

3 METODOLOGÍA DE SIMULACIÓN

3.1 RECOGIDA DE LA INFORMACIÓN DE ENSAYOS CLÍNICOS.

3.1.1 SUJETOS DE ESTUDIO

Se han analizado diferentes tipos de ensayos clínicos ya realizados previamente (retrospectivos) y en realización (prospectivos), así como datos de la bibliografía y se han utilizado estos datos para el diseño, validación y ajuste del simulador basado en el entorno LeanSim. Estos datos están en soporte informático, en bases de datos o contenidos en ficheros de paquetes estadísticos estándar como SAS 8.02 y SPSS 11.5.

Se han estudiado diferentes ensayos clínicos a fin de simular sus datos y ajustar el sistema de simulación realista.

3.1.2 VARIABLES DE ESTUDIO

Se han estudiado todas las variables que puedan tener repercusión en la simulación realista de ensayos clínicos. Algunas de éstas son: diseño del ensayo clínico (controlado, aleatorizado, medidas repetidas, "cross-over", grupos en paralelo, diseños factoriales, etc), rango de respuestas del ensayo, variables principales, variables secundarias, covariantes, número de centros, número de tratamientos, criterios de inclusión / exclusión, dosis, farmacocinética / farmacodinámica, número de pacientes que abandonan, número de visitas, cumplimiento de la medicación, tipología de las variables del estudio (nominales, discretas, continuas), datos faltantes, distribución estadística de las variables del estudio, hipótesis del estudio, métodos estadísticos para las pruebas de hipótesis, patología estudiada, tamaño de la muestra, tipo de pacientes, distribución temporal de los pacientes, etc.

3.2 MÉTODOS DE SIMULACIÓN

3.2.1 MODELIZACIÓN DE LOS ENSAYOS CLÍNICOS

Como se ha mencionado, en este trabajo se ha recogido información de una manera directa de las particularidades y diseños de ensayos clínicos a lo largo de tres años 2000-2033 (Hipócrates) o a partir de matrices de datos de ensayos clínicos proporcionados por las diferentes entidades biomédicas que han colaborado en este estudio y de la bibliografía estudiada.

Este trabajo se centra en los diseños experimentales de medidas repetidas ya que constituyen uno de los diseños más ampliamente utilizado en los ensayos clínicos. Los modelos de medidas repetidas se ajustan muy bien a la mayoría de los ensayos clínicos donde se plantea un modelo de variables experimentales (tratamiento, tiempo, paciente, covariantes, centro, etc) que intentan explicar una o más variables de respuesta (variable principal del estudio).

DISEÑOS DE MEDIDAS REPETIDAS

Se caracterizan por ser unos diseños donde las observaciones de medidas parecidas se toman durante una serie de tiempos. Algunas características de este diseño son:

- La misma variable es medida en la misma unidad observacional más de una vez. Las respuestas no son independientes tal y como se suele suponer en lo modelos de regresión habituales.
- Se recogen datos para más de una unidad observacional: las respuestas no forman una serie temporal simple, sino un conjunto de series temporales.

Son diseños muy frecuentes en el campo biológico y recogen la mayoría de los utilizados en los ensayos clínicos de fármacos.

Son necesarias medidas repetidas de una variable por los siguientes motivos:

- Las medidas repetidas son la única vía de obtener mediciones, contando las ocurrencias del mismo fenómeno.
- El interés puede estar centrado en la evolución de la misma respuesta, dadas unas condiciones iniciales que pueden ser o no fijadas experimentalmente.
- El investigador puede estar interesado en observar el efecto continuado de un tratamiento en el tiempo.
- Pueden ser necesarias medidas diferentes para realizar comparaciones en una situación donde la variabilidad entre las unidades es un factor incontrolable. Para incrementar la precisión, son necesarias las comparaciones intra-unidad de los diferentes tratamientos.
- Para estudiar el efecto total de las diferentes secuencias de tratamientos, tal y como pasa en los estudios cruzados ("cross-over").

La unidad observacional de los modelos de medidas repetidas son los pacientes a los que se realizarán medidas de respuestas en diversos periodos de tiempo.

Usualmente los datos clínicos se analizan usando modelos de efectos fijos, pero es más beneficioso el uso de los modelos mixtos, donde se aísla una parte fija y una parte aleatoria.

El uso de estos modelos mixtos estaba muy limitado hace años por el poco software existente, pero actualmente existen procedimientos bien validados y fáciles de usar para la estimación de modelos mixtos, como PROC MIXED del paquete SAS.

A modo de ejemplo introductorio, se describe la modelización de un ensayo clínico para el tratamiento del SIDA, descrito en Monleón y col. (2003^a) basado, con algunas modificaciones, en el modelo general de ensayo clínico con medidas repetidas propuesto por Vonesh y Cinchilli (1997).

Modelo Lineal General

En el caso estudiado, la variable de respuesta es el logaritmo de la carga viral. Vonesh y Chinchilli (1997) proponen un modelo de ensayo clínico de medidas repetidas basado en una modificación del modelo clásico de medidas repetidas de efectos fijos al que se ha incorporado efectos aleatorios univariantes:

$$\log CV_{ijk} = m + t_i + d_{j(i)} + V_k + (tV)_{ik} + e_{ijk} \quad (1)$$

Donde

$\log CV_{ijk}$ = Logaritmo de la carga viral para individuo i , tratamiento j y visita k .

m = media general

t_i = efecto fijo del tratamiento i

V_k = efecto fijo de la visita k .

$(tV)_{ik}$ = efecto fijo del tratamiento i y de la visita k .

$d_{j(i)}$ = efecto aleatorio del paciente j en el tratamiento i .

e_{ijk} = error aleatorio.

Se asume que e_{ijk} y $d_{j(i)}$ son independientes y se distribuyen según $e_{ijk} \sim N(0, \sigma_e^2)$, $d_{j(i)} \sim N(0, s_d^2)$. Mediante la transformación descrita en Vonesh y Chinchilli (1997) se obtiene como resultando un modelo más interpretable y sencillo de simular:

$$\log CV^*_{ijk} = V^*_k + (tV)^*_{ik} + e^*_{ijk} \quad (2)$$

Equivalente a (1), donde,

V^*_k = efecto fijo de la visita k.

$(tV)^*_{ik}$ = efecto de la interacción entre el tratamiento i de la visita k

e^*_{ijk} = error aleatorio.

El principal inconveniente del modelo (2) (ó 1) es su excesiva parametrización (p.e. es necesario estimar un parámetro para describir el efecto de cada instante de visita), con los consiguientes problemas de inferencia simultánea. Tampoco representa de una manera natural la heterogeneidad en las respuestas individuales. Estos defectos están en gran parte paliados bajo un enfoque de modelos mixtos.

Modelo Lineal Mixto

El modelo lineal mixto se puede plantear, en términos generales (véase, por ejemplo, Verbeke y Molenberghs, 2001) de la siguiente forma:

$$Y_i = X_i b + Z_i b_i + e_i \quad (3)$$

Donde Y_i es el vector de observaciones correspondiente al individuo i , X_i y Z_i son matrices de diseño, b es un vector de parámetros fijos, $b \sim N(0, D)$ es un vector de parámetros aleatorios, $e_i \sim N(0, \hat{\sigma}_i)$ es el vector de residuos para el individuo i y $b_1, \dots, b_N, e_1, \dots, e_N$ son independientes.

Un posible modelo para el caso estudiado es:

$$\log CV_{ij} = (b_0 + b_{0i}) + (b_1 + b_{1i}) + (b_2 + b_{2i}) \text{Visita}_{ij} + e_{ij} \quad (4)$$

Donde:

$\log CV_{ij}$ = logaritmo de la carga viral del individuo i en la visita j ,

b_0, b_{0i} = Parte fija y aleatoria de la carga viral inicial,

b_1, b_{1i} = Parte fija y aleatoria del efecto del tratamiento,

b_2, b_{2i} = Parte fija y aleatoria de efecto de la visita,

Visita_{ij} = Tiempo (meses) desde inicio del tratamiento, y

e_{ij} = error aleatorio.

Para el análisis del modelo de medidas repetidas propuesto y la estimación de parámetros se ha empleado los procedimientos PROC MIXED de SAS.

En este caso concreto la modelización y simulación de los ensayos clínicos se ha realizado también utilizando el paquete estadístico SAS 8.

3.2.2 GENERACIÓN DE NORMALES MULTIVARIANTES MEDIANTE EL MÉTODO DE BÚSQUEDA DE JACOBI DE DERIVACIÓN DE VALORES PROPIOS Y VECTORES PROPIOS

Para minimizar el efecto de la variabilidad inter-respuesta e intentar simular al máximo la dependencia entre las medidas de un mismo paciente a los largo de las diferentes visitas, se han probado diferentes métodos de generación de variables normales multivariantes y finalmente por su facilidad computacional se ha optado por el método de Jacobi, tal como se describe en Golub y Van Loan (1996). Las ventajas computacionales en cuanto a velocidad y simplicidad se discuten en Press y col. (2002).

Ilustrándolo, para mayor simplicidad, en el caso bivalente, el algoritmo seguido es el siguiente:

Paso 1: Dada la matriz de correlaciones $\begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}$ que se desea simular, se

obtiene la siguiente matriz de vectores propios: $\begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$ con valores propios

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix},$$

Paso 2: Se calcula la raíz cuadrada de los valores propios y se organiza en

forma de matriz diagonal: $\begin{bmatrix} \sqrt{V_1} & 0 \\ 0 & \sqrt{V_2} \end{bmatrix}$,

Paso 3: Se multiplica la matriz diagonal anterior por la traspuesta de la

matriz de vectores propios: $\begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \times \begin{bmatrix} \sqrt{V_1} & 0 \\ 0 & \sqrt{V_2} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = X$,

Paso 4: Se multiplican los números aleatorios normales estándar e

independientes $\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}$ por la traspuesta de X , $\begin{bmatrix} X_{11} & X_{21} \\ X_{12} & X_{22} \end{bmatrix}$, para obtener un par

de valores generados de acuerdo con la distribución normal bivalente

asociada a la matriz de correlaciones inicial y centrada $\begin{bmatrix} Y_1 = X_{11}Z_1 + X_{21}Z_2 \\ Y_2 = X_{12}Z_1 + X_{22}Z_2 \end{bmatrix}$

En el anexo I se puede encontrar el código de programa en lenguaje Visual Basic que se ha desarrollado a partir del trabajo de Sun (1997) y que permite la generación de distribuciones normales multivariantes. Este método se utilizará para simular variabilidad multivariante inter-paciente en algunos de los modelos que se presentan en esta tesis.

3.2.3 SIMULACIÓN DE EVENTOS DISCRETOS

En relación con el tratamiento que realizan del parámetro "tiempo", se suelen considerar 3 tipos de sistemas y/o de modelos:

- Continuos.
- Discretos.
- De eventos discretos.

Un sistema continuo es aquel que puede ser descrito mediante un sistema de ecuaciones diferenciales, tal como pueden ser los parámetros

farmacocinéticos y farmacodinámicos que se miden continuamente en un periodo de tiempo en voluntarios sanos o en pacientes de una enfermedad.

Un sistema discreto es aquel que es observado en momentos de tiempo discretos prefijados, como por ejemplo en aquellos casos en que se inspecciona un proceso una vez semanalmente, o en una serie de visitas como es el caso de los ensayos clínicos. En los congresos y workshops en los que se ha participado (ISCB, ISPOR, COMSTAT, SEIO, ENBIS, etc.) y que han sido mencionado en esta tesis, se ha observado un aumento sustancial del número de trabajos presentados con relación a este enfoque.

La principal característica de un sistema de eventos discretos (Barceló, 1996; Guasch y col., 2002) es que el sistema está determinado por una secuencia de eventos que ocurren en puntos temporales aleatorios t_1, t_2, \dots y el cambio de estado del sistema tiene lugar en esos instantes. En el caso de los ensayos clínicos los pacientes serán las entidades del sistema y los diferentes eventos serán las visitas y cambios de estado de salud (respuesta), desde un nivel basal al final tras la toma de un tratamiento farmacológico que modificará la fisiología del paciente.

Cuando se escribe un programa de simulación para un modelo de eventos discretos se puede realizar una aproximación del esquema temporal de funcionamiento o de modificación de las entidades en el sistema. Así deberán describirse la secuencia de eventos y actividades que realizarán las entidades durante su estancia en el sistema y cómo se modificarán. Algunos de los sistemas más estudiados de simulación de eventos discretos son los problemas de colas que en el caso de los ensayos clínicos corresponderán a la espera que deben tener los pacientes durante visita y visita, si los centros donde son tratados no pueden absorber toda la demanda de aquellos (En un centro e instante de tiempo sólo puede ser tratado un paciente a la vez).

En general, en la simulación de eventos discretos existen N diferentes eventos y a cada uno le corresponde una actividad: A_1, \dots, A_N . En la lista de eventos se

puede indicar los puntos temporales: T_1, \dots, T_n en los que ocurren los diferentes eventos. La simulación consistirá en encontrar el menor T_i , dado el momento actual, hasta un evento concreto y ejecutar las actividades correspondientes. El planteamiento de una simulación de eventos discretos es la siguiente:

INICIALIZACION

Dado un tiempo = 0;

Inicializar estado del sistema y los contadores estadísticos;

Inicializar lista de eventos;

PROGRAMA PRINCIPAL

Mientras tiempo de simulación < regla de tiempo

Hacer:

Determinar tipo del próximo evento;

Avanzar tiempo de simulación;

Actualizar estado del sistema + contadores estadísticos;

Generar futuros eventos + añadirlos a la lista de eventos;

Finalizar;

RESULTADOS

Computar + imprimir estimaciones de interés;

3.2.4 ENTORNO LEANSIM

Para el desarrollo de los modelos de simulación de esta tesis se han utilizado dos metodologías, una basada en el pseudo-lenguaje de propósito general del paquete estadístico SAS (SAS Institute, Cary, NC), ampliamente utilizado en el ámbito biosanitario y otra utilizando el entorno de simulación LeanSim (Guasch y col., 2003) que se describirá a continuación. No se describirá el entorno de programación del pseudo-lenguaje SAS ya que es de ámbito genérico y existen manuales específicos de su uso.

LeanSim brinda un entorno abierto y flexible de diseño y desarrollo de modelos de simulación genéricos agrupados por ámbitos de trabajo. La realización de esta tesis doctoral, conjuntamente con el proyecto PETRI ha permitido ampliar el conjunto de ámbitos con uno nuevo que permite al usuario final crear y modelar procesos referentes al análisis de ensayos clínicos.

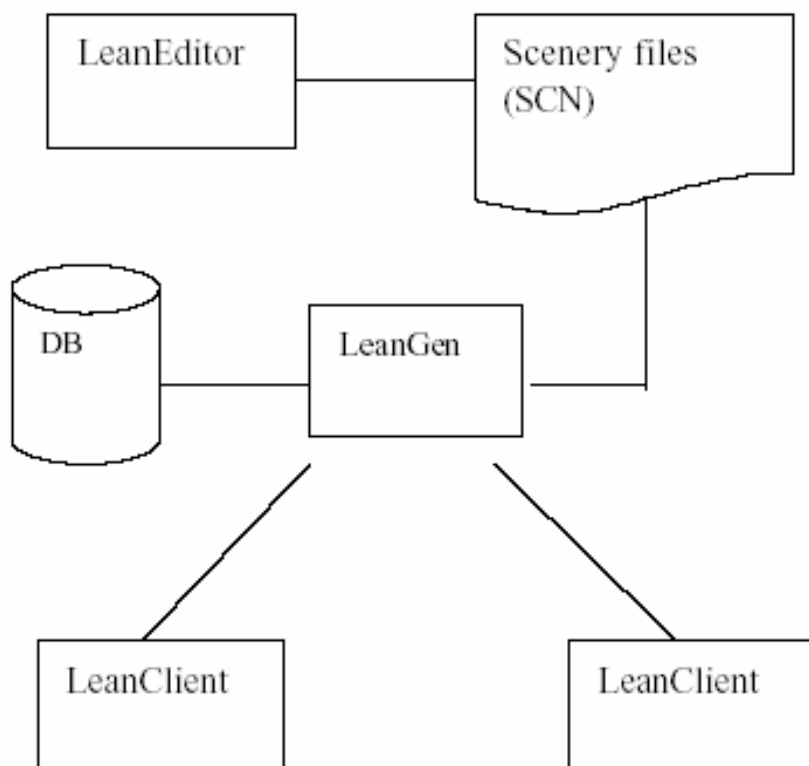


Figura 11: Entorno de la suite de LeanSim (de Fonseca y col., 2003).

La herramienta LeanSim es un entorno de simulación discreta formado por una biblioteca de clases en C++ creada por el grupo de investigación de simulación del Dr. Josep Casanovas de la UPC (Fonseca y col., 2003, 2004^{a, b}). Contiene componentes para el manejo de recursos de simulación, un sistema de sincronización entre las actividades que se realizan en cada evento y los objetos, un motor de simulación (núcleo de simulación) adaptado para

soportar múltiples ejecuciones (réplicas), recogida y análisis estadístico de los datos obtenidos.

LeanSim combina dos de los principales paradigmas de la simulación, como son, la orientación a procesos y la programación de eventos. El primero de ellos se utiliza en la fase de desarrollo del modelo de simulación, a partir de la definición de los elementos de simulación significativos de ser examinados y las acciones asociadas a cada una de las entidades estudiadas se definen los procesos que se analizarán en el modelo. La programación de eventos permite avanzar el reloj de simulación y priorizar las acciones asociadas a las entidades, recursos y demás elementos de simulación en función del tiempo de ejecución o programación y la prioridad del evento, este paradigma está vinculado al proceso de experimentación del modelo.

En la **Figura 11** puede verse los diferentes componentes de LeanSim: Un programa editor (LeanEditor) desde el que se crean los modelos conceptuales de simulación y la representación visual de la simulación (“Scenery Files”), un programa generador del modelo de simulación ejecutable (LeanGen) y un programa cliente donde se distribuye la representación (LeanClient) que será finalmente el que realice las simulaciones sobre el modelo de simulación generado.

Además, en el plano de la representación visual de la simulación se añade un conjunto de clases que permiten una representación en un universo tridimensional. Esta tecnología permite analizar el comportamiento de modelos de simulación muy complejos, como es el caso de los ensayos clínicos, y observarlos en un entorno de realidad virtual.

Como se detallará en el capítulo de la validación de los modelos de simulación estudiados se ha validado el uso de LeanSim para la simulación de ensayos clínicos en el trabajo presentado en el congreso COMSTAT 2004 (Monleón y col., 2004) y en trabajos precedentes como Monleón y col. (2003^{a, b, c}), donde

se presentaron los principales aspectos de la simulación de ensayos clínicos basada en un modelo general de simulación ejecutado por LeanSim.

Las entidades de simulación son aquellos componentes o partes del sistema real que se desea observar como evolucionan a lo largo del estudio de simulación, éstas pueden ser de dos tipos claramente diferenciados en función de su persistencia en el modelo: entidades temporales que típicamente representan los elementos temporales que fluyen en el modelo dentro del ámbito de ensayos clínicos y que corresponderían a los sujetos que participan del ensayo; y las entidades permanentes donde se realizan las actividades que condicionan los estados asociados a las entidades temporales. En la aproximación planteada en LeanSim se ha adoptado la terminología de entidades como los elementos temporales y se agrupan las entidades permanentes como objetos o elementos de simulación agrupados en categorías según sus funcionalidades. Las entidades pueden tener sus propios atributos, o tipos de datos abstractos que permiten al usuario definir sus diferentes características.

Los tipos definidos de atributos son:

- Int.
- Double.
- Boolean.
- Distribution (Constant, Probabilistic distributions {Erlang, Exponential, Geometrical, LogNormal, Normal, Poisson, Triangular, Uniform, Weibull}).
- Vector.

En cuanto a los detalles de visualización del modelo de simulación se siguen los siguientes pasos:

- Creación de los objetos del mundo real y las entidades que se mueven por el sistema.
- Introducir luces, cámaras, sonidos, definir las interacciones y movimientos de los objetos.
- Asociar el objeto real al objeto de simulación, teniendo en cuenta los eventos y estados que hacen cambiar la vista del objeto.
- Integrar todos los componentes en el entorno de simulación LeanSim.

A las librerías de simulación del programa se ha añadido una clase de generación de normales multivariantes desarrollada por el autor según el método de Jacobi y que se puede ver en el CD anexo. Esta clase permite generar variabilidad inter-paciente y fue presentada por Monleón y col. (2003^a) en el congreso de la Sociedad Española de Biometría en 2003.

3.2.5 FUNCIONAMIENTO INTERNO DE LEANSIM

A continuación se describe un resumen del funcionamiento interno del entorno LeanSim que ya se ha presentado en la *Figura 11* y como se construyen los modelos de simulación en el mismo (LCFIB, 2005).

LeanSim posee una serie de objetos de simulación, o elementos, que el usuario puede utilizar en el modelado de sistemas y que se clasifican según su funcionalidad en diversas familias. A continuación pueden verse los diferentes elementos de simulación asociados a LeanSim comentados en su manual de instrucciones (LCFIB, 2005):

Funcionalidad	Familia del objeto	Elementos	Modo de Ejecución
Creación de entidades que entran en el sistema.	Generador	Generador Estocástico Generador Bajo Demanda Generador Frecuencia Generador Script	Básico o Múltiple
Destrucción de entidades	Destrucción	Destructor	Básico o

que abandonan el sistema			Múltiple
Creación de entidades a partir de otras entidades	Empaquetadores	Empaquetador Desempaquetador Transformadores	Básico
Elementos de control de operaciones	Recursos	Recursos Estáticos Recursos Dinámicos Recursos Humanos	Básico o Múltiple
Almacenamiento de entidades	Almacenes	Silo	Básico o Múltiple
	Estantes	Estantes	Básico o Múltiple
	Almacén de baja estacionalidad	Colas con Time Out	Básico o Múltiple
		Colas Estándares	Básico
Procesadores de tiempo	Servidores	Servidor Estático Servidor Dinámico	Básico
Control de procesos	Controladores	Bifurcadores Selectores Enrutadores Señalizadores	Básico

Existe una correspondencia entre estos objetos con las operaciones realizadas durante los ensayos clínicos, así el objeto generador correspondería a los médicos que incluyen a pacientes, el objeto destructor a los pacientes que finalizan con el ensayo clínico y los objetos de servicios a los hospitales.

Durante el movimiento de las entidades a través del sistema pueden realizarse diferentes actividades. Estas actividades representan las transformaciones con las que pueden ser modificadas las entidades a lo largo de su vida en el modelo. Estas actividades se denominan operaciones o funciones y son las siguientes:

- ☉ Funciones definidas por el usuario
- ☉ Función Generar

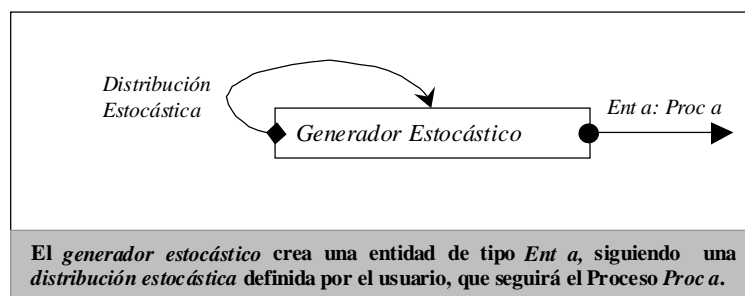
- ☺ Destruir
- ☺ Modificar
- ☺ Enviar señal
- ☺ Bifurcar
- ☺ Selección Máquina
- ☺ Pedir Recurso
- ☺ Liberar Recursos
- ☺ Mover
- ☺ GoTo
- ☺ Transformar
- ☺ Almacenar
- ☺ Pack
- ☺ Unpack
- ☺ Recolectores (Operaciones de recogida de información)

3.2.5.1.1 Generadores estocásticos

La creación de entidades se define a partir de una distribución estocástica relacionada con el tiempo entre llegadas. Las posibles distribuciones utilizadas se recogen en la siguiente tabla:

El tiempo entre llegadas se define como ...
Un valor Constante
Una distribución Erlang
Una distribución Exponencial
Una distribución Geométrica
Una distribución Log normal
Una distribución Normal
Una distribución Poisson
Una distribución Triangular
Una distribución Uniforme
Una distribución Weibull

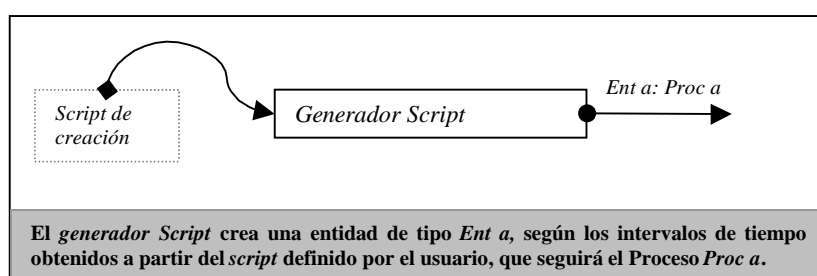
El siguiente diagrama describe la operativa de trabajo asociada a un generador estocástico bajo demanda:



3.2.5.1.2 Generadores de "script"

Aquellos procesos de generación de entidades asociados a la evaluación de una o más variables de estado del sistema se pueden simular mediante el uso de los generadores de "script", para cada uno de ellos se define un "script", que es evaluado en tiempo de ejecución y, que define el instante de tiempo en que una nueva entidad entrará en el sistema.

El siguiente diagrama describe la operativa de trabajo asociada a un generador de "script":



Los "scripts" se utilizarán para introducir en el simulador el modelo de comportamiento del fármaco y sus parámetros, así como el modelo conceptual que lo acompaña (centros, visitas, número de pacientes, abandonos, aleatorización de los tratamientos, etc).

3.2.5.2 LEANEDITOR

LeanEditor es el editor de modelos de simulación de LeanSim y permite desarrollar estudios de simulación a partir de las plantillas de diseño. En la *figura 12* puede verse un detalle de la interfaz del mismo y se compone de un sistema de menús, diferentes barras de herramientas, y áreas gráficas para el diseño y representación de los modelos.

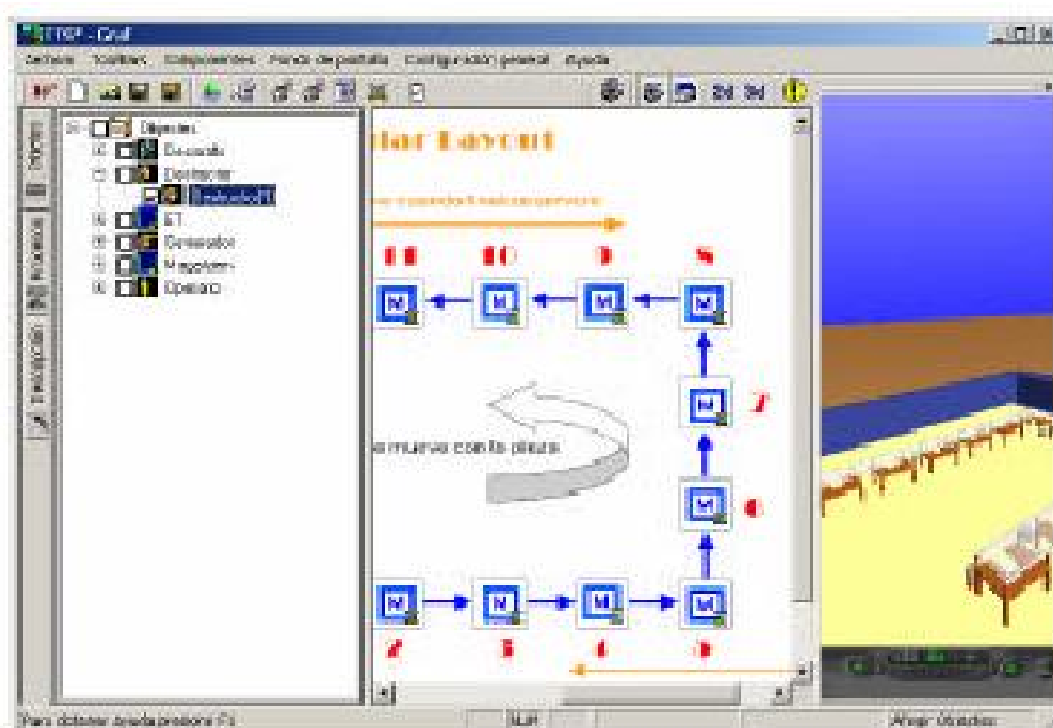


Figura 12: Aspecto del editor de LeanSim con esquema de simulación de proceso.

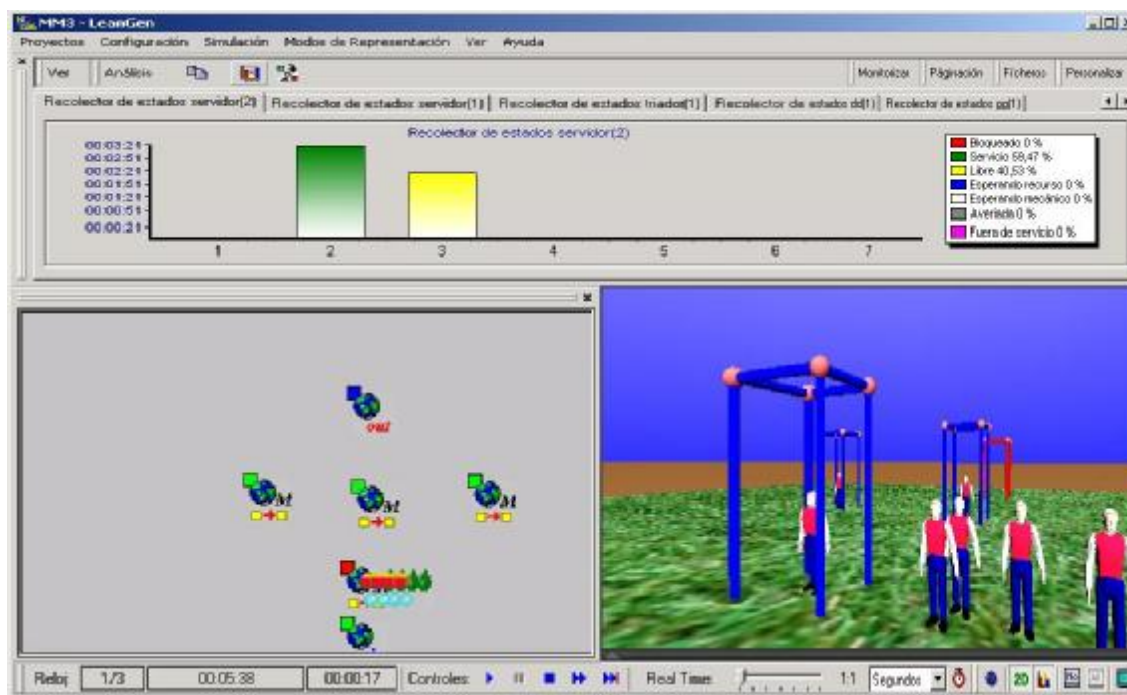
LeanEditor no ofrece un mecanismo lineal y único para la definición de los sistemas que se desean simular, el usuario puede seleccionar diferentes caminos o procedimientos para la elaboración de un modelo de simulación. La definición de modelos de simulación se basa en los siguientes cinco puntos:

- ☉ 1-Ámbito del proyecto: Define los elementos de simulación y sus propiedades por defecto, que están disponible en un determinado sector, como en los ensayos clínicos. La posibilidad de disponer de diferentes ámbitos disminuye el tiempo de desarrollo de los modelos de simulación al acortar las fases de representación y configuración del modelo, recordar que Leansim construye la representación gráfica al mismo tiempo que se genera el modelo.
- ☉ 2-Entidades y procesos: Una vez seleccionado el ámbito del proyecto es necesario identificar las entidades (Por ejemplo pacientes) y procesos que se desean simular (Por ejemplo abandonos, asignación de los tratamientos, etc). La configuración de cada una de las entidades y procesos se realiza mediante la interacción, por parte del usuario, con la ventana de edición de entidades y de procesos del programa.
- ☉ 3-Elementos del modelo: Se han de identificar los diferentes elementos de simulación que se observan en el modelo a simular. Para ello se utiliza la tabla de funcionalidades anterior (Creación de entidades que entran en el sistema, destrucción de entidades que abandonan el sistema, petición de entidades, especificación de atributos y de operaciones, etc.)
- ☉ 4-Acciones por proceso: Se han de definir los diferentes procesos donde intervienen las entidades, ya que las entidades se mueven o definen su comportamiento en el sistema según los procesos en que intervienen, entendiéndose por proceso el conjunto de las fases sucesivas de un fenómeno natural o de una operación asociada a una entidad.
- ☉ 5-Configuración general: La configuración general del modelo permite definir que elementos pueden ser modificados en tiempo de experimentación, mediante LeanGen, y el tiempo total de la simulación.

3.2.5.3 LEANGEN

LeanGen es el motor de simulación de Leansim y permite desarrollar experimentos de simulación a partir de los proyectos generados con LeanEditor. En los siguientes capítulos se analizará la interfaz gráfica de LeanGen y las diferentes funcionalidades del producto.

La interfaz gráfica que acompaña a LeanGen, se compone de un sistema de menús, diferentes barras de herramientas, áreas gráficas para la representación y análisis de experimentos:



Las diferentes funcionalidades que ofrece LeanGen son accesibles mediante la barra de menús integrada en el programa. La siguiente tabla presenta las diferentes funcionalidades del motor de simulación:

Opción de menú		Icono	Descripción funcionalidad
Proyectos	Nuevo		Creación de un nuevo proyecto a partir de un proyecto de LeanEditor
	Abrir		Abrir un proyecto generado anteriormente a partir de un ámbito
	Cerrar		Guardar el proyecto actual con el nombre establecido en la apertura
	Eliminar		Guardar el proyecto actual con un nuevo nombre
	Salir		Eliminar un proyecto de la lista de proyectos generados
Configuración	Modelo		Mostrar o Ocultar la barra de herramientas de LeanEditor
	Procesos		Mostrar o Ocultar la barra de Modelos de LeanEditor
	Estadísticos		Mostrar o Ocultar la barra de Estado de LeanEditor
Simulación	Simular		Mostrar o Ocultar la ventana de configuración de entidades
	Pausa		Mostrar o Ocultar la ventana de configuración de procesos
	Terminar		Mostrar o Ocultar la ventana de control de animaciones
	Simular sin RV		Ejecutar LeanGen, motor de simulación
	Simular hasta ...		Seleccionar el archivo gráfico que se utilizará como fondo 2D
Modos de Representación	Estadísticos		Seleccionar el archivo gráfico que se utilizará como fondo 3D
	Vista 2D		Mostrar el Visor VRML
	Mundo Virtual		Mostrar u Ocultar el decorado en realidad virtual
	Con RV		Representar elementos dentro del mundo virtual
	Sin RV		No representar elementos dentro del mundo virtual
	Pantalla Completa		Visualizar la representación a pantalla completa
Ver	Barra de Acceso rápido		Mostrar o Ocultar la ventana de configuración de objetos desde LeanGen
	Barra de Simulación		Mostrar o Ocultar la ventana de configuración del escenario base
	Barra de Representación		Activar o desactivar la representación virtual en fase de diseño
	Barra de estado		Mostrar la ayuda del programa
Ayuda	Sugerencias		Información del producto
	Temas de ayuda		Muestra el menú de ayuda del usuario
	Acerca de ...		Información relativa al producto

Las barras de herramientas permiten acceder mediante los elementos iconográficos que la componen a las principales funcionalidades de menú descritas en el capítulo anterior. LeanGen consta de tres barras de herramientas:

- ☞ Acceso rápido.
- ☞ Simulación.
- ☞ Representación.

3.2.5.4 LEANCLIENT

LeanClient es el tercer componente de LeanSim, su objetivo es distribuir la representación de una simulación en diferentes terminales permitiendo que varios usuarios puedan controlar paralelamente distintas zonas del mundo simulado.

3.2.5.4.1 Análisis de los resultados

El resultado de las simulaciones se realiza a través de LeanGen, que no es una herramienta específica de análisis de resultados, no obstante dispone de componentes gráficos que permiten un primer estudio de los resultados con los estadísticos más comunes: desviación estándar, tendencia, media o exponencial y la exportación de datos en formato de archivo de texto plano para el tratamiento de los datos obtenidos con programas específicos de análisis estadísticos.

Cada uno de los diferentes elementos de simulación, diseñados en un modelo LeanSim, puede, automáticamente, registrar una serie de valores que permiten determinar el grado de aceptación del modelo simulado. Estos valores son almacenados en un control asociado al área de representación de estadísticos y la interacción por parte del usuario con este control se analiza en los siguientes apartados.

3.3 MODELIZACIÓN DE UN ENSAYO CLÍNICO GENÉRICO (MECG) MEDIANTE LEANSIM

Tras observar los diseños utilizados por diferentes investigadores, el funcionamiento, los datos y el proceder de los ensayos clínicos recogidos en el sistema Hipócrates (*Figura 13*), referencias bibliográficas y ensayos clínicos adicionales aportados por los Laboratorios Dr. Esteve, se ha diseñado un modelo conceptual general de ensayo clínico en LeanSim. Este modelo se

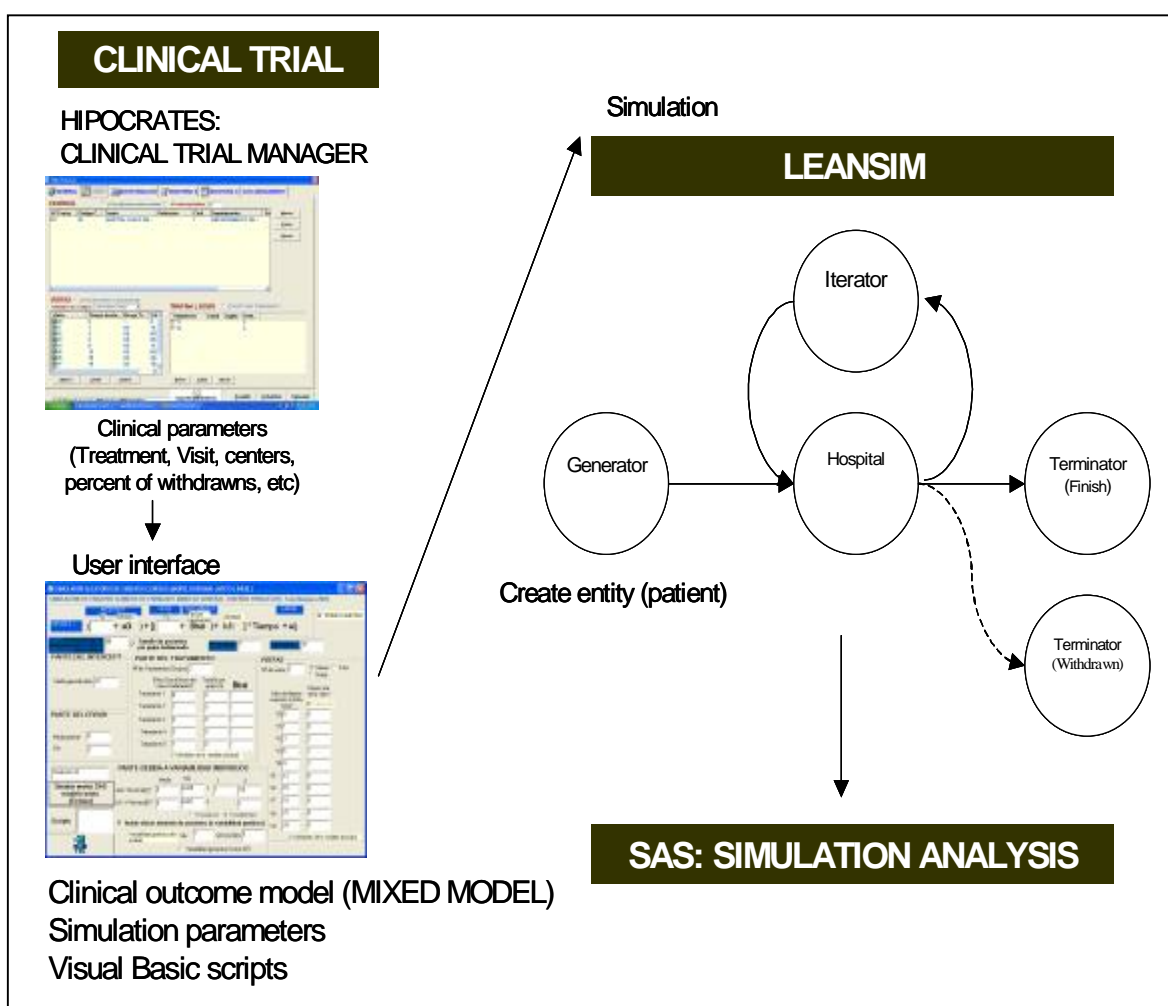


Figura 13: Diagrama de eventos de un modelo genérico de ensayos clínicos y su interacción con las especificaciones de los ensayos clínicos almacenadas en Hipócrates. El modelo conceptual en LeanSim describe el ciclo de vida de una entidad (paciente) en el ensayo clínico (de Monleón y col., 2004).

desarrolló durante 2002-2003 y se mejoró en una segunda etapa durante 2003 y 2004. Fue presentado durante el congreso de la Sociedad Española de Biometría en 2003 (Monleón y col., 2003^c) y en el congreso COMSTAT (Monleón y col., 2004). Bajo el marco conceptual que proporciona este modelo, el simulador de LeanSim no solo proporciona réplicas de simulación de la variable de respuesta del ensayo (variable principal), mediante la teoría estadística de modelos lineales, sino que proporciona otros datos de ensayo como:

- Identificación de las entidades generadas.
- Abandonos.
- Tratamiento.
- Variable de respuesta por visita de tratamiento.
- Covariantes: Sexo, edad, etc.
- Momento de generación de la entidad (según patrón predeterminado por el usuario que puede ser una función aleatoria de tiempo).

Otros modelos de ensayos clínicos basados en simulaciones discretas, modelos autoregresivos y cadenas de Markov fueron presentados en el congreso de ingeniería ENBIS por Abbas y col. (2003).

3.3.1 ACTIVIDAD ANTIRETROVIRAL DE DOS COMBINACIONES FARMACOLÓGICAS. HOSPITAL CLÍNICO. DEP FARMACOLOGÍA. 2003 (SIDA-1).

En esta sección se describe el ensayo clínico utilizado como base para diseñar el modelo genérico de simulación de ensayos clínicos.

Características:

Ensayo clínico con 155 pacientes afectados de SIDA en el Hospital Clínico de Barcelona donde se comparaba la eficacia de dos combinaciones de fármacos anti-retrovirales ((Nucleósido + inhibidor de proteasa) y (Nucleósidos + no

nucleósidos)) para disminuir la carga viral de los pacientes durante 2 años. Los tratamientos fueron suministrados de manera aleatoria y abierta a los pacientes y se midieron los niveles de carga viral (copias/ml) cada 3 meses (10 visitas). La variable de respuesta se transformó en logarítmica con el propósito de normalizar su comportamiento. Un aspecto destacable de estos estudios es la altísima tasa de abandonos por parte de los pacientes (80%).

Las cargas virales reales por combinación farmacológica fueron las que se presentan en la **Figura 14**.

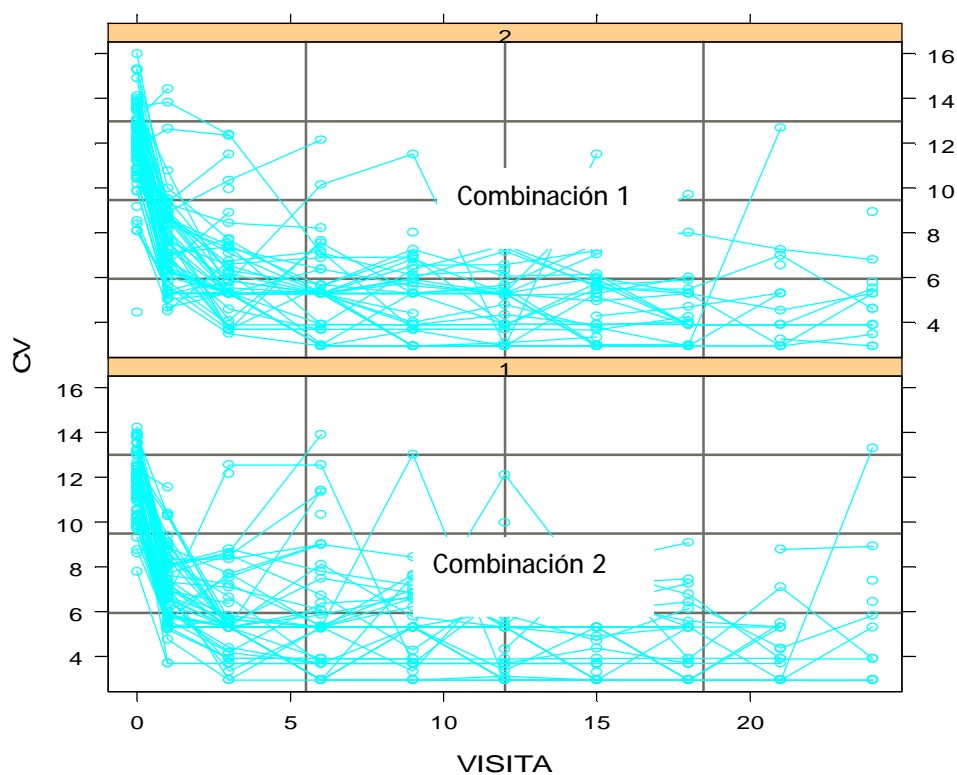


Figura 14: Evolución de la carga viral (CV) de los pacientes durante los 2 años de duración del ensayo clínico de SIDA. Se ha separado el gráfico por tratamiento.

3.3.2 VERSIÓN INICIAL

La *Figura 15* refleja la lógica del modelo asociado al estudio SIDA-1, algunos de cuyos resultados de simulación se presentan posteriormente en la sección de experimentos retrospectivos. El diagrama está compuesto de 4 objetos de simulación: un generador o inicializador de entidades, dos terminadores o destructores de entidades y un iterador. Por otra parte se pueden ver dos archivos XML (“Assaig” y “Visita”) que parametrizan el experimento a realizar.

Como se puede observar, el modelo presenta un diseño muy sencillo, la complejidad de los estudios reside en el objeto Iterador. El modelo describe el ciclo de vida de un sujeto (paciente) que se somete a un tratamiento que es asignado aleatoriamente durante el ensayo clínico y se recogen las medidas de la variable carga viral a lo largo de una serie de visitas (modelo de medidas repetidas):

- Generación de la entidad y asignación de sus propiedades o atributos¹ (1).
- Transferencia al objeto iterador (2), donde se calcula la carga viral del paciente en cada visita. Éste puede abandonar el ensayo en función de una función probabilística, haciendo un salto al proceso de Finalizaciones (3).
- Registro de los sujetos que abandonan el tratamiento (3).
- Registro de los pacientes que abandonan el tratamiento (4).

La duración de la simulación viene condicionada por el comportamiento del objeto de simulación CleanIteratorMachine, en concreto por el tiempo asociado a la última visita programada por el último sujeto implicado en el estudio de simulación.

¹ **Sexo:** Hombre (0) o Mujer (1); **Tener SIDA:** No (0) o Sí (1); **Edad:** Dentro del rango 20-40 [20-40] años; **Tratamiento:** Combinación T₁ (0) o T₂ (1)

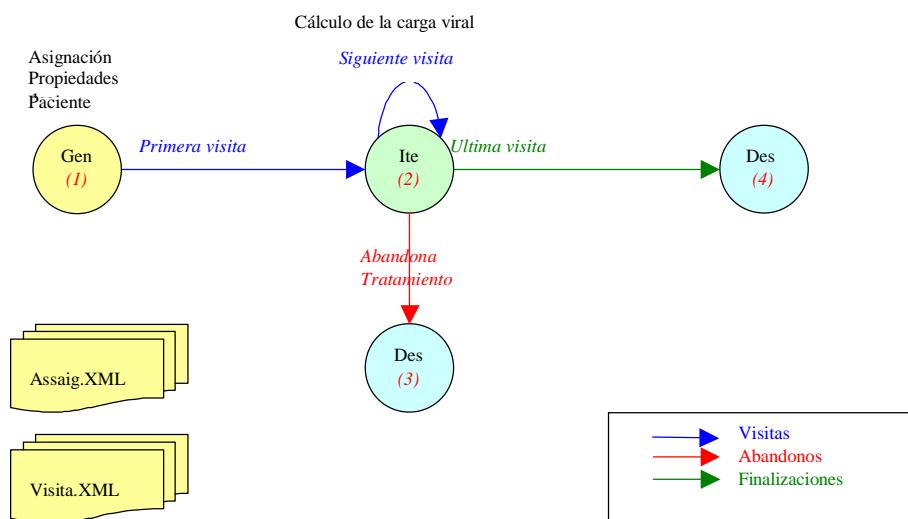


Figura 15: Lógica del modelo asociado al estudio de simulación del ensayo clínico SIDA-1.

En el modelo de simulación cada paciente corresponde a un proceso. Cada nuevo paciente generado se asigna aleatoriamente a uno de los tratamientos y queda caracterizado por un conjunto de parámetros correspondiente a los parámetros fijos del modelo mixto descrito anteriormente, a los que se suman los parámetros aleatorios generados de acuerdo con las distribuciones indicadas más arriba (b_0 , b_1 y b_2), en concreto de acuerdo con una distribución normal multivariante con vector de medias nulo y la matriz de covarianzas S . Estos parámetros así obtenidos caracterizan a aquel individuo en toda la simulación. Una vez creado, cada individuo (proceso) se ejecuta en paralelo y pasa por las diferentes visitas previstas o, con cierta probabilidad, abandona definitivamente o se salta alguna de las visitas. En cada visita se reevalúa el modelo lineal. Esto proporciona la predicción de carga vírica para cada visita e instante concreto, a la cual se añade un residuo generado aleatoriamente generando así la carga vírica observada.

Formato de los archivos asociados a la simulación:

Existen 2 archivos en formato XML que permiten parametrizar el modelo de simulación:

Assaig.xml

Visita.xml

Fichero Assaig.xml

Este archivo recoge la información relacionada con las propiedades generales del ensayo clínico y que tienen que ver con la duración final del estudio de simulación:

- Calendario asociado al ensayo: inicio y fin de ensayo, días festivos.
- Tamaño de la muestra asociado al ensayo.
- Otros parámetros a determinar.

Fichero Visita.xml

Este archivo recoge toda la información relacionada con un tratamiento asociado a un paciente en un centro hospitalario. Es un recolector de visitas y para cada una de ellas se ha considerado necesario recoger la siguiente información:

- § El elemento *Descripción* de tipo *TipusCadena*, que contiene una cadena que describe la vista.
- § El elemento *Función* de tipo *TipusCadena*, que contiene la función de ejecución de la vista.
- § El elemento *Temps* de tipo *TipusDouble*, que contiene el tiempo del siguiente evento.
- § El elemento *Atribut* de tipo *TipusCadena*, que contiene el identificador del atributo de la entidad a la cual se asignará la función.

§ El elemento *Salts*, contiene un conjunto de elementos *Salt* donde cada uno de ellos contiene:

- ∅ El elemento *ProcesSalt* de tipo *IdProces*, que contiene el identificador del proceso que irá a parar a la entidad.
- ∅ El atributo *Probabilitat* de tipo *TipusDouble*, que contiene la probabilidad de realizar el salto.
- ∅ El atributo *Temps* de tipo *TipusDouble*, que contiene el tiempo que se tarda en realizar un salto.

En la *Figura 16* se muestra la descripción gráfica del elemento *Visites*:

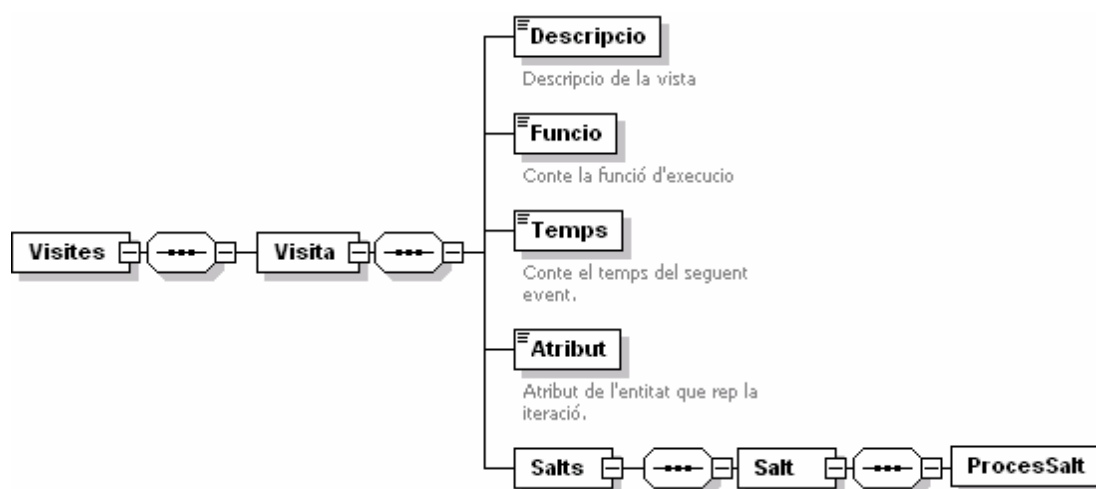


Figura 16: Esquema XML del elemento Visitas.

Un ejemplo del fichero *Visita.XML* es el siguiente:

```

<Visites xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Projectes\Petri\Visites.xsd">
  <Visita>
    <Descripcio> La visita traumatoleg </Descripcio>
    <Funcio> Funcio1 </Funcio>
    <Temps>3.14159</Temps>
    <Atribut> GlobulsBlancs </Atribut>
    <Salts>
  
```

```

<Salt Probabilitat="0.25" Temps="1.222">
  <ProcesSalt IdProces="Primer" IdEntitat="1"/>
</Salt>
<Salt Probabilitat="0.75" Temps="0.222">
  <ProcesSalt IdProces="Segon" IdEntitat="1"/>
</Salt>
</Salts>
</Visita>
<Visita>
  <Descripcio> Visita a medicina General </Descripcio>
  <Funcio> Funcio2 </Funcio>
  <Temps>3.242</Temps>
  <Atribut> TempsTractament </Atribut>
  <Salts>
    <Salt Probabilitat="0.25" Temps="3.222">
      <ProcesSalt IdProces="Primer" IdEntitat="2"/>
    </Salt>
    <Salt Probabilitat="0.15" Temps="0.222">
      <ProcesSalt IdProces="Segon" IdEntitat="2"/>
    </Salt>
    <Salt Probabilitat="0.45" Temps="2.222">
      <ProcesSalt IdProces="Tercer" IdEntitat="2"/>
    </Salt>
  </Salts>
</Visita>
</Visites>

```

Como se puede ver, la estructura del XML permite un grado de definición muy por encima de lo realmente esperado para el estudio del modelo propuesto. En el estudio propuesto una visita a un centro hospitalario seguiría la siguiente codificación:

```

<Visita>
  <Descripcio> Sessió de control nº5 </Descripcio>
  <Funcio> CV</Funcio>
  <Temps>30</Temps>
  <Atribut> Index_5_CV</Atribut>
  <Salts>
    <Salt Probabilitat="0.25" Temps="0 ">
      <ProcesSalt IdProces="Abandonament" IdEntitat="Pacient"/>
    </Salt>
  </Salts>
</Visita>

```

La lectura que se haría de la visita sería la siguiente: Visita de un paciente para realizar la quinta sesión de control <Descripcio> donde se mide la carga viral <Funcio> y se registra en la ficha del paciente dentro del campo Index_5_CV <Atribut>. Una vez finalizada la visita, un 25% de los pacientes <Probabilitat de salt> abandonan el ensayo, y al resto se les programa la visita para dentro de 30 días <Temps>. Una vez finalizada la última de las visitas, contenidas en el archivo XML, el paciente abandona el sistema y finaliza.

Se han implementado una serie de opciones que permiten escribir los ficheros XML a partir de los datos del diseño del experimento contenido en el programa Hipócrates pasando de esta manera la información de los ensayos reales a LeanSim para validar que éste realice bien las simulaciones.

INCONVENIENTES DE ESTA VERSIÓN DEL MODELO

El principal inconveniente de esta versión de MECG es la excesiva rigidez en la especificación del modelo de generación de la variable principal, en este caso la carga viral (1), ya que un cambio de función o de parámetros implica una costosa modificación del código C++ del simulador, y recompilación de éste.

Otro inconveniente es la enorme dificultad que tienen los investigadores biomédicos para utilizar un simulador con la complejidad del construido, ya que la parametrización y generalización de todos los casos y funciones es prácticamente imposible. Para ello se ha propuesto la utilización de una interfaz de usuario que traslade las peticiones de simulación y los parámetros desde el investigador biomédico al simulador.

3.3.3 VERSIÓN MODIFICADA DEL MODELO SIMULADOR MECG

A partir de la experiencia con la primera versión de simulador de ensayos clínicos de 2002-2003, se propuso un cambio en el modelo genérico de ensayo clínico y en su parametrización. Durante 2003-2004 se ha realizado un segundo modelo (*Figura 17*) que flexibiliza y mejora el diseño de las simulaciones y que convierte a LeanSim en una herramienta capaz de simular cualquier tipo de ensayo clínico.

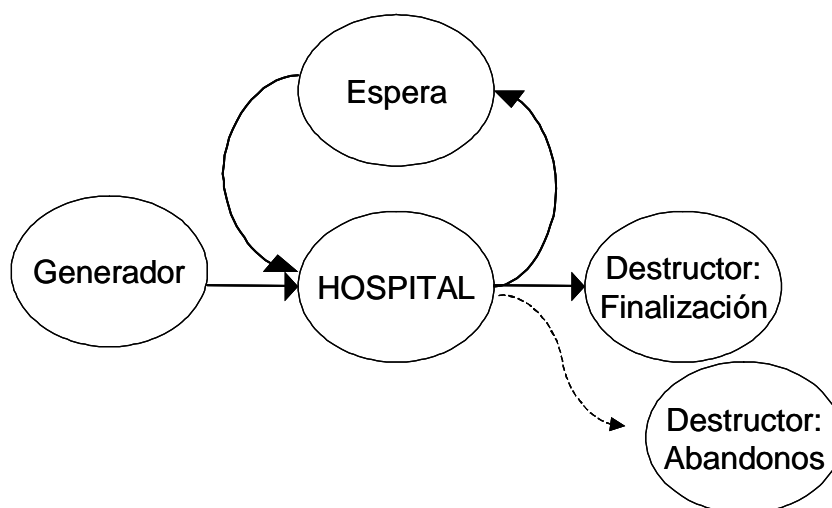


Figura 17: Lógica de la 2ª versión del modelo general de ensayo clínico utilizado en los estudios de simulación clínica.

Particularidades del nuevo diseño:

La especificación del cálculo de la fórmula de las variables principales se puede realizar ahora de forma mucho más flexible, con un lenguaje de alto nivel (VBScript).

La espera entre visita y visita se realiza en otra máquina, antes era en la misma máquina.

No existe la máquina de abandono.

El tiempo entre visitas se calcula durante la visita y se almacena en el atributo "temps" del paciente. Este tiempo se expresa en segundos.

¿Qué se puede parametrizar en este modelo y cómo?

Este modelo permite parametrizar el sistema, utilizando un conjunto de "scripts" (actualmente 4, pero se pueden ampliar a 2 más) que se ejecutan sobre una máquina virtual dentro del motor de simulación. De estos 4 "scripts", 3 hacen referencia a la máquina *Centro* y el cuarto a la máquina de *Espera*.

Las operaciones que se llevan a cabo en cada una de estas máquinas y los "scripts" que se ejecutan en cada una de ellas se describen a continuación.

Máquina Centro

En ella se ejecuta la operación señalada por el diagrama de flujo de la **Figura 18**.

Con esta operación se busca un comportamiento similar al que supone la ejecución en un lenguaje de alto nivel de una estructura del tipo:

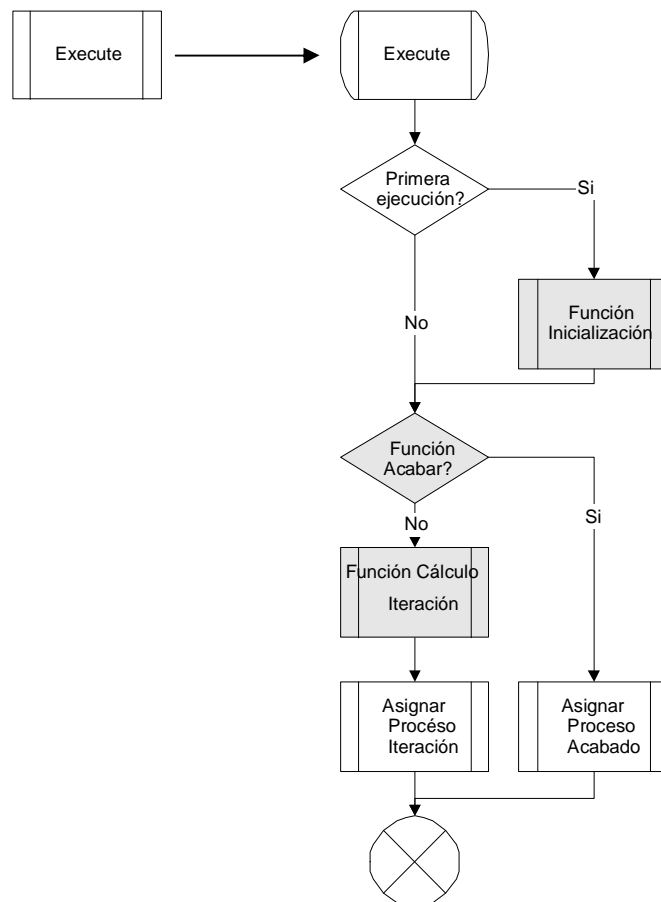


Figura 18: Diagrama de flujo de la máquina centro.

```

[inicilalitzación]
mientras [acabar] hacer
    [cuerpo]
fmientras
  
```

Se puede observar la ejecución de los 3 "scripts" implicados (marcados en gris en el diagrama SDL ("description language standardized") y con corchetes en el código anterior). Los "scripts" de "inicialización" y "cuerpo" son ejecutados y el "script acabado" es interpretado.

Máquina Espera

Esta máquina representa el hecho de esperar. El "script" que se ejecuta en esta máquina ha de retornar el tiempo (en días) que el paciente tiene que esperar para ir a una nueva visita. Este "script" es evaluado o interpretado.

Escritura de "scripts"

Es necesario diferenciar dos tipos de "scripts":

- **"Scripts" ejecutados.** Por ejemplo y al igual que en el primer simulador se presenta la ejecución de una simulación del modelo SIDA-1 que posteriormente será comentada:

```
Dim b

b = Simulator.GetUniform(0, 1)
if (b < 0.75) then
    CurrentEntity.SetAttributeValue "tesida", 0
else
    CurrentEntity.SetAttributeValue "tesida", 1
end if
```

- **"Scripts" interpretados o evaluados.** Por ejemplo:

```
10*CurrentEntity.GetAttributeValue "tesida"
```

La diferencia entre unos y otros es que el segundo tipo de "scripts" retorna un cierto valor al acabar su ejecución. El primer "script" sería erróneo como "script" evaluado, y viceversa.

Como se puede ver en los ejemplos, se puede establecer una comunicación entre la aplicación y el "script" que se realiza mediante "call-backs" desde el "script", que entiende su motor de ejecución y permite la interacción desde la aplicación principal.

Para permitir al “script” interactuar con la aplicación, se han definido dos espacios de nombres que permiten introducir y recibir valores hacia y desde el motor de simulación: **Simulator** y **CurrentEntity**.

Las propiedades que se ejecutan sobre estos espacios de nombres son las siguientes:

Simulator

<p>Simulator.GetCurrentTime() : retorna double Obtiene el tiempo actual del reloj de simulación.</p>
<p>Simulator.GetUniform(double a, double b) : retorna double Obtiene un valor aleatorio siguiendo una variable aleatoria uniforme entre a y b.</p>
<p>Simulator.GetNormal(double mu, double sigma) : retorna double Obtiene un valor aleatorio siguiendo una variable aleatoria normal de parámetros mu i sigma.</p>

En un futuro se implementarán otras distribuciones tales como exponencial, Poisson, Bernuilli, Weibull, Erlang, etc.

CurrentEntity

<p>CurrentEntity.SetAttributeValue (BSTR variable, double valor) Asigna el valor <i>valor</i> a la variable <i>variable</i> de la entidad actual. Si se asigna un valor a un atributo que no existe, este se crea.</p>
<p>CurrentEntity.SetAttributeString (BSTR variable, BSTR valor) Igual que el anterior pero se asigna a la entidad un atributo de tipo String.</p>
<p>CurrentEntity.GetAttributeValue (BSTR variable) : retorna double Obtiene el valor de la variable <i>variable</i> de la entidad. Si se lee un valor de un atributo que no existe el resultado leído será 0.</p>

Los dos últimos "scripts" realizan las siguientes funciones:

- Número de pacientes a simular.
- Tiempo entre generación y generación de pacientes (para simular el tiempo de inclusión), parecido al "script" de espera.
- Pueden implementarse también rutinas desde el "script" de inicialización o iteración como pueden ser:

Function Suma(a, b)

Suma = a + b

End Function

CurrentEntity.SetAttributeValue "suma", Suma(3, 4)

3.3.3.1 EJEMPLO SIDA-1

Para mostrar cómo se pueden aplicar todas estas funciones a la simulación del modelo de ensayo clínico de SIDA-1 se presenta el siguiente desarrollo. La base del modelo conceptual es el siguiente modelo lineal mixto:

$$\text{LogCV} = (a_1 + b_{0i}) + ((k - 1)a_2 + b_{1i}) + \text{Visita}_{ij} (a_3 + b_{2i}) + e_{ij}$$

Dónde los parámetros se definen de la manera siguiente:

- a_1, a_2, a_3 : constantes.
- k : tratamiento actual que sigue el paciente
- $b_{0i}, b_{1i}, b_{2i}, e_{ij}$: variables aleatorias que se distribuyen según una distribución Normal, cada una con sus respectivos parámetros m y s .
- Visita_{ij} : Tiempo (en meses) que el paciente lleva realizando el ensayo.

Los "scripts" resultantes que leerá el simulador de LeanSim son los siguientes:

1-FuncioInicialitzacio.vbs

```
Dim b
Dim edat

b = Simulator.GetUniform(0, 1)
If (b < 0.72) then
    'Homes
    CurrentEntity.SetAttributeValue "sexe", 0
    edat = Simulator.GetUniform(0, 1)
    edat = 40 + edat*20
    CurrentEntity.SetAttributeValue "edat", edat
    CurrentEntity.SetAttributeValue "pes", edat
else
    'Dones
    CurrentEntity.SetAttributeValue "sexe", 1
    edat = Simulator.GetUniform(0, 1)
    edat = 10 + edat*10
    CurrentEntity.SetAttributeValue "edat", edat
    CurrentEntity.SetAttributeValue "pes", edat
end if

b = Simulator.GetUniform(0, 1)
if (b < 0.5) then
    CurrentEntity.SetAttributeValue "tractament", 0
    CurrentEntity.SetAttributeString "forma", "HumanResource.wrl"
else
    CurrentEntity.SetAttributeValue "tractament", 1
    CurrentEntity.SetAttributeString "forma",
"HumanResourceBlue.wrl"
end if

b = Simulator.GetUniform(0, 1)
if (b < 0.75) then
    CurrentEntity.SetAttributeValue "tesida", 0
else
    CurrentEntity.SetAttributeValue "tesida", 1
end if

CurrentEntity.SetAttributeValue "visita", 0

CurrentEntity.SetAttributeValue "b0i", Simulator.GetNormal(0, 0.618)
CurrentEntity.SetAttributeValue "b1i", Simulator.GetNormal(0, 0.197)
CurrentEntity.SetAttributeValue "b2i", Simulator.GetNormal(0, 0.236)

CurrentEntity.SetAttributeValue "generateTime",
Simulator.CurrentTime()
```

2-FuncioCondicio.vbs: Controla que se realicen 8 visitas.

```
CurrentEntity.GetAttributeValue("visita") < 9
```

3-FuncioIteracio.vbs

```
Dim numVisita
numVisita = CurrentEntity.GetAttributeValue("visita")

Dim segonsXmes
Dim tt
\ .....
```

En este "script" se calcula el tiempo entre visitas mediante la variable "temps". Este, en la primera visita es de un mes, en la segunda visita es de dos meses y de tres meses para el resto de visitas, tal como muestra el código del "script". Además se han añadido unas variaciones aleatorias según una Normal(0, 0.5). La variable "segonsXmes" sirve para hacer la conversión de tiempo a segundos.

```
\ .....
```

```
Dim temps
tt = Simulator.GetNormal(0, 0.5)

segonsXmes = 365.25*24*60*60/12
If (numVisita = 0) Then
    temps = segonsXmes*1
ElseIf (numVisita = 1) Then
    temps = segonsXmes*2+segonsXmes*tt
Else
    temps = segonsXmes*3+segonsXmes*tt
End If

If temps < 0 Then temps = 0 End If
CurrentEntity.SetAttributeValue "temps", temps

'Cálculo de parámetros aleatorios de los pacientes:
Dim b0i
Dim bli
Dim b2i
b0i = CurrentEntity.GetAttributeValue("b0i")
bli = CurrentEntity.GetAttributeValue("bli")
b2i = CurrentEntity.GetAttributeValue("b2i")
```

Esta es la manera de calcular la variable "Visita_ij". Por otra parte el paciente tiene un atributo "generateTime" que se ha almacenado es el

"script" de inicialización. Sólo es necesario recoger el tiempo y restarle éste y convertirlo en meses.

```

Dim currentTime
Dim initialTime
Dim totalTime
Dim Visitaij
currentTime = Simulator.CurrentTime()
initialTime = CurrentEntity.GetAttributeValue("generateTime")
totalTime = currentTime - initialTime
Visitaij = totalTime / 365.25*12

dim k
k = CurrentEntity.GetAttributeValue("tractament")

dim eij
eij = Simulator.GetNormal(0, 2.52)

dim logCV
logCV = (8.34 + b0i)+((k-1)*0.13 + b1i)+(-0.24 + b2i) * Visitaij + eij

dim nom

nom = "logCV"
nom = nom & numVisita
CurrentEntity.SetAttributeValue nom, logCV

numVisita = numVisita + 1
CurrentEntity.SetAttributeValue "visita", numVisita

```

4-FuncioTempsSalt.vbs

Este "script" hace referencia a la máquina *Espera* y es evaluado. Se ha de apuntar que la asignación del valor correspondiente de la variable *temps* se realiza en el cuerpo de la iteración.

```
CurrentEntity.GetAttributeValue("temps")
```

Este nuevo modelo ha perdido la parametrización de ciertos parámetros con respecto al anterior. De todas maneras no es difícil reintroducirlos utilizando la nueva filosofía de los "scripts". Estos parámetros son los siguientes:

- Abandonos: Ya no existe la máquina de abandonos. Se pueden calcular los abandonos añadiendo al "script" *FuncioIteracio.vbs* un fragmento que calcule una función de probabilidades uniforme y que decida abandonar el paciente en función de la visita. El resultado de un abandono se puede almacenar en un atributo de la entidad paciente.

Se prevee que a finales de 2005 se disponga de una versión reducida de LeamSim, actualmente desarrollándose, que simplifique la especificación del modelo de simulación.

3.3.3.2 ESPECIFICACIÓN DEL MODELO DE SIMULACIÓN EN PSEUDO-LENGUAJE DE PROGRAMACIÓN SAS

Dada la fuerte carga de conceptos estadísticos y de cálculo estadístico asociada a las simulaciones realizadas en esta memoria, a lo largo de la misma también se evalúa la posibilidad de utilizar una herramienta no específicamente de simulación pero con amplias facilidades estadísticas, como el lenguaje de programación del paquete estadístico SAS, para la simulación de ensayos clínicos. La especificación del modelo de simulación en lenguaje SAS del ejemplo de SIDA desarrollado, corresponde al siguiente algoritmo:

```

data SIMULACION;
DO REPLICA=1 TO 30;
drug="Tratam0";
TRATAM=1;
coef_ab = 0.20;
do NUM_PAC=1 to 70;
  abandona=0;
  b1=0.151*rannor(0);
  b2= 0.119 *rannor(0);
  *mes 0;
  do visita=0 to 1 by 1;
    eij = 2.543 *rannor(0);
    log_CV = 8.2535 +((TRATAM-1)*0.1813 + b2)+(-0.2395 + b2) * visita + eij;
    *abandonos;
    ABAND=uniform(0);
    if visita=1 AND (ABAND <= coef_ab or abandona = 1) then abandona=1;
    if abandona = 1 then LOG_CV = .;
    output;
  end;
  *resto de visitas;
  do visita=3 to 24 by 3;
    eij = 2.543 *rannor(0);
    log_CV = 8.2535 +((TRATAM-1)*0.1813 + b2)+(-0.2395 + b2) * visita + eij;
    *abandonos;
    ABAND=uniform(0);
    if ABAND <= coef_ab or abandona = 1 then abandona=1;
    if abandona = 1 then LOG_CV = .;

  output;

  end;
end;

drug="Tratam1";
TRATAM=2;
do NUM_PAC=1 to 70;
  abandona=0;
  b1=0.151*rannor(0);
  b2= 0.119 *rannor(0);
  *mes 0;
  do visita=0 to 1 by 1;
    eij = 2.543 *rannor(0);
    log_CV = 8.2535 +((TRATAM-1)*0.1813 + b2)+(-0.2395 + b2) * visita + eij;
    *abandonos;
    ABAND=uniform(0);
    if visita=1 AND (ABAND <= coef_ab or abandona = 1) then abandona=1;
    if abandona = 1 then LOG_CV = .;
    output;
  end;
  *resto de visitas;
  do visita=3 to 24 by 3;
    eij = 2.543 *rannor(0);
    log_CV = 8.2535 +((TRATAM-1)*0.1813 + b2)+(-0.2395 + b2) * visita + eij;
    *abandonos;
    ABAND=uniform(0);
    if ABAND <= coef_ab or abandona = 1 then abandona=1;
    if abandona = 1 then LOG_CV = .;

  output;

  end;
end;
end;
run;

```

Como puede observarse, la especificación del modelo se ha realizado siguiendo el algoritmo genérico de simulación discreta, descrito en el apartado 3.1.3. El modelo de simulación, aunque no es exactamente igual al señalado anteriormente en el entorno LeanSim, es más sencillo en SAS que en LeanSim, dado el propósito estadístico de la simulación que se pretendía, aunque resultaría más complejo si la cantidad de factores y operaciones a realizar durante la simulación creciera (colas de espera en los hospitales por sólo poder atender a una cantidad limitada de pacientes en un periodo de tiempo, periodos de vacaciones de los médicos, diversos centros, tal vez especializados en determinados tipos de pacientes, etc).

3.3.4 INTERFAZ DE USUARIO PARA GENERAR EL MODELO DE LA VARIABLE PRINCIPAL DEL ENSAYO CLÍNICO SEGÚN MODELOS LINEALES MIXTOS

Con el fin de facilitar el trabajo, a los usuarios biomédicos no expertos en simulación ni en estadística, de construir los modelos de la variable principal

Figura 19: Interfaz de usuario que genera scrips de Visual Basic para simular modelos en LeanSim de modelos lineales mixtos (IMLM).

de los ensayos clínicos a simular, se ha desarrollado una interfaz de usuario para modelos lineales mixtos (IMLM) que contiene los factores principales (tratamiento, paciente y visita) de los modelos de ensayos clínicos de respuesta continua, paralelos y de medidas repetidas estándar.

La interfaz puede verse en funcionamiento en la **Figura 19** y su base conceptual en la **Figura 20**.

Este interfaz, una vez introducidos los parámetros del modelo los transforma en unos ficheros de "scripts" de Visual Basic 6, que son interpretados por el simulador LeanSim o bien utilizados en las simulaciones con lenguaje SAS.

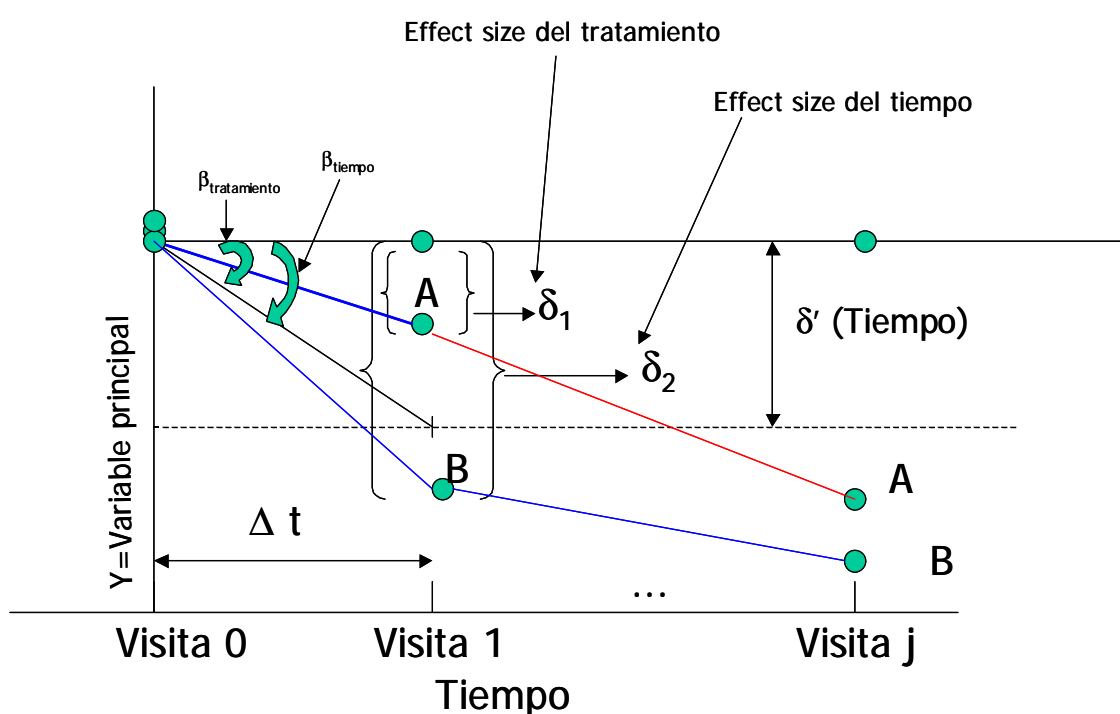


Figura 20: Esquema de funcionamiento de interfaz de usuario para el cálculo de parámetros de un modelo lineal mixto sencillo de un ensayo clínico.

Este programa de interfaz de usuario ha sido realizado en Visual Basic 6. Se incluye una versión ejecutable de la misma en el CD anexo.

3.4 VALIDACIÓN, VERIFICACIÓN Y ACREDITACIÓN (VV&A) DE LOS MODELOS DE SIMULACIÓN

3.4.1 INTRODUCCIÓN

Como se ha comentado en el capítulo introductorio de esta memoria, los modelos de simulación están cada vez más presentes en la toma de decisiones. El grado de acierto de estas decisiones depende, lógicamente, de si el modelo y sus resultados son "correctos" y "útiles". Es necesario validar y verificar los modelos de simulación, especialmente en cuestiones tan críticas como la optimización de futuras pruebas con pacientes. Esta cuestión se ha abordado desde múltiples ópticas, por ejemplo en Robinson (1997) quien distingue un modelo conceptual de un modelo computacional y separa el proceso de verificación del proceso de validación. En esta sección se abordará básicamente la óptica estadística, aunque se expone sucintamente el planteamiento del problema y las múltiples visiones de cómo abordarlo.

En una primera aproximación, demasiado simplista, desde el punto de vista de la diferencia entre los datos reales y los simulados, el mejor modelo es aquel que proporciona un mejor ajuste. Pero es prácticamente imposible que un modelo reproduzca exactamente los datos reales, siendo necesario establecer el modelo más simple que explique al máximo los datos reales y que sea aceptado por los expertos en el tema.

En trabajos como Sargent (2003) o Nance y Sargent (2002) se realiza una amplia recopilación de los métodos de validación, verificación y acreditación de los modelos de simulación. Nos basaremos en ellos, en el presente capítulo, como base de una revisión general de estas técnicas.

El modelo conceptual describe la representación lógico-matemática de la realidad que se quiere modelizar (Barceló, 1996). En esta tesis la mayoría de

modelos conceptuales corresponden a modelos mixtos que representan la variación en el tiempo de un indicador de la respuesta al tratamiento farmacológico experimental administrado a los pacientes durante el proceso del ensayo clínico, en función de una serie de factores como el tipo de tratamiento, edad, sexo, etc.

El modelo computacional es la plasmación del modelo conceptual en forma de un programa informático, en esta tesis correspondería a los "scripts" de LeanSim o a los ficheros de sintaxis de SAS. El modelo conceptual de ensayo clínico general que se presenta es el descrito como "Ensayo clínico genérico o MCEG" en los capítulos anteriores.

La validación se referiría al proceso (o conjunto de procesos) encaminado a establecer que se ha obtenido el modelo correcto, una adecuada representación del sistema a estudiar. El modelo computacional ha de poseer una precisión adecuada según el propósito con el que ha sido construido (Schlesinger y col., 1979). En el caso de los ensayos clínicos correspondería a estimar una respuesta al tratamiento y simular todos los eventos que se producen: iniciación, asignación a un tratamiento, finalización, abandono, etc. Esta definición de validación es la adoptada en la presente memoria.

La validación denominada "Black-box" indica que los resultados del modelo deben ser parecidos a los del mundo real cuando las condiciones de entrada (factores) son similares. Esta comparación puede hacerse visualmente.

La validación puede ser retrospectiva o prospectiva en función de la fuente de datos con los que se ha estimado el modelo de simulación. Según Holford y col. (2000) la validación de modelos retrospectivos tiene que venir completada por una validación prospectiva. En el caso de la validación retrospectiva, la mejor manera de comparar los datos simulados con los datos reales es mediante intervalos de confianza (Law y Kelton, 1991; Barceló, 1996 y Guasch y col., 2002), intentando que los modelos computacionales

presenten intervalos parecidos a los de la realidad y del modelo estadístico estimado. En el caso de modelos de simulación prospectivos no es posible compararlos con datos reales, ya que no se dispone aún de éstos, pero sí pueden aproximarse con datos de ensayos clínicos retrospectivos semejantes, estableciendo intervalos de confianza de los parámetros del modelo, así como analizando los residuos del mismo.

Para validar los modelos de simulación con datos retrospectivos y poder comparar los resultados con los resultados ya obtenidos, existen algunos métodos estadísticos específicos. Kleijnen (1999), Kleijnen y Deflandre (2005) y Whitner y Balci (1989) revisan las técnicas estadísticas comúnmente utilizadas en la validación de modelos de simulación. Reynolds y Deaton (1982) presentan una revisión de estas técnicas desde el punto de vista del contraste de hipótesis. Los trabajos de Wang (1988^{a,b}, 1990^{a,b}, 1991, 1992^{a,b}, 1994), Wang y Graybill (1981) y Wang y Iyer (1994), desde un punto de vista de los intervalos de confianza, serían aplicables a las simulaciones descritas en la presente memoria y por ello se hace más énfasis a esta metodología a lo largo del apartado correspondiente a los intervalos de confianza. Kleijnen y col. (2004) es una revisión actualizada de la problemática del diseño de experimentos de simulación.

Por verificación se entendería el proceso (o conjunto de procesos) encaminado a establecer que la lógica operacional del modelo computacional se corresponda adecuadamente con la lógica del modelo conceptual, es decir, si el programa informático que lo representa funciona adecuadamente. En el caso de los ensayos clínicos, el modelo computacional debe funcionar correctamente y cumplir correctamente todos los pasos detallados en el modelo conceptual. Se tendrá que comprobar si tras la ejecución del modelo se producen unos resultados razonables. Habitualmente se desarrollarán varias versiones del modelo para obtener un modelo válido satisfactorio. Durante cada iteración del modelo, se realizará una validación y una verificación, en los casos en que sea posible.

Finalmente, si un modelo satisface las especificaciones preestablecidas se habla de acreditación del modelo. Un concepto relacionado es el de la credibilidad, que corresponde al desarrollo de la confianza de los usuarios del modelo en el uso del mismo y de la información que de él se derive.

Los modelos se desarrollan para un propósito o aplicación específica y su validez viene determinada con respecto a ese propósito. Si el propósito del modelo es contestar a determinadas cuestiones, la validez del modelo deberá determinarse para cada una de las cuestiones.

Un modelo puede ser válido para un conjunto de condiciones experimentales y no válido en otro. En cualquier caso sólo se considerará válido si la precisión del modelo está dentro de un rango de precisión aceptable, fijada según el propósito del modelo planteado inicialmente, lo que demandará que se identifiquen las variables de respuesta del modelo ("outputs") y su rango de precisión, previamente a la simulación o bien en las fases más iniciales de la misma.

Si las variables "output" de interés son variables aleatorias, como es habitual en los ensayos clínicos (respuesta al tratamiento, eficacia, seguridad...), se usarán las propiedades y funciones de aquellas, tales como la media y la varianza, para validar el modelo.

Es habitual construir diferentes versiones de los modelos para obtener un modelo satisfactoriamente válido, pero suele ser muy costoso en cuanto a tiempo y recursos obtener un modelo absolutamente válido. Existen algunas técnicas y pruebas estadísticas que permiten determinar la confianza del modelo en función de la precisión requerida. Así, si la prueba determina que el modelo no presenta suficiente precisión para un determinado conjunto de condiciones experimentales, el modelo se considerará no válido, pero si se

determina que es válido no existe garantía que lo sea siempre, en todo su dominio de aplicación (ver apartado de pruebas de hipótesis).

La relación entre el coste económico/cantidad de tiempo para realizar una validación del modelo y la "confiabilidad" del modelo para el usuario, puede verse en la *Figura 21*. A medida que se incrementa la confianza en el modelo, se incrementa también su coste económico y la cantidad de tiempo empleada.

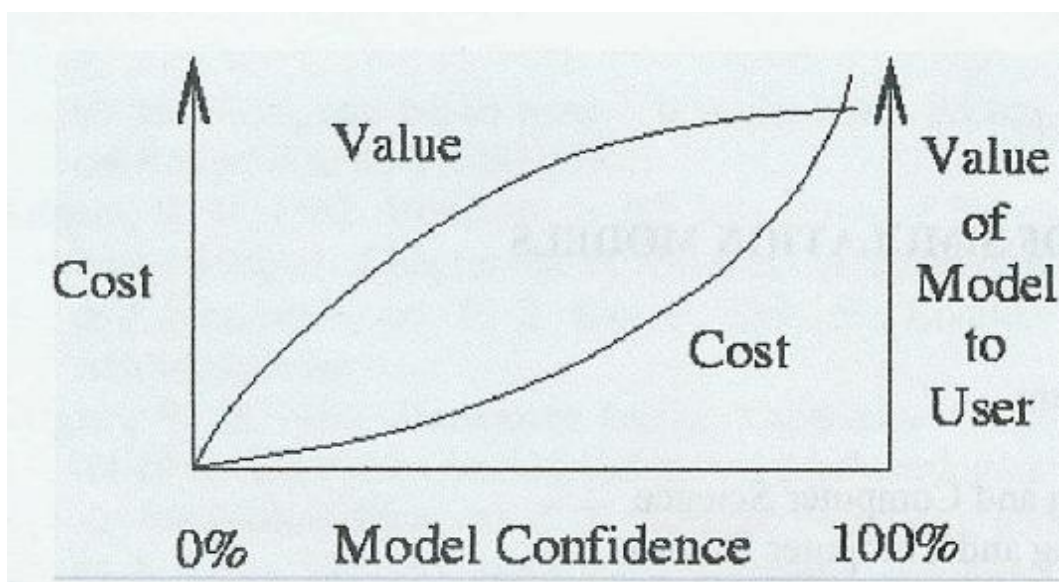


Figura 21: Confianza del modelo versus coste de tiempo y coste económico (de Sargent, 2003).

Para decidir si un modelo es válido o no, existen 4 aproximaciones en función de quién las utiliza:

- El equipo que realiza la simulación decide si el modelo es válido o no. Es una decisión subjetiva basada en los resultados de pruebas y evaluaciones que se consideran como parte del proceso de desarrollo del modelo.

- Si el equipo que desarrolla la simulación es pequeño, es mejor que el/los usuario/s final/es del modelo de simulación determinen su validez. Esta aproximación aumenta la credibilidad del modelo.
- Utilización de una tercera persona o equipo, independiente tanto del equipo de desarrollo como del usuario final y que realizará el denominado proceso de “verificación y validación independiente” (IVV). Esta aproximación sólo se suele utilizar en aquellos casos en los que se realizan modelos de simulación a gran escala, en los que puede haber uno o más equipos de desarrolladores de modelos de simulación. Es necesario que esta tercera parte conozca perfectamente el propósito de la simulación. El equipo de desarrollo de los modelos de simulación recibe los resultados del equipo de IVV y se va avanzando a medida que se satisfacen los requisitos de validación y verificación. El equipo de IVV también puede realizar la validación y verificación de los modelos de simulación una vez desarrollados los mismos y ofrecer un diagnóstico final. Esta aproximación es la que ofrece un mejor resultado (Sargent, 2003).
- Cálculo de unas puntuaciones “score” ponderadas, subjetivamente y a partir de diferentes aspectos del proceso de validación y que se combinan para determinar las puntuaciones parciales y una puntuación global para el modelo de simulación. Se considerará un modelo de simulación válido si supera una cierta puntuación mínima.

3.4.2 PARADIGMAS DE LA VERIFICACIÓN Y VALIDACIÓN DE LOS MODELOS DE SIMULACIÓN

En la *Figura 22* se ha representado de forma simplificada el proceso de modelización de un sistema y corresponde a la explicación del primer paradigma de la verificación y validación de sistemas, propuesto por Sargent (2003).

La entidad-problema es el sistema o fenómeno (real o no) que debe ser simulado. El modelo conceptual es su representación matemática/lógica/verbal que se desarrolla para un caso particular y el modelo computacional es la implementación del modelo conceptual en un ordenador. Así el modelo conceptual se desarrolla a través de una fase de análisis y modelización y el modelo computacional se desarrolla tras una fase de programación e implementación. En la fase experimental se obtienen

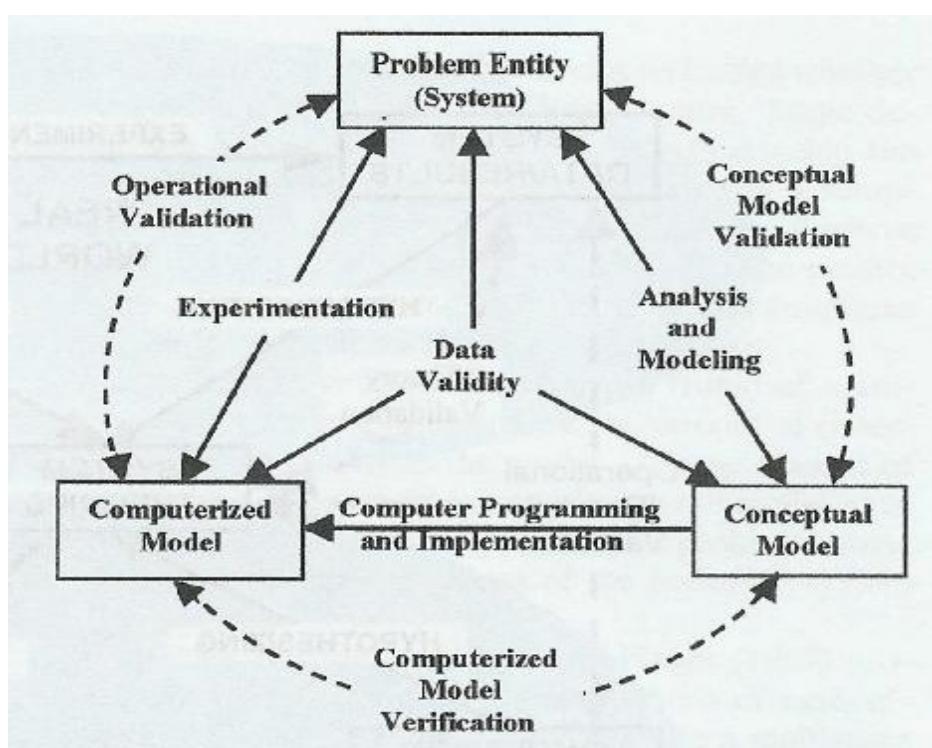


Figura 22: Esquema simplificado del proceso de modelización de un sistema (de Sargent, 2003).

inferencias sobre el sistema tras la realización de experimentos realizados en el ordenador sobre el modelo computacional. Sargent (2003) define de manera formal la validación y verificación en los siguientes términos:

- La validación del modelo conceptual se define como el proceso de comprobación de la veracidad de las teorías y suposiciones realizadas bajo el modelo conceptual. La representación del sistema debe ser correcta con relación al propósito del modelo.

- La verificación del modelo computacional se define como el proceso de asegurar que la programación computacional realizada y la implementación del modelo conceptual sean correctas.

- La validación operacional se define como el proceso que determina si el comportamiento del resultado ("output") tiene suficiente precisión para el propósito del modelo. En ese tipo de validación los datos generados experimentando con el modelo (simulado) se comparan con los datos obtenidos a partir del sistema real. Se trata de la etapa de validación en la que puede incidir más claramente la Estadística, por sus evidentes relaciones con el diseño experimental y el análisis de datos.

- La validación de los datos se define como el proceso de asegurar que los datos con los que se construirá, evaluará y comprobará el modelo, sean apropiados y correctos. Como se ha comentado en el capítulo introductorio, los errores en los datos con los que se construyen los modelos generan una gran cantidad de problemas y se dedicará uno de los capítulos de esta tesis a la gestión de los datos clínicos.

En la **Figura 23** se presenta un esquema detallado de cómo realizar los procesos de validación y verificación para el desarrollo de modelos de simulación según el segundo paradigma. En el esquema se muestran las interconexiones entre el mundo real y el mundo simulado.

La teoría de sistemas describe las características del sistema y su comportamiento en el mundo real, especialmente en el caso de los datos

experimentales u observacionales obtenidos en él. Las teorías sobre el sistema se desarrollan para abstraer lo que se está observando y para establecer hipótesis a partir de los datos y resultados del mismo. Estas teorías se validan cuando convergen, al comparar datos reales y los resultados del sistema, en el ámbito en el que la teoría es aplicable. Este proceso requiere numerosos experimentos que se deben realizar en el sistema real.

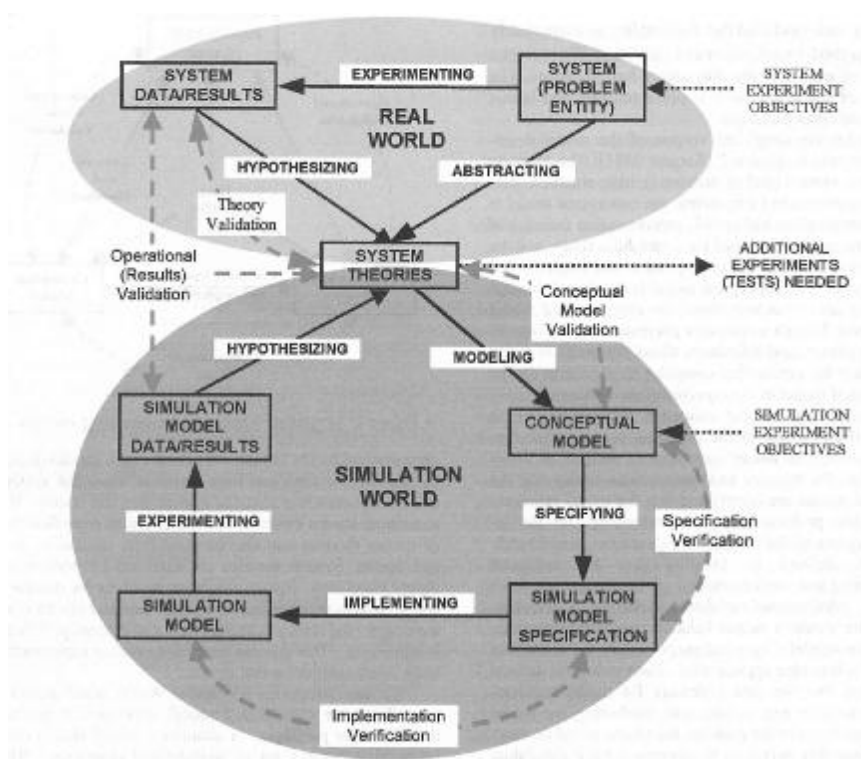


Figura 23: Representación de las relaciones entre el mundo real (superior) y mundo simulado (inferior) con los procesos de validación y verificación (de Sargent, 2003).

En el mundo simulado del segundo paradigma, un modelo de simulación sólo se puede desarrollar si se dispone de un conjunto de objetivos bien definidos. Según el paradigma, el modelo conceptual es el resultado de la modelización de un sistema (entidad problema) y es la representación

matemática/lógica/verbal del sistema, desarrollado según los objetivos de un estudio concreto y el modelo computacional o modelo de simulación es la implementación en un "computador" de las especificaciones del modelo conceptual. Posteriormente se realiza una inferencia acerca del sistema al realizar experimentos en el modelo computacional y analizar los mismos.

Al igual que en el primer paradigma, en el segundo paradigma se aplican los mismos procedimientos de validación y verificación del modelo, aunque con mayor cantidad de pasos por su mayor complejidad conceptual.

3.4.3 TÉCNICAS DE VALIDACIÓN

Las técnicas de validación y verificación de modelos pueden ser objetivas o subjetivas. Una técnica se considera como "objetiva" si está basada en métodos matemáticos o estadístico-inferenciales, como pruebas de hipótesis o intervalos de confianza. En general se aplicará una combinación de técnicas y no sólo una de ellas.

Las técnicas que menciona Sargent (2003) son:

- Trazas: El uso de trazas es el seguimiento del flujo de las entidades (pacientes, en el caso de los ensayos clínicos) a través de cada submodelo y en la totalidad del modelo para determinar si su lógica es correcta y si la precisión necesaria se mantiene. Un buen ejemplo es la implementación del modelo general de ensayo clínico realizado en el entorno LeanSim, que permite un perfecto seguimiento del flujo de cada entidad a través de los diferentes procesos del ensayo (visitas, centros, cambio en la enfermedad, modificación de la enfermedad debida a un tratamiento, etc).
- Validación según el aspecto del modelo (Validación de aspecto): Requiere consultar a expertos en el sistema real estudiado y que

evalúen el modelo conceptual y/o su comportamiento para determinar si éste es correcto para el propósito con el que ha sido construido. Esta validación requiere, normalmente, examinar el diagrama de flujo del modelo, el modelo gráfico, o el conjunto de ecuaciones del modelo. Es uno de los métodos más empleados.

- Animación: Se representa el comportamiento en el tiempo del modelo operacional de una manera gráfica. Esta técnica se utiliza ampliamente en el sector industrial. Por ejemplo el movimiento de una serie de piezas de materia prima durante su procesado en una fábrica. El entorno LeanSim permite la representación gráfica del flujo de las entidades a medida que se produce su simulación.
- Comparación con otros modelos: Los resultados del modelo ("outputs") que se esté simulando se comparan con los resultados de un modelo válido (que ya se ha validado).
- Pruebas de degradación Se estudia el comportamiento del modelo en condiciones extremas, tanto de posibles valores del "input" como de parámetros internos, para determinar hasta que punto se degrada su comportamiento.
- Validación de eventos: Se compara la ocurrencia de eventos en el modelo de simulación y en la realidad y se determina si son similares.
- Prueba de condiciones extremas: La estructura del modelo y el "output" deben ser plausibles para cualquier combinación posible de condiciones extremas y poco frecuentes que puedan producirse en el sistema.
- Validación histórica de datos: Si se han recogido datos históricos, se pueden utilizar parte de los datos para construir el modelo (conjunto para modelización) y el resto para determinar si el modelo de simulación se comporta como lo hace el sistema real (conjunto de validación).
- Métodos históricos: Los tres métodos históricos de validación son el racionalismo, el empiricismo y la economía positiva. En el racionalismo se utilizan deducciones lógicas a partir de suposiciones para desarrollar

el modelo correcto o válido. El empiricismo requiere que cada suposición y resultado sea empíricamente validado. La economía positiva requiere sólo que el modelo sea capaz de predecir el futuro y no está relacionado con las premisas del modelo o su estructura.

- Validación interna: Se realizan réplicas de un modelo estocástico para determinar la cantidad de variabilidad estocástica en el modelo. Una gran cantidad de inconsistencias (pérdida de consistencia) puede causar resultados cuestionables.
- Gráficos operacionales: Cuando el modelo se ejecuta a lo largo del tiempo se muestran de manera gráfica los valores de diferentes medidas del modelo mientras se ejecuta la simulación
- Variabilidad de los parámetros – Análisis de sensibilidad: Esta técnica consiste en cambiar los valores del “input” y de los parámetros internos del modelo para determinar el efecto que produce en el comportamiento o en el “output”. Deben ocurrir las mismas relaciones en el modelo y en el sistema real. Los parámetros del modelo “sensibles” pueden causar cambios en su comportamiento o en el “output”.
- Validación predictiva: El modelo se usa para predecir el comportamiento del sistema y posteriormente se realizan comparaciones entre el comportamiento del sistema y la predicción realizada por el modelo para determinar si son los mismos. Los datos del sistema pueden proceder de un sistema operacional u obtenerse mediante experimentos sobre el sistema real (pruebas de campo).
- Prueba de Turing: Se les pregunta a diferentes expertos si pueden discriminar entre los “outputs” del sistema y modelo.

3.4.4 VALIDACIÓN DE LOS DATOS

Este proceso no se incluye dentro de la validación del modelo porque habitualmente es un proceso complejo, consume tiempo y es costoso obtener datos precisos y válidos para simular posteriormente los modelos.

Frecuentemente es la razón que hace que un modelo de simulación válido falle.

Para construir un buen modelo conceptual es necesaria una cantidad suficiente de datos procedentes del sistema real. Estos han de permitir desarrollar las teorías con las que construir el modelo de simulación y representar adecuadamente la entidad-problema según el propósito con el que ha sido construido. Además, es necesario conocer el comportamiento de los datos dentro del sistema para poder realizar la validación operacional.

Los datos para construir los modelos deben ser precisos, apropiados y suficientes. Si es necesaria su transformación, ésta debe ser correcta (transformación logarítmica, normalización, etc).

En el caso concreto de los ensayos clínicos y principalmente por motivos de confidencialidad, es una tarea sumamente complicada conseguir datos. Por otra parte, deben desarrollarse procedimientos adecuados para recoger, mantener y comprobar que los datos sean correctos. Para ello pueden usarse técnicas como las pruebas de consistencia internas (rangos, valores extremos, etc). Las técnicas implementadas para la recogida de datos de los ensayos clínicos, se centran en facilitar la realización de este proceso pero no garantizan la calidad de los datos que se han grabado informáticamente. Es necesario disponer de herramientas para poder detectar y corregir los posibles errores, que se describen en trabajos como Bonillo-Martin (2003), Davis y col. (1999), Richardson y Chen (2001), y también en <http://www.nhlbi.nih.gov/funding/policies/dataqual.htm> .

Como se demostrará, éste es uno de los principales problemas detectados en el trabajo de investigación realizado en esta tesis y en el siguiente capítulo se trata específicamente la problemática de los datos en los ensayos clínicos y como recoger datos válidos con los que poder estimar los modelos y posteriormente poder simularlos de manera adecuada.

3.4.5 VALIDACIÓN DEL MODELO CONCEPTUAL

La validez del modelo conceptual se puede determinar siempre que las teorías y las premisas subyacentes bajo el modelo conceptual sean las correctas y que la representación del modelo real y de su estructura, así como de las relaciones lógicas, matemáticas y causales sean razonables con la intención que persigue el modelo. Estas teorías y premisas subyacentes bajo el modelo, deben ser comprobadas mediante análisis matemático y métodos estadísticos inferenciales, usando los datos del sistema real. Algunos ejemplos de premisas y de teorías serían asumir que existe linealidad, independencia de los datos, la llegada de los pacientes durante un ensayo clínico siga una distribución de Poisson, Normal, Lagrange, Weibull, etc.

Ejemplos de métodos estadísticos que pueden aplicarse serían la estimación de distribuciones estadísticas a partir de los datos, estimación de los valores de los parámetros a partir de los datos, y la representación de los datos para determinar si estos son estacionarios en el tiempo.

Todas las teorías que se utilicen deberían ser revisadas, así si se utiliza una cadena de Markov, el sistema debe tener la propiedad de Markov y los estados y transiciones entre los mismos deben ser los correctos.

Cada submodelo y la totalidad del modelo deben ser evaluados para determinar si es razonable y acorde con el propósito del modelo.

Las técnicas que primero se utilizarán para la validación del modelo conceptual son las de "validación de aspecto" y "trazas".

Deberán revisarse los errores en el modelo conceptual, si se detectan, así como en su validación.

3.4.6 VERIFICACIÓN DEL MODELO COMPUTACIONAL

La verificación del modelo computacional asegura que la programación computacional y la implementación del modelo computacional sean las correctas. El factor que afecta en mayor medida a la verificación es el uso de un lenguaje de simulación o un lenguaje de alto nivel, como C++, Fortran, etc. El uso de un lenguaje de simulación de propósito específico puede dar lugar a cometer menor cantidad de errores y a utilizar un menor tiempo de programación que el uso de un lenguaje de propósito general.

Cuando se utiliza un lenguaje de simulación, el objetivo de la verificación es asegurar que se ha utilizado un lenguaje de simulación sin errores, que éste haya sido instalado correctamente en el ordenador, que se haya probado el funcionamiento de su generador de números pseudo-aleatorios y que el programa que representa el modelo computacional sea correcto en el lenguaje de simulación.

Las técnicas más frecuentemente utilizadas durante la verificación de los modelos computacionales son la verificación mediante "pasos estructurados" y mediante "trazas".

El programa de ordenador deberá ser diseñado, desarrollado e implementado utilizando técnicas de ingeniería del "software" (técnicas de diseño orientado a objetos, programación estructurada y modularidad en los programas) si se utiliza un lenguaje de programación de alto nivel. En este caso la verificación, está relacionada con determinar si las funciones de simulación (mecanismos de flujo de tiempo, generadores de números pseudo-aleatorios, generadores de variables aleatorias) y el modelo computacional han sido programados e implementados correctamente.

Existen dos aproximaciones para probar los programas informáticos de simulación: pruebas estáticas y pruebas dinámicas. En las primeras se analiza

el programa informático para determinar si es correcto gracias al uso de técnicas como pasos estructurados, pruebas de corrección y examen de las propiedades de la estructura del programa. En las pruebas dinámicas los programas informáticos se ejecutan bajo diferentes condiciones y los valores obtenidos (incluyendo aquellos obtenidos durante la ejecución) se usan para determinar si el programa informático y su implementación son correctos. Las técnicas frecuentemente usadas en las pruebas dinámicas son las trazas, la investigación de las relaciones entre entradas-salidas ("input"- "output"), comprobaciones de consistencia interna y reprogramación de componentes críticos para determinar si se obtienen los mismos resultados. Si hay un gran número de variables, algunas de las variables se pueden agregar a fin de reducir el número de pruebas a realizar mediante diseño de experimentos.

3.4.7 VALIDACIÓN OPERACIONAL

La validación operacional consiste en determinar si los datos generados experimentando con el modelo (simulando) son comparables con los datos obtenidos a partir del sistema real con una determinada precisión y bajo el dominio de la aplicabilidad del modelo. Este tipo de validación es de naturaleza básicamente estadística.

Si se encuentra alguna deficiencia durante la validación operacional, puede estar causada por cualquiera de los pasos en los que está implicado el modelo de simulación, como la teoría con la que ha sido desarrollado o los datos utilizados.

Todas las técnicas de validación señaladas anteriormente son aplicables para realizar la validación operacional. Qué técnicas y cuándo utilizarlas objetiva o subjetivamente deberá ser decidido por el equipo de simulación y por todas las partes interesadas. El factor que afecta en mayor medida a la validación

operacional es que el sistema sea observable, es decir que sea posible observar datos del comportamiento de la entidad-problema.

Algunas de las aproximaciones a la validación operacional son la comparación utilizando representación gráfica y la exploración del comportamiento del modelo en el caso de los sistemas observables y aproximaciones objetivas. En el caso de los sistemas no observables y de aproximaciones subjetivas, las técnicas más usadas son la exploración del comportamiento de los modelos y la comparación con otros modelos, en el resto de situaciones es preferible el uso de una técnica estadística.

Se utilizarán principalmente 3 tipos de aproximaciones para realizar una validación operacional:

- Una decisión subjetiva basada en el uso de gráficos del comportamiento del modelo y de los datos.
- El uso de intervalos de confianza.
- El uso de pruebas de hipótesis.

Es preferible el uso de intervalos de confianza y de pruebas de hipótesis para las comparaciones, ya que permiten tomar decisiones más objetivas. Pero con frecuencia en la práctica, no es posible utilizar ni lo uno ni lo otro, porque las premisas estadísticas requeridas no cumplen o lo hacen con grandes dificultades (por ejemplo las de independencia y normalidad) o bien puede suceder que no haya suficiente cantidad de datos disponibles lo que casua que los resultados estadísticos no sean significativos (por ejemplo que el intervalo de confianza desarrollado en la comparación del sistema y del modelo sea demasiado grande para ser útil en la práctica). La aproximación más corriente utilizada en la validación operacional es el uso de gráficos.

3.4.7.1 USO DE GRÁFICOS EN LA VALIDACIÓN OPERACIONAL

Se representa el comportamiento del modelo y del sistema a través de gráficos en varios conjuntos de condiciones experimentales, a fin de determinar si el resultado del comportamiento del modelo tiene suficiente precisión para el propósito con el que ha sido diseñado.

Se suelen utilizar gráficos del tipo histograma, diagramas de cajas (“Box-plot”) y gráficos de dispersión. Una revisión sobre la representación gráfica para la validación operacional puede encontrarse en Anderson y Sargent (1974).

Los gráficos se pueden utilizar en las diferentes etapas de la validación del modelo. El equipo de trabajo puede utilizar los gráficos durante el desarrollo del modelo para realizar una evaluación subjetiva sobre la precisión del modelo y determinar si ésta es suficiente para el propósito con el que ha sido construido. En segundo lugar se pueden utilizar durante la etapa de validación del aspecto mediante la consulta a grupo de expertos. Los gráficos también se pueden utilizar durante la prueba de Turing y durante el proceso de IV&V

3.4.7.2 USO DE INTERVALOS DE CONFIANZA EN LA VALIDACIÓN OPERACIONAL

Los intervalos de confianza pueden servir como rango de precisión del modelo durante la validación del modelo. Así puede obtenerse un intervalo de confianza para cada conjunto de condiciones experimentales, un intervalo de confianza conjunto o una región de confianza conjunta para las diferencias entre medias, varianzas y distribuciones de los diferentes resultados del modelo y del sistema.

Existen numerosos métodos descritos para la construcción de intervalos de confianza, univariantes o multivariantes, como los que se encuentran en Law y Kelton (2000) para el caso univariante. Banks y col. (2000) y Law y Kelton (2000) describen el uso de intervalos de confianza para determinar la precisión del modelo. También Balci y Sargent (1984) describen la metodología general del uso de intervalos de confianza para la validación operacional de los modelos.

Habitualmente se desea construir un rango de precisión lo más pequeño posible a partir del intervalo de confianza. Cuanta más pequeña la longitud del intervalo de confianza más preciso el rango de confianza. El rango de precisión viene determinado por los valores de los intervalos de confianza, la varianza de las variables del modelo y de las variables de resultados. Por otro lado este rango puede hacerse más pequeño al hacer más pequeño el nivel de confianza o incrementar el tamaño muestral (Monleón y col., 2003^e).

Para determinar el rango de precisión del modelo deben realizarse pruebas previas entre el tamaño muestral, intervalos de confianza y las estimaciones de la longitud y rango de precisión del modelo. Pueden construirse curvas de precisión para los factores señalados.

A continuación se presenta la base matemática de la estimación de 2 tipos de intervalos de confianza utilizados posteriormente en la metodología de esta tesis donde la mayoría de ejemplos utilizados se refieren a modelos mixtos lineales/no lineales.

Intervalos de confianza normales de la predicción

Para construir el intervalo de confianza de la predicción de un modelo lineal (Chatterjee y Hadi, 1986) se parte de la ecuación general del modelo:

$$Y = Xb + e \quad (6)$$

Donde: $e \sim N(0, s^2)$

La resolución del modelo corresponde a la obtención de las siguientes estimaciones y predicciones:

$$b = \hat{b} = (X^T X)^{-1} y \quad (7)$$

$$\hat{y} = Xb = Hy \quad (8)$$

$$H = X(X^T X)^{-1} X^T \quad (9)$$

El cálculo de los residuos se obtiene mediante:

$$r = y - \hat{y} = (1 - H)y \quad (10)$$

Donde: $r \sim N(0, s^2)$

Los residuos "estudentizados" se obtienen mediante la expresión:

$$t_i = \frac{r_i}{s \sqrt{1 - h_i}} \quad (11)$$

Donde $t_i \sim t$ Student $(n - p - 1)$ g.l. y su varianza puede calcularse como:

$$s_{(i)}^2 = \frac{|r|^2}{n - p - 1} - \frac{r_i^2}{(n - p - 1)(1 - h_i)} \quad (12)$$

Donde n es el tamaño muestral, p el número de parámetros del modelo y h_i es el i -ésimo elemento de la diagonal de H .

De esta manera puede obtenerse un intervalo de confianza de la predicción como:

$$c_i = r_i \pm t_{(1-\frac{\alpha}{2}, n)} s_{(i)} \sqrt{1 - h_i} \quad (13)$$

Los intervalos de confianza (13) que no incluyen el valor 0 son equivalentes a rechazar la hipótesis nula (H_0) de que no hay diferencias entre una

observación y la predicha por el modelo, al nivel de α . Estos intervalos de confianza son una buena evidencia de que la observación es un valor fuera de rango para el modelo estimado.

Un inconveniente de este tipo de intervalos es la suposición de que los residuos son normales y cumplen las suposiciones realizadas en (5) y (6).

Intervalos de confianza bootstrap

Algunos autores han estudiado la validación de modelos computacionales de simulación mediante intervalos de confianza basados en remuestreo bootstrap, aunque no se ha descrito la utilización de estos en la validación de simulaciones de ensayos clínicos. En esta tesis se intentará demostrar su utilidad en la validación de los modelos computacionales frente a los datos reales y el modelo conceptual.

En el CD anexo se transcribe un programa macro de SAS para el cálculo de intervalos de confianza de la media y la mediana para las estimaciones de las simulaciones con y sin reducción de sesgo basado en los trabajos de Efron y Tibshirani (1993) y Shao y Tu (1995) que realizan 200 remuestreos de la matriz de datos original.

La principal ventaja de la utilización de este tipo de intervalos es que no se basan en una distribución estadística conocida previamente y por lo tanto es aplicable a modelos que generen datos no gaussianos o bien a modelos estadísticos estimados que no presenten unos residuos normales. La complejidad computacional puede ser importante cuando el número de muestras es elevado como se ha podido comprobar durante las pruebas realizadas con la macro de SAS mencionada.

Intervalos de confianza de los parámetros fijos y aleatorios del modelo mixto

Algunos autores han estudiado la variabilidad de la estimación de los parámetros para los modelos lineales mixtos como es el caso de Jørgensen (2004) y Sánchez y Ocaña (2002).

Autores como Walker (2002) y Neter y col. (1996) han estudiado ensayos clínicos modelizados mediante modelos lineales mixtos en SAS (PROC MIXED) donde se muestran pormenorizadamente los cálculos de los intervalos de confianza de los parámetros del modelo, al nivel α , para los parámetros estimados. Un ejemplo de cálculo de intervalo de confianza de la predicción mediante el programa SAS es la opción "cl" ("t-type confidence limits": www.id.unizh.ch/software/unix/statmath/sas/sasdoc/stat/chap41/sect23.htm#mixedestbg) que se calcula para cada uno de los parámetros fijos estimados:

```
proc mixed cl;  
  class treatment block;  
  model y = treatment/ddfm=satterth;  
  random block;  
run;
```

En el procedimiento "PROC MIXED" los valores predcidos por el modelo, una vez estimados a partir de los datos originales son:

$$\hat{y} = X \hat{b} + Z \hat{g} \quad (14)$$

Donde \hat{b} son los parámetros fijos y \hat{g} los aleatorios, cuya solución puede ser escrita como:

$$\begin{aligned}\hat{\beta} &= (\mathbf{X}'\hat{\mathbf{V}}^{-1}\mathbf{X})^{-1}\mathbf{X}'\hat{\mathbf{V}}^{-1}\mathbf{y} \\ \hat{\gamma} &= \hat{\mathbf{G}}\mathbf{Z}'\hat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\beta})\end{aligned}\tag{15}$$

Donde $V = Var(y)$ y $V = ZGZ' + R$. g y ϵ se distribuyen según una normal con parámetros:

$$\begin{aligned}\mathbf{E} \begin{bmatrix} \gamma \\ \epsilon \end{bmatrix} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ \mathbf{Var} \begin{bmatrix} \gamma \\ \epsilon \end{bmatrix} &= \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}\end{aligned}\tag{16}$$

Los intervalos de confianza de la predicción (de manera estándar al 95%) se construyen como se ha indicado anteriormente suponiendo una distribución t-Student para los residuos “estudentizados” y se corresponden a los intervalos calculados mediante los procedimientos de SAS GLM (General Linear Models) y REG (Regresión) para factores de efectos fijos. Este intervalo presenta la siguiente forma:

$$\mathbf{L} \begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix} \pm t_{\mathcal{D}, \alpha/2} \sqrt{\mathbf{L}\hat{\mathbf{C}}\mathbf{L}'}\tag{17}$$

Donde $\mathbf{L}\beta$ es la función general estimable y $\hat{\mathbf{L}}$ es la menor raíz triangular Cholesky de $\hat{\mathbf{G}}$ que satisface $\hat{\mathbf{G}} = \hat{\mathbf{L}}\hat{\mathbf{L}}'$. $t_{\mathcal{D}, \alpha/2}$ es el percentil $100(1 - \frac{\alpha}{2})$ de la distribución t_u y $\hat{\mathbf{C}}$ es la matriz de covarianzas estimada.

Para el cálculo de los valores predichos según el modelo estimado, el programa SAS utiliza el algoritmo EBLUP (“Empirical Best Linear Unbiased Prediction”), especialmente en aquellos casos con valores faltantes. La fórmula de EBLUP es:

$$\hat{m} = X_m \hat{b} + \hat{C}_m \hat{V}^{-1} (y - X \hat{b}) \quad (18)$$

Donde m representa una hipotética realización de un vector de datos ausentes (missing) con una matriz de diseño asociado X_m . La matriz C_m es la matriz de covarianzas basada en el modelo entre m y el dato observado y .

La varianza de la predicción estimada es:

$$Var(\hat{m} - m) = \hat{V}_m - \hat{C}_m \hat{V}^{-1} \hat{C}_m^T + \left[X_m - \hat{C}_m \hat{V}^{-1} X \right] \left(X^T \hat{V}^{-1} X \right)^{-1} \left[X_m - \hat{C}_m \hat{V}^{-1} X \right]^T \quad (19)$$

Donde V_m es la matriz de covarianzas de m basada en el modelo.

3.4.7.3 USO DE PRUEBAS DE HIPÓTESIS EN LA VALIDACIÓN OPERACIONAL

Se pueden utilizar las pruebas de hipótesis para comparar, entre otras características, medias, varianzas, distribuciones y características de series temporales de las variables de resultados del modelo y del sistema real para cada conjunto de condiciones experimentales, para determinar si el comportamiento de los resultados del modelo posee un rango de precisión adecuado.

Se define el rango de precisión aceptable como aquel que se requiere para que el modelo sea válido según la intención con la que ha sido construido.

Las hipótesis que se contrastan habitualmente durante la validación operacional son:

H_0 : El modelo es válido para un rango de precisión aceptable, bajo el conjunto de condiciones experimentales estudiadas
 H_1 : El modelo es inválido para un rango de precisión aceptable, bajo el conjunto de condiciones experimentales estudiadas

Como ocurre siempre que se realizan pruebas de hipótesis, pueden darse 2 tipos de errores. El error de tipo I, bajo el planteamiento anterior, correspondería a rechazar la validez de un modelo válido y el error de tipo II a aceptar la validez de un modelo no válido. La probabilidad del error de tipo I, a , en el ámbito de la simulación se denomina habitualmente riesgo de construcción del modelo y la probabilidad del error de tipo II, b , se denomina riesgo del usuario del modelo (Balci y Sargent, 1981). Así para la validación de modelos, el riesgo del usuario del modelo es muy importante y debería ser pequeño. Se deberán tener muy en cuenta los errores de tipo I y de tipo II cuando se utilicen pruebas de hipótesis para la validación de los modelos.

A la cantidad de distancia entre el modelo y el sistema se la denomina medida de validación, I , y se utiliza como medida de precisión del modelo. Se puede usar para determinar el rango de validez: $0 \leq I \leq I^*$.

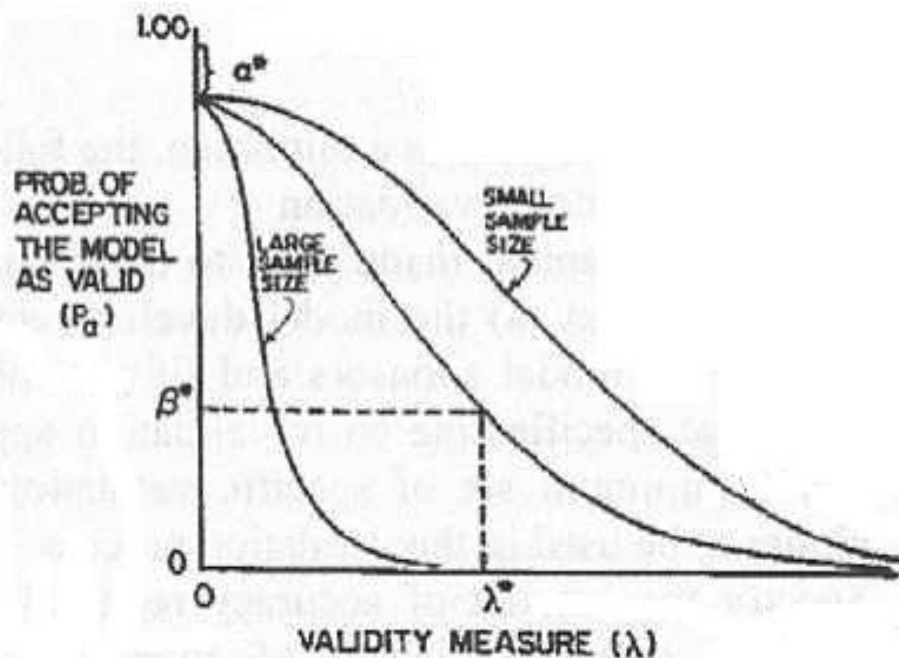


Figura 24: Curvas características entre la medida de validez de un modelo y la probabilidad de aceptar el modelo como válido P_a , (de Sargent, 2003).

La probabilidad de aceptar un modelo siendo válido, P_a , se puede entender como una función de la medida de la validez mediante el uso de su curva característica (Johnson, 1994). En la **Figura 24** se representan 3 curvas características que ilustran cómo afecta el tamaño muestral a la función P_a , donde ésta se debe entender como una función de I . Puede observarse que un modelo inadecuado tiene una probabilidad alta de ser aceptado si se utiliza un número pequeño de muestras, y un modelo adecuado tiene una baja probabilidad de ser aceptado si se utiliza un gran tamaño muestral.

La localización y la forma de las curvas características son función por una parte de las técnicas estadísticas utilizadas, del valor de α elegido para $I = 0$ (α^*) y del tamaño muestral de las observaciones. Una vez se han construido las curvas características, se puede determinar para un valor I^* los intervalos

para el riesgo del usuario del modelo b (I) y el riesgo de construcción del modelo (a) de la siguiente manera:

$$a^* \leq a \leq (1-b^*) \quad (20)$$

$$0 \leq \text{riesgo usuarios del modelo} \leq b^*.$$

Existe una relación directa entre el riesgo de la construcción del modelo, el riesgo del usuario del modelo, el rango de validez aceptable y el tamaño muestral de las observaciones. Si se utilizan pruebas de hipótesis para la validación del modelo debe realizarse entre los parámetros señalados.

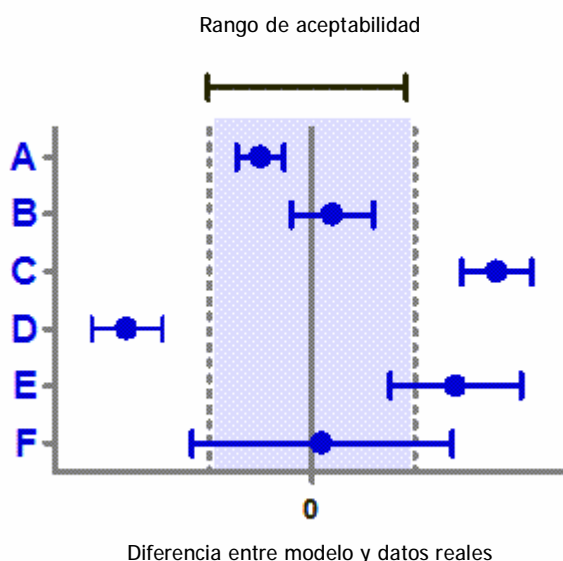
La metodología para la validación de modelos utilizando pruebas de hipótesis puede verse en Balci y Sargent (1981) y en Reynolds y Deaton (1982).

Los métodos estadísticos basados en la realización de contrastes de hipótesis (o en la obtención de intervalos o regiones de confianza pero con la intención de acabar "aceptando" o "rechazando" la hipótesis de validez del modelo) abordan siempre la validación operacional desde una perspectiva "de bondad de ajuste", en el sentido de que se basan en alguna medida de concordancia entre los datos obtenidos en el sistema real y los pseudo-datos generados por el modelo de simulación y en los que la validez exacta del modelo corresponde a la hipótesis nula, y su no-validez a la hipótesis alternativa.

Las discusiones anteriores en las que se aconseja utilizar tamaños muestrales de réplicas de simulación no demasiado grandes, para no rechazar un modelo válido, resultan, como mínimo, chocantes. Sería interesante estudiar un enfoque de las pruebas de hipótesis de validación desde el punto de vista de la equivalencia, en la misma línea que las pruebas de bioequivalencia (Schuirmann, 1987; Chow y Liu, 1992), en donde se planteen la siguiente hipótesis:

- H_0 : El modelo es inválido para un rango de precisión aceptable, bajo el conjunto de condiciones experimentales
- H_1 : El modelo es válido para un rango de precisión aceptable, bajo el conjunto de condiciones experimentales (*Validez \pm precisión*)

Es decir, con la hipótesis de validez del modelo (equivalencia entre los datos experimentales y el modelo) como hipótesis alternativa. Parece que entonces si tendría sentido decir que “se ha demostrado” la validez de un modelo y se corregiría el defecto de las pruebas de hipótesis en cuanto al tamaño muestral, expuestos anteriormente. En forma de intervalos de confianza podrían darse las siguientes situaciones:



En los casos A y B, las diferencias entre el modelo conceptual y el de simulación son muy pequeñas, y se considera la equivalencia entre ambos: validación.

En los casos C y B, las diferencias entre el modelo conceptual y el de simulación son grandes y se considera la no equivalencia: no validación.

En los casos E y F existen pruebas tanto a favor de la equivalencia entre el modelo conceptual y el simulado como a favor de la no equivalencia, por lo

que los datos son ambiguos y se necesitan más pruebas u otros datos para decidir entre una u otra hipótesis.

3.4.8 DOCUMENTACIÓN

Debe realizarse una documentación de la validación del modelo donde se recojan las pruebas realizadas, datos, resultados, validación de los datos, verificación del modelo computacional, validación del modelo conceptual, validación operacional, etc.

3.4.9 ACREDITACIÓN

Un tema muy vinculado a la validación y verificación de modelos de simulación es el de su "acreditación" por parte de algunos organismos oficiales, como sucede en el Departamento de Defensa de Canadá (Department of Defense, 1996), que ha establecido normas y una oficina para acreditar modelos de simulación, en el sentido de certificar su validez para un determinado fin. En este certificado se acredita de forma oficial que el modelo, simulación o conjunto de modelos y simulaciones son aceptables para el uso de una aplicación específica.

Habitualmente la acreditación es responsabilidad de una tercera parte, independiente y que frecuentemente incluye en la valoración no sólo la verificación y la validación, sino ítems como la documentación y la amigabilidad de los programas de simulación. Algunos autores como Balci (1996) ofrecen una guía detallada de cómo acreditar un modelo de simulación, pero en un ámbito general sin entrar a tratar situaciones concretas.

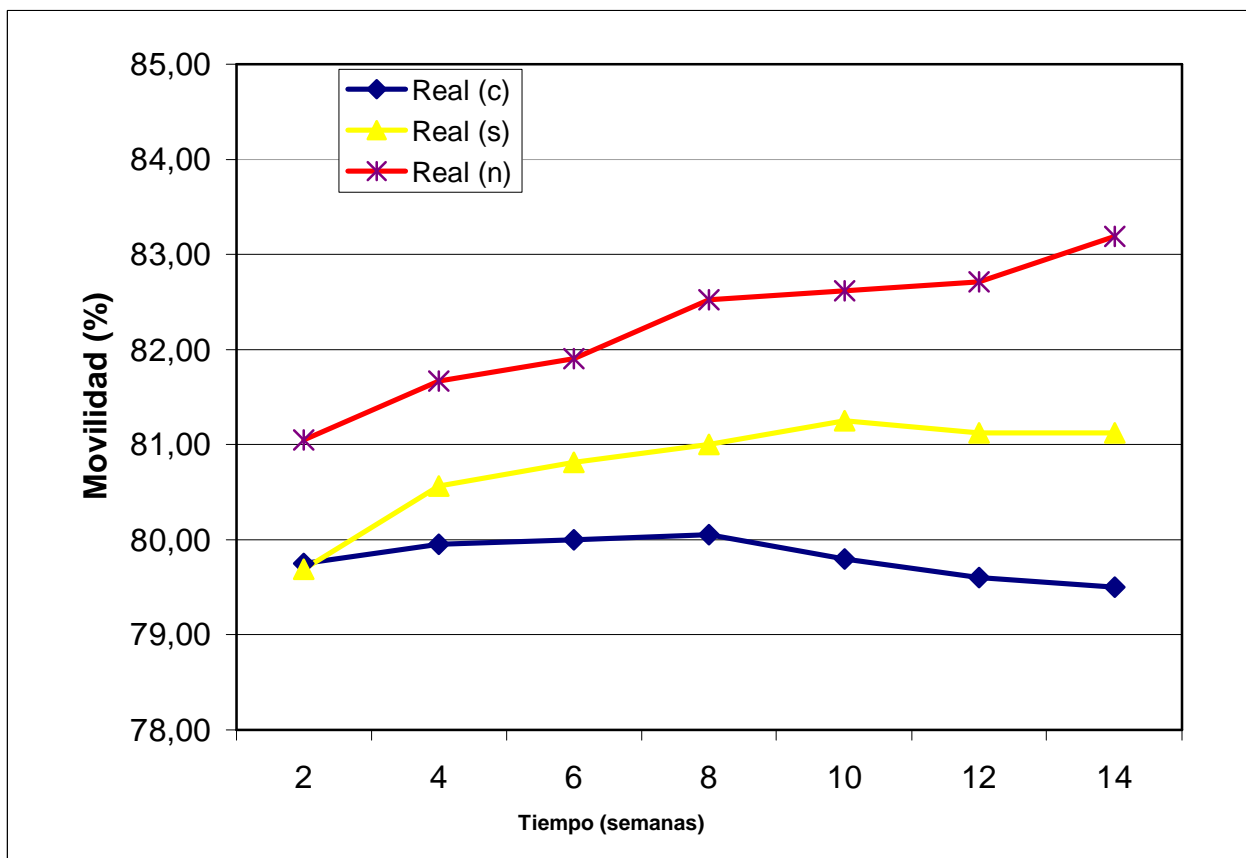
Las siglas VV&A corresponden al proceso de validación, verificación y acreditación (Balci, 1986).

3.4.10 DESARROLLO DE UNA METODOLOGÍA PARA LA ESTIMACIÓN DE MODELOS Y SIMULACIÓN DE ENSAYOS CLÍNICOS

En esta sección se pretende ilustrar el proceso de modelado con un ejemplo real de ensayo clínico como hilo conductor.

Neter y col. (1996) describen un ensayo clínico sobre la limitación que produce la artritis reumatoide en las personas afectadas por ella. Dicho ensayo pretendía demostrar la eficacia de un nuevo fármaco para su tratamiento, comparándolo con un tratamiento estándar y placebo. Estos tratamientos fueron administrados durante 14 semanas a 57 pacientes (20 placebo, 16 tratamiento estándar y 21 nuevo tratamiento). El diseño elegido para este ensayo fue el de medidas repetidas y las mediciones se realizaron cada 2 semanas. La variable de respuesta analizada fue una escala de movilidad, entre 1 y 100%.

Las observaciones medias de los diferentes grupos de tratamiento aparecen en la siguiente gráfica:



c = Placebo (1), s = Tratamiento estándar (2), n = Nuevo tratamiento (3)

3.4.10.1 ESTIMACIÓN DEL MODELO CONCEPTUAL

En SAS, a partir de unos datos se puede ajustar un modelo mixto lineal mediante la sintaxis:

```
proc mixed data = arthrit2 method = ml covtest cl;
class subj;
model y = drug time time*drug/solution cl;
random intercept time /subject =subj type = un g gcorr;
run;
```

Se indica que el término independiente y el tiempo en que se produce la visita poseen unos parámetros aleatorios y existe una posible interacción entre el tiempo y medicación. Se ha considerado el tiempo como un factor de tipo continuo (variable independiente), ya que ofrece una menor sobreparametrización de los modelos estimados, como ya se mostró en

Monleón y col. (2003^a). Se ha considerado la medicación como un factor discreto, con tres niveles: *n*, *c* y *s*.

Existen múltiples opciones en el procedimiento de cálculo de SAS y una de ellas permite especificar el tipo de estimación de la matriz de covarianza. En este caso se ha optado por el método "UN" (unstructured covariance) porque es la recomendada por Neter y col. (1996). De esta manera y como primera aproximación, no se presupone ninguna estructura en la matriz de covarianzas y si se dispone de *q* factores, por lo que se han de estimar $(q^2 - q)/2$ parámetros. Suponiendo otras estructuras de covarianza sería necesario estimar menos parámetros, por ejemplo *q* en el caso de suponer que la matriz es de tipo diagonal. En el CD anexo de esta memoria pueden verse diferentes estructuras de covarianzas, admitidas por el programa SAS.

También los algoritmos de cálculo implementados en SAS permiten el cálculo de diferentes tipos de criterios de bondad de ajuste como *-2 Log Verosimilitud (-LL)*, más optimista cuando existen muchos parámetros en el modelo, ya que con este criterio no se penaliza para el número de parámetros y no se prima la parsimonia del modelo. Los criterios AIC (Akaike's Information Criterion) y BIC (Schwartz's Bayesian criterion) sí penalizan para el número de parámetros, especialmente este último (Neter y col., 1996).

El listado de resultados, para la estimación de un modelo lineal mixto con los datos analizados, es el siguiente:

Proc Mixed datos H. Clinic 12:20 Friday, April 2, 2004 166

The Mixed Procedure

Model Information

Data Set	WORK.ARTHRIT2
Dependent Variable	y
Covariance Structure	Unstructured
Subject Effect	subj
Estimation Method	ML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Class Level Information

Class	Levels	Values
subj	21	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
drug	3	1 2 3

Dimensions

Covariance Parameters	4
Columns in X	8
Columns in Z Per Subject	2
Subjects	21
Max Obs Per Subject	21
Observations Used	399
Observations Not Used	0
Total Observations	399

Iteration History

Iteration	Evaluations	-2 Log Like	Criterion
0	1	2061.02469933	
1	4	2000.57278402	0.00128119
2	1	2000.34783821	0.00048893
3	1	1999.98827125	0.00004436
4	1	1999.95836169	0.00000049
5	1	1999.95804802	0.00000000

Convergence criteria met.

Estimated G Matrix

Row	Effect	subj	Col 1	Col 2
1	Intercept	1	0.9801	0.07844
2	time	1	0.07844	

Estimated G Correlation Matrix

Row	Effect	subj	Col 1	Col 2
1	Intercept	1	1.0000	1.0000
2	time	1	1.0000	1.0000

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z	Alpha	Lower	Upper
UN(1, 1)	subj	0.9801	1.0197	0.96	0.1682	0.05	0.2568	50.2447
UN(2, 1)	subj	0.07844	0.07020	1.12	0.2639	0.05	-0.05916	0.2160
UN(2, 2)	subj	5.87E-18
Residual		8.0337	0.5839	13.76	<.0001	0.05	7.0017	9.3133

Fit Statistics

-2 Log Likelihood	2000.0
AIC (smaller is better)	2018.0
AICC (smaller is better)	2018.4
BIC (smaller is better)	2027.4

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > Chi Sq
2	61.07	<.0001

Solution for Fixed Effects

Effect	drug	Estimate	Standard Error	DF	t Value	Pr > t	Alpha	Lower	Upper
Intercept		80.9184	0.5656	20	143.06	<.0001	0.05	79.7385	82.0982
drug	1	-0.8993	0.7488	353	-1.20	0.2305	0.05	-2.3719	0.5732
drug	2	-0.9954	0.7954	353	-1.25	0.2116	0.05	-2.5598	0.5690
drug	3	0
time		0.1650	0.05844	20	2.82	0.0105	0.05	0.04305	0.2869
time*drug	1(c)	-0.1958	0.08365	353	-2.34	0.0198	0.05	-0.3603	-0.03128
time*drug	2(s)	-0.06197	0.08869	353	-0.70	0.4852	0.05	-0.2364	0.1124
time*drug	3(n)	0

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
drug	2	353	1.03	0.3576
time	1	20	4.92	0.0384
time*drug	2	353	2.83	0.0604

Los resultados indican que existe una interacción significativa ($\alpha=0,1$) entre la medicación y el tiempo.

El modelo mixto estimado, que es la base del “modelo conceptual” de la simulación, es el siguiente:

$$\% \text{ Movilidad} = (80,92 [79,7;82,1] + b_0) + b_i + (0,165 [0,04;0,29] + b_1) \text{Tiempo}_j + (f_i \text{ Visita}_j) + e_{ij}$$

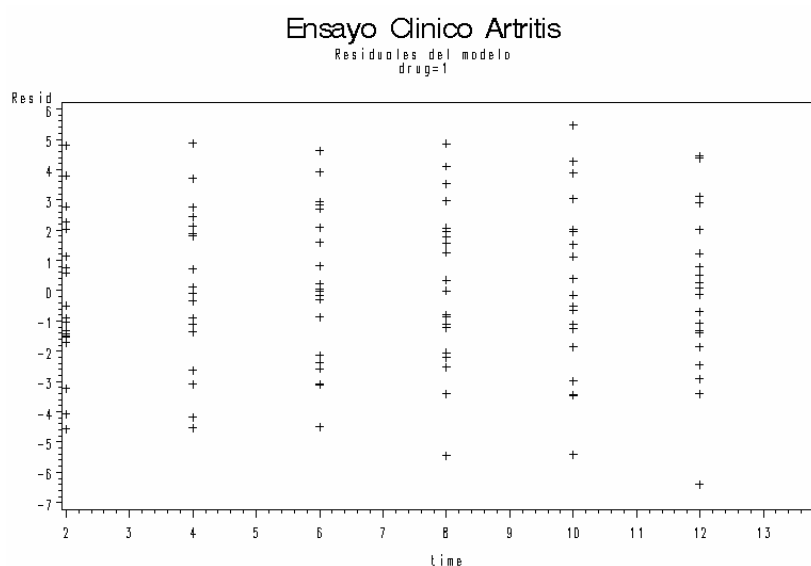
Donde b_0 se considera en principio con distribución $N(0; 0,99 [0,51;7])$, $b_1 \sim N(0,0,00 [0,00; 0,00])$ por lo que es descartado del modelo, (Covarianza d entre b_0 y $b_1 = 0,07844$) y $e_{ij} \sim N(0; 2,83 [2,65;3,05])$ y $\beta_i = (-0,8998 [-2,37;0,57]; -0,9954 [-2,55;0,57]; 0)$ representa los coeficientes de los efectos fijos de los tratamientos c, s y n respectivamente. $f_i = (-0,1958 [-0,36;-0,03]; -0,06197 [-0,24;-0,11]; 0)$ representa los coeficientes de los efectos fijos de las interacciones del tiempo con los tratamientos c, s y n

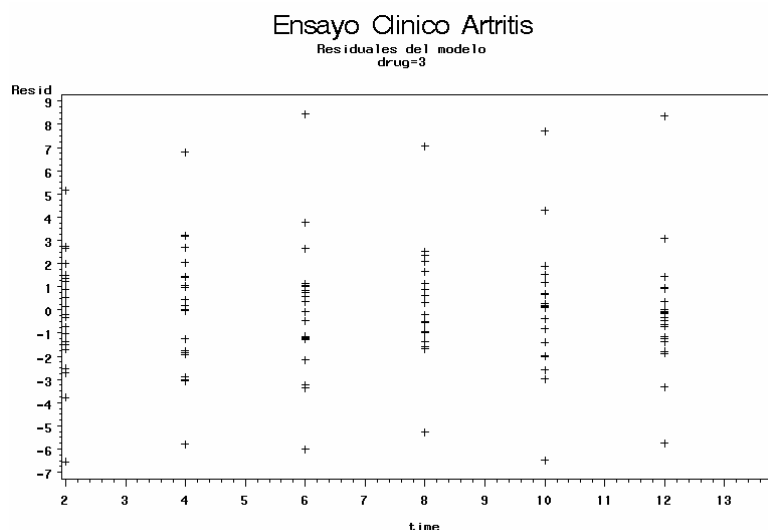
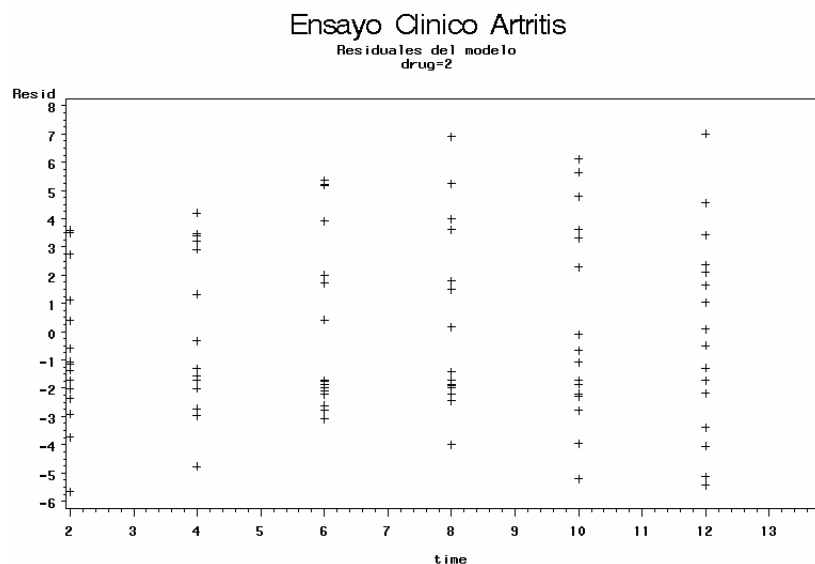
respectivamente. Se indican entre corchetes y cursiva los intervalos de confianza de nivel 95% de los parámetros estimados. Es aconsejable que dichos intervalos de confianza acompañen la ecuación del modelo a fin de poner de manifiesto el grado de incertidumbre y posteriormente realizar un análisis de sensibilidad del modelo computacional cambiando los parámetros estimados por otros parámetros dentro de los rangos de variación fijados por su intervalo de confianza.

3.4.10.2 VALIDACIÓN DEL MODELO CONCEPTUAL

También es conveniente la realización de análisis de residuos de los valores predichos por el modelo, a fin de conocer cómo se comporta respecto a los datos reales y perfilar una posible estrategia de validación.

Se ha realizado un análisis de los residuos del modelo obteniéndose las gráficas de los residuos del modelo respecto al factor tiempo, considerado como un predictor:





Un análisis estadístico de los residuos revela que para los tratamientos s y n estos residuos no son normales (Test de Kolmogorov, aunque el gráfico de Levene no esté muy desviado):

Ensayo Clínico Artritis 12: 20 Friday, April 2, 2004 206
resids from ar model, quadratic

----- drug=1 -----

The UNIVARIATE Procedure
Variable: Resid

Moments

N	140	Sum Weights	140
Mean	0	Sum Observations	0
Std Deviation	2.58407768	Variance	6.67745748

Skewness	- 0. 0316944	Kurtosis	-0. 4199619
Uncorrected SS	928. 166589	Corrected SS	928. 166589
Coeff Variation	.	Std Error Mean	0. 21839442

Basic Statistical Measures

Location		Variability	
Mean	0. 00000	Std Deviation	2. 58408
Median	-0. 15671	Variance	6. 67746
Mode	.	Range	11. 84949
		Interquartile Range	3. 63501

Tests for Location: $\mu_0=0$

Test	-Statistic-	-----p Value-----		
Student's t	t	0	Pr > t	1. 0000
Sign	M	-5	Pr >= M	0. 4470
Signed Rank	S	-6	Pr >= S	0. 9901

Ensayo Clinico Artritis

12: 20 Friday, April

2, 2004 207

resids from ar model, quadratic

----- drug=1 (Placebo)-----

The UNIVARIATE Procedure

Variable: Resid

Tests for Normality

Test	--Statistic--	-----p Value-----		
Shapiro-Wilk	W	0. 989206	Pr < W	0. 3514
Kolmogorov-Smirnov	D	0. 056145	Pr > D	>0. 1500
Cramer-von Mises	W-Sq	0. 070037	Pr > W-Sq	>0. 2500
Anderson-Darling	A-Sq	0. 394757	Pr > A-Sq	>0. 2500

Quantiles (Definition 5)

Quantile	Estimate
100% Max	5. 464035
99%	5. 298200
95%	4. 420110
90%	3. 629869
75% Q3	2. 009840
50% Median	-0. 156705
25% Q1	-1. 625171
10%	-3. 166039

5% -4.330159
 1% -6.349177
 0% Min -6.385453

Ensayo Clinico Artritis

12: 20 Friday, April

2, 2004 208

resids from ar model, quadratic

----- drug=1 -----

The UNIVARIATE Procedure

Variable: Resid

Extreme Observations

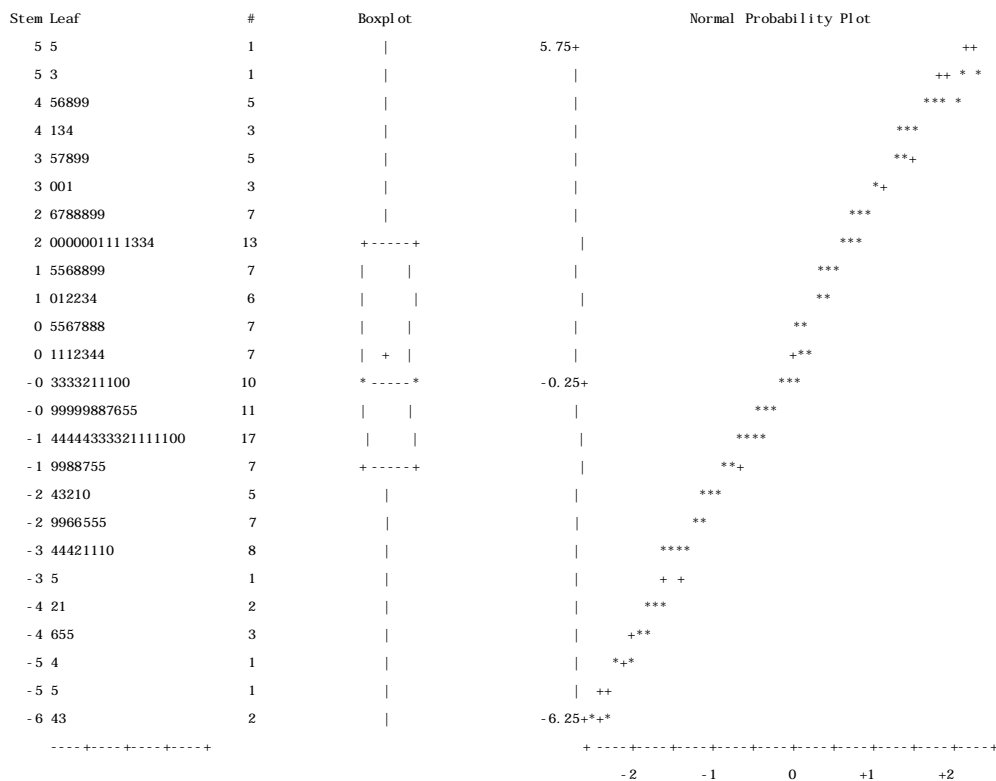
-----Lowest-----		-----Highest-----	
Value	Obs	Value	Obs
-6.38545	108	4.80632	14
-6.34918	128	4.85967	77
-5.45800	68	4.86154	34
-5.42173	88	5.29820	121
-4.56683	8	5.46403	81

resids from ar model,

----- drug=1 -----

The UNIVARIATE Procedure

Variable: Resid



resids from ar model, quadratic

----- drug=2 -(Standar) -----

The UNIVARIATE Procedure

Variable: Resid

Moments

N	112	Sum Weights	112
Mean	0	Sum Observations	0
Std Deviation	3.18275169	Variance	10.1299083
Skewness	0.40220348	Kurtosis	-0.8682406
Uncorrected SS	1124.41983	Corrected SS	1124.41983
Coeff Variation	.	Std Error Mean	0.30074177

Basic Statistical Measures

Location		Variability	
Mean	0.00000	Std Deviation	3.18275
Median	-1.10605	Variance	10.12991
Mode	.	Range	12.67569
		Interquartile Range	5.38612

Tests for Location: Mu0=0

Test	-Statistic-	-----p Value-----	
Student's t	t 0	Pr > t	1.0000
Sign	M -9	Pr >= M	0.1078
Signed Rank	S -43	Pr >= S	0.9013

Ensayo Clinico Artritis

12:20 Friday, April

2, 2004 211

resids from ar model, quadratic

----- drug=2 -----

The UNIVARIATE Procedure

Variable: Resid

Tests for Normality

Test	--Statistic---	-----p Value-----	
Shapiro-Wilk	W 0.947859	Pr < W	0.0003
Kolmogorov-Smirnov	D 0.14874	Pr > D	<0.0100
Cramer-von Mises	W-Sq 0.473213	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq 2.422638	Pr > A-Sq	<0.0050

Quantiles (Definition 5)

Quantile	Estimate
100% Max	7.00585
99%	6.91954
95%	5.34964
90%	4.36363
75% Q3	3.14063
50% Median	-1.10605
25% Q1	-2.24550
10%	-3.71420
5%	-4.63986
1%	-5.42743
0% Min	-5.66984

resids from ar model, quadratic

----- drug=2 -----

The UNIVARIATE Procedure

Variable: Resid

Extreme Observations

-----Lowest-----		-----Highest-----	
Value	Obs	Value	Obs
-5.66984	10	5.64142	73
-5.42743	94	6.12045	66
-5.21499	78	6.89125	98
-5.12874	81	6.91954	57
-4.77373	26	7.00585	82

Ensayo Clinico Artritis

12:20 Friday, April

2, 2004 213

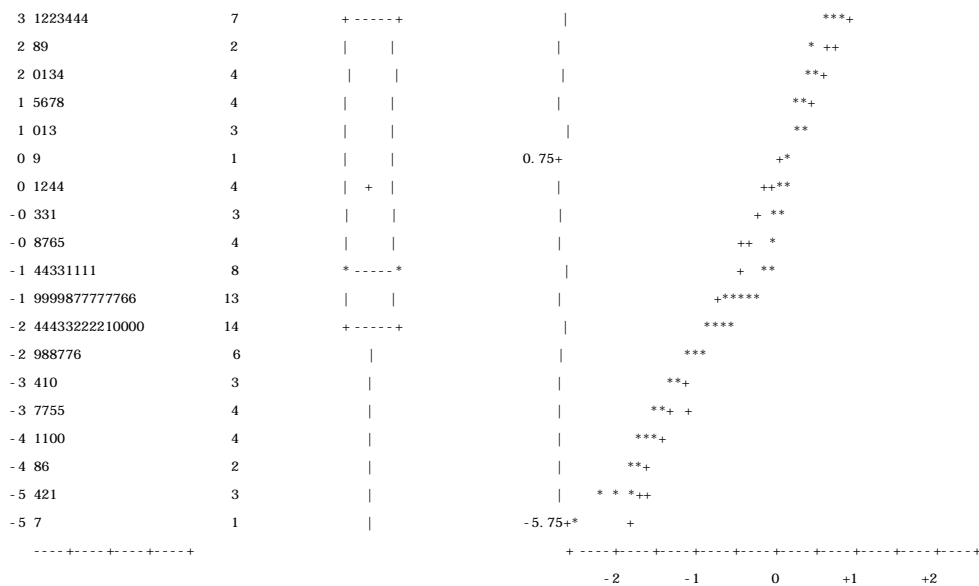
resids from ar model, quadratic

----- drug=2 -----

The UNIVARIATE Procedure

Variable: Resid

Stem Leaf	#	Box plot	Normal Probability Plot
7 0	1		7.25+ ++ *
6 99	2		++*
6 1	1		*++
5 6	1		*+
5 2223	4		***+
4 68	2		**+
4 024	3		**+
3 55566669	8		***



Ensayo Clinico Artritis

12: 20 Friday, April

2, 2004 214

resids from ar model, quadratic

----- drug=3 -(New)-----

The UNIVARIATE Procedure

Variable: Resid

Moments

N	147	Sum Weights	147
Mean	0	Sum Observations	0
Std Deviation	2.65747094	Variance	7.06215179
Skewness	0.55616443	Kurtosis	2.00406524
Uncorrected SS	1031.07416	Corrected SS	1031.07416
Coeff Variation	.	Std Error Mean	0.21918451

Basic Statistical Measures

Location		Variability	
Mean	0.00000	Std Deviation	2.65747
Median	-0.11833	Variance	7.06215
Mode	.	Range	14.97211
		Interquartile Range	2.51951

Tests for Location: Mu0=0

Test	-Statistic-	-----p Value-----	
Student's t	t	0	Pr > t 1.0000
Sign	M	-3.5	Pr >= M 0.6208
Signed Rank	S	-296	Pr >= S 0.5688

```

----- drug=3 -----
The UNIVARIATE Procedure
Variable: Resid

Tests for Normality

Test          --Statistic--   ----p Value-----

Shapiro-Wilk      W      0.942586   Pr < W   <0.0001
Kolmogorov-Smirnov D      0.105516   Pr > D   <0.0100
Cramer-vonMises   W-Sq   0.356743   Pr > W-Sq <0.0050
Anderson-Darling  A-Sq   2.379601   Pr > A-Sq <0.0050

Quantiles (Definition 5)

Quantile      Estimate

100% Max      8.431601
99%           8.365645
95%           4.843792
90%           2.676253
75% Q3        1.126239
50% Median    -0.118334
25% Q1        -1.393273
10%           -2.970641
5%            -3.779646
1%            -6.494662
0% Min        -6.540509
    
```

2, 2004 216 Ensayo Clinico Artritis 12:20 Friday, April

resids from ar model, quadratic

```

----- drug=3 -----
The UNIVARIATE Procedure
Variable: Resid

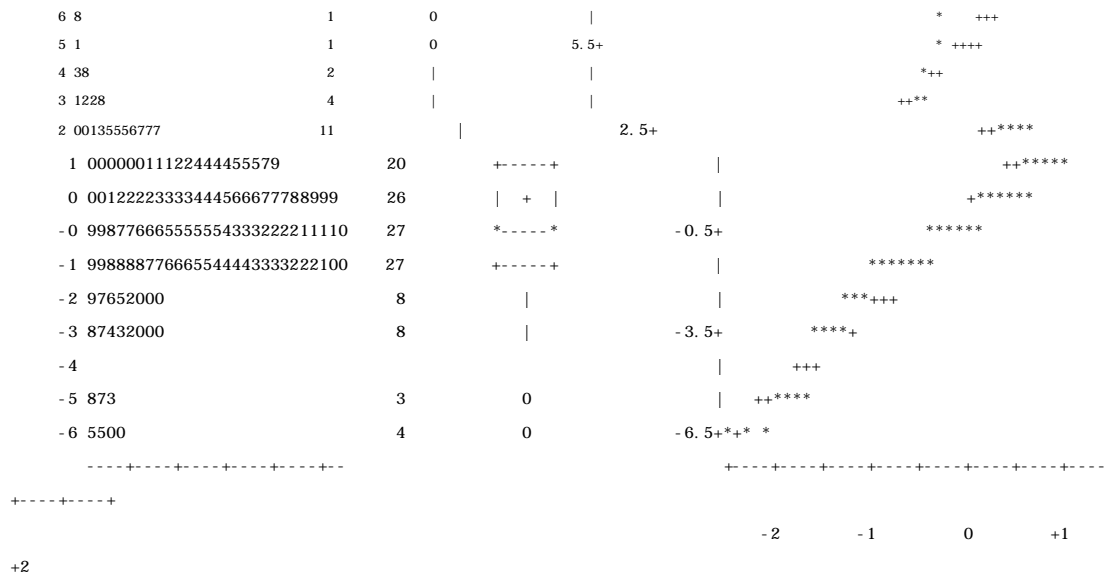
Extreme Observations

-----Lowest-----   -----Highest-----

Value   Obs      Value   Obs

-6.54051   2      7.07628   71
-6.49466   86      7.72096   92
-6.01759   44      8.01033   134
-5.97174   128     8.36565   113
-5.77905   23      8.43160   50

Stem Leaf      #      Boxplot      Normal Probability Plot
8 044          3      0      8.5+      * * *
7 17          2      0      |      **
    
```



Ensayo Clinico Artritis

12: 20 Friday, April

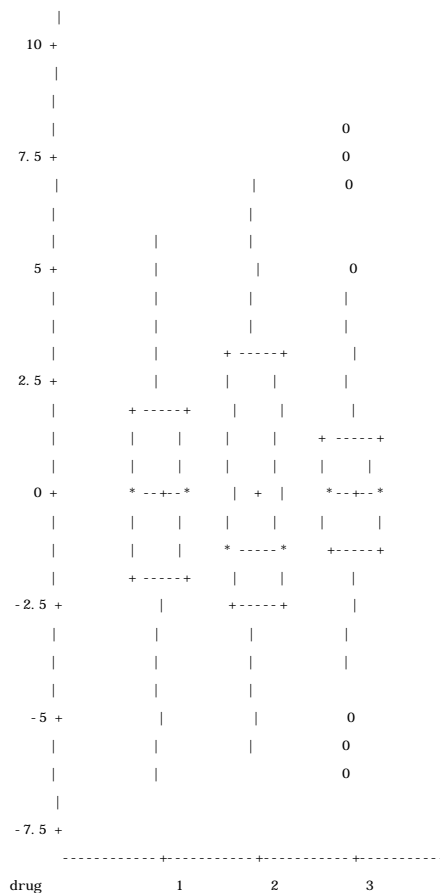
2, 2004 217

resids from ar model, quadratic

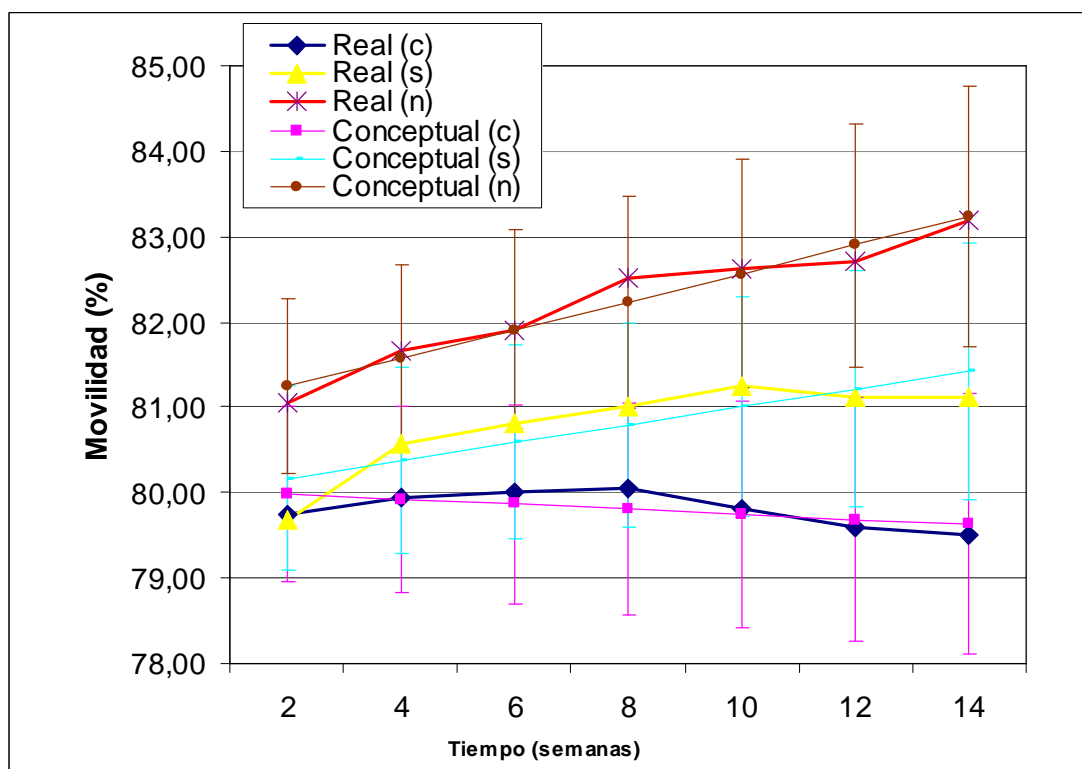
The UNIVARIATE Procedure

Variable: Resid

Schematic Plots



Tras el análisis de los residuos respecto de la predicción, se obtienen los resultados medios de las estimaciones para cada factor del modelo conceptual y los intervalos de confianza para la predicción, que se han representado en la siguiente gráfica frente a los valores reales:



Puede observarse que los datos reales y estimados medios del modelo conceptual son bastante parecidos. La precisión del semi-intervalo de confianza estimado por SAS es de aproximadamente $\pm 1,3\%$, es decir el intervalo de confianza es aproximadamente:

$$IC = Media \pm 1,3\%.$$

3.4.10.3 CONSTRUCCIÓN Y SIMULACIÓN DEL MODELO CONCEPTUAL

Se ha construido un modelo computacional a partir de los parámetros estimados en el modelo conceptual, mediante la escritura de 6 “scripts” de Visual Basic que el simulador de LeanSim puede interpretar. El modelo computacional también se ha construido en lenguaje SAS.

Los “scripts” de la simulación en LeanSim y la sintaxis de SAS, pueden verse en el CD anexo. Constituyen la base del modelo computacional que se ejecutará hasta obtener un resultado satisfactorio. La mayoría de simulaciones de éste y otros capítulos se han realizado mediante SAS, por la simplificación que presenta durante su especificación y la inmediatez en la realización de los cálculos estadísticos, aunque se han realizado pruebas comparativas con LeanSim, no detectándose diferencias en los resultados obtenidos.

Al ejecutar el modelo computacional, por la propia naturaleza del método de Monte Carlo, cada réplica dará un resultado diferente e independiente debido a las componentes aleatorias del modelo. Es posible repetir el resultado de una réplica utilizando la misma semilla de aleatorización por ser un método pseudo-aleatorio.

El modelo computacional puede ejecutarse de 2 maneras:

RÉPLICAS: Se ejecutará la simulación repetidamente, con un cierto tamaño muestral en cada réplica. Cada réplica del ensayo clínico simulado es una posible solución dentro de la distribución de probabilidades de todas las soluciones posibles. Holford y col. (1999) en la guía para la simulación de ensayos clínicos “Simulation in Drug Development: Good Practices”, dedican un apartado a las réplicas de simulación, donde afirman que su número debe estar basado en la precisión y los objetivos del estudio. Para analizar cada

réplica de simulación se usarán los mismos análisis estadísticos planeados para el ensayo clínico real, que en general corresponden a pruebas estadísticas inferenciales y que podrá variar en función del diseño del ensayo.

Si se supone que se realizan R réplicas independientes de una variable aleatoria Y de interés (variable de respuesta clínica, diferencia de respuesta basal final, variable de respuesta en un periodo de tiempo, etc), se denominará Y_{ri} a la i ésima observación de la réplica r , para $i = 1, 2, \dots, n_r$ y $r = 1, 2, \dots, R$

Si para un número de réplicas r , $Y_{r1}, Y_{r2}, \dots, Y_{rn}$ es una secuencia autocorrelacionada de una variable de interés Y , obtenida tras la simulación, para diferentes réplicas Y_{ri} y Y_{rj} son valores estadísticamente independientes.

Se define como media muestral de las réplicas de simulación de Y a la expresión:

$$\hat{q}_r = \frac{1}{n_r} \sum_{i=1}^{n_r} Y_{ri}, \quad r = 1, 2, 3, \dots, R \quad (21)$$

Por lo que habrá R muestras y R medias muestrales. La media muestral general es:

$$\hat{q} = \frac{1}{R} \sum_{r=1}^R \hat{q}_r \quad (22)$$

La varianza muestral de la simulación se define como:

$$\hat{s}^2(\hat{q}) = \frac{1}{R} \sum_{r=1}^R \frac{(\hat{q}_r - \hat{q})^2}{R-1} \quad (23)$$

El intervalo de confianza basado en una distribución normal, de θ para un cubrimiento del $100(1-\alpha)\%$ se obtiene de:

$$\hat{q} - \frac{t_{\alpha/2, R-1} S}{\sqrt{R}} \leq q \leq \hat{q} + \frac{t_{\alpha/2, R-1} S}{\sqrt{R}} \quad (24)$$

Donde la desviación estándar se define como:

$$S = \sqrt{R} \hat{s}(\hat{q}). \quad (25)$$

Se define el semi-intervalo de confianza de q como:

$$g = \frac{t_{a/2, R-1} S}{\sqrt{R}}. \quad (26)$$

Si se asume un criterio de error especificado según el intervalo de confianza $1-a$ como:

$$P(|\hat{q} - q| < e) \geq 1 - a \quad \text{y} \quad g - \frac{t_{a/2, R-1} S_0}{\sqrt{R}} \leq e \quad (27)$$

$$\text{De lo que se deduce que: } R \geq \left(\frac{t_{a/2, R-1} S_0}{e} \right)^2 \quad (28)$$

Algunos autores como Guasch y col. (2002) sugieren fórmulas para calcular el número de réplicas de simulación según la precisión requerida (aceptabilidad del modelo) de la variable de respuesta y el análisis de una serie de réplicas previas (simulaciones piloto). En variables de respuesta de tipo continuo el número de réplicas puede calcularse como:

$$R = k \left[\frac{(g/2)}{e} \right]^2 \quad (29)$$

Donde k es el número de réplicas inicial realizado, g es el semi-intervalo de confianza (26) de la variable de respuesta, e es la precisión deseada (1%, 5%, 10%, etc). Si se desea, en el caso de la artritis reumatoide, saber cuál es el número de réplicas adecuado para una precisión del 10% y se ha visto en una prueba con 10 simulaciones que el semi-intervalo de confianza es de un 1% de la variable movilidad, se calculará el número de réplicas como:

$$R = 10 \left[\frac{(1/2)}{0,1} \right]^2 = 250 \text{ réplicas}$$

Esta forma de calcular el número de réplicas es complejo, ya que en muchas situaciones es difícil estimar el semi-intervalo de confianza, ya que dependerá

del modelo y de la forma en que se establezca su cálculo. Es evidente que a medida que se va aumentando el número de réplicas y se van acumulando casos en la matriz de datos, el tamaño del intervalo de confianza disminuye.

Para una variable de tipo binario puede calcularse como:

$$R = k \left[\frac{g}{p} \right]^2 \quad (30)$$

CONVERGENCIA DE LA VARIABILIDAD: El modelo se ejecuta aumentando el número de individuos para disminuir la variabilidad de la variable de respuesta hasta que converge a un valor (valor de convergencia) (Law y Kelton, 1999). Conceptualmente es equivalente a simular toda la población de entidades de procedencia del modelo de simulación y puede resultar útil en el caso de estudios epidemiológicos donde se desee estimar la incidencia de una determinada enfermedad en toda la población, o en el caso de estudios observacionales.

Un método combinado a partir de los 2 anteriores es el de simular una sola réplica de gran tamaño y dividirla en subconjuntos más pequeños de tamaño adecuado, utilizando remuestreo bootstrap.

Es complejo establecer cuándo deben realizarse réplicas de simulación de tamaño parecido al de los datos reales o bien una gran réplica grande (muchos pacientes), quizás empleando mucho tiempo en su simulación. Durante la simulación, debe imitarse al máximo todos los aspectos del ensayo clínico real y prácticamente siempre deberían realizarse réplicas de simulación a fin de tener una idea de la variabilidad del ensayo clínico real. Uno de los aspectos que deben conservarse, es utilizar tamaños muestrales de réplicas parecidos a los del ensayo real, aunque en algunas situaciones en las que se estudia la sensibilidad o robustez de la simulación si deberán utilizarse

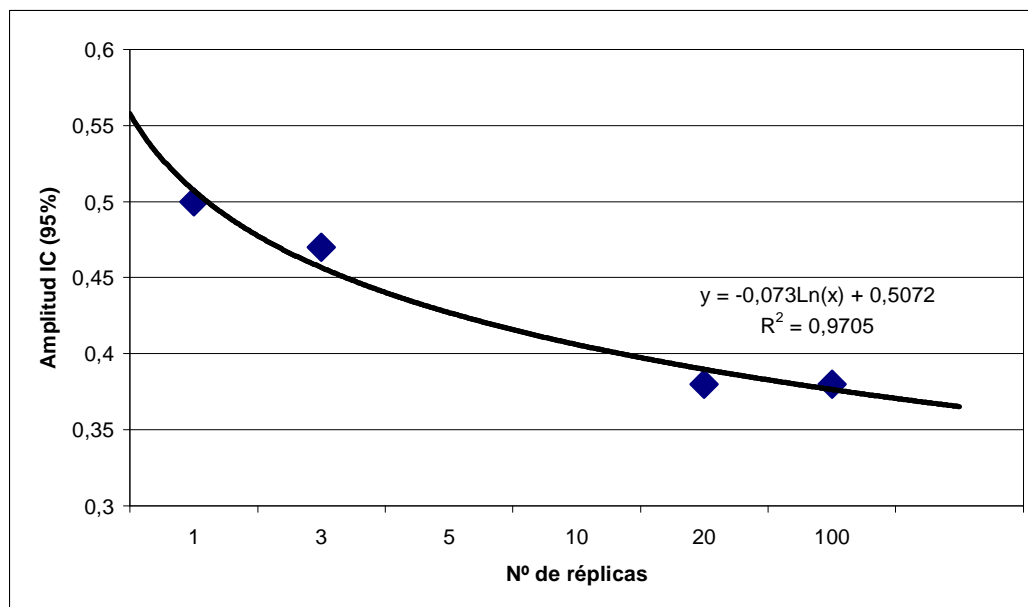
réplicas de gran tamaño, donde se establezca la convergencia de la variabilidad.

En los ensayos clínicos existen condicionantes insalvables como el número de centros, duración, diseño, variable de respuesta, etc. y se ha de procurar diseñar un ensayo clínico simulado (gracias a simulaciones replicadas) lo mejor posible.

3.4.10.4 PRUEBAS DE SIMULACIÓN. AJUSTE DEL MODELO COMPUTACIONAL.

Se han realizado diferentes pruebas con réplicas y convergencia de variabilidad con LeanSim y el modelo construido. Una de las observaciones de rendimiento computacional del simulador es que son necesarios 3 minutos para generar 3000 entidades (pacientes) del modelo anterior en un ordenador con procesador Pentium IV a 2,5MHz y 512 Mb de memoria RAM. Este tiempo en un paquete estadístico como SAS, que actúa como un intérprete, es teóricamente superior, aunque como el modelo de simulación es sencillo no se han detectado diferencias apreciables.

También puede realizarse un estudio previo de cómo varía la amplitud de los intervalos de confianza para la respuesta media (% movilidad) y así determinar cuál es el número de réplicas necesarias. En el presente estudio se ha realizado un estudio con 1, 3, 20 y 100 réplicas de 60 pacientes para conocer cuál es el número de réplicas requerido, relacionando la amplitud y el número de réplicas mediante un modelo logarítmico:



En la gráfica anterior puede observarse que con 20 réplicas es suficiente para que se establezca la variabilidad de la variable de respuesta hasta un valor de $IC = \text{Media del \%Movilidad} \pm 0,38\%$ que es el valor al que convergen las simulaciones. Los intervalos de confianza corresponden al método basado en la suposición de normalidad, descrito anteriormente.

En las simulaciones "definitivas" el modelo estimado anteriormente ha sido simulado 100 veces y en cada réplica se han simulado 60 pacientes (20 pacientes por grupo de tratamiento), lo cual hace un total de 6000 entidades generadas. Se ha medido el tiempo que tarda el simulador en generar una entidad y éste ha sido de unos 0,06 segundos (6 minutos en total) en un ordenador tipo Pentium IV de 2,4MHz y 512 MB de memoria RAM.

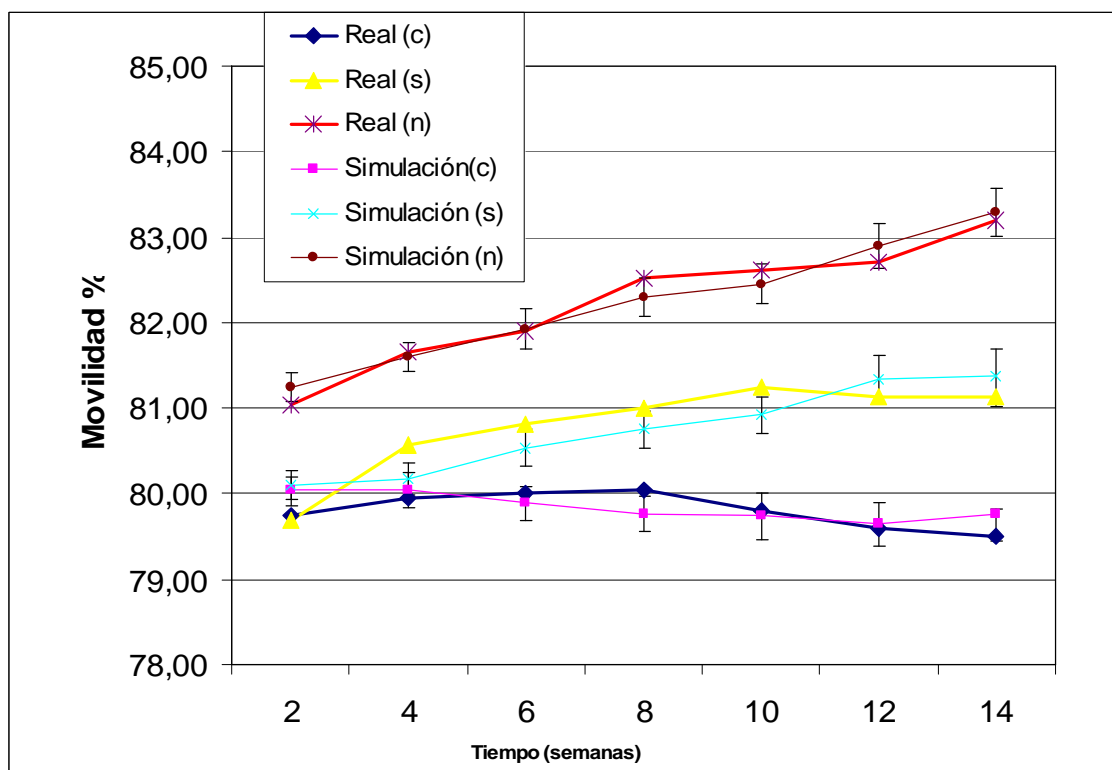
Es enormemente costoso en cuanto a tiempo exportar los resultados obtenidos en la simulación al paquete estadístico SAS para su análisis posterior, debido a que el simulador de LeanSim genera unos ficheros de datos que deben ser transformados a un fichero que sea capaz de reconocer el entorno SAS. Es por ello que en las pruebas de verificación de los modelos computacionales, sólo se presentan los resultados de un porcentaje de las réplicas de simulación y no de todas ellas.

3.4.10.5 VALIDACIÓN Y VERIFICACIÓN DEL MODELO COMPUTACIONAL

Tras el análisis de los datos simulados por el modelo, se han obtenido los intervalos de confianza de los efectos medios de los factores para la variable "porcentaje de movilidad". Los intervalos presentados corresponden a los intervalos percentil bootstrap (básico y corregido para el sesgo, BCa) descritos, por ejemplo, en Efron y Tibshirani (1993), con un recubrimiento nominal del 95%. Tras la simulación, los resultados comparativos de los diferentes intervalos de confianza estimados se han presentado en la siguiente tabla:

		DATOS REALES	MODELO ESTIMADO PROC MIXED SAS			SIMULACION CON LeanSim Y/O SAS N= 3000 (1 réplica)					SIMULACION CON LeanSim Y/O SAS N= 60 (100 réplicas)					
					Intervalo de confianza asintótico normal			Intervalo de confianza Percentil Bootstrap		Intervalo de confianza Percentil Bootstrap BCa			Intervalo de confianza Percentil Bootstrap		Intervalo de confianza Percentil Bootstrap BCa	
Drug	Tiempo	Movilidad media (%)	Movilidad media (%)	Error estandar	Limite inferior 95%	Limite superior 95%	Movilidad media (%)	Limite inferior 95%	Limite superior 95%	Limite inferior 95%	Limite superior 95%	Movilidad media (%)	Limite inferior 95%	Limite superior 95%	Limite inferior 95%	Limite superior 95%
c	2	79,75	79,98	0,52	78,96	81,01	80,04	79,87	80,18	79,88	80,22	80,0026	79,7623	80,2820	79,6968	80,3084
c	4	79,95	79,93	0,55	78,84	81,01	80,05	79,80	80,24	79,84	80,26	79,7745	79,4995	80,0230	79,4590	80,0899
c	6	80,00	79,87	0,58	78,72	81,01	79,89	79,69	80,05	79,69	80,09	79,8103	79,5111	80,1298	79,4327	80,1879
c	8	80,05	79,81	0,62	78,58	81,03	79,77	79,59	79,95	79,57	79,98	79,7713	79,4617	80,1613	79,4325	80,1101
c	10	79,80	79,75	0,67	78,44	81,06	79,74	79,52	80,06	79,47	80,01	79,6300	79,2715	79,9072	79,2863	79,9738
c	12	79,60	79,69	0,71	78,29	81,09	79,65	79,47	79,92	79,41	79,44	79,4413	79,6423	79,1078	79,0787	79,8039
c	14	79,50	79,63	0,76	78,13	81,13	79,76	79,52	80,02	79,70	80,07	79,4667	79,0642	80,0202	79,0019	79,9315
s	2	79,69	80,17	0,55	79,08	81,25	80,11	79,95	80,22	79,95	80,28	79,9510	79,6989	80,1638	79,6839	80,2181
s	4	80,56	80,38	0,56	79,28	81,47	80,18	79,97	80,34	79,99	80,37	80,4047	80,0572	80,8068	80,0712	80,7383
s	6	80,81	80,58	0,58	79,45	81,72	80,54	80,36	80,75	80,32	80,75	80,5859	80,3995	80,8769	80,3435	80,8283
s	8	81,00	80,79	0,61	79,59	81,99	80,76	80,54	80,95	80,55	80,98	80,5859	80,2563	81,1176	80,2563	81,1176
s	10	81,25	81,00	0,65	79,72	82,29	80,93	80,73	81,13	80,72	81,15	80,8716	80,6009	81,2483	80,5120	81,2312
s	12	81,13	81,21	0,71	79,82	82,61	81,34	80,97	81,61	81,06	81,63	80,8158	80,3783	81,1732	80,3766	81,2549
s	14	81,13	81,42	0,77	79,92	82,93	81,37	81,07	81,72	81,04	81,71	81,0164	80,6258	81,4628	80,6179	81,4150
n	2	81,05	81,25	0,52	80,22	82,28	81,24	81,13	81,40	81,07	81,41	81,4291	81,1439	81,7220	81,1328	81,7254
n	4	81,67	81,58	0,56	80,49	82,67	81,60	81,45	81,77	81,42	81,77	81,6880	81,2633	81,9954	81,2855	82,0906
n	6	81,90	81,91	0,59	80,74	83,07	81,93	81,74	82,19	81,70	82,16	81,8851	81,5198	82,2611	81,5526	82,2176
n	8	82,52	82,24	0,63	80,99	83,48	82,30	82,08	82,30	82,07	82,53	82,0380	81,6166	82,3807	81,6369	82,4391
n	10	82,62	82,57	0,68	81,24	83,90	82,45	82,28	82,76	82,21	82,68	82,7037	82,2114	83,0370	82,2828	83,1246
n	12	82,71	82,90	0,72	81,47	84,32	82,90	82,69	83,20	82,63	83,17	83,0626	82,3817	83,4836	82,5256	83,5996
n	14	83,19	83,23	0,77	81,71	84,75	83,29	83,06	83,59	83,01	83,56	83,4490	82,7642	83,8810	82,8702	84,0278

A partir de la tabla anterior se han construido una serie de gráficos para el estudio de los resultados de la simulación. La siguiente gráfica representa los datos reales y los simulados, con los intervalos de confianza bootstrap (intervalos BCa) calculados a partir de las réplicas de simulación:



El resultado del análisis de la varianza para la simulación de 3000 pacientes (1 réplica) indica que existen diferencias estadísticamente significativas entre tratamientos, tiempo y la interacción entre tiempo y tratamiento al igual que ocurriría con los datos reales. Estos resultados son:

1) 3000 pacientes y 1 sola réplica:

Type 1 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
TRACTAME	2	15E3	215.23	<.0001
time	1	2977	131.41	<.0001
time*TRACTAME	2	15E3	67.67	<.0001

Diferencias significativas entre todos los factores estudiados.

2) Con 60 pacientes y 1ª réplica de simulación:

Type 1 Tests of Fixed Effects

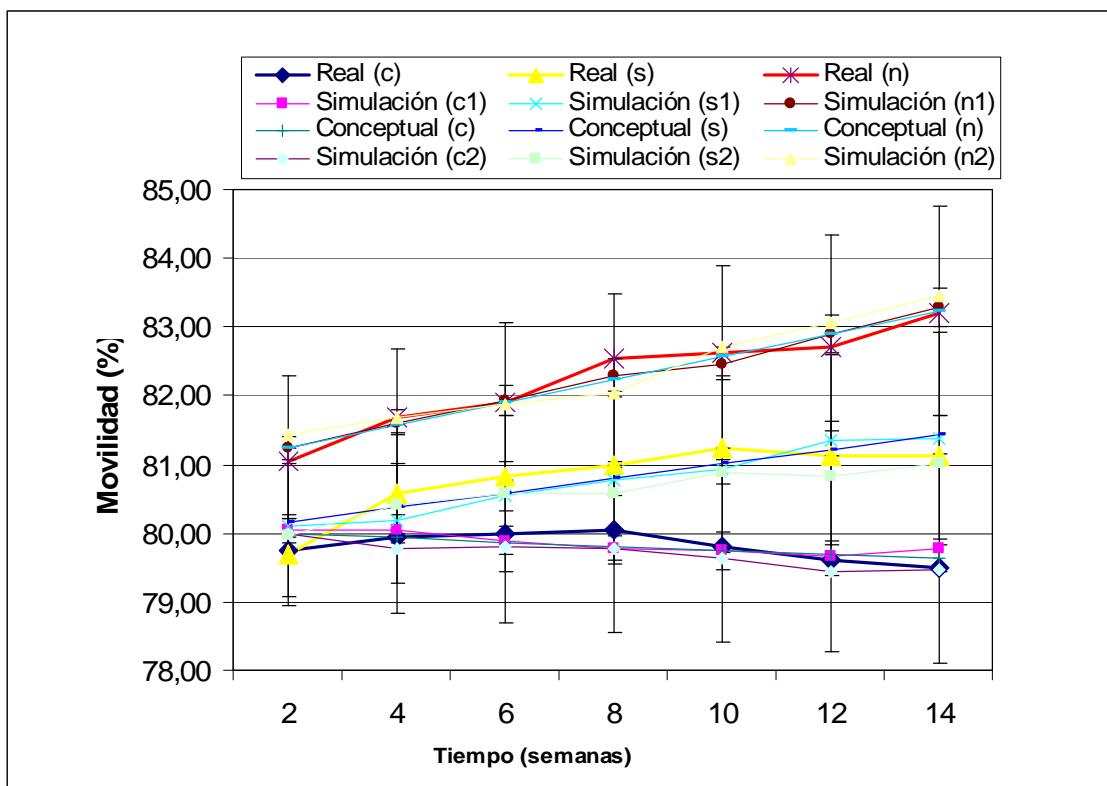
Effect	Num DF	Den DF	F Value	Pr > F
TRACTAME	2	300	7.02	0.0010
time	1	57	0.01	0.9043
time*TRACTAME	2	300	3.89	0.0216

3) Con 60 pacientes y 2ª réplica de simulación:

Type 1 Tests of Fixed Effects

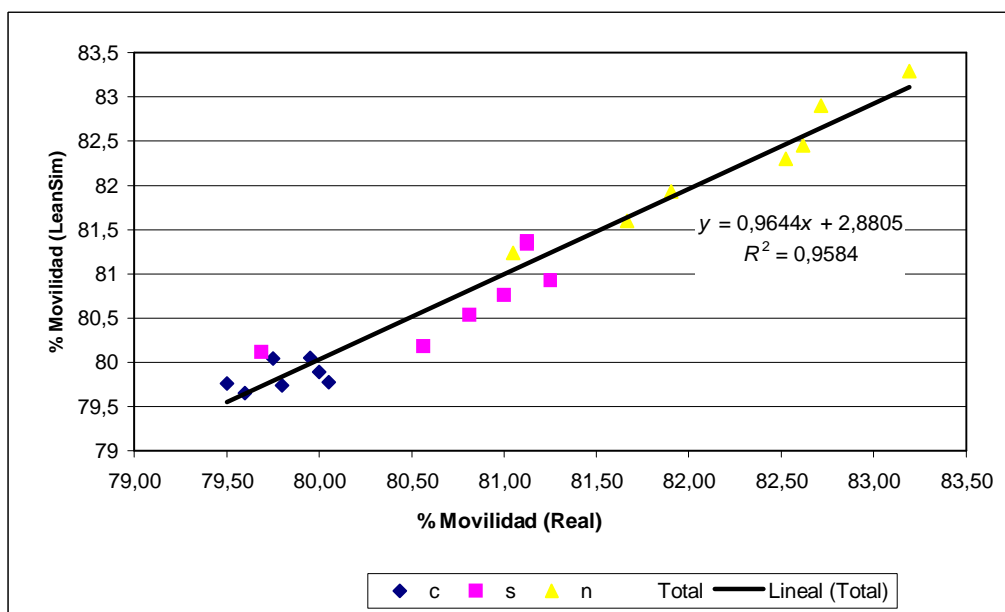
Effect	Num DF	Den DF	F Value	Pr > F
TRACTAME	2	300	6.86	0.0012
time	1	57	0.37	0.5477
time*TRACTAME	2	300	0.40	0.6678

Como se puede comprobar, según la cantidad de entidades generadas y el número de réplicas generadas, los intervalos de confianza son diferentes, no así las medias como puede verse en la siguiente gráfica donde se han representado, los valores reales, el modelo conceptual (e intervalos de confianza de la predicción calculados por SAS) y las simulaciones (1:3000 pacientes/1 réplica) y (2: 60 pacientes/100 réplicas) (e intervalos de confianza bootstrap):



Cuando se genera una gran cantidad de entidades se produce una situación de falso positivo o Error de tipo I, es decir que se detectan diferencias entre los grupos de tratamiento cuando en realidad no es así en el sistema real.

Otro método de validación del modelo aconsejado por autores como Kleijnen (2003) es la relación de ajuste lineal entre los valores medios reales y los valores medios según la simulación del modelo presentado anteriormente, obteniéndose las siguientes gráficas:



El coeficiente de determinación R^2 es extremadamente alto e indica la gran concordancia entre los datos obtenidos mediante el modelo computacional en LeanSim y los datos reales.

VERIFICACIÓN DEL MODELO COMPUTACIONAL BASADO EN PARÁMETROS Y SIGNIFICACIÓN ESTADÍSTICA DE LAS RÉPLICAS DE SIMULACIÓN (VMCBP)

A fin de verificar el modelo computacional, se han estudiado los intervalos de confianza de los parámetros fijos y parámetros (varianzas y covarianzas) de la parte aleatoria del modelo conceptual lineal mixto, presentados anteriormente, así como el error y las significaciones estadísticas para los factores principales (tratamiento, tiempo e interacción tiempo x tratamientos). Dichos parámetros han sido estimados mediante SAS en las réplicas de simulación generadas por LeanSim o SAS a partir del modelo computacional construido. Se propone estudiar si la variabilidad de las réplicas de simulación generadas concuerda con la del modelo estimado.

Se presentan a continuación los resultados de 10 réplicas de simulación con 60 pacientes cada uno (20 pacientes por grupo de tratamiento). En la primera fila de la tabla se presentan los resultados del modelo conceptual lineal mixto real, estimado en la primera parte de este capítulo.

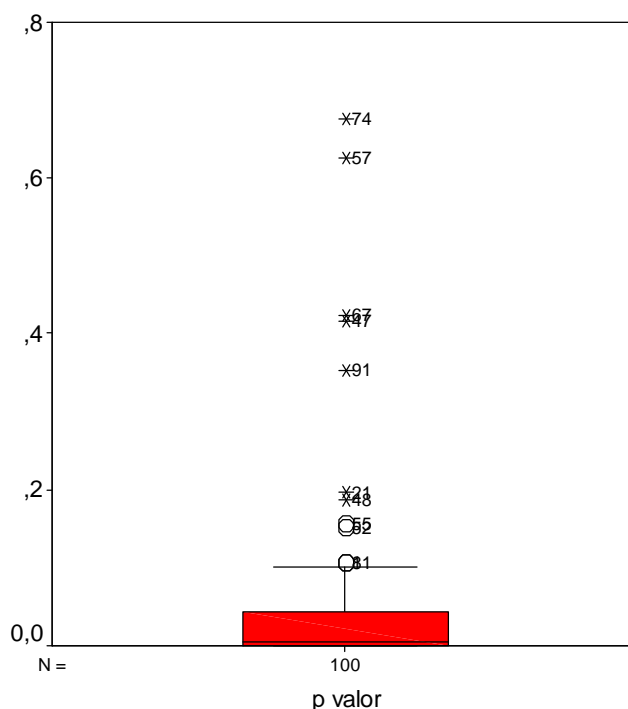
Nº de réplica	Parámetro fijo del término independiente 80,92 [79,7;82,1]	Parámetro aleatorio del término independiente con distribución N(0; 0,99[0,51;7,00]) \hat{S}	Parámetro fijo del factor tratamiento β_i (-0,8998[-2,37;0,57] c; -0,9954[-2,55;0,57] s; 0 n)	Parámetro fijo del factor tiempo 0,165 [0,04;0,29]	f_i (-0,1958 [-0,36;-0,03] c, -0,06197[-0,24;-0,11] s, 0 n)	$e_{ij} \sim$ N(0; 2,83 [2,65;3,05])	Tests of Fixed Effects Tratamiento ($p = <.0001$) Tiempo (0,0376) Tiempo*Tratamiento (0,0604)	Verosimilitud: -LL (2000)
1	80,9132	1,2922	-0,4999 -1,0920 0	0,2154	-0,3664 -0,1670 0	2,91	0,0010 0,9043 0,0216	2228
2	81,1472	1,2573	-1,5291 -1,6146 0	0,06920	-0,09350 0,01424 0	2,66	0,0012 0,5477 0,6678	2138,7
3	81,2282	<0,0001*	-0,8034 -1,1822 0	0,2277	-0,2157 -0,3257 0	3,101	<.0001 0,4296 0,0150	2236,3
4	80,5560	<0,0001*	-0,9809 -1,5365 0	0,1677	-0,2330 0,07491 0	2,892	0,0011 0,0161 0,0479	2191,5
5	80,6149	1,2111	-0,3147 0,2515 0	0,2003	-0,2357 -0,03955 0	2,878	0,0129 0,0100 0,0909	2196,1
6	81,6373	2,0092	-1,9212 -2,0876 0	0,1819	-0,2444 -0,02159 0	2,805	<.0001 0,0966 0,1680	2203,5
7	81,3624	1,3710	-0,7441 -1,3900 0	0,08824	-0,1331 0,1234 0	2,713	0,0531* 0,2000 0,0957	2166,8
8	80,9338	1,2515	-0,8593 -0,1720 0	0,1274	-0,1989 -0,1072 0	2,927	0,0089 0,7316 0,3396	2214,5
9	81,2813	2,1835	-0,9666 -1,2667 0	0,1896	-0,09401 -0,01290 0	2,611	0,0529* 0,0058 0,7089	2156,2
10	80,6600	1,2831	-0,7668 -1,3134 0	0,2154	-0,2224 -0,04723 0	2,846	0,0053 0,0177 0,0996	2191,7

El resultado de la prueba VMCBP indica que existen unos pocos parámetros (marcados con *) donde la réplica del modelo computacional presenta diferencias con respecto al modelo conceptual estimado, o está fuera del intervalo de confianza.

Al respecto de las diferencias estadísticas para la variable simulada movilidad entre tratamientos farmacológicos, se han encontrado en 2 (20%) de las 10 réplicas de simulación que no presentan diferencias estadísticamente significativas ($p=0,0531$ y $p=0,0529$), aunque por un escaso margen de probabilidad. Aunque es necesario realizar una mayor cantidad de réplicas de

simulación, esta primera aproximación indica que el modelo computacional verifica razonablemente el modelo conceptual lineal mixto estimado anteriormente.

Se han calculado los p-valores para el factor tratamiento para cada una de las 100 réplicas de simulación. El gráfico de dispersión, en forma de diagrama de caja, es:



En el diagrama anterior puede observarse que los p-valores son coherentes con lo esperable cuando la hipótesis alternativa es cierta, ya que su distribución es muy asimétrica, dándose una acumulación de valores cercanos a 0, muy pequeños, aunque también pueden darse p-valores grandes correspondientes a los casos en que se produce error de tipo II.

Si el intervalo de confianza, en el modelo conceptual, abarca aproximadamente un 95% de los parámetros estimados en las 100 réplicas de simulación (p-valor para el factor tratamiento: $0,04981[0,02688; 0,07274]$, indicando que existen diferencias estadísticamente significativas en este factor para la mayoría de réplicas estudiadas, concretamente en 88% de los casos $p < 0,05$, que es el principal factor en éste ensayo clínico), es un argumento a favor de la

verificación del modelo computacional, ya que la simulación no es más que una ejecución aleatoria independiente del modelo computacional. Es tanto más creíble cuanto más podamos confiar en la validez de los intervalos de confianza que ha calculado SAS (Monleón y col., 2005).

En realidad, si no se cumplen las condiciones de normalidad, como sucede en el ejemplo de datos propuesto, estos intervalos de confianza tampoco tienen un recubrimiento o confianza del 95%.

VERIFICACIÓN DEL MODELO COMPUTACIONAL BASADO EN PARÁMETROS Y SIGNIFICACIÓN ESTADÍSTICA DE LAS RÉPLICAS DE SIMULACIÓN (VMCBP)

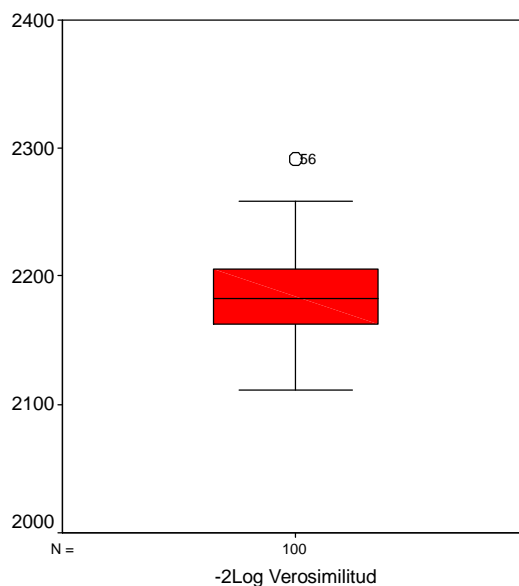
Otro método que se presenta para la verificación del modelo computacional, es utilizar la verosimilitud de modelo simulado, estimada mediante SAS en cada réplica de simulación.

La verosimilitud estimada en SAS para el modelo conceptual lineal mixto, presentado en el apartado de estimación del modelo conceptual, es de 2000. Esta verosimilitud ha sido calculada utilizando la base de una distribución normal y su fórmula es (Hartley y Rao, 1967):

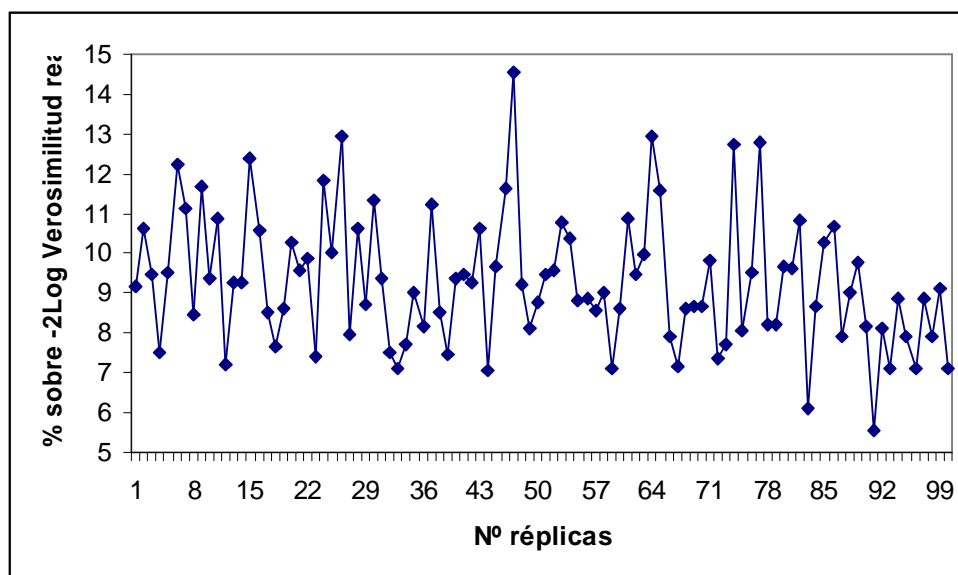
$$L = (2\pi)^{-\frac{n}{2}} |V|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(y - Xq)' V^{-1}(y - Xq)\right] \quad (31)$$

Donde $y \sim N(Xq, V)$ con $V = Z_{GZ'+I\sigma_e^2}$

Se ha estimado el intervalo de confianza normal de la verosimilitud (-LL) de las 100 réplicas de simulación en 2185,623[2179,1094; 2192,1365]. Para la verosimilitud (-LL) también se ha calculado su diagrama de dispersión:



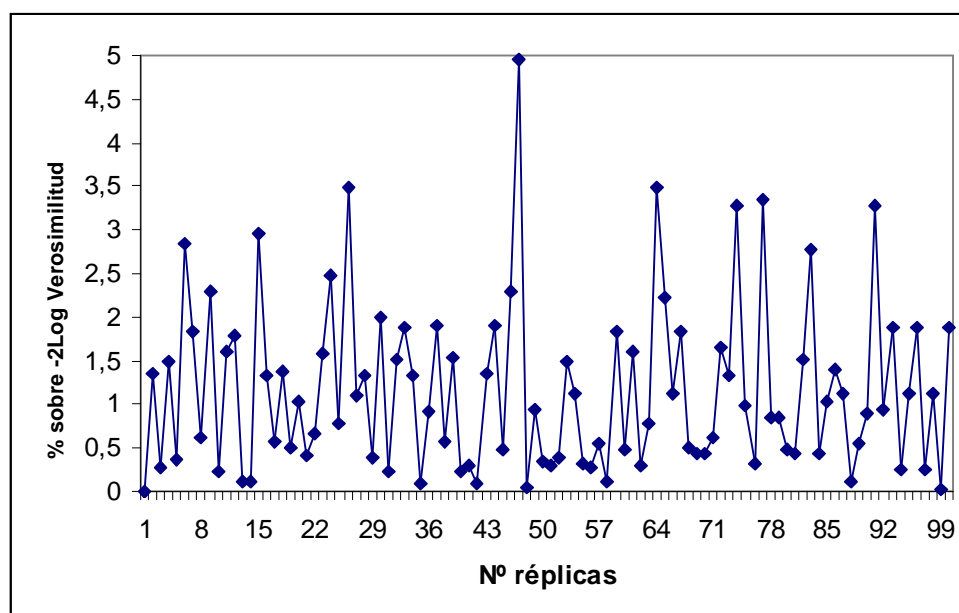
A continuación se presenta la variación porcentual de la verosimilitud con respecto a la verosimilitud del modelo real para las 100 réplicas de simulación realizadas:



En el gráfico anterior puede observarse que la variación porcentual de la verosimilitud oscila entre un 6 a un 14% lo que se considerará normal para las réplicas de simulación que verifican el modelo conceptual, aunque este porcentaje debe verificarse en posteriores experimentos. La verosimilitud

diferencial es un indicador válido a fin de verificar las simulaciones, de una forma rápida y sencilla. Se recomienda utilizar un rango de oscilación para la variación porcentual de la verosimilitud entre la obtenida en el modelo conceptual estimado y la de la simulación entre un 4 y un 18% y a esta prueba se la denominará Prueba de Verificación del Modelo Computacional Basada en la Verosimilitud (PVMCBV) y ha sido presentada en Monleón y col.(2005).

Se ha podido calcular también que la verosimilitud (-LL) intra-réplicas oscila entre un 0 a 5%, como puede verse en el gráfico siguiente:



Este rango de variación (0-5%) se ha podido corroborar en diversos experimentos de simulación realizados mediante SAS con simulaciones de modelos lineales normales.

Se propone incluir en la siguiente versión del simulador LeanSim, durante el proceso de análisis de los resultados y de una forma automatizada los resultados de las pruebas PVMCBV y VMCBP. No se ha encontrado referencia a estos métodos dentro de los métodos de verificación de las simulaciones en la bibliografía consultada.

Parece adecuado también realizar verificaciones para las predicciones medias cuando se tienen trayectorias medias por paciente. Aunque la idea mejoraría conceptualmente si se realizara un intervalo de confianza simultáneo para todos los parámetros o para todas las trayectorias, y no parámetro a parámetro y periodo a periodo como se ha presentado anteriormente. Este problema no se da en la prueba de verosimilitud, ya que sólo se presenta un valor para cada réplica de simulación.

El profesor Esteban Vegas del Departamento de Estadística de la UB ha propuesto mejorar la imparcialidad del proceso de validación operacional utilizando validación cruzada. No ha habido tiempo de realizar pruebas suficientes para mencionar aquí los resultados, en un futuro se espera poder utilizar esta técnica en la simulación de ensayos clínicos.

Estas pruebas presentadas en este apartado de verificación son útiles sólo cuando el modelo conceptual queda totalmente especificado por una teoría matemática-estadística (como por ejemplo en un modelo mixto) que permite definir intervalos de confianza, verosimilitud, etc. En ocasiones, que suelen coincidir con los casos más interesantes de simular, el modelo conceptual y su implementación en forma de modelo operacional incluyen aspectos que no se contemplan en la teoría matemático-estadística como abandonos, tiempo entre visitas, hechos fortuitos, etc., que hacen que del modelo conceptual no se deriven directamente intervalos de confianza y otras propiedades estadísticas, teniendo en este caso gran importancia la validación y verificación basadas en aspecto y trazas comentadas en la revisión de métodos VV&A, en la que las verificaciones y validaciones a nivel gráfico cobran gran importancia, así como la opinión de los expertos en el tema y en las que el ensayo clínico presentado cumple también adecuadamente. De todas formas sí que puede tener interés en situaciones en las cuales, para determinada/as combinación/es de parámetros, el modelo conceptual se simplifique lo suficiente como para tener una solución analítica.

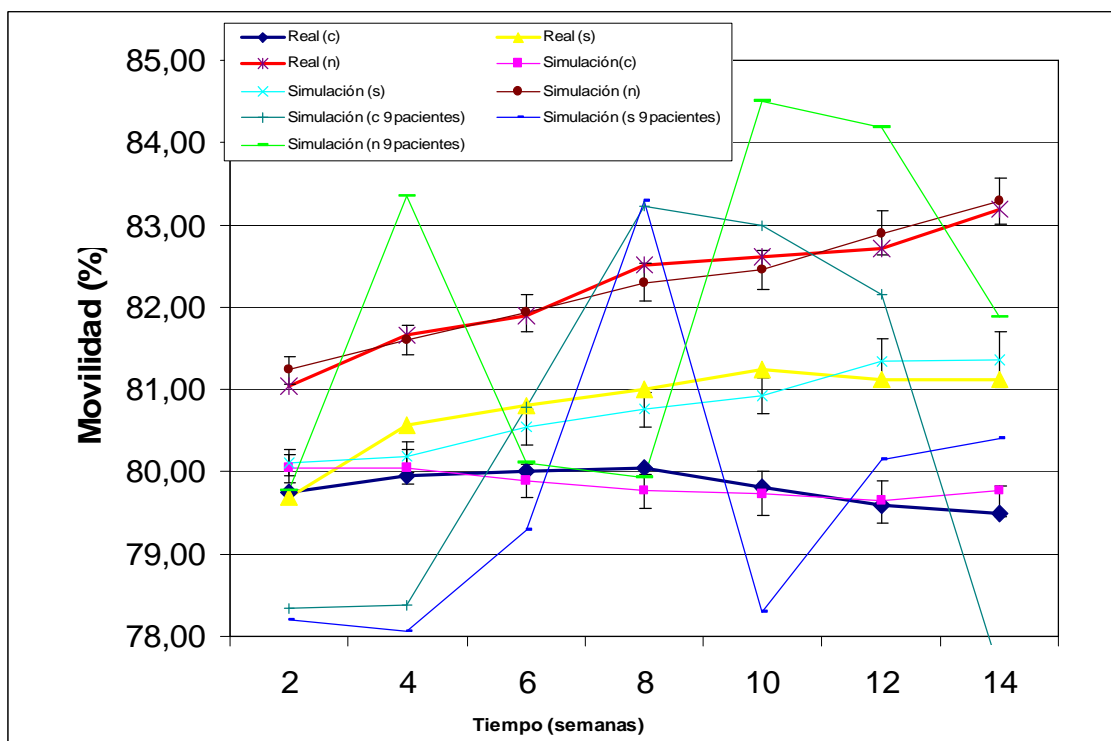
3.4.10.6 ANÁLISIS DE SENSIBILIDAD DEL MODELO COMPUTACIONAL

A fin de comprobar la robustez del modelo computacional y plantear posibles utilidades de la simulación, así como optimizar el ensayo clínico se presentan otros escenarios de simulación:

Escenario de falso negativo (Error de tipo II o *b*) / Falta de potencia de test

Se pueden simular situaciones conducentes a un posible error de tipo II, generando una menor cantidad de pacientes de los previstos en el cálculo del tamaño muestral del ensayo clínico, de tal manera que la variabilidad de las entidades generadas confunda a la prueba estadística e impida conocer si existen diferencias entre los efectos de los tratamientos debidos a su variabilidad inter-sujetos o inter-tratamiento.

Se ha simulado 1 réplica con sólo 9 pacientes (3 por cada grupo) y el resultado comparado con los datos reales y la simulación de la población de pacientes anterior ($n=60$, 100 réplicas) puede verse en la siguiente gráfica:



El resultado tras el análisis de la varianza de medidas repetidas en SAS indica que no existen diferencias significativas entre tratamientos ni interacción entre tratamiento y tiempo:

Type 1 Tests of Fixed Effects

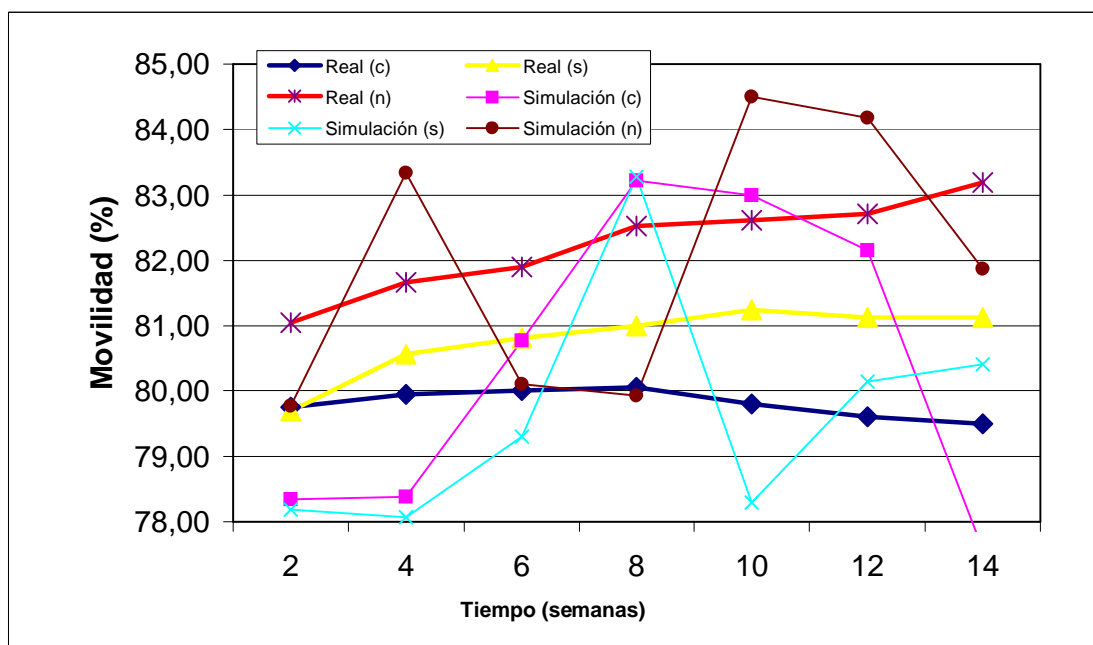
Effect	Num DF	Den DF	F Value	Pr > F
TRACTAME	2	45	3.03	0.0583
time	1	6	3.02	0.1330
time*TRACTAME	2	45	0.05	0.9486

Se ha simulado una sola réplica de 20 pacientes por grupo (Igual que datos reales). El resultado del análisis de la varianza es:

Type 1 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
TRACTAME	2	300	2.72	0.0675
time	1	57	9.77	0.0028
time*TRACTAME	2	300	0.50	0.6084

En el gráfico puede verse el efecto de gran variabilidad en los datos simulados frente a los datos reales:



Con 300 pacientes generados (100 pacientes por grupo) los resultados son:

The Mixed Procedure

Type 1 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
TRACTAME	2	300	2.72	0.0675
time	1	57	9.77	0.0028
time*TRACTAME	2	300	0.50	0.6084

Otro análisis de robustez del modelo podría ser realizar simulaciones con los valores extremos de los intervalos de confianza del modelo.

