

objetos de datos fluyen por el sistema. Un segundo enfoque del modelado de análisis, llamado *análisis orientado a objetos*, se centra en la definición de clases y en el modo en el que colaboran una con otra para cumplir con los requerimientos del cliente.

Aunque el modelo de análisis que se propone en este libro combina características de ambos enfoques, es frecuente que los equipos del software elijan uno de ellos y excluyan las representaciones del otro. La pregunta no es cuál es mejor, sino qué combinación de representaciones dará a los participantes el mejor modelo de los requerimientos del software y cuál será el mejor puente para cruzar la brecha hacia el diseño del software.

7.2 MODELADO ORIENTADO AL FLUJO

Aunque algunos ingenieros de software perciben el modelado orientado al flujo como una técnica obsoleta, sigue siendo una de las notaciones más usadas actualmente para hacer el análisis de los requerimientos.¹ Si bien el *diagrama de flujo de datos* (DFD) y la información relacionada no son una parte formal del UML, se utilizan para complementar los diagramas de éste y amplían la perspectiva de los requerimientos y del flujo del sistema.

El DFD adopta un punto de vista del tipo entrada-proceso-salida para el sistema. Es decir, los objetos de datos entran al sistema, son transformados por elementos de procesamiento y los objetos de datos que resultan de ello salen del software. Los objetos de datos se representan con flechas con leyendas y las transformaciones, con círculos (también llamados burbujas). El DFD se presenta en forma jerárquica. Es decir, el primer modelo de flujo de datos (en ocasiones llamado DFD de nivel 0 o *diagrama de contexto*) representa al sistema como un todo. Los diagramas posteriores de flujo de datos mejoran el diagrama de contexto y dan cada vez más detalles en los niveles siguientes.



CONSEJO

Algunas personas afirman que los DFD son obsoletos y que no hay lugar para ellos en la práctica moderna. Ese punto de vista excluye un modo potencialmente útil de representación en el nivel del análisis. Si ayuda, use DFD.



Cita:

"El propósito de los diagramas de flujo de datos es proveer un puente semántico entre los usuarios y los desarrolladores de sistemas."

Kenneth Kozar

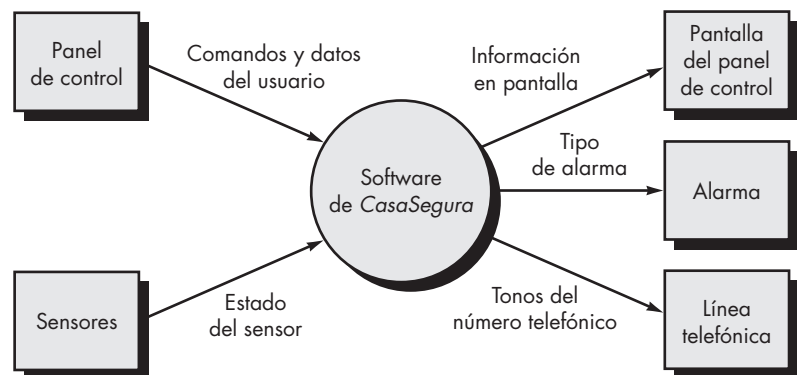
7.2.1 Creación de un modelo de flujo de datos

El diagrama de flujo de datos permite desarrollar modelos del dominio de la información y del dominio funcional. A medida que el DFD se mejora con mayores niveles de detalle, se efectúa la descomposición funcional implícita del sistema. Al mismo tiempo, la mejora del DFD da como resultado el refinamiento de los datos conforme avanzan por los procesos que constituyen la aplicación.

Unos cuantos lineamientos sencillos ayudan muchísimo durante la elaboración del diagrama de flujo de los datos: 1) el nivel 0 del diagrama debe ilustrar el software o sistema como una sola

FIGURA 7.1

DFD en el nivel de contexto para la función de seguridad de CasaSegura



¹ El modelado del flujo de datos es una actividad fundamental del *análisis estructurado*.

PUNTO CLAVE

Conforme se mejora cada nivel del DFD, debe mantenerse la continuidad del flujo de la información. Esto significa que las entradas y salidas en cierto nivel deben ser las mismas en un nivel mejorado.

CONSEJO

El análisis gramatical no es a prueba de todo, pero da un impulso excelente para arrancar si se tienen dificultades para definir objetos de datos y las transformaciones que operan sobre ellos.

CONSEJO

Asegúrese de que la narración del procesamiento que se va a analizar gramaticalmente está escrita con el mismo nivel de abstracción.

burbuja; 2) debe anotarse con cuidado las entradas y salidas principales; 3) la mejora debe comenzar por aislar procesos candidatos, objetos de datos y almacenamiento de éstos, para presentarlos en el siguiente nivel; 4) todas las flechas y burbujas deben etiquetarse con nombres significativos; 5) de un nivel a otro, debe mantenerse la *continuidad del flujo de información*,² y 6) debe mejorarse una burbuja a la vez. Existe la tendencia natural a complicar innecesariamente el diagrama de flujo de los datos. Esto sucede cuando se trata de ilustrar demasiados detalles en una etapa muy temprana o representar aspectos de procedimiento del software en lugar del flujo de la información.

Para ilustrar el uso del DFD y la notación relacionada, consideremos de nuevo la función de seguridad de *CasaSegura*. En la figura 7.1 se muestra un DFD de nivel 0 para dicha función. Las *entidades externas* principales (cuadrados) producen información para uso del sistema y consumen información generada por éste. Las flechas con leyendas representan objetos de datos o jerarquías de éstos. Por ejemplo, los **comandos** y **datos del usuario** agrupan todos los comandos de configuración, todos los comandos de activación/desactivación, todas las diferentes interacciones y todos los datos que se introducen para calificar o expandir un comando.

Ahora debe expandirse el DFD de nivel 0 a un modelo de flujo de datos de nivel 1. Pero, ¿cómo hacerlo? Según el enfoque sugerido en el capítulo 6, debe aplicarse un “análisis gramatical” [Abb83] a la narración del caso de uso que describe la burbuja en el nivel del contexto. Es decir, se aíslan todos los sustantivos (y frases sustantivadas) y verbos (y frases verbales) en la narración del procesamiento de *CasaSegura* obtenida durante la primera reunión realizada para recabar los requerimientos. Recordemos el análisis gramatical del texto que narra el procesamiento presentado en la sección 6.5.1:

La función de seguridad CasaSegura permite que el propietario configure el sistema de seguridad cuando se instala, vigila todos los sensores conectados al sistema de seguridad e interactúa con el propietario a través de internet, una PC o un panel de control.

Durante la instalación, la PC de CasaSegura se utiliza para programar y configurar el sistema. Se asigna a cada sensor un número y un tipo, se programa una clave maestra para activar y desactivar el sistema, y se introducen números telefónicos para marcar cuando ocurre un evento de sensor.

Cuando se reconoce un evento de sensor, el software invoca una alarma audible instalada en el sistema. Después de un tiempo de retraso que especifica el propietario durante las actividades de configuración del sistema, el software marca un número telefónico de un servicio de monitoreo, proporciona información acerca de la ubicación y reporta la naturaleza del evento detectado. El número telefónico vuelve a marcarse cada 20 segundos hasta que se obtiene la conexión telefónica.

