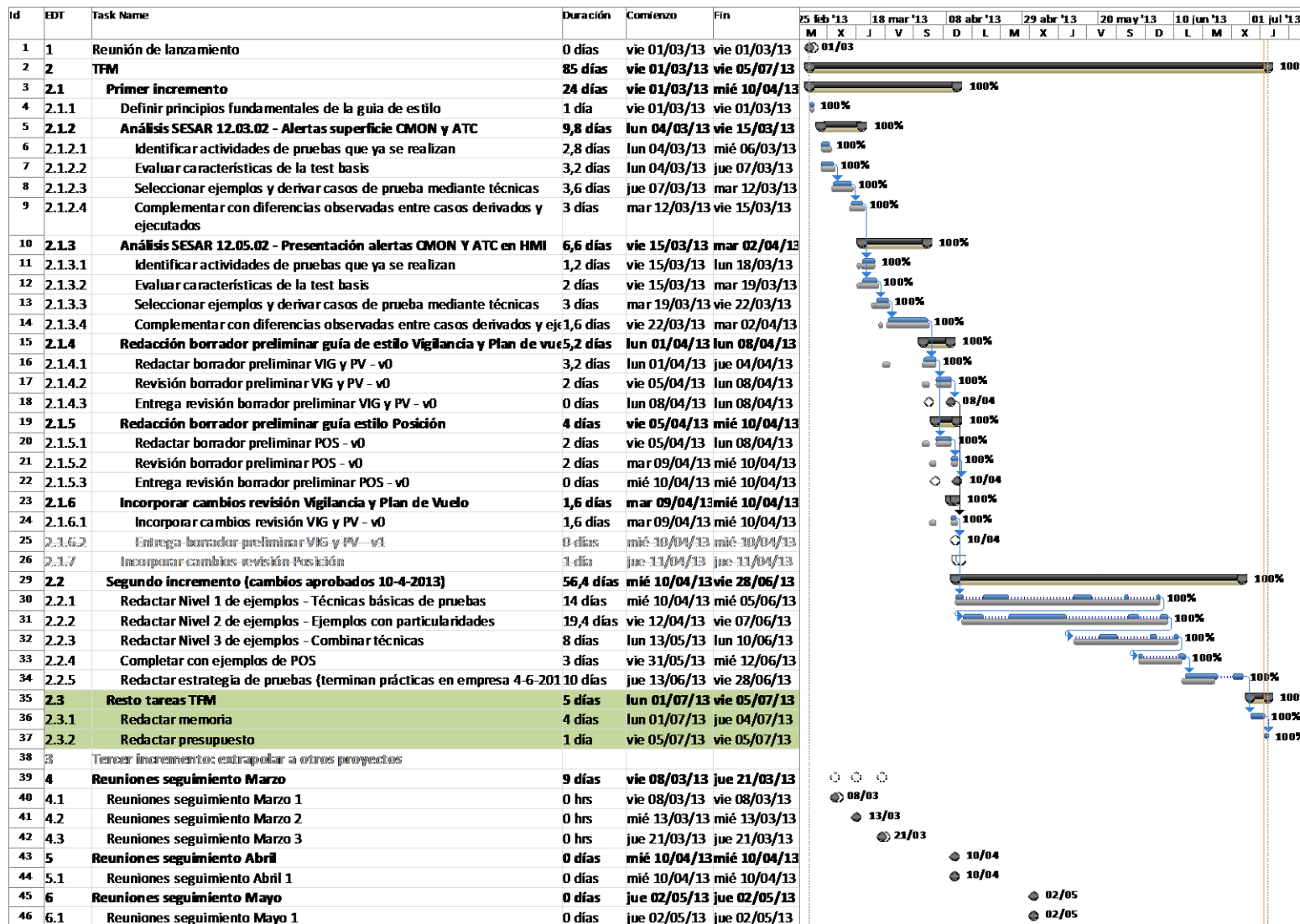


Figura 8. Seguimiento final - 05/07/2013

## 2.4. Presupuesto

El presupuesto real se ha obtenido contabilizando las horas reales empleadas en cada tarea del EDT.

Se dispone de un único recurso humano que tiene como coste unitario de 20 €/ hora. El coste/hora de revisión se establece en una media de 30 €/ hora, y se imputan las horas aproximadas correspondientes a las revisiones realizadas por todos los revisores en conjunto.

Nombre de tarea	Trabajo	Costo total	Previsto	Variación	Real
<b>Primer incremento</b>	<b>138 hrs</b>	<b>2.360,00 €</b>	<b>600,00 €</b>	<b>1.760,00 €</b>	<b>2.360,00 €</b>
Definir principios fundamentales de la guía de estilo	5 hrs	100,00 €	30,00 €	70,00 €	100,00 €
Análisis SESAR 12.03.02 - Alertas superficie CMON y ATC	48 hrs	960,00 €	225,00 €	735,00 €	960,00 €
Análisis SESAR 12.05.02 - Presentación alertas CMON Y ATC en HMI	31 hrs	620,00 €	135,00 €	485,00 €	620,00 €
Redacción borrador preliminar guía de estilo Vigilancia y Plan de vuelo	26 hrs	320,00 €	60,00 €	260,00 €	320,00 €
Redacción borrador preliminar guía estilo Posición	20 hrs	200,00 €	60,00 €	140,00 €	200,00 €
Incorporar cambios revisión Vigilancia y Plan de Vuelo	8 hrs	160,00 €	60,00 €	100,00 €	160,00 €
Incorporar cambios revisión Posición	5 hrs	100,00 €	30,00 €	70,00 €	0,00 €
<b>Segundo incremento</b>	<b>268 hrs</b>	<b>5.360,00 €</b>	<b>0,00 €</b>	<b>5.360,00 €</b>	<b>5.360,00 €</b>
Redactar Nivel 1 de ejemplos - Técnicas básicas de pruebas	70 hrs	1.400,00 €	0,00 €	1.400,00 €	1.400,00 €
Redactar Nivel 2 de ejemplos - Ejemplos con particularidades	93 hrs	1.860,00 €	0,00 €	1.860,00 €	1.860,00 €
Redactar Nivel 3 de ejemplos - Combinar técnicas	40 hrs	800,00 €	0,00 €	800,00 €	800,00 €
Completar con ejemplos de POS	15 hrs	300,00 €	0,00 €	300,00 €	300,00 €
Redactar estrategia de pruebas	50 hrs	1.000,00 €	0,00 €	1.000,00 €	1.000,00 €
<b>Resto tareas TFM</b>	<b>25 hrs</b>	<b>500,00 €</b>	<b>0,00 €</b>	<b>500,00 €</b>	<b>500,00 €</b>
Redactar memoria	20 hrs	400,00 €	0,00 €	400,00 €	400,00 €
Redactar presupuesto	5 hrs	100,00 €	0,00 €	100,00 €	100,00 €
<b>Revisores</b>	<b>72 hrs</b>	<b>2.160,00 €</b>			
<b>TOTAL</b>	<b>503,00 hrs</b>	<b>10.380,00 €</b>			

Tabla 4 se muestra el desglose de costes del proyecto, respecto a la línea base de la planificación inicial.

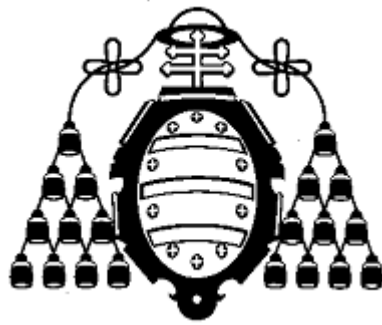
Nombre de tarea	Trabajo	Costo total	Previsto	Variación	Real
<b>Primer incremento</b>	<b>138 hrs</b>	<b>2.360,00 €</b>	<b>600,00 €</b>	<b>1.760,00 €</b>	<b>2.360,00 €</b>
Definir principios fundamentales de la guía de estilo	5 hrs	100,00 €	30,00 €	70,00 €	100,00 €
Análisis SESAR 12.03.02 - Alertas superficie CMON y ATC	48 hrs	960,00 €	225,00 €	735,00 €	960,00 €
Análisis SESAR 12.05.02 - Presentación alertas CMON Y ATC en HMI	31 hrs	620,00 €	135,00 €	485,00 €	620,00 €
Redacción borrador preliminar guía de estilo Vigilancia y Plan de vuelo	26 hrs	320,00 €	60,00 €	260,00 €	320,00 €
Redacción borrador preliminar guía estilo Posición	20 hrs	200,00 €	60,00 €	140,00 €	200,00 €
Incorporar cambios revisión Vigilancia y Plan de Vuelo	8 hrs	160,00 €	60,00 €	100,00 €	160,00 €
Incorporar cambios revisión Posición	5 hrs	100,00 €	30,00 €	70,00 €	0,00 €
<b>Segundo incremento</b>	<b>268 hrs</b>	<b>5.360,00 €</b>	<b>0,00 €</b>	<b>5.360,00 €</b>	<b>5.360,00 €</b>
Redactar Nivel 1 de ejemplos - Técnicas básicas de pruebas	70 hrs	1.400,00 €	0,00 €	1.400,00 €	1.400,00 €
Redactar Nivel 2 de ejemplos - Ejemplos con particularidades	93 hrs	1.860,00 €	0,00 €	1.860,00 €	1.860,00 €
Redactar Nivel 3 de ejemplos - Combinar técnicas	40 hrs	800,00 €	0,00 €	800,00 €	800,00 €
Completar con ejemplos de POS	15 hrs	300,00 €	0,00 €	300,00 €	300,00 €
Redactar estrategia de pruebas	50 hrs	1.000,00 €	0,00 €	1.000,00 €	1.000,00 €
<b>Resto tareas TFM</b>	<b>25 hrs</b>	<b>500,00 €</b>	<b>0,00 €</b>	<b>500,00 €</b>	<b>500,00 €</b>
Redactar memoria	20 hrs	400,00 €	0,00 €	400,00 €	400,00 €
Redactar presupuesto	5 hrs	100,00 €	0,00 €	100,00 €	100,00 €
<b>Revisores</b>	<b>72 hrs</b>	<b>2.160,00 €</b>			
<b>TOTAL</b>	<b>503,00 hrs</b>	<b>10.380,00 €</b>			

Tabla 4. Costes del proyecto (sin IVA)

El presupuesto final, incluyendo gastos indirectos y aplicando el IVA se muestra en la Tabla 5:

<b>Presupuesto ejecución material</b>	<b>10.380,00 €</b>
<b>Presupuesto de ejecución por contrata (21% IVA)</b>	<b>12.559,80 €</b>

**Tabla 5. Presupuesto de ejecución por contrata**



**UNIVERSIDAD DE OVIEDO**

**ESCUELA POLITÉCNICA DE INGENIERÍA DE  
GIJÓN**

**MÁSTER EN INGENIERÍA INFORMÁTICA**

**TRABAJO FIN DE MÁSTER**

**PROCEDIMIENTOS DE PRUEBAS EN DESARROLLO DE UNA  
FACTORIA DE SOFTWARE**

**DOCUMENTO Nº 3**

**GUIA DE PRUEBAS EN DESARROLLO**



**RUBÉN VALDÉS SECO  
JULIO 2013**

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

## **3.GUÍA DE PRUEBAS EN DESARROLLO**

### **3.1.Introducción**

En este documento se adjunta íntegramente la guía técnica de pruebas en desarrollo particularizada para la factoría Indra Software Labs Gijón.

### **3.2. Alcance**

El alcance fijado es la presentación de conceptos de diseño de pruebas y utilización de técnicas de diseño de pruebas a una audiencia con perfil de desarrollador de software. Los conceptos se introducen sobre ejemplos extraídos de los proyectos de referencia:

- SESAR 12.03.02 DISEÑO FUNCIONAL DE ALERTAS “CONFORMANCE MONITORING (CMON)” y “CONFLICTING ATC CLEARANCES (ATC)”, relativo a la detección de alertas en la superficie de un aeródromo y su presentación gráfica al controlador de torre.
- SESAR 12.05.02 cuyo objetivo es la presentación de las alertas anteriores en HMI (Human Machine Interface).

Así mismo también se presentan algunos criterios para asistir en la elaboración de estrategias de pruebas basadas en riesgos, dirigidos a una audiencia con perfil de gestor de proyecto.



**indra**

Indra Software Labs Gijón

# GUIA DE PRUEBAS EN DESARROLLO

Áreas de Plan de vuelo, Vigilancia y Posición

Edición: 1

Fecha: 12/07/2013

Páginas: 95

Este documento ha sido:

**Preparado por:**

Rubén Valdés

Becario

**Revisado por:**

Ramón Lema

Gerente proyecto

---

Iván García

Gerente proyecto

---

Carmen Méndez

Ingeniero software

---

Javier Tuya

Universidad Oviedo

**Aprobado por:**

**Autorizado por:**



# Índice de contenido

<b>1. PROPÓSITO .....</b>	<b>5</b>
<b>2. ALCANCE .....</b>	<b>5</b>
<b>3. DEFINICIONES Y SIGLAS .....</b>	<b>5</b>
3.1. DEFINICIONES .....	5
3.2. SIGLAS .....	6
<b>4. CONCEPTOS ESENCIALES DE DISEÑO DE PRUEBAS .....</b>	<b>7</b>
4.1. OBJETIVO DE PRUEBA .....	7
4.2. SITUACIÓN DE PRUEBA Y CASO DE PRUEBA .....	9
4.3. TEST BASIS .....	13
4.4. MODELO .....	13
<b>5. TÉCNICAS DE PRUEBA .....</b>	<b>16</b>
5.1. INTRODUCCIÓN A LAS TÉCNICAS BÁSICAS .....	16
5.1.1. <i>Partición en clases de equivalencia</i> .....	16
5.1.2. <i>Valores límite</i> .....	19
5.1.3. <i>Cobertura de puntos de decisión</i> .....	21
5.1.3.1. Cobertura de condiciones .....	22
5.1.3.2. Cobertura de decisiones .....	23
5.1.3.3. Cobertura de condición/decisión (C/D) .....	23
5.1.3.4. Cobertura de múltiple condición (MC) .....	24
5.1.3.5. Cobertura de condición/decisión modificada (MCDC) .....	25
5.1.4. <i>Cobertura de caminos</i> .....	27
5.1.4.1. Concepto de nivel de cobertura .....	27
5.1.4.2. Pasos para obtener nivel 2 .....	29
5.2. APLICAR TÉCNICAS .....	32
5.2.1. <i>Acoplamiento</i> .....	32
5.2.1.1. Condiciones acopladas repetidas .....	33
5.2.1.2. Acoplamiento dependiente del contexto .....	34
5.2.2. <i>Inferir modelos a partir de ejemplos</i> .....	40
5.3. COMBINAR TÉCNICAS .....	48
5.3.1. <i>Priorizar técnicas</i> .....	50
5.3.2. <i>Ejemplo: puntos de decisión + CE + VL</i> .....	51
5.3.2.1. Derivar situaciones de prueba con cada técnica .....	53
5.3.2.2. Diseño de los casos de prueba .....	61
5.3.3. <i>Ejemplo: cobertura de caminos + tabla de decisión</i> .....	64
<b>6. PRUEBAS EN HMI .....</b>	<b>69</b>
6.1. REVISAR TEST BASIS .....	69
6.2. IDENTIFICAR ELEMENTOS DE HMI .....	69
6.3. IDENTIFICAR INTERACCIONES DE USUARIO .....	69
6.4. IDENTIFICAR INTERFACES CON OTROS COMPONENTES .....	69
6.5. IDENTIFICAR ASPECTOS DE CONFIGURACIÓN .....	69
6.6. IDENTIFICAR PRODUCTOS DE PRUEBAS REUTILIZABLES .....	70
6.7. EJEMPLO: FUNCIONALIDADES GENÉRICAS DE ALERTAS CMON .....	70
6.7.1. <i>Revisar test basis</i> .....	70
6.7.2. <i>Identificar elementos de HMI</i> .....	71
6.7.3. <i>Identificar interacciones de usuario</i> .....	73
6.7.4. <i>Identificar interfaces con otros componentes</i> .....	74
6.7.5. <i>Identificar aspectos de configuración</i> .....	75
6.7.6. <i>Identificar productos de pruebas reutilizables</i> .....	77
6.7.7. <i>Diseño de pruebas</i> .....	77
6.7.7.1. Extender el modelo básico: configuración .....	81
<b>7. PRIORIZACIÓN DE PRUEBAS .....</b>	<b>83</b>
7.1. NIVEL 1: ESTRATEGIA DE PRUEBAS .....	83
7.1.1. <i>Identificar objetivos de prueba</i> .....	83

7.1.2. Definición de riesgo.....	83
7.1.3. Clases de riesgo .....	84
7.1.4. Intensidad de pruebas .....	85
7.1.5. Planificación de las pruebas .....	86
7.1.5.1. Tiempo de reparación, regresión y re-test .....	87
7.1.5.2. Criterios de entrada y salida .....	87
7.1.5.3. Criterios de suspensión y continuación.....	87
7.2. NIVEL 2: DISEÑO Y EJECUCIÓN DE PRUEBAS .....	87
7.2.1. Diseño de pruebas.....	88
7.2.1.1. Probar funcionalidad de un componente.....	88
7.2.1.2. Probar integración de varios componentes .....	88
7.2.1.3. Probar HMI.....	89
7.2.2. Ejecución de pruebas .....	89
7.2.2.1. Opcional: pruebas de regresión basadas en riesgo .....	90
<b>8. DOCUMENTACIÓN DE REFERENCIA .....</b>	<b>93</b>
<b>9. CONTROL DE CAMBIOS.....</b>	<b>94</b>

## Lista de figuras

FIGURA 1. OBJETIVO DE PRUEBA: UN REQUISITO .....	7
FIGURA 2. OBJETIVO DE PRUEBA: UN COMPONENTE.....	7
FIGURA 3. OBJETIVO DE PRUEBA: UNA FUNCIONALIDAD ESPECÍFICA.....	8
FIGURA 4. OBJETIVO DE PRUEBA: UN RIESGO .....	8
FIGURA 5. RELACIÓN ENTRE CONCEPTOS OBJETIVO, SITUACIÓN Y CASO DE PRUEBA .....	9
FIGURA 6. EXTRACTO DE LA ESPECIFICACIÓN DE LA ALERTA NO LANDING CLEARANCE.....	10
FIGURA 7. TRAZABILIDAD DE LOS CASOS DE PRUEBA CON LAS SITUACIONES DE PRUEBA.....	10
FIGURA 8. CASOS DE PRUEBA CP01 Y CP02 DE ALERTA NO LANDING CLEARANCE .....	12
FIGURA 9. EJEMPLO DE MODELO: UN DIAGRAMA DE TRANSICIÓN DE ESTADOS .....	14
FIGURA 10. ESPECIFICACIÓN PARCIAL DE ALERTA SAME RUNWAY.....	14
FIGURA 11. EJEMPLO DE MODELO: TABLA DE DECISIÓN.....	15
FIGURA 12. EJEMPLO DE MODELO: ESQUEMA HOLDING POINTS EN UN AERÓDROMO.....	15
FIGURA 13. ESPECIFICACIÓN DE ALERTA NO CONTACT .....	18
FIGURA 14. SITUACIONES DE PRUEBA DERIVADAS MEDIANTE CLASES DE EQUIVALENCIA .....	18
FIGURA 15. DETALLE ATRIBUTO TIME TO THRESHOLD.....	19
FIGURA 16. DISEÑO DE CASOS DE PRUEBA ALERTA NO CONTACT .....	19
FIGURA 17. VALORES LÍMITE SOBRE ATRIBUTO TIME TO THRESHOLD.....	20
FIGURA 18. DISEÑO DE PRUEBAS APLICANDO VALORES LÍMITE .....	20
FIGURA 19. ESTRUCTURA DE UN PUNTO DE DECISIÓN.....	21
FIGURA 20. ESPECIFICACIÓN DE ALERTA NO TAKE OFF CLEARANCE .....	22
FIGURA 21. COBERTURA DE CONDICIONES.....	22
FIGURA 22. COBERTURA DE CONDICIONES (CON SITUACIONES NO FACTIBLES) .....	22
FIGURA 23. DETALLE DE CP01 DE ALERTA NO TAKE-OFF CLEARANCE .....	23
FIGURA 24. COBERTURA DE CONDICIÓN/DECISIÓN MODIFICADA.....	25
FIGURA 25. DETALLE DE CP01 DISEÑADO APLICANDO MCDC .....	27
FIGURA 26. SITUACIONES DE PRUEBA PARA EL NIVEL 1 .....	28
FIGURA 27. DETALLE DE CP01 CORRESPONDIENTE AL NIVEL 1 .....	28
FIGURA 28. SITUACIONES DE PRUEBA PARA EL NIVEL 2.....	28
FIGURA 29. DETALLE CP01 Y CP02 PARA NIVEL 2 DE COBERTURA DE CAMINOS .....	29
FIGURA 30. TRAZABILIDAD CP01 Y CP02 PARA NIVEL 2 .....	29
FIGURA 31. ESPECIFICACIÓN DE ALERTA NO PUSH BACK CLEARANCE .....	30
FIGURA 32. MODELO DE ESTADOS ALERTA NO PUSH BACK CLEARANCE .....	31
FIGURA 33. DETALLE DE CP01 NIVEL 1 PARA NO PUSH BACK CLEARANCE .....	31
FIGURA 34. OBTENCIÓN SITUACIONES DE PRUEBA NIVEL 2 DE COBERTURA CAMINOS.....	31
FIGURA 35. DETALLE CP01 Y CP02 NIVEL 2 NO PUSH BACK CLEARANCE .....	32
FIGURA 36. EJEMPLO DE CONDICIONES ACOPLADAS REPETIDAS.....	33
FIGURA 37. EJEMPLO DE ACOPLAMIENTO DEPENDIENTE DEL CONTEXTO .....	35
FIGURA 38. ESPECIFICACIÓN DE ALERTA STATIONARY AFTER CLEARANCE .....	36
FIGURA 39. MCDC SOBRE CONDICIONES ALERTA STATIONARY AFTER CLEARANCE .....	36
FIGURA 40. MCDC SEPARANDO POR TIPO DE MÓVIL = AERONAVE.....	37
FIGURA 41. MCDC SEPARANDO POR TIPO DE MÓVIL = VEHÍCULO .....	37



FIGURA 42. MCDC SEPARANDO POR FASE CUANDO TIPO MÓVIL = AERONAVE .....	38
FIGURA 43. TRAZABILIDAD CP01 EJERCITANDO SITUACIONES DE LA FIGURA 40.....	38
FIGURA 44. DETALLE CP01 EJERCITANDO SITUACIONES DE LA FIGURA 40 .....	39
FIGURA 45. ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	41
FIGURA 46. MODELO INFERIDO A PARTIR DE ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	43
FIGURA 47. DERIVAR SITUACIONES DE PRUEBA NIVEL 2 A PARTIR DEL MODELO DE ESTADOS.....	44
FIGURA 48. SITUACIÓN A) DESCRITA EN LA ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	45
FIGURA 49. SITUACIÓN B) DESCRITA EN LA ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	45
FIGURA 50. SITUACIÓN C) DESCRITA EN LA ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	45
FIGURA 51. SITUACIÓN D) DESCRITA EN LA ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	46
FIGURA 52. SITUACIÓN E) DESCRITA EN LA ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	46
FIGURA 53. SITUACIÓN F) DESCRITA EN LA ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	46
FIGURA 54. SITUACIÓN G) DESCRITA EN LA ESPECIFICACIÓN DE ALERTA RED STOP BAR .....	47
FIGURA 55. OTROS CASOS DE PRUEBA DE ALERTA RED STOP BAR (1 DE 2) .....	47
FIGURA 56. OTROS CASOS DE PRUEBA DE ALERTA RED STOP BAR (2 DE 2) .....	48
FIGURA 57. ESPECIFICACIÓN DEL EJEMPLO SINTÉTICO.....	49
FIGURA 58. ESPECIFICACIÓN DE ALERTA OPPOSITE RUNWAYS .....	53
FIGURA 59. CONSTRUCCIÓN DE UNA TABLA DE DECISIÓN .....	54
FIGURA 60. TABLA DE DECISIÓN ALERTA OPPOSITE RUNWAYS POR LU (LINE UP) .....	55
FIGURA 61. TABLAS DE DECISIÓN ALERTA POR AUTORIZACIONES LU & TO Y TO & TO.....	55
FIGURA 62. COMBINAR SITUACIONES DE MCDC CON SITUACIONES DE REVOCACIONES.....	56
FIGURA 63 CLASES DE EQUIVALENCIA ALERTA OPPOSITE RUNWAYS POR LU & LU.....	57
FIGURA 64 CLASES DE EQUIVALENCIA ALERTA OPPOSITE RUNWAYS POR LU & TO .....	58
FIGURA 65. CLASES DE EQUIVALENCIA ALERTA OPPOSITE RUNWAYS POR TO & TO.....	59
FIGURA 66. DETALLE CASO DE PRUEBA QUE EJERCITA PUNTOS DE DECISIÓN + CE + VL .....	63
FIGURA 67. ESPECIFICACIÓN DE LA FUNCIÓN DE CÁLCULO DE ALERTA RED STOP BAR .....	65
FIGURA 68 TABLA DECISIÓN CONDICIONES CÁLCULO ALERTA RED STOP BAR .....	66
FIGURA 69. SITUACIONES DE PRUEBA FACTIBLES DE RED STOP BAR .....	66
FIGURA 70. DETALLE DE CP01 QUE EJERCITA SITUACIONES DE LA TABLA DE DECISIÓN .....	68
FIGURA 71. COTEJAR ALERTAS CMON Y LAS DE INCURSIÓN EN PISTA (FUNCIONALIDAD).....	71
FIGURA 72. COTEJAR ALERTAS CMON Y LAS DE INCURSIÓN EN PISTA (PRESENTACIÓN).....	71
FIGURA 73. ESQUEMA GENERAL DE ZONAS DE INTERACCIÓN EN HMI .....	72
FIGURA 74. ESPECIFICACIÓN DE ALERTA OPPOSITE STOP POINTS .....	72
FIGURA 75. ESTRUCTURA DE ALGUNOS ELEMENTOS DE HMI.....	73
FIGURA 76. ESPECIFICACIÓN RECONOCIMIENTO DE ALERTAS .....	74
FIGURA 77. IDENTIFICAR INTERFACES CON OTROS COMPONENTES .....	74
FIGURA 78. PUBLICACIÓN DEL EVENTO DE ALERTA SAME/NEARBY STOP POINTS .....	75
FIGURA 79. ESPECIFICACIÓN FILTRO ON-GROUND DE PISTAS DE SUPERFICIE .....	75
FIGURA 80. ESPECIFICACIÓN FILTROS DE LAS LÍNEAS 1,2,3,4 Y CFL DE LAS ETIQUETAS DE PISTA...76	76
FIGURA 81. ESPECIFICACIÓN FILTROS DE CORR Y NO CORR .....	76
FIGURA 82. ESPECIFICACIÓN FUNCIONALIDAD SUPRIMIR ETIQUETA DE PISTA .....	76
FIGURA 83. ESPECIFICACIÓN DESACTIVACIÓN TOTAL DE FILTROS .....	76
FIGURA 84. ESPECIFICACIÓN SELECCIÓN DE LÍMITES DE LOS FILTROS DE ALTITUD .....	76
FIGURA 85. ESPECIFICACIÓN SELECCIÓN DE OPERATIVA.....	77
FIGURA 86. PRIORIDAD DE ALERTAS CMON MOSTRADAS EN OG GENÉRICO.....	78
FIGURA 87. MODELO INFERIDO A PARTIR DE LA ESPECIFICACIÓN DE PRESENTACIÓN DE ALERTAS....	79
FIGURA 88. DETALLE CP01 PRESENTACIÓN DE ALERTAS.....	80
FIGURA 89. FLUJO CP01 PRESENTACIÓN DE ALERTAS.....	81
FIGURA 90. PRIORIZACIÓN EN LA EJECUCIÓN DE CASOS DE PRUEBA .....	90
FIGURA 91. ASIGNACIÓN DE PESO A LOS CASOS DE PRUEBA DE REGRESIÓN .....	91

## Lista de tablas

TABLA 1. PROYECTOS INCLUIDOS EN EL ALCANCE .....	5
TABLA 2. ALGUNAS PAUTAS GENERALES PARA IDENTIFICAR SITUACIONES DE PRUEBA .....	11
TABLA 3. PAUTAS PARA IDENTIFICAR CLASES DE EQUIVALENCIA .....	17
TABLA 4. VARIANTES DE LA TÉCNICA DE COBERTURA DE PUNTOS DE DECISIÓN.....	21
TABLA 5. COBERTURA DE DECISIONES .....	23
TABLA 6. COBERTURA DE CONDICIÓN/DECISIÓN .....	24
TABLA 7. COBERTURA DE MÚLTIPLE CONDICIÓN.....	24
TABLA 8. MCDC EN COMPARACIÓN CON RESTO DE VARIANTES .....	26
TABLA 9. DESCRIPCIÓN INTERACCIONES DE USUARIO RELATIVAS A ALERTAS SOBRE UNA PISTA.....	74
TABLA 10. EJEMPLO DE LISTADO DE OBJETIVOS DE PRUEBA .....	83
TABLA 11. EJEMPLO DE CLASIFICACIÓN DE RIESGO CON 2 NIVELES DE CRITICIDAD .....	85
TABLA 12. EJEMPLO DE CLASIFICACIÓN DE OBJETIVOS DE PRUEBA EN NIVELES DE RIESGO .....	85
TABLA 13. EJEMPLO DE CLASIFICACIÓN DE NIVELES DE INTENSIDAD DE PRUEBA .....	86
TABLA 14. EJEMPLO DE ESTRATEGIA DE PRUEBAS BASADA EN RIESGOS.....	86
TABLA 15. PROBLEMAS ESTRATEGIA DE PRUEBAS DE REGRESIÓN BASADAS EN RIESGO.....	92

# 1. PROPÓSITO

Este documento tiene como objetivo definir un conjunto de criterios y recomendaciones a seguir en el diseño y ejecución de pruebas realizadas en el proceso de desarrollo de software de *Indra Software Labs Gijón* (ISL Gijón). Con esto se pretende disponer de unas buenas prácticas que ayuden a los desarrolladores a realizar pruebas.

El presente documento corresponde al borrador preliminar de la guía de estilo de pruebas en desarrollo para las áreas de ISL Gijón de Plan de Vuelo, Vigilancia y Posición.

## 2. ALCANCE

El objetivo de este documento es introducir al desarrollador en la aplicación de técnicas de prueba para el tipo de desarrollos que se llevan a cabo en ISL Gijón.

Para ello, se realizó un estudio de los siguientes proyectos, para realizar una valoración inicial sobre las tareas que ya se vienen realizando e identificar las técnicas de prueba aplicables:

Proyecto	Áreas afectadas	Descripción
SESAR 12.03.02	Plan de Vuelo y Vigilancia	Cálculo de alertas de superficie CMON y ATC
SESAR 12.05.02	Posición	Presentación de alertas de superficie CMON y ATC

**Tabla 1.** Proyectos incluidos en el alcance

Aunque el conjunto de prácticas que recoge este documento se ha obtenido solamente en base al estudio de los proyectos indicados en la Tabla 1, se considera suficientemente representativo del tipo de desarrollos que se llevan a cabo en esta factoría.

## 3. DEFINICIONES Y SIGLAS

### 3.1. Definiciones

En este apartado se recogen los términos básicos cuya explicación es necesaria para comprender el dominio del proyecto:

- **Pruebas en desarrollo:** dada una funcionalidad implementada por un desarrollador, se refiere a las pruebas que este realiza para verificar que el código construido cumple la especificación. Se trata por tanto de pruebas que involucran uno o varios componentes de un sistema.
- **Objetivo de prueba:** se refiere a lo que va ser probado, como por ejemplo un requisito, un componente, una funcionalidad específica de un componente o un determinado riesgo de un producto.
- **Situación de prueba:** se refiere a lo que debería ser probado respecto a un objetivo de prueba, como por ejemplo un valor concreto, un rango de valores, una combinación de valores o una transición de un estado a otro.
- **Caso de prueba:** es la forma de ejercitar situaciones de prueba y tiene la siguiente estructura:
  - *Situación inicial:* es la situación de partida en la que tiene que estar el sistema y su entorno de ejecución.
  - Uno o varios *pasos de prueba:* cada uno tiene un conjunto de valores de entrada y una descripción de las acciones que es necesario realizar para procesarlos. Los pasos de prueba se ejecutan secuencialmente y cada uno tiene una salida esperada que podría ser la entrada del siguiente paso de prueba.

Cada caso de prueba puede ejercitar una o varias situaciones de prueba o varias combinaciones de situaciones de prueba.

- **Test basis:** cualquier tipo de información que se considere relevante para el diseño de pruebas, como por ejemplo especificaciones de requisitos, documentos de diseño funcional, documentos que describen la arquitectura del sistema a probar o manuales de usuario.
- **Modelo:** una forma de representar la especificación que facilita la derivación de situaciones de prueba, como por ejemplo diagramas de transición de estados, tablas de decisión o condiciones complejas.
- **Técnica de diseño de pruebas:** es un procedimiento estándar para identificar situaciones de prueba de forma sistemática.
- **Riesgo:** probabilidad de que ocurra un fallo en relación al daño esperado si este se produce. Se puede definir mediante la fórmula

$$\text{Riesgo} = \text{Probabilidad de fallo} \times \text{Daño}$$

donde:

- Probabilidad de fallo = Probabilidad de presencia de defectos x Frecuencia de uso
- Daño: un daño directo podría involucrar pérdidas de ingresos, daños a terceros, daños físicos o en el medio ambiente (en especial cuando se trata de sistemas empotrados) o sobrecostos por correcciones.

Un daño indirecto podría involucrar daño en la imagen corporativa o pérdida de confianza de los clientes entre otros.

- **Intensidad de pruebas:** nivel de detalle con el que se decide probar un determinado objetivo de prueba, atendiendo al nivel de riesgo identificado para el mismo (normalmente a mayor riesgo mayor intensidad de pruebas). Involucra la aplicación de una o varias técnicas de prueba.
- **Estrategia de pruebas basada en riesgos:** documento que recoge el desglose de tareas de desarrollo para un proyecto, donde cada tarea tiene asociado una intensidad de pruebas determinada en función de los riesgos identificados para ella. Es la base para la planificación de las tareas, priorizándolas de acuerdo a su riesgo asociado (las de mayor riesgo primero).
- **Regresión:** se experimenta cuando la calidad de un sistema se deteriora como consecuencia de haber introducido cambios.
- **Pruebas de regresión:** su objetivo es verificar que tras la implementación de un cambio el resto de partes del sistema que no han sido modificadas funcionan correctamente.

## 3.2. Siglas

- **CE:** Clases de equivalencia
- **VL:** Valores límite
- **MC:** Múltiple condición
- **MCDC:** Múltiple condición/decisión modificada

## 4. CONCEPTOS ESENCIALES DE DISEÑO DE PRUEBAS

En esta sección se definen algunos conceptos que serán utilizados en el resto del documento.

### 4.1. Objetivo de prueba

En el contexto de una tarea de desarrollo, el término *objetivo de prueba* se refiere a *qué va ser probado* p.e. un requisito, un componente, una funcionalidad específica de un componente o un riesgo.

#### Ejemplo 1

Un requisito, en este caso del usuario final (REQ-06).

#### EXTRACTO DE LA ESPECIFICACIÓN

...  
PCI: SC#7047

...  
RESUMEN:

La presente UCR SC#7047 establece la definición de las alertas necesarias para una implementación de A-SMGCS de nivel 2. Dichas alertas consisten en las Alertas de Incursión en Pista y las Alertas por Zonas de Rodaje Restringidas.

...  
REQ-06: La función de detección de alertas de incursión en pista será aplicable a los conflictos aeronave-aeronave y aeronave-vehículo. No considerará conflictos vehículo-vehículo.

Figura 1. Objetivo de prueba: un requisito

#### Ejemplo 2

Una parte de un componente, en este caso el objetivo de prueba sería el bloque funcional de las "Alertas CMON" (resaltado en color rojo) que forma parte del componente SNETS.

#### EXTRACTO DE LA ESPECIFICACIÓN

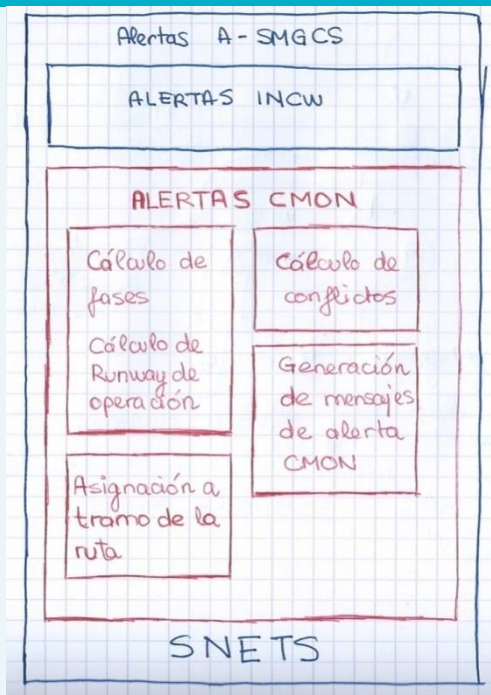
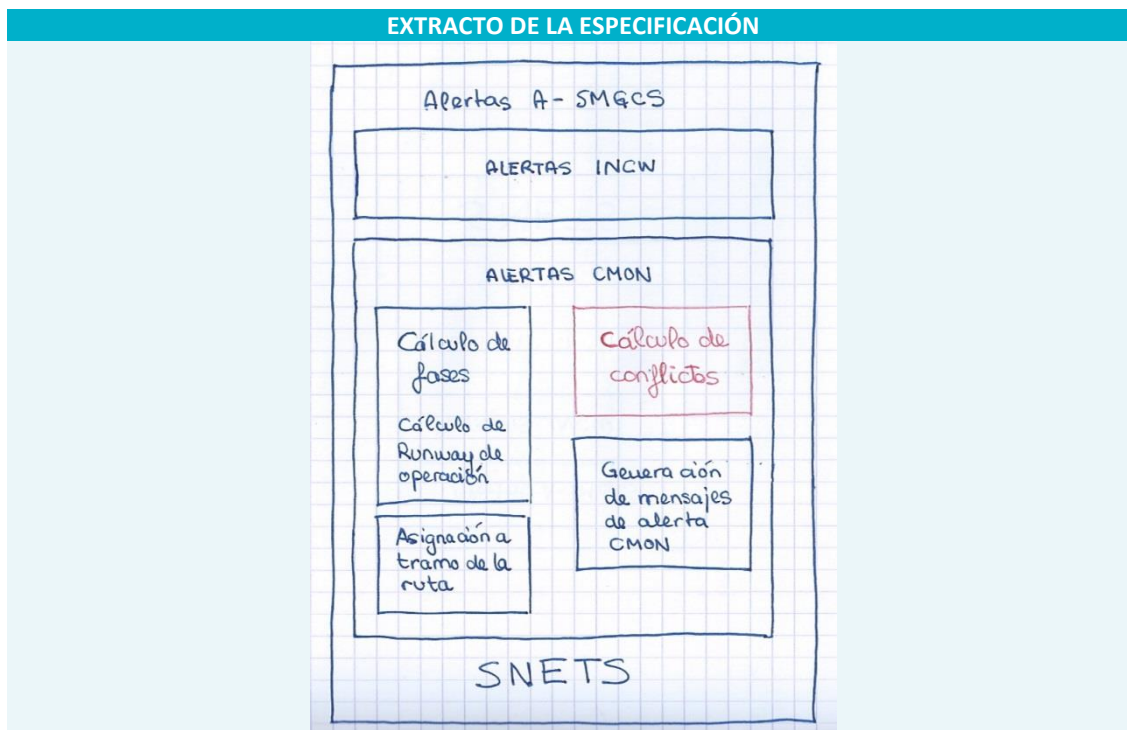


Figura 2. Objetivo de prueba: un componente

### Ejemplo 3

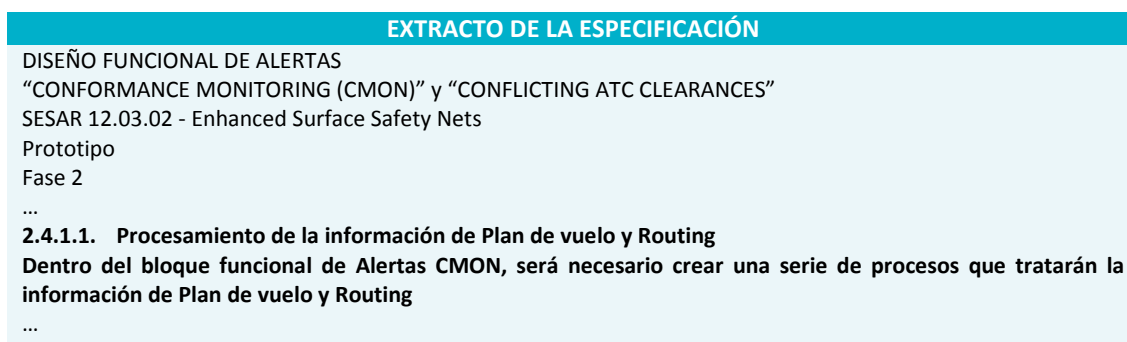
Una funcionalidad específica de un componente, en este caso el objetivo de prueba sería la funcionalidad “Cálculo de conflictos” (resaltado en color rojo) que forma parte del bloque funcional de las alertas CMON, el cual pertenece al componente SNETS.



