VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE

Bibliografía

Fisher M. Software Verification and Validation. An Engineering and Scientific Approach. Ed. Springer. USA. 2007.

Agenda

- Verificación y Validación.
  - ✓Análisis de la Interfaz

Verificación y Validación del software

Verificación y validación del software

3.2	Análisis de Interfaz
3.2.1	V&V debe asegurar que han sido identificados los elementos de interfaz correctos.
3.2.2	V&V debe asegurar que todos los elementos de interfaz están completamente definidos.
3.2.3	V&V debe asegurar que cada elemento de interfaz es empleado consistentemente.
3.2.4	V&V debe asegurar que los elementos de la interfaz mantienen el rendimiento necesitado por el sistema.
3.2.5	V&V debe asegurar que todos los elementos de la interfaz son testeables.

Verificación y validación del software

Tareas genéricas:

Tarea 1. Identificar los datos que deben ser transferidos entre módulos.

Tarea 2. Identificar las interfaces que manejan las transacciones de los datos.

Tarea 3. Comparar las interfaces definidas por la V&V con las de los desarrolladores y evaluar las inconsistencias.

Verificación y validación del software

Tarea 4. Analizar cada dato y determinar si se encuentra definido completamente.

Tarea 5. Graficar la ubicación de los datos y determinar si se están empleando de forma consistente.

Tarea 6. Identificar las necesidades de rendimiento del sistema.

Verificación y validación del software

Tarea 7. Modelar y simular la comunicación entre las interfaces para determinar si mantienen el rendimiento requerido.

Tarea 8. Desarrollar pruebas para las interfaces e identificar cuáles no son testeables.

Verificación y validación del software

- Ejemplo: Proyecto MUGSEY

Objetivos

- Analiza y mantiene la salud del sistema de forma adecuada.
- Identifica y gestiona las fallas de forma adecuada.
- Adquiere, almacena y conserva los datos.
- Establece comunicación de forma confiable con la Tierra.
- Puede mantenerse fácilmente.

Verificación y validación del software

- Proveer la seguridad de que las interfaces del software soportan la identificación y gestión de fallas

Verificación y validación del software

- Los **requerimientos del sistema** que son de interés para la VyV, de acuerdo a este objetivo se definen en la siguiente tabla:

Verificación y validación del software

System Reqt. Number	System Requirement	Software Reqt. Number	Software Requirement
3.4.3.1	Stored Commanding	OS 2.4.5.1	FDIR (Fault Detection Isolation and Recovery)
6.3	Fault Handling	OS 2.1.6	Guidance Element – Update
		OS 2.1.7	Guidance Element – Invalidate
6.3.1	Science Faults	OS 2.4.2.1	Memory Manager
		OS 2.4.4	Watchdog Timer
6.3.2	Abort Mission	OS 2.4.5.1	FDIR

Verificación y validación del software

System Reqt. Number	System Requirement	Software Reqt. Number	Software Requirement
3.4.3.1	Stored Commanding	OS 2.4.5.1	FDIR (Fault Detection Isolation and Recovery)
6.3	Fault Handling	OS 2.1.6	Guidance Element – Update
		OS 2.1.7	Guidance Element – Invalidate
6.3.1	Science Faults	OS 2.4.2.1	Memory Manager
		OS 2.4.4	Watchdog Timer
6.3.2	Abort Mission	OS 2.4.5.1	FDIR

Verificación y validación del software

- Existen 5 datos de interés para la VyV en este ejemplo:

Data Item	Sent From	Sent Where
Position	GPS	Guidance Element
Validate Bit	GPS	Guidance Element
Altitude	C&DH Element	FDIR
Stored Command Index	FDIR	Command Processing
Commands	Command Processing	C&DH Element

Verificación y validación del software

Tarea 4. Determinar si los datos se encuentran definidos por completo.

Iniciamos con un checklist. El cuál se aplicará a cada uno de los ítem encontrados en la tabla “Datos de interés”:

1. Las unidades de medida que el dato representa.
2. La precisión requerida para el dato.
3. El rango de valores que el dato puede tomar.
4. El lapso de tiempo en el que el dato debe ser procesado.
5. La fuente del dato.
6. El destino del dato.

Verificación y validación del software

Dato	Item
Dato	Las unidades de medida que el dato representa.
	La precisión requerida para el dato.
	El rango de valores que el dato puede tomar.
	El lapso de tiempo en el que el dato debe ser procesado.
	La fuente del dato
	El destino del dato.

Verificación y validación del software

Dato	Condición
altitud	Pies
	Precisión de 5 dígitos
	Rango de 0 a 99000
	10 segundos
	GPS
	C&DH

Verificación y validación del software

Tarea 5. Graficar la ubicación de los datos y determinar si se están empleando de forma consistente.

Ejecutada para asegurar que el dato siempre es utilizado en la forma que fue previamente definido.

Continuando con el ejemplo Mugsey. El equipo VyV debe buscar todas las ocurrencias del dato *altitud*, y asegurarse que ha sido empleado como se definió en la Tarea 4.

Verificación y validación del software

Dato	Condición	Fuente	
altitud	Pies	Nombre del archivo	☑
	Precisión de 5 dígitos		☑
	Rango de 0 a 99000		x
latitud	Pies	Nombre del archivo	☑
	Precisión de 5 dígitos		x
	Rango de 0 a 99000		x

Verificación y validación del software

Las siguientes tareas son realizadas para cubrir el requerimiento 3.2.4. V&V debe asegurar que los elementos de la interfaz mantienen el rendimiento necesitado por el sistema.

VyV debe asegurar que los elementos de la interfaz mantengan el rendimiento requerido por el sistema.

Tarea 6. Identificar las necesidades de rendimiento del sistema.

Tarea 7. Modelar y simular la comunicación entre las interfaces para determinar si mantienen el rendimiento requerido.

Para ello se deben extraer los requerimientos de rendimiento del sistema y evaluar las interfaces para asegurar que mantienen estas condiciones de rendimiento.

Verificación y validación del software

La Tarea 6 y 7 pueden convertirse en evaluaciones muy especializadas.

Para las fases de requerimientos y diseño se evaluarán a través del enfoque de **Modelado** y posiblemente de **Simulación**.

Para sistemas de software implementados, puede emplearse Análisis estático o ejecutarlo para extraer los comportamientos de rendimiento.

