

Modelado y simulación de eventos discretos

ALFONSO URQUÍA MORALEDA
CARLA MARTÍN VILLALBA

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

OBJETIVOS DOCENTES

Una vez estudiado el contenido del tema y realizados los ejercicios prácticos, debería saber:

- Discutir el significado de los conceptos “sistema”, “modelo”, “experimento” y “simulación”.
- Discutir en qué situaciones puede ser imposible o desaconsejable experimentar con el sistema real.
- Describir y comparar las diferentes formas de estudiar un sistema y los diferentes tipos de modelos.
- Comparar y reconocer los distintos tipos de modelos matemáticos.
- Discutir los niveles en el conocimiento de los sistemas de la clasificación de Klir, y el significado de los conceptos “análisis”, “inferencia” y “diseño” de sistemas con relación a dicha clasificación del conocimiento.
- Discutir un marco formal para el modelado y la simulación en el que se definen 5 entidades (sistema fuente, base de datos del comportamiento, modelo, simulador y marco experimental) y dos relaciones entre ellas (relación de modelado y relación de simulación).
- Describir las diferencias entre los distintos tipos de modelos matemáticos y saber reconocer a qué tipo pertenece cualquier modelo.
- Discutir qué características debe tener un modelo para ser de tiempo discreto y cómo se realiza la descripción y simulación de este tipo de modelos.
- Discutir qué es un autómata celular y cómo se realiza su descripción y simulación.
- Discutir qué características tienen los modelos de eventos discretos y las diferencias entre su descripción orientada a la planificación de eventos y orientada a los procesos.
- Discutir los pasos de que típicamente consta un estudio de simulación en el cual se emplean modelos estocásticos de eventos discretos.
- Emplear el lenguaje R para realizar programas sencillos y representaciones gráficas sencillas de los datos.

1.1. INTRODUCCIÓN

En este primer tema se definen algunos de los conceptos más comúnmente empleados en el ámbito del modelado y la simulación, como son “modelo”, “sistema”, “experimento” y “simulación”. Se describen cuatro tipos diferentes de modelos, uno de los cuales es el empleado en simulación: el modelo matemático.

En la teoría de sistemas se diferencian dos aspectos fundamentales, ortogonales entre sí. Por una parte está el **conocimiento** del que se dispone acerca del comportamiento de un sistema. Por otra parte está la **especificación** de dicho conocimiento, que puede realizarse empleando uno u otro tipo de modelo matemático.

En este tema se muestra una clasificación en cuatro niveles del **conocimiento** que puede poseerse acerca de un sistema. Esta clasificación facilita la definición de las tres actividades fundamentales relacionadas con los sistemas: el análisis, la inferencia y el diseño.

Se describe asimismo un marco formal para el modelado y la simulación, en el cual se establecen unas entidades, como son el sistema fuente, el modelo y el simulador, y también la relación entre ellas. De este marco formal se deriva de forma natural la definición de conceptos básicos como son “marco experimental”, “simulador”, “validez” del modelo, “corrección” del simulador, “complejidad” del modelo y “simplificación” del modelo, entre otros.

Ahondando en lo que respecta a la **especificación** del conocimiento, se describen diferentes clasificaciones de los modelos matemáticos que responden a diferentes criterios. Se pone especial énfasis en la descripción de los tipos de modelos que aquí nos ocupan: los modelos matemáticos de tiempo discreto y de eventos discretos.

Se explican también los pasos de los que típicamente consta un estudio de simulación, cuando en él se emplean modelos estocásticos de eventos discretos. Estos pasos comprenden la definición del problema, la planificación del proyecto, la definición del sistema, las diferentes etapas en la definición del modelo y del experimento, la experimentación con el modelo, el análisis e interpretación de los resultados, y la documentación de las conclusiones.

Finalmente, se introduce una de las múltiples herramientas software existentes para el análisis de datos: el lenguaje R. Dicho análisis es una parte esencial en el planteamiento del modelo matemático y también en la toma de decisiones basada en los resultados de la simulación de los modelos.

gicos, de evolución de los ecosistemas y sociológicos, en los cuales los experimentos deberían tener una duración de cientos, miles o millones de años.

Aun siendo posible experimentar directamente con el sistema real, en ocasiones es desaconsejable debido a su elevado coste económico. Consideremos un empresario que debe decidir si ampliar o no las instalaciones de su fábrica, para lo cual necesita estimar si la ganancia potencial que supondrán las nuevas instalaciones justifica el coste que tendrá realizar la ampliación. Experimentar directamente con el sistema real supondría realizar la ampliación con el fin de evaluar su rendimiento económico, lo cual no parece razonable.

Otro posible motivo para no experimentar con el sistema real es que el experimento produzca perjuicio, incomodidad o sea peligroso. Así, experimentar con un nuevo sistema de facturación en un aeropuerto puede producir retrasos y problemas imprevisibles que perjudiquen al viajero. Un ejemplo de experimento peligroso sería usar el sistema real para entrenar a los operarios de una central nuclear acerca de cómo deben reaccionar ante situaciones de emergencia.

Una alternativa a la experimentación con el sistema real consiste en realizar un modelo del sistema y experimentar con el modelo. En el sentido amplio del término, un **modelo** es *una representación de un sistema desarrollada para un propósito específico*. Volviendo al ejemplo de la ampliación de la fábrica, en lugar de experimentar con el sistema real puede realizarse un modelo de la operación de cada una de las configuraciones de la fábrica (la actual y la ampliada) y comparar el comportamiento de los modelos.

Como hemos visto, en ocasiones experimentar con un modelo resulta menos costoso y más seguro que experimentar directamente con el sistema real. Otra ventaja de la experimentación con modelos es que, con un modelo adecuado, se pueden ensayar condiciones de operación extremas que son impracticables en el sistema real. Este es el caso cuando el experimento requiere modificar variables que en el sistema real o bien no están accesibles, o bien no pueden ser modificadas en el rango requerido.

Resumiendo lo anterior, en la Figura 1.1 se muestran las dos formas de conocer el comportamiento de un sistema: experimentando con el sistema real y experimentando con un modelo del sistema. Obsérvese también que en la figura se señalan cuatro tipos diferentes de modelo: mental, verbal, físico y matemático. A continuación se explican las características de cada uno de ellos.

En nuestra vida cotidiana empleamos continuamente **modelos mentales** para comprender y predecir el comportamiento de los sistemas. Considerar que alguien es

1.2. CONCEPTOS FUNDAMENTALES

Una forma de aproximarse a una disciplina es entendiendo el significado de los términos que describen sus conceptos fundamentales. Con este propósito se explica a continuación el significado de términos que describen conceptos fundamentales en el ámbito del modelado y la simulación.

1.2.1. Sistema, experimento y modelo

El modelado matemático y la simulación por ordenador es una forma de adquirir conocimiento acerca del comportamiento de los sistemas. A la vista de ello, cabe plantearse en primer lugar qué se entiende por sistema. Puede considerarse que un **sistema** es *cualquier objeto cuyas propiedades se desean estudiar*. De acuerdo con esta definición, *cualquier fuente potencial de datos* puede considerarse que es un sistema.

Así, por ejemplo, una fábrica con máquinas, personal y almacén sería un sistema. También lo sería el servicio de emergencias de un hospital, incluyendo el personal, las salas, el equipamiento y el transporte de los pacientes. Sería igualmente un sistema una red de ordenadores con servidores, clientes, dispositivos de disco e impresoras, como también lo sería un supermercado con control de inventario, cajeros y atención al cliente, o un parque temático con atracciones, tiendas, restaurantes, trabajadores, clientes y aparcamientos, por citar únicamente algunos ejemplos.

Una manera de conocer el comportamiento de un sistema es experimentar con él. De hecho, éste ha sido el método empleado durante siglos para avanzar en el conocimiento: plantear preguntas acerca del comportamiento de los sistemas y responderlas mediante experimentación. Un **experimento** es *el proceso de extraer datos de un sistema sobre el cual se ha ejercido una acción externa*. Por ejemplo, el encargado de un supermercado puede ensayar diferentes procedimientos de control del inventario y de distribución del personal para determinar qué combinación muestra un mejor equilibrio entre coste y calidad de servicio.

Experimentar directamente con el sistema real presenta indudables ventajas. Sin embargo, en ocasiones no es posible. Quizá el motivo más evidente es que el sistema real aun no exista físicamente. Esto sucede por ejemplo en la fase de diseño de nuevos sistemas, cuando el ingeniero necesita predecir el comportamiento de los mismos antes de que sean construidos. Otro motivo es que la escala de tiempo del experimento lo haga inviable. Esto es habitual en los estudios geológicos, cosmológicos

matemáticos. Igualmente, en todas las ramas de la Ingeniería se emplean modelos matemáticos para describir el funcionamiento de los sistemas naturales y artificiales.

En algunos casos, las relaciones matemáticas que constituyen los modelos son sencillas y puede encontrarse una **solución analítica** del modelo. Sin embargo, en la mayoría de los casos los modelos no pueden resolverse analíticamente y deben estudiarse con ayuda del ordenador, aplicando métodos numéricos. Este *experimento numérico realizado sobre el modelo matemático* recibe el nombre de **simulación**.

A partir de este punto nos enfocaremos en los modelos de tipo matemático y en su resolución mediante simulación por ordenador. Los modelos mentales, verbales y físicos quedan fuera del alcance de este texto. Emplearemos en lo sucesivo el término “modelo” para referirnos a modelo matemático. También está fuera del alcance de este texto la resolución analítica de los modelos matemáticos.

1.2.2. Niveles en el conocimiento de los sistemas

Existen varias formas de clasificar el conocimiento que puede poseerse de un sistema. A principios de la década de 1970, G.J. Klir propuso una clasificación del conocimiento en cuatro niveles, que resulta muy adecuada en el ámbito del modelado y la simulación. A continuación se indica qué tipo de conocimiento se posee acerca del sistema en cada nivel, de acuerdo con esta clasificación. A medida que se asciende de nivel, se conocen aspectos importantes del sistema que no se conocen en los niveles inferiores del conocimiento.

- **Nivel 0 - Fuente.** En este nivel identificamos la porción del mundo real a modelar y las maneras mediante las cuáles vamos a observarlo. Dicha porción del mundo real, que es nuestra fuente de datos, se denomina sistema fuente.
- **Nivel 1 - Datos.** En este nivel disponemos de una base de datos de medidas y observaciones de nuestro sistema fuente.
- **Nivel 2 - Generación.** En este nivel somos capaces de recrear estos datos usando una representación más compacta. Por ejemplo, mediante fórmulas matemáticas o algoritmos. Dado que un mismo conjunto de datos puede ser generado empleando diferentes fórmulas, algoritmos u otros procedimientos, haber determinado el procedimiento para reproducir los datos es un conocimiento que no teníamos al Nivel 1 (datos).
- **Nivel 3 - Estructura.** En este último nivel sabemos cómo recrear los datos observados en el Nivel 1 (datos) de una manera específica: en términos de

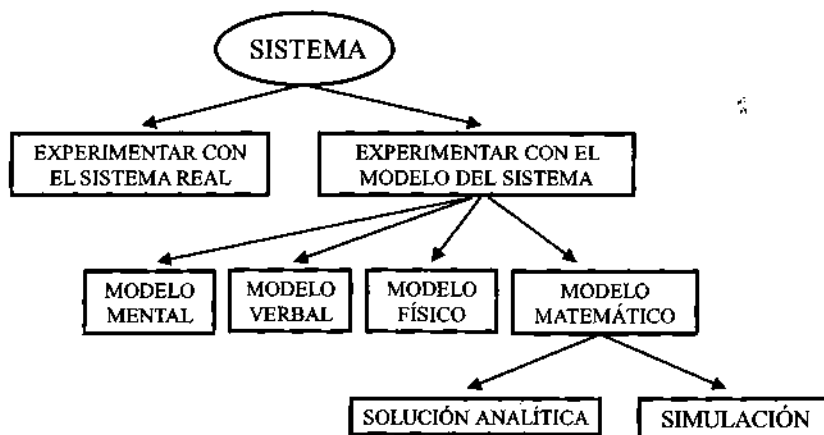


Figura 1.1: Formas de estudiar un sistema.

“amable” constituye un modelo del comportamiento de esa persona. Este modelo nos ayuda a responder, por ejemplo, a la pregunta de cómo reaccionará si le pedimos un favor. También, mediante la intuición y la experiencia, desarrollamos modelos mentales de los sistemas técnicos. Por ejemplo, aprender a conducir un coche consiste parcialmente en desarrollar un modelo mental de las propiedades de la conducción del coche. Asimismo, un operario trabajando en determinado proceso industrial sabe cómo el proceso reacciona ante diferentes acciones: el operario, mediante el entrenamiento y la experiencia, ha desarrollado un modelo mental del proceso.

Otro tipo de modelo es el **modelo verbal**, en el cual el comportamiento del sistema es descrito mediante palabras. Por ejemplo, un modelo mental es: si se aprieta el freno, entonces la velocidad del coche se reduce. Los sistemas expertos son ejemplos de modelos verbales formalizados. Es importante diferenciar entre los modelos mentales y los verbales. Por ejemplo, usamos un modelo mental de la dinámica de la bicicleta cuando la conducimos. Sin embargo, no es sencillo convertirlo en un modelo verbal.

Además de los modelos mentales y verbales, existe otro tipo de modelos que tratan de imitar al sistema real. Son los **modelos físicos**, tales como las maquetas a escala que construyen los arquitectos, diseñadores de barcos o aeronaves, para comprobar las propiedades estéticas, aerodinámicas, etc.

Finalmente, el cuarto tipo de modelo es el **modelo matemático**. En los modelos matemáticos las relaciones entre las magnitudes de interés del sistema (distancias, velocidades, flujos, etc.) son descritas mediante relaciones matemáticas. La mayoría de las teorías sobre las Leyes de la Naturaleza son descritas empleando modelos

1.2.3. Marco formal para el modelado y la simulación

En esta sección se presenta un marco formal para el modelado matemático y la simulación, en el cual se definen cinco entidades (sistema fuente, base de datos del comportamiento, modelo, simulador y marco experimental) y dos relaciones entre ellas (relación de modelado y relación de simulación). En la Figura 1.2 se han representado esquemáticamente las cinco entidades y las dos relaciones. Se describen a continuación.

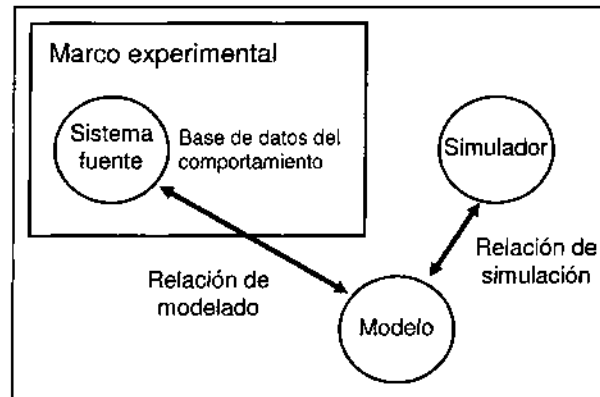


Figura 1.2: Entidades básicas del modelado y simulación, y su relación.

El **sistema fuente** es el entorno real o virtual que estamos interesados en modelar, el cual constituye una *fuentes de datos observables*, en la forma de trayectorias (observaciones indexadas en el tiempo) de variables. Esta entidad es conocida en el Nivel 0 (fuente) de conocimiento del sistema.

Los datos que se han recogido a partir de observaciones o experimentando con el sistema se llaman **base de datos del comportamiento** del sistema. Estas observaciones son características del conocimiento en el Nivel 1 (datos). Los datos son observados o adquiridos a través de marcos experimentales de interés para quien realiza el modelo. Un **marco experimental** es una especificación de las condiciones bajo las cuales el sistema es observado o se experimenta con él. Es la formulación operacional de los objetivos que motivan un proyecto de modelado y simulación.

Hay dos formas igualmente válidas de entender qué es un marco experimental. En la primera, el marco experimental es la definición del tipo de los datos que se incluirán en la base de datos del comportamiento del sistema. En la segunda, el marco experimental es un sistema que interactúa con el sistema de interés para obtener observaciones bajo determinadas condiciones experimentales. En esta se-

componentes interconectados entre sí. Esto implica un conocimiento acerca de la estructura interna del sistema.

Los modelos constituyen una formalización del conocimiento en los Niveles 2 (generación) y 3 (estructura). Por otra parte, esta clasificación permite definir en qué consisten los tres tipos básicos de problemas relacionados con el conocimiento de los sistemas: análisis, inferencia y diseño. Cada uno de estos problemas implica un determinado movimiento entre los niveles de conocimiento del sistema.

- Al realizar el **análisis** de un sistema se intenta comprender el comportamiento del sistema, existente o hipotético, empleando para ello el conocimiento que se tiene de su estructura.
- En la **inferencia** sobre un sistema se intenta conocer la estructura del sistema a partir de las observaciones que pueden realizarse del mismo.
- En el **diseño** de un sistema se investigan diferentes estructuras alternativas para un sistema completamente nuevo o para el rediseño de uno ya existente.

Cuando nos movemos hacia niveles inferiores del conocimiento, como sucede en el caso del análisis de sistemas, no estamos generando conocimiento nuevo. Estamos únicamente haciendo explícito lo que ya está implícito en la descripción que tenemos. Por el contrario, la inferencia y el diseño son problemas que requieren ascender en los niveles de conocimiento. Al hacer inferencia disponemos de una base de datos del comportamiento del sistema fuente y tratamos de encontrar una representación del conocimiento al Nivel 2 (generación) o al Nivel 3 (estructura), que nos permita recrear los datos de que disponemos. Este proceso se denomina **construcción del modelo**.

En el caso de la inferencia, el sistema fuente existe. Sin embargo, en el caso del diseño el sistema fuente no existe y el objetivo es construir un sistema que se comporte de la manera deseada. Si el objetivo es llegar a construir el sistema, debe llegarse a un Nivel 3 (estructura) del conocimiento, puesto que la construcción se realizará mediante la interconexión de diferentes componentes tecnológicos.

Finalmente, el proceso denominado **ingeniería inversa** tiene elementos tanto de inferencia como de diseño. Para hacer ingeniería inversa de un sistema existente, en primer lugar se realiza un gran número de observaciones de él. A partir de estas observaciones, se infiere el comportamiento del sistema y se diseña una estructura alternativa que tenga ese comportamiento.

2. Para calcular esas medidas, el modelo deberá incluir ciertas variables (llamadas **variables de salida**) cuyo valor deberá calcularse durante la simulación del modelo.
3. El cálculo de las medidas de salida a partir de las variables de salida se realiza en el componente transductor del marco experimental.

Las otras dos entidades que componen el marco formal son el modelo y el simulador. En el contexto de este marco formal, se entiende que un **modelo** de un sistema es *un conjunto de instrucciones, reglas, ecuaciones o ligaduras para reproducir el comportamiento del sistema*. Se trata de una especificación del conocimiento acerca del comportamiento del sistema al Nivel 2 (generación) o al Nivel 3 (estructura).

Dado que un modelo es un conjunto de instrucciones, reglas, ecuaciones o ligaduras, es necesario disponer de un agente capaz de obedecer las instrucciones y reglas, y de evaluar las ecuaciones, con el fin de generar el comportamiento descrito en el modelo. Este agente se denomina **simulador**. Así pues, puede definirse un simulador como *cualquier agente computacional (tal como un único procesador, una red de procesadores, la mente humana, o de manera más abstracta, un algoritmo) capaz de ejecutar el modelo para generar su comportamiento*.

La **relación de modelado**, denominada **validez**, se refiere a la relación entre el modelo, el sistema y el marco experimental. A menudo se piensa en la validez como el grado en el cual el modelo representa fielmente al correspondiente sistema. Sin embargo, resulta más práctico requerir que el modelo capture de forma fiel el comportamiento del sistema sólo hasta el punto demandado por los objetivos del estudio de simulación. De esta forma, el concepto de validez responde a la pregunta de si es posible distinguir entre el modelo y el sistema en el marco experimental de interés.

El tipo más básico de validez, la **validez replicativa**, se afirma si, para todos los posibles experimentos del marco experimental, el comportamiento del modelo y del sistema se corresponden dentro de una tolerancia aceptable. Formas más estrictas de validez son la validez predictiva y la validez estructural. En la **validez predictiva** no sólo requerimos validez replicativa, sino también la habilidad de predecir. Finalmente, para que exista **validez estructural** el modelo no sólo debe ser capaz de replicar los datos observados del sistema, sino también reproducir el comportamiento componente a componente.

La **relación de simulación**, denominada la **corrección** del simulador, es la relación entre el simulador y el modelo. Un simulador es correcto si ejecuta sin

gunda aproximación, el marco experimental está caracterizado por ser un sistema de medida y observador, que típicamente está compuesto por tres componentes (véase la Figura 1.3): generador, receptor y transductor. El **generador** genera las secuencias de entrada al sistema. El **receptor** monitoriza el experimento, para comprobar que se satisfacen las condiciones experimentales requeridas. El **transductor** observa y analiza las secuencias de salida del sistema.

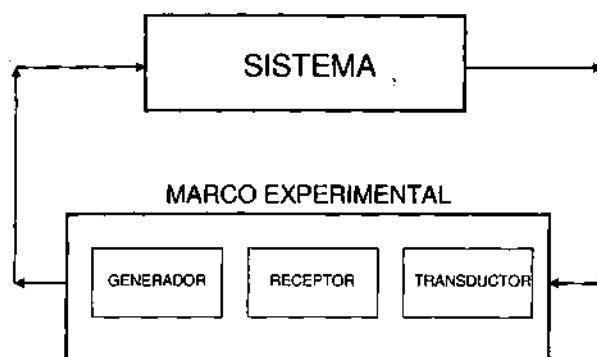


Figura 1.3: Marco experimental y sus componentes.

Es importante establecer lo antes posible en el proceso de desarrollo del modelo cuáles son los objetivos del estudio, ya que los objetivos sirven para enfocar el modelo en aquellos aspectos del sistema que son relevantes para el propósito del estudio. En otras palabras, conocer los objetivos del estudio permite plantear los marcos experimentales adecuados. Los marcos experimentales trasladan los objetivos a condiciones de experimentación más precisas para el sistema fuente y sus modelos.

Una vez fijados los objetivos, presumiblemente existirá un nivel en la especificación del sistema que será el más adecuado para contestar la cuestión planteada. Cuanto más exigentes sean las preguntas, normalmente mayor es la resolución necesaria (nivel en la descripción del modelo) para contestarlas. Por ello, la elección de un nivel de abstracción apropiado repercute en la consecución de los objetivos.

Un procedimiento para transformar los objetivos en marcos experimentales, en aquellos casos en que los objetivos se refieren al diseño del sistema, es el siguiente:

1. Se establece qué medidas van a usarse para evaluar las diferentes alternativas de diseño. Estas medidas deben cuantificar la eficacia con la que el sistema cumple con sus objetivos. Llamaremos a estas medidas las **medidas de salida**.

y en el cual el tiempo de servicio a cada cliente está igualmente preestablecido de antemano. No existe incertidumbre acerca del instante de inicio o de finalización de cada servicio.

Por el contrario, un modelo es **estocástico** cuando *alguna de sus variables de entrada es aleatoria*. Las variables del modelo calculadas a partir de variables aleatorias son también aleatorias. Por ello, la evolución de este tipo de sistemas debe estudiarse en términos probabilísticos.

Por ejemplo, considérese el modelo de un parking, en el cual las entradas y salidas de coches se producen en instantes de tiempo aleatorios. La aleatoriedad de estas variables se propaga a través de la lógica del modelo, de modo que las variables dependientes de ellas también son aleatorias. Este sería el caso, por ejemplo, del tiempo que transcurre entre que un cliente deja aparcado su vehículo y lo recoge (tiempo de aparcamiento), el número de vehículos que hay aparcados en un determinado instante, etc.

Es importante tener en cuenta que realizar una única réplica de la simulación de un modelo estocástico es equivalente a realizar un único experimento sobre el sistema real. Por ejemplo, si se realiza una simulación del comportamiento del parking durante 24 horas, es equivalente a observar el funcionamiento del parking real durante 24 horas. Si se repite la observación al día siguiente, seguramente los resultados obtenidos serán diferentes, y lo mismo sucede con la simulación: si se realiza una segunda réplica independiente de la primera (usando una secuencia diferente de números aleatorios), seguramente los resultados serán diferentes. La consecuencia que debe extraerse de ello es que el diseño y el análisis de los experimentos de simulación estocásticos debe hacerse teniendo en cuenta esta incertidumbre en los resultados. Es decir, debe hacerse empleando técnicas estadísticas.

Modelo estático o dinámico

Un **modelo estático** es un modelo en el cual *el tiempo no juega ningún papel*. Los modelos estáticos pueden ser deterministas y estocásticos. El primer tipo se emplea frecuentemente en Ingeniería para describir el comportamiento de los sistemas en el estado estacionario. El segundo tipo se emplea en la realización de **simulaciones de Monte Carlo**. Reciben este nombre las *simulaciones que emplean números aleatorios para resolver problemas estáticos* (se verá un ejemplo en los ejercicios de autocomprobación).

errores las instrucciones, reglas y ecuaciones que componen el modelo, generando correctamente el comportamiento descrito en el modelo.

La **complejidad** del modelo son los recursos requeridos por un determinado simulador para interpretar el modelo correctamente. Si bien la complejidad se mide relativa a un determinado simulador, a menudo las propiedades intrínsecas al modelo están fuertemente correlacionadas con su complejidad, la cual es prácticamente independientemente del simulador. El modelado exitoso puede verse como *la simplificación válida*. A menudo es necesario reducir la complejidad de los modelos (simplificar los modelos) para facilitar que sean ejecutados en un tiempo razonable en el simulador, de recursos limitados, de que disponemos.

En el proceso de la **simplificación** están implicados dos modelos: el modelo base y el modelo simplificado. El modelo base es “más capaz”, pero requiere más recursos para ser simulado que el modelo simplificado. En este contexto, “más capaz” significa que el modelo base es válido dentro de un conjunto de marcos experimentales (con respecto a un sistema real) más amplio que el modelo simplificado. Sin embargo, el punto importante es que el modelo simplificado y el modelo base sean igualmente válidos dentro del marco experimental particular de interés.

1.2.4. Clasificaciones de los modelos matemáticos

En la Sección 1.2.2 se describió una clasificación en cuatro niveles del conocimiento que puede poseerse de un sistema. Asimismo, se explicó que el conocimiento al Nivel 2 y 3 puede especificarse empleando modelos. En esta sección se muestran varias clasificaciones de los modelos matemáticos. Con ello se pretende proporcionar una primera visión general acerca de qué opciones hay a la hora de seleccionar el modelo matemático que más se adecúe a un determinado estudio. Obsérvese que el tipo de modelo más adecuado en cada caso no depende del sistema en sí, ni del nivel de conocimiento que se posea sobre él, sino de de las preguntas a las que se pretende contestar mediante la experimentación con el modelo.

Modelo determinista o estocástico

Un modelo es **determinista** cuando *todas sus variables de entrada son deterministas*. Es decir, el valor de cada una de ellas es conocido en cada instante. Un ejemplo de modelo determinista es un servicio al cual los clientes acceden ordenadamente, cada uno a una hora preestablecida (de acuerdo, por ejemplo, con un libro de citas),

donde x_i representa el valor de x en el instante t_i , y x_{i-1} y u_{i-1} representan el valor de x y u en el instante t_{i-1} , respectivamente. Despejando x_i se obtiene:

$$x_i = x_{i-1} + \Delta t \cdot f(x_{i-1}, u_{i-1}, t_{i-1}) \quad \text{con } i = 1, 2, \dots \quad (1.3)$$

Esta expresión permite calcular el valor de la variable x en un instante de tiempo t_i , conocido el valor de dicha variable y de la variable de entrada en un instante anterior t_{i-1} . Es decir, permite calcular la evolución en el tiempo de x . La simulación del modelo consiste en el cálculo del valor de la variable x en los instantes:

$$t_i = t_0 + i \cdot \Delta t \quad \text{con } i = 1, 2, \dots \quad (1.4)$$

donde el valor de la variable x en el instante t_i , que se representa x_i , se calcula de la Ec. (1.3). Obsérvese que para ello, es preciso conocer:

- El valor de la variable x en el instante t_0 . Dicho valor, que se denomina *valor inicial de la variable*, se ha representado x_0 .
- El valor de la variable de entrada en los instantes de tiempo t_0, t_1 , etc. Estos valores se han representado u_0, u_1 , etc.

Como resultado de la simulación del modelo, se obtienen los pares de valores (t_i, x_i) para $i = 0, 1, 2, \dots$, que describen la evolución en el tiempo de la variable x . Como la variable x es de tiempo continuo, su evolución suele graficarse interpolando linealmente entre los valores calculados.

1.3. MODELADO Y SIMULACIÓN DE TIEMPO DISCRETO

Los modelos de tiempo discreto son el tipo de modelo más fácil de entender de manera intuitiva, ya que sus variables van cambiando de valor únicamente en instantes de tiempo equiespaciados. Así pues, para simular estos modelos el **reloj de la simulación** (que *indica el valor del tiempo simulado*) avanza saltando un cierto intervalo de tiempo denominado **paso de avance en el tiempo**, que es constante a lo largo de la simulación.

Si el instante inicial de la simulación es t_0 y el paso de avance en el tiempo es Δt , entonces el reloj de la simulación va saltando sucesivamente a los instantes t_0, t_1, t_2, \dots , donde

Un **modelo dinámico** es un modelo *en el cual interviene el tiempo*. Los modelos dinámicos se clasifican en tres tipos: de tiempo discreto, de eventos discretos y de tiempo continuo. Se describen a continuación las características de cada uno de ellos.

En los **modelos de tiempo discreto** y en los **modelos eventos discretos** el valor de las variables sólo puede cambiar en instantes específicos, permaneciendo constante el resto del tiempo. Estos cambios se denominan **eventos** y debe satisfacerse que el número de eventos en un intervalo de tiempo finito debe ser finito. La diferencia entre los modelos de tiempo discreto y de eventos discretos es que en el primer tipo los eventos se producen en instantes de tiempo equiespaciados y predefinidos, cosa que en general no sucede en el segundo. Las variables de ambos tipos de modelos se denominan **variables de tiempo discreto**.

En un **modelo de tiempo continuo** el valor de las variables puede cambiar de manera continua a lo largo del tiempo. Estas variables se denominan **variables de tiempo continuo**. Un ejemplo de variable de tiempo continuo es el nivel de agua en un depósito. Sin embargo, al simular mediante ordenador un modelo de tiempo continuo debe discretizarse el eje temporal. Esta discretización es una aproximación, que transforma el modelo de tiempo continuo en un modelo de tiempo discreto. Se explica esto con más detalle a continuación.

En cualquier intervalo de tiempo de longitud mayor que cero existen infinitos instantes de tiempo. Dado que es imposible calcular el valor de las variables del modelo en infinitos instantes de tiempo, la simulación de los modelos de tiempo continuo se realiza aplicando algoritmos que calculan el valor de las variables del modelo únicamente en determinados instantes de tiempo. Se dice que se realiza una **discretización temporal**.

Consideremos, por ejemplo, el modelo de tiempo continuo

$$\frac{dx}{dt} = f(x, u, t) \quad (1.1)$$

el cual describe que la variación en el tiempo de la variable x es una función f del valor de la propia variable, de una variable de entrada u y del valor del tiempo t . Aplicando el método de integración explícito de Euler con un **intervalo de discretización temporal** Δt , se obtiene el modelo de tiempo discreto

$$\frac{x_i - x_{i-1}}{\Delta t} = f(x_{i-1}, u_{i-1}, t_{i-1}) \quad \text{con } i = 1, 2, \dots \quad (1.2)$$

La primera fila de la tabla indica que cuando el estado actual es 0 y la entrada es 0, entonces la salida del sistema es 0 y el estado en el siguiente instante de tiempo es 0. La segunda fila indica que cuando el estado actual es 0 y la entrada actual es 1, entonces la salida actual es 0 y el estado siguiente es 1, y así sucesivamente. Obsérvese que en el biestable D la salida es igual al estado actual y el estado siguiente es igual a la entrada actual.

Conocido el estado inicial del modelo, la tabla de transición/salidas permite calcular las trayectorias de estado y de salida correspondientes a una trayectoria de entrada. A continuación se muestra un ejemplo del funcionamiento del biestable D. Se comprueba que la salida sigue a la entrada, pero retrasada un paso en el tiempo.

Tiempo	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
Trayectoria de entrada	1	0	1	0	1	0	1	0	1	0
Trayectoria de estado	0	1	0	1	0	1	0	1	0	1
Trayectoria de salida	0	1	0	1	0	1	0	1	0	1

La tabla de transición/salidas es útil para representar el comportamiento de modelos sencillos, con un número pequeño de estados y unas entradas con un número pequeño de posibles valores. Una forma más general de representar el comportamiento de un modelo de tiempo discreto es indicando su función de transición de estado y su función de salida. Sean x_i la entrada, q_i el estado e y_i la salida del modelo en el instante t_i , donde el índice temporal i toma los valores $0, 1, 2, \dots$. Las funciones de transición de estado y de salida se definen de la forma indicada a continuación.

- La **función de transición de estado**, δ , describe cómo se calcula el estado siguiente a partir del estado y la entrada actuales.

$$q_{i+1} = \delta(q_i, x_i) \qquad \text{para } i : 0, 1, 2, \dots \qquad (1.6)$$

- La **función de salida**, λ , describe cómo se calcula la salida actual a partir del estado y la entrada actuales.

$$y_i = \lambda(q_i, x_i) \qquad \text{para } i : 0, 1, 2, \dots \qquad (1.7)$$

Las funciones de transición de estado y de salida permiten calcular la trayectoria del estado (q_1, q_2, \dots) y la trayectoria de la salida (y_0, y_1, y_2, \dots) , conocidos el estado inicial del modelo (q_0) y la trayectoria de entrada (x_0, x_1, x_2, \dots) .

$$t_{i+1} = t_i + \Delta t \quad \text{para } i = 0, 1, 2, \dots \quad (1.5)$$

En cada uno de esos instantes, el modelo se encuentra en un estado, recibe unas entradas y genera unas salidas. El modelo permite calcular, a partir de su estado actual y de sus entradas actuales, cuáles son sus salidas actuales y cuál será su estado en el siguiente instante de tiempo.

Los modelos de tiempo discreto tienen numerosas aplicaciones. Una aplicación importante es la descripción de circuitos digitales síncronos, en los cuales el periodo del reloj del circuito define el paso de avance en el tiempo. Como hemos visto anteriormente, los modelos de tiempo discreto también se emplean como aproximaciones de modelos de tiempo continuo. En este caso, el paso de avance en el tiempo viene determinado por el intervalo de discretización temporal.

1.3.1. Descripción de modelos de tiempo discreto

Cuando el modelo tiene un número finito de estados y sus entradas pueden tomar un número finito de posibles valores, una forma sencilla de especificar el comportamiento del modelo es mediante su **tabla de transición/salidas**. En ella se escriben todas las posibles combinaciones de valores de los estados y las entradas, y para cada una de éstas se indica el valor de la salida y del siguiente estado. Este tipo de representación se emplea frecuentemente para describir el comportamiento de los circuitos digitales. Se muestra a continuación un ejemplo sencillo.

Consideremos el siguiente modelo de un biestable de tipo D. El modelo tiene dos estados, representados mediante 0 y 1, una entrada y una salida. La entrada puede tomar dos posibles valores, representados mediante 0 y 1. Hay cuatro combinaciones entre los estados y los posibles valores de la entrada. Estas cuatro combinaciones se escriben en las dos primeras columnas de la tabla. El número de combinaciones determina el número de filas de la tabla. Para cada una de estas combinaciones, se escribe en la tercera columna el valor del estado siguiente. En la cuarta columna se escribe el valor de la salida actual del sistema.

Estado actual	Entrada actual	Estado siguiente	Salida actual
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

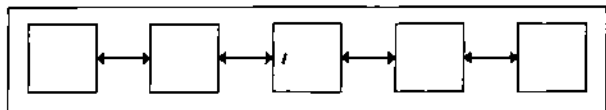


Figura 1.4: Espacio celular unidimensional.

tribución espacial de las células puede ser un mallado unidimensional, bidimensional o multidimensional, conectado de manera uniforme. Las células que influyen sobre una célula en particular, denominadas sus **vecinas**, son a menudo aquellas situadas más cerca en el sentido geométrico.

Puede definirse un autómata unidimensional conectando las células en fila, como se muestra en la Figura 1.4, de modo que cada célula tenga conectada una célula a su izquierda y otra a su derecha. Supongamos que cada célula de este autómata unidimensional puede estar en dos estados, 0 y 1, y que recibe como entrada los estados de las células vecinas. En este caso, hay 8 posibles combinaciones de los valores de las 2 entradas y del estado de la célula. La **tabla de transición de estados** de la célula, que describe el nuevo estado de la célula en función de sus entradas y del estado actual, tendrá la forma mostrada a continuación.

Entrada izquierda actual	Estado actual	Entrada derecha actual	Estado siguiente
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

La cuarta columna contiene interrogaciones. Dado que dicha columna tiene 8 filas y que la célula puede estar en 2 estados, hay $2^8 = 256$ posibles tablas de transición de estados. Cada una de estas posibles tablas representa una **regla**. Hay, por tanto, 256 posibles reglas. El criterio para designar las reglas es el siguiente. La cuarta columna puede interpretarse como un número binario de 8 bits, de modo que la primera fila sea el bit menos significativo y la octava fila el más significativo. Se obtiene el nombre de la regla convirtiendo este número de decimal.

Para simular el autómata hay que asignar valor inicial al estado de cada célula del espacio celular y aplicar el algoritmo simulador, que calcula en cada instante

La información representada mediante una tabla de transición/salidas puede ser descrita de forma más compacta empleando las funciones de transición de estado y de salida. Por ejemplo, el comportamiento del biestable B (la salida es igual al estado actual y el estado siguiente es igual a la entrada actual) puede representarse de la forma siguiente:

$$\delta(q, x) = x \quad (1.8)$$

$$\lambda(q, x) = q \quad (1.9)$$

1.3.2. Simulación de modelos de tiempo discreto

El siguiente algoritmo es un **simulador** para un modelo de tiempo discreto descrito mediante las funciones de transición de estado y de salida. El algoritmo calcula las trayectorias del estado y de salida del modelo, a partir de la trayectoria de entrada y del estado inicial.

```

iFin = 9           índice temporal final
x(0) = 1, ..., x(9) = 0  trayectoria de entrada
q(0) = 0           estado inicial
i = 0              inicialización del índice temporal
while ( i <= iFin ) {
    y(i) = λ( q(i), x(i) )
    q(i+1) = δ( q(i), x(i) )
    i = i + 1
}

```

1.3.3. Autómatas celulares

El **autómata celular** es un tipo de modelo con dos propiedades características. La primera es que posee una estructura regular: está compuesto de componentes iguales, conectados de acuerdo a un cierto patrón espacial. La segunda es que el comportamiento de todos los componentes está regido por el mismo conjunto de reglas.

Este tipo de modelo fue empleado originalmente por von Neumann y Ulam para describir la autorreproducción de sistemas biológicos. De ahí proviene su denominación. Cada uno de los componentes iguales que compone el autómata celular se denomina una **célula** y el conjunto de células se denomina **espacio celular**. La dis-

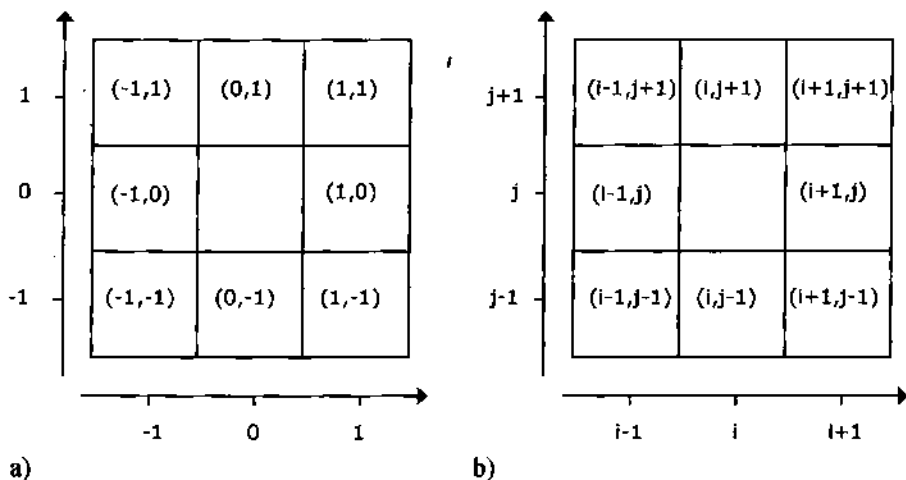


Figura 1.6: Células vecinas en el *Juego de la Vida* a la situada en: a) $(0, 0)$; y b) (i, j) .

el estado de cada célula a partir del estado en el instante anterior de esa célula y de sus vecinas. Por ejemplo, en la Figura 1.5 se muestra la tabla de transición de estados del autómatas con regla 90 y un ejemplo de su simulación. Para facilitar la visualización del resultado de la simulación, el estado 1 se ha representado mediante un aspa y el estado 0 mediante un guión.

Los autómatas celulares unidimensionales muestran comportamientos muy interesantes y diversos. Básicamente pueden distinguirse cuatro tipos diferentes de comportamiento. En algunos autómatas cualquier dinámica se extingue rápidamente. Otros tienen enseguida un comportamiento periódico. Hay otros que muestran un comportamiento caótico. Finalmente, está el tipo más interesante de autómatas: aquellos autómatas cuyo comportamiento no es periódico ni predecible, pero que muestran patrones regulares interesantes.

Un ejemplo interesante de autómatas bidimensional es el denominado **Juego de la Vida de Conway**. El juego tiene lugar en un espacio celular bidimensional, cuyo tamaño puede ser finito o infinito. Cada celda está acoplada a aquellas que se encuentran más próximas a ella, tanto lateral como diagonalmente. Esto significa que, para una célula situada en el punto $(0, 0)$, sus células vecinas laterales están situadas en los puntos $(0, 1)$, $(1, 0)$, $(0, -1)$ y $(-1, 0)$, y sus células vecinas diagonales están situadas en $(1, 1)$, $(-1, 1)$, $(1, -1)$ y $(-1, -1)$, como se muestra en la Figura 1.6a. En la Figura 1.6b se muestran las células vecinas de una célula situada en la posición (i, j) .

El estado de cada célula puede tomar dos valores: 1 (viva) y 0 (muerta). Cada una de las células puede sobrevivir (está viva y permanece viva), nacer (su estado

Entrada izquierda actual	Estado actual	Entrada derecha actual	Estado siguiente
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

i=0	-----X-----
i=1	-----X-X-----
i=2	-----X--X-----
i=3	-----X-X-X-X-----
i=4	-----X--X-----
i=5	-----X-X--X-X-----
i=6	-----X--X--X--X-----
i=7	-----X-X-X-X-X-X-X-----
i=8	-----X--X-----X-----
i=9	-----X-X-----X-X-----
i=10	-----X--X-----X--X-----
i=11	-----X-X-X-X-----X-X-X-X-----
i=12	-----X--X--X--X-----X-----X-----
i=13	-----X-X--X-X--X-X--X-X-----
i=14	-----X--X--X--X--X--X--X--X-----
i=15	-----X-X-X-X-X-X-X-X-X-X-X-X-X-X-----
i=16	-----X-----X-----
i=17	-----X-X-----X-X-----
i=18	-----X--X-----X--X-----
i=19	-----X-X-X-X-----X-X-X-X-----
i=20	-----X--X-----X-----X-----
i=21	-----X-X--X-X-----X-X--X-X-----
i=22	-----X--X--X--X-----X--X--X--X-----
i=23	-----X-X-X-X-X-X-X-----X-X-X-X-X-X-X-----
i=24	-----X-----X-----X-----X-----
i=25	-----X-X-----X-X-----X-X-----X-X-----
i=26	-----X--X-----X--X-----X--X-----X--X-----
i=27	-----X-X-X-X-----X-X-X-X-----X-X-X-X-----
i=28	-----X-----X-----X-----X-----X-----X-----
i=29	-----X-X--X-X--X-X--X-X--X-X--X-X--X-X-----

Figura 1.5: Autómata celular con regla 90: tabla de transición de estados (arriba) y simulación de un espacio celular con esa regla compuesto por 61 células (abajo).

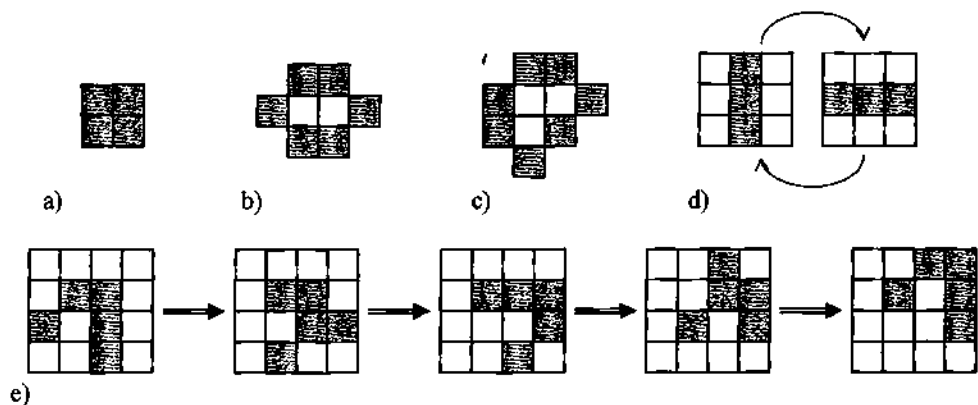


Figura 1.7: Algunos patrones que aparecen en el *Juego de la Vida de Conway*. Los patrones a), b) y c) son estables, ya que no cambian. El patrón d) es oscilante. En e) se muestra un ciclo de patrones que se mueve.

La versión optimizada del algoritmo sigue la lógica siguiente. En una transición de estado deben señalarse aquellas células cuyo estado cambia. A continuación, se establece el conjunto de todas las células vecinas de aquellas. Este conjunto contiene todas las células que pueden posiblemente cambiar su estado en el siguiente paso de tiempo. El estado de las células no pertenecientes a este conjunto permanecerá inalterado en el siguiente paso de tiempo. De esta forma se determina el conjunto de células que posiblemente cambiarán su estado. Para saber si cada una de ellas efectivamente cambia de estado, es necesario evaluar su función de transición de estado.

Los autómatas celulares se emplean para describir sistemas cuyo comportamiento está determinado por la interacción entre sus componentes. Algunas aplicaciones son el estudio del crecimiento de cristales, la propagación de incendios forestales y de manchas en el mar, las reacciones químicas y el transporte en medios fluidos y sólidos, el tráfico de vehículos en las ciudades, el comportamiento de colonias de seres vivos, y el crecimiento de corales y conchas, por citar sólo algunos ejemplos.

1.4. MODELADO Y SIMULACIÓN DE EVENTOS DISCRETOS

Los modelos de tiempo discreto y de eventos discretos tienen una característica en común: las variables del modelo cambian únicamente en determinados instantes de tiempo, manteniéndose constante el resto del tiempo. La diferencia entre ambos tipos de modelo es la siguiente. En los modelos de tiempo discreto estos instantes están equiespaciados en el eje temporal y por ello el reloj de la simulación avanza en

pasa de 0 a 1) y morir (su estado pasa de 1 a 0) a medida que el juego progresa. Las reglas, tal como fueron definidas por Conway, son las siguientes:

1. Una célula permanece viva si tiene en su vecindad 2 ó 3 células vivas.
2. Una célula muere debido a superpoblación si hay más de 3 células vivas en su vecindad.
3. Una célula muere a causa del aislamiento si hay menos de 2 células vivas en su vecindad.
4. Una célula muerta vuelve a la vida si hay exactamente 3 células vivas en su vecindad.

El espacio celular recubre una región finita. Por ejemplo, el cuadrado ocupado por $N \times N$ células, de modo que el autómata está compuesto por N^2 células. Obsérvese que debe especificarse el comportamiento de las células del borde del espacio, ya que no poseen todos sus vecinos. Una posibilidad es asumir que las células situadas en los bordes de la región mantienen un estado constante: por ejemplo, todas están muertas. Otra solución es envolver el espacio de manera toroidal. Es decir, interpretar el índice N también como 0.

El modelo del Juego de la Vida evoluciona sobre una base de tiempo discreto. El tiempo avanza a pasos: 0, 1, 2, ... El juego comienza con cierta configuración de células vivas. A medida que la simulación va avanzando, la forma de los grupos de células vivas va cambiando. La idea del juego es encontrar nuevos patrones y estudiar su comportamiento. Por ejemplo, se muestran algunos patrones en la Figura 1.7.

Un algoritmo para simular el autómata sería el siguiente. Se almacena el estado inicial de las células en una estructura de datos. En cada instante de tiempo se inspeccionan todas las células, aplicando la función de transición de estado a cada una de ellas, y salvando el siguiente estado en una segunda copia de la estructura de datos. Una vez calculado el siguiente estado, éste se convierte en el estado actual y se avanza el reloj de la simulación un paso.

Este algoritmo que calcula el nuevo estado de todas las células en cada paso de tiempo. Una forma de optimizarlo consiste en calcular únicamente el nuevo estado de aquellas células cuyo nuevo estado pueda ser diferente del actual. Si el estado de las células vecinas no cambia en el instante actual, entonces el estado de la célula en cuestión no cambiará en el siguiente instante. Obsérvese que en esta versión optimizada del algoritmo es necesario llevar un registro de qué células son susceptibles de cambiar de estado en la transición siguiente.

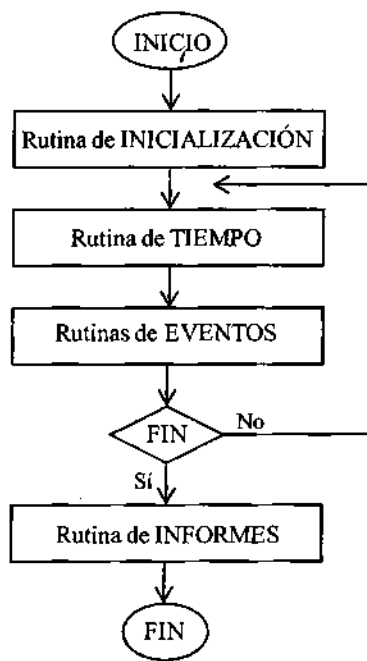


Figura 1.8: Flujo de la simulación de un modelo orientado a la planificación de eventos.

- **Rutina de informes.** Al finalizar la simulación, calcula y muestra el valor de las variables de salida.

El programa principal controla el flujo de control de la simulación, tal como se muestra en la Figura 1.8.

1. Comienza la simulación.
2. El programa principal pasa el control a la **rutina de inicialización**, para que ésta inicialice el reloj de la simulación, las variables de estado, el calendario de eventos y los acumuladores estadísticos. Se activa el evento “Inicio de la Simulación”. Como parte de las acciones asociadas a la ejecución de este evento, se planifican determinados eventos para su ejecución en instantes futuros. Estos eventos son añadidos al calendario de eventos, ordenados de menor a mayor instante de ejecución.
3. Una vez ejecutadas las acciones de inicialización, el programa principal transfiere el control a la **rutina de tiempo**. El reloj de la simulación es avanzado hasta el instante de ejecución del primer evento del calendario, que es el más inminente, el cual es entonces borrado del calendario. A continuación, se activa dicho evento.

pasos de tiempo de duración constante durante toda la simulación. En los modelos de eventos discretos no existe esta restricción. El tiempo que transcurre entre eventos consecutivos puede ser cualquiera, siempre que se satisfaga que el número de eventos en cualquier intervalo finito de tiempo sea finito. Por ello, el algoritmo de simulación de eventos discretos debe mantener un **calendario de eventos**, donde ir registrando el instante de activación de los eventos planificados para instantes futuros.

Existen varias metodologías para la descripción de modelos de eventos discretos. En esta sección se introducirán dos de ellas: el modelado orientado a la planificación de eventos y el modelado orientado a los procesos.

1.4.1. Modelado orientado a la planificación de eventos

Una manera de definir los modelos de eventos discretos es especificando qué tipos de eventos pueden ocurrir y cuál es la relación causal entre ellos. Un evento puede producir un cambio en el estado del modelo, planificar eventos para instantes futuros y cancelar la planificación de eventos.

A continuación se describe una manera sencilla de programar, empleando un lenguaje de programación imperativo (C, FORTRAN, etc.), el algoritmo de la simulación de un modelo de eventos discretos descrito mediante planificación de eventos. El programa podría constar de las rutinas siguientes:

- **Rutina de inicialización.** Asigna valores iniciales a las variables de estado, inicializa los acumuladores estadísticos y planifica eventos añadiéndolos al calendario de eventos. Los **acumuladores estadísticos** son variables intermedias a partir de las cuales se calculan las variables de salida de la simulación. Algunos ejemplos de acumuladores estadísticos son la suma de tiempos de espera en una cola, el tiempo máximo hasta el momento de espera en una cola, el número de entidades procesadas hasta el momento por cierto recurso, etc.
- **Rutina de tiempo.** Determina cuál es el evento más inminente de los planificados en el calendario de eventos y avanza el reloj de la simulación hasta ese instante.
- **Rutinas de eventos.** Son las rutinas, una para cada tipo de evento, que realizan el flujo de acciones asociado al evento. Entre estas acciones puede estar modificar el valor de las variables y acumuladores estadísticos, así como añadir o quitar eventos del calendario de eventos.

en cualquier contexto de aplicación en el cual su comportamiento sea apropiado y tenga sentido conectarlos a otros componentes.