**Figura 3.** Objetivo de prueba: una funcionalidad específica

### Ejemplo 4

Un riesgo del producto, en este caso el malfuncionamiento del proceso que envía información de Plan de vuelo y Routing al proceso que hace el tratamiento de las alertas CMON. Por lo tanto aquí el objetivo de prueba sería hacer pruebas de integración entre el bloque funcional de las alertas CMON con Plan de Vuelo y Routing.



**Figura 4.** Objetivo de prueba: un riesgo

## 4.2. Situación de prueba y caso de prueba

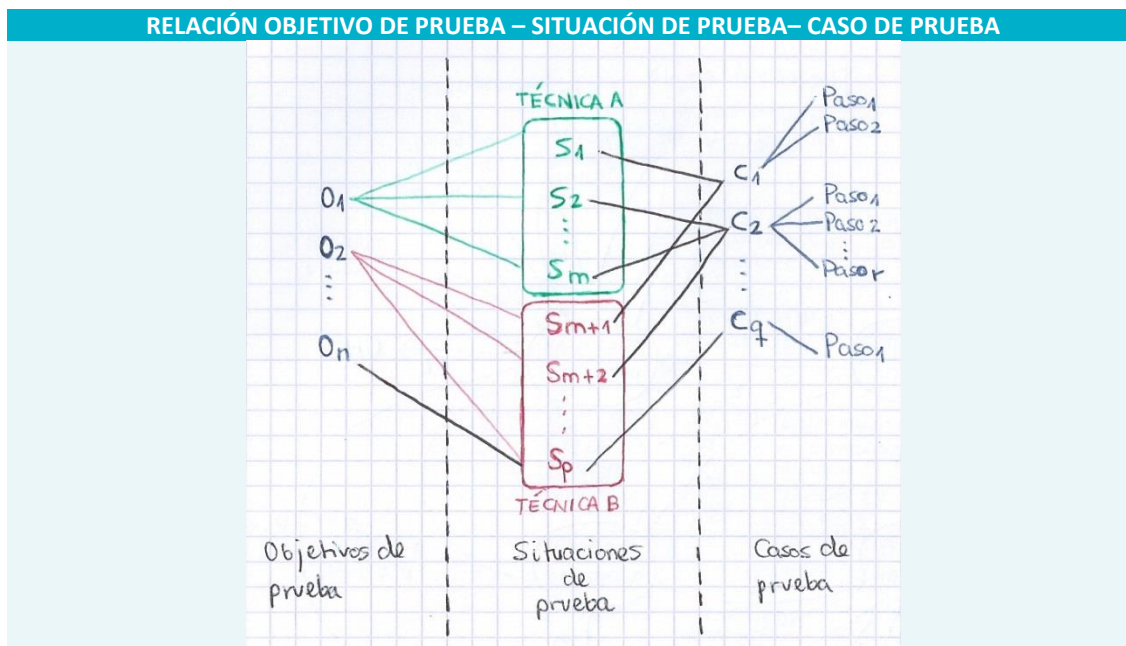
Una situación de prueba se puede definir como *lo que debería ser probado* respecto a un objetivo de prueba, como por ejemplo un valor concreto, un rango de valores, una combinación de valores o una transición de un estado a otro.

Como se verá en secciones posteriores, la utilización de técnicas de prueba permite derivar situaciones de forma sistemática.

Un *caso de prueba* es la forma de ejercitar situaciones de prueba y tiene la siguiente estructura:

- *Situación inicial*: es la situación de partida en la que tiene que estar el sistema y su entorno de ejecución.
- *Uno o varios pasos de prueba*: cada uno tiene un conjunto de valores de entrada y una descripción de las acciones que es necesario realizar para procesarlos. Los pasos de prueba se ejecutan secuencialmente y cada uno tiene una salida esperada que podría ser la entrada del siguiente paso de prueba.

Cada caso de prueba puede ejercitar una o varias situaciones de prueba o varias combinaciones de situaciones de prueba:



**Figura 5.** Relación entre conceptos objetivo, situación y caso de prueba

### Ejemplo 5

El desarrollador dispone de la siguiente información sobre las condiciones que se tienen que dar para detectar la alerta NO LANDING CLEARANCE (CMON):

#### EXTRACTO DE LA ESPECIFICACIÓN

##### Requisitos SESAR asociados

The SSN server shall generate an alert (ALARM) if an aircraft is approaching to a runway without having received a Landing clearance.

##### Descripción extraída del diseño funcional

...

Detección del conflicto:

Si se cumplen todas las condiciones siguientes para el móvil:

- Es una Aeronave

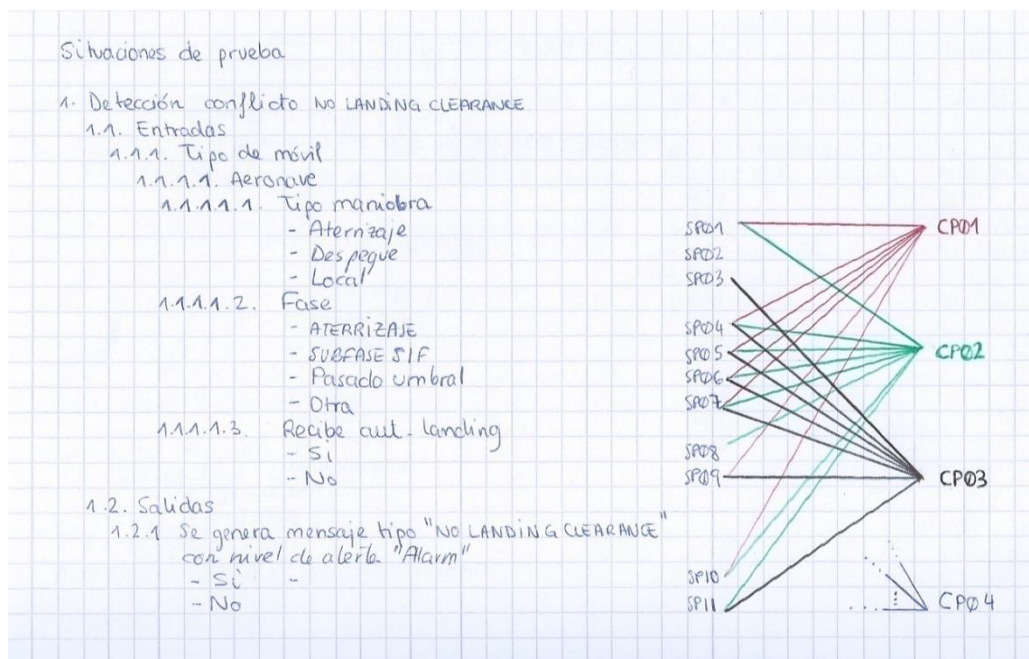
- Es un Aterrizaje (en PV\_Phase).
- Está en fase ATERRIZAJE, subfase S/F o PASADO EL UMBRAL
- No ha recibido Autorización Landing (no aparece en PV\_Clearances\_List).

Se generará un conflicto tipo "No Landing Clearance" y nivel de alerta "Alarm".

...

**Figura 6.** Extracto de la especificación de la alerta NO LANDING CLEARANCE

Teniendo en cuenta la información anterior, se podría pensar en las siguientes situaciones, que serían un subconjunto de todas las posibles:



**Figura 7.** Trazabilidad de los casos de prueba con las situaciones de prueba

Existe cierta clase de aspectos que es necesario considerar cuando se piensa en posibles situaciones de prueba. Podemos destacar algunos de ellos sobre el ejemplo anterior:

En la test basis se indica que el tipo de maniobra que interviene en la detección del conflicto NO LANDING CLEARANCE es "aterrizaje", pero ¿qué pasaría con "otros" tipos de maniobra? ¿Los hay?

En este caso hay 3 tipos de maniobra posibles:

- Aterrizaje (cuando aeródromo actual = destino del PV): es el tipo de maniobra que interviene en la detección del conflicto.
- Despegue (cuando aeródromo actual = origen del PV): este tipo de maniobra no se va a poder cubrir debido a que las fases que intervienen se salen del alcance de la alerta, que está vinculada a aeronaves que descienden hacia pista.
- Local (cuando en el PV hay mismo aeródromo origen y destino): según la especificación para este tipo de maniobra intervienen todas las fases posibles, de manera que habrá que considerarla.

Se podría analizar de forma similar otras condiciones que intervienen en la detección de la alerta:

- ¿Qué pasa con el resto de fases que no se citan explícitamente entre las condiciones que detectan el conflicto? De ahí que en la lista de situaciones de prueba se haya indicado "otra", en referencia a cualquier otra fase que pudiese intervenir.



- Respecto a la autorización de landing, ¿haberla recibido es suficiente para que no se pueda generar esta alerta? ¿Qué pasaría si se da la autorización y luego se revoca? ¿Se seguiría generando la alerta en ese supuesto?
- En línea con la anterior, ¿qué pasa si cuando el vuelo está aterrizando se da autorización de despegue? ¿Es esto posible?

Todas estas preguntas dejan entrever una idea general muy importante: dada una especificación para la que se va a diseñar pruebas, resulta necesario tener una predisposición a cuestionarse todos los aspectos relacionados con ella, tanto los que se indican explícitamente como los que no están especificados.

Es muy importante plantearse este tipo de cuestiones y no presuponer nada: si se desconoce algo, será necesario confirmarlo por algún medio.

En este ejemplo se puede ver que algunas cuestiones ya surgen de forma natural, simplemente atendiendo a las características de los atributos que intervienen en la detección del conflicto.

Aunque en secciones posteriores se presentará una serie de técnicas que facilitan la identificación de situaciones de prueba, se pueden adelantar ya algunas pautas generales:

Aspecto	Se podría comprobar ...
Entradas	<p>¿Qué valores introduce el usuario?</p> <p>¿Qué ocurre con los campos que se dejan vacíos?</p> <p>¿Hay valores que no introduce el usuario pero son relevantes para determinar la salida?</p>
Salidas	<p>¿Qué valores se deben actualizar en el interfaz de usuario?</p> <p>¿Hay otros valores que se deben actualizar que no se reflejan en el interfaz? (por ejemplo, logs o bases de datos)</p>
Atributos que toman un valor concreto (o varios) de un conjunto delimitado de valores discretos (por ejemplo, una enumeración o lista)	<p>¿Qué ocurre cuando el atributo toma el valor indicado?</p> <p>¿Qué ocurre cuando toma otro distinto?</p>
Atributos booleanos	<p>¿Qué ocurre si toma valor cierto?</p> <p>¿Y si toma valor falso?</p> <p>¿Es posible una situación en donde no tome ningún valor de verdad (por ejemplo, el valor NULL)?</p>
Atributos que toman un valor en un rango de valores continuos	<p>¿Qué ocurre cuando el atributo toma un valor dentro del rango?</p> <p>¿Y cuando toma un valor fuera del rango?</p>
Volumen	<p>¿Qué ocurre cuando se procesa el máximo número de entidades relativas a algún cálculo? (por ejemplo, máximo número de alertas o pistas configurado)</p> <p>¿Qué ocurre cuando se está procesando el máximo y aparecen nuevas entidades?</p>
Rendimiento	<p>¿Cuál es el tiempo de respuesta en situaciones de volumen máximo?</p> <p>¿Y en situaciones de máximo uso de recursos del sistema? (por ejemplo, procesador o memoria)</p>

**Tabla 2.** Algunas pautas generales para identificar situaciones de prueba

El primer paso para diseñar un caso de prueba consiste en pensar qué situaciones va a ejercitar. Posteriormente se pensará en los datos y la secuencia de acciones necesarias para llevarlo a cabo.

Mantener la trazabilidad con las situaciones de prueba ejercitadas facilita además comprobar qué situaciones quedan pendientes de cubrir con casos de prueba.

En este ejemplo se podrían diseñar los siguientes casos de prueba:

- CP01: una maniobra de aterrizaje en la que no se da autorización de landing.
- CP02: una maniobra de aterrizaje en la que se da autorización de landing, posteriormente se revoca y se vuelve a dar dicha autorización.
- CP03: un vuelo local que está en maniobra de aterrizaje y no recibe autorización de landing.
- CP04: igual al CP03, pero dando autorización de landing.

A continuación se presenta el detalle de los dos primeros casos de prueba:

CP01	Paso 1	Paso 2	Paso 3	Paso 4
Entrada	Fase PV = SELECTED	Fase PV = SELECTED	Fase PV = ATERRIZAJE LIF	Fase PV = ATERRIZAJE SIF
Descripción	Fase PV = SELECTED Autorización = ninguna	Fase PV = ATERRIZAJE LIF Autorización = ninguna	Fase PV = ATERRIZAJE SIF Autorización = ninguna	Fase PV = Pasado umbral Autorización = ninguna
Salida	No se genera alerta	Al cambiar de fase se genera alerta	se mantiene la alerta	se mantiene alerta
	Paso 5			
Entrada	Fase PV = Pasado umbral			
Descripción	Fase PV = TAXI Autorización = ninguna			
Salida	Desaparece alerta			
CP02	Paso 1	Paso 2	Paso 3	Paso 4
Entrada	Fase PV = SELECTED	Fase PV = SELECTED	Fase PV = ATERRIZAJE LIF	Fase PV = ATERRIZAJE SIF
Descripción	Fase PV = SELECTED Autorización = landing	Fase PV = ATERRIZAJE LIF Autorización = ninguna	Fase PV = ATERRIZAJE SIF Revocar aut. landing	Fase PV = Pasado umbral Autorización = landing
Salida	No se genera alerta	Al cambiar de fase no se genera alerta	Al cambiar de fase no se genera alerta Al revocar aut. se genera alerta	Al cambiar de fase se mantiene alerta Al dar aut. desaparece alerta
	Paso 5			
Entrada	Fase PV = Pasado umbral			
Descripción	Fase PV = TAXI Revocar aut. landing			
Salida	Al cambiar de fase no se genera alerta Al revocar aut. tampoco se genera alerta			

**Figura 8.** Casos de prueba CP01 y CP02 de alerta NO LANDING CLEARANCE

### 4.3. Test basis

La *test basis* se refiere a cualquier tipo de información que se considere relevante para el diseño de pruebas.

En general, las entradas más directas que tiene el desarrollador para comenzar la implementación de sus tareas serán especificaciones de requisitos y documentos de diseño funcional y/o diseño técnico, pero las fuentes de información relevantes para acometer el diseño de pruebas podrían no limitarse a estas.

Por este motivo puede ser necesaria una mínima labor inicial de análisis de cada tarea particular para identificar qué otra información puede ser significativa de cara al diseño de pruebas.

A continuación se enumera una serie de tareas que pueden ser de utilidad para completar la test basis, tales como inspeccionar existencia de:

- Interfaces entre el componente (o parte del mismo) sujeto a pruebas y otros componentes, que puede hacer necesaria la recopilación de información sobre esos componentes relacionados.
- Documentación relativa a aspectos no funcionales no indicados explícitamente en el diseño funcional o técnico, p.e. modos de funcionamiento del sistema que puedan condicionar el tratamiento de la información que se realiza en alguna funcionalidad (modo normal/autónomo).
- Conocimiento funcional tácito (no formalizado en documentos) de personas con mayor experiencia previa en el sistema, que no aparezca documentado.

Ejemplos de fuentes de información que pueden formar parte de la test basis son:

- Especificaciones de requisitos
  - Requisitos de usuario
  - Requisitos de sistema
    - Funcionales
    - No funcionales (rendimiento, seguridad, volumen, back-up recuperación, etc)
- Requisitos de la arquitectura del sistema
- Documentos de diseño funcional
- Otros documentos referenciados en los propios documentos de requisitos o de diseño funcional.
- Manuales de usuario
- Diseños de pruebas realizados en proyectos anteriores.

### 4.4. Modelo

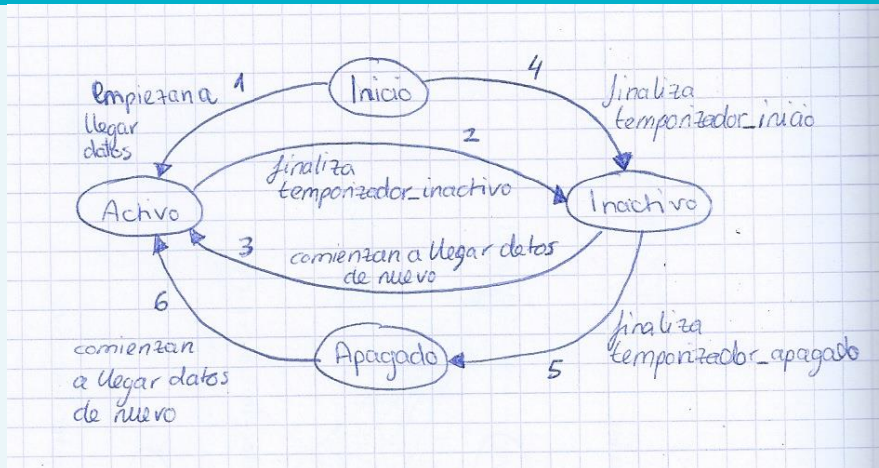
En el contexto de pruebas un *modelo* es una forma de representar la especificación que facilita la derivación de situaciones de prueba.

En algunos casos puede ocurrir que para un objetivo de prueba particular ya aparezca incluido algún modelo en la test basis, mientras que en otros casos puede ser necesario inferirlos a partir de la test basis, completándola, p.e. diagramas de transición de estados, tablas de decisión o condiciones complejas.

## Ejemplo 6

El siguiente diagrama de transición de estados modelaría los distintos estados que podría tener un proceso encargado de dar cierto tratamiento al tráfico de datos que recibe:

### MODELO A PARTIR DE LA ESPECIFICACIÓN DE UN PROCESO



**Figura 9.** Ejemplo de modelo: un diagrama de transición de estados

A partir de este diagrama se podría identificar las secuencias de estados que están permitidos y las que no.

## Ejemplo 7

En el siguiente gráfico se representa la evolución de una situación real que puede dar lugar a una alerta ATC SAME RUNWAY cuando una aeronave está autorizada para aterrizar y otra aeronave está autorizada para despegar sobre una misma pista.

Este modelo podría servir de ayuda para construir una tabla de decisión que represente la generación de la alerta en función del estado de los móviles involucrados.

### EXTRACTO DE LA ESPECIFICACIÓN



Ejemplo:

- Móvil 2: La aeronave está aterrizando (en fase ATERRIZAJE).
- Móvil 1:
  - o Caso (1): La aeronave está empezando a alinearse (fase TAXI o ALINEACION) => se generaría alerta.
  - o Caso (2): La aeronave está despegando y en la pista (fase DESPEGUE, dentro de la Runway)
  - o Caso (3): La aeronave está despegando y ha abandonado la pista (fase DESPEGUE, fuera de la Runway) => no se generaría la alerta.

**Figura 10.** Especificación parcial de alerta SAME RUNWAY

La tabla de decisión mencionada podría ser la siguiente:

Tabla decisión

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
móvil 2 en fase ATERRIZAJE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
móvil 1 en fase TAXI	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
móvil 1 en fase ALINEACIÓN	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
móvil 1 en fase DESPEGUE	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
móvil 1 dentro de la pista se genera alerta	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
																	X	X														X

Figura 11. Ejemplo de modelo: tabla de decisión

En esta tabla se combinan los valores de las condiciones que detectan el conflicto, identificando las combinaciones que detectan el conflicto. Cada combinación se correspondería con una situación de prueba. De todas las situaciones de prueba, las que generan alerta aparecen marcadas con una "x".

**Ejemplo 8**

En este caso se adjunta una representación de la topología de un aeródromo y se indican explícitamente tablas con datos para realizar pruebas de las alertas ATC del grupo OPPOSITE STOP POINTS.

Esta alerta se genera cuando ocurren determinadas combinaciones de autorizaciones sobre 2 móviles que implican entrar en una misma pista por holding points opuestos.

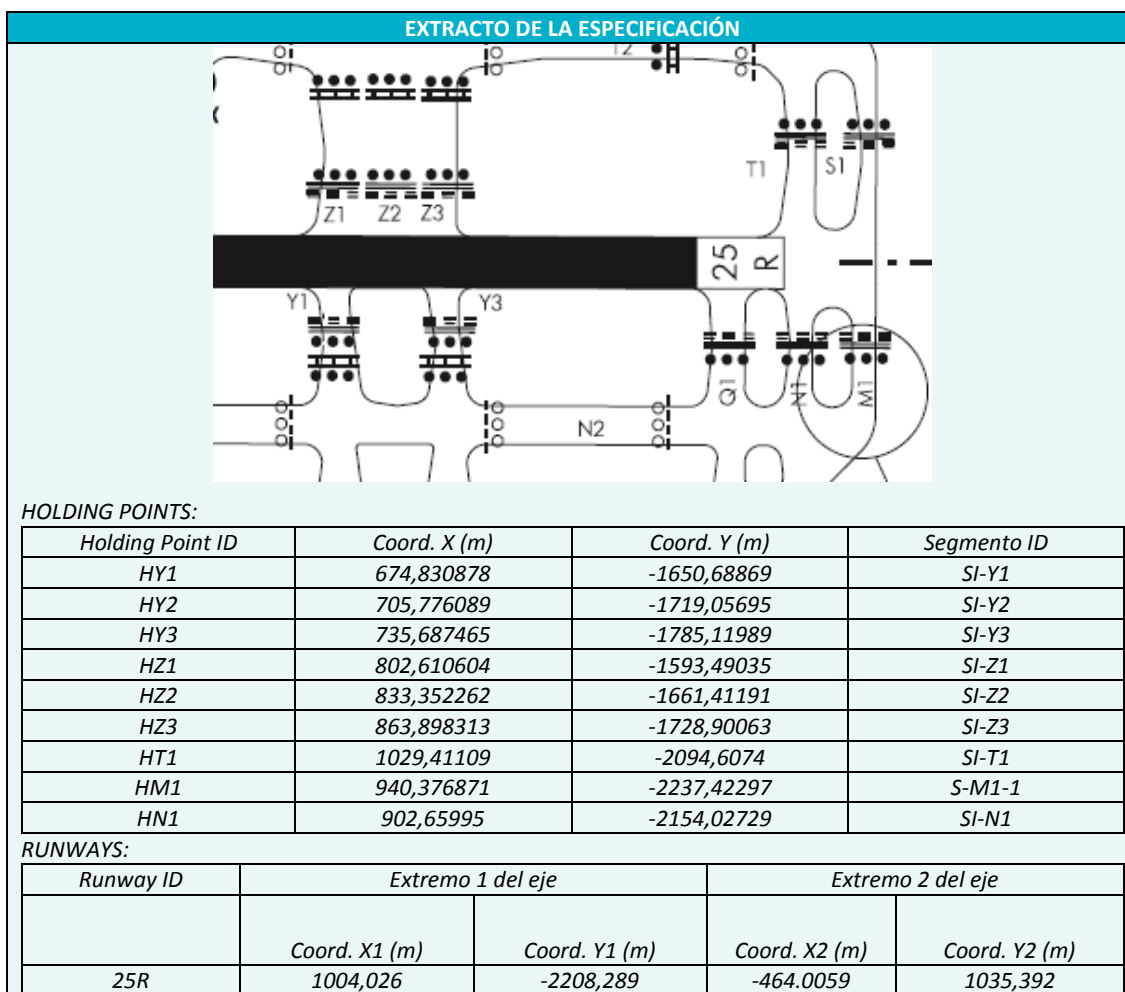


Figura 12. Ejemplo de modelo: esquema holding points en un aeródromo

El gráfico de la topología del aeródromo puede servir para derivar situaciones de prueba, como por ejemplo las situaciones que se pueden dar teniendo en cuenta la posición relativa de los segmentos donde están los holding points:

- Segmentos adyacentes (Z1, Z2 y Z3), p.e. uno de los móviles implicados está situado en el segmento correspondiente al holding point Z1 pero cruza al segmento del holding point Z3 y espera autorización en dicho punto
- Segmentos paralelos (Y1 e Y3), p.e. uno de los móviles implicados tiene asignado el holding point Y1 pero se detiene esperando autorización en holding point Y3

## 5. TÉCNICAS DE PRUEBA

Esta sección se estructura de la siguiente manera:

- Un primer apartado en donde se introduce un conjunto de técnicas de prueba básicas mediante pequeños ejemplos extraídos de especificaciones reales.
- A continuación, en un segundo apartado, se presenta cómo se aplicarían las técnicas anteriores en casos reales más complejos.
- En un último apartado se presentarían cómo es posible combinar algunas técnicas para derivar situaciones de prueba.

### 5.1. Introducción a las técnicas básicas

Una *técnica básica de pruebas* es un método estándar para derivar situaciones de prueba. Las características de la test basis determinan el conjunto de técnicas que es posible utilizar.

#### 5.1.1. Partición en clases de equivalencia

Consiste en particionar el rango de valores que admite un parámetro en clases en las que el comportamiento del sistema es equivalente.

Una situación de prueba será un valor concreto de una clase que será representativo de todos los valores de la clase. Un caso de prueba puede ejercitar una o varias situaciones de prueba, las cuales podrían corresponder a varias clases de equivalencia.

La partición en clases de equivalencia se puede aplicar en general a cualquier tipo de test basis, independientemente de cómo esté descrita.

Las clases de equivalencia se clasifican generalmente en válidas (valores de entrada que cumplen la especificación) e inválidas (entradas erróneas o no esperadas).

Existe un matiz importante a tener en cuenta para identificar clases de equivalencia inválidas: debe describir el conjunto de valores que constituye una entrada no esperada atendiendo al objetivo de la prueba que se ha fijado.

Por ejemplo, para un campo en el que un usuario introduce valores, cuando el objetivo de la prueba es:

- validar los valores introducidos, entonces clase inválida sería el conjunto de valores que no se pueden o que no se deberían introducir atendiendo al tipo o formato de datos que admite el campo de entrada.
- verificar una funcionalidad, entonces clase inválida sería el conjunto de valores que constituye una entrada no esperada atendiendo a la especificación de dicha funcionalidad.

Algunas pautas para identificar clases de equivalencia podrían ser las siguientes:

Descripción	Clases de equivalencia
<p>Condición de entrada que toma valor dentro de un rango</p> <p>Ejemplo: -100 &lt; X &lt; 100</p>	<p>Una clase válida para los valores contenidos en el rango: <math>X \in [-99, 99]</math></p> <p>Dos clases inválidas: <math>X \in [\text{MIN}, -100]</math> <math>X \in [100, \text{MAX}]</math> donde MIN y MAX se refieren respectivamente a los límites negativo y positivo del tipo de datos con el que se debe representar X</p>
<p>Condición de entrada que toma uno varios valores dentro de un conjunto de valores posibles</p> <p>Ejemplo: X puede ser TIPO2, TIPO3 o TIPO5</p>	<p>Una clase válida para los valores que tienen el mismo tratamiento: <math>X \in [\text{TIPO2}, \text{TIPO3}, \text{TIPO5}]</math></p> <p>Una clase inválida para el resto: <math>X \in [\text{TIPO1}, \text{TIPO4}]</math></p>
<p>Condición de entrada del tipo “debe ser/tener”</p> <p>Ejemplo: X debe ser de tipo real en doble precisión</p>	<p>Una clase válida: X es de tipo real en doble precisión</p> <p>Una clase inválida X no es de tipo real en doble precisión</p>

**Tabla 3.** Pautas para identificar clases de equivalencia

Lo más habitual al diseñar los casos de prueba es intentar cubrir, si es posible, el máximo número posible de clases válidas con cada caso. Para cada una de las clases inválidas podría llegar a ser necesario diseñar casos de prueba independientes. En cualquier caso, cada clase de equivalencia se debería ejercitar al menos una vez.

Si no se identificasen clases de equivalencia podría ocurrir que quedasen sin probar rangos de valores donde el sistema se comporta de una manera particular. En el peor caso podría ocurrir que además el diseño de pruebas sea ineficiente, por haber construido casos de prueba que ejercitan varios valores que pertenecerían a la misma clase de equivalencia, si esta se hubiese identificado.

### Ejemplo 9

A continuación se presenta un extracto de parte de la test basis de la alerta CMON NO CONTACT necesaria para probar la funcionalidad “detección del conflicto”.

#### EXTRACTO DE LA ESPECIFICACIÓN

##### Requisitos SESAR asociados

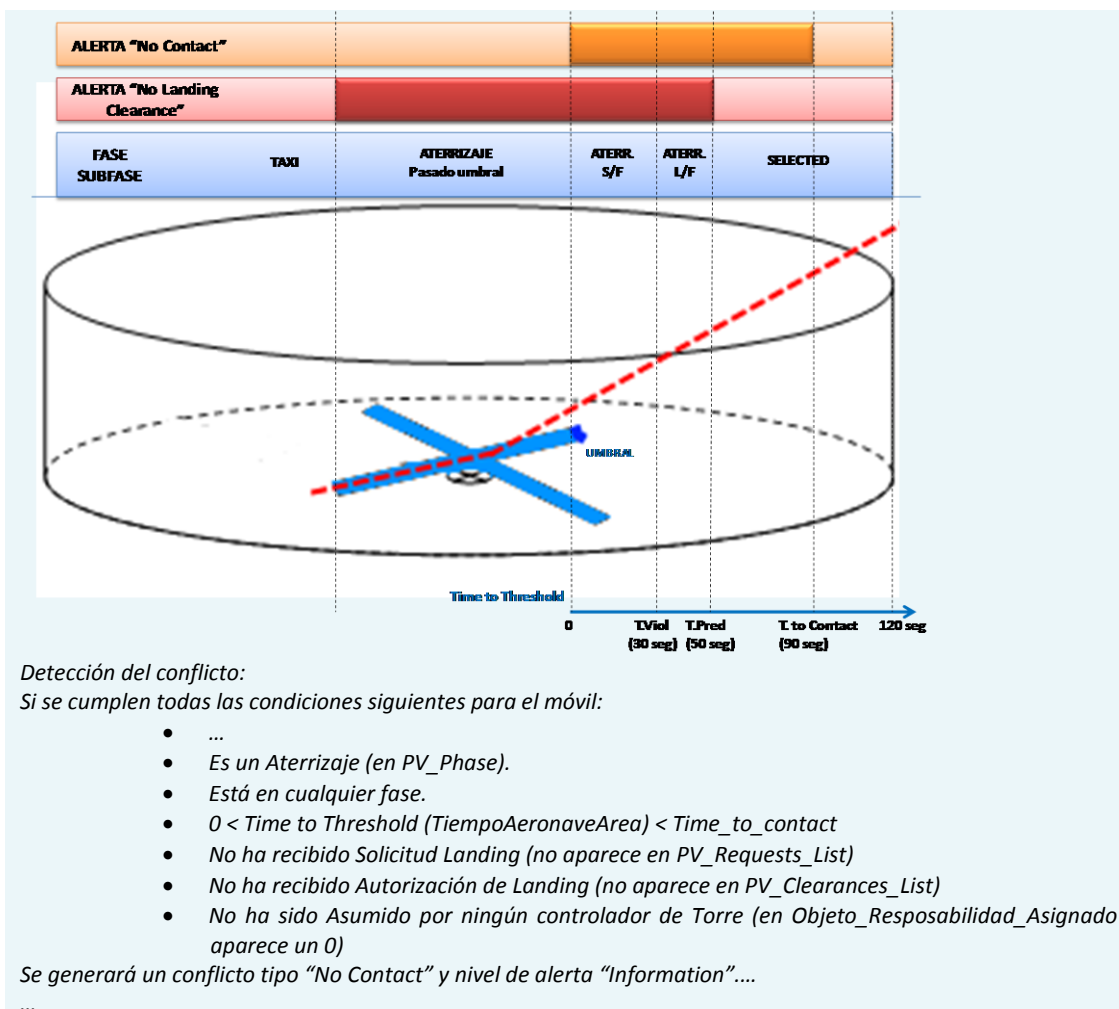
*The SSN server shall trigger an alert (INFORMATION) for an arriving aircraft, if there has been no "Contact indication", and the arriving aircraft is at less than A1 seconds from threshold.*

*Note: A1 should be configurable, depending upon optimisation at the aerodrome.*

##### Descripción extraída del diseño funcional

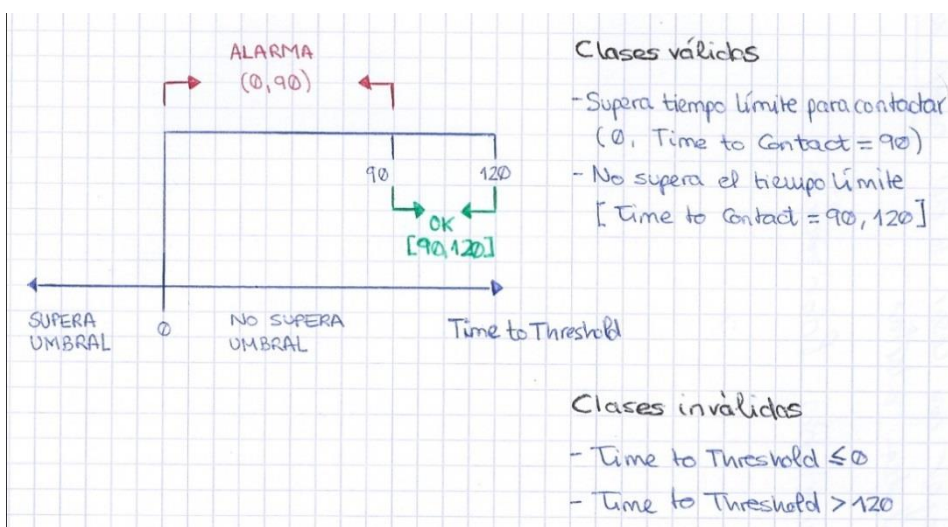
...

El atributo Time to Threshold solamente se calcula cuando la aeronave se dirige a un área, por tanto no se calcula (se establece el valor de inicialización, -1) si no está alineándose para aterrizar en un área concreta o si ya ha superado el umbral (ver función Filtros::calcularAreaSeDirige)



**Figura 13.** Especificación de alerta NO CONTACT

En la detección del conflicto el "tiempo restante hasta alcanzar el umbral (*Time to threshold*) es un parámetro que admite valores comprendidos en el rango (0, L], donde se supone que por defecto L=120 (es un valor configurable). Si se observa cómo se comporta el sistema en función del valor de este parámetro se puede observar lo siguiente:



**Figura 14.** Situaciones de prueba derivadas mediante clases de equivalencia



Las dos clases válidas, *supera tiempo límite para contactar* y *no supera el tiempo límite* corresponden a los valores que cumplen la especificación.

La clase inválida *Time to Threshold  $\leq 0$*  se deriva teniendo en cuenta la descripción dada en la test basis:

**EXTRACTO DE LA ESPECIFICACIÓN**

...

*El atributo Time to Threshold solamente se calcula cuando la aeronave se dirige a un área, por tanto no se calcula (se establece el valor de inicialización, -1) si no está alineándose para aterrizar en un área concreta o si ya ha superado el umbral (ver función Filtros::calcularAreaSeDirige)*

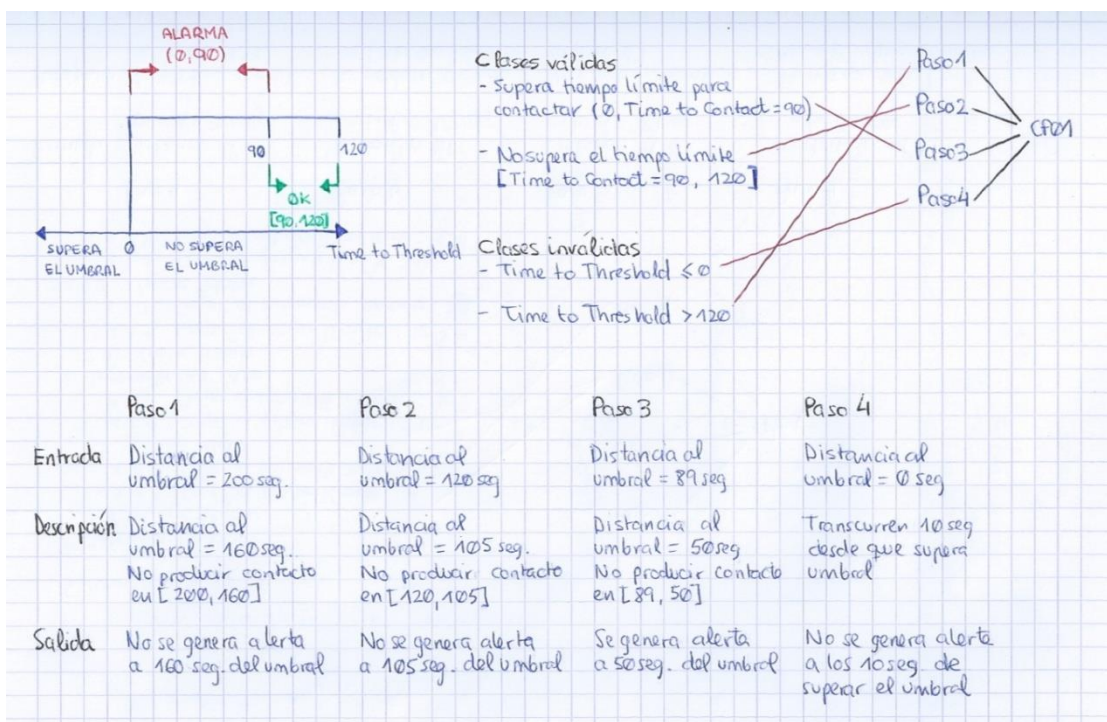
...

**Figura 15.** Detalle atributo Time to Threshold

La otra clase inválida, *Time to Threshold  $\geq 120$* , se deriva teniendo en cuenta que en el modelo proporcionado en la test basis esa situación no es esperada.

Otra condición que interviene en la detección del conflicto y que es interesante observar es la *fase en que se encuentra el vuelo*. Si prestamos atención a la forma en que está descrita esa condición para la detección del conflicto, al leer “*Está en cualquier fase*” se podría pensar en que son admisibles todas las fases en que se puede encontrar un vuelo de cara a identificar clases válidas, pero esto sería incorrecto.

Es importante tener en cuenta el contexto de la frase, ya que tenemos otra condición que dice “*Es un Aterrizaje (en PV\_Phase)*” lo cual restringe las fases admisibles, dado que hay fases que serán específicas de despegues y que por tanto no haya que tener en cuenta aquí. Se podrían diseñar el siguiente caso de prueba para cubrir todas las situaciones derivadas aplicando clases de equivalencia:



**Figura 16.** Diseño de casos de prueba alerta NO CONTACT

### 5.1.2. Valores límite

El objetivo de la técnica es probar el comportamiento del sistema cuando el valor que toman ciertos parámetros traspasa un determinado límite. El modo de proceder para aplicar esta técnica consiste básicamente en:

1. Determinar los límites o fronteras relativos a clases de equivalencia o condiciones
2. Definir las situaciones de prueba siguientes:
  - a. Exactamente en el límite
  - b. El valor adyacente "externo" al límite (según la precisión del tipo de datos admitida)
  - c. Opcionalmente, incluir valores muy altos (límites según la precisión del tipo de datos admitida) y otros puntos singulares como el 0

La técnica de valores límite se puede aplicar en general a cualquier tipo de test basis, independientemente de cómo esté descrita y normalmente se utiliza sobre parámetros que admiten valores continuos (p.e. tiempo, velocidad, altitud, etc).

### Ejemplo 10

Continuando con el Ejemplo 9 anterior, se podría aplicar valores límite sobre el "tiempo restante hasta alcanzar el umbral".

Si la precisión del parámetro Time to Threshold es de segundos (tener en cuenta que podrían darse otros escenarios en donde los parámetros sobre los que se aplica valores límite admiten otras precisiones, p.e. milésimas de segundo):

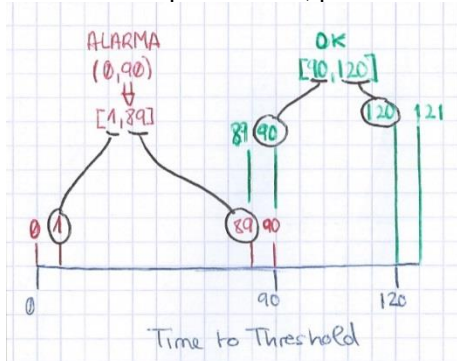


Figura 17. Valores límite sobre atributo Time to Threshold

Atendiendo a esto último se podrían diseñar las siguientes situaciones:

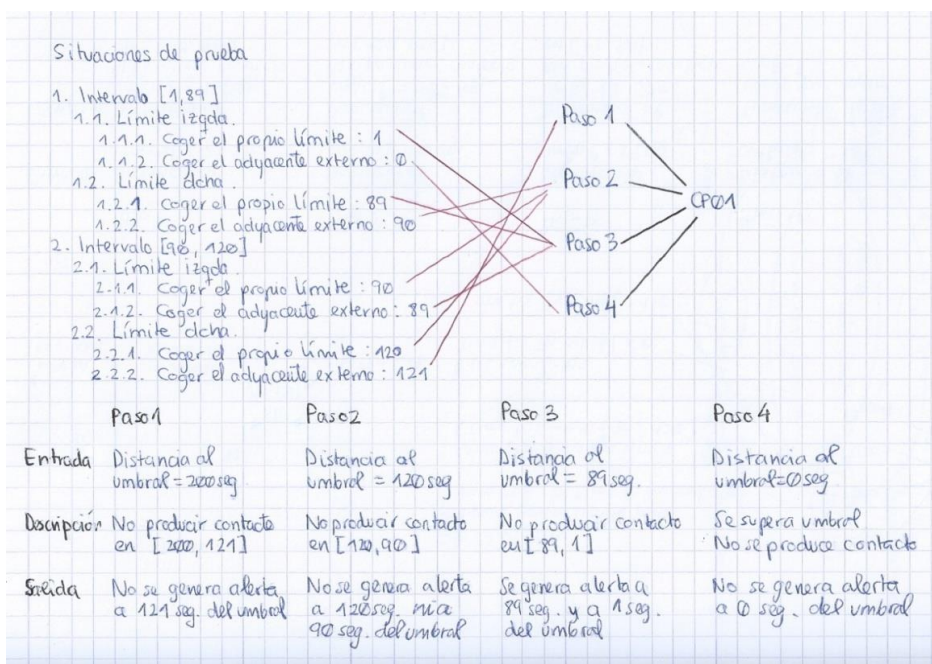


Figura 18. Diseño de pruebas aplicando valores límite

Se puede comprobar que con 1 caso de prueba sería posible ejercitar todas las situaciones de prueba descritas.

### 5.1.3. Cobertura de puntos de decisión

Consiste en cubrir las distintas alternativas en un punto de decisión con el objetivo de obtener las salidas TRUE y FALSE. La test basis necesaria consistirá en un conjunto de condiciones descritas formalmente y combinadas mediante los operadores lógicos AND, OR y NOT.

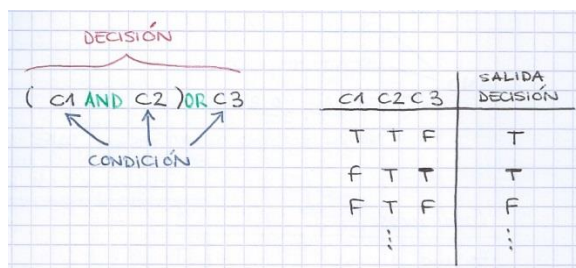


Figura 19. Estructura de un punto de decisión

Los tipos de cobertura de puntos de decisión más utilizados son los siguientes:

Variante	Descripción
<b>condiciones</b>	Las salidas posibles de cada condición (TRUE y FALSE), se prueban al menos una vez.
<b>decisiones</b>	Las salidas posibles de la decisión se prueban al menos una vez.
<b>condición/decisión (C/D)</b>	Las salidas posibles de cada condición y de la decisión se prueban al menos una vez.  Implica las dos anteriores.
<b>múltiple condición (MC)</b>	Se prueban al menos una vez todas las combinaciones posibles de salidas de todas las condiciones de la decisión.  Implica cobertura de condición/decisión modificada.  Se obtienen $2^N$ situaciones de prueba, donde N es el número de condiciones que intervienen en la decisión.
<b>condición/decisión modificada (MCDC)</b>	Al variar la salida de cada condición sin cambiar el resto de condiciones, la salida de la decisión cambia.  Implica cobertura de condición/decisión.  En general, se obtienen N+1 situaciones de prueba, donde N es el número de condiciones que intervienen en la decisión

Tabla 4. Variantes de la técnica de cobertura de puntos de decisión

Las variantes indicadas en la tabla anterior están ordenadas de forma creciente en cuanto al número de situaciones de prueba que permiten derivar y la probabilidad de encontrar fallos. Según esto, la cobertura de condiciones es la más básica que se podría aplicar, mientras que la cobertura de múltiple condición sería la más completa.

#### Ejemplo 11

A continuación se presenta un extracto de la especificación para la alerta CMON NO TAKE-OFF CLEARANCE, que servirá de base para la explicación de las distintas variantes de cobertura de puntos de decisión.

**Requisitos SESAR asociados**

The SSN server shall generate an alert (ALARM), if an aircraft is taking off without having received a Take-Off clearance.

**Descripción extraída del diseño funcional**

...

Detección del conflicto:

Si se cumplen todas las condiciones siguientes para el móvil:

- Es una Aeronave
- Es un Despegue (en PV\_Phase)
- Está en fase DESPEGUE
- No ha recibido Autorización Take-off

Se generará un conflicto tipo "No Take-off Clearance" y nivel de alerta "Alarm".

...

**Figura 20.** Especificación de alerta NO TAKE OFF CLEARANCE

En los siguientes apartados se aplican las distintas variantes.

**5.1.3.1. Cobertura de condiciones**

**Definición:** las salidas posibles de cada condición (TRUE y FALSE), se prueban al menos una vez.

Se podría cubrir con las siguientes situaciones:

Cobertura	Condiciones	
	1	2
Es una aeronave	T	F
Es un despegue	T	F
Está en fase DESPEGUE	T	F
No ha recibido aut. TAKE-OFF	T	F

**Figura 21.** Cobertura de condiciones

La anterior es una forma de conseguir que cada condición tome los valores TRUE y FALSE al menos una vez, y se consigue con 2 situaciones de prueba.

Pueden existir otras combinaciones que permitan conseguir que cada condición tome los valores TRUE y FALSE al menos una vez. A continuación se presenta una de ellas que da lugar a situaciones que no son factibles (situaciones para las que no es posible ejecutar pruebas atendiendo a la especificación):

Cobertura	Condiciones	
	1	2
Es una aeronave	T	F
Es un despegue	T	F
Está en fase DESPEGUE	F	T
No ha recibido aut. TAKE-OFF	F	T

**Figura 22.** Cobertura de condiciones (con situaciones no factibles)

Se puede observar que en la Figura 22 la situación de prueba 2 no es factible (un vehículo no puede cumplir estar en fase de desapegue o recibir autorización de take off).

Para ejercitar las 2 situaciones derivadas en la Figura 21 es necesario diseñar 2 casos de prueba, uno para cubrir cada una situación:

- CP01: aeronave despegando sin autorización take off.
- CP02: escenario en donde un vehículo (no aeronave) se desplaza por las pistas de rodadura del aeródromo hasta un holding point de entrada a una pista.

CP01	Paso 1	Paso 2	Paso 3	Paso 4
Entrada	Aeronave parada en stand	Aeronave parada en calle rodadura	Aeronave parada en holding point	Aeronave parada en pto. espera
Descripción	Autorización = push-back Aeronave se mueve hasta calle rodadura	Autorización = taxi Aeronave se mueve hasta holding point	Autorización = line-up Aeronave se mueve hasta pto. espera aut. take-off	Autorización = ninguna Aeronave acelera hasta vel > vel-despegue
Salida	Aeronave parada en calle rodadura	Aeronave parada en holding point	Aeronave parada en pto. espera	Cuando se pasa a fase DESPEGUE se genera a la hora NO TAKE OFF CLEARANCE

**Figura 23.** Detalle de CP01 de alerta NO TAKE-OFF CLEARANCE

### 5.1.3.2. Cobertura de decisiones

Definición: las salidas posibles de la decisión se prueban al menos una vez.

Se podría cubrir con las siguientes situaciones (en la figura se compara con la variante anterior):

Variantes anteriores	Cobertura de decisiones																																	
<p>Cobertura Condiciones</p> <table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> </tr> </thead> <tbody> <tr> <td>Es una aeronave</td> <td>T</td> <td>F</td> </tr> <tr> <td>Es un despegue</td> <td>T</td> <td>F</td> </tr> <tr> <td>Está en fase DESPEGUE</td> <td>T</td> <td>F</td> </tr> <tr> <td>No ha recibido aut. TAKE-OFF</td> <td>T</td> <td>F</td> </tr> </tbody> </table>		1	2	Es una aeronave	T	F	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F	<p>Cobertura Decisiones</p> <table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> </tr> </thead> <tbody> <tr> <td>Es una aeronave</td> <td>T</td> <td>T</td> </tr> <tr> <td>Es un despegue</td> <td>T</td> <td>F</td> </tr> <tr> <td>Está en fase DESPEGUE</td> <td>T</td> <td>F</td> </tr> <tr> <td>No ha recibido aut. TAKE-OFF</td> <td>T</td> <td>F</td> </tr> <tr> <td>SALIDA</td> <td>T</td> <td>F</td> </tr> </tbody> </table>		1	2	Es una aeronave	T	T	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F	SALIDA	T	F
	1	2																																
Es una aeronave	T	F																																
Es un despegue	T	F																																
Está en fase DESPEGUE	T	F																																
No ha recibido aut. TAKE-OFF	T	F																																
	1	2																																
Es una aeronave	T	T																																
Es un despegue	T	F																																
Está en fase DESPEGUE	T	F																																
No ha recibido aut. TAKE-OFF	T	F																																
SALIDA	T	F																																

**Tabla 5.** Cobertura de decisiones

Con estas 2 situaciones de prueba se consigue que la decisión tenga las salidas TRUE y FALSE al menos una vez, atendiendo a cómo se combinan las condiciones mediante el operador AND.

Se puede observar que sería necesario definir 2 casos de prueba para ejercitar las 2 situaciones de prueba:

- El CP01 sería el mismo caso de prueba que ejercitaba la cobertura de condiciones.
- El CP02 podría ser, por ejemplo, un aterrizaje (según su PV) en fase de ATERRIZAJE y que ha recibido autorización LANDING.

### 5.1.3.3. Cobertura de condición/decisión (C/D)

Definición: las salidas posibles de cada condición y de la decisión se prueban al menos una vez.

En este caso se podría cubrir con las mismas situaciones de prueba derivadas para el ejemplo anterior de cobertura de condiciones (en la figura se compara con las variantes anteriores):

Variantes anteriores	Cobertura Condición/Decisión																																																			
<p>Cobertura Condiciones</p> <table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>Es una aeronave</td><td>T</td><td>F</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> </table> <p>Cobertura Decisiones</p> <table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>Es una aeronave</td><td>T</td><td>T</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> <tr><td>SALIDA</td><td>T</td><td>F</td></tr> </table>		1	2	Es una aeronave	T	F	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F		1	2	Es una aeronave	T	T	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F	SALIDA	T	F	<p>Cobertura Condición / Decisión</p> <table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>Es una aeronave</td><td>T</td><td>F</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> <tr><td>SALIDA</td><td>T</td><td>F</td></tr> </table>		1	2	Es una aeronave	T	F	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F	SALIDA	T	F
	1	2																																																		
Es una aeronave	T	F																																																		
Es un despegue	T	F																																																		
Está en fase DESPEGUE	T	F																																																		
No ha recibido aut. TAKE-OFF	T	F																																																		
	1	2																																																		
Es una aeronave	T	T																																																		
Es un despegue	T	F																																																		
Está en fase DESPEGUE	T	F																																																		
No ha recibido aut. TAKE-OFF	T	F																																																		
SALIDA	T	F																																																		
	1	2																																																		
Es una aeronave	T	F																																																		
Es un despegue	T	F																																																		
Está en fase DESPEGUE	T	F																																																		
No ha recibido aut. TAKE-OFF	T	F																																																		
SALIDA	T	F																																																		

**Tabla 6.** Cobertura de condición/decisión

Con las 2 situaciones de prueba indicadas se cumple simultáneamente la cobertura de condiciones y de decisiones.

Los casos de prueba necesarios para ejercitar la cobertura de condición/decisión podrían ser, en este caso, los mismos que permitían ejercitar la cobertura de condiciones.

### 5.1.3.4. Cobertura de múltiple condición (MC)

**Definición:** se prueban al menos una vez todas las combinaciones posibles de salidas de todas las condiciones de la decisión.

Se podría cubrir con las siguientes situaciones (en la figura se compara con las variantes anteriores):

Variantes anteriores	Cobertura de múltiple condición																																																																																																																																																									
<p>Cobertura Condiciones</p> <table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>Es una aeronave</td><td>T</td><td>F</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> </table> <p>Cobertura Decisiones</p> <table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>Es una aeronave</td><td>T</td><td>T</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> <tr><td>SALIDA</td><td>T</td><td>F</td></tr> </table>		1	2	Es una aeronave	T	F	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F		1	2	Es una aeronave	T	T	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F	SALIDA	T	F	<p>Cobertura Mc</p> <table border="1"> <tr><td></td><td colspan="16">Situaciones</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr> <tr><td>Es una aeronave</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>Es un despegue</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>Está en fase DESPEGUE</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> </table> <p>Cobertura Condición / Decisión</p> <table border="1"> <tr><td></td><td>1</td><td>2</td></tr> <tr><td>Es una aeronave</td><td>T</td><td>F</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> <tr><td>SALIDA</td><td>T</td><td>F</td></tr> </table>		Situaciones																	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Es una aeronave	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	Es un despegue	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	Está en fase DESPEGUE	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	No ha recibido aut. TAKE-OFF	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0		1	2	Es una aeronave	T	F	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F	SALIDA	T	F
	1	2																																																																																																																																																								
Es una aeronave	T	F																																																																																																																																																								
Es un despegue	T	F																																																																																																																																																								
Está en fase DESPEGUE	T	F																																																																																																																																																								
No ha recibido aut. TAKE-OFF	T	F																																																																																																																																																								
	1	2																																																																																																																																																								
Es una aeronave	T	T																																																																																																																																																								
Es un despegue	T	F																																																																																																																																																								
Está en fase DESPEGUE	T	F																																																																																																																																																								
No ha recibido aut. TAKE-OFF	T	F																																																																																																																																																								
SALIDA	T	F																																																																																																																																																								
	Situaciones																																																																																																																																																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16																																																																																																																																										
Es una aeronave	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1																																																																																																																																										
Es un despegue	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0																																																																																																																																										
Está en fase DESPEGUE	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0																																																																																																																																										
No ha recibido aut. TAKE-OFF	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0																																																																																																																																										
	1	2																																																																																																																																																								
Es una aeronave	T	F																																																																																																																																																								
Es un despegue	T	F																																																																																																																																																								
Está en fase DESPEGUE	T	F																																																																																																																																																								
No ha recibido aut. TAKE-OFF	T	F																																																																																																																																																								
SALIDA	T	F																																																																																																																																																								

**Tabla 7.** Cobertura de múltiple condición

Como se puede observar, se derivan 16 situaciones de prueba ( $2^N$ , siendo  $N=4$  condiciones). Del total de situaciones derivadas las siguientes no serían factibles:

- situación de prueba 1, debido a que la autorización take-off solo se puede dar a aeronaves.
- situaciones 3 y 4, debido a que no es posible que un móvil que no sea aeronave esté en fase DESPEGUE.
- situaciones 5, 6, 7 y 8, debido a que no es posible que un móvil que no sea una aeronave se trate de un despegue.

Atendiendo a esto último se podrían diseñar los siguientes casos de prueba:

- CP01, que ejercitará la situación de prueba 2: se podría reutilizar el CP02 de cobertura de condiciones (vehículo que se desplaza por las pistas de rodadura del aeródromo hasta un holding point de entrada a una pista).
- CP02, que ejercitará las situaciones 9, 10, 11 y 12: se podría reutilizar el CP01 de MCDC.
- CP03, que ejercitará las situaciones 13, 14, 15 y 16: se podría reutilizar el CP02 de cobertura de decisiones (aterrizaje en fase de ATERRIZAJE y que ha recibido autorización LANDING).

### 5.1.3.5. Cobertura de condición/decisión modificada (MCDC)

**Definición:** el objetivo es probar la sensibilidad que tiene cada condición sobre la salida que toma la decisión. Para abordar esto es necesario ir dando valores a cada condición manteniendo en el resto de condiciones un *valor neutro*.

A continuación se explica el concepto de valor neutro mediante un ejemplo:

Se tiene la condición compuesta  $R = A \text{ AND } B$ . Suponiendo que se quiere probar cómo A determina el valor de R, el valor neutro de B sería TRUE, debido a que:

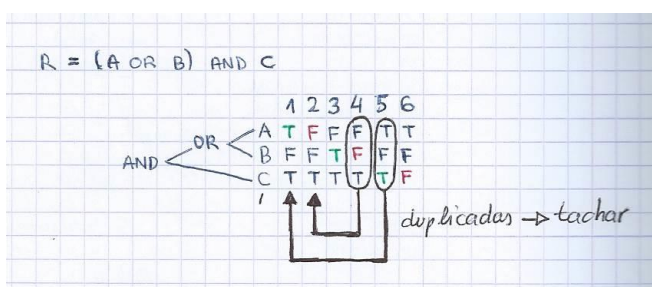
- $A \text{ AND } \text{TRUE} = A$
- $A \text{ AND } \text{FALSE} = \text{FALSE}$ , es decir, si B fuese TRUE el valor de R dependería por completo del valor de A, que es lo que se pretende conseguir con MCDC

De forma análoga, si la condición fuese  $R = A \text{ OR } B$ , valor neutro sería FALSE, dado que  $A \text{ OR } \text{FALSE} = A$ .

Las situaciones de prueba a cubrir serían entonces las siguientes:

- Cada condición toma valor TRUE manteniendo el resto a TRUE, que es el valor neutro atendiendo a cómo se combinan las condiciones en la decisión con AND
- Cada condición toma valor FALSE manteniendo el resto a TRUE

A continuación se explica cómo aplicar MCDC sobre un ejemplo genérico. Resulta muy práctico representar las condiciones mediante un grafo, ya que facilita el proceso de rellenar valores neutros.



**Figura 24.** Cobertura de condición/decisión modificada

Se comienza probando la sensibilidad de la condición A (resaltado en color rojo) sobre la salida de la decisión R. Dado que B está conectado con A mediante el operador OR, su valor neutro es FALSE (resaltado en color verde). En cambio C está conectado con A mediante el operador AND, así que su valor neutro es TRUE (resaltado también en color verde). Esto da lugar a las situaciones de prueba 1 y 2.

A medida que se rellena el grafo es útil ir anotando las situaciones de prueba derivadas hasta el momento. Para la condición B el proceso sería similar, dando lugar a las situaciones de prueba 3 y 4.

Sin embargo, cuando se prueba la sensibilidad de la condición C es necesario tener en cuenta que las condiciones A y B no están conectadas con C directamente. Lo que conecta con C es la combinación (A OR B) mediante el operador AND, de manera que (A OR B) debe asumir el valor neutro TRUE.

Para que (A OR B) tome el valor neutro TRUE hay 3 combinaciones posibles:

- A=TRUE y B=TRUE
- A=TRUE y B=FALSE
- A=FALSE y B=TRUE

Solo se necesita una de las 3, la única diferencia es que si se escoge TRUE-FALSE o FALSE-TRUE se obtendrá una situación de prueba duplicada, lo cual no ocurre si se escoge TRUE-TRUE. En este caso se escogió A=TRUE y B=FALSE, dando lugar a las situaciones 5 y 6.

Se puede observar que, sin tener en cuenta duplicados, se han generado 4 situaciones de prueba (N+1, donde N=3 de condiciones).

Continuando con el Ejemplo 5, al aplicar MCDC se obtendrían las siguientes situaciones de prueba (en la figura se compara con las variantes anteriores):

Variantes anteriores	MCDC																																																																																																																																																																										
<p><b>Cobertura Condiciones</b></p> <table border="1"> <tr><td>Es una aeronave</td><td>1</td><td>2</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> </table> <p><b>Cobertura Decisiones</b></p> <table border="1"> <tr><td>Es una aeronave</td><td>1</td><td>2</td></tr> <tr><td>Es un despegue</td><td>T</td><td>T</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> </table> <p><b>SALIDA</b></p> <table border="1"> <tr><td></td><td>T</td><td>F</td></tr> </table> <p><b>Cobertura Condición/Decisión</b></p> <table border="1"> <tr><td>Es una aeronave</td><td>1</td><td>2</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>F</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>F</td></tr> </table> <p><b>SALIDA</b></p> <table border="1"> <tr><td></td><td>T</td><td>F</td></tr> </table> <p><b>Cobertura MC</b></p> <table border="1"> <tr><td>Es una aeronave</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr> <tr><td>Es un despegue</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>Está en fase DESPEGUE</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> </table>	Es una aeronave	1	2	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F	Es una aeronave	1	2	Es un despegue	T	T	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F		T	F	Es una aeronave	1	2	Es un despegue	T	F	Está en fase DESPEGUE	T	F	No ha recibido aut. TAKE-OFF	T	F		T	F	Es una aeronave	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Es un despegue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Está en fase DESPEGUE	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	No ha recibido aut. TAKE-OFF	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	<p><b>Cobertura MCDC</b></p> <table border="1"> <tr><td>Es una aeronave</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>T</td><td>F</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>F</td></tr> </table> <p><i>duplicadas → tachar</i></p> <p><b>Cobertura MCDC</b></p> <table border="1"> <tr><td>Es una aeronave</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>Es un despegue</td><td>T</td><td>F</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Está en fase DESPEGUE</td><td>T</td><td>T</td><td>F</td><td>T</td><td>T</td></tr> <tr><td>No ha recibido aut. TAKE-OFF</td><td>T</td><td>T</td><td>T</td><td>F</td><td>T</td></tr> </table> <p><i>reaprovechando la 1</i></p>	Es una aeronave	1	2	3	4	5	6	7	8	Es un despegue	T	F	T	T	T	T	T	T	Está en fase DESPEGUE	T	T	F	T	T	T	T	T	No ha recibido aut. TAKE-OFF	T	T	T	T	T	T	T	F	Es una aeronave	1	2	3	4	5	Es un despegue	T	F	T	T	T	Está en fase DESPEGUE	T	T	F	T	T	No ha recibido aut. TAKE-OFF	T	T	T	F	T
Es una aeronave	1	2																																																																																																																																																																									
Es un despegue	T	F																																																																																																																																																																									
Está en fase DESPEGUE	T	F																																																																																																																																																																									
No ha recibido aut. TAKE-OFF	T	F																																																																																																																																																																									
Es una aeronave	1	2																																																																																																																																																																									
Es un despegue	T	T																																																																																																																																																																									
Está en fase DESPEGUE	T	F																																																																																																																																																																									
No ha recibido aut. TAKE-OFF	T	F																																																																																																																																																																									
	T	F																																																																																																																																																																									
Es una aeronave	1	2																																																																																																																																																																									
Es un despegue	T	F																																																																																																																																																																									
Está en fase DESPEGUE	T	F																																																																																																																																																																									
No ha recibido aut. TAKE-OFF	T	F																																																																																																																																																																									
	T	F																																																																																																																																																																									
Es una aeronave	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16																																																																																																																																																											
Es un despegue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																											
Está en fase DESPEGUE	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0																																																																																																																																																											
No ha recibido aut. TAKE-OFF	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0																																																																																																																																																											
Es una aeronave	1	2	3	4	5	6	7	8																																																																																																																																																																			
Es un despegue	T	F	T	T	T	T	T	T																																																																																																																																																																			
Está en fase DESPEGUE	T	T	F	T	T	T	T	T																																																																																																																																																																			
No ha recibido aut. TAKE-OFF	T	T	T	T	T	T	T	F																																																																																																																																																																			
Es una aeronave	1	2	3	4	5																																																																																																																																																																						
Es un despegue	T	F	T	T	T																																																																																																																																																																						
Está en fase DESPEGUE	T	T	F	T	T																																																																																																																																																																						
No ha recibido aut. TAKE-OFF	T	T	T	F	T																																																																																																																																																																						

Tabla 8. MCDC en comparación con resto de variantes



Como se puede observar, tras eliminar duplicados se obtienen 5 situaciones de prueba (N+1, siendo N=4 condiciones). Sería necesario un caso de prueba para cubrir cada de las situaciones de prueba.

Una forma de evitar situaciones de prueba duplicadas consiste en revisar las situaciones de prueba anteriores al derivar cada nueva situación de prueba, reutilizando las que permitan variar la condición manteniendo fijo el resto. En este ejemplo la situación de prueba 1 se puede reutilizar para demostrar la sensibilidad de las condiciones *Es un despegue* (situaciones 1 y 3), *Está en fase DESPEGUE* (situaciones 1 y 4) y *No ha recibido aut. TAKE-OFF* (situaciones 1 y 5).

Atendiendo a las situaciones de prueba derivadas se podrían diseñar los siguientes casos de prueba:

- CP01, con el que sería posible ejercitar las situaciones 1, 4 y 5.
- CP02, para ejercitar la situación de prueba 3 (el vuelo no es un despegue). Este caso de prueba sería prácticamente igual al CP01 descrito para ejercitar cobertura de condiciones, salvo que en el PV el vuelo figuraría como "local" en lugar de como "despegue".
- La situación de prueba 2 no es factible (no es posible que un móvil que no sea una aeronave se trate de un despegue), así que no es posible ejecutar un caso que la ejercite (no es posible que un móvil que no sea una aeronave sea un despegue).

CP01	Paso 1	Paso 2	Paso 3	Paso 4
Entrada	Aeronave parada en stand	Aeronave parada en calle rodadura	Aeronave parada en holding point	Aeronave parada en pt. espera
Descripción	Autorización = push-back Aeronave se mueve hasta calle rodadura	Autorización = taxi Aeronave se mueve hasta holding point	Autorización = line-up Aeronave se mueve hasta pt. espera con take-off	Autorización = take-off Aeronave acelera hasta vel > vel. despegue
Salida	Aeronave parada en calle rodadura	Aeronave parada en holding point	Aeronave parada en pt. espera	Se entra en fase DESPEGUE
	Paso 5			
Entrada	Aeronave en fase DESPEGUE (siguen tierra)			
Descripción	Autorización = ninguna Reovar aut. take-off			
Salida	No genera alertas tras reovar aut.			

Figura 25. Detalle de CP01 diseñado aplicando MDCD

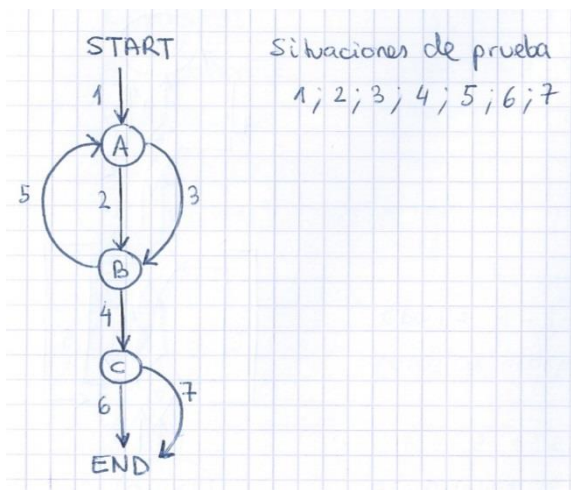
### 5.1.4. Cobertura de caminos

La cobertura de caminos es aplicable si el comportamiento del sistema está descrito mediante puntos de decisión y caminos (*paths*) o mediante alguna representación similar que describa un flujo en el que es posible seguir caminos diferentes para llegar de un estado inicial a otro final.

Las situaciones de prueba se describen mediante combinaciones de caminos que se deben seguir de forma consecutiva.

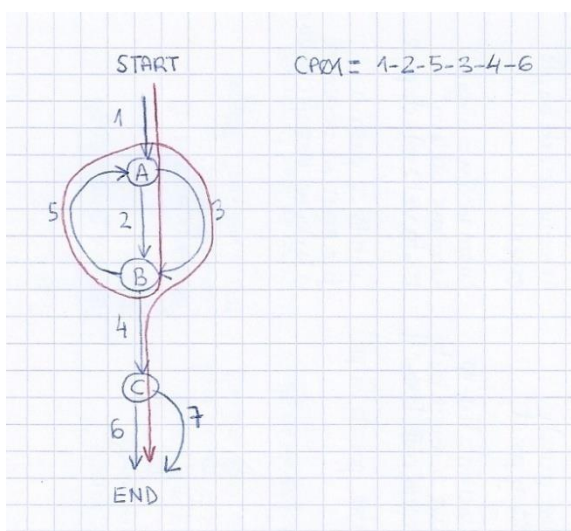
#### 5.1.4.1. Concepto de nivel de cobertura

El nivel más básico consiste en aquel en donde cada situación de prueba consiste en cada camino considerado de forma individual (N=1).



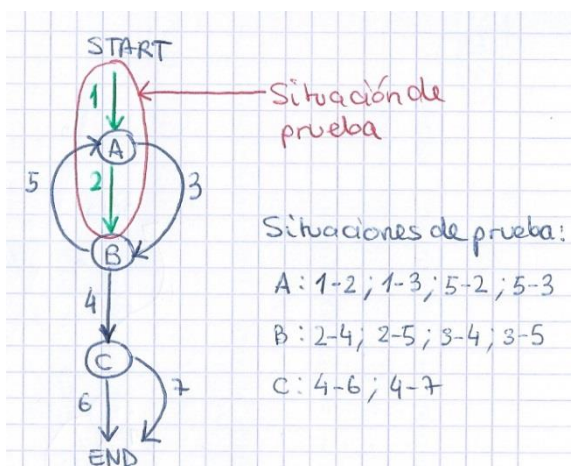
**Figura 26.** Situaciones de prueba para el nivel 1

Un caso de prueba consiste en la secuencia de combinaciones de caminos que va desde el principio hasta el final.



**Figura 27.** Detalle de CP01 correspondiente al nivel 1

En el siguiente nivel (N=2), se cubrirían secuencias de combinaciones de dos caminos consecutivos. En general, a mayor nivel mayor probabilidad de encontrar fallos más difíciles de detectar.



**Figura 28.** Situaciones de prueba para el nivel 2

### 5.1.4.2. Pasos para obtener nivel 2

Serían los siguientes:

**1) Puntos de decisión y caminos**

Identificar de manera única cada punto de decisión (A, B, etc) y numerar los caminos. Para cada punto de decisión agrupar caminos entrantes y salientes.

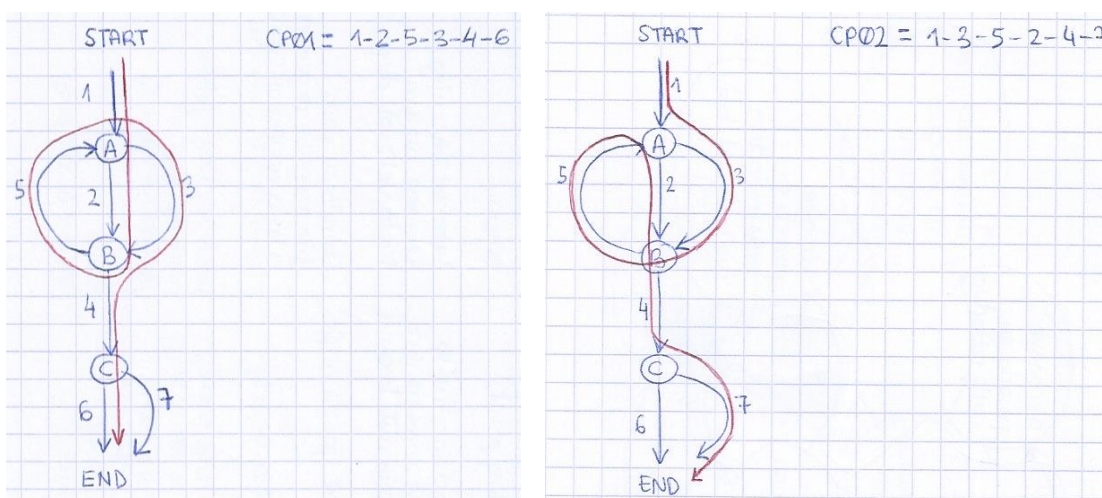
**2) Combinaciones de caminos**

Realizar todas las combinaciones de caminos entrantes y salientes en cada punto de decisión. Si hay P caminos entrantes y Q caminos salientes, se obtienen P x Q combinaciones.

**3) Casos de prueba**

Crear los casos de prueba hasta que se cubran todas las combinaciones de caminos.

En el siguiente ejemplo se cubren todas las combinaciones (situaciones de prueba) con 2 casos de prueba:



**Figura 29.** Detalle CP01 y CP02 para nivel 2 de cobertura de caminos

Para mantener la trazabilidad entre los casos de prueba y las situaciones de prueba que se ejercitan se puede optar por:

- La manera más sencilla, que consiste en ir tachando las combinaciones a medida que se van ejercitando con cada caso de prueba
- Utilizar como apoyo una matriz de trazabilidad, marcando para cada caso de prueba diseñado las situaciones de prueba que se cubren, de manera que en los sucesivos casos de prueba se busque cubrir las situaciones que faltan.

	CP01	CP02
1-2	X	
1-3		X
5-2		X
5-3	X	
2-4		X
2-5	X	
3-4	X	
3-5		X
4-6	X	
4-7		X

**Figura 30.** Trazabilidad CP01 y CP02 para nivel 2

A continuación se presenta un extracto de la especificación para la alerta CMON NO PUSH-BACK CLEARANCE.

#### EXTRACTO DE LA ESPECIFICACIÓN

##### Requisitos SESAR asociados

*The SSN server shall trigger an alert (INFORMATION) if an aircraft is moving from its stand without having received a Push-back clearance.*

##### Descripción extraída del diseño funcional

...

*Transcurso de una maniobra habitual de Push-back y luego Taxi:*

*T1) Está parada en el stand*

...

*T2) Empieza a moverse*

...

*T3) Está moviéndose en maniobra push-back*

...

*T4) Se para, una vez se ha alineado con la calle de rodadura*

...

*T5) Está parada, esperando la autorización TAXI*

...

*T6) Empieza a moverse*

...

*T7) Está rodando en TAXI*

...

*Una vez transcurrida esta secuencia, hay que tener en cuenta también que cada vez que la aeronave se para mientras está por las calles de rodadura, también puede pararse y volver a arrancar. Sería:*

*T8) Se para,*

...

*T9) Está parado,*

...

*T10) Empieza a moverse*

...

*T11) Está rodando en TAXI*

...

*Situaciones no cubiertas:*

*Con la utilización de este parámetro no se cubrirían ciertas situaciones hipotéticas, ya que se considerará para el prototipo una situación "ideal" en la maniobra de push-back.*

*Las situaciones no cubiertas en esta fase del prototipo serán:*

- *La aeronave se para en medio de la maniobra de push-back.*
- *Se pierde la pista de vigilancia en medio de la maniobra y se recupera después.*

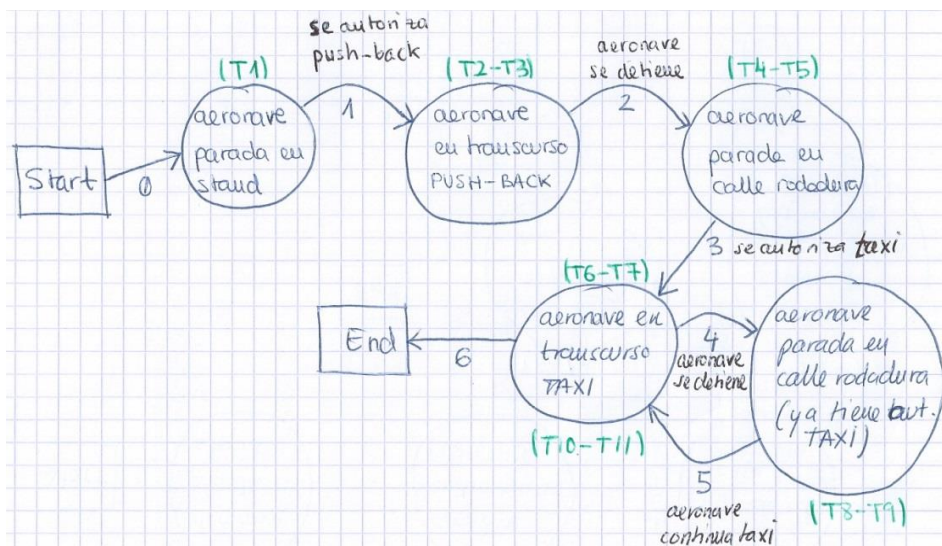
...