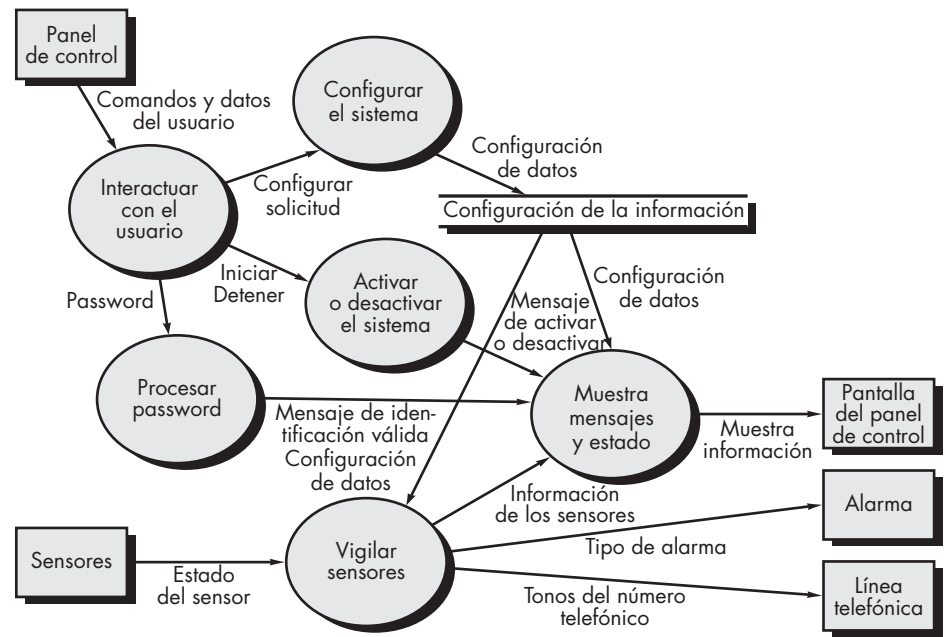
El propietario recibe información de seguridad a través de un panel de control, la PC o un navegador, lo que en conjunto se llama interfaz. La interfaz despliega en el panel de control, en la PC o en la ventana del navegador mensajes de aviso e información del estado del sistema. La interacción del propietario tiene la siguiente forma...

En relación con el análisis gramatical, los verbos son los procesos de *CasaSegura* y se representarán como burbujas en un DFD posterior. Los sustantivos son entidades externas (cuadros), datos u objetos de control (flechas) o almacenamiento de datos (líneas dobles). De lo estudiado en el capítulo 6 se recuerda que los sustantivos y verbos se asocian entre sí (por ejemplo, a cada sensor se asigna un número y tipo; entonces, **número** y **tipo** son atributos del objeto de datos **sensor**). De modo que al realizar un análisis gramatical de la narración de procesamiento en cualquier nivel del DFD, se genera mucha información útil sobre la manera de proceder para la mejora del nivel siguiente. En la figura 7.2 se presenta un DFD de nivel 1 con el empleo de esta información. El proceso en el nivel de contexto que se ilustra en la figura 7.1 ha sido expandido

2 Es decir, los objetos de datos que entran al sistema o a cualquier transformación en cierto nivel deben ser los mismos objetos de datos (o sus partes constitutivas) que entran a la transformación en un nivel mejorado.

FIGURA 7.2

DFD de nivel 1
para la función
de seguridad de
CasaSegura



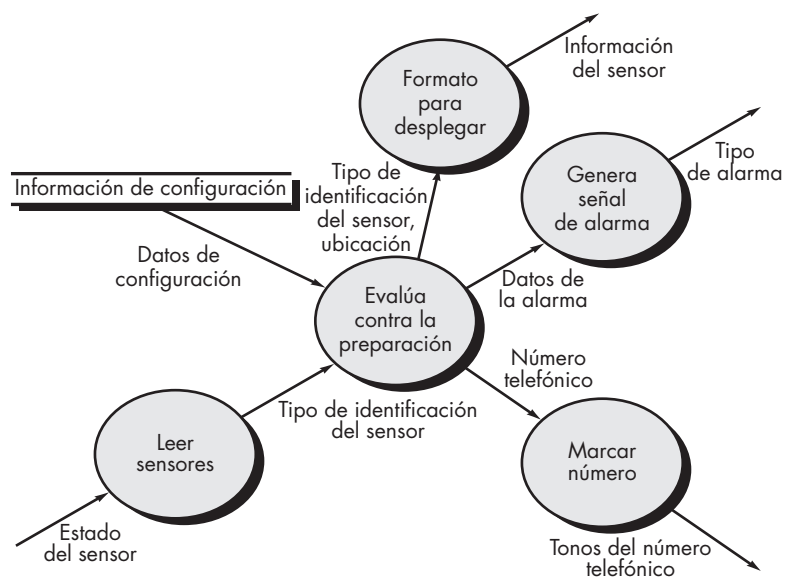
a seis procesos derivados del estudio del análisis gramatical. De manera similar, el flujo de información entre procesos del nivel 1 ha surgido de dicho análisis. Además, entre los niveles 0 y 1 se mantiene la continuidad del flujo de información.