Para describir modelos de manera modular y jerárquica es preciso definir los componentes que componen el modelo y especificar cómo los componentes interactúan entre sí. La interacción entre los componentes del modelo puede describirse mediante los dos tipos siguientes de acoplamiento:

- **Acoplamiento modular.** Cada componente tiene una interfaz, compuesta por puertos de entrada y puertos de salida, a través de los cuales se produce la interacción con el resto de componentes. El componente envía información a través de sus puertos de salida y la recibe a través de sus puertos de entrada. La interacción (transmisión de información) entre componentes es descrita mediante la conexión de los puertos de salida a los correspondientes puertos de entrada. La transmisión de información entre puertos se produce instantáneamente. Es decir, en el mismo instante en que se envía información por un puerto de salida, ésta es recibida en los puertos de entrada conectados al de salida.

Los modelos compuestos, definidos como componentes conectados entre sí, también tienen su interfaz, con puertos de entrada y de salida. La transmisión de información desde las entradas del modelo compuesto hasta las entradas de sus componentes y desde las salidas de los componentes hasta las salidas del modelo compuesto también se produce instantáneamente.

- **Acoplamiento no modular.** La interacción entre los componentes se describe mediante la influencia que tiene el estado de unos componentes (componentes influenciadores) sobre la transición del estado de otros (componentes influenciados). Los componentes con acoplamiento no modular no tienen interfaz, ya que el estado de los componentes influenciadores interviene directamente en las funciones de transición de estados de los componentes influenciados. Por ejemplo, las células de los autómatas celulares están acopladas de manera no modular.

En el Tema 2 se describirá un formalismo, denominado DEVS (Discrete Event system Specification), que está basado en la planificación de eventos y facilita el modelado modular y jerárquico.

4. El programa principal transfiere el control a la rutina correspondiente al tipo de evento activado. Las acciones realizadas por la **rutina de eventos** dependen de cada caso, pero en general la rutina actualizará las variables de estado, los acumuladores estadísticos y añadirá nuevos eventos al calendario de eventos. También puede eliminar eventos del calendario de eventos.

Existe un evento especial, denominado “Finalización de la Simulación”, que se activa cuando se satisfacen las condiciones fijadas para la finalización de la simulación. Por ejemplo, que se alcance determinado valor del tiempo, que el estado del sistema satisfaga determinadas condiciones, etc. Una de las acciones asociadas al evento de finalización de la simulación es el cálculo de las variables de salida de la simulación a partir del valor de los acumuladores estadísticos.

5. Si el evento ejecutado es “Finalización de la Simulación”, el programa principal transfiere el control a la rutina generadora de informes. En caso contrario, el programa principal devuelve el control a la rutina de tiempo.
6. La **rutina generadora de informes** muestra el valor de las variables de salida de la simulación.
7. Finaliza la simulación.

Esta forma de describir los modelos de eventos discretos y de programar su algoritmo de simulación es conceptualmente sencilla cuando se aplica a modelos sencillos. Sin embargo, este procedimiento implica describir globalmente el comportamiento del sistema completo, lo cual resulta excesivamente complejo y es propenso a errores cuando se aplica a la realización de modelos de grandes dimensiones, en los que hay que definir muchos tipos de eventos y la relación causal entre ellos.

Para facilitar la descripción y simulación de modelos de grandes dimensiones, se han propuesto formalismos orientados a la planificación de eventos que permiten describir los modelos de manera **modular y jerárquica**, como componentes que interactúan entre sí, de modo que el modelo pueda reproducir la estructura del sistema.

El planteamiento es el siguiente. En lugar de intentar describir globalmente el comportamiento del sistema completo, es más sencillo realizar un **análisis por reducción**. Es decir, dividir el sistema en partes, modelar las partes independientemente y finalmente describir la interacción entre las partes. Esta metodología de modelado modular y jerárquico facilita desarrollar y probar los modelos de las partes como unidades independientes, situarlos en un repositorio de modelos y reutilizarlos

La metodología del modelado orientado a los procesos se explicará con más detalle en el Tema 3, en el cual se realiza una introducción al manejo de Arena.

1.4.3. Modelado de una oficina de atención al público

El modelo descrito a continuación permite ilustrar las diferencias entre el modelado orientado a la planificación de eventos y el modelado orientado a los procesos. Se trata del modelo de una oficina de atención al público en la cual trabaja un empleado. La estructura lógica del modelo es la siguiente (véase la Figura 1.9):

- Si llega un nuevo cliente y el empleado está ocupado, el cliente se pone al final de la cola. Si el empleado está libre, el cliente es atendido inmediatamente. La cola que se forma frente al empleado tiene disciplina FIFO. El término FIFO es un acrónimo del inglés “First In, First Out”, que significa “primero en entrar, primero en salir”. Es decir, los clientes abandonan la cola en el mismo orden en que llegan a ella.
- Si el empleado termina de atender a un cliente, éste se marcha y comienza a ser atendido el primer cliente de la cola. Si la cola está vacía, el empleado permanece libre hasta la llegada de un nuevo cliente.

En la parte superior de la Figura 1.10 se muestra un esquema de la estructura del modelo, que está compuesto por una cola y un proceso con un recurso: el empleado. Se muestran los eventos de entrada y de salida al sistema. Los eventos de entrada, ordenados sobre el eje temporal, se denominan **trayectoria de entrada**, mientras que los eventos de salida ordenados sobre el eje temporal se denominan **trayectoria de salida**. Cada evento de entrada (llegada de un cliente a la oficina) y de salida (un cliente que abandona la oficina después de ser atendido) se representa mediante una flecha vertical situada sobre el eje temporal, en el instante en que se produce el evento.

Los procesos de llegada y de atención al cliente se modelan de la forma siguiente. Supongamos que los clientes llegan de uno en uno y que el intervalo de tiempo que transcurre entre llegadas sucesivas está distribuido exponencialmente, con media 10 minutos. Supongamos también que el tiempo que tarda el empleado en atender a un cliente es una variable aleatoria distribuida uniformemente en el intervalo $[5, 10]$ minutos.

Para describir el estado del sistema, hacen falta al menos dos variables de estado: la fase del empleado y el número de clientes que están en cola. Los valores posibles

1.4.2. Modelado orientado a los procesos

La metodología del modelado orientado a los procesos pretende facilitar la descripción de los modelos, permitiendo que ésta se realice de manera más próxima al razonamiento humano. Consiste en tomar el punto de vista de las entidades y describir su circulación a través del sistema. Este enfoque se centra en los procesos que llevan a cabo las entidades. Por ello se llama modelado orientado a los procesos.

La práctica de esta metodología de la orientación de los procesos comenzó a aplicarse en la década de 1970, gracias a la aparición de **lenguajes de simulación** de propósito general para modelos de eventos discretos. Los compiladores de estos lenguajes de simulación traducen automáticamente la descripción orientada a los procesos del modelo a una descripción orientada a la planificación de eventos escrita en algún lenguaje de programación. En última instancia, el código ejecutable de la simulación siempre está orientado a la planificación de los eventos. Algunos de estos lenguajes de simulación se usan todavía en la actualidad, como es el caso de GPSS, SIMSCRIPT, SLAM, SIMAN, etc.

Hoy en día el modelado orientado a los procesos suele realizarse empleando **entornos de simulación**, que son una capa software construida sobre un lenguaje de simulación a fin de facilitar la descripción del modelo mediante interfaces de usuario muy intuitivas, con menús, diálogos, etc. Entre los entornos de simulación más populares están AnyLogic, Arena, AutoMod, Enterprise Dynamics, ExtendSim, Flexsim, ProModel y SIMUL8.

Estos entornos permiten al usuario construir el modelo instanciando módulos predefinidos (pinchando y arrastrando el icono desde la librería de módulos a la ventana de edición del modelo) y conectándolos gráficamente. La animación y otras capacidades gráficas permiten visualizar la evolución del modelo durante la simulación. La interfaz gráfica de usuario del entorno de simulación también permite acceder a los niveles inferiores en la descripción del modelo: a la descripción de partes del modelo usando el lenguaje de simulación e incluso al lenguaje de programación.

Por ejemplo, el entorno de simulación Arena soporta el uso combinado de diferentes niveles de descripción: elementos de alto nivel parametrizables por el usuario y elementos de bajo nivel que el usuario puede definir usando el lenguaje de simulación SIMAN, y los lenguajes de programación Visual Basic y C/C++.

Asimismo, algunos entornos proporcionan facilidades para el modelado estadístico de las entradas aleatorias al modelo, para definir experimentos y problemas de optimización sobre el modelo, y para analizar estadísticamente los resultados.

de estas variables son los siguientes. La fase del empleado puede tomar dos posibles valores: {libre, ocupado}. El número de clientes que esperan en la cola puede tomar valores enteros positivos, incluido el cero. En la Figura 1.10 se muestra un ejemplo de cómo evoluciona el estado del modelo a medida que los clientes van llegando y abandonando la oficina.

El objetivo de la simulación es estimar el tiempo medio de espera en cola y el número medio de clientes que esperan en la cola. Los cálculos a realizar son los siguientes.

- El **tiempo medio de espera del cliente en la cola**, se define de la forma siguiente:

$$\hat{d}(n) = \frac{1}{n} \sum_{i=1}^n D_i \quad (1.10)$$

donde n es el número total del clientes que han abandonado la cola y D_i es el tiempo de espera en la cola del cliente i . Para calcularlo es preciso llevar registro a lo largo de la simulación tanto del número de clientes que han abandonado la cola hasta ese momento, como de la suma de sus tiempos de espera.

- El **número medio de clientes en la cola** se define de la forma siguiente:

$$\hat{q}(T) = \frac{1}{T} \int_0^T Q(\tau) \cdot d\tau \quad (1.11)$$

donde T es el tiempo que ha durado la simulación y $Q(\tau)$ es el número de clientes que había en la cola en el instante τ . Obsérvese que si se representa el número de clientes en cola Q en función del tiempo, la integral es igual al área bajo la curva entre los instantes inicial y final de la simulación. Dicho área debe irse calculando a lo largo de la simulación, para lo cual se emplea el acumulador estadístico R . Inicialmente R vale cero. Se actualiza el valor de R cada vez que cambia el tamaño de la cola y en el instante final la simulación. Para ello, se suma al valor actual de R el producto del número de clientes en cola antes del cambio por el tiempo transcurrido desde el anterior cambio. Obsérvese que es preciso definir un segundo acumulador estadístico, t_{evento} , que almacene el instante en que se produjo el cambio en el tamaño de la cola anterior al actual.

La descripción del modelo siguiendo la metodología de la **planificación de eventos** se realiza describiendo la condición de disparo y las acciones asociadas a

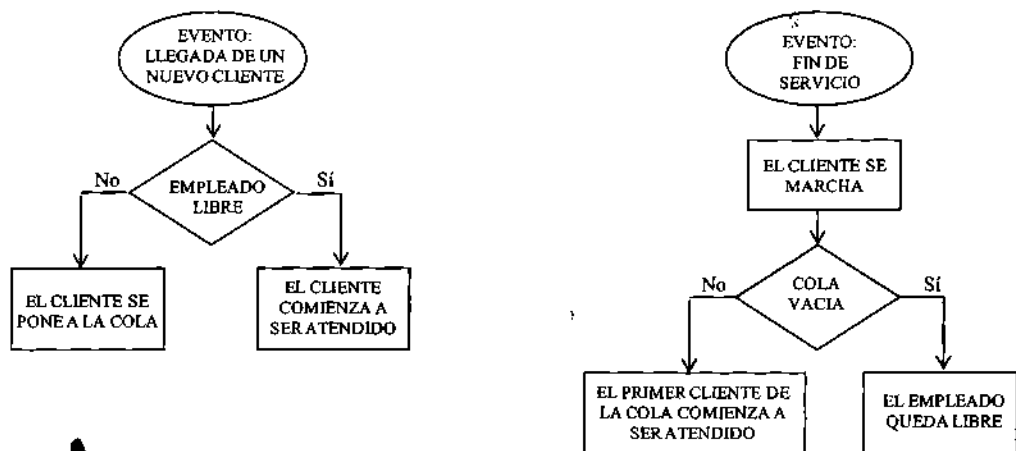


Figura 1.9: Esquema de la estructura lógica del modelo.

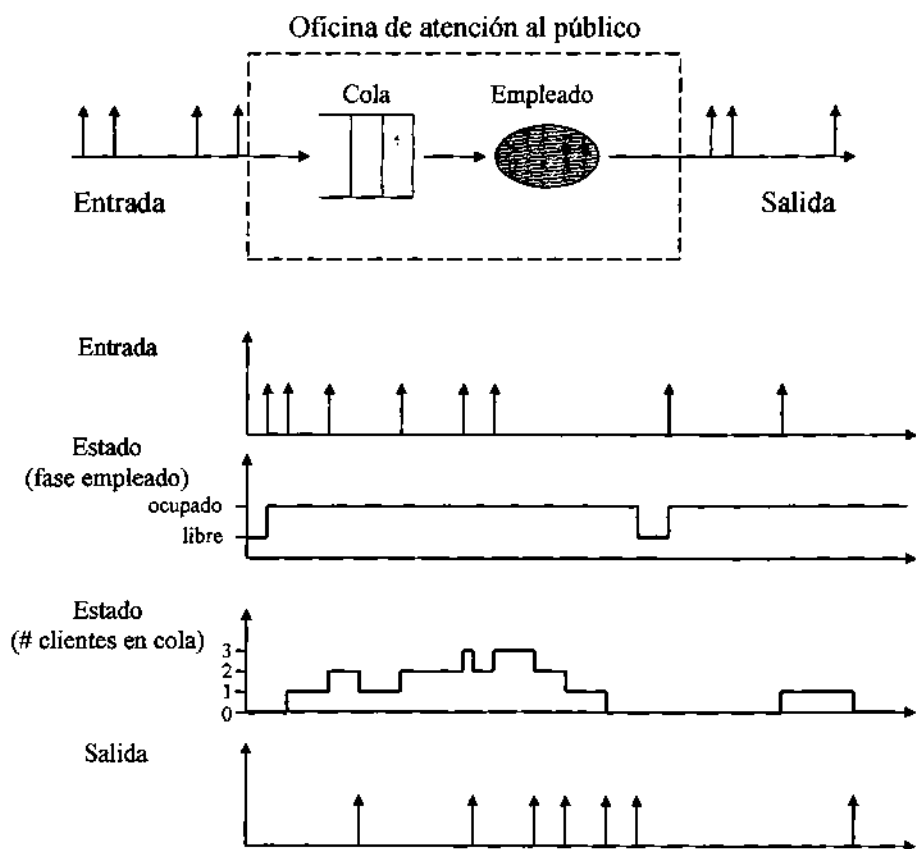


Figura 1.10: Ejemplo del comportamiento del modelo de la oficina.

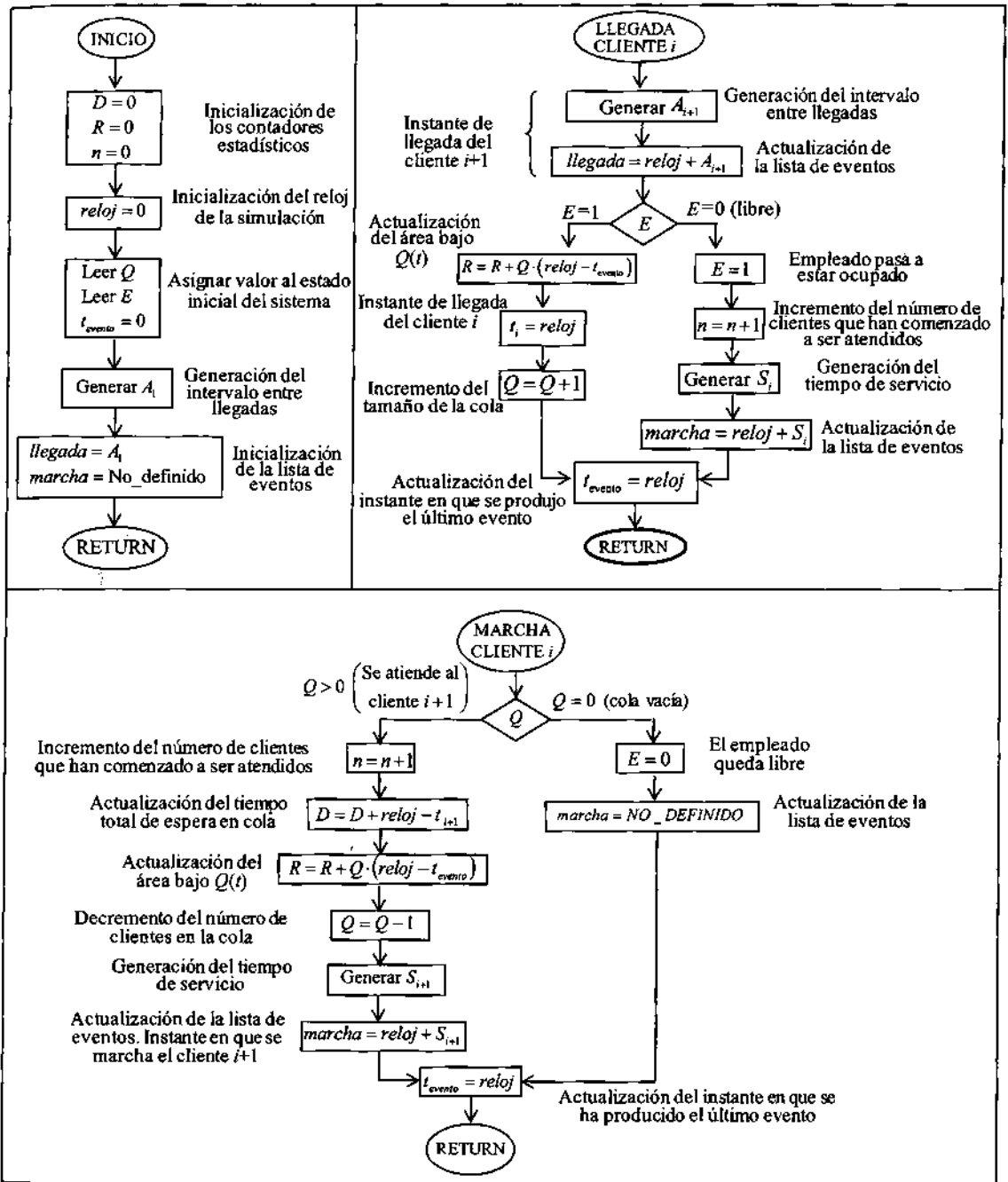


Figura 1.11: Flujos de acciones asociadas a los eventos.

cada tipo de evento. En este caso hay cuatro tipos de eventos: inicio de la simulación, llegada a la oficina de un nuevo cliente, fin de servicio a un cliente y final de la simulación. Los flujos de acciones asociados a los tres primeros tipos se muestran en la Figura 1.11.

Obsérvese que la fase del empleado se representa mediante la variable de estado E , que puede tomar los valores 0 (libre) y 1 (ocupado). El modelo tiene otras dos variables de estado además de E . Estas son: Q (número de clientes en cola) y t_{evento} (instante en que se produjo el último evento).

El calendario de eventos guarda dos valores: *llegada* (instante en que está planificada la llegada del próximo cliente) y *marcha* (instante en que está planificado que el cliente que está siendo atendido abandone el sistema).

El modelo tiene tres contadores estadísticos: D (suma de los tiempos de espera en cola de todos los clientes que la han abandonado), n (número de clientes que han comenzado a ser atendidos) y R (área bajo Q).

Veamos a continuación cómo se describe el funcionamiento de la oficina empleando la metodología de la **orientación al proceso** y en particular el entorno de simulación Arena. En esta metodología se toma el punto de vista de los clientes y se describe su circulación a través del sistema. Los pasos que sigue un cliente en la oficina son los siguientes:

1. Llego a la oficina.
2. Me pongo al final de la cola. Espero hasta que yo sea el primero de la cola y el empleado esté libre (si tengo suerte, el tiempo de espera será cero).
3. El empleado me atiende durante el tiempo que requiero.
4. Finaliza mi tiempo de servicio y abandono la oficina.

El modelo puede describirse de manera sencilla, ya que Arena dispone de un módulo predefinido que permite describir el proceso de llegada, otro módulo que permite describir el proceso de atención al cliente con su cola, y otro módulo que permite describir que los clientes abandonan la oficina. En la Figura 1.12a se muestra el diagrama de módulos del modelo con la animación correspondiente a un cierto instante de la ejecución de la simulación. El diagrama del modelo está compuesto por la instanciación y conexión de los tres tipos de módulo anteriormente indicados. La conexión entre los módulos define el flujo de los clientes en la oficina. En el preciso instante de la simulación mostrado en la figura habían llegado ya 109 clientes, 3

esperaban en la cola, uno estaba siendo atendido por el empleado y 105 había abandonado la oficina (véanse los números que hay cerca de los módulos en la Figura 1.12a).

Haciendo doble clic sobre cada módulo del diagrama se abre un menú de configuración. En la Figura 1.12b se muestra el menú de configuración del módulo que describe la llegada de clientes (módulo de tipo Create) y en la Figura 1.12c el menú del módulo que describe el proceso de atención al cliente (módulo de tipo Process).

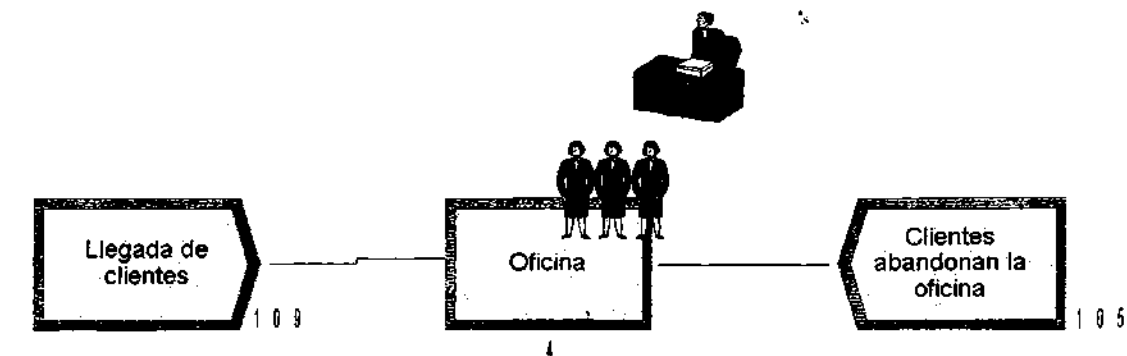
Arena calcula por defecto determinadas medidas estadísticas del comportamiento del sistema, como son el tiempo medio, máximo y mínimo en las colas, la ocupación de los recursos, etc. y permite al usuario definir sus propios cálculos. Al finalizar el experimento de simulación Arena genera automáticamente informes en los cuales se muestra esta información.

En este punto no es importante entender los detalles acerca de la definición del modelo con Arena, ya que eso se explicará con detalle en el Tema 3. Lo importante es comprender la diferencia entre el modelado orientado a la planificación de eventos y el modelado orientado a los procesos.

1.5. PASOS EN UN ESTUDIO DE SIMULACIÓN

Avanzando en la breve introducción al modelado y la simulación ofrecida en este primer tema, en esta sección se describen los pasos que típicamente se siguen al realizar un estudio de simulación empleando modelos de eventos discretos de tipo estocástico. A grandes rasgos, los pasos son los siguientes:

1. **Definición del problema.** Consiste en definir claramente por qué se estudia el sistema, qué objetivos se persigue con ello y cuáles son las preguntas a contestar.
2. **Planificación del proyecto.** Tiene como objetivo estimar qué recursos son necesarios para llevar a cabo el estudio: dinero, personal, hardware de computación, recursos software, etc. Si estos recursos no están disponibles debe replantearse el alcance del estudio.
3. **Definición del sistema.** Debe definirse qué aspectos de la realidad constituyen el sistema bajo estudio.
4. **Formulación conceptual del modelo.** Desarrollo de un modelo preliminar, bien gráficamente (mediante diagramas de bloques) o en pseudocódigo, en el



a)

Configuración de la llegada de clientes:

Name:			Entity Type:
Llegada de clientes			Cliente
Time Between Arrivals:			
Type:	Expression:	Units:	
Expression	EXPON(10)	Minutes	
Entities per Arrivat	Max Arrivals	First Creation	
1	Infinite	0.0	

Botones: OK, Cancel, Help

b)

Configuración del proceso de atención:

Name:		Type:
Oficina		Standard
Logic:		
Action:	Priority:	
Seize Delay Release	Medium(2)	
Resources:		
Resource: resource 1	Add	
<End of list>	Edit	
	Delete	
Delay Type:		
Uniform	Units:	Allocation:
	Minutes	Value Added
Minimum:	Maximum:	
5	10	
<input checked="" type="checkbox"/> Report Statistics		

Botones: OK, Cancel, Help

c)

Figura 1.12: Modelado en Arena de la oficina atendida por un empleado: a) diagrama de módulos del modelo; y b) y c) propiedades de los módulos que definen los procesos de llegada de los clientes y de atención a los clientes, respectivamente.

del esfuerzo y del tiempo del proyecto en los pasos 1 a 6 (comprensión del problema, diseño del modelo y del experimento), el 20 % del esfuerzo en el paso 7 (programación del modelo) y el 40 % restante en los pasos 8 a 12 (verificación, validación, diseño y realización del experimento, análisis de los resultados y documentación). A continuación se describe con más detalle cada uno de estos pasos.

1.5.1. Definición del problema y planificación del proyecto

Antes de intentar resolver un problema, es esencial formularlo adecuadamente. Ésta es una afirmación que parece evidente, de sentido común, pero que no siempre se aplica. Para encontrar una solución a un problema, primero es preciso conocer cuál es el problema. En particular, el primer paso en todo estudio de simulación es definir explícitamente cuál es su objetivo: la construcción del modelo se realiza con el fin de ayudar a responder ciertas preguntas acerca del sistema real.

El objetivo de un estudio de simulación puede ser de lo más diverso. Algunos ejemplos son:

- **Evaluación** del comportamiento de determinado diseño del sistema en términos absolutos frente a determinados criterios específicos.
- **Comparación** del comportamiento de diferentes diseños alternativos del sistema, o comparación de diferentes políticas de operación o procedimientos.
- **Predicción** del comportamiento del sistema bajo cierto conjunto de condiciones.
- **Análisis de la sensibilidad**, es decir, determinar entre varios factores cuál es el que más afecta al comportamiento del sistema.
- **Optimización**, esto es, cálculo de los valores de determinados parámetros del modelo que producen que el sistema tenga la mejor respuesta (atendiendo a un determinado criterio).
- **Estudio de las relaciones funcionales** entre determinadas variables del modelo. Por ejemplo, la determinación de la relación funcional entre una de las respuestas del modelo y determinadas variables de entrada.
- **Localización y análisis de los cuellos de botella**. Consiste en descubrir la localización de los cuellos de botella del sistema y evaluar propuestas para descongestionar alguno de ellos.

que se definen los componentes, las variables descriptivas y las interacciones (la lógica) que constituyen el modelo del sistema.

5. **Diseño preliminar del experimento.** Consiste en definir qué acción se va a ejercer sobre el modelo y cómo se va a medir su comportamiento. Se trata, por tanto, de definir qué variables son las entradas y cuáles las salidas, cómo va a modificarse el valor de las entradas y cómo van a recogerse los datos de salida.
6. **Preparación de los datos de entrada.** Debe establecerse el procedimiento a seguir para asignar valores a cada una de las variables de entrada durante la simulación.
7. **Traducción del modelo a código ejecutable**, bien sea programando directamente el algoritmo de simulación del modelo en algún lenguaje de programación, o bien empleando un lenguaje o entorno de simulación. Esta descripción del modelo ya puede ser simulada en el ordenador.
8. **Verificación y validación.** Se trata de estudiar si el modelo opera como debiera, y si la salida del modelo es creíble y representativa del comportamiento del sistema.
9. **Diseño experimental final.** En este punto se diseña un experimento que proporcione la información necesaria para poder contestar a las preguntas planteadas en el estudio. Típicamente el experimento consiste en cierto número de réplicas de la simulación realizadas en condiciones bien definidas.
10. **Experimentación.** Realización del experimento de simulación diseñado anteriormente.
11. **Análisis e interpretación.** Consiste en extraer conclusiones de los resultados de la simulación.
12. **Documentación y actuación (si procede).** Una vez obtenidos los resultados, analizados y extraídas las conclusiones, éstas deben ser presentadas de la forma más adecuada. El modelo debe ser documentado y preparado de modo que pueda ser reutilizado para posteriores estudios. Finalmente, deben tomarse acciones (si ésta era la finalidad del estudio) a partir de las conclusiones obtenidas.

La denominada “Regla 40 – 20 – 40” proporciona una orientación acerca de cómo distribuir el esfuerzo en un estudio de simulación. Esta regla aconseja dedicar el 40 %

- ¿Dispone el estudio del soporte necesario por parte de la dirección? Las partes involucradas se implicarán en mayor medida si son conscientes del respaldo de la dirección. ¿Se les ha informado adecuadamente? ¿Son conscientes de que ese respaldo existe?
- ¿Se han definido las tareas de que consta el proyecto, la vinculación entre ellas, quién debe realizarlas y en qué fechas?
- ¿Posee cada uno de los miembros del proyecto la formación necesaria para realizar con éxito su cometido? ¿Está suficientemente capacitado el coordinador del proyecto?
- ¿Se han identificado los usuarios potenciales o clientes de los resultados del estudio? ¿Han sido todos ellos consultados? ¿Se ha establecido el procedimiento para comunicarles regularmente el estado en que se encuentra el proyecto y los progresos realizados?
- ¿Se han establecido los adecuados canales de comunicación entre los miembros del equipo, la dirección y los clientes?

1.5.2. Definición del sistema y formulación del modelo

La esencia del arte del modelado son la **abstracción** y la **simplificación**. Se trata de identificar el pequeño conjunto de características o propiedades del sistema suficientes para servir los objetivos específicos del estudio. A grandes rasgos, la metodología para la construcción del modelo podría ser la siguiente:

1. Escoger las variables de salida, lo cual resulta relativamente sencillo una vez definido el objetivo del estudio.
2. La construcción del modelo se realiza identificando qué componentes del sistema afectan a estas variables de salida y decidiendo, para cada uno de ellos, si debe ser incluido en el modelo o si debe ser considerado parte del entorno del modelo. En este último caso, el componente es representado mediante entradas al modelo.
3. Una vez se ha decidido qué componentes constituyen de modelo y qué variables describen el comportamiento de cada componente, deben determinarse las relaciones funcionales entre los componentes, es decir, la lógica del modelo. Normalmente se usan gráficos o pseudocódigo para describir el sistema como un diagrama de flujo lógico.

Estos objetivos pretenden responder preguntas del tipo:

- ¿Funciona el sistema del modo que queremos que lo haga? (Evaluación y predicción).
- ¿Por qué no funciona el sistema y qué podemos hacer para que funcione? (Determinación de los cuellos de botella y optimización).
- ¿Cuál es la mejor alternativa? (Comparación y análisis de sensibilidad).

El propósito del estudio de simulación tiene implicaciones decisivas en el diseño del modelo y del experimento. Por ejemplo, si el propósito es evaluar el comportamiento de un sistema en términos absolutos, deberá existir un alto grado de correspondencia entre el comportamiento del modelo y del sistema. Por el contrario, si el propósito del estudio es comparar varios diseños, el modelo puede ser válido en un sentido relativo incluso cuando sus respuestas en un sentido absoluto difieran considerablemente de las del sistema real. En consecuencia, el proceso completo de diseño del modelo, validación, diseño de los experimentos y análisis de los resultados debe estar íntimamente relacionado con el propósito del estudio.

No obstante, el objetivo del estudio debe definirse con más detalle que mediante una simple pregunta. También es necesario conocer todos los aspectos relevantes que rodean al proyecto de simulación. Frecuentemente el estudio es realizado por un grupo de ingenieros que deben recomendar una solución al problema, pero que no son quienes deben tomar la decisión final. En estos casos, los resultados del estudio deben servir simultáneamente a dos propósitos: ayudar a formular la recomendación y ayudar a “vender” la recomendación. Con este fin, deben considerarse otros factores condicionantes: cuáles son los intereses de la persona que debe tomar la decisión final, qué aspectos de la situación están bajo su control y cuáles no, en qué va a consistir el proceso de la toma de decisión, qué otras personas participan en la toma de la decisión (en especial aquellas que se oponen a que se realicen cambios en el sistema) y cuáles son sus objetivos más relevantes, etc.

Como parte de la planificación deben considerarse todos aquellos factores que son críticos para el éxito del proyecto, tales como:

- ¿Están claramente definidos los objetivos del estudio? ¿Por qué se realiza el estudio? ¿Qué se pretende obtener del estudio?
- ¿Se dispone de los recursos adecuados y suficientes para poder realizar el estudio: tiempo, personal, software, ordenadores, etc.?

El segundo ejemplo de estudio de simulación fallido tuvo lugar en el año 1985. Se realizó un modelo de una fábrica de grandes dimensiones (la fábrica tenía entorno a 2.5 km de longitud), con un coste de 250.000 dólares. Sin embargo, el modelo era tan detallado que jamás pudo llegarse a ejecutar, dados los inmensos recursos de memoria que requería.

Una buena práctica consiste en realizar el modelo de manera iterativa: comenzar con un modelo muy simple, cuya complejidad puede posteriormente ir aumentándose fácilmente. Para ello, el modelo debe realizarse de manera **modular** y **jerárquica**, dividiendo el sistema en submodelos y modelando todos ellos con un nivel semejante de complejidad. Este modelo inicial puede construirse muy rápido y puede servir de punto de discusión sobre posteriores refinamientos en el modelado, entendiéndose por **refinamiento** del modelo *el aumento en su nivel de detalle*. Añadiendo progresivamente los refinamientos al modelo, y comparando los resultados obtenidos con los del modelo más sencillo, puede estimarse el impacto de cada conjunto de refinamientos sobre la respuesta del modelo. En determinado punto de este proceso de aumento gradual de la complejidad del modelo, los refinamientos añadidos tienen un efecto pequeño, es decir, influyen despreciablemente en las conclusiones del estudio, con lo cual se concluirá que no es preciso incorporarlos.

Existen varias técnicas que se pueden aplicar simultáneamente para obtener un modelo simplificado de un sistema. Algunas de ellas son:

- La **omisión** de ciertos detalles del modelo. La idea que hay detrás de esta aproximación es que no todos los factores son igualmente importantes para determinar el comportamiento del sistema. Se trata de determinar qué factores son críticos (efectos de primer orden) y cuáles no (efectos de segundo orden).

Omitir los detalles supone introducir deliberadamente “imperfecciones” y por tanto podría considerarse que es una mala práctica. Sin embargo, nada más lejos de la realidad: solamente omitiendo los factores de segundo orden puede construirse un modelo útil. El aumento en el nivel de detalle debe hacerse atendiendo al impacto que la complejidad añadida tiene sobre los resultados del estudio. Es decir, debe emplearse el esfuerzo de modelado en la representación de aquellos aspectos del sistema que más influencia tienen sobre los resultados del estudio.

- **Agregar** determinados procesos del sistema en un proceso equivalente. Ejemplos típicos de agregación son:

- **Agregación temporal**: se trata un intervalo de tiempo como una unidad. Por ejemplo, se consideran que todos los eventos ocurridos en un día

Para diseñar el modelo de simulación conviene disponer de la máxima información posible acerca del sistema. Existen diversas fuentes de información, tales como:

- **La documentación** de las especificaciones de diseño del sistema. Aunque a menudo esta información está desactualizada y es incompleta, suele ser un punto de partida razonable para comprender el sistema.
- **Entrevistas con operarios e ingenieros** que conozcan el sistema o con sus diseñadores si éste todavía no ha sido construido. En efecto, durante el diseño del modelo puede invitarse a especialistas en el sistema a que examinen datos de salida, procedentes unos del sistema real y otros del modelo, y que intenten, sin conocer la respuesta, identificar (razonadamente) cuáles provienen del modelo y cuáles del sistema real. Esta información puede emplearse para mejorar el modelo.
- **Observaciones y medidas** realizadas de primera mano sobre el sistema. Sin duda ésta es la fuente de información más fiable: una vez leída la documentación y discutida la operación del sistema con operarios e ingenieros, es recomendable observar el sistema en operación. De hecho, en los primeros años de la investigación de operaciones se consideraba un procedimiento habitual que los ingenieros a cargo del modelado participaran durante cierto tiempo en la operación del sistema bajo estudio a fin de conocerlo mejor.

La tarea de modelado implica la búsqueda de un punto de equilibrio: el modelo debe representar los aspectos del sistema real con el grado de detalle requerido, pero de modo que sea lo más sencillo posible. No debe hacerse un modelo demasiado detallado, que incluya componentes que contribuyan poco o nada a la comprensión del problema. Debe tenerse presente que el objetivo del modelo es responder a las preguntas planteadas y no imitar de manera precisa el comportamiento del sistema real. Además, aumentando la complejidad del modelo, no necesariamente se consigue aumentar su realismo.

Los dos siguientes casos reales son descritos en (Law & Kelton 2000). En el primer caso, la dirección de una fábrica de comida para perros encargó a una consultoría la realización de un modelo de su línea de fabricación, que producía entorno a un millón de latas de comida por día. En el modelo se representó cada lata de comida como una entidad independiente, con lo cual era extremadamente costoso ejecutar el modelo. Por ello, el modelo no fue muy útil. El modelo fue reescrito unos años más tarde, tratando el proceso de fabricación como un flujo continuo. El nuevo modelo produjo resultados precisos y se ejecutaba en una pequeña fracción del tiempo necesario para ejecutar el modelo original.

- En ocasiones se considera que la distribución de probabilidad de algunos de los parámetros del modelo no varía con el tiempo. Esta aproximación es razonable si la velocidad de cambio es muy pequeña comparada con el periodo de tiempo de interés.

El impacto de las aproximaciones de modelado debe estimarse considerando el propósito específico del estudio de simulación:

- Si el propósito es estimar de manera precisa el comportamiento del sistema, deben evaluarse las simplificaciones considerando su impacto sobre la medida del comportamiento.
- Por otro lado, si el propósito del estudio es comparar diferentes sistemas o procedimientos, debe evaluarse si la simplificación afecta a todos los modelos aproximadamente de la misma manera (los errores son aproximadamente iguales en magnitud).

Por este motivo, estimar el comportamiento absoluto de un sistema normalmente requiere modelos considerablemente más detallados que comparar varios sistemas. Una consecuencia práctica de este hecho es que los modelos matemáticos suelen ser más precisos en la comparación de sistemas alternativos que en la obtención de respuestas absolutas.

1.5.3. Diseño de los experimentos

El diseño de los experimentos se realiza en dos etapas diferentes del estudio de simulación. En primer lugar, antes de que el diseño del modelo haya finalizado. Tan pronto como sea posible deben seleccionarse qué medidas del comportamiento del sistema van a usarse en el estudio, qué factores van a variarse y qué niveles de cada uno de estos factores van a investigarse. Una vez que el modelo ha sido desarrollado, verificado y validado, se revisa el diseño experimental teniendo en cuenta el conocimiento que se ha ganado durante el proceso de diseño, construcción, verificación y validación del modelo.

El experimento debe diseñarse cuidadosamente, atendiendo a las siguientes dos consideraciones:

- El diseño del experimento determina el tipo de análisis estadístico que debe aplicarse a los resultados.

han ocurrido simultáneamente en determinado instante (por ejemplo, a las 12 de la mañana).

- **Agregación de recursos:** se tratan varios recursos como una unidad. Por ejemplo, si un operario de una cadena de montaje realiza cierto número de trabajos consecutivos, puede modelarse la operación completa como un proceso simple en lugar de modelar cada una de las tareas individualmente.