Verificación y validación del software

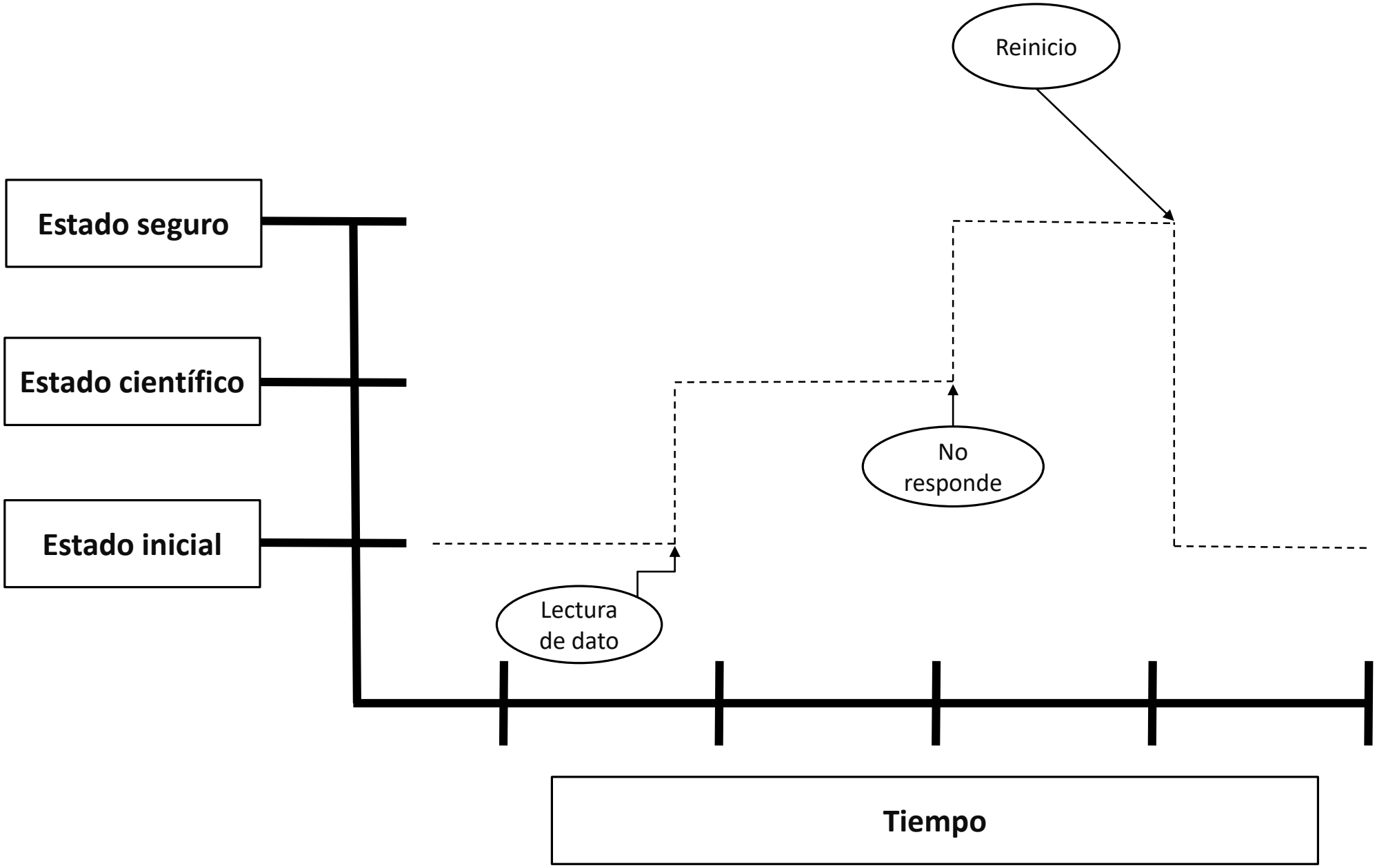
En la fase de análisis de requerimientos es posible modelar los requerimientos y analizar el tiempo. Para ello se emplean diagramas de secuencia y diagramas de tiempo.

Los diagramas de tiempo han sido muy populares sobretodo entre los ingenieros eléctricos.

Verificación y validación del software

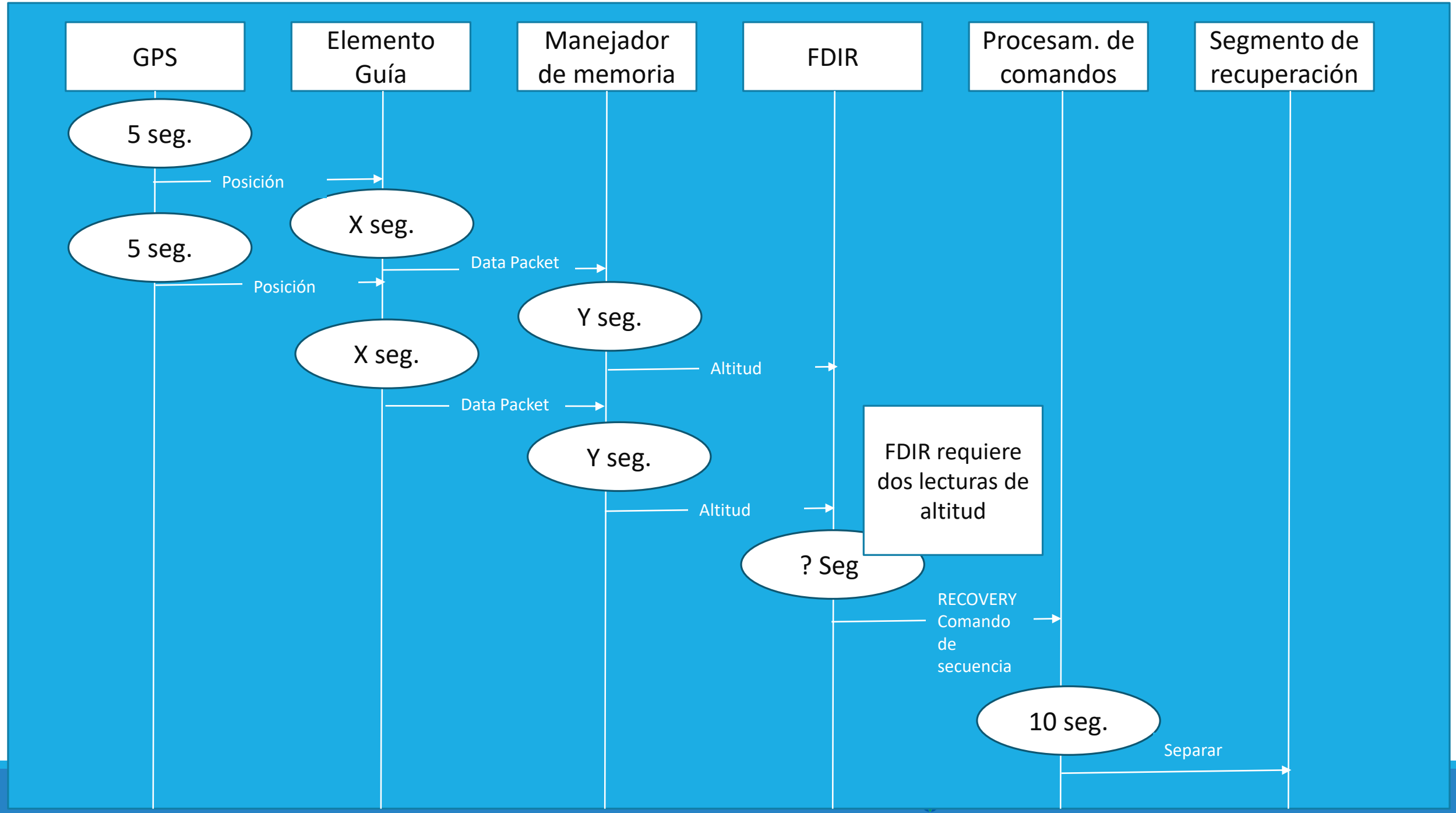
Consisten en un gráfico donde el tiempo es el eje horizontal y los posibles estados del software se ubican en el eje vertical. El evento que dispara la transición entre estados se representa en burbujas.

El eje vertical también puede representar tareas o módulos.


Reinicio
Estado seguro
Estado científico
Estado inicial
No responde
Lectura de dato
Tiempo

Verificación y validación del software

La comunicación entre módulos puede ser representada a través de diagramas de secuencia.


GPS
Elemento Guía
Manejador de memoria
FDIR
Procesam. de comandos
Segmento de recuperación
5 seg.
Posición
X seg.
5 seg.
Data Packet
Posición
Y seg.
X seg.
Altitud
Data Packet
Y seg.
FDIR requiere dos lecturas de altitud
Altitud
? Seg
RECOVERY Comando de secuencia
10 seg.
Separar

Verificación y validación del software

La Tarea 8. Consiste en determinar si las interfaces pueden ser verificadas mediante testeo.

Tarea 8. Desarrollar pruebas para las interfaces e identificar cuáles no son testeables.

Planear/desarrollar pruebas para los datos; de forma indirecta obtendremos cuáles no son testeables…(Última fase del documento).

Verificación y validación del software

1. ¿Los desarrolladores han definido las interfaces necesarias?
2. ¿Estas interfaces se encuentran completamente definidas?
3. ¿Las interfaces son usadas de manera consistente?
4. ¿Las interfaces mantienen el rendimiento requerido por el sistema?
5. ¿Estas interfaces pueden ser verificadas mediante testeo?

GRACIAS POR SU ATENCIÓN