Los procesos representados en el nivel 1 del DFD pueden mejorarse más hacia niveles inferiores. Por ejemplo, el proceso *vigilar sensores* se mejora en el DFD de nivel 2, como se aprecia en la figura 7.3. De nuevo, observe que entre los niveles se ha mantenido la continuidad del flujo de información.

La mejora de los DFD continúa hasta que cada burbuja realiza una función simple. Es decir, hasta que el proceso representado por la burbuja ejecuta una función que se implementaría fácilmente como componente de un programa. En el capítulo 8 se estudia un concepto llamado

FIGURA 7.3

DFD de nivel 2
que mejora el
proceso *vigilar
sensores*



cohesión, que se utiliza para evaluar el objeto del procesamiento de una función dada. Por ahora, se trata de mejorar los DFD hasta que cada burbuja tenga “un solo pensamiento”.

7.2.2 Creación de un modelo de flujo de control

Para ciertos tipos de aplicaciones, el modelo de datos y el diagrama de flujo de datos es todo lo que se necesita para obtener una visión significativa de los requerimientos del software. Sin embargo, como ya se dijo, un gran número de aplicaciones son “motivadas” por eventos y no por datos, producen información de control en lugar de reportes o pantallas, y procesan información con mucha atención en el tiempo y el desempeño. Tales aplicaciones requieren el uso del *modelado del flujo de control*, además de modelar el flujo de datos.

Se dijo que un evento o aspecto del control se implementa como valor booleano (por ejemplo, verdadero o falso, encendido o apagado, 1 o 0) o como una lista discreta de condiciones (vacío, bloqueado, lleno, etc.). Se sugieren los lineamientos siguientes para seleccionar eventos candidatos potenciales:

? ¿Cómo seleccionar los eventos potenciales para un diagrama de flujo de control, de estado o CSPEC?

- Enlistar todos los sensores que son “leídos” por el software.
- Enlistar todas las condiciones de interrupción.
- Enlistar todos los “interruptores” que son activados por un operador.
- Enlistar todas las condiciones de los datos.
- Revisar todos los “aspectos de control” como posibles entradas o salidas de especificación del control, según el análisis gramatical de sustantivos y verbos que se aplicó a la narración del procesamiento.
- Describir el comportamiento de un sistema con la identificación de sus estados, identificar cómo se llega a cada estado y definir las transiciones entre estados.
- Centrarse en las posibles omisiones, error muy común al especificar el control; por ejemplo, se debe preguntar: “¿hay otro modo de llegar a este estado o de salir de él?”

Entre los muchos eventos y aspectos del control que forman parte del software de *CasaSegura*, se encuentran **evento de sensor** (por ejemplo, un sensor se descompone), **bandera de cambio** (señal para que la pantalla cambie) e **interruptor iniciar/detener** (señal para encender o apagar el sistema).

7.2.3 La especificación de control

Una *especificación de control* (CSPEC) representa de dos maneras distintas el comportamiento del sistema (en el nivel desde el que se hizo referencia a él).³ La CSPEC contiene un diagrama de estado que es una especificación secuencial del comportamiento. También puede contener una tabla de activación del programa, especificación combinatoria del comportamiento.

La figura 7.4 ilustra un diagrama de estado preliminar⁴ para el nivel 1 del modelo de flujo de control para *CasaSegura*. El diagrama indica cómo responde el sistema a eventos conforme pasa por los cuatro estados definidos en este nivel. Con la revisión del diagrama de estado se determina el comportamiento del sistema, y, lo que es más importante, se investiga si existen “agujeros” en el comportamiento especificado.

Por ejemplo, el diagrama de estado (véase la figura 7.4) indica que las transiciones del estado **Ocioso** ocurren si el sistema se reinicia, se activa o se apaga. Si el sistema se activa (por ejem-

³ En la sección 7.3 se presenta notación adicional de modelado por comportamiento.

⁴ La notación del diagrama de estado que se emplea aquí sigue la del UML. En el análisis estructurado se dispone de un “diagrama de transición de estado”, pero el formato UML es mejor en contenido y representación de la información.