El Teorema del Límite Central a menudo ayuda a determinar la distribución de probabilidad resultante del proceso agrupado. Si la variable aleatoria del proceso agrupado (por ejemplo, el tiempo en realizar la tarea) es suma de las variables aleatorias de los subprocesos y si éstas son independientes entre sí, entonces la variable del proceso agrupado está aproximadamente distribuida de forma normal cuando el número de subprocesos es grande (sean cuales sean las distribuciones de probabilidad de las variables de los subprocesos).

- El tercer tipo de simplificación consiste en **sustituir** un proceso complejo por un proceso diferente, más sencillo, que se comporte de manera equivalente al proceso original. Un ejemplo es la representación de un proceso complejo, cuyo comportamiento no se comprende o no se desea modelar, por medio de determinada distribución de probabilidad. De este modo, se sustituye un componente del sistema por entradas al sistema. Por ejemplo, para modelar los pedidos realizados por determinado cliente puede, o bien modelarse toda la estructura de negocio del cliente, o bien considerar que sus demandas están distribuidas aleatoriamente de acuerdo a determinada distribución de probabilidad.

Además de la aplicación de las técnicas anteriores, comúnmente se realizan las siguientes aproximaciones:

- La definición de los "límites" del modelo y de su interacción con el entorno implica la realización de aproximaciones, entre ellas la decisión de qué factores externos influyen sobre el funcionamiento del modelo (entradas al modelo) y cuáles son ignorados.
- Los fenómenos de naturaleza aleatoria son modelados mediante funciones de probabilidad (frecuentemente distribuciones teóricas) que los representan sólo de manera aproximada.
- Al modelar la interacción entre los fenómenos aleatorios, en ocasiones se considera (de forma aproximada) que dos variables aleatorias son estadísticamente independientes con el fin de simplificar el modelo.

Por ejemplo, algunas de las herramientas software que facilitan la aplicación del formalismo DEVS son ADEVS, CD++, BEVS/C++, DEVSJAV, DEVSim++, JDEV, PowerDEV, Python DEV, SmallDEV, etc.

1.5.6. Verificación y validación del modelo

La finalidad de la **verificación** es comprobar que no se han cometido errores al codificar el modelo usando un lenguaje o entorno de simulación. La **validación** consiste en comprobar que el modelo supone una aproximación adecuada de la realidad para los objetivos particulares del estudio de simulación. La verificación y validación de un modelo son análisis conceptualmente distintos, si bien en la práctica aparecen relacionados. Cuando los resultados de la simulación parecen “extraños” o erróneos, debe investigarse si es debido a equivocaciones al traducir el modelo (de su definición formal a su descripción en lenguaje de simulación) o a que las hipótesis de modelado no son las adecuadas.

Algunos de los procedimientos que se emplean para verificar un modelo son los siguientes:

- **Verificación manual de la lógica.** Consiste en ejecutar la simulación durante un periodo de tiempo corto y comprobar manualmente los resultados obtenidos.
- **Comprobación submodelo a submodelo.** Se trata de verificar individualmente que cada submodelo produce los resultados esperados para todos los posibles tipos de entradas.
- **Comprobación con soluciones conocidas.** Consiste en ajustar el modelo de modo que represente un sistema de solución conocida y comparar ésta con los resultados de la simulación.
- **Test de sensibilidad.** Puede modificarse el valor de un parámetro, dejando los demás fijos, con el fin de medir la sensibilidad del modelo respecto a ese parámetro. La comparación de la sensibilidad observada en las simulaciones, con la que sería de esperar en el sistema real, puede proporcionar pistas útiles.

Respecto a la validación de los modelos, puede considerarse que ésta tiene tres vertientes diferentes. Consiste en determinar:

1. Si la estructura del modelo representa adecuadamente al sistema real.

- Los experimentos de simulación deben diseñarse de modo que permitan obtener la máxima información acerca del comportamiento del sistema al menor costo posible (tiempo y trabajo del experimentador, y tiempo de computación).

1.5.4. Datos de entrada

Determinar qué datos emplear como entrada al modelo durante la simulación es una tarea muy delicada, de la que depende en gran medida el éxito del estudio. De hecho, frecuentemente la recogida y el análisis de los datos del sistema, y el modelado de las entradas a partir de estos datos, consume más tiempo que el propio diseño y programación del modelo.

El diseño de un modelo estocástico de simulación siempre implica decidir si determinados aspectos del sistema son representados de forma estocástica o determinista. Si se opta por una descripción probabilística de determinado aspecto, puede suceder que:

- Puedan obtenerse datos experimentales del sistema. En este caso, debe decidirse si durante la simulación se muestrea directamente de los datos experimentales, o bien si se muestrea de una distribución de probabilidad ajustada a partir de éstos.
- No puedan obtenerse datos experimentales, bien porque el sistema aun no ha sido construido o porque no es posible recogerlos. En este caso, puede uno basarse en consideraciones teóricas y en estimaciones de operarios, ingenieros, diseñadores, etc. familiarizados con el funcionamiento del sistema.

1.5.5. Traducción del modelo

El modelo puede describirse formalmente empleando diferentes metodologías. En las secciones precedentes se han introducido dos de ellas: el modelado orientado a la planificación de eventos y el modelado orientado a los procesos. Cada una de ellas es soportada por diferentes herramientas software, que ayudan a definir y simular el modelo. Por ejemplo, el modelado orientado a los procesos es facilitado por lenguajes de simulación (GPSS, SIMSCRIPT, SLAM, SIMAN, etc.) y entornos de simulación (AnyLogic, Arena, AutoMod, Enterprise Dynamics, ExtendSim, Flexsim, ProModel, SIMUL8, etc.) Igualmente, se han desarrollado herramientas software para facilitar la aplicación de los formalismos basados en la planificación de eventos.

- **Simulación con terminación.** La simulación se dice que tiene terminación cuando existe un evento que señalará de forma natural el final de la misma. La finalidad es estudiar el comportamiento del sistema cuando éste evoluciona, partiendo de unas determinadas condiciones iniciales, hasta que se satisface determinada condición de finalización.
- **Simulación sin terminación.** La simulación es sin terminación cuando no existe un evento que de manera natural marque el final de la misma. La finalidad de este tipo de estudios es analizar el comportamiento del sistema una vez éste ha adquirido un régimen de funcionamiento independiente de las condiciones iniciales de la simulación, es decir, el estado estacionario.

Puesto que la salida de un modelo de simulación es una muestra de datos de su comportamiento, pueden aplicarse los conceptos de inferencia estadística de muestras. Sin embargo, la mayoría de los tests estadísticos requieren que los datos de la muestra sean independientes entre sí. Es decir, no estén correlacionados. Frecuentemente los datos obtenidos de la simulación de los modelos no cumplen esta condición, con lo cual deben aplicarse ciertos procedimientos antes de poder extraer conclusiones válidas. Respecto al tamaño de la muestra (duración de la simulación o número de réplicas), éste debe ser lo suficientemente grande como para proporcionar el nivel de confianza necesario en la estimación de la medida del comportamiento.

1.5.8. Documentación y aplicación de los resultados

Llegado este punto, ya se han completado todos los pasos del diseño y desarrollo del modelo, ejecución de la simulación, así como del análisis de los resultados. Los elementos finales del estudio de simulación son la aplicación de sus resultados y la documentación. No puede considerarse que un proyecto de simulación ha sido completado con éxito hasta que los resultados obtenidos no hayan sido entendidos, aceptados y usados.

Las conclusiones del estudio deben ser documentadas de manera clara, concisa y convincente, debiéndose abordar todos aquellos aspectos que el cliente considera importantes. De otra forma, difícilmente serán aplicadas, con lo cual el proyecto será un fracaso. La presentación de los resultados es una parte del estudio tan crítica como las demás, y merece el mismo cuidado en su planificación y diseño.

Además de los resultados del estudio, debe documentarse el desarrollo y operación del modelo. Esto puede contribuir a aumentar considerablemente su vida útil, así

2. Si los datos calculados al simular el modelo (es decir, el comportamiento del modelo) reproducen de forma adecuada el comportamiento del sistema real.
3. Si el usuario del modelo tiene confianza en los resultados obtenidos de las simulaciones. Involucrar al usuario final en todas las fases del diseño y la construcción del modelo generalmente hace que este aspecto de la validación del modelo sea mucho más sencillo. En (Law & Kelton 2000) se describe el siguiente caso real. Un analista trabajó durante 6 meses en un modelo, sin interactuar con la persona que lo había solicitado. En la reunión mantenida para la presentación de los resultados del estudio, cuando tan sólo habían transcurrido 5 minutos, la persona que había encargado el estudio afirmó: “Este no es el problema en el que yo estoy interesado”.



Puesto que el modelo se construye para un propósito específico, la validez sólo puede ser evaluada con relación a este propósito. La validación del modelo es un proceso continuo durante su diseño, desarrollo y uso. Existen diferentes grados de validación: la confianza en el modelo va acumulándose según el modelo va superando pruebas y se van encontrando más puntos de coincidencia entre el comportamiento del modelo y el del sistema real. La verificación y la validación de un modelo son procesos que realmente nunca finalizan.

En todo este proceso de validación, no debe perderse de vista que el objetivo del ingeniero dedicado al modelado es la realización de modelos útiles, en un tiempo razonable y con un coste razonable. Por este motivo, más que preguntarse en qué medida se ajusta el comportamiento simulado al comportamiento real del sistema, es más adecuado preguntarse en qué medida las diferencias entre el modelo y el sistema son lo suficientemente significativas como para afectar a las conclusiones derivadas del uso del modelo.

1.5.7. Experimentación y análisis de los resultados

En el caso de los modelos estocásticos, la planificación del experimento y el análisis de los resultados se realiza aplicando técnicas estadísticas. Cabe distinguir dos tipos de simulación de modelos estocásticos: “con terminación” y “sin terminación”. El tipo de simulación (con o sin terminación) más adecuado en cada caso depende del objetivo del estudio, no de la naturaleza del sistema. Las características de estos dos tipos de simulación se describen a continuación:

```
R version 2.14.0 (2011-10-31)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

> |
```

Figura 1.13: Ejemplo de la consola de R en Windows.

defecto. Otras se encuentran en paquetes, que deben ser abiertos (esta acción se denomina *attach*) para poder usar las funciones que contienen.

Las sentencias consisten en funciones y asignaciones. R usa el símbolo `<-` para las asignaciones, en lugar del típico símbolo `=`. Por ejemplo,

```
> x <- rnorm(10)
```

crea un objeto de tipo vector llamado `x`, que contiene 10 observaciones de la distribución normal estándar.

Los comentarios son precedidos por el símbolo `#`. El intérprete de R ignora todo el texto que aparezca tras el símbolo `#`. Para finalizar una sesión debe ejecutarse la función `q()`.

La función `c()` convierte sus argumentos en un vector o una lista. Por ejemplo, supongamos que se realizan 5 réplicas independientes de la simulación del modelo de la oficina, obteniéndose los siguientes 5 valores del tiempo medio de espera en cola: 1.2, 2.4, 1.4, 2.2 y 3.2 minutos. La primera de las siguientes dos sentencias crea un objeto del tipo vector llamado `tCola`, cuyos componentes son esos cinco valores. La segunda sentencia dibuja el valor del componente del vector frente al índice. En la Figura 1.14 se muestra el gráfico.

```
> tCola <- c(1.2, 2.4, 1.4, 2.2, 3.2)
> plot(tCola)
```

como aumentar la probabilidad de que recomendaciones basadas en él sean aceptadas. Una buena documentación facilita las modificaciones y asegura que el modelo puede ser usado o algunas de sus partes reutilizadas, aun si no están presentes sus desarrolladores.

1.6. ANÁLISIS DE DATOS: INTRODUCCIÓN A R

El análisis de los datos es una parte esencial en un proyecto de simulación. Por una parte es preciso analizar los datos del sistema bajo estudio, con el fin de modelar su comportamiento. Por otra parte es preciso analizar los datos obtenidos como resultado de la simulación del modelo, con el fin de extraer conclusiones.

Como ya se ha indicado anteriormente, algunos entornos de simulación incorporan herramientas de ayuda para el análisis de los datos. Estas herramientas típicamente permiten representar los datos gráficamente (por ejemplo, mediante gráficas X-Y, boxplots, histogramas y gráficas Q-Q) y realizar ajustes (por ejemplo, ajuste de distribuciones de probabilidad y superficies de respuesta). Sin embargo, un buen número de entornos de simulación no proporcionan estas ayudas, limitándose a facilitar la descripción y simulación del modelo. En este segundo grupo se encuentran la mayoría de los entornos y herramientas de simulación gratuitas, las cuales deben proporcionar una combinación de alguna herramienta para el análisis y modelado de los datos.

Existe una amplia variedad de herramientas especializadas en el análisis y modelado de los datos. En esta sección se presenta una de ellas: el lenguaje R y la plataforma software que lo soporta. Se trata de software gratuito y de código abierto, con versiones para los sistemas operativos Windows, Mac OS X y Linux. Puede descargarse del Comprehensive R Archive Network (CRAN), que está ubicado en <http://cran.r-project.org/>.

R es un lenguaje interpretado, en el cual se distingue entre letras mayúsculas y minúsculas. Es posible ir introduciendo los comandos uno a uno en la línea de comandos de la consola, tras el símbolo del sistema (>), o bien ejecutar un conjunto de comandos escritos en un fichero. En la Figura 1.13 se muestra un ejemplo de la consola de R en Windows.

El lenguaje R tiene una gran variedad de tipos de datos, incluyendo vectores, matrices, data frames y listas. La mayor parte de la funcionalidad se consigue mediante el uso de funciones, tanto las proporcionadas por el lenguaje como las definidas por el propio usuario. Algunas funciones básicas están disponibles por