#### **Figura 31.** Especificación de alerta NO PUSH BACK CLEARANCE

Si se observa la test basis, no se dispone directamente de una descripción dada mediante puntos de decisión y caminos para esta maniobra de push-back. Es bastante común encontrarse con este tipo de situaciones en las que la forma en que está descrita la test basis no parece adecuada en un primer momento para aplicar una determina técnica.

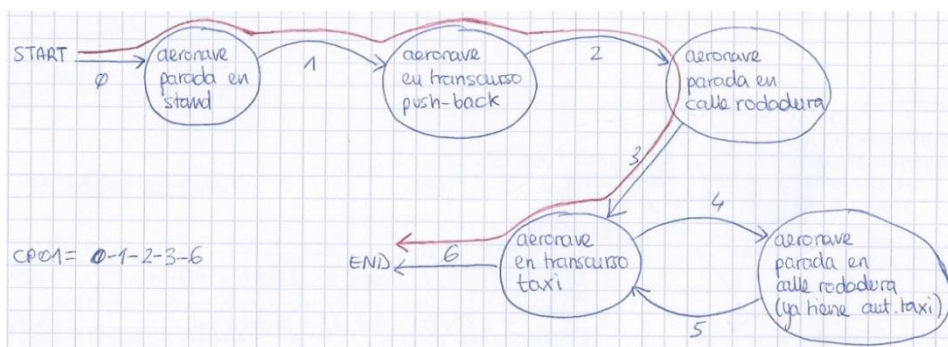
En este caso se describe de forma textual una secuencia de estados por los que va pasando una aeronave en función de determinados eventos, lo que encaja con los requisitos para aplicar cobertura de caminos.

Haciendo una pequeña labor de abstracción es posible extraer un modelo que permita aplicar la técnica (no tienen por qué ser exactamente los mismos que se indican en la test basis, en este caso se han agrupado algunos estados para simplificar el modelo):



**Figura 32.** Modelo de estados alerta NO PUSH BACK CLEARANCE

Una vez extraído un modelo, es posible aplicar cobertura de caminos de nivel 1:



**Figura 33.** Detalle de CP01 nivel 1 para NO PUSH BACK CLEARANCE

Dado que el camino tiene que cubrir cada situación de prueba (en el nivel 1 una situación de prueba era un arco) con un único caso de prueba es posible ejercitar la cobertura de caminos de nivel 1, lo cual se puede comprobar con la trazabilidad del CP01.

A continuación se muestra el diseño de casos de prueba aplicando el nivel 2. En primer lugar se desarrollan los pasos necesarios para derivar las situaciones de prueba:

1. Identificar para cada estado sus arcos entrantes y salientes	2. Combinar arcos
"aeronave parada en stand" IN: 0 OUT: 1	"aeronave parada en stand": 0-1
"aeronave en transcurso PUSH-BACK" IN: 1 OUT: 2	"aeronave en transcurso PUSH-BACK": 1-2
"aeronave parada en calle rodadura" IN: 2 OUT: 3	"aeronave parada en calle rodadura": 2-3
"aeronave en transcurso TAXI" IN: 3,5 OUT: 4,6	"aeronave en transcurso TAXI": 3-4; 3-6; 5-4; 5-6
"aeronave parada en calle rodadura (ya tiene aut. de TAXI)" IN: 4 OUT: 5	"aeronave parada en calle rodadura (ya tiene aut. de TAXI)": 4-5

**Figura 34.** Obtención situaciones de prueba nivel 2 de cobertura caminos

El conjunto de situaciones derivadas se podría ejercitar con 2 casos de prueba:

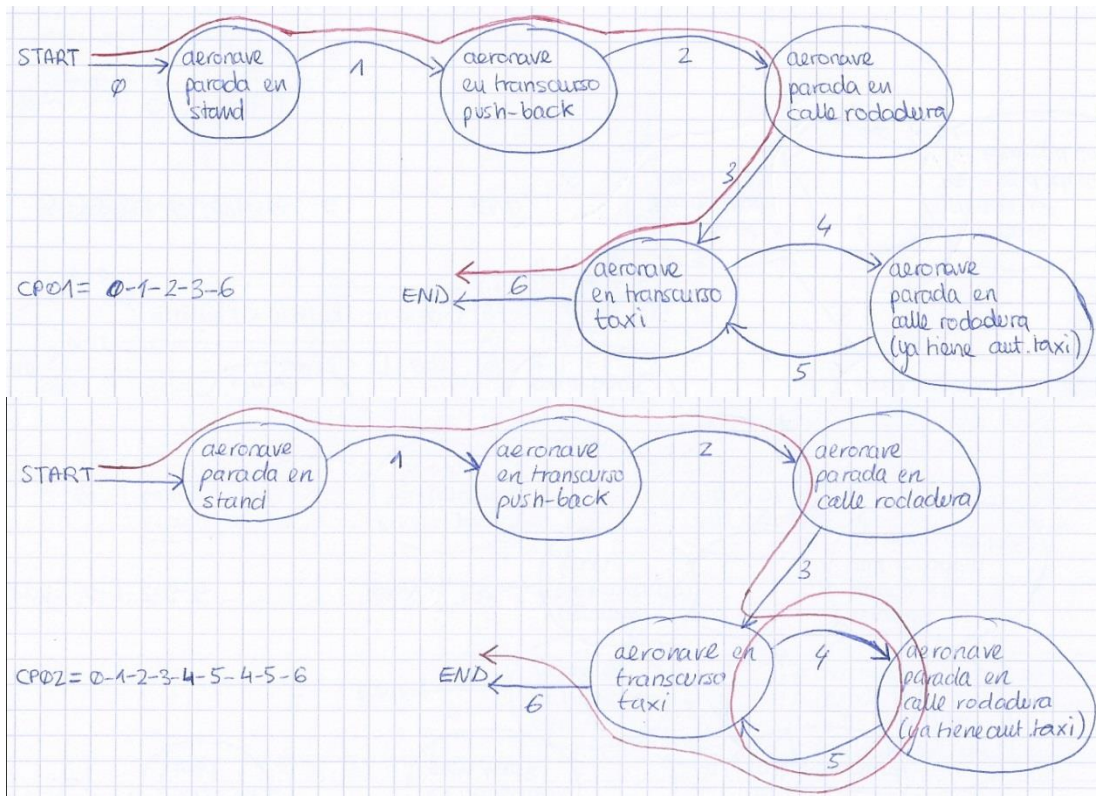


Figura 35. Detalle CP01 y CP02 nivel 2 NO PUSH BACK CLEARANCE

Este caso es sencillo dado que en general en cada estado solo existe un arco de entrada y otro de salida (excepto para el ciclo que se puede identificar en el modelo en el estado "aeronave en transcurso TAXI").

Se ha conseguido ejercitar la cobertura de caminos de nivel 2 con 2 casos de prueba. El primer caso representa un escenario en el que la aeronave no realiza ninguna parada una vez empieza a rodar en taxi, mientras que el segundo caso de prueba representa un escenario en el que se realizan dos paradas en el transcurso de la maniobra de taxi.

## 5.2. Aplicar técnicas

En el apartado anterior se introducían ejemplos sencillos y bastante acotados, donde la aplicación de las técnicas básicas era bastante directa. Normalmente no se darán estas circunstancias tan ideales y será necesario derivar situaciones de prueba en escenarios que tienen particularidades y que requieren alguna consideración adicional.

### 5.2.1. Acoplamiento

Cuando la test basis está descrita mediante un conjunto de condiciones puede darse el caso de que entre algunas de ellas exista *acoplamiento*.

Dos o más condiciones están acopladas cuando al variar una de ellas cambia también alguna de las otras (o todas).

Cuando variar el valor de una de las condiciones implica que *siempre* varía el valor de las otras, se dice que hay *acoplamiento fuerte*. Por ejemplo, en la siguiente expresión:

$$(X=0 \text{ AND } A) \text{ OR } (X \neq 0 \text{ AND } B)$$

las condiciones  $X=0$  y  $X \neq 0$  tienen acoplamiento fuerte. Suponiendo que la variable  $X$  tiene valor 0, cambiar el valor de  $X$  a cualquier otro valor siempre hace que cambien las dos condiciones, ya que  $X=0$  era cierta y pasa a ser falsa y  $X \neq 0$  era falsa y pasa a ser cierta.

Cuando variar el valor de una de las condiciones implica que *algunas veces*, pero no siempre, varía el valor de las otras, se dice que hay *acoplamiento débil*. Por ejemplo en la expresión:

$$X=0 \text{ OR } X=1 \text{ OR } X=3$$

existe acoplamiento débil entre las condiciones  $X=0$ ,  $X=1$  y  $X=3$ . Si el valor inicial de  $X$  fuese 0, cambiar el valor de la variable  $X$  de 0 a 2 sólo cambia la primera condición (de cierto a falso), mientras que cambiar el valor de  $X$  de 0 a 1 hace que cambien las dos primeras condiciones.

El acoplamiento entre condiciones puede dificultar el cumplimiento del criterio MCDC, debido a que no es posible variar el valor de una condición manteniendo fijo el resto de condiciones.

Cuando se aplica MCDC sobre una expresión existen dos formas de acoplamiento que pueden dar lugar a que no sea posible derivar un conjunto de situaciones de prueba que permita cumplir el criterio:

- existencia de *condiciones acopladas repetidas*
- acoplamiento *dependiente del contexto*

### 5.2.1.1. Condiciones acopladas repetidas

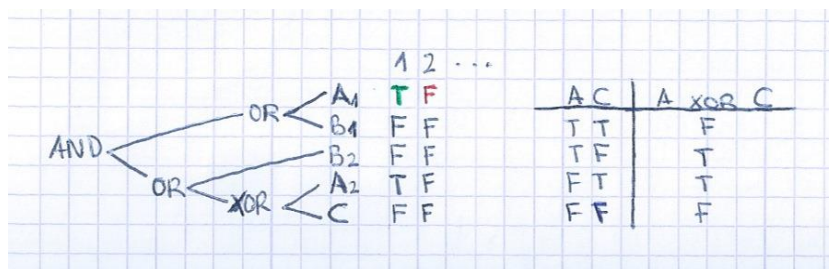
Un posible enfoque para abordar la existencia de condiciones acopladas repetidas se denomina *MCDC fuerte*, que consiste en intentar demostrar la independencia de cada aparición de la condición como si se tratase de una instancia independiente. Para ello es necesario que al variar cada instancia de la condición varíe la salida de la decisión, al mismo tiempo que se *enmascaran* los efectos de las demás instancias de la condición.

El concepto de enmascarar una condición se aplica a condiciones que están compuestas mediante operadores lógicos:

- $X \text{ AND } \text{FALSE} = \text{FALSE}$ , sin importar el valor de la condición  $X$ , de modo que el valor FALSE enmascara el valor de  $X$ .
- $X \text{ OR } \text{TRUE} = \text{TRUE}$ , sin importar el valor de la condición  $X$ , de modo que el valor TRUE enmascara el valor de  $X$ .

Cuando existe acoplamiento puede ocurrir que sean necesarias más de  $N+1$  situaciones de prueba para cumplir el criterio MCDC, en cualquier caso, si se está manejando una expresión testable (que sea posible ejecutar pruebas sobre ella), se requerirán como máximo  $2N$  situaciones de prueba (2 por cada condición, siendo  $N$  el número de condiciones).

Para algunas expresiones resulta imposible cumplir el criterio MCDC, como por ejemplo en la expresión  $(A \text{ OR } B) \text{ AND } (B \text{ OR } (A \text{ XOR } C))$ :



**Figura 36.** Ejemplo de condiciones acopladas repetidas

En la expresión anterior las condiciones A y B aparecen repetidas y dado que la técnica requiere probar el efecto independiente de cada instancia de cada condición, se les ha añadido un subíndice para diferenciarlas.

A continuación se explica cómo se rellenarían las 2 primeras columnas, que se corresponderían a las situaciones de prueba que deberían permitir demostrar el efecto independiente de  $A_1$  sobre la salida de la expresión:

- Columna 1:
  - a. Se comienza asignando a  $A_1$  el valor TRUE.
  - b. El valor de  $B_1$  debe ser FALSE dado que está conectado con  $A_1$  por medio del operador OR.
  - c. La condición  $B_2$  toma valor FALSE porque  $B_2=B_1$ .
  - d. La condición  $A_2$  toma valor TRUE porque  $A_2=A_1$ .
  - e. Dado el valor anterior de  $A_2$ , para que  $A_1$  cumpla el criterio MCDC es necesario que C tenga valor FALSE, ya que el resultado de ( $B_2$  OR ( $A_2$  XOR C)) debe ser necesariamente TRUE porque  $A_1$  se conecta con esta condición compuesta por medio del operador AND.
- Columna 2:
  - a. Se cambia el valor de  $A_1$  a FALSE, de modo que  $A_2$  tendrá que ser FALSE también.
  - b.  $B_1$  se debe mantener al valor FALSE, de modo que  $B_2$  tendrá que ser FALSE también.
  - c. El valor de C se debe mantener al valor FALSE de la columna 1. Debido a esto, el valor de ( $B_2$  OR ( $A_2$  XOR C)) va a ser FALSE, lo cual incumple el criterio dado que se debería mantener al valor TRUE.

### 5.2.1.2. Acoplamiento dependiente del contexto

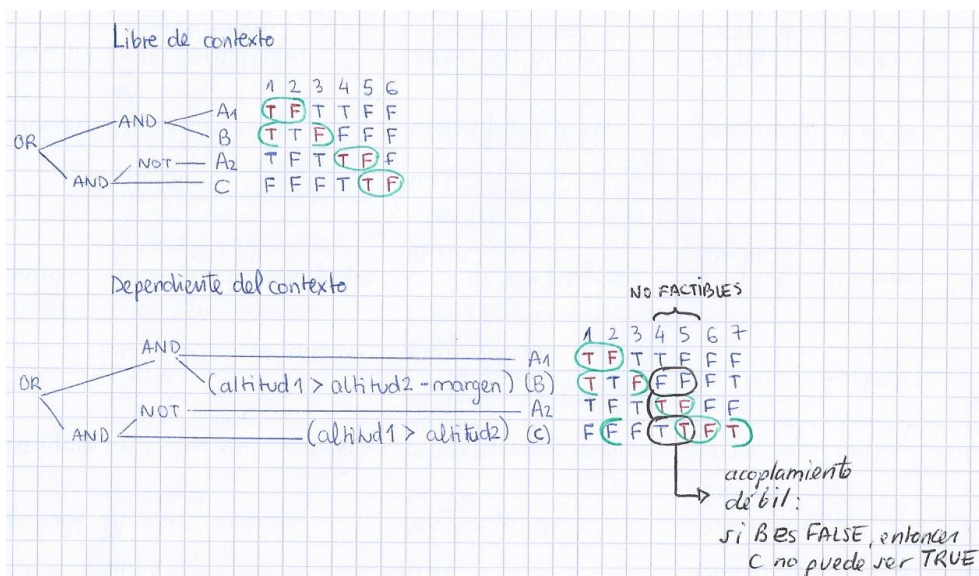
El acoplamiento dependiente del contexto ocurre cuando aparecen condiciones que incorporan operadores relacionales (<, >, <=, >=, ==, !=) sobre no booleanos. Por ejemplo en la siguiente expresión:

$$(A \text{ AND } (\textit{altitud1} > \textit{altitud2} - \textit{margen})) \text{ OR } (\text{NOT } A \text{ AND } (\textit{altitud1} > \textit{altitud2}))$$

las condiciones  $(\textit{altitud1} > \textit{altitud2} - \textit{margen})$  y  $(\textit{altitud1} > \textit{altitud2})$  presentan acoplamiento débil: si  $(\textit{altitud1} > \textit{altitud2} - \textit{margen})$  es FALSE entonces  $(\textit{altitud1} > \textit{altitud2})$  no puede ser TRUE. Esto sucede porque si un número ( $\textit{altitud1}$ ) es menor o igual que otro ( $\textit{altitud2}$ ) menos un margen positivo ( $\textit{margen}$ ), entonces es imposible que  $\textit{altitud1} > \textit{altitud2}$ .

El siguiente ejemplo presenta la diferencia entre acoplamiento no dependiente del contexto y acoplamiento dependiente del contexto. Se aplica MCDC a la expresión  $(A \text{ AND } B) \text{ OR } (\text{NOT } A \text{ AND } C)$ :





**Figura 37.** Ejemplo de acoplamiento dependiente del contexto

En el caso de estar libre de contexto se puede observar que:

- es posible demostrar el efecto independiente de la primera instancia de A (A<sub>1</sub>) si se mantiene B a TRUE y C a FALSE (C enmascara el valor de A<sub>2</sub>).
- se puede demostrar el efecto independiente de la segunda instancia de A (A<sub>2</sub>) manteniendo B a FALSE y C a TRUE.

Cuando se aplica MCDC a la expresión dependiente del contexto, ocurre que las situaciones de prueba 4 y 5 no son factibles debido a que es imposible que B sea FALSE y C sea TRUE simultáneamente.

A pesar de tener que eliminar la situación de prueba 5, todavía es posible demostrar el efecto independiente de C añadiendo la situación de prueba 7. Sin embargo no es posible hacerlo para A<sub>2</sub>, dado que precisamente las situaciones de prueba 4 y 5 son las que permiten demostrar el efecto independiente de esa condición.

Cuando no es posible cumplir MCDC para todas las condiciones de una expresión debido a problemas de acoplamiento y el problema no se soluciona con el enmascaramiento de condiciones, la pauta a seguir sería *relajar* dicho criterio y continuar diseñando pruebas con las situaciones que se hayan derivado para las condiciones que sí cumplen el criterio.

### Ejemplo 13

A continuación se presenta parte de la especificación de la alerta STATIONARY AFTER CLEARANCE, donde aparecen condiciones acopladas y se explica cómo proceder para cumplir el criterio MCDC.

#### EXTRACTO DE LA ESPECIFICACIÓN

##### Requisitos SESAR asociados

The SSN server shall trigger an alert (INFORMATION) when a mobile does not move after X seconds (parameter), having received an instruction to push-back, taxi, line-up, cross, enter or take-off.

##### Descripción extraída del diseño funcional

...

##### Detección del conflicto:

Si se cumplen todas las condiciones siguientes para el móvil :

Si es una Aeronave,

- Si está en fase TAXI
  - Última autorización recibida es Push-back / Taxi / Line-up / Cross
- Si está en fase ALINEACIÓN

- Última autorización recibida es Take-off
- Está parado (estaParado = true)
- horaEmpezoMovimiento < Hora última autorización recibida
- Tiempo que lleva parado el móvil = (Hora actual – Hora última autorización recibida) > TmaxParadoAutorizado

Si es un Vehículo,

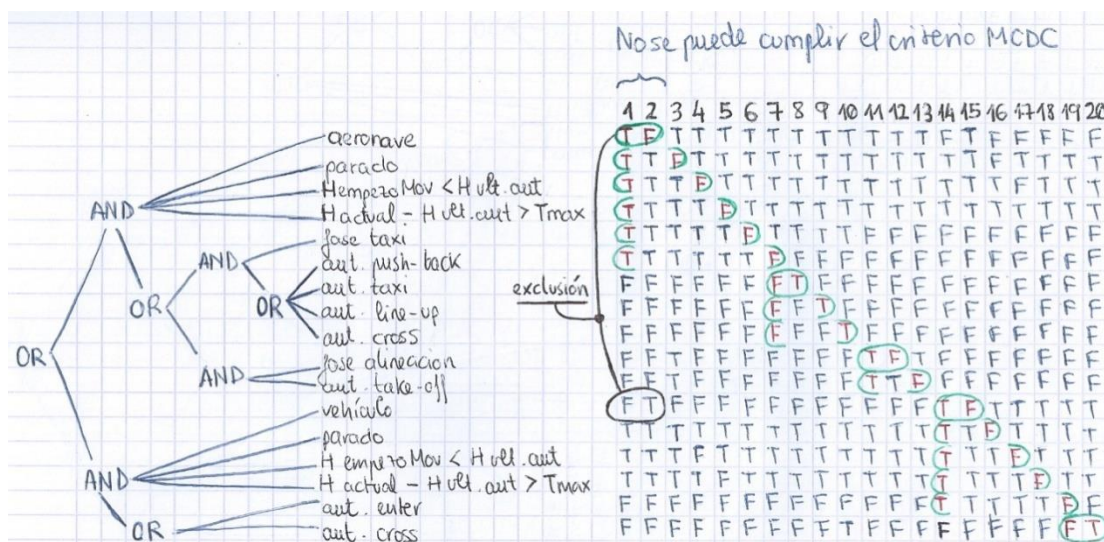
- Última autorización recibida es Cross / Enter
- Está parado (estaParado = true)
- horaEmpezoMovimiento < Hora última autorización recibida
- Tiempo que lleva parado el móvil = (Hora actual – Hora última autorización recibida) > TmaxParadoAutorizado

Se generará un conflicto tipo “Stationary after Clearance” y nivel de alerta “Information”.

...

**Figura 38.** Especificación de alerta STATIONARY AFTER CLEARANCE

Representar las condiciones de la test basis mediante un árbol puede servir de ayuda para aplicar el criterio:



**Figura 39.** MCDC sobre condiciones alerta STATIONARY AFTER CLEARANCE

En este caso se observa una condición excluyente, que es el *tipo de móvil*: cuando el móvil es una aeronave es imposible que pueda ser un vehículo y viceversa. Al aplicar la técnica directamente se puede observar que, debido a esto último, no es posible cumplir el criterio MCDC:

Con las situaciones de prueba 1 y 2 ya se puede afirmar que, debido a la exclusión mutua, no es posible cumplir el criterio MCDC.

Una solución consiste en separar la expresión anterior atendiendo a las condiciones mutuamente excluyentes. Una forma de hacer esto puede ser la siguiente:

1. Numerar las condiciones excluyentes.
2. Tomar cada condición excluyente que corresponda a una hoja del árbol e ir componiendo con el resto de condiciones que no están acopladas, mediante los operadores lógicos que se van encontrando en el camino hasta la raíz del árbol.

En general cuando se separan expresiones se debe tener la certeza de que las distintas partes (subexpresiones) son completamente independientes para que no haya *efectos laterales*, que ocurren cuando se hacen cambios en los valores de las condiciones de una parte y esto tiene algún efecto en alguna otra de las partes en que se ha separado la expresión original.

De esta manera se obtienen dos expresiones sobre las que aplicar el criterio:

- La expresión que representa las condiciones que se tienen que dar para detectar el conflicto cuando el tipo de móvil = aeronave.

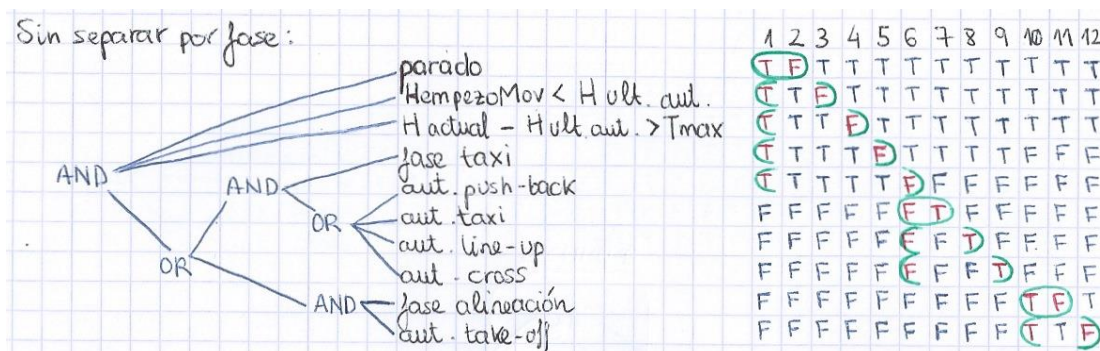


Figura 40. MCDC separando por tipo de móvil = aeronave

- Idem para cuando el tipo de móvil = vehículo.

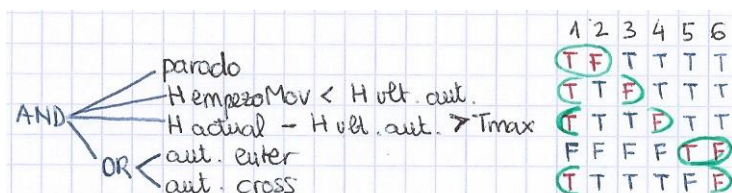


Figura 41. MCDC separando por tipo de móvil = vehículo

Para la expresión de la Figura 40 parece posible cumplir el criterio, siendo necesarias 12 situaciones de prueba: aquí no se cumple la regla general de N+1 situaciones de prueba (dado que N=10).

Una vez derivadas las situaciones de prueba es necesario comprobar si todavía existe alguna *no ejecutable* (para la que no se puede ejecutar una prueba debido a los valores que tienen que tomar las condiciones).

En este caso se da la circunstancia de que sería necesario verificar si posible o no ejecutar en el entorno de pruebas casos que ejerciten las situaciones 5 y 11, debido a que:

- las únicas fases posibles para un despegue se supone que son SELECTED (por defecto), TAXI, ALINEACIÓN y DESPEGUE.
- las aeronaves que son despegues (según su PV) cumplen las condiciones de la fase TAXI prácticamente al iniciar la simulación en el entorno de pruebas, de modo que en la práctica no será posible probar una situación donde una aeronave esté en fase SELECTED.
- la fase DESPEGUE exige que la aeronave esté en movimiento (estaParado = FALSE).
- las situaciones 5 y 11 representan situaciones en las que *fase en que encuentra el móvil* no es TAXI ni ALINEACIÓN.
- teniendo en cuenta las razones expuestas anteriormente habría que verificar (comprobar que ya ha sido probado antes) que no es posible probar con las fases SELECTED o DESPEGUE: si finalmente no es posible, entonces las situaciones 5 y 11 no serán factibles (no es posible ejercitarlas en el entorno de pruebas).

Las situaciones 5 y 11 son precisamente las que permiten demostrar la influencia que tienen las condiciones *fase en que encuentra el móvil = taxi* y *fase en que se encuentra el móvil = alineación* sobre el resultado de la expresión que detecta el conflicto. Opcionalmente se podría volver a separar la expresión por la condición *fase en que se encuentra el móvil* con el objetivo de cumplir el criterio de forma estricta:



CP01	Paso 1	Paso 2	Paso 3	Paso 4
Entrada	Aeronave parada en el stand	Aeronave parada en T13	Aeronave parada en T24	Aeronave parada en T33
Descripción	En T3 dar aut. push-back En T4 aeronave sigue ruta PV En T10 parar Mantener parada hasta T13	En T14 aeronave continúa ruta PV En T15 dar aut. push-back En T20 parar Mantener parada hasta T24	En T25 seguir ruta PV En T26 dar aut. push-back En T30 parar Mantener parada hasta T33	En T34 seguir ruta PV En T35 dar aut. push-back En T41 parar
Salida	No se genera alerta porque $HempzoMor < HultAut$ es FALSE	En [T20, T24] se genera alerta	En [T31, T35] se genera alerta	En T41 se genera alerta
	Paso 5	Paso 6	Paso 7	Paso 8
Entrada	Aeronave parada en T41	Aeronave parada en T46	Aeronave parada en T58	Aeronave parada en T70
Descripción	En T41 dar aut. enter (no afecta a alerta) Mantener parada hasta T46	En T47 seguir ruta PV En T48 dar aut. taxi En T53 parar Mantener parada hasta T58	En T59 seguir ruta PV En T60 dar aut. cross En T65 parar Mantener parada hasta T70	En T71 seguir ruta PV En T72 dar aut. line-up En T77 parar Mantener parada hasta T82
Salida	Se genera alerta en T46	En [T55, T58] se genera alerta	En [T65, T70] se genera alerta	En [T77, T82] se genera alerta
	Paso 9	Paso 10		
Entrada	Aeronave parada en T82	Aeronave parada en T94		
Descripción	En T83 seguir ruta PV En T84 dar aut. enter (no afecta a alerta) En T89 parar Mantener parada hasta T94	En T95 seguir ruta PV En T96 dar aut. push-back En T101 parar Mantener parada hasta T106		
Salida	No se genera alerta	En [T101, T106] se genera alerta		

**Figura 44.** Detalle CP01 ejercitando situaciones de la Figura 40

Como se puede observar, el CP01 tiene 10 pasos de prueba con los que es posible cubrir las 10 situaciones de prueba para las que es posible ejecutar pruebas en el entorno de pruebas (situaciones 1, 2, 3, 4, 6, 7, 8, 9, 10 y 12).

Lo anterior no quiere decir que en general haya una relación directa entre el número de situaciones de prueba y el de pasos de prueba necesarios, ya que puede ocurrir que con unos pocos pasos de prueba se puedan ejercitar varias situaciones de prueba.

Un detalle importante en este tipo de pruebas interactivas, donde el tester emula las acciones que haría un controlador con el sistema, es que cuando se indican valores relativos a tiempos no siempre será posible que al ejecutar las pruebas se consigan valores exactos.

En este caso hemos supuesto una precisión de segundos, y a la hora de ejecutar los distintos pasos de prueba seguramente habrá alguna diferencia entre el instante en que se realizan las acciones en la realidad y en el instante indicado en el diseño, ya que el tester estará limitado por el tiempo de reacción que tenemos las personas, por ejemplo para hacer un clic de ratón en un momento preciso.

Debido a esta particularidad las pruebas de valores límite no tienen mucho sentido cuando se trata de pruebas interactivas. Para que esto fuese posible sería necesario poder ejecutar los casos de prueba de forma automatizada.

Para ejercitar las situaciones derivadas para la expresión que detecta el conflicto cuando el móvil es un vehículo sería necesario diseñar un nuevo caso de prueba (CP02), que no se

desarrolla aquí dado que se procedería de forma muy similar al caso de prueba CP01 explicado anteriormente.

## 5.2.2. Inferir modelos a partir de ejemplos

En el uso real de un sistema interactivo puede suceder que el usuario trabaje sobre un escenario que evoluciona en función de determinadas acciones, pasando por otros escenarios distintos.

En el diseño de pruebas es importante tener presente lo anterior, ya que de lo contrario podrían quedar sin detectar fallos que solamente ocurren cuando se producen dichas transiciones entre escenarios.

Cuando en la test basis se describen escenarios en los que pueden realizarse diferentes tipos de acciones que pueden tener un impacto mutuo, es interesante pensar si es posible representar dichos escenarios mediante un modelo de transición de estados. Esto puede servir de ayuda para derivar situaciones más completas, al encadenar escenarios de forma consecutiva.

### Ejemplo 14

En este ejemplo se describe una alerta que se genera cuando se detecta un móvil cruzando una stop bar en rojo (stop bar en estado activo).

La descripción dada (test basis) consiste en una serie de gráficos que representan un conjunto de escenarios en los que se debe generar la alerta, pero cada escenario se presenta de forma independiente.

#### EXTRACTO DE LA ESPECIFICACIÓN

##### Requisitos SESAR asociados

*The SSN server shall trigger an alert (ALARM) if a mobile is detected crossing a red stop bar (wherever the location of the stop bar is).*

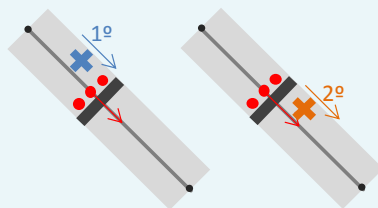
##### Descripción extraída del diseño funcional

...

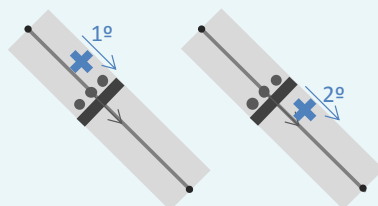
*Detección del conflicto:*

*Situaciones cubiertas:*

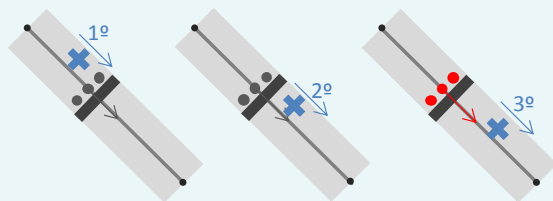
- a) *El móvil supera una Stop bar activa (en su dirección de funcionamiento) -> Se genera la alerta.*



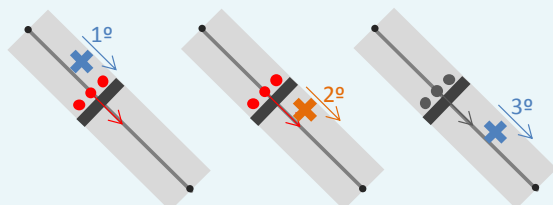
- b) *El móvil supera una Stop bar inactiva -> No se genera la alerta.*



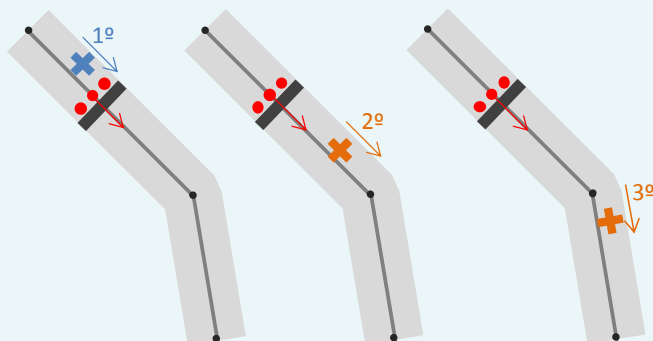
c) El móvil supera una Stop bar inactiva y después ésta se activa -> No se genera alerta, porque el momento en el que superó la Stop bar es anterior al momento de activación de la Stop bar.



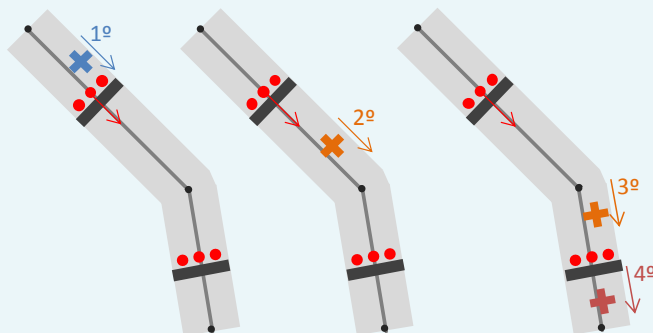
d) El móvil supera una Stop bar activa y después ésta se desactiva -> Se genera alerta hasta el momento en el que la Stop bar se desactiva. Entonces se para de generar la alerta.



e) El móvil supera una Stop bar activa y continúa por los siguientes segmentos -> Se genera la alerta y se mantiene aunque esté en otros segmentos y estos tengan Stop bars (ver caso siguiente de dos Stop bars superadas en rojo).



f) El móvil supera dos Stop bar en rojo consecutivas -> Se generará una alerta con la primera Stop bar, hasta que supere la segunda. Entonces se generará la alerta con la segunda Stop bar y desaparecerá la primera (el móvil no podrá generar dos alertas de tipo Stop bar simultáneas).



g) El móvil accede por la parte del segmento posterior a una Stop bar activa y la supera en dirección contraria a la de su funcionamiento -> No se genera alerta.

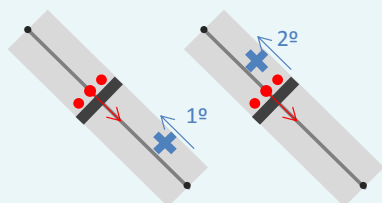


Figura 45. Especificación de alerta RED STOP BAR

A la hora de crear un modelo de transición de estados a partir de la test basis anterior puede ser útil responder a una serie de preguntas:

1) *¿La descripción hace referencia a elementos que pueden tener impacto mutuo?*

Es importante responder esta cuestión, porque cuando se describe el comportamiento de varios elementos diferentes que tienen impacto mutuo puede tratarse de un escenario difícil de modelar.

Aquí se podrían identificar en primera instancia varios elementos:

- Un móvil que va atravesando barreras.
- Barreras que va atravesando el móvil.

En este caso, el estado de la alarma del móvil depende del estado de activación de las barreras, pero no sucede lo contrario y por tanto no existe impacto mutuo (por ejemplo una barrera activa no se desactiva al ser superada por un móvil).

De esta manera el elemento que se va a modelar será un móvil cualquiera.

2) *¿Cuáles son los estados del elemento?*

Aquí se tratarían dos estados, que se combinan:

- el estado de la alarma RED STOP BAR del móvil: activa/inactiva.
- si el móvil se encuentra antes de una barrera o acaba de atravesarla.

3) *¿Cuáles son los eventos que intervienen y que permiten transicionar de un estado a otro?*

Se pueden identificar varios:

- Activar/desactivar una barrera.
- Cuando el móvil supera una barrera en una determina dirección: DF (Dirección de Funcionamiento) o DCF (Dirección Contraria a la de Funcionamiento).
- Cuando el móvil cambia del segmento actual a otro que puede tener (o no tener) otras barreras.

4) *¿Hay acciones que se ejecutan ante la ocurrencia de determinados eventos?*

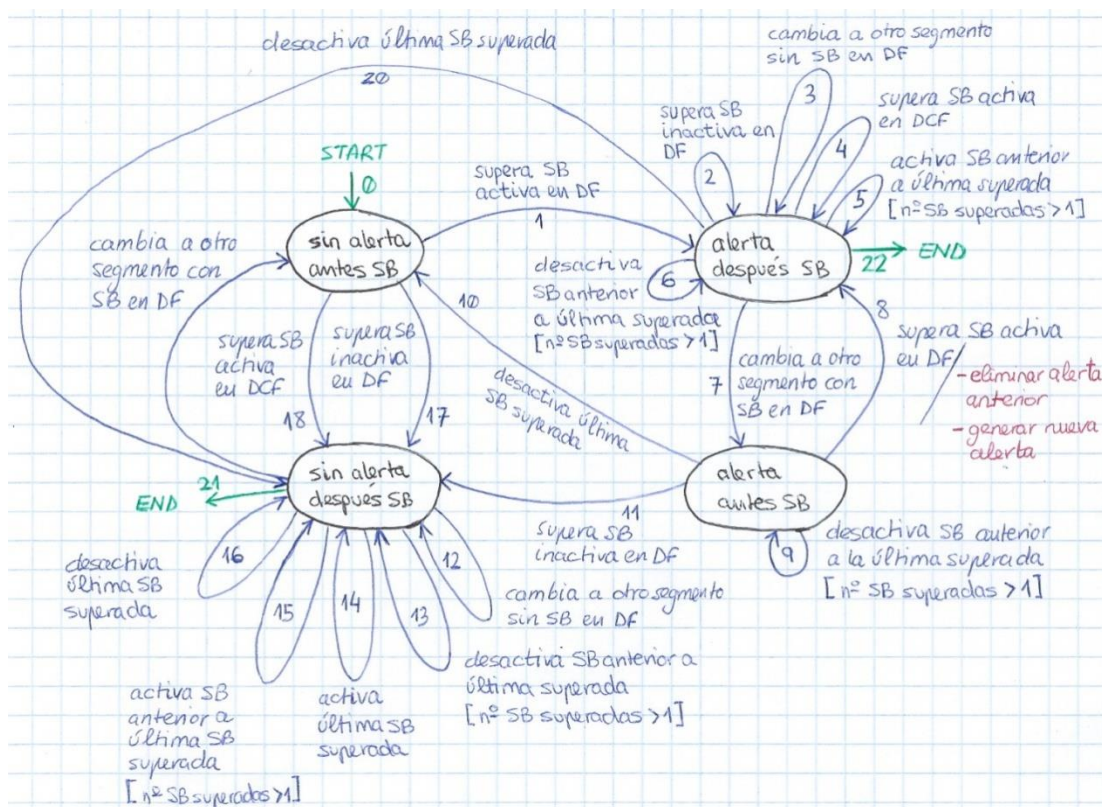
Una acción es un resultado derivado de la ocurrencia de un evento. En este caso, cuando se atraviesan 2 stop bar en rojo consecutivas, tras superar la segunda stop bar suceden 2 acciones:

- se elimina la alerta activa actual del móvil.
- se genera una nueva alerta para el móvil.

Con esta información se puede comenzar a modelar el estado de activación de la alarma RED STOP BAR para un móvil en función de los eventos descritos. Normalmente se tratará de una labor iterativa e incremental, de manera que es probable que se generen algunos modelos intermedios que se irán mejorando hasta obtener el modelo final.

En ocasiones puede ser necesario representar escenarios muy complejos y esto podría ser causa de rechazo hacia realizar esta labor de modelado. Ante esto hay que plantearse que siempre será mejor tener un modelo incompleto y que permita derivar algunas situaciones de prueba más realistas, frente a no tener ningún modelo y por tanto no derivar ese tipo de situaciones.





**Figura 46.** Modelo inferido a partir de especificación de alerta RED STOP BAR

Una vez construido el modelo se pueden añadir estados auxiliares de inicio y de fin para facilitar la aplicación de la técnica de cobertura de caminos (estados START y END), ya que cada caso de prueba debe atravesar una secuencia de arcos desde un principio hasta un final.

Aquí se ha considerado que el estado inicial del móvil siempre será que “no tiene alerta” y que para terminar es necesario “haber atravesado alguna stop bar” (para que por ejemplo no haya secuencias de arcos que representen que el móvil se detiene justo antes de atravesar una stop bar).

A partir del modelo se podrían derivar las siguientes situaciones de prueba aplicando cobertura de caminos de nivel 2:

## 1. Identificar para cada estado sus arcos entrantes y salientes

"sin alerta antes SB"

IN: 0, 10, 19

OUT: 1, 17, 18

"sin alerta después SB"

IN: 17, 18, 20, 11, 12, 13, 14, 15, 16

OUT: 12, 13, 14, 15, 16, 19, 21

"alerta después SB"

IN: 1, 2, 3, 4, 5, 6, 8

OUT: 2, 3, 4, 5, 6, 7, 20, 22

"alerta antes SB"

IN: 7, 9

OUT: 8, 10, 9, 11

## 2. Combinar arcos

"sin alerta antes SB": 0-1; 0-17; 0-18; 10-1; 10-17;  
10-18; 19-1; 19-17; 19-18

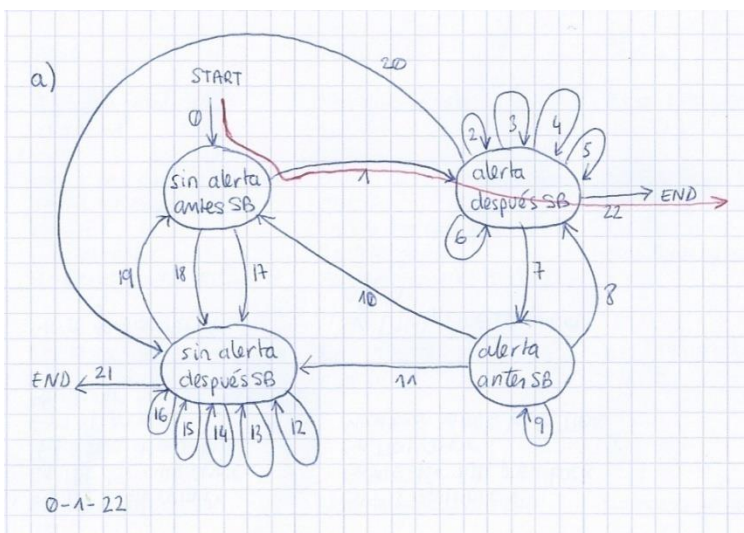
"sin alerta después SB": 17-12; 17-13; 17-14; 17-15;  
17-16; 17-19; 17-21; 18-12; 18-13;  
18-14; 18-15; 18-16; 18-19; 18-21;  
20-12; 20-13; 20-14; 20-15; 20-16;  
20-19; 20-21; 11-12; 11-13; 11-14;  
11-15; 11-16; 11-19; 11-21; 12-12;  
12-13; 12-14; 12-15; 12-16; 12-19;  
12-21; 13-12; 13-13; 13-14; 13-15;  
13-16; 13-19; 13-21; 14-12; 14-13;  
14-14; 14-15; 14-16; 14-19; 14-21;  
15-12; 15-13; 15-14; 15-15; 15-16;  
15-19; 15-21; 16-12; 16-13; 16-14;  
16-15; 16-16; 16-19; 16-21

"alerta después SB": 1-2; 1-3; 1-4; 1-5; 1-6; 1-7; 1-20;  
1-22; 2-2; 2-3; 2-4; 2-5; 2-6; 2-7;  
2-20; 2-22; 3-2; 3-3; 3-4; 3-5; 3-6;  
3-7; 3-20; 3-22; 4-2; 4-3; 4-4; 4-5;  
4-6; 4-7; 4-20; 4-22; 5-2; 5-3; 5-4;  
5-5; 5-6; 5-7; 5-20; 5-22; 6-2; 6-3;  
6-4; 6-5; 6-6; 6-7; 6-20; 6-22; 8-2;  
8-3; 8-4; 8-5; 8-6; 8-7; 8-20; 8-22

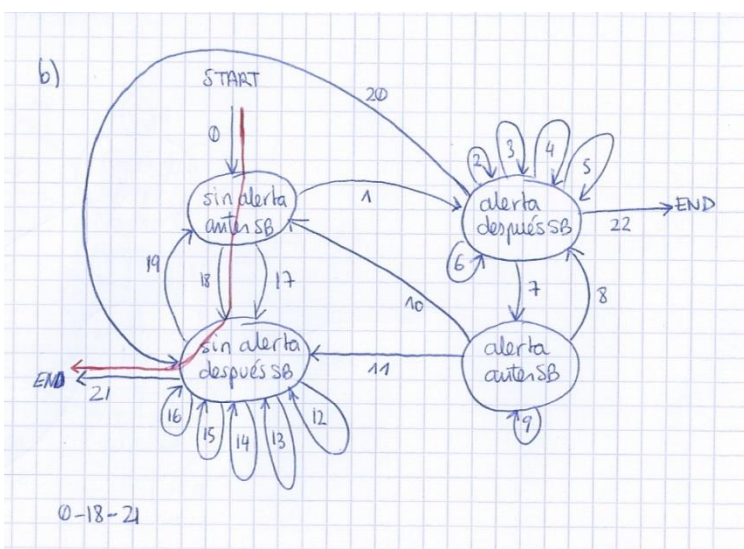
"alerta antes SB": 7-8; 7-10; 7-9; 7-11; 9-8; 9-10;  
9-9; 9-11

**Figura 47.** Derivar situaciones de prueba nivel 2 a partir del modelo de estados

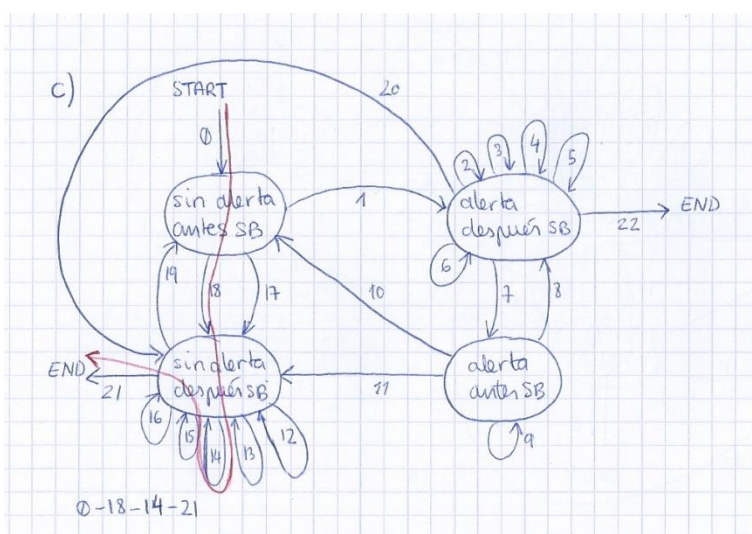
Los siguientes casos de prueba corresponden a las situaciones independientes descritas en la test basis (situaciones a, b, c, d, e, f y g):



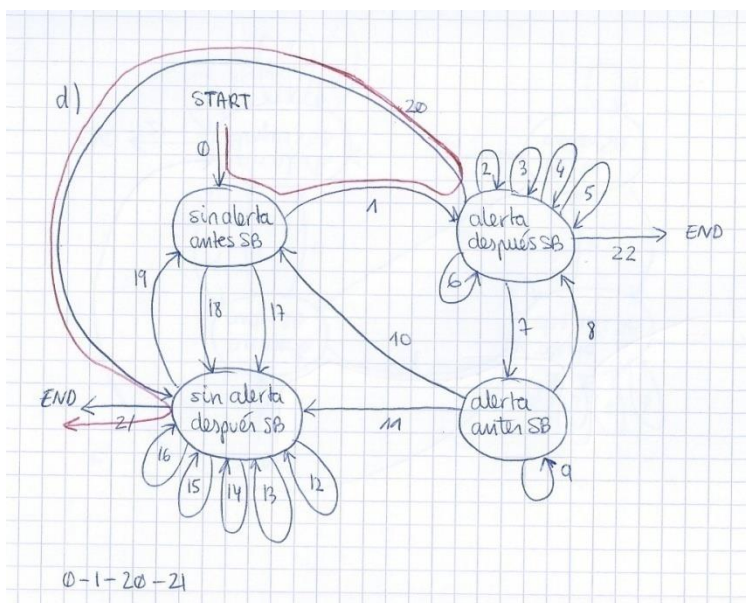
**Figura 48.** Situación a) descrita en la especificación de alerta RED STOP BAR



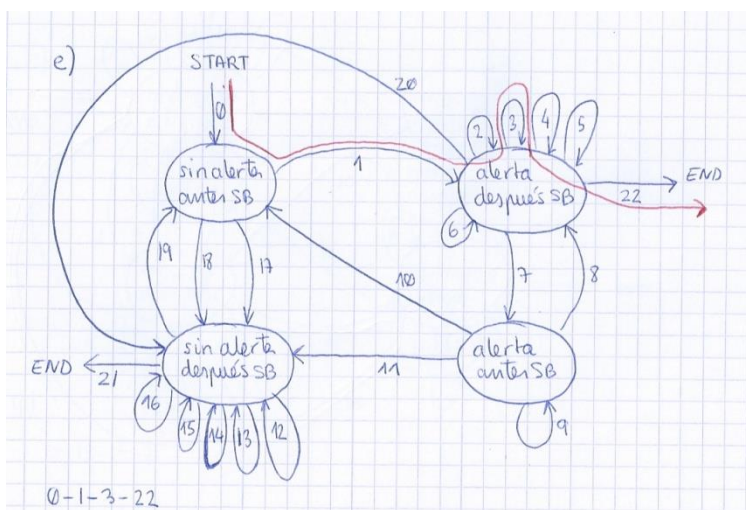
**Figura 49.** Situación b) descrita en la especificación de alerta RED STOP BAR



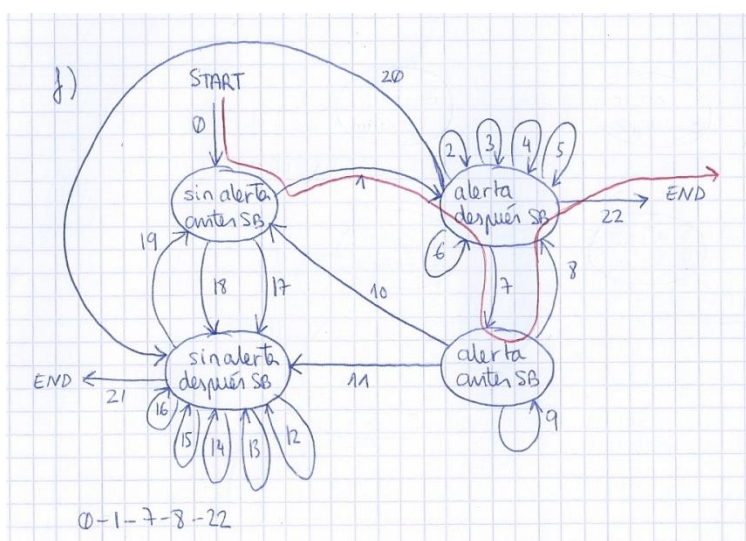
**Figura 50.** Situación c) descrita en la especificación de alerta RED STOP BAR



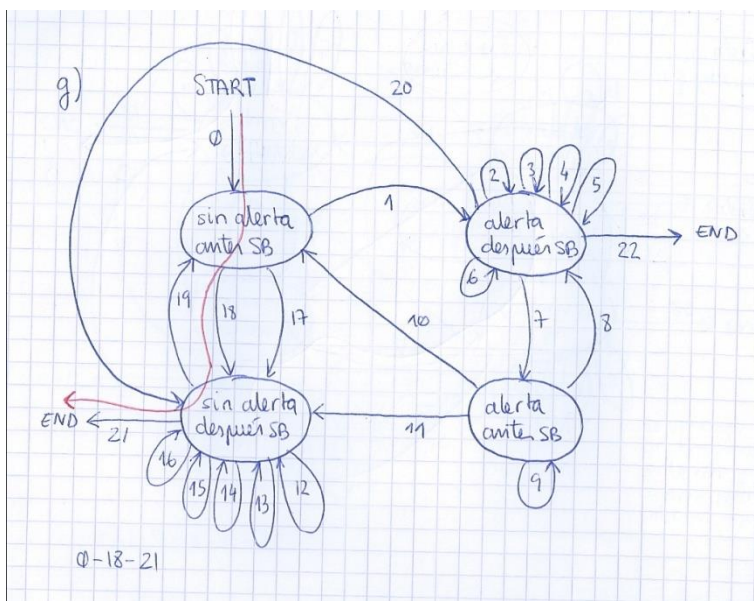
**Figura 51.** Situación d) descrita en la especificación de alerta RED STOP BAR



**Figura 52.** Situación e) descrita en la especificación de alerta RED STOP BAR



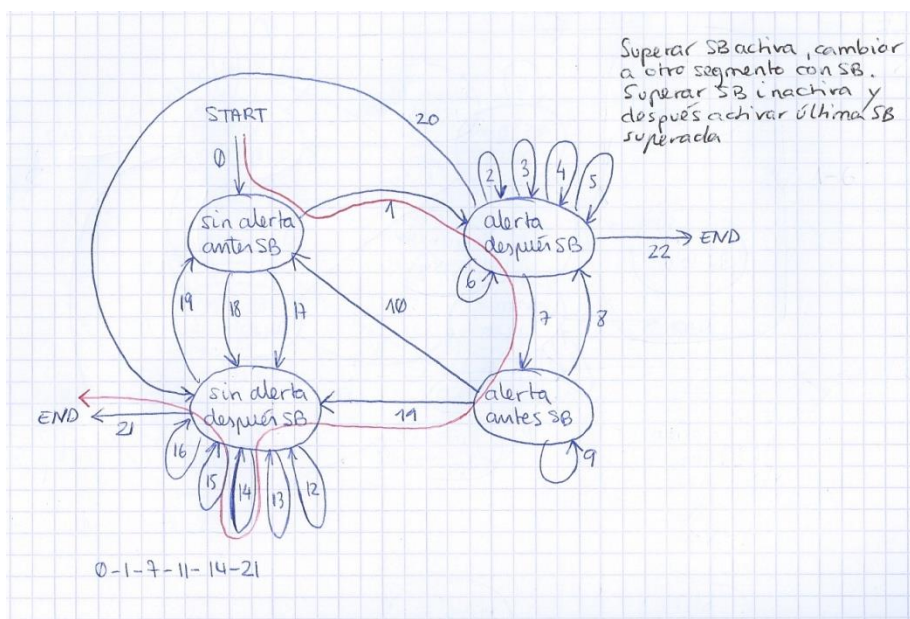
**Figura 53.** Situación f) descrita en la especificación de alerta RED STOP BAR



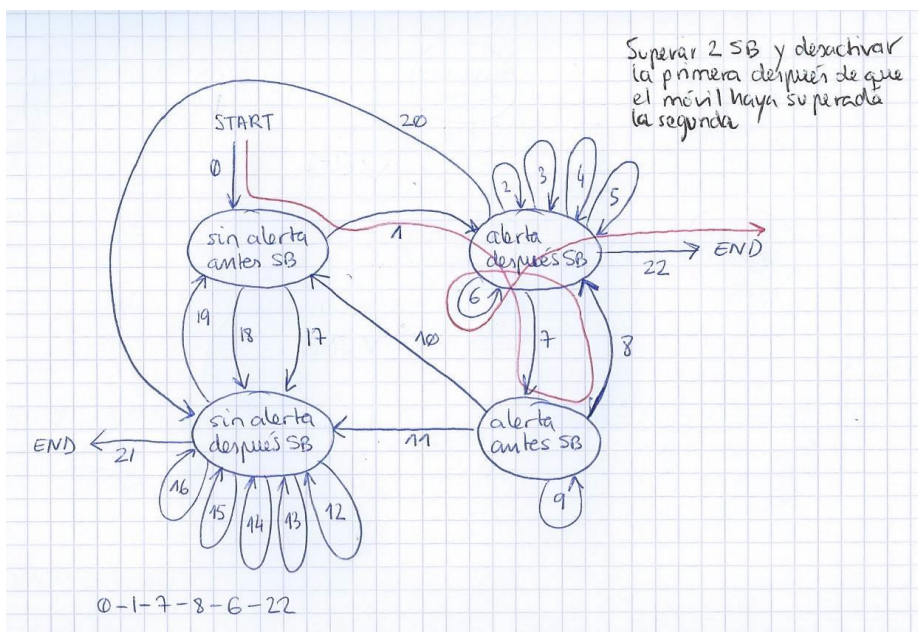
**Figura 54.** Situación g) descrita en la especificación de alerta RED STOP BAR

A medida que se van diseñando los casos de prueba es imprescindible mantener actualizada la trazabilidad respecto a las situaciones de prueba (las derivadas en el paso 2 de la técnica: combinar arcos de entrada y salida). La forma más sencilla de hacer esto es ir tachando las situaciones de prueba (combinaciones de arcos) que se ejercitan con cada caso de prueba, de forma que sea fácil identificar cuales quedan pendientes.

A continuación se presentan algunos ejemplos del resto casos de prueba necesarios (no se han desarrollado todos):



**Figura 55.** Otros casos de prueba de alerta RED STOP BAR (1 de 2)



**Figura 56.** Otros casos de prueba de alerta RED STOP BAR (2 de 2)

Al diseñar casos de prueba es necesario tener en cuenta la *testabilidad*, es decir, si es posible ejecutar las pruebas diseñadas (ya sea aplicando cobertura de caminos o cualquier otra técnica).

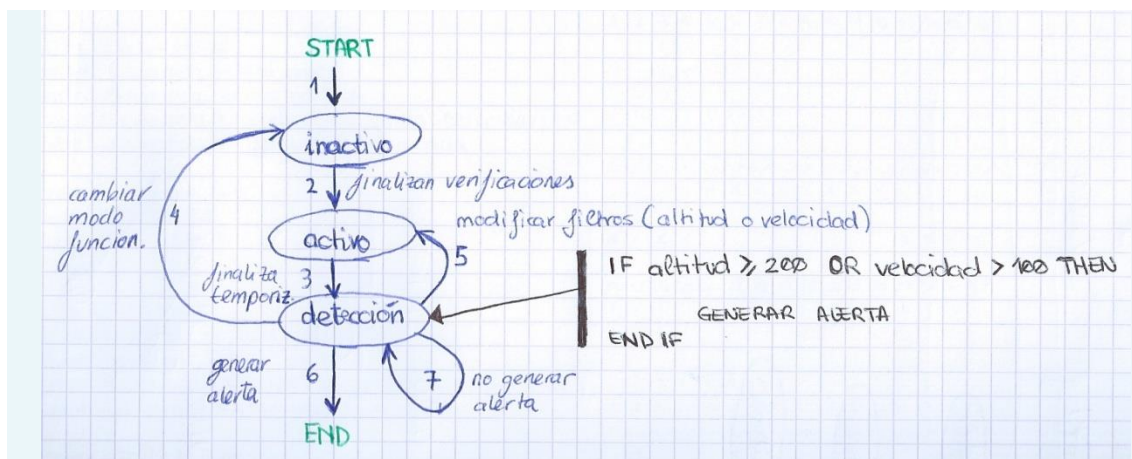
En este ejemplo habrá que tener en cuenta si se exige hacer las pruebas sobre un conjunto particular de aeródromos, ya que debido a la topología de cada aeródromo podría haber secuencias que no se puedan probar (por ejemplo una situación en la que la aeronave atravesase 4 segmentos, habiendo una barrera activa en cada uno de ellos).

### 5.3. Combinar técnicas

Cuando se aplican varias técnicas de forma combinada sobre un objetivo de prueba lo que se consigue es complementar las situaciones derivadas. Esto permite diseñar pruebas más completas y realistas que tengan más probabilidades de detectar aquellos defectos más difíciles de encontrar.

A continuación se presenta un ejemplo sintético, en el que se describe un determinado proceso a partir de una especificación de requisitos y un diagrama de flujo:

EXTRACTO DE LA ESPECIFICACIÓN	
Requisito	Prioridad
<b>REQ-1:</b> REQ-1-1: Cuando finalicen las verificaciones iniciales del estado "inactivo" se pasará al estado "activo". REQ-1-2: Cuando finalice el temporizador del estado "activo" se pasará automáticamente al estado "detección". REQ-1-3: Al cambiar el modo de funcionamiento siendo "detección" el estado actual, se pasará automáticamente al estado "inactivo". REQ-1-4: En el estado "detección" el sistema comprueba periódicamente la condición de detección de conflicto. REQ-1-4-1: Si la condición de detección de conflicto se cumple, se genera una alerta y se finaliza. REQ-1-4-2: Si la condición no se cumple, se continúa en el estado "detección".	MAX
<b>REQ-2:</b> Al pasar por cada estado se registra en un log una marca horaria, el estado el estado de procedencia y el estado actual.	MIN



**Figura 57.** Especificación del ejemplo sintético

Sobre esta especificación se podría aplicar la técnica de cobertura de caminos, que está orientada a encontrar defectos que solo se manifiestan cuando se siguen determinados flujos. Supongamos que se decide aplicar cobertura de caminos de nivel 2 y se obtienen los siguientes casos de prueba:

- CP01: 1-2-3-4-2-3-5-3-6
- CP02: 1-2-3-7-4-2-3-7-5-3-7-7-6

Si existe algún defecto que se manifiesta al seguir cualquiera de las combinaciones posibles de 2 arcos consecutivos, dicho defecto será detectado con una probabilidad bastante alta (aunque nunca se garantiza nada al 100%) al ejecutar los casos de prueba CP01 y CP02 anteriores.

Evidentemente también podría ocurrir que existiese algún defecto que solo se manifestase al seguir combinaciones de 3 arcos consecutivos. En general, a mayor nivel de cobertura de caminos, mayor certeza de que se va a encontrar defectos más complejos, pero a costa también de un mayor coste en la aplicación de la técnica.

El máximo nivel de cobertura sería el nivel “infinito”, pero no sería factible en la práctica. No obstante, ni tan siquiera en el caso de que el nivel infinito fuese factible sería posible afirmar que de esta manera se iban a encontrar “todos” los defectos existentes. Vamos a ver esto sobre un ejemplo:

En el punto de decisión correspondiente al estado “detección”, hay 4 situaciones posibles que generan alerta:

- situación 1 (T T): altitud  $\geq 200$ ; velocidad  $> 100$  → GENERAR ALERTA
- situación 2 (T F): altitud  $\geq 200$ ; velocidad  $\leq 100$  → GENERAR ALERTA
- situación 3 (F T): altitud  $< 200$ ; velocidad  $> 100$  → GENERAR ALERTA
- situación 4 (F F): altitud  $< 200$ ; velocidad  $\leq 100$  → NO HAY ALERTA

Supongamos que existe un defecto en la implementación del proceso descrito anteriormente, de manera que en la situación 3 no se genera alerta (lo cual sería incorrecto) y supongamos también que disponemos de un conjunto de casos de prueba de nivel infinito.

El defecto ocurriría en la transición “generar alerta” (arco 6), cuando se supone que se va a generar la alerta. Existen sin embargo otras 2 situaciones (situaciones 1 y 2) que también están asociadas con el arco 6. En este caso, el conjunto de casos de prueba de nivel infinito podría haber pasado por el arco 6 muchas veces, pero no habría garantía de que al menos uno de dichos casos de prueba ejercita la situación 3 (todas las veces que se pasa por ese arco es porque cumple la situación 1 o la 2).

Lo anterior sucede porque la técnica de cobertura de caminos no ha sido concebida para ese propósito. El conjunto de casos de prueba se diseñó para ejercitar *combinaciones de caminos*, pero el defecto descrito no tiene relación con esto.

El defecto podría haberse encontrado combinando cobertura de caminos con la técnica de cobertura de puntos de decisión.

En términos de los casos de prueba anteriores, “combinar” significa que la transición “generar alerta” se debería probar varias veces con valores distintos de las condiciones, en este caso cubriendo todas las combinaciones posibles de las condiciones del punto de decisión correspondiente al estado “detección” (Múltiple Condición).

### 5.3.1. Priorizar técnicas

En ocasiones pueden darse circunstancias que pueden conducir a tener que elegir entre varias técnicas posibles, por ejemplo debido a que no se pueda dedicar suficiente tiempo a pruebas, que haya que justificar costes ante un cliente o por otras cuestiones en esta línea.

Sobre el proceso utilizado como ejemplo hasta ahora habría que responder a algunas preguntas:

¿En el proceso se ejecutan diferentes tipos de acciones cuyas consecuencias afectan la una a la otra, hasta llegar a una situación en la que se debe generar una alerta?

En el ejemplo anterior esto es así y se corresponde con el grupo de requisitos REQ-1 de la Figura 57, que tienen la prioridad más alta. En estas circunstancias, la técnica de cobertura de caminos de nivel 2 sería una elección recomendable, basándonos en la prioridad.

Digamos que en cambio el requisito más prioritario fuese REQ-2, considerando por ejemplo que es mucho más importante que tras evaluar cada punto de decisión se registre o no cierto dato en un determinado log.

En este último caso, las combinaciones de caminos no son interesantes ya que el registro de datos que se produce al pasar de un estado al otro no tiene un impacto mutuo, de manera que en estas circunstancias sería más recomendable utilizar alguna variante de cobertura de puntos de decisión.

Si se diese la circunstancia de que es necesario hacer una elección entre técnicas pero no es posible basarse en la prioridad (porque se desconoce o es la misma), entonces un criterio de selección alternativo podría ser que el desarrollador estime el esfuerzo de aplicación de cada técnica (estas ya vendrían determinadas por la estrategia de pruebas) y decida aplicar aquella que implique menor esfuerzo (en horas), de cara a obtener resultados rápidos.

En el resto de esta sección se presentan algunos ejemplos en donde es posible aplicar técnicas de forma combinada.



### 5.3.2. Ejemplo: puntos de decisión + CE + VL

Es bastante habitual reforzar la cobertura de puntos de decisión con clases de equivalencia y valores límite.

#### Ejemplo 15

A continuación se toma parte de la especificación correspondiente a la alerta ATC OPPOSITE RUNWAYS para explicar en siguientes apartados cómo se van aplicando las distintas técnicas y cómo se diseña un conjunto de casos de prueba que ejercita combinaciones de las situaciones derivadas.

#### EXTRACTO DE LA ESPECIFICACIÓN

##### Requisitos SESAR asociados

a) *Alerta LINE-UP & LINE-UP*

*The SSN server shall trigger an alert (ALARM) when an aircraft is cleared to line up and another aircraft is cleared to line up on the same runway on the opposite threshold.*

b) *Alerta LINE-UP & TAKE-OFF*

*The SSN server shall trigger an alert (ALARM) when an aircraft is cleared to line up and another aircraft is cleared to take-off from the opposite threshold of the same runway.*

c) *Alerta LINE-UP & LAND*

*The SSN server shall trigger an alert (ALARM) when an aircraft is cleared to line up and another aircraft is cleared to land at the opposite threshold.*

d) *Alerta TAKE-OFF & TAKE-OFF*

*The SSN server shall trigger an alert (ALARM) when an aircraft is cleared to take-off on a runway and another aircraft is cleared to take-off on the same runway but on opposite threshold.*

e) *Alerta TAKE-OFF & LAND*

*The SSN server shall trigger an alert (ALARM) when an aircraft is cleared to take-off on a runway and another aircraft is cleared to land on the opposite threshold.*

f) *Alerta LAND & LAND*

*The SSN server shall trigger an alert (ALARM) when two aircraft are cleared to land on the same runway at opposite threshold.*

##### Descripción extraída del diseño funcional

...

*Detección del conflicto:*

*Si para una pareja de móviles se cumplen las condiciones siguientes:*

- *Alerta LINE-UP & LINE-UP*
- *Móvil 1 y Móvil 2 son Aeronaves.*
- *Móvil 1 ha recibido autorización Line-up (aparece en PV\_Clearances\_List).*
- *Móvil 2 ha recibido autorización Line-up (aparece en PV\_Clearances\_List).*
- *Runway asignada para despegue (Departure\_Runway) de móvil 1 y Runway asignada para despegue (Departure\_Runway) de móvil 2 son opuestas (getIdPistaOpuesta).*

*Esta alerta no se generará si para los mismos móviles se está produciendo una alerta de tipo ATC-Opposite Runways prioritaria. Ver cuadro de prioridades al final de este apartado.*



- *Alerta LINE-UP & TAKE-OFF*
- *Móvil 1 y Móvil 2 son Aeronaves.*
- *Móvil 1 ha recibido autorización Line-up (aparece en PV\_Clearances\_List).*
- *Móvil 2 ha recibido autorización Take-off (aparece en PV\_Clearances\_List).*

Se excluirá si ya ha despegado y ha abandonado la pista => está en Fase DESPEGUE, y a la vez AreaDespegando = ninguna.

- Runway asignada para despegue (Departure\_Runway) de móvil 1 y Runway asignada para despegue (Departure\_Runway) de móvil 2 son opuestas (getIdPistaOpuesta).

Esta alerta no se generará si para los mismos móviles se está produciendo una alerta de tipo ATC-Opposite Runways prioritaria. Ver cuadro de prioridades al final de este apartado.



- Alerta TAKE-OFF & TAKE-OFF
- Móvil 1 y Móvil 2 son Aeronaves.
- Móvil 1 ha recibido autorización Take-off (aparece en PV\_Clearances\_List).  
Se excluirá si ya ha despegado y ha abandonado la pista => está en Fase DESPEGUE, y a la vez AreaDespegando = ninguna.
- Móvil 2 ha recibido autorización Take-off (se excluirá si ya ha despegado y ha abandonado la pista => está en Fase DESPEGUE, y a la vez AreaDespegando = ninguna)
- Runway asignada para despegue (Departure\_Runway) de móvil 1 y Runway asignada para despegue (Departure\_Runway) de móvil 2 son opuestas (getIdPistaOpuesta).

Esta alerta es prioritaria sobre otras alertas de tipo ATC-Opposite Runways. Ver cuadro de prioridades al final de este apartado.



- Alerta TAKE-OFF & LAND
- ...
- Alerta LAND & LAND
- ...
- Alerta LINE-UP & LAND
- ...

Se generará un conflicto tipo "ATC-Opposite runways" y nivel de alerta "Alarm".

**Tabla de prioridades en la generación de conflictos**

ALERTAS GRUPO ATC- OPPOSITE RUNWAYS			
Situación 1:	Despegue	Despegue	Orden de prioridad
Combinación de autorizaciones para generación de Alertas	Take-off	Take-off	↑ Prioritaria
	Line-up	Take-off	
	Line-up	Line-up	
Situación 2:	Despegue	Aterrizaje	Orden de prioridad
Combinación de autorizaciones	Take-off	Landing	↑ Prioritaria

<i>para generación de Alertas</i>	<i>Line-up</i>	<i>Landing</i>	
<b>Situación 3:</b>	<b>Aterrizaje</b>	<b>Aterrizaje</b>	<b>Orden de prioridad</b>
<i>Combinación de autorizaciones para generación de Alertas</i>	<i>Landing</i>	<i>Landing</i>	<i>N/A</i>

...  
**Figura 58.** Especificación de alerta OPPOSITE RUNWAYS

### 5.3.2.1. Derivar situaciones de prueba con cada técnica

#### 5.3.2.1.1. Aplicar cobertura de puntos de decisión

Se puede ver que en principio sería posible aplicar cobertura de puntos de decisión sobre la condición general para la detección del conflicto de cada tipo de alerta.

Normalmente el desarrollador tendrá como entrada una decisión sobre la intensidad de pruebas que es necesario aplicar a las distintas tareas de desarrollo que tiene asignadas. Esto se traducirá en un conjunto de técnicas que debería utilizar, pero pueden darse casos donde se permita aplicar distintas variantes de una misma técnica (en las técnicas presentadas, las variantes de cobertura de puntos de decisión o el nivel seleccionado en la técnica de cobertura de caminos).

Por ejemplo, para una determinada tarea se le indica al desarrollador que puede aplicar MCDC o MC, que son variantes de la técnica de cobertura de puntos de decisión, asociadas normalmente a una intensidad de pruebas alta. En estas circunstancias, el desarrollador debe escoger una de esas 2 variantes.

Un criterio que puede ser bastante útil a la hora de seleccionar la técnica más adecuada es intentar prever la *rentabilidad* que se va a obtener, referida a la cantidad e importancia de los defectos detectados con las situaciones de prueba que se van a ejercitar.

Si se deriva un pequeño conjunto de situaciones de prueba que detecta ciertos defectos bastante difíciles de encontrar, dicho conjunto de situaciones será muy rentable.

Si por el contrario se genera un conjunto muy grande de situaciones, habrá que valorar si merece la pena el esfuerzo necesario para ejercitarlas todas, ya que si finalmente la mayoría de defectos detectados fuesen triviales entonces probablemente el esfuerzo invertido podría haberse dedicado a otras tareas.

No resulta práctico ni sencillo prever cuántos defectos van a encontrar las pruebas ni su severidad, pero en cambio si es abordable prever el esfuerzo necesario para diseñar las pruebas que permitan encontrarlos con mayor probabilidad. Con esa estimación es posible tomar una decisión sobre la variante que es más conveniente aplicar.

A continuación se describen los pasos seguidos para seleccionar la técnica más adecuada atendiendo al criterio anterior:

En primer lugar se construye una tabla de decisión para las alertas que se pueden producir cuando los dos móviles están en maniobra de despegue (en la test basis, los tipos de alerta correspondientes a la situación 1 de la tabla de prioridades de la Figura 58):

De las condiciones que intervienen en el tipo de alerta LINE-UP & LINE-UP hay 2 que merecen especial atención: *Móvil 1 ha recibido autorización LINE-UP* y *Móvil 2 ha recibido autorización LINE-UP* (aparecen condiciones similares en los tipos de alerta LINE-UP & TAKE-OFF y TAKE-OFF & TAKE-OFF).

El fragmento de texto “ha recibido” de las condiciones anteriores especifica que, en el instante del cálculo del conflicto, se cumple que cada móvil ha recibido *en algún momento anterior* la autorización que hace cierta la condición. Lo anterior da pie a tener en cuenta aquellas acciones relacionadas con las autorizaciones que pudiesen afectar al cumplimiento de esas 2 condiciones.

En este caso particular, podría ser interesante probar el efecto que tiene la *revocación* de las autorizaciones que hacen ciertas las condiciones, dado que en este caso el efecto que tiene revocar es que a todos los efectos es como si “la autorización no se hubiese dado”.

Por esto resulta interesante añadir condiciones sobre la revocación de autorizaciones al conjunto de condiciones que se deben cumplir para detectar el conflicto.

En el supuesto de que la revocación de una autorización no se tradujese en considerar que “la autorización asociada no se dio”, entonces habría que tener en cuenta las revocaciones igualmente, pero en este caso el enfoque sería comprobar que la acción de revocar no tiene influencia en la detección del conflicto.

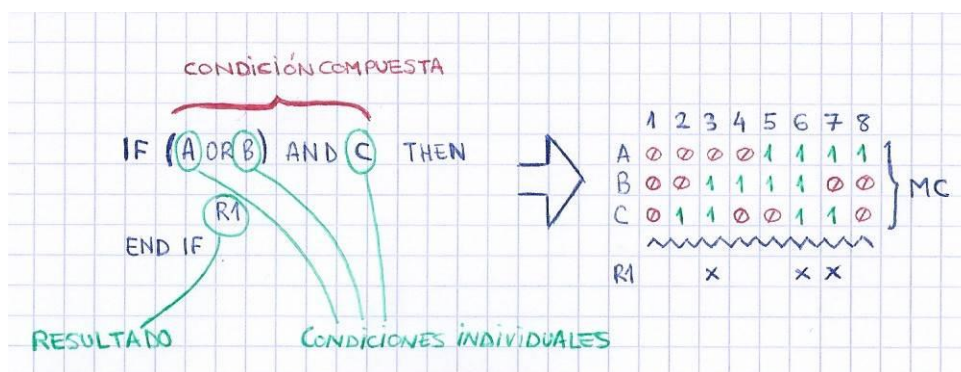
En general, cuando se pretende “comprobar si algo que no debe influir en una salida influye realmente o no”, se está hablando de hacer *pruebas negativas*.

El matiz anterior en relación a cómo se tiene que interpretar si “haber recibido una autorización” se invalida o no en función de la revocación correspondiente es un ejemplo de un suceso bastante habitual: a veces el diseño de pruebas no se basa exclusivamente en la información que aparece los requisitos (que además pueden estar incompletos en el momento de hacer el diseño).

Para probar en detalle atendiendo a cobertura de puntos de decisión se suele aplicar MCDC, pero también se puede considerar la variante de Múltiple Condición (MC).

Cuando se aplica MC normalmente se habla de construir una “tabla de decisión”, donde además de identificar las condiciones que intervienen para hacer las combinaciones, es necesario identificar los resultados que se producen.

A continuación se explica cómo se construye una tabla de decisión sobre un ejemplo sintético, en este caso la expresión compuesta (A OR B) AND C:



**Figura 59.** Construcción de una tabla de decisión

A las condiciones identificadas (A, B y C) se les aplica MC (Múltiple Condición) obteniéndose 8 situaciones de prueba, asignando los resultados identificados a aquellas situaciones de prueba que lo producen.

Para compactar el conjunto de situaciones derivadas (recordar que con MC se derivan del orden de  $2^N$  situaciones, donde N es el número de condiciones), no se tienen en cuenta las 2 primeras condiciones de cada tipo de alerta: *Móvil 1 y Móvil 2 son Aeronaves*.

Alerta LINE-UP & LINE-UP		No factibles																																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
móvil 1 ha recibido aut. LU		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
móvil 2 ha recibido aut. LU		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
runways abiertos		0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
se revoca aut. LU móvil 1		0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
se revoca aut. LU móvil 2		0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Generar alerta (condicionada a prioridades)																																		

Figura 60. Tabla de decisión alerta OPPOSITE RUNWAYS por LU (LINE UP)

Del total de situaciones derivadas para la alerta LINE-UP & LINE-UP, las situaciones 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 26, 27, 30 y 31 no serían factibles debido a que representan escenarios en los que se revoca una autorización que no se ha dado (se ha verificado que eso no es posible). Para esta alerta podría ser razonable derivar las situaciones de prueba que requiere utilizar una tabla de decisión (2<sup>5</sup> situaciones).

Alerta LINE-UP & TAKE-OFF	1	2	3	...	128
móvil 1 ha recibido aut. LU	...				
móvil 2 ha recibido aut. TO	...				
móvil 2. fase 1 = DESPEGUE	...				
móvil 2. Area Despegando != ninguna	...				
runways abiertos	...				
se revoca aut. LU móvil 1	...				
se revoca aut. TO móvil 2	...				
Generar alerta (condicionada a prioridades)	...				

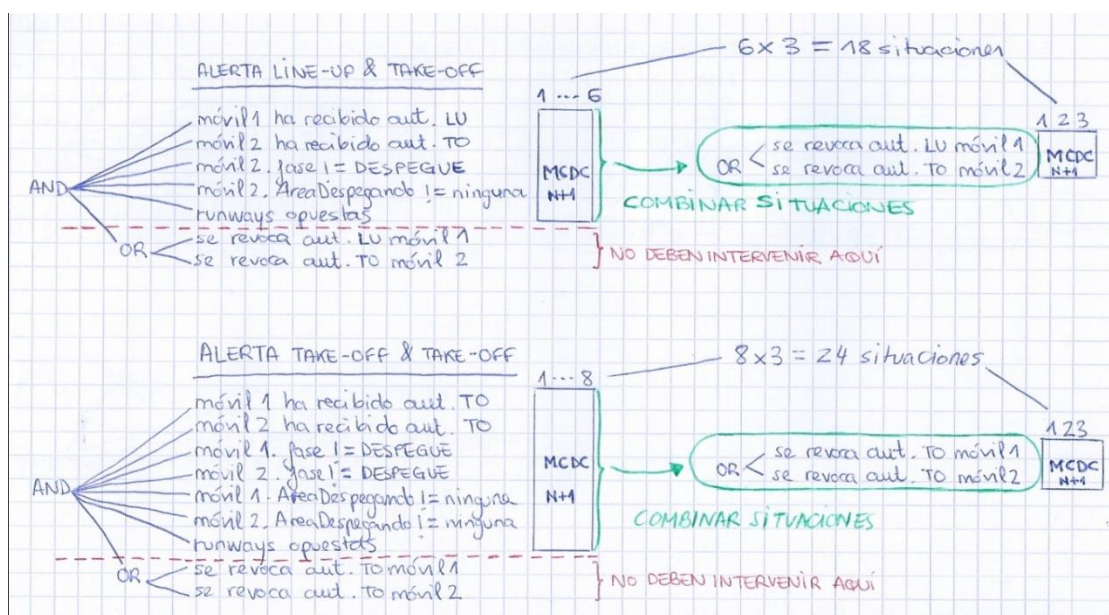
Alerta TAKE-OFF & TAKE-OFF	1	2	3	...	128	...	512
móvil 1 ha recibido aut. TO	...						
móvil 2 ha recibido aut. TO	...						
móvil 1. fase 1 = DESPEGUE	...						
móvil 2. fase 1 = DESPEGUE	...						
móvil 1. area Despegando != ninguna	...						
móvil 2. area Despegando != ninguna	...						
runways abiertas	...						
se revoca aut. TO móvil 1	...						
se revoca aut. TO móvil 2	...						
Generar alerta	...						

Figura 61. Tablas de decisión alerta por autorizaciones LU & TO y TO & TO

En cambio, en la Figura 61 se puede observar que en el caso de las alertas LINE-UP & TAKE-OFF (2<sup>7</sup> situaciones → 128 situaciones, donde el exponente viene de las 5 condiciones de la especificación + 2 condiciones sobre revocaciones) y TAKE-OFF & TAKE-OFF (2<sup>9</sup> situaciones → 512 situaciones) el esfuerzo necesario para ejercitar todas las situaciones de prueba será demasiado grande (incluso habiendo excluido del total aquellas situaciones de prueba que pudiesen no ser factibles).

Por este motivo se podría descartar utilizar tablas de decisión para las alertas LINE-UP & TAKE-OFF y TAKE-OFF & TAKE-OFF (porque el esfuerzo previsto no parece muy rentable) y se podría optar por utilizar MCDC.

Es importante tener en cuenta que las condiciones sobre la revocación de las autorizaciones que se añadieron en la tabla de decisión se podían añadir porque en la variante MC no tiene en cuenta cómo se componen las condiciones mediante los operadores lógicos. En MCDC sí que se tiene en cuenta como están compuestas las condiciones, así que no deberían intervenir en la expresión sobre la que se aplica, ya que esto iría en contra de la especificación:



**Figura 62.** Combinar situaciones de MCDC con situaciones de revocaciones

Teniendo en cuenta este último detalle, se aplicaría MCDC de la forma habitual, pero combinando además las situaciones que se deriven con las situaciones relativas a las revocaciones.

### 5.3.2.1.2. Añadir clases de equivalencia

Atendiendo a la tabla de prioridades de las alertas, se podría además combinar las situaciones de pruebas derivadas anteriormente aplicando cobertura de puntos de decisión con las situaciones de derivadas aplicando clases de equivalencia sobre “la presencia de alertas prioritarias en el momento del cálculo del conflicto”.

A continuación se representan las clases de equivalencia identificadas por medio de una escala de tiempo que refleja:

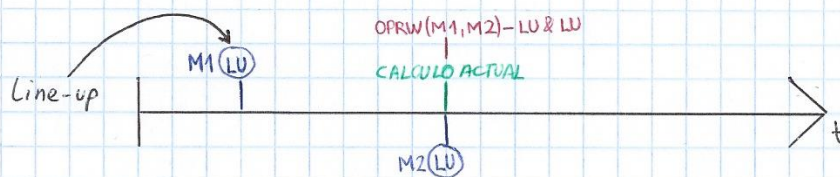
- el instante de tiempo en que se dan las autorizaciones a los móviles involucrados.
- el momento de cálculo actual.
- el tipo conflicto OPRW que se genera, suponiendo que se cumple el resto de condiciones que intervienen en la detección del tipo de conflicto correspondiente.

1. Presencia de alertas en el momento de cálculo del conflicto

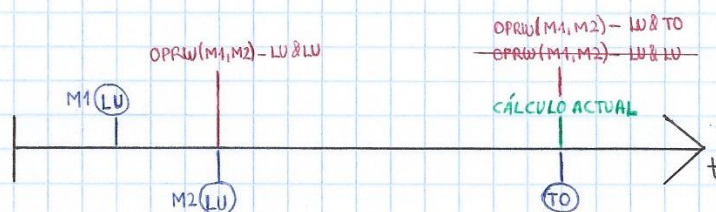
1.1. Alerta LINE-UP & LINE-UP

1.1.1. Teniendo en cuenta solo los mismos 2 móviles

1.1.1.1. No hay otras prioridades



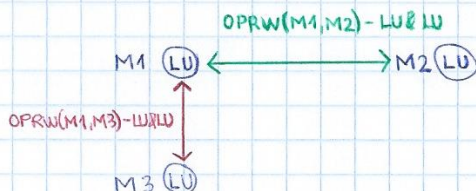
1.1.1.2 Hay alguna prioridad



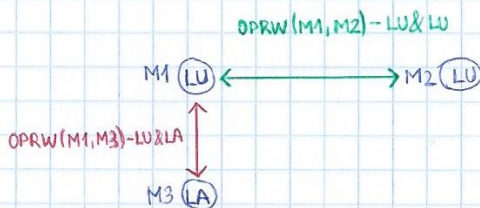
1.1.2. Teniendo en cuenta otros móviles distintos a los 2 sobre los que se hace el cálculo del conflicto

1.1.2.1. No hay prioridades

1.1.2.1.1. Alguna de la misma prioridad



1.1.2.1.2. Alguna de la Situación 2 de la tabla de prioridades (es decir, alguna de menor prioridad)



1.1.2.2. Hay prioridades

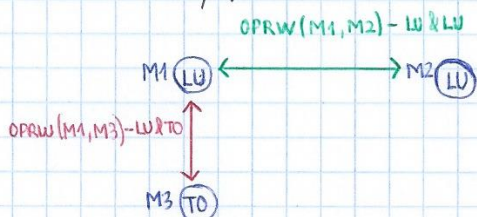
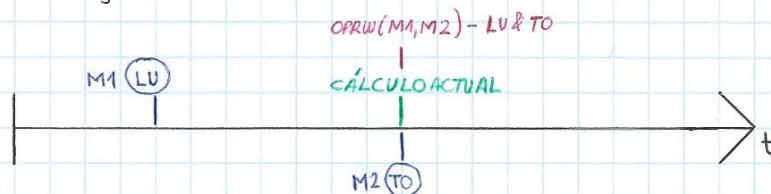


Figura 63 Clases de equivalencia alerta OPPOSITE RUNWAYS por LU & LU

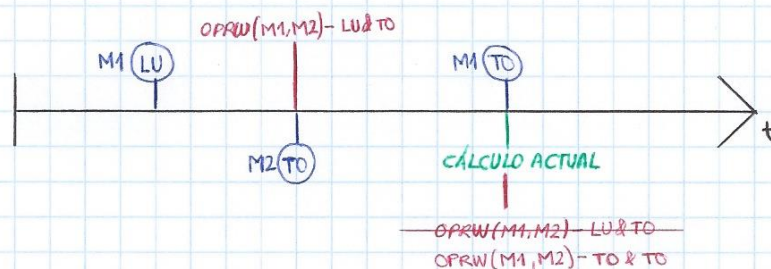
1.2. Alerta LINE-UP & TAKE-OFF

1.2.1. Teniendo en cuenta solo los mismos 2 móviles

1.2.1.1. No hay prioridades



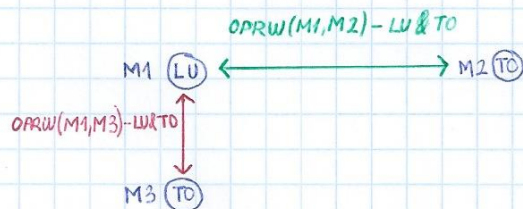
1.2.1.2. Hay alguna prioridad



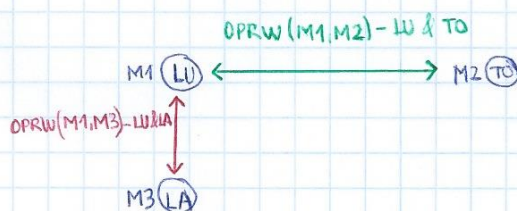
1.2.2. Teniendo en cuenta además otros móviles distintos a los 2 sobre los que se hace el cálculo del conflicto

1.2.2.1. No hay prioridades

1.2.2.1.1. Alguna de la misma prioridad



1.2.2.1.2. Alguna de la situación 2



1.2.2.2. Hay prioridades

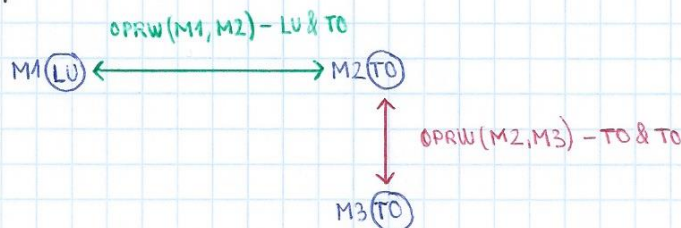


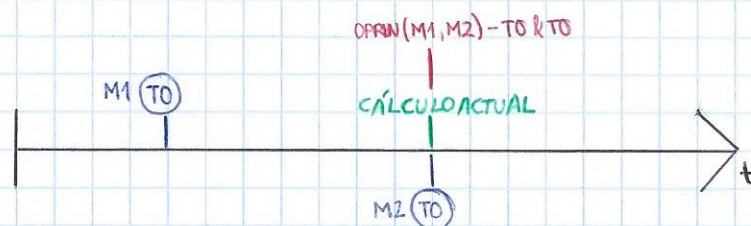
Figura 64 Clases de equivalencia alerta OPPOSITE RUNWAYS por LU & TO



### 1.3. Alerta TAKE-OFF & TAKE-OFF

#### 1.3.1. Teniendo en cuenta solo los mismos 2 móviles

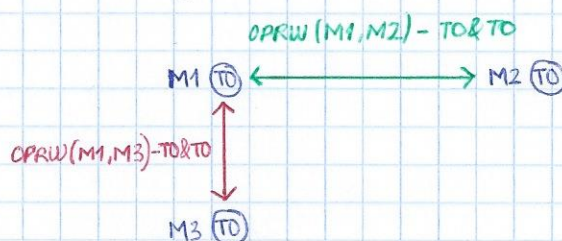
##### 1.3.1.1. No hay prioridades



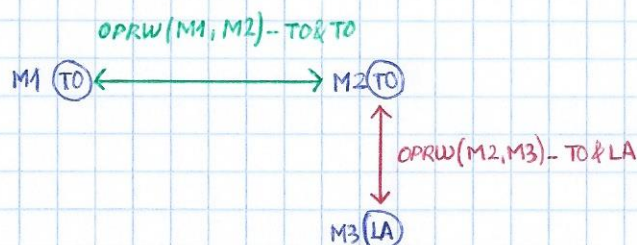
#### 1.3.2. Teniendo en cuenta además otros móviles distintos a los dos sobre los que se hace el cálculo del conflicto

##### 1.3.2.1. No hay prioridades

##### 1.3.2.1.1. Alguna de la misma prioridad



##### 1.3.2.1.2. Alguna de la situación 2



**Figura 65.** Clases de equivalencia alerta OPPOSITE RUNWAYS por TO & TO

Merece especial atención el grupo de situaciones de prueba en donde se tiene en cuenta otros móviles aparte de los dos involucrados en el cálculo del conflicto.

La especificación indicaba que para generar una alerta solo se debería tener en cuenta alertas prioritarias sobre “los mismos dos móviles”, de modo que considerar la influencia que tienen otros móviles en el cálculo es otra forma de hacer pruebas negativas basándose en la especificación.

Este grupo de situaciones tiene como objetivo probar si la existencia de alertas que involucren a alguno de dichos móviles con “otros móviles distintos” influye o no en el cálculo.

#### 5.3.2.1.3. Completar con valores límite

Adicionalmente también se podría hacer la combinación de las situaciones anteriores con valores límite, teniendo en cuenta el “número de alertas máximo que puede tratar el sistema de forma simultánea” (parámetro que se indica en la configuración).

Suponiendo que por configuración el sistema puede tratar un máximo de 32 alertas ATC, se necesita confeccionar un escenario en el que se den las condiciones para que se esté detectando dicho límite de conflictos en el mismo instante en que ocurre un nuevo conflicto.

Puede llegar a ser muy laborioso preparar este tipo de escenarios, de modo que puede ser recomendable seguir alguna estrategia que minimice el coste de preparación.

Una posible estrategia para abordar este problema podría ser la siguiente:

- Dado que existen varios tipos de alertas ATC y es necesario generar 32 simultáneas, se podría intentar aprovechar para generar al menos una de cada tipo:
  - a. OPPOSITE RUNWAYS (OPRW)
  - b. OPPOSITE STOP POINTS (OPSP)
  - c. SAME/NEARBY HOLDING POINTS (NEHP)
  - d. SAME RUNWAY (SMRW)
  - e. CROSSED RUNWAYS (CRRW)
  - f. CONVERGING RUNWAYS (CORW)
  - g. PARALLEL RUNWAYS (PARW)
- Para que sea posible generar una alerta de cada uno de los tipos anteriores habrá que prestar atención a la topología de aeródromo necesaria para poder probarlas todas juntas:
  - a. La alerta CRRW requiere de una topología en la que existan pistas cruzadas
  - b. La alerta CORW requiere que existan pistas convergentes
  - c. La alerta PARW requiere que existan pistas paralelas

Teniendo en cuenta que el sistema debe satisfacer cualquier topología de aeródromo, el motivo por el que se tiene en cuenta el aspecto de la topología es evitar que para ejercitar todas las situaciones derivadas sea necesario ejecutar pruebas en varios aeródromos distintos, ya que eso probablemente implicará repetición de tareas de preparación del entorno.

- Dado que algunos de los tipos de alertas anteriores se pueden generar debido a varias combinaciones de autorizaciones, puede ser conveniente identificar aquellas que impliquen menor preparación. Por ejemplo, la alerta CRRW se puede detectar debido a los siguientes combinaciones de autorizaciones:
  - a. TAKE-OFF & TAKE-OFF
  - b. TAKE-OFF & LAND
  - c. LAND & LAND

De estas combinaciones, las que requieren autorización LAND implican tener una aeronave en maniobra de aterrizaje, lo cual requiere mayor tiempo de preparación en el entorno de pruebas en comparación con la preparación necesaria para incluir en la simulación una aeronave en maniobra de despegue.

Debido a esto se podría optar por la combinación TAKE-OFF & TAKE-OFF frente a las otras combinaciones, ya que se necesitarán 2 aeronaves en maniobra de despegue y requerirá menor preparación que las otras.

- Una vez se preparan las 32 alertas simultáneas será necesario añadir una nueva alerta para observar el comportamiento del sistema cuando se supera este límite. En este caso se tratará de una alerta de tipo OPRW, que es la alerta ATC sobre la que se estaba aplicando la combinación de técnicas.

Una variante de la estrategia podría consistir en no intentar ejercitar todos los tipos de alertas ATC, de forma que se prepararían los datos necesarios para forzar la ocurrencia simultánea

de 33 alertas de un solo tipo, en este caso OPRW. De esta manera se podría ahorrar cierto trabajo de preparación y la topología de aeródromo necesaria sería menos restrictiva, ya que solo se necesitan 2 pistas para este tipo de alerta.

En general, cuando es necesaria una gran labor de preparación y existen varias alternativas posibles para dejar el sistema en el estado inicial, puede ser útil identificar aquellas alternativas menos restrictivas y que impliquen menor esfuerzo/tiempo de preparación.

### 5.3.2.2. Diseño de los casos de prueba

A continuación se diseñarían casos de prueba que ejerciten combinaciones de las situaciones de prueba derivadas aplicando las 3 técnicas:

En este caso se propone un único caso de prueba que ejercita todas las situaciones de prueba derivadas (las que eran factibles) para las alertas correspondientes a la situación 1 de la tabla de prioridades (Despegue-Despegue):

- Cobertura de condiciones (tabla de decisión y MCDG): se ejercita en los pasos de prueba 1 a 8
- VL
  - a. VL=0: se ejercita en los pasos de prueba 1 a 7
  - b. VL=1: se ejercita en los pasos de prueba 4 a 5
  - c. VL=32: se ejercita en el paso de prueba 15
  - d. VL=Forzar 33: se ejercita en el paso de prueba 16
- CE
  - a. LINE-UP & LINE-UP: se ejercita en los pasos de prueba 6,9 y 10
  - b. LINE-UP & TAKE-OFF: se ejercita en los pasos de prueba 10 a 13
  - c. TAKE-OFF & TAKE-OFF: se ejercita en los pasos de prueba 11 a 14

Al conjunto de situaciones derivadas habría que añadir las correspondientes a las situaciones 2 y 3 de la tabla de prioridades de la Figura 58, aunque estas no se desarrollan aquí:

CP01	Paso 1	Paso 2	Paso 3	Paso 4
Entrada	Num. Alertas ATC=0 M1 parada en stand M2 parada en stand	Num. Alertas ATC=0	Num. Alertas ATC=0 M2 tiene aut. line-up	Num. Alertas ATC=0
Descripción	M1 y M2 tienen ruta para despegar por pistas opuestas Mantener paradas M1 y M2	M2. autorización = line-up	Revocar aut. line-up de M2	M2. autorización = line-up M1. autorización = line-up Revocar aut. line-up de M1
Salida	No hay alertas OPRW Num. Alertas ATC=0	No hay alertas OPRW Num. Alertas ATC=0	No hay alertas OPRW Num. Alertas ATC=0	OPRW(M1, M2) hasta revocación de M1. Num. Alertas ATC=0
Ejercita	Tabla decisión: 1, 8 VL: 0	Tabla decisión: 9 VL: 0	Tabla decisión: 10 VL: 0	Tabla decisión: 21 VL: 0, 1
	Paso 5	Paso 6	Paso 7	Paso 8
Entrada	Num. Alertas ATC=0 M2 tiene aut. line-up	Num. Alertas ATC=0	Num. Alertas ATC=0 M3 parada en stand M1 tiene aut. line-up	Num. Alertas ATC=0 M1 tiene aut. line-up
Descripción	M1. autorización = line-up Revocar aut. line-up de M2 Revocar aut. line-up de M1	M2. autorización = line-up M1. autorización = line-up Revocar aut. line-up de M2	M3 tiene ruta PV para despegar por misma pista de M1. Mantener parada M3	Revocar aut. line-up de M1. M3. autorización = line-up Revocar aut. line-up de M3
Salida	OPRW (M1, M2) hasta revocaciones. Num. Alertas ATC=0	OPRW (M1, M2) hasta revocación Num. Alertas ATC=0	Num. Alertas ATC=0	OPRW (M2, M3) hasta revocar aut. de M3
Ejercita	Tabla decisión: 22, 24 VL: 0, 1	Tabla decisión: 23 VL: 0 CE: LU & LU - mismos móviles - No hay prioridades	Tabla decisión: 16, 15 VL: 0	Tabla decisión: 17, 18, 20, 19
	Paso 9	Paso 10	Paso 11	Paso 12
Entrada	Num. Alertas ATC=0 Preparar datos aterrizaje M4 para que tenga pista opuesta a M1	Num. Alertas ATC=2 OPRW (M1, M4) OPRW (M1, M2) M1 tiene aut. line-up M2 tiene aut. line-up M4 tiene aut. landing	Num. Alertas ATC=2 OPRW (M1, M2) OPRW (M2, M3) M1 tiene aut. line-up M2 tiene aut. line-up M3 tiene aut. take-off	Num. Alertas ATC=2 OPRW (M1, M2) OPRW (M2, M3) M4 tiene aut. line-up M2 tiene aut. take-off M3 tiene aut. take-off M5 parada en stand
Descripción	M4. autorización = landing M1. autorización = line-up M2. autorización = line-up	M3. autorización = take-off Esperar hasta que M4 esté en fase TAXI	M2. autorización = line-up	M5 tiene ruta PV para despegar por misma pista que M2 M5. autorización = take-off
Salida	OPRW (M1, M4) - LU & LA OPRW (M1, M2) - LU & LU Num. Alertas ATC=2	OPRW (M1, M2) - LU & LU OPRW (M2, M3) - LU & TO	OPRW (M1, M2) - LU & TO OPRW (M2, M3) - TO & TO Num. Alertas ATC=2	OPRW (M1, M2) - LU & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - LU & TO Num. Alertas ATC=3
Ejercita	CE: LU & LU - otros - No hay prioridades - Alguna de la situación 2	CE: LU & LU - otros - Hay prioridades CE: LU & TO - mismos - No hay prioridades	CE: LU & TO - mismos - Hay alguna prioridad CE: LU & TO - otros - Hay prioridades CE: TO & TO - mismos - No hay prioridades	CE: TO & TO - mismos - No hay prioridades CE: LU & TO - otros - misma prioridad CE: LU & TO - otros - Hay prioridades

	Paso 13	Paso 14	Paso 15	Paso 16
Entrada	Num. Alertas ATC = 3 OPRW (M1, M2) - LU & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - LU & TO Reparar datos almacenaje M6 para que pistas de M6 y M1 sean opuestas	Num. Alertas ATC = 4 OPRW (M1, M2) - LU & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - LU & TO OPRW (M1, M6) - LU & LA	Num. Alertas ATC = 4 OPRW (M1, M2) - TO & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - TO & TO OPRW (M1, M6) - TO & LA M7-M35 paradas en stand	Num. Alertas ATC = 32 OPRW (M1, M2) - TO & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - TO & TO OPRW (M1, M7) - TO & TO ... OPRW (M1, M35) - TO & TO M36 parada en stand
Descripción	M6. autorización = landing	M1. autorización = take-off	Esperar hasta que M6 esté en fase TAXI M7-M35 tienen ruta PV para despegar por pista opuesta a M1 M7-M35 autorización = take-off	M36 tiene ruta PV para despegar por pista opuesta a M1 M36. autorización = take-off
Salida	Num. Alertas ATC = 4 OPRW (M1, M2) - LU & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - LU & TO OPRW (M1, M6) - LU & LA	Num. Alertas ATC = 4 OPRW (M1, M2) - TO & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - TO & TO OPRW (M1, M6) - TO & LA	Num. Alertas ATC = 32 OPRW (M1, M2) - TO & TO OPRW (M1, M5) - TO & TO OPRW (M2, M3) - TO & TO OPRW (M1, M7) - TO & TO ... OPRW (M1, M35) - TO & TO	Num. Alertas ATC = 32 OPRW (M1, M2) - TO & TO OPRW (M2, M3) - TO & TO OPRW (M1, M5) - TO & TO OPRW (M1, M7) - TO & TO ... OPRW (M1, M35) - TO & TO
Ejercita	CE: LU & TO - otros - alguna situación 2, LU & TO - otros - Hay prioritarias, TO & TO - mismos - No hay prioritarias	CE: TO & TO - otros - no hay prioritarias, misma prioridad, TO & TO - otros - no hay prioritarias - alguna sit. 2	VL = 32	VL: Forzar 33

**Figura 66.** Detalle caso de prueba que ejercita puntos de decisión + CE + VL

En aquellos casos en los que se observe que el número de pasos de prueba necesarios empieza a ser inmanejable, se podría optar por separar en varios casos de prueba, agrupando en función de alguno de los siguientes criterios:

- **preparación previa común:** si un conjunto de situaciones requiere específicamente la preparación de un gran conjunto de datos, o de una gran labor de configuración del estado inicial del entorno de pruebas.

En este ejemplo, una alternativa podría consistir en separar en un caso de prueba independiente las situaciones que ejercitan los VL.

- **evitar enmascaramiento entre situaciones de prueba:** por ejemplo para ejercitar las situaciones relativas a valores límite para el máximo número de alertas, se podría optar por diseñar un caso de prueba aparte.

El motivo de esto sería evitar que esta situación en la que se maneja el máximo de alertas “enmascare” los fallos que se podrían detectar al ejercitar otras.

Supongamos que se decide que los primeros pasos de prueba deben ejercitar todas las situaciones de VL: podría suceder, por ejemplo, que cuando ya se está detectando el máximo de 32 alertas se ejecute un nuevo paso de prueba, donde se pretende ejercitar alguna situación correspondiente a CE.

Dado que las ocurrencias de alertas que exceden el límite se descartan, si hubiese un fallo asociado a la situación que pretende ejercitar CE, este no sería detectado en la salida del paso de prueba porque el máximo de alertas lo enmascararía. En el ejemplo desarrollado se ha evitado el enmascaramiento dejando las situaciones correspondientes al máximo de alertas para los últimos pasos de prueba.

### 5.3.3. Ejemplo: cobertura de caminos + tabla de decisión

En el Ejemplo 14 se presentaba cómo inferir un modelo de transición de estados para los escenarios individuales descritos en la test basis de la alerta RED STOP BAR, que facilitaba la derivación de situaciones de prueba aplicando la técnica de cobertura de caminos de nivel 2.

#### Ejemplo 16

Atendiendo al siguiente fragmento adicional de test basis para esta alerta, sería posible complementar la cobertura de caminos anterior confeccionando una tabla de decisión para las condiciones que intervienen en la detección del conflicto:

#### EXTRACTO DE LA ESPECIFICACIÓN

```

...
//CÁLCULO DEL CONFLICTO
//Si el móvil no estaba en alerta "Red Stop bar" en la anterior actualización, se comprueba si su posición en la
nueva actualización sí que provoca una alerta.
Si movil.AlertaRSB = false
    Si movil.NombreSBsuperada y movil.NombreSBnosuperada no están vacíos
        //Se descartan los móviles que acceden directamente desde el extremo de la Stop bar que se considera
superada, poniendo como condición que el nombre de la Stop bar superada sea el mismo que la no
superada (estuvo antes por delante y se cruzó en su dirección de funcionamiento)
        Si movil.NombreSBsuperada = movil.NombreSBnosuperada
            //Se comprueba si la Stop bar superada está activa
            Si movil.NombreSBsuperada.Active = true,
                //Se comprueba si la Stop bar activó antes de haberla superado
                Si movil.TiempoSBsuperada > movil.NombreSBsuperada.ActivationTime,
                    // El móvil ha superado la Stop bar cuando estaba activa. Se genera el evento de
alerta y se activa el flag de mantención.
                    Generar conflicto "Red Stop Bar"
                    movil.AlertaRSB == true
                    else [si movil.TiempoSBsuperada ≤ movil.NombreSBsuperada.ActivationTime -> el móvil ha
superado la Stop bar cuando estaba inactiva, no se hace nada.
                    else [si movil.NombreSBsuperada.Active = false] -> la Stop bar superada está inactiva, no se hace
nada.
                    else [si movil.NombreSBsuperada ≠ movil.NombreSBnosuperada] -> no se hace nada
                    else [si movil.NombreSBsuperada y/o movil.NombreSBnosuperada está vacío] -> no se hace nada
//Si el móvil estaba en alerta "Red Stop bar" en la anterior actualización, se comprueba si la Stop bar sigue
activa o se ha desactivado en la nueva actualización.
else [si movil.AlertaRSB = true]
//Se comprueba si la Stop bar sigue activa
    Si movil.NombreSBsuperada.Active = true,
        //Se genera el evento de alerta "Red Stop Bar" y el flag de mantención se mantiene true (tal y como
estaba).
        Generar conflicto "Red Stop bar"
    else [si movil.NombreSBsuperada.Active = false]
        //La Stop bar se ha desactivado. No se genera evento de alerta, y se establece a false en indicador de
alerta.
        movil.AlertaRSB == false

```

#### Nuevos atributos en el móvil:

Para cubrir todas las situaciones posibles y calcular correctamente la alerta se deberán crear los siguientes atributos en el móvil:

- **NombreSBnosuperada:** una cadena, que se inicializará vacía. Se guardará el nombre de la Stop bar que no se ha superado, cuando se detecte que el móvil está en un segmento que contiene una Stop bar

- pero todavía no se ha superado.
- *NombreSBsuperada*: una cadena, que se inicializará vacía. Se guardará el nombre de la Stop bar que se haya superado, cuando se detecte que el móvil está en un segmento que contiene una Stop bar y ya se ha superado.
  - *IntentosSBsuperada*: un contador. Se irá sumando uno cada vez que en una actualización se detecte que se ha superado una Stop bar. Servirá para asegurarse de que no es una actualización de la posición incorrecta, ya que se esperará al número de intentos configurado (*Nmax\_IntentosSBsuperada*) para confirmarse que se ha superado la Stop bar y se guarde en *NombreSBsuperada*.
  - *TiempoSBsuperada*: un valor temporal (hora:min:seg). Se inicializará vacío y se almacenará el instante en el que se confirma que el móvil ha superado la Stop bar.
  - *AlertaRSB*: un booleano (true/false). Sirve para mantener la alerta “Red Stop Bar” activada cuando se ha detectado que el móvil ha superado una Stop bar en rojo.
- ...

**Figura 67.** Especificación de la función de cálculo de alerta RED STOP BAR

Partimos de la suposición de que se ha podido verificar que la funcionalidad que comprueba si un móvil supera una barrera ha sido suficientemente probada y su funcionamiento es el esperado.

A partir de la especificación anterior se podría construir una tabla de decisión con los siguientes parámetros:

- Condiciones
  1. *AlertaRSB*==FALSE
  2. *NombreSBsuperada* no está vacío
  3. *NombreSBnosuperada* no está vacío
  4. *NombreSBsuperada.Active* == TRUE
  5. *TiempoSBsuperada* > *NombreSBnosuperada.ActivationTime*
  6. *NombreSBsuperada* == *NombreSBnosuperada*
- Resultados
  - Generar conflicto RSB
  - *AlertaRSB* = TRUE
  - *AlertaRSB* = FALSE

Analizando las condiciones que intervendrían en la tabla de decisión se puede observar que existe acoplamiento débil entre las condiciones *NombreSBnosuperada no está vacío* y *NombreSBsuperada no está vacío* con la condición *NombreSBsuperada == NombreSBnosuperada*, dado que:

- si *NombreSBnosuperada* no está vacío es TRUE y *NombreSBsuperada no está vacío* es FALSE entonces es imposible que *NombreSBsuperada == NombreSBnosuperada* sea TRUE.
- si *NombreSBnosuperada* no está vacío es FALSE y *NombreSBsuperada no está vacío* es TRUE entonces también es imposible que *NombreSBsuperada == NombreSBnosuperada* sea TRUE.

Este problema se podría resolver quitando la condición *NombreSBnosuperada no está vacío* de la tabla de decisión ya que manteniendo solo las otras 2 condiciones *NombreSBsuperada no está vacío* y *NombreSBsuperada == NombreSBnosuperada*, no se alteraría la lógica de la especificación, dado que:

- cuando *NombreSBsuperada == NombreSBnosuperada* sea TRUE y *NombreSBsuperada no está vacío* es TRUE, entonces necesariamente se cumplirá que *NombreSBnosuperada no está vacío* es TRUE.

- cuando  $NombreSBsuperada == NombreSBnosuperada$  sea FALSE, no será relevante el valor de  $NombreSBnosuperada$  no está vacío.

Teniendo en cuenta estas consideraciones, la tabla de decisión construida podría ser la siguiente (se trata de la misma tabla de decisión, dividida en 2):

**Top Section (Steps 1-16):**

Alerta RSB == FALSE  
 NombreSBsuperada no está vacío  
 NombreSBsuperada.Active == TRUE  
 TiempoSBsuperada > NombreSBsuperada.ActivationTime  
 NombreSBsuperada == NombreSBnosuperada

Generar conflicto RSB  
 Alerta RSB = TRUE  
 Alerta RSB = FALSE

**Bottom Section (Steps 17-32):**

Alerta RSB == FALSE  
 NombreSBsuperada no está vacío  
 NombreSBsuperada.Active == TRUE  
 TiempoSBsuperada > NombreSBsuperada.ActivationTime  
 NombreSBsuperada == NombreSBnosuperada

Generar conflicto RSB  
 Alerta RSB = TRUE  
 Alerta RSB = FALSE

**Figura 68** Tabla decisión condiciones cálculo alerta RED STOP BAR

En este punto es necesario identificar posibles situaciones no factibles: las situaciones 1 a 8 y 25 a 32 no serían factibles porque, si la condición  $NombreSBsuperada$  no está vacío es FALSE, entonces no sería posible evaluar a un valor de verdad (TRUE o FALSE) las condiciones en donde intervienen los parámetros  $NombreSBsuperada.Active$  y  $NombreSBsuperada.ActivationTime$ .

A continuación se podría comenzar el diseño de casos de prueba. El objetivo es construir una secuencia de pasos de prueba que represente un recorrido de un móvil que va atravesando stop bars activas o inactivas, en donde además se tendrá que activar o desactivar stop bars en un determinado instante para ir ejercitando las situaciones de prueba anteriores.

Puede ser útil analizar las situaciones que es necesario cubrir, para ver si es posible encontrar una forma de agruparlas que permita obtener un conjunto compacto de pasos de prueba. El sentido de esto es reducir el esfuerzo necesario para poder ejecutar las pruebas y que los casos de prueba tengan un mantenimiento lo más sencillo posible (los casos de prueba con un número exagerado de pasos no serán fácilmente mantenibles).

A continuación se describen las situaciones de prueba que hay que ejercitar:

	9	10	11	12	13	14	15	16
Alerta act. anterior	SI	SI	SI	SI	SI	SI	SI	SI
Supera SB	SI	SI	SI	SI	SI	SI	SI	SI
Estado actual SB	ACTIVA	ACTIVA	ACTIVA	ACTIVA	INACTIVA	INACTIVA	INACTIVA	INACTIVA
Estado cuando superó SB	INACTIVA	INACTIVA	ACTIVA	ACTIVA	ACTIVA	ACTIVA	INACTIVA	INACTIVA
NombreSBnosuperada != NombreSBsuperada	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	17	18	19	20	21	22	23	24
	NO	NO	NO	NO	NO	NO	NO	NO
	SI	SI	SI	SI	SI	SI	SI	SI
	INACTIVA	INACTIVA	INACTIVA	INACTIVA	ACTIVA	ACTIVA	ACTIVA	ACTIVA
	INACTIVA	INACTIVA	ACTIVA	ACTIVA	ACTIVA	ACTIVA	INACTIVA	INACTIVA
	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE

**Figura 69.** Situaciones de prueba factibles de RED STOP BAR



Una posible forma de agruparlas podría ser por el estado de la alerta en la actualización anterior a la actual (*Alerta act. Anterior*) y por el estado de la stop bar cuando fue superada (*Estado cuando superó SB*).

Lo ideal sería que fuese posible configurar un recorrido del móvil sin restricciones en cuanto a dónde están dispuestas las stop bars, dado que así resultaría más sencillo confeccionar la secuencia de pasos de prueba. En la realidad las pruebas se tendrán que realizar en uno o varios aeródromos, que podrían tener topologías que impidiesen ejercitar algunas situaciones concretas.

Suponiendo que no hubiese restricciones, se podría diseñar el caso de prueba CP01 representa el recorrido de un móvil en donde:

- las stop bars que aparecen en rojo representan que estaban activas cuando el móvil las superó.
- las stop bars resaltadas en negro representan que estaban inactivas cuando las superó.
- las situaciones cubiertas se resaltan en color verde.
- el trazo continuo rojo representa la parte del recorrido en el que se genera alerta para el móvil.

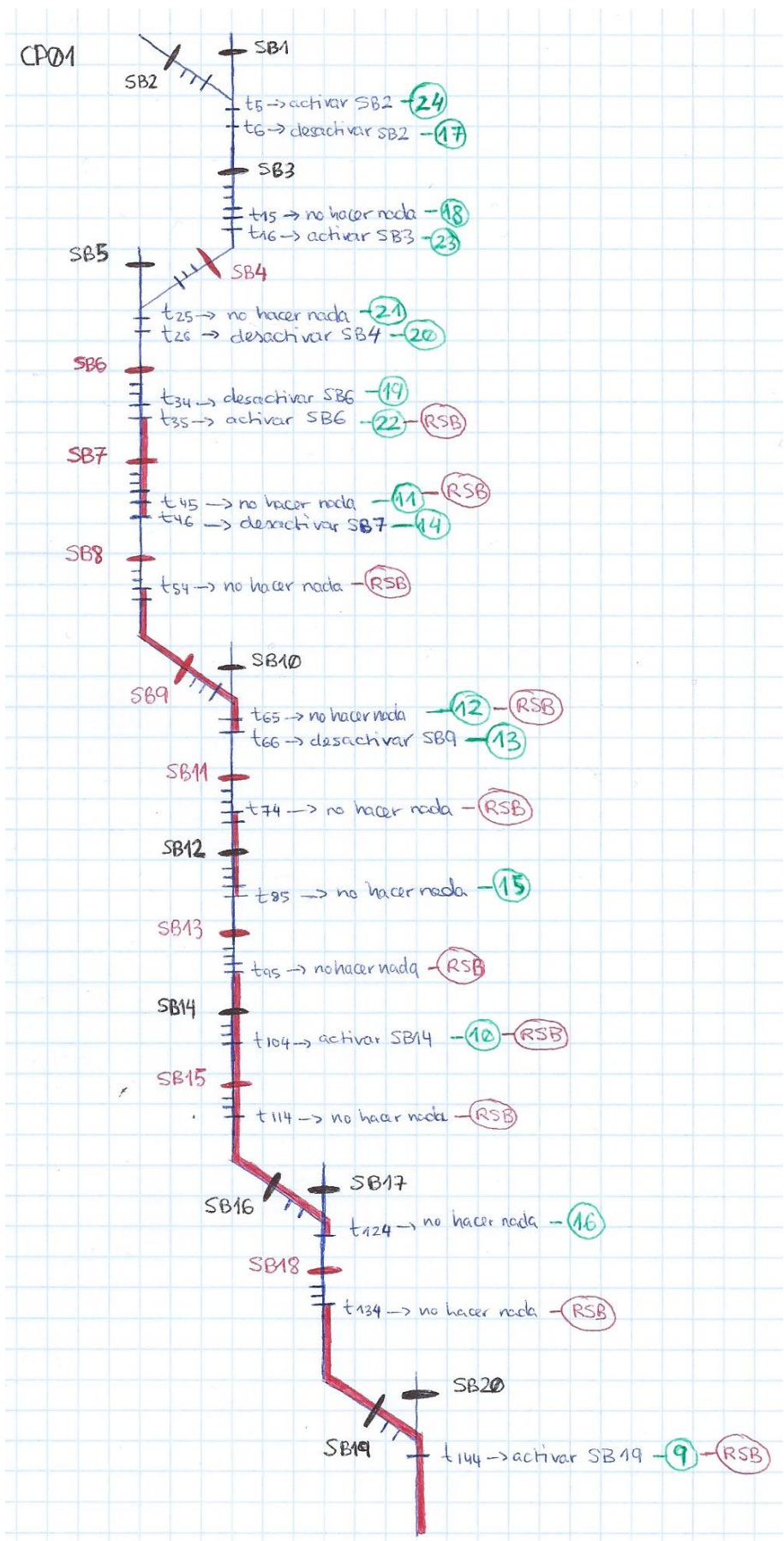


Figura 70. Detalle de CP01 que ejercita situaciones de la tabla de decisión

## 6. PRUEBAS EN HMI

El área de Posición corresponde a desarrollos sobre HMI, es decir, presentación de información e interacción del usuario con el sistema.

En este contexto se debe tener en cuenta que ejercitar las situaciones a probar implicará utilizar distintas operaciones y/o eventos del interfaz de usuario.

Pueden darse escenarios en donde es posible ejercitar funcionalidades mediante varias secuencias de acciones alternativas o incluso secuencias de acciones con cierta complejidad, en cuanto al número de pasos que involucran.

Debido a estas particularidades las pruebas de interfaces de usuario implican, en general, algunas consideraciones adicionales a la hora de diseñar pruebas.

Los apartados siguientes se corresponden con la descripción de tareas de preparación previas al diseño de pruebas, las cuales se centran en revisar, completar y preparar la test basis necesaria.

### 6.1. Revisar test basis

Es muy importante revisar la test basis para identificar si está completa o es necesario consultar documentación adicional. Si no se hace, se puede estar perdiendo información relevante para el diseño de pruebas.

Siempre supone una buena práctica cuestionarse qué aspectos se heredan y cuáles no cuando se trabaja en un proyecto que utiliza funcionalidades ya implementadas en otro proyecto anterior.

### 6.2. Identificar elementos de HMI

Se entenderá por elemento de HMI al conjunto de elementos de interfaz de usuario (menús, ventanas, pistas etc) que se utilizan para realizar alguna tarea concreta. Implica la inspección de la test basis una vez ha sido revisada (y completada si fuese necesario).

### 6.3. Identificar interacciones de usuario

Una vez enumerados los elementos que intervienen, se identificarán las interacciones de usuario como un conjunto de eventos que pueden ocurrir sobre los elementos (hacer clic con el ratón en un botón, introducir texto en un campo, desplegar un menú, etc.) y las acciones resultantes.

Esta tarea y la anterior son importantes porque sirven para identificar secuencias alternativas de elementos del interfaz de usuario que permiten realizar una misma funcionalidad (y que por tanto también se deberían ejercitar).

### 6.4. Identificar interfaces con otros componentes

El componente de la Posición de Torre (POS-T) actúa principalmente como una capa de presentación para la información que suministran otros componentes. Es necesaria una labor de identificación de los interfaces existentes con dichos componentes para conocer:

- el detalle de la información recibida y las condiciones para considerarla válida
- si es necesario realizar algún tratamiento o transformación previo de esas entradas
- si es necesario registrar esas entradas de alguna forma (logs e históricos) y bajo qué condiciones

### 6.5. Identificar aspectos de configuración

En esta tarea es necesario identificar todos aquellos aspectos de configuración que pueden condicionar la presentación de la información en los elementos de la Posición (y que por

tanto también será necesario probar). Para identificarlos puede ser especialmente útil revisar manuales de usuario.

De forma general se puede hablar de dos tipos de configuración:

- *Online*: aquellos aspectos que el usuario puede configurar en cualquier momento durante la ejecución del sistema
- *Offline*: aspectos que se configuran antes de la ejecución del sistema y que no pueden ser modificados durante la ejecución

## 6.6. Identificar productos de pruebas reutilizables

Dada una funcionalidad cualquiera dividida en una parte de lógica de negocio y otra de presentación, puede ocurrir que cada parte esté a cargo de personas diferentes. En este contexto la comunicación entre los responsables de desarrollar cada parte puede ser muy importante para evitar diseñar pruebas repetidas.

Puede convertirse en una buena práctica verificar si es posible reutilizar:

- casos de prueba ya existentes, extendiéndolos o adaptándolos
- parte de la configuración de casos existentes, como por ejemplo entradas de casos de prueba que contienen muchos datos
- notas de preparación del entorno para poder ejecutar pruebas

## 6.7. Ejemplo: funcionalidades genéricas de alertas CMON

En este apartado se aplica la secuencia de tareas indicadas anteriormente sobre un ejemplo real.

### 6.7.1. Revisar test basis

En la siguiente tabla se muestra en la parte izquierda la test basis para la funcionalidad de alertas CMON relativas a la presentación en POS. En la parte derecha se detallan los requisitos para las *alertas de incursión en pista* y las *alertas de zonas de rodaje restringidas*, que son las alertas ya implementadas (las actuales en el momento de comienzo del desarrollo de las alertas CMON y ATC):

EXTRACTO DE LA ESPECIFICACIÓN	
<i>Alertas CMON [SESAR 12.03.02]</i>	<i>Alertas de incursión en pista [SC#7047]</i>
...	...
24. FUNCIONALIDADES GENÉRICAS DE ALERTAS	2.4.2. ALERTA DE INCURSIÓN EN PISTA
24.1. RECONOCIMIENTO DE ALERTAS	...
<b>Se replicarán las funcionalidades de Reconocimiento de alertas usadas para las alertas actuales, sobre cada uno de los nuevos OGs definidos.</b>	2.4.2.3.2. <b>Inhibición de alertas</b>
Las reglas para el reconocimiento sobre OGs genéricos se definen en el apartado "OG GENÉRICO PARA LAS ALERTAS CMON".	REQ-94: En la VIG existirá un botón configurable indicando el estado de activación de las alertas de incursión en pista con las siglas RW. Los colores de activación serán los mismos que para el resto de alertas.
24.2. INHIBICIÓN DE ALERTAS EN POS (SÓLO INDIVIDUAL)	REQ-95: Desde el botón de la VIG, previa selección de un vuelo/vehículo (asumido) se podrá deshabilitar individualmente para las alertas de incursión en pista.
<b>La función de inhibición de alertas CMON y ATC se replicará de las alertas INCW y SAPW. Se podrá realizar únicamente sobre una track seleccionada.</b>	REQ-96: Tras la inhibición individual de un vuelo/vehículo, la POS emitirá un registro histórico de dicha acción manual.
24.3. ACTIVAR/DESACTIVAR UN GRUPO DE ALERTAS	REQ-97: Dicho registro histórico se grabará para su explotación posterior en Palestra (PAL-TDVT/SPV).
Se realizará a través de los ficheros de configuración, ya que no se realizarán desarrollos en PST, que es donde se realiza esta acción	...
	2.4.2.3.3. <b>Reconocer y silenciar alertas</b>
	REQ-101: En las posiciones en las que se presente el resalte de alerta no reconocida, se podrá reconocer individualmente el resalte pulsando con el BI del ratón

<p>actualmente.</p> <p>...</p>	<p>sobre el objeto ARWY.</p> <p>REQ-102: En las posiciones en las que se presente el resalte de predicción no reconocida, se podrá reconocer individualmente el resalte pulsando con el BI del ratón sobre el objeto ARWY.</p> <p>REQ-103: El reconocimiento de una alerta implica que cese el resalte de alerta no reconocida y se silencie el aviso acústico de dicha alerta.</p> <p>REQ-104: El reconocimiento de una predicción implica que cese el resalte de predicción no reconocida y se silencie el aviso acústico de dicha alerta.</p> <p>...</p>
--------------------------------	---

**Figura 71.** Cotejar alertas CMON y las de incursión en pista (funcionalidad)

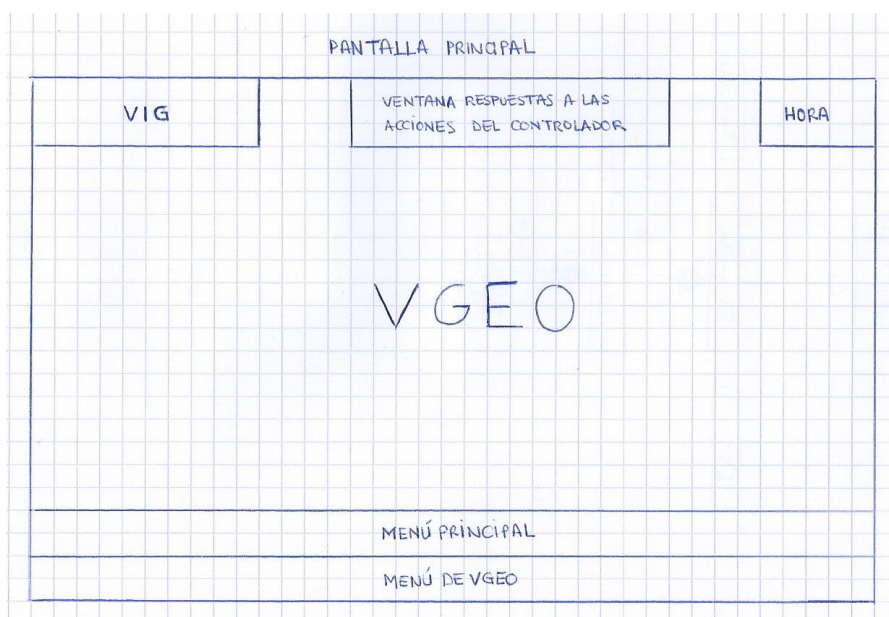
Además de las *funcionalidades* de reconocimiento e inhibición de alertas, en este caso también es interesante tener en cuenta los requisitos generales de presentación de las *alertas de incursión en pista* y las *alertas de zonas de rodaje restringidas*, para verificar si hay requisitos que hay que heredar. A continuación se muestra otro extracto:

EXTRACTO DE LA ESPECIFICACIÓN	
Alertas CMON [SESAR 12.03.02]	Alertas de incursión en pista [SC#7047]
<p>...</p> <p>24. FUNCIONALIDADES GENÉRICAS DE ALERTAS</p> <p>24.1. RECONOCIMIENTO DE ALERTAS</p> <p>Se replicarán las funcionalidades de Reconocimiento de alertas usadas para las alertas actuales, sobre cada uno de los nuevos OGs definidos.</p> <p>Las reglas para el reconocimiento sobre OGs genéricos se definen en el apartado "OG GENÉRICO PARA LAS ALERTAS CMON".</p> <p>24.2. INHIBICIÓN DE ALERTAS EN POS (SÓLO INDIVIDUAL)</p> <p>La función de inhibición de alertas CMON y ATC se replicará de las alertas INCW y SAPW. Se podrá realizar únicamente sobre una track seleccionada.</p> <p>24.3. ACTIVAR/DESACTIVAR UN GRUPO DE ALERTAS</p> <p>Se realizará a través de los ficheros de configuración, ya que no se realizarán desarrollos en PST, que es donde se realiza esta acción actualmente.</p>	<p>...</p> <p>2.4.2.3.1. <b>Presentación de alertas</b> de incursión en pista.</p> <p>REQ-74: Las alertas se presentarán en las etiquetas de los vuelos/vehículos afectados, en todas las ventanas geográficas, y con una indicación clara entre los elementos afectados.</p> <p>REQ-75: Si para un mismo móvil se producen situaciones de alerta sobre una pista abierta y una pista cerrada, sólo se presentarán las alertas de la pista abierta.</p> <p>...</p> <p>REQ-80: Cuando se presente una alerta de incursión en pista que afecte a dos vuelos y/o móviles se presentará un vector de conflicto que una los símbolos de los dos vuelos y/o móviles afectados.</p> <p>...</p> <p>REQ-85: Las etiquetas de vuelos en situación de alerta de incursión en pista dejarán de estar afectadas por filtros mientras dure la situación de alerta.</p> <p>REQ-86: Se podrán configurar resaltes acústicos independientes, en presencia y tono, para los dos niveles de alerta (predicción y violación).</p> <p>...</p>

**Figura 72.** Cotejar alertas CMON y las de incursión en pista (presentación)

## 6.7.2. Identificar elementos de HMI

Del manual de usuario de la Posición de Torre [MUPICT] se puede extraer la siguiente representación general de la pantalla principal que utiliza el usuario:



**Figura 73.** Esquema general de zonas de interacción en HMI

Sobre la pantalla principal anterior, en el área correspondiente a la *Ventana de Información Geográfica (VGEO)* se representan elementos gráficos relacionados con alertas de superficie tales como pistas de móviles, etiquetas y tabulares. El controlador interactuará con esos elementos y otros correspondientes a opciones en menús, o campos que requieren entrada de texto.

A continuación se presenta un extracto de la test basis correspondiente a la alerta ATC OPPOSITE STOP POINTS, resaltando sobre el texto los elementos de HMI relativos a alertas. EL formato en que describen el resto de alertas ATC es similar a este:

EXTRACTO DE LA ESPECIFICACIÓN					
...					
La POS-T recibirá cada actualización de alerta e identificará que se trata de una alerta "ATC-Opposite Stop Points" a través del Tipo de mensaje = 061. Presentará la información relativa al conflicto en:					
<ul style="list-style-type: none"> <li>• <b>Etiqueta del vuelo de cada track:</b> Se mostrará el indicador de alerta "ATC-Opposite Stop Points", siempre en el mismo color asociado al nivel Alarm (rojo), ya que este tipo de alerta siempre será de este mismo nivel.</li> <li>• <b>Tabular de alertas – sección "Conflicting ATC Clearances" (ATC):</b> Se presentarán dos líneas con datos de la alerta:</li> </ul>					
ATC					
Contenido de la línea	Primera línea	Indicador de alerta "ATC-Opposite Stop Points"	Autorización Conflictiva móvil 1	Identificador de plan de vuelo 1	Taxiway de conflicto 1
	Segunda línea		Autorización Conflictiva móvil 2	Identificador de plan de vuelo 2	Taxiway de conflicto 2
Ejemplo presentación	Primera línea	OPSP	LUP	IBE0001	M1
	Segunda línea		CRS	IBE0002	S1
...					
12. <b>OG GENÉRICO PARA LAS ALERTAS CMON (GCMON)</b> Con el objetivo de minimizar el espacio ocupado por las alertas en los elementos de presentación de la posición (EFS y etiquetas), se creará un OG genérico para las alertas CMON.					
...					

**Figura 74.** Especificación de alerta OPPOSITE STOP POINTS

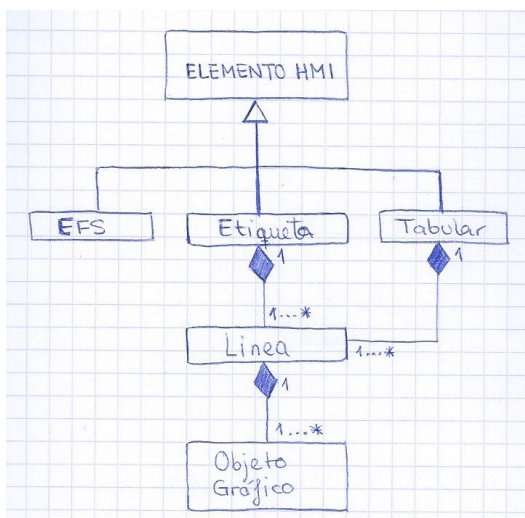
Se pueden identificar directamente como elementos de presentación en todos los tipos de alertas: la *etiqueta* de cada pista y el *tabular* de alertas.

De la revisión del resto de la test basis, se identifica otro elemento adicional: la ficha electrónica de vuelo (EFS por sus siglas en inglés):

Algunos elementos de presentación pueden estar compuestos a su vez por otros, que se corresponden físicamente con zonas del elemento en las que puede interactuar el usuario.

Por ejemplo, en la etiqueta de vuelo se pueden identificar distintos objetos gráficos (OGs) para representar alertas (en este caso uno u otro dependiendo de la configuración): OG genérico y OG detallado.

Todo esto se puede resumir en la siguiente figura:



**Figura 75.** Estructura de algunos elementos de HMI

En general tanto etiqueta como tabular serán elementos que se estructuran en un conjunto de 1 o varias líneas, en cada una de las cuales podrá haber varios objetos gráficos. [Nota: la estructura de la ficha electrónica de vuelo no estaba suficientemente definida en el momento de elaborar este documento y por esto no se refleja en la Figura 75].

### 6.7.3. Identificar interacciones de usuario

Las principales interacciones de usuario serían las siguientes:

Funcionalidad	Interacción asociada	Consecuencias de la interacción
<p><b>Reconocimiento:</b> permite al controlador dejar constancia de que ha advertido la existencia de la alerta, que está activa.</p> <p>La alerta activa tiene asociado el resalte de alerta no reconocida y opcionalmente podría tener asociada un aviso acústico.</p>	<p>Para reconocer la alerta el controlador tiene que pulsar con el BI (Botón Izquierdo) del ratón sobre el objeto gráfico que muestra el resalte.</p>	<p>Reconocer una alerta implica silenciar el aviso acústico en caso de que esté activo y que cese el resalte de alerta no reconocida en el objeto gráfico correspondiente</p>
<p><b>Silenciado:</b> hacer que cese el aviso acústico de una alerta no reconocida.</p>	<p>Se podrán silenciar pulsando sobre el botón de silenciar alarmas del menú principal o del táctil [MUPICT].</p>	<p>Sobre una alerta no reconocida que tiene un aviso acústico, implica que cese dicho aviso, pero no que cese el resalte de alerta no reconocida.</p>

<b>Inhibición:</b> en relación a una pista, implica excluirla para los posteriores cálculos de alertas.	Desde el botón de la VIG [MUPICT], previa selección de un vuelo/vehículo se puede deshabilitar individualmente.	Como consecuencia de una inhibición individual de un vuelo/vehículo, la POS emitirá un registro histórico de dicha acción manual.
---	---	---

**Tabla 9.** Descripción interacciones de usuario relativas a alertas sobre una pista

También es importante identificar con precisión el efecto de las interacciones anteriores sobre otros elementos de interfaz de usuario. Un tipo de efecto bastante habitual es la *sincronización* entre elementos, que se refiere a que “algo” que se hace sobre un elemento se debe “reflejar” también en otros. Un ejemplo puede ser el siguiente:

**EXTRACTO DE LA ESPECIFICACIÓN**

...

12.1. REGLAS PARA EL RECONOCIMIENTO DE ALERTAS

...

*Configuración simultánea de OGs genéricos y detallados en la misma POS:*  
*Podrían darse casos en los que el **OG genérico esté configurado para mostrarse en un lugar de la posición** (por ejemplo en la EFS), **y simultáneamente, los OGs detallados estén configurados para mostrarse en otro lugar de la misma posición** (por ejemplo en la etiqueta).*

*De cara a estas situaciones, **el OG genérico de alertas CMON estará ligado al OG detallado de la alerta que esté mostrando por prioridad, de forma que al reconocerse la alerta en uno de ellos, se reconocerá automáticamente en el otro, y viceversa.***

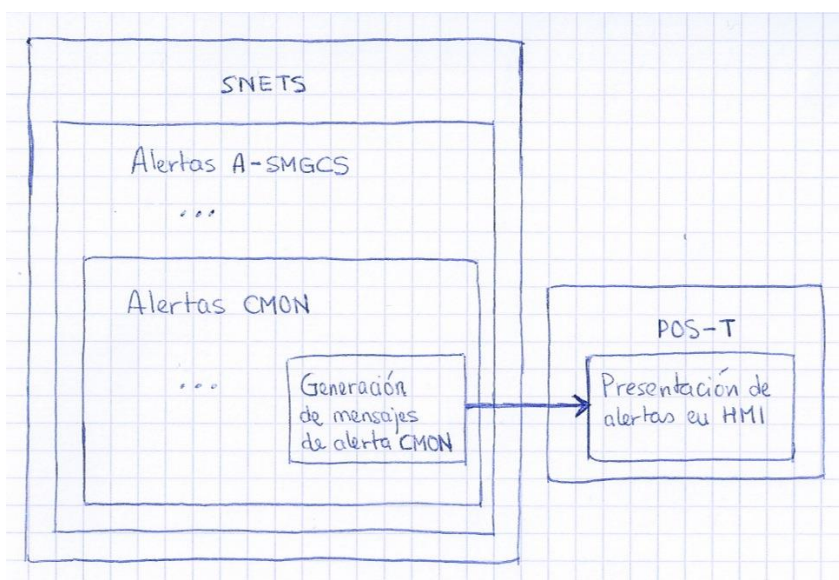
...

**Figura 76.** Especificación reconocimiento de alertas

En esta configuración reconocer una alerta activa en el OG genérico implica que además de cesar el resalte de la alerta en dicho OG genérico (texto parpadeante en color rojo y aviso acústico), también debe cesar el resalte en el OG detallado asociado.

### 6.7.4. Identificar interfaces con otros componentes

En el caso del componente de presentación de alertas de POS-T, existen interfaces con los componentes relativos a la generación de mensajes Asterix correspondientes a alertas:



**Figura 77.** Identificar interfaces con otros componentes

A continuación se muestra un extracto de la información que recibe el componente POS-T desde el componente *Generación de mensajes de alertas CMON*, para observar el conjunto de datos recibido para una alerta, en este caso de tipo ATC SAME/NEARBY STOP POINTS:



EXTRACTO DE LA ESPECIFICACIÓN	
...	
Por cada conflicto detectado, se enviará un evento de alerta que contendrá la información:	
I004/000 Message Type	062 – ATC-Same/Nearby Stop points
I004/010 Data Source Identifier	SAC / SIC (Replicar de las alertas existentes)
I004/020 Time of message	Tiempo de creación del conflicto (Replicar de las alertas existentes)
I004/030 Track number 1	Id Pista del Asterix 62 (Replicar de las alertas existentes)
I004/035 Track number 2	Id Pista del Asterix 62 (Replicar de las alertas existentes)
I004/040 Alert Identifier	Identificador del conflicto (calculado internamente por Safety Net) (Replicar de las alertas existentes)
I004/045 Alert Status	Nivel de alerta calculado: 010 = Alarm
I004/170 Aircraft Identification & Characteristics 1	Información del Asterix 62 del móvil 1 (se completa con la información disponible)
I004/171 Aircraft Identification & Characteristics 2	Información del Asterix 62 del móvil 2 (se completa con la información disponible)
I004/100 Area definition	Taxiways en las que se detecta el conflicto (los Tramos en los que está situado el Holding point de cada móvil, almacenados en TramoConflictoNEHP) -Subfield #3: Taxiway Designator 1 (Tramo en el que está el Holding point 1) -Subfield #4: Taxiway Designator 2 (Tramo en el que está el Holding point 2)
I004/SP (Special Purpose)	Se completa con la información necesaria para la alerta (la marcada): <ul style="list-style-type: none"> <li>• Byte 1- Fase de vuelo (PHA) <ul style="list-style-type: none"> <li>- Bits 1..4 – Fase de vuelo Track 1</li> <li>- Bits 5..8 – Fase de vuelo Track 2</li> </ul> </li> <li>• Byte 3 – Autorización Conflictiva (CLR) <ul style="list-style-type: none"> <li>- Bits 1..4 – Autorización Conflictiva Track 1</li> <li>- Bits 5..8 – Autorización Conflictiva Track 2</li> </ul> </li> </ul>
...	

**Figura 78.** Publicación del evento de alerta SAME/NEARBY STOP POINTS

La tabla anterior enumera y describe cada campo que debería contener el mensaje recibido en el componente de presentación de alertas ATC de POS-T y por tanto se trata de información muy importante a tener en cuenta de cara a identificar situaciones de prueba relacionadas por ejemplo con la recepción, tratamiento y registro de dicha información.

### 6.7.5. Identificar aspectos de configuración

La siguiente lista incluye algunos de los aspectos más relevantes que pueden afectar de alguna manera a la funcionalidad de presentación de alertas de superficie, por ejemplo modificando la posición que ocupan los OG asociados u ocultándolos [MUPICT]:

- **Configuración online** (interacciones del controlador)
  - Filtros o funcionalidades que permiten aumentar/disminuir el nivel de detalle con que se muestra información en los elementos de posición.
    - filtro ON-GROUND de pistas de superficie:

EXTRACTO DE LA ESPECIFICACIÓN	
...	
Filtra las pistas de superficie que están "en tierra" (ON-GROUND), para tener una presentación más limpia, sólo con información de pistas de secundario y del multilateración de aire. Opción disponible para torres con sistema de vigilancia de superficie.	
...	

**Figura 79.** Especificación filtro ON-GROUND de pistas de superficie

- filtros de las líneas 1,2,3,4 y CFL de las etiquetas de pista:

#### EXTRACTO DE LA ESPECIFICACIÓN

...

*LIN 1: Habilitar y deshabilitar la presentación de la línea 1 de las etiquetas de pista, excepto el INDICATIVO. Al activarse, permite ver la primera línea entera de la etiqueta de pista en vuelos correlados.*

*LIN 2: Habilitar y deshabilitar la presentación de la línea 2 de las etiquetas de pista. Al activarse, permite ver la segunda línea de la etiqueta de pista en vuelos correlados.*

*LIN 3: Habilitar y deshabilitar la presentación de la línea 3 de las etiquetas de pista, excepto el CFL. Al activarse, permite ver la tercera línea de la etiqueta de pista en vuelos correlados.*

*LIN4: Habilitar y deshabilitar la presentación de línea 4 de las etiquetas de pista. Al activarse, permite ver la cuarta línea de la etiqueta de pista. Esta línea presenta la distancia y el tiempo que tardaría la aeronave en llegar a la cabecera de pista seleccionada en la VIG en caso de dirigirse a ese punto con la velocidad actual. Esta línea solamente aparece cuando la aeronave está a menos de quince millas de la cabecera seleccionada.*

*CFL: Habilitar y deshabilitar la presentación del CFL de las etiquetas de pista. Al activarse, permite ver el CFL de la etiqueta de pista en vuelos correlados.*

...

**Figura 80.** Especificación filtros de las líneas 1,2,3,4 y CFL de las etiquetas de pista

- filtros de CORR y no CORR:

#### EXTRACTO DE LA ESPECIFICACIÓN

...

*Permite filtrar de la presentación de VGEO de las etiquetas de las pistas correladas o no correladas, para tener una presentación más limpia, sólo con información de las pistas deseadas.*

...

**Figura 81.** Especificación filtros de CORR y no CORR

- funcionalidad suprimir etiqueta de pista:

#### EXTRACTO DE LA ESPECIFICACIÓN

...

*Mediante este pulsador "SEP", se inhibe la presentación de la etiqueta y el vector de asociación de una pista. Para que reaparezca, debe pulsarse con el botón derecho del ratón sobre la pista en cuestión.*

...

**Figura 82.** Especificación funcionalidad suprimir etiqueta de pista

- desactivación total de filtros:

#### EXTRACTO DE LA ESPECIFICACIÓN

...

*Permite visualizar toda la información no aplicando a los filtros establecidos.*

...

**Figura 83.** Especificación desactivación total de filtros

- selección de límites de los filtros de altitud:

#### EXTRACTO DE LA ESPECIFICACIÓN

...

*Establece los límites superior e inferior de los filtros de altitud.*

...

**Figura 84.** Especificación selección de límites de los filtros de altitud

- Formatos de presentación de etiquetas (p.e. básico ↔ extendido).
- Selección de operativa (MODO NORMAL ↔ MODO AUTÓNOMO):

#### EXTRACTO DE LA ESPECIFICACIÓN

...

*a) En el modo NORMAL, en la posición se presentan Pistas de los Radares de Aproximación y de superficie según se indicará más adelante.*

...

Respuestas de la función:

- El cambio a Modo de Funcionamiento Autónomo hace desaparecer toda la información mantenida por el TDR en Modo Normal (pistas, alertas AC, alarmas de correlación, estado de radares y áreas especiales) así como las ventanas de apertura manual utilizadas para mostrar alguna de estas informaciones. Se pierden los ficticios creados en modo normal.

Seguidamente aparece la información mantenida por el TDR en Modo Autónomo de la posición (pistas, alarmas de correlación y estado de radares).

...

b) En el modo AUTONOMO, sólo se presentan Pistas y Plots sintéticos de los Radares de Aproximación. El sistema funciona en Modo Autónomo, cuando no tienen conexión con los servidores de información radar.

Es esta situación, la Posición estará funcionando con un tratamiento de datos radar local, que realiza los mismos procesos que el sistema central pero en cada Posición de forma independiente.

...

Respuestas de la función

- El cambio a Modo de Funcionamiento Normal hace desaparecer toda la información mantenida por el TDR en Modo Autónomo (pistas, alarmas de correlación, estado de radares) así como las ventanas de apertura manual utilizadas para mostrar alguna de estas informaciones. Se pierden los ficticios creados en modo autónomo.

Seguidamente aparece la información mantenida por el TDR en Modo Normal (pistas, alertas, alarmas de correlación, estado de radares y áreas especiales).

...

**Figura 85.** Especificación selección de operativa

- **Configuración offline** (configuración en SEI)
  - Configuración de la ubicación de OGs en elementos de la posición (p.e. del OGs genérico y detallado).
  - Límites
    - Tiempos máximos/mínimos de validez de la información presentada (p.e. tiempo máximo transcurrido desde última actualización sin que se haya recibido el mensaje CMON o ATC que contiene la información de todas las alertas detectadas).
    - Volumen de información (p.e. número máximo de alertas simultáneas que es posible detectar).
  - Configuración global del nivel de detalle de la funcionalidad (p.e. niveles de alerta que se presentan en la posición: información y alarma o solo alarma).

### 6.7.6. Identificar productos de pruebas reutilizables

En el caso del proyecto SESAR 12.03.02, entre las situaciones a probar se debería haber contemplado las relativas a la *generación* correcta de mensajes Asterix de cada tipo de alerta.

Seguramente muchos de esos casos de prueba (o al menos parte de sus datos) podrían ser reutilizados para las pruebas relativas al *tratamiento* de los mensajes Asterix recibidos en POS-T (tareas típicas, como validación o registro de datos), ya que de esta manera se evita tener que preparar nuevos datos de prueba.

### 6.7.7. Diseño de pruebas

Una vez completadas las tareas de preparación previas se puede comenzar a diseñar casos de prueba. Este apartado se centrará en el diseño de pruebas relativas a las *interacciones de usuario sobre el OG genérico para las alertas CMON*.

En este contexto, para una misma pista pueden ocurrir simultáneamente varios conflictos, pero en el OG genérico asociado a cada pista solo se muestra el texto asociado a la alerta más prioritaria. Cuando la alerta mostrada desaparece, se muestra automáticamente la alerta siguiente en orden de prioridad.

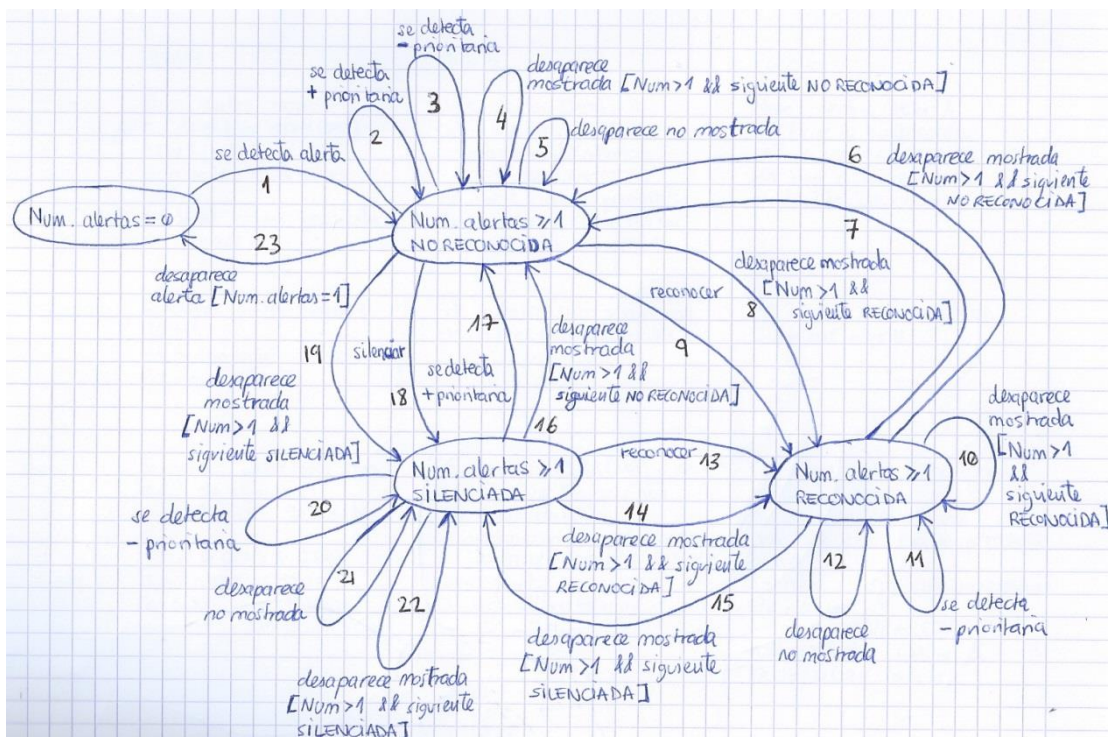
	<b>Tipo de alerta CMON</b>	<b>Prioridad</b>
...		
12. OG GENÉRICO PARA LAS ALERTAS CMON	<b>No Landing Clearance</b>	↑
Con el objetivo de minimizar el espacio ocupado por las alertas en los elementos de presentación de la posición (EFS y etiquetas), se creará un OG genérico para las alertas CMON.	<b>Landing on Wrong Runway</b>	
	<b>No Contact</b>	
	<b>No Take-off Clearance</b>	
	<b>No Line-up Clearance</b>	
	<b>Lining-up on Wrong Runway</b>	
Mostrará el OG definido para una única Alerta CMON, aunque se detecten varias alertas de este tipo simultáneamente sobre la misma track, siguiendo la siguiente tabla de prioridades	<b>No Cross Clearance</b>	
	<b>No Enter Clearance</b>	
	<b>Entering Wrong Runway</b>	
	<b>Red Stop bar</b>	
	<b>No Taxi Clearance</b>	
Algunas alertas nunca se podrán dar simultáneamente por ejemplo por el tipo de vuelo, Despegue o Arribada, o por ser Vehículo o Aeronave, pero se incluyen priorizadas para facilitar su implementación en la posición.	<b>No Push-back Clearance</b>	
	<b>Taxi Route Deviation</b>	
	<b>High Speed Movement</b>	
	<b>Closed Runway</b>	
	<b>Closed Taxiway</b>	
...	<b>Stationary After Clearance</b>	

**Figura 86.** Prioridad de alertas CMON mostradas en OG genérico

Dadas las características de la test basis parece que se reúnen las condiciones necesarias para inferir un modelo de transición de estados que represente el funcionamiento del OG genérico de una pista de superficie en función de la detección de alertas.

Se podría definir un modelo básico inicial, teniendo en cuenta lo siguiente:

- Estados: 4 estados que representan la combinación del número de alertas CMON detectadas y el estado de reconocimiento de la alerta mostrada (la más prioritaria) en el OG genérico:
  - No hay alertas → Número alertas = 0
  - Hay una o más alertas
    - Número alertas  $\geq 1$  y la alerta mostrada no está reconocida ni silenciada.
    - Número alertas  $\geq 1$  y la alerta mostrada está silenciada y no reconocida.
    - Número alertas  $\geq 1$  y la alerta mostrada está reconocida.
- Eventos:
  - Detección de una nueva alerta
    - de mayor prioridad
    - de menor prioridad
  - Desaparición de una alerta
    - la mostrada (la más prioritaria)
    - alguna no mostrada
  - El usuario reconoce la alerta mostrada.
  - El usuario silencia la alerta mostrada (ésta tenía asociado aviso acústico).



**Figura 87.** Modelo inferido a partir de la especificación de presentación de alertas

Las acciones derivadas de la detección/desaparición de alertas serían las relativas a:

- la selección de la alerta que se muestra en el OG genérico.
- la ordenación de las alertas detectadas de acuerdo a las prioridades establecidas.
- cese del resalte gráfico (deja de parpadear el texto indicativo de la alerta) y/o acústico (deja de reproducirse el aviso acústico).

Este modelo de transición de estados se puede utilizar para derivar situaciones de prueba aplicando la técnica de cobertura de caminos.

A continuación se presenta un ejemplo de caso de prueba suponiendo:

- una configuración de presentación en la que solo se muestra el OG genérico (no se desarrolla aquí la derivación de situaciones de prueba mediante la técnica).
- se parte de un estado inicial donde no hay alertas y se termina también en un estado sin alertas.

CP01	Paso 1	Paso 2	Paso 3
Entrada	Num. alertas CMON=0 M1 parada en el stand (es un despegue)	Num. alertas CMON=1 Se está mostrando NPBK en OG genérico M1 tiene tramos pendientes de recorrer	Num. alertas CMON=1 Se está mostrando NTAX en OG genérico M1 tiene tramos pendientes de recorrer
Descripción	M1 necesita push-back. M1 autorización = ninguna. M1 empieza a moverse siguiendo ruta PV.	Reconocer alerta. M1 autorización = ninguna. Detener M1. Continuar ruta según PV	M1 autorización ninguna. Se cierra último tramo del recorrido de M1
Salida	Se genera alerta NPBK(M1) Num. alertas CMON=1 Se muestra NPBK en OG genérico y se actualiza tabular	Al reconocer, cesa el resalte Al continuar movimiento tras parar, desaparece NPBK Se genera NTAX (M1) Num. alertas CMON=1 Se actualiza tabular de alertas	Al cerrar el tramo se genera CTWY (M1). Se sigue detectando NTAX(M1) y se muestra en OG genérico Num. alertas CMON=2 Se actualiza tabular
	Paso 4	Paso 5	Paso 6
Entrada	Num. alertas CMON=2 Se detectan NTAX(M1) y CTWY (M1) Se muestra NTAX en OG genérico. M1 tiene tramos pendientes de recorrer	Num. alertas CMON=3 Se tiene la siguiente ordenación de alertas: NTAX ← CRWY ← CTWY M1 tiene tramos pendientes de recorrer y está en un segmento con SB activa	Num. alertas CMON=4 Se tiene la ordenación: RSBA ← NTAX ← CRWY ← CTWY M1 tiene algunos tramos pendientes de recorrer
Descripción	Se cierra pista despegue de M1. M1 autorización = line-up M1 sigue ruta PV	M1 autorización = ninguna M1 atraviesa SB activa	M1 autorización = taxi Silenciar alerta
Salida	Se genera CRWY (M1) tras aut. de line-up. La ordenación de alertas es: NTAX ← CRWY ← CTWY Num. alertas CMON=3 Se actualiza el tabular	Se genera RSBA (M1) y se muestra en OG genérico con aviso acústico. La ordenación de alertas es: RSBA ← NTAX ← CRWY ← CTWY Num. alertas CMON=4 Se actualiza tabular	Desaparece alerta NTAX cuando se da aut. taxi. Al silenciar cesa aviso acústico pero resalte permanece activo. Se tiene la siguiente ordenación de alertas: RSBA ← CRWY ← CTWY Num. alertas CMON=3 Se actualiza tabular
	Paso 7	Paso 8	Paso 9
Entrada	Num. alertas CMON=3 Se tiene la ordenación de alertas: RSBA ← CRWY ← CTWY M1 tiene tramos pendientes por recorrer	Num. alertas CMON=2 Se tiene la ordenación: CRWY ← CTWY M1 tiene tramos pendientes por recorrer	Num. alertas CMON=1 Alerta mostrada es CRWY
Descripción	Desactivar SB superada por M1 M1 autorización = ninguna	Se abre tramo cerrado en ruta de M1	Abrir pista despegue para M1. M1 autorización = take-off
Salida	Al desactivar SB desaparece alerta RSBA Se tiene la siguiente orden: CRWY ← CTWY La alerta mostrada tiene el resalte activo y no hay aviso acústico Num. alertas CMON=2 Se actualiza el tabular	Tras abrir tramo desaparece alerta CTWY Se muestra alerta CRWY en el OG Num. alertas CMON=1 Se actualiza el tabular	Al dar aut. take-off desaparece alerta CRWY. OG genérico está vacío Num. alertas CMON=0 Se actualiza el tabular

Figura 88. Detalle CP01 presentación de alertas

Este caso de prueba representa el siguiente camino sobre el modelo de estados anterior:

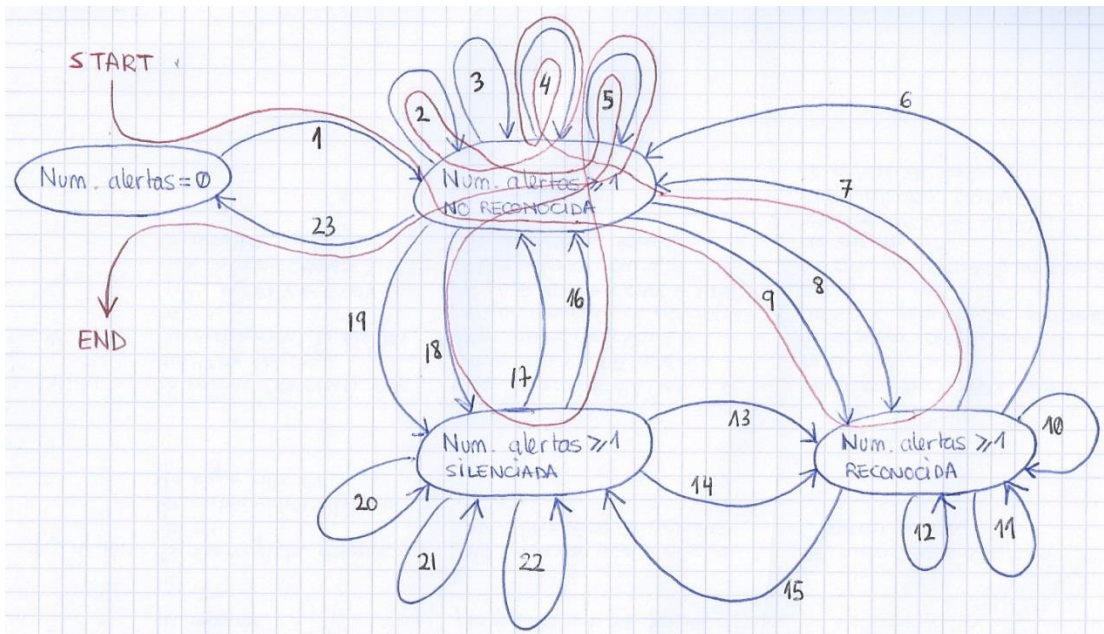


Figura 89. Flujo CP01 presentación de alertas

### 6.7.7.1. Extender el modelo básico: configuración

Aparte de cubrir aspectos funcionales también es importante tener en cuenta aquellos aspectos de configuración que pueden afectar a la presentación de alertas que hayan sido identificados en tareas de preparación previas.

Las situaciones de prueba derivadas a partir del modelo extendido se podrían combinar con alguno o varios de los siguientes aspectos:

- Configuración online:
  - Inhibir las alertas sobre la pista.
  - Formato configurado para el OG del elemento desde el que se reconoce la alerta:
    - OG genérico
    - OG detallado
  - Filtros sobre pistas:
    - ON-GROUND
    - Suprimir Etiqueta de Pista
    - ...
  - Alternando el modo de operativa seleccionado:
    - NORMAL → AUTÓNOMO
    - AUTÓNOMO → NORMAL
    - ...
- Configuración offline:
  - Desactivar grupos de alertas en ficheros de configuración.
  - ...

Mediante la combinación de todos estos aspectos lo que se consigue es probar situaciones que se aproximen lo más posible a la operación de un usuario real, que puede llegar a ser bastante completa y compleja.

Un ejemplo de combinación podría consistir en extender el caso de prueba CP01 anterior, añadiendo un paso de prueba intermedio entre los pasos 5 y 6, que consista en inhibir las alertas CMON (configuración online) para la pista correspondiente al móvil M1 y transcurrido un breve instante deshacer esa acción.

La salida esperada en este caso sería:

- tras inhibir las alertas se dejaría de detectar cualquier alerta para dicha pista.
- el OG genérico debería aparecer vacío.
- tampoco se debería reproducir el aviso acústico.
- el tabular de alertas CMON debería actualizarse eliminando las alertas correspondientes a la pista del móvil M1.
- al deshacer la inhibición de alertas, se debería de volver a detectar la ordenación de alertas anterior a la inhibición: **RSBA** ← NTAX ← CRWY ← CTWY.
- se vuelve a reproducir el aviso acústico asociado a la alerta RSBA, que es la mostrada en el OG genérico asociado a la pista del móvil M1.
- se actualiza el tabular de alertas CMON, volviendo a aparecer las alertas detectadas.



## 7. PRIORIZACIÓN DE PRUEBAS

Esta sección detalla algunas recomendaciones relativas a aspectos de priorización aplicables en pruebas en desarrollo. A continuación se presenta una propuesta estructurada en dos niveles:

- priorización en la fase de planificación, en donde se define la estrategia global de pruebas y se planifican las tareas de desarrollo.
- priorización en la fase de diseño y ejecución de pruebas.

### 7.1. NIVEL 1: estrategia de pruebas

Es beneficioso e importante planificar las tareas de desarrollo teniendo en cuenta criticidades en función de su riesgo asociado (generalmente a mayor riesgo, mayor criticidad) y determinar la *intensidad* de pruebas necesaria para verificar que el trabajo realizado en cada tarea cubre la mínima calidad exigible (de acuerdo a la criticidad).

La intensidad de pruebas se refiere a la exhaustividad con que se decide probar un determinado objetivo de prueba, atendiendo al nivel de riesgo identificado para el mismo. Combinar diferentes técnicas permite pruebas más o menos exhaustivas, aunque una sola técnica también puede conducir a pruebas más exhaustivas (o menos si se decide no cubrir todas las situaciones).

Normalmente a los objetivos de prueba que se consideran críticos se les aplica una mayor intensidad, mientras que a los que no implican riesgo importante (y por tanto no son críticos) se les aplica menor intensidad de pruebas (o incluso se puede decidir que no se van a probar).

Los aspectos anteriores se podrían resumir en dos principios: “probar primero lo más crítico” y “probar con mayor intensidad lo más crítico”.

La clasificación de tareas de desarrollo en niveles de criticidad puede no ser trivial y puede requerir un análisis de riesgos previo, por tanto la responsabilidad de esta actividad debería recaer en personas (o grupo de personas) que reúnan en un alto conocimiento funcional y técnico de cada desarrollo particular, como por ejemplo responsables de área de desarrollo y/o de proyecto.

#### 7.1.1. Identificar objetivos de prueba

En el contexto particular de ISL Gijón las entregas al cliente implican normalmente un determinado conjunto de tareas de desarrollo que cubren las distintas funcionalidades a entregar. Cada tarea de desarrollo podría dar lugar a uno o varios *objetivos de prueba*.

En el caso del proyecto SESAR 12.03.02 incluido en el alcance de este documento, algunos ejemplos de objetivos de prueba podrían ser los siguientes:

Objetivos de prueba
Integración con subsistema de Plan de Vuelo
Integración con subsistema de Vigilancia
Cálculo de conflictos CMON
Cálculo de conflictos ATC
Generación de mensajes ASTERIX

**Tabla 10.** Ejemplo de listado de objetivos de prueba

#### 7.1.2. Definición de riesgo

Cada objetivo de prueba debería ser asignado a una clase de riesgo y por este motivo es importante utilizar una misma definición de lo que se debe entender por *riesgo*.

*Definición:* se entiende por riesgo la probabilidad de que ocurra un fallo en relación al daño esperado si este se produce.

El riesgo se puede representar mediante la siguiente fórmula:

$$\text{Riesgo} = \text{Probabilidad de fallo} \times \text{Daño}$$

donde:

$$\text{Probabilidad de fallo} = \text{Frecuencia de uso} \times \text{Probabilidad de presencia de defectos}$$

Los factores que intervienen en un riesgo presentan ciertas propiedades que es necesario tener en consideración a la hora de proponer una clasificación:

Frecuencia de uso: en un sistema (o componente) que es accedido (para lectura o escritura) cientos de veces al día, la probabilidad de que se manifieste un defecto es bastante mayor que en un sistema (o componente) que se accede unas pocas veces al día.

Probabilidad de presencia de defectos: para evaluar la presencia de defectos puede ser de utilidad el siguiente resumen de situaciones [SCHAEFER1996] en donde se pueden concentrar defectos (se hace referencia a funciones, pero se podría extender al concepto de componente o subsistema):

- Funcionalidades complejas.
- Funciones completamente nuevas.
- Funciones sobre las que se realizan adaptaciones con cierta frecuencia.
- Funciones en las que se utilizan herramientas o técnicas específicas por primera vez.
- Funciones que son reasignadas de un desarrollador a otro.
- Funciones implementadas bajo gran presión de tiempo de entrega.
- Funciones que requieren optimizaciones de velocidad y/o uso de recursos.
- Funciones en las que previamente se haya detectado defectos.
- Funciones con muchos interfaces.

Otras circunstancias que pueden aumentar la probabilidad de defectos son:

- Desarrolladores con poca experiencia en desarrollo.
- Desarrolladores con experiencia en desarrollo que cambian a un área de conocimiento desconocida.
- Intensidad de pruebas insuficiente de las pruebas en desarrollo.
- Utilización de nuevas herramientas o entornos.
- Equipos de desarrollo compuestos por muchas personas
- Problemas que afectan a la comunicación de los miembros de los equipos de desarrollo (por causas geográficas o personales).

Daño: es conveniente hacer la distinción entre daño *directo* e *indirecto*. Un daño directo podría involucrar pérdidas de ingresos, daños a terceros, daños físicos o en el medio ambiente o sobrecostos por correcciones. Un daño indirecto podría involucrar daño en la imagen corporativa o pérdida de confianza de los clientes.

### 7.1.3. Clases de riesgo

El daño y riesgo asignado a un objetivo de prueba se traducirá en una determinada *clase de riesgo*.

Para hacer esto es útil definir una tabla de clasificación del riesgo, donde el primer paso consiste en decidir el número de niveles de riesgo que se van a utilizar.

Lo más común suele ser utilizar 3 niveles de riesgo (p.e. ALTO, MEDIO, BAJO), aunque debido al alcance de esta guía se decide por simplicidad utilizar una clasificación en 2 niveles de riesgo (CRÍTICO y NORMAL).

En general lo más recomendable es adaptar el número de niveles de riesgo a las necesidades de cada momento, de manera que si en el futuro cambiase el tipo de proyectos que se llevan a cabo, se podría reconsiderar la utilización de otra clasificación con un número diferente de niveles.

Una vez definidos los niveles de riesgo que se van a utilizar, el siguiente paso consiste en definir una clasificación del riesgo.

En este sentido, se podría optar por definir una clasificación genérica, utilizable en cualquier tipo de proyecto. Si hubiese proyectos con características singulares para los que la clasificación genérica no resulta adecuada, se podría definir otras clasificaciones particulares para ellos.

Un ejemplo de una posible clasificación de riesgo podría ser el siguiente:

Clasificación de riesgo		Probabilidad de fallo	
		Alto	Bajo
Daño	Alto	CRITICO	CRITICO
	Bajo	NORMAL	NORMAL

**Tabla 11.** Ejemplo de clasificación de riesgo con 2 niveles de criticidad

En esta clasificación se han utilizado en concreto:

- 2 niveles de daño: Alto y Bajo
- 2 niveles de probabilidad de fallo: Alto y Bajo

La complejidad de la clasificación de riesgo se puede ajustar a las necesidades: por ejemplo, se podría decidir que para el factor *Daño* el número de niveles debería ser 3 (Alto, Medio y Bajo), manteniendo 2 niveles (Alto y Bajo) para el factor *Probabilidad de fallo*.

Suponiendo que se estima un daño y una probabilidad de fallo para los objetivos de prueba identificados en la Tabla 10, la clase de riesgo obtenida en base a la clasificación de la Tabla 11 sería la siguiente:

Objetivo de prueba	Daño	Probabilidad fallo	Clase
Integración con subsistema de Plan de Vuelo	Alto	Bajo	CRITICO
Integración con subsistema de Vigilancia	Alto	Bajo	CRITICO
Cálculo de conflictos CMON	Alto	Alto	CRITICO
Cálculo de conflictos ATC	Alto	Alto	CRITICO
Generación de mensajes ASTERIX	Bajo	Bajo	NORMAL

**Tabla 12.** Ejemplo de clasificación de objetivos de prueba en niveles de riesgo

#### 7.1.4. Intensidad de pruebas

La intensidad de pruebas se determina para cada objetivo de prueba en función de la clase de riesgo. Para ello es necesario establecer previamente una clasificación de los distintos niveles de intensidad que se aplicarán.

Un ejemplo de una posible clasificación de niveles de intensidad podría ser la siguiente:

Nivel de intensidad	Descripción
●●	Intensidad máxima
●	Intensidad básica
Test estático	Revisión de la test basis, inspección de código, ...

**Tabla 13.** Ejemplo de clasificación de niveles de intensidad de prueba

El test estático no implicaría la ejecución de software, sino la revisión de documentación relevante o la inspección manual de código.

La intensidad básica estaría asociada a cubrir las situaciones de prueba que se consideran mínimas para cada objetivo de prueba.

La aplicación de intensidad máxima supondría la necesidad de cubrir situaciones de prueba más completas y complejas, por tanto normalmente implicará utilizar alguna combinación de técnicas de prueba.

Es importante tener en cuenta que el número de niveles de intensidad de pruebas no tiene que coincidir necesariamente con el número de niveles de riesgo que se haya definido previamente. Por ejemplo, para 2 niveles de riesgo se podría decidir utilizar 3 niveles de intensidad de pruebas (máxima, media, básica) si se considerase necesario.

Con la asignación de los niveles de intensidad se termina de definir la estrategia de pruebas. Un ejemplo de estrategia podría ser el siguiente:

Estrategia de pruebas proyecto 12.03.02				
Objetivo de prueba	Daño	Probabilidad fallo	Clase	Intensidad
Integración con subsistema de Plan de Vuelo	Alto	Bajo	CRITICO	●●
Integración con subsistema de Vigilancia	Alto	Bajo	CRITICO	●●
Cálculo de conflictos CMON	Alto	Alto	CRITICO	●●
Cálculo de conflictos ATC	Alto	Alto	CRITICO	●●
Generación de mensajes ASTERIX	Bajo	Bajo	NORMAL	●
...	...	...	...	...

**Tabla 14.** Ejemplo de estrategia de pruebas basada en riesgos

### 7.1.5. Planificación de las pruebas

A causa del alcance de esta guía, se hablará en términos de “tareas de desarrollo”, en las cuales estarían ya incluidas todas las tareas de pruebas realizadas por el desarrollador. Por tanto, de aquí en adelante cuando se haga referencia al “riesgo de una tarea” se estará haciendo referencia a tareas de desarrollo.

Sobre la base de la clasificación de riesgos, sería recomendable poner en primer lugar las tareas correspondientes a riesgos altos. Implementar en primer lugar dichas tareas permite probarlas primero y con ello detectar (y corregir) los defectos más importantes primero.

Este enfoque permitiría obtener la certeza de que las funcionalidades más importantes tienen la calidad requerida ya en los comienzos de cada desarrollo concreto.

Esto puede ser útil desde el punto de vista de posibles interrupciones del proceso de desarrollo: si sucede algo no previsto a mitad del desarrollo se va a poder entregar funcionalidades importantes que han sido verificadas, porque ya se probaron al principio y no al final.

Es importante tener en cuenta aspectos técnicos al realizar la priorización, dado que en algunos casos el propio diseño y/o la arquitectura del sistema podrían introducir dependencias que impidiesen poner en primer lugar los riesgos importantes.

### 7.1.5.1. Tiempo de reparación, regresión y re-test

Hacer una previsión del esfuerzo que puede ser necesario (al menos para las tareas con riesgos altos) en tareas de desarrollo como son re-test, corrección de defectos y regresiones (si aplicasen) podría ayudar a afinar la estimación de horas dedicadas a la realización de pruebas que va asociada a las tareas de desarrollo que se realizan en ISL Gijón.

### 7.1.5.2. Criterios de entrada y salida

De cara a la planificación puede ser importante tener en cuenta la siguiente información:

- Disponibilidad actual y futura de los recursos (personas e infraestructura)
- Disponibilidad de los documentos de requisitos, diseño y otra documentación necesaria para comenzar el desarrollo (no disponibles, poco estables,...)
- Tiempo total disponible, de cara a fijar deadlines de las tareas del proyecto.
- Dependencias técnicas entre tareas de desarrollo.
- Hitos

Del mismo modo, es necesario evaluar qué criterios se van a establecer para decidir cuándo se ha probado suficientemente como para dar por terminadas las pruebas. Existen varias alternativas válidas:

- A. Número de defectos en una determinada clase de riesgo que permanecen abiertos:** p.e. no pueden quedar defectos asociados a la clase de riesgo más alta y un máximo de 5 defectos para la clase de riesgo más baja (siguiendo con la clasificación en 2 niveles propuesta para la estrategia de pruebas), o en total no debe haber más de 10 defectos entre todas las clases de riesgo.
- B. Criterio adoptado para cubrir una determinada clase de riesgo:** p.e. todas las funcionalidades de la categoría de riesgo más alta han sido probadas usando técnicas de diseño de pruebas formales.
- C. Intensidad con que se debería probar los requisitos:** p.e. para cada funcionalidad que requiere interacción del usuario se han probado todos los flujos correctos.

En el contexto de las pruebas en desarrollo de ISL Gijón se podría establecer con carácter general el criterio C, es decir, terminar cuando se ejerciten las situaciones de prueba derivadas en función de la intensidad de prueba que se haya fijado en la estrategia.

### 7.1.5.3. Criterios de suspensión y continuación

Pensar en estos dos aspectos con antelación podría ser beneficioso. Ejemplos de criterios de suspensión comunes son:

- Determinados componentes de la infraestructura necesaria para ejecutar las pruebas no están disponibles
- Encontrarse defectos que bloquean la ejecución de otras pruebas

Criterios para la continuación típicos podrían ser los siguientes:

- Cuando finalicen las condiciones que dieron lugar a la suspensión, se volverán a ejecutar todas las pruebas que ya se habían ejecutado antes de la suspensión
- Realización de pre-test (pruebas mínimas para evaluar si el sistema es suficientemente estable como para realizar otras pruebas más completas)

## 7.2. NIVEL 2: diseño y ejecución de pruebas

De cara al diseño de pruebas el desarrollador tendrá principalmente 2 entradas que le servirán como referencia:

- Un conjunto de tareas de desarrollo, cada una de las cuales implicará la implementación de un determinado conjunto de funcionalidades.
- La criticidad e intensidad de pruebas asignada en la estrategia de pruebas a cada objetivo de prueba.

Cada tarea de desarrollo supone un determinado esfuerzo, parte del cual se dedicará a la implementación/corrección de funcionalidades y el resto se dedicará a trabajo de pruebas propiamente dicho (diseño y ejecución de pruebas).

El desarrollador podría encontrarse dos situaciones:

- A. Las tareas de desarrollo asignadas ya tienen restricciones y dependencias que determinan el orden en que se deben abordar. En este caso se debe seguir dicho orden.
- B. Se le asignan algunas tareas de desarrollo que forman parte de una determinada entrega, pero no están ordenadas de ninguna manera.

En este caso el desarrollador tendría la responsabilidad de priorizar sus tareas de desarrollo, atendiendo a la criticidad asignada en la estrategia de pruebas: se pondrán en primer lugar aquellas tareas de desarrollo que tienen asociados objetivos de prueba con mayor riesgo.

A igual criticidad, un criterio válido podría consistir en comenzar por aquellas tareas que requieren mayor intensidad de pruebas, dado que es un indicador de la importancia que tienen desde el punto de vista de la calidad mínima que espera el cliente a la entrega (si una tarea requiere mayor intensidad de pruebas que otras, es que es importante).

En caso de igual criticidad e intensidad de pruebas, se podrían seleccionar tareas atendiendo a menor esfuerzo estimado para completarlas (en horas).

### **7.2.1. Diseño de pruebas**

Con vistas al diseño de pruebas el desarrollador podría utilizar como referencia el siguiente conjunto de prácticas.

#### **7.2.1.1. Probar funcionalidad de un componente**

En general, considerar el uso de las técnicas de CE y MCDC, así como de tablas de decisión cuando sea abordable. Normalmente el uso de tablas de decisión será abordable cuando el número de condiciones sea 5 como máximo, ya que hay que considerar que el número total de situaciones que se generan (y que por tanto se debería ejercitar) crece exponencialmente:  $2^N$ , siendo N el número de condiciones.

Tener en cuenta que el grado de exhaustividad de las pruebas aumenta cuando se combina:

- Más o menos atributos en el contexto de una misma técnica, para derivar situaciones de prueba que consistan en la combinación de varios atributos (frente a derivar situaciones de prueba más simples, donde no se combina nada).
- Varias técnicas distintas, como por ejemplo CE+ VL, CE+ MCDC + VL o tablas de decisión + cobertura de caminos

#### **7.2.1.2. Probar integración de varios componentes**

En general, considerar el uso de las técnicas de CE y cobertura de caminos, teniendo en cuenta que para aumentar el grado de exhaustividad de las pruebas se recurrirá a la combinación de atributos en el caso de CE, a aumentar el nivel utilizado en el caso de cobertura de caminos o a la combinación de técnicas.

### 7.2.1.3. Probar HMI

En general, considerar el uso de las técnicas de CE y cobertura de caminos, teniendo en cuenta los mismos criterios relativos a la exhaustividad citados anteriormente para las pruebas de integración de componentes.

Prestar atención a aspectos relativos al formato de campos donde el usuario final introduce datos: tipos de datos admitidos, longitud que admiten los campos, opcionalidad u obligatoriedad de los campos y uso del valor cero.

Considerar situaciones de prueba relacionadas con la sincronización de la información visualizada entre varios elementos de HMI en respuesta a acciones del usuario.

Tener en cuenta aspectos de configuración (online y offline) que afectan a la presentación de información.

Pensar en situaciones de prueba relacionadas con la ubicación de los elementos en el HMI, como por ejemplo solapamientos entre ventanas o diálogos.

### 7.2.2. Ejecución de pruebas

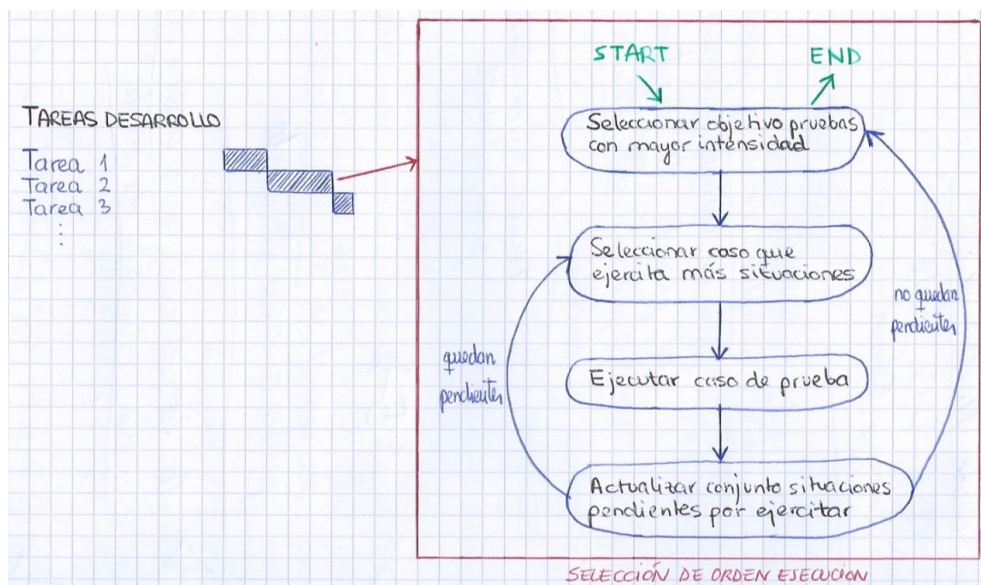
La salida del diseño de pruebas correspondiente a cada tarea de desarrollo consistirá en un determinado conjunto de casos de prueba.

Del total de situaciones de prueba derivadas (a partir de los objetivos de prueba que tenga asociados cada tarea de desarrollo), cada caso de prueba ejercitará un determinado número de ellas, de manera que el conjunto de todos los casos ejercite globalmente todas las situaciones.

Es importante tener en cuenta que, en la práctica, podría haber situaciones de prueba que no se puedan ejercitar por ser complejas de ejecutar y proporcionar un beneficio incremental pequeño. Cuando sucede esto, se puede relajar el objetivo ideal de ejercitar todas las situaciones de prueba.

El orden de ejecución de los casos de prueba es relevante desde el punto de vista del feedback que proporcionan: tener constancia de la existencia de fallos lo antes posible permite estimar el esfuerzo necesario para corregirlos y de esta manera es posible mantener actualizada la planificación del trabajo pendiente (podría ser necesario cambiar la prioridad de algunas tareas).

Un criterio para priorizar la ejecución de los casos en el entorno de pruebas puede consistir en ordenarlos. Suponiendo que las tareas de desarrollo están ordenadas atendiendo a riesgos:



**Figura 90.** Priorización en la ejecución de casos de prueba

El proceso de selección es iterativo e incremental, de manera que los resultados de ejecución de cada caso servirán para seleccionar el siguiente. Cuando el caso de prueba seleccionado pasa con éxito, se actualizará el conjunto de situaciones pendientes de ejercitar (situaciones que no cubre el caso de prueba ejecutado) y se seleccionará como siguiente caso de prueba el que ejercite un número mayor.

En caso de que varios objetivos de prueba tengan la misma intensidad, se podría deshacer el empate seleccionando aquel con la combinación *Daño x Probabilidad de fallo* más alta (por ejemplo, según la clasificación de riesgo de la Tabla 11, para el riesgo CRÍTICO las combinaciones posibles eran *Alto x Alto* y *Alto x Bajo*). El desarrollador podría consultar esta información en la estrategia de pruebas.

Si continúa existiendo empate, se podría deshacer escogiendo el objetivo de prueba para el que se estime menor esfuerzo de ejecución (en horas).

### 7.2.2.1. Opcional: pruebas de regresión basadas en riesgo

La regresión se experimenta cuando la calidad de un sistema se deteriora como consecuencia de haber introducido cambios.

El objetivo de las pruebas de regresión es verificar que tras la implementación de un cambio el resto de partes del sistema que no han sido modificadas funcionan correctamente.

La ejecución de estas pruebas consume por lo general bastante tiempo, de manera que es una buena práctica que los casos de prueba para hacer regresiones sean independientes entre sí, principalmente en escenarios donde se pueda verificar que es posible ejecutar pruebas en paralelo.

Una aproximación a la construcción de un conjunto de pruebas de regresión eficiente se basa en *etiquetar* con información adicional los casos de prueba que se van obteniendo del diseño.

En la práctica esto se podría hacer manteniendo un *repositorio* (no tiene porqué tratarse necesariamente de una base de datos, podría consistir por ejemplo en una hoja de cálculo) donde se van añadiendo referencias a los casos de prueba diseñados. Para cada referencia se anotará además un determinado conjunto de etiquetas.

Las etiquetas se utilizarán para confeccionar una clasificación de casos de prueba bajo algún criterio que permita seleccionarlos con vistas a un tipo concreto de pruebas (regresión, rendimiento, seguridad,...).



De cara a hacer pruebas de regresión, la clasificación más importante es en función del *peso* que tiene cada caso respecto a los demás del repositorio.

Otras clasificaciones posibles (para tipos de pruebas diferentes a regresión) podrían ser por ejemplo:

- objetivo de prueba
- componente (o subsistema)
- clase de riesgo
- requisito
- versión (release)

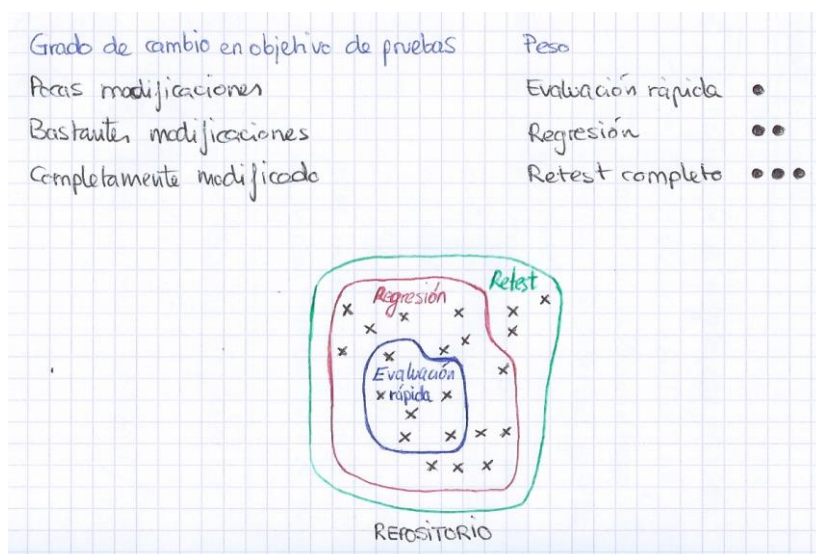
A continuación se describen los pasos necesarios para definir el peso:

### Determinar valores de la categoría peso

El peso estará directamente asociado al grado de cambios que sufre cada objetivo de prueba que se quiere evaluar.

En general, cuando se introducen cambios que afectan a algún objetivo de prueba, los factores daño y frecuencia de uso varían bastante poco, pero la probabilidad de presencia de defectos puede ser mayor. Debido a esta diferencia, en regresión se utiliza una clasificación en base a la categoría peso en lugar de una clasificación basada en las clases de riesgo determinadas en la estrategia de pruebas.

Se podría utilizar como base la siguiente graduación, añadiendo nuevos niveles en caso de considerarlo necesario:



**Figura 91.** Asignación de peso a los casos de prueba de regresión

De la figura anterior se puede deducir que existe una ordenación y que cada valor de peso incluye al anterior. Por ejemplo, al hacer una selección de casos de prueba de regresión para un objetivo de prueba que ha sufrido bastantes modificaciones, se seleccionarían los casos etiquetados con peso “Regresión” junto con los etiquetados con peso “Evaluación rápida”.

### Seleccionar número de casos de prueba para cada valor de la categoría peso

Para cada valor de peso sería necesario determinar el porcentaje de los casos de prueba del repositorio que resultan suficientes para evaluar ese nivel.

Para ello puede servir de referencia responder a las siguientes cuestiones:

- Evaluación rápida: ¿qué subconjunto de casos del repositorio sería necesario ejecutar para verificar que las partes del sistema que no han sufrido cambios se comportan exactamente igual que antes de introducir los cambios?
- Regresión: ¿qué subconjunto de casos del repositorio sería necesario ejecutar para verificar que las partes del sistema en las que se han introducido (unos pocos) cambios están experimentando una regresión?
- Re-test completo: ¿qué subconjunto de casos del repositorio sería necesario ejecutar para evaluar el resultado de profundas modificaciones en el sistema?

Las respuestas a estas preguntas permiten definir un margen para cada valor de la categoría peso, que se aplica a todo el repositorio. Unos márgenes orientativos podrían ser los siguientes:

- Evaluación rápida: 10-15%
- Regresión: 60-70%
- Re-test completo: 100%

### Asignar casos de prueba a cada valor de la categoría peso

Para cada objetivo de prueba se seleccionarían los casos que permitan evaluar dicho objetivo al nivel deseado:

- Evaluación rápida: los casos de prueba más elementales.
- Regresión: casos de prueba que ejercitan toda la funcionalidad *core* del sistema no ejercitada en la categoría anterior.
- Re-test completo: se seleccionarían todos los casos que no han sido seleccionados en categorías anteriores, priorizando aquellos que tienen un carácter muy excepcional o que ejercitan situaciones ideales.

En cada nivel, si el número de casos crece demasiado y se supera el margen prefijado, se identificaría si existen casos que ejercitan prácticamente la misma funcionalidad. Si ocurre esto, se pasarían los casos que introducen duplicidad al siguiente valor de la categoría peso.

Si el conjunto de casos seleccionados no llega al margen fijado, se podría dejar tal cual o se podrían diseñar nuevos casos de prueba.

De esta manera lo que se consigue es obtener un ranking de casos de prueba para cada objetivo de prueba en función del valor de la categoría peso.

### Prerrequisitos para la aplicación de pruebas de regresión basadas en riesgo

Las tareas descritas para la elaboración de un ranking de casos de prueba de regresión implican una serie de aspectos que es necesario tener en cuenta.

Los siguientes aspectos implican un esfuerzo adicional, dado que sería necesario implantarlos en el proceso actual de desarrollo de ISL Gijón y son el motivo por el que las pruebas de regresión basadas en riesgo se proponen como algo opcional:

Aspecto	Descripción	Afectados
Mantenimiento del repositorio de casos	Fundamental para poder aplicar de forma efectiva este enfoque de pruebas de regresión basadas en riesgo	Desarrolladores
Implantación de actividades en el proceso de desarrollo	Todo cambio en un proceso ya establecido implica costes que pueden ser difíciles de estimar	Responsables de área/proyecto

**Tabla 15.** Problemas estrategia de pruebas de regresión basadas en riesgo

## 8. Documentación de referencia

Los documentos siguientes servirán como referencia en este informe:

[BACH 2002] Bach, James, (2002) *Exploratory testing explained*, [Enlace](#)

[ISTQB] *Glosario Estándar de términos utilizados en Pruebas de Software*, V1.3.ES.0.915, [Enlace](#)

[MYERS 2004] Myers J., G., *The Art of Software Testing*, Second Edition, 2004

[MUPICT] *MANUAL DE USUARIO DE LA POSICIÓN INTEGRADA DE CONTROL DE TORRE (PICT)*, M.U. Posición PICT SPICTMUE.059\_1.pdf

[SC#7047] *SC#7047: SNETS A-SMGCS nivel 2*, SC#7047 - SNETS A-SMGCS nivel 2-JUNIO11 CAP 1 (25%) - (Rev4+Estimaciones).doc

[SCHAEFER1996] Schaefer, H., *Surviving under time and Budget pressure*, in Proceedings EuroSTAR Conference 1996, Amsterdam. [Enlace](#)

[SESAR12.03.02-12.05.03] *DISEÑO FUNCIONAL DE ALERTAS “CONFORMANCE MONITORING (CMON)” y “CONFLICTING ATC CLEARANCES”*, 12.03.02 Diseño funcional alertas prototipo.docx

[TESTCUBE] Specialist group Test Cube, *THE TEST CUBE Approach to structure and manage test sets*. [Enlace](#)

[TESTTYPES] *Overview "Applied test types"*. [Enlace](#)

[TMAP]. *TMap Next for result-driven testing*, Tim Koomen, Leo var der Aalst, Bart Broekman, Michiel Vroon, 2006

[TPI] Koomen, T., Pol, M., *Test Process Improvement: A Step-by-step Guide to Structured Testing*, 1999

## 9. CONTROL DE CAMBIOS

Fecha	Edic./Rev.	Razón del cambio
DD.MM.AAAA	Nº	

### REGISTRO DE EDICIONES Y REVISIONES DE PÁGINAS

Página	Edic/Rev.	Página	Edic/Rev.	Página	Edic/Rev.	Página	Edic/Rev.
1 a nº total	1						

Este documento es propiedad de INDRA no pudiendo ser usado con fines distintos de aquellos para los que ha sido entregado, ni reproducido, total o parcialmente, ni transmitido o comunicado a ninguna persona sin autorización del propietario.