

## Capítulo 8

---

# Lenguajes de Simulación

## Capítulo 8

### Lenguajes de Simulación

Los objetivos de este capítulo son los siguientes:

- Presentar los diferentes conceptos básicos que tienen los lenguajes de simulación.
- Conocer los diferentes tipos de formas de hacer simulación.

#### 8.1 Introducción

Los lenguajes de simulación están especialmente preparados para determinadas aplicaciones de la simulación. Así suelen venir acompañados de una metodología de programación apoyada por un sistema de símbolos propios para la descripción del modelo, por ejemplo, mediante diagramas de flujo u otras herramientas que simplifican notablemente la modelación y facilitan la posterior depuración del modelo.

#### 8.2 Clasificación de los Lenguajes

Para los modelos de simulación se pueden usar diferentes tipos de lenguajes de programación, tales como: Arena, ProModel, Mimic, Cobloc, Midas, etc.

Muchos de estos lenguajes tienen varias versiones, pero se usan los nombres genéricos o familiares para no listar todas las versiones.



En el Apéndice A se muestra una clasificación de lenguajes basada en las tres técnicas de simulación que trataremos a continuación.

#### 8.3 Técnicas de Simulación Análoga

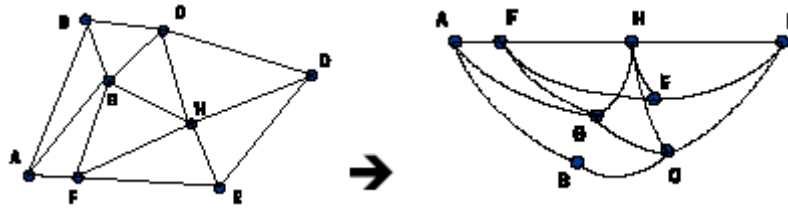
Los modelos análogos son modelos de flujo de tiempo continuo, aunque la escala de tiempo puede ser acelerada o desacelerada, usando factores de escala apropiada.

Los modelos análogos poseen algunas propiedades similares a los objetos representados, pero sin ser una réplica morfológica de los mismos: Un mapa es un modelo de la realidad establecido mediante un conjunto de convenciones relativamente complejo que conduce a un resultado final claramente distinto del objeto representado.

Ejemplo:

Modelo análogo utilizado para resolver el problema del camino más corto entre dos nodos de un grafo.

Los lugares se representan mediante pequeños anillos y los caminos entre ellos se representan mediante hilos de la misma longitud que el camino real. Para localizar la ruta más corta entre dos puntos, A y D, se toman los anillos correspondientes y se tensa la red. Los hilos tensos definen la ruta mínima: A-F-H-D.



La simulación análoga básica consiste en resolver una ecuación diferencial de alto orden mediante la integración repetida de una variable analógica que se iguala o relaciona con la variable real del problema.

La diferencia básica entre los modelos digitales y los modelos análogos reside en que los primeros están codificados en cifras, lo que permite su tratamiento por medios informáticos.

Aunque el uso de computadoras análogas ha disminuido debido a la aparición de las computadoras digitales, los conceptos y técnicas de la simulación análoga siguen aplicándose en forma amplia en muchos dispositivos de sistemas electrónicos y de control.

Algunos lenguajes de computadoras digitales, tales como APACHE y APSE, pueden usarse para ayudar a programar simulaciones análogas. Usando alguno de estos lenguajes se puede reducir el tiempo de programación en el computador análogo. Este lenguaje está orientado a ecuaciones y es una extensión de un lenguaje de propósitos generales.

## 8.4 Técnicas de Simulación Digital

Modelo digital: El objeto real se codifica en cifras y se maneja con medios informáticos.

Ejemplo

La reconstrucción de un edificio antiguo es un ejemplo de modelo digital. Parte del edificio ha sido representado a partir de un levantamiento simulado basado en restos de cimientos y muros. A Continuación, se presenta la reconstrucción simulada de la Biblioteca Celso en Éfeso, Turquía.



Biblioteca Celso en Éfeso, Turquía



Reconstrucción simulada de la Biblioteca Celso en Éfeso, Turquía

## Ventajas y Riesgos de los Modelos Digitales

### Ventajas:

- **No ambigüedad:** Cada parte del modelo debe expresarse explícitamente mediante valores o secuencias de operaciones (algoritmos) programadas en un lenguaje concreto.

- **Verificabilidad:** Los modelos pueden ser revisados por personas independientes ya que se basan en datos y algoritmos.
- **Repetibilidad:** Los resultados son los mismos para los mismos datos de entrada; no están sometidos a factores aleatorios o incontrolados y pueden ser replicados indefinidamente.

Los riesgos específicos de los modelos digitales están derivados de una mala concepción de su naturaleza y, en especial:

De la confusión entre exactitud y precisión:

- Exactitud es la coincidencia de un valor medido o estimado con un valor “real” o de referencia.
- Precisión es la potencialidad del método de medida para dar resultados con n dígitos significativos reales.

Aplicado a las computadoras: Las computadoras pueden generar resultados muy precisos, pero eso no significa que sean exactos en absoluto y, frecuentemente, se confía en que lo primero implica lo segundo.

Las técnicas de simulación digital incluyen el uso de lenguajes de propósitos generales y de propósitos especiales. Estas técnicas aplican los procedimientos de integración numérica para evaluar la respuesta en el tiempo de una ecuación diferencial dada.

A continuación, se presentan las ventajas y desventajas de estos lenguajes:

#### 8.4.1 Lenguajes de Propósitos Generales

Ventajas:

- Este lenguaje presenta un número mínimo de restricciones en un formato de salida. Por lo general se tiene ya conocimiento del lenguaje.
- Ofrece al programador una flexibilidad máxima en el diseño, implementación y uso del modelo.
- Tienen mayor capacidad expresiva, nos permite expresar ideas complejas con pocas líneas de código. Esto deriva, en una mayor productividad del programador.
- Se puede aplicar en la mayoría de las actividades que desarrolla el hombre, además de ser muy eficientes en empresas de trabajo para un mayor desarrollo.

Desventajas:

- El tiempo de programación de una simulación resulta demasiado largo.
- Se hace difícil la programación.
- Los lenguajes de propósito general deben tener un rendimiento aceptable para una gran variedad de operaciones, mientras que un lenguaje de consulta sobre XML se centra en optimizar este tipo de operaciones.
- El mayor inconveniente de este tipo de programación se presenta al interpretar los mensajes de error del sistema de desarrollo, puesto que se requiere un alto grado de especialización en el desarrollo de software.

#### 8.4.1.1 Plataformas que utiliza Java

- **Java Card**
- **Java Micro Edition**
- **Java Standard Edition**
- **Java Enterprise Edition**

#### 8.4.2 Lenguajes de Propósitos Especiales: Lenguajes de Simulación

Ventajas:

- Estos lenguajes requieren menos tiempo de programación y proveen técnicas de verificación de errores superiores a las provistas por los lenguajes de propósitos generales.
- Poseen la facilidad para construir subrutinas de usuarios.
- Generan automáticamente ciertos datos necesarios en las corridas de simulación y controlan la administración y localización del almacenamiento del computador durante la corrida de simulación.

Desventajas:

- Se reduce la flexibilidad en la creación del modelo e incrementa el tiempo de corrida del computador.
- El analista debe amoldarse a los requerimientos del formato de salida del lenguaje.
- En el estudio de la simulación, se debe definir el sistema a modelar y debe describirse en términos de diagramas de flujos lógicos y relaciones funcionales.
- El procedimiento usual de escoger el lenguaje apropiado para una aplicación es el de adoptar un lenguaje conocido por el analista.

Cualquier algoritmo general puede expresar el modelo deseado, sin embargo, uno de los lenguajes de simulación especializado puede tener distintas ventajas en términos de facilidad, eficiencia y efectividad.

### **.4.3 Técnicas Híbridas**

Algunas técnicas híbridas son más análogas en su naturaleza y algunas son más digitales. En ambos casos son capaces de contener eventos continuos y discretos.

La simulación híbrida implica el uso de computadoras digitales de propósitos generales y computadoras análogas de propósitos generales completamente separados y unidos sólo por un equipo de interface.

Este equipo de interface que une a las computadoras de propósitos generales consiste en convertidores de digital a análogo y de análogo a digital.

La mayoría de las simulaciones híbridas se llevan a cabo mediante un procedimiento orientado, en lenguaje digital de propósito general como un programa supervisor, a menudo en combinación con subrutinas diseñadas por un usuario (programador análogo), para revisar la porción análoga, conversión de datos, diagnóstico y control.

Un lenguaje desarrollado para la simulación híbrida fue HYTRAN. Este sirvió para el desarrollo de HOI (HYTRAN Operations Interpreter), el cual es utilizado para la automatización, organización y revisión de subsistemas análogos y para tener un control completo del programa híbrido.

#### **.4.3.1 Técnicas híbridas en Tailandia**

Desarrollaron una técnica de iniciador de secuencia de reacción de cadena de polimerasa específica (PCR-SSP) para detectar moléculas complejas en el grupo sanguíneo MNS en la población tailandesa utilizando dos juegos de cebadores de nuevo diseño específicos para cuatro híbridos GYP (BAB), GP.Mur, GP.Hop, GP.Bun y GP.HF, y dos híbridos GYP (ABA), GP.Vw y GP.Hut. Se analizaron mil cuarenta y una muestras de sangre con anti-Mi humano mediante una técnica de tubo convencional y con computadoras análogo-híbrida se sacaban los resultados o el valor semilla, se analizaron 598 muestras mediante la técnica en un emulador digital verificaban que células eran complejas y que otras no si estaban en el rango del valor semilla.

#### **.4.3.2 Fábricas en Tailandia**



- Todos los fabricantes de **automóviles** más importantes del mundo, los ensambladores y los fabricantes de partes y componentes están presentes en Tailandia y muchos están planeando nuevas inversiones. Ford, General Motores, BMW, DaimlerChrysler, Mitsubishi, Mazda, Toyota, Isuzu, Honda y Nissan tienen una presencia establecida en Tailandia, produciéndose cerca de 1,4 millones de vehículos.
- El Reino de Tailandia es en la actualidad el mayor productor mundial de camionetas de una tonelada y el séptimo exportador más grande de automóviles.
- El Grupo Charo en Pokphand, con sede en Bangkok, Tailandia fue fundado en 1921 por los hermanos Chia y hoy es uno de los conglomerados más importantes de Asia. Con empresas y filiales internacionales que operan en los sectores de la agroindustria, el comercio minorista y las telecomunicaciones; actualmente emplean a más de 250.000 personas en sus filiales (fábricas y oficinas) en todo el mundo.

#### 8.4.4 Facilidades de los Lenguajes de Simulación

Entre las facilidades que poseen los lenguajes de simulación podemos mencionar:

- Reducen la tarea de programación.
- Ayudan a definir las clases de entidades dentro del sistema.
- **Proveen un natural ambiente para modelamiento de la simulación.**
- Proporcionan la flexibilidad de cambios.
- Proveen un significado de diferenciación entre las entidades de la misma clase por características o propiedades.
- Describen la relación entre entidades y su ambiente común.
- **Algunos modelos en lenguajes de simulación permiten interfaces con lenguajes generales, específicamente FORTRAN (ocurre con SLAM II, SIMAN, GPSS).**
- **Proveen una gran interacción entre edición, depuración y ejecución. Alcanzando algunos de ellos implantación de la ingeniería de software.**

Los procesos de simulación requieren de ciertas funciones comunes que diferencian los lenguajes de simulación de los lenguajes algebraicos generales o de los lenguajes de programación para los negocios. Entre los requerimientos de los procesos de simulación que satisfacen los lenguajes especiales tenemos: La creación de números y variables aleatorios, el avance automático del tiempo por una unidad o para el próximo evento,



registrar datos para la salida; arreglar salidas en formatos específicos; detectar y reportar inconsistencias lógicas y otras condiciones de error; ejecutar análisis estadísticos en los datos registrados.

#### **8.4.5 Clasificación de los Lenguajes de Simulación**

Los lenguajes de simulación se clasifican en dos clases: Los lenguajes orientados a procesos de cambios continuos y los lenguajes orientados a cambios discretos.

**8.4.5.1 Lenguajes orientados a Procesos de Cambios Continuos**  
**generalmente son apropiados cuando el analista considera al sistema bajo estudio, como un flujo continuo de información. En simulación continua, generalmente el reloj de simulación se incrementa a intervalos fijos de tiempo.**

**8.4.5.2 Lenguajes orientados a cambios discretos**  
**Por otra parte, en modelos de simulación discreta, al analista le interesa lo que sucede a entidades individuales del sistema. Por consiguiente, el reloj de la simulación en simulación discreta se incrementa cada vez que ocurre un evento.**

**Pueden dividirse en tres enfoques: 1) Enfoque de eventos, 2) Enfoque de actividades, 3) Enfoque de procesos. Un evento se define como un cambio en el estado de una entidad del sistema. Una actividad es una colección de operaciones que transforman el estado de una entidad. Un proceso es una secuencia de eventos que ocurren en un tiempo determinado.**

#### **8.5 Lenguajes de Simulación de Ecuaciones**

Entre los lenguajes de simulación orientados a bloque podemos mencionar: MIDAS, PACTOLUS, SLAUS, MADLOC, COBLOC, 1130 CSMP.

Estos lenguajes simulan el comportamiento de las computadoras análogas e híbridas sobre la base de componente por componente, ejemplo: Un sumador es reemplazado por un código operacional de suma, un integrador es reemplazado por un código operacional de integración, etc. Estos complementos de los componentes funcionales tienen sus interconexiones especificadas por un lenguaje orientado a bloques, lo mismo que el tablero

de un computador análogo eléctricamente unido a los componentes de computación análoga.

### 8.5.1 Lenguajes para la Solución de Ecuaciones Diferenciales

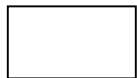
Uno de los lenguajes que se utiliza para resolver ecuaciones diferenciales es el DYNAMO (Dynamic Models), el cual utiliza ecuaciones de diferenciación de primer orden para aproximar procesos continuos.



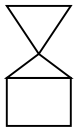
Las variables esenciales de un estudio de un sistema dinámico que son variables de estado y variables de salida son descritas en DYNAMO como ecuaciones de nivel y ecuaciones de rata, respectivamente. Las variables de estado (niveles) describen el estado y condición del sistema en un tiempo dado, mientras que las variables de salida (ratas) describen los cambios de estado al transcurrir el tiempo.

Las ecuaciones auxiliares son los componentes de las ecuaciones de rata y describen la función de estas ecuaciones.

Otro lenguaje de simulación utilizado para resolver ecuaciones diferenciales y que utiliza variables de estado y variables de salida es el VENSIM® (Ambiente de simulación Ventana).



Variable de nivel: Valores cruciales para el estudio del comportamiento del sistema. Se refieren a un punto específico en el tiempo.



Variable de flujo o rata: Afectan a las variables de nivel. Se refieren al tiempo transcurrido entre dos instantes, hacen que los niveles aumenten o disminuyan.



Canal de material: Representan las acciones de los flujos sobre los niveles.

### 8.5.2 Lenguajes Basados en Ecuaciones

Estos lenguajes se escapan de las restricciones impuestas por la construcción de un bloque completo de los lenguajes de un simulador análogo y se dan directamente de ecuaciones.

Los lenguajes de un simulador análogo agregan el poder y la conveniencia de instrucciones algebraicas y lógicas.

Los lenguajes orientados a ecuaciones se desarrollaron más recientemente que los lenguajes de un simulador análogo.

Los lenguajes desarrollados después de 1966 tienen la orientación de una ecuación directa, puesto que en estos últimos años se trató de estandarizar el formato y la estructura de los programas de un simulador análogo-digital.

Entre los lenguajes basados en ecuaciones podemos mencionar: DSL/90, MIMIC, BHSL, DIHSYS y el S/360 CSMP.

## 8.6 Lenguajes orientados a Cambios Discretos

Los lenguajes orientados a cambios discretos están divididos en cuatro categorías que son:

- Lenguajes orientados a un evento
- Lenguajes orientados a un proceso

(Se eliminaron dos porque con evento y proceso son suficientes).

Antes de describir cada una de estas categorías debemos mencionar que los lenguajes de flujo de transacciones son actualmente lenguajes de proceso, ya que ellos emplean un programa sinóptico del sistema, pero se establece que son considerados como otra categoría por su orientación de flujograma.

Los lenguajes de evento, actividad y proceso, con excepción del SIMCOM, usan instrucciones de programación para describir relaciones de causa y efecto entre los elementos del sistema.

## 8.6 Algunos programas de simulación

### 8.6.1 Stella

Es un programa que permite realizar la modelación o simulación de sistemas de una manera gráfica, a través de símbolos. sus componentes principales son barra de menú, el stock. el Flow. convector, se le puede agregar gráficas y tablas, su área de trabajo cuenta con una interface, mapa, modelo y también pueden incluirse ecuaciones

### 8.6.2 VENSIM

Es una herramienta grafica de creación de modelos de simulación que permite conceptualizar, documentar, simular, analizar y optimizar modelos de Dinámica de Sistemas. En otras palabras, este software proporciona una forma simple y flexible de crear modelos de simulación, ya sean con diagramas causales, diagramas de flujos, organigramas, cartas de organización y diagramas de la espina de pez.

### 8.6.3 PROMODEL

Este software permite simular cualquier tipo de sistemas ya sea manufactura, logística, manejo de materiales entre otros, también se pueden simular bandas de transporte, grúas viajeras, ensamblaje, corte talleres logística, entre otros, Lo único que se necesita para modelar es esfuerzo e ingenio.

### 8.6.4 Arena

El software de simulación Arena es un “Seguro de vida” para las empresas ya que les ayuda a predecir el impacto en una organización de nuevas ideas, estrategias y políticas de negocio antes de implementarlas, sin poner en peligro el nivel de servicio actual, sus procesos y sus relaciones con clientes y proveedores; además de esto Arena es una ayuda para ahorrar gastos innecesarios, optimizar sus inversiones y fortalecer las relaciones con los clientes.

## 8.7 Lenguajes orientados a una Actividad

Estos lenguajes representan hechos que dependen del tiempo como ocurrencia instantánea en un tiempo simulado.

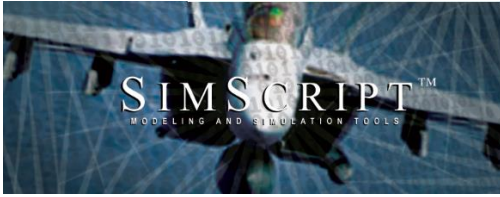
Usando estos lenguajes no se fija el tiempo de la ocurrencia dentro de un programa, pero se especifica sobre cuales condiciones pueden suceder estas ocurrencias.

En estos lenguajes aparecen instrucciones de no fijar el tiempo de una actividad, pero ellos contienen programas que revisan grupos de condiciones antes de cada tiempo de avance de simulación, para determinar si cualquiera de las actividades puede realizarse.

En este tipo de lenguaje, el programa está compuesto de una sección de prueba y una sección de acción. Siempre que el tiempo de simulación avance, todos los programas de actividad son revisadas para posibles ejecuciones. Todas las condiciones de prueba deben ser encontradas para los cambios de estados y en la sección de acción las instrucciones puestas por el tiempo deben llevarse a cabo. En esta categoría se encuentran los lenguajes CSL, ESP, FORSIM-IV, GSP y MILITRAN.

## 8.8 Lenguajes orientados a un Evento

En estos lenguajes cada evento debe representarse como una ocurrencia instantánea en el tiempo simulado. Podemos mencionar aquí lenguajes como el SIMSCRIPT, GASP, SIMCOM y SIMPAC.



SIMPAC, a Dassault Systèmes company



### 8.8.1 Lenguajes orientados a un Proceso

Este tipo de lenguaje intenta combinar la eficiencia de los lenguajes orientados a un evento con lo conocido de los lenguajes orientados a una actividad.

Es necesario mencionar que se entiende por proceso un grupo de eventos asociados con la descripción del comportamiento de un sistema.

Un proceso existe en el tiempo y puede tener un comportamiento dinámico. Los procesos son flexibles ya que su ocurrencia puede programarse.

Se pueden interrumpir y tienen subprocesos que obedecen a ellos, por lo tanto, se pueden programar para retrasarse ellos mismos y que además encuentren ciertas condiciones. Una de sus características fundamentales es que hacen que un programa individual actúe como si se tratara de varios programas, controlado independientemente por las observaciones orientadas a una actividad.

Esta programación hace posible el concepto de un punto de reactivación que significa lo siguiente: Un apuntador llama a una rutina de proceso donde se inicia una ejecución, después que algún comando que tenga un retraso de tiempo se ha llevado a cabo. Aquí se encuentran lenguajes como SIMULA, OPS y SOL.



### **8.8.2 Lenguajes orientados a Flujo de Transacciones**

En estos lenguajes, los avances del tiempo son simulados como pasos de transacciones a través de bloques y las decisiones son hechas de acuerdo con la lógica del sistema simulado.

El sistema está representado en términos de bloques, el programa crea transacciones, moviéndolas para bloques específicos y ejecutando las acciones asociados con cada bloque.

Entre los lenguajes orientados a flujo de transacciones podemos mencionar: GPSS y BOSS.

### **Diagrama de Flujo de Decisión**

El diagrama de flujo de decisión sirve de ayuda al analista sobre cuál es el lenguaje por seleccionar (técnicas de simulación) para la debida solución de su problema, sistema o modelo.

## 8.9 Ejercicios Propuestos

### 8.9.1 Población de focas

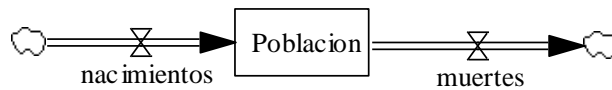
Una población de 500 focas vive en un sector del norte de Canadá. Cada año nacen 100 focas y cada año mueren unas 120. ¿Cuál será la población en unos 10 años?

**Solución:**

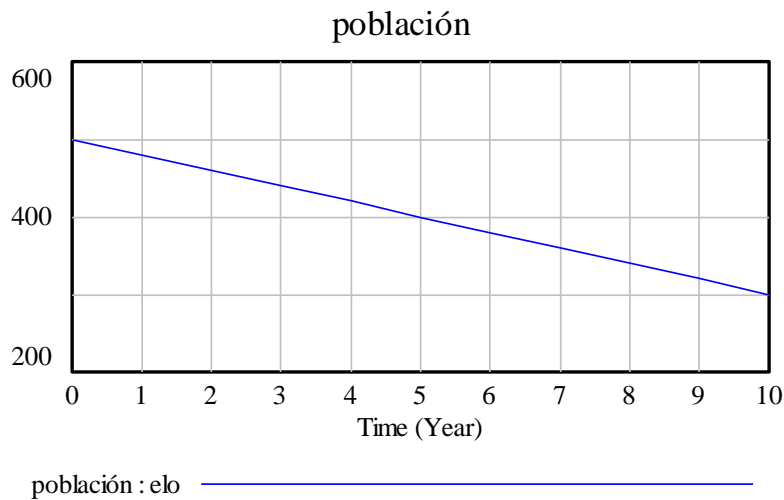
Nacimientos = 100

Muertes = 120

Población inicial = 500



Time (Years)	Población
0	500
1	480
2	460
3	440
4	420
5	400
6	380
7	360
8	340
9	320
10	300



### 8.9.2 Población de Abetos



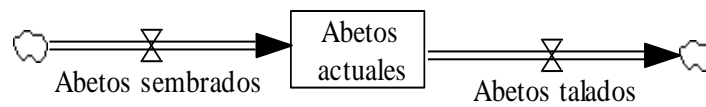
Hay aproximadamente 5 millones de abetos en el bosque de Hardwood. Una compañía maderera ha estado talando los árboles a razón de unos 100,000 cada año. Un grupo de ambientalistas ha denunciado que a ese paso el bosque será destruido, por lo que ellos han decidido trabajar duramente para sembrar tantos árboles como les sea posible. El grupo tiene una capacidad de plantar unos 5000 árboles cada año. ¿Cuántos árboles habrá en el bosque en 30 años?

**Solución:**

Abetos sembrados = 5000

Abetos talados = 100,000

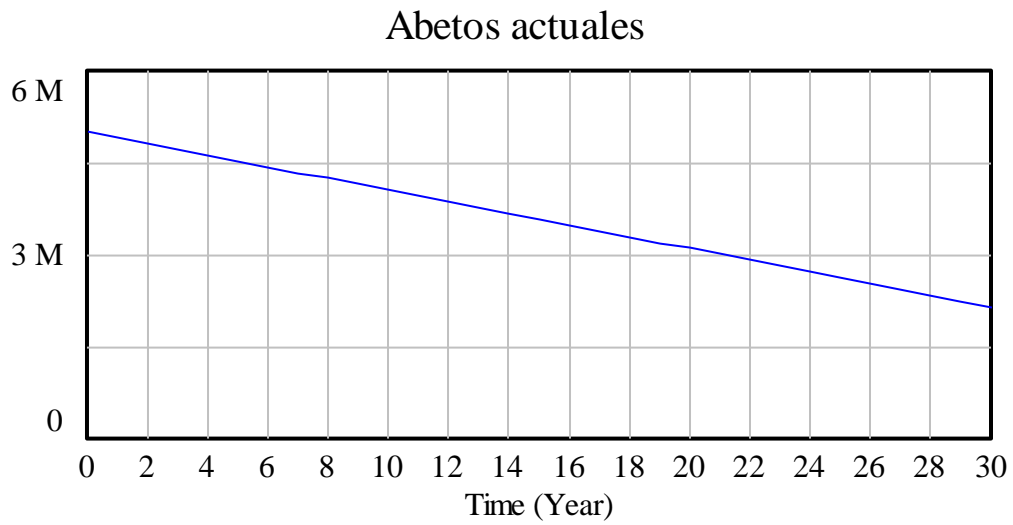
Abetos actuales = 5 millones



Time (Year)	Abetos actuales
0	5 E +6
1	4.905 E +6
2	4.81 E +6
3	4.715 E +6
4	4.62 E +6
5	4.525 E +6
6	4.43 E +6
7	4.335 E +6
8	4.24 E +6
9	4.145 E +6
10	4.05 E +6
11	3.955 E +6
12	3.86 E +6
13	3.765 E +6
14	3.67 E +6
15	3.575 E +6
16	3.48 E +6
17	3.385 E +6
18	3.29 E +6
19	3.195 E +6
20	3.1 E +6
21	3.005 E +6
22	2.91 E +6
23	2.815 E +6
24	2.72 E +6
25	2.625 E +6
26	2.53 E +6
27	2.435 E +6

28	2.34 E +6
29	2.245 E +6
30	2.15 E +6

e+006 = millones



Abetos actuales : Current

### 8.9.3 Bizcochos en el Estómago

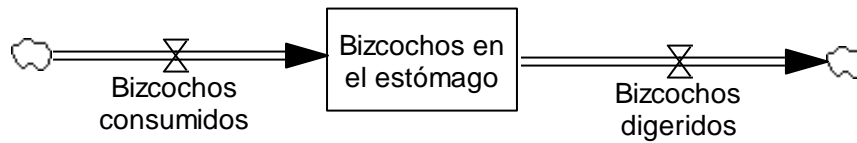
Es sábado y Karla está trabajando duramente en su proyecto de grupo y su mejor amiga y compañera de equipo ha decidido contribuir con su esfuerzo por lo que le hornea una gran plancha de bizcochos. Karla mordisquea los bizcochos a medida que va trabajando y se come un bizcocho cada hora. A medida que ella trabaja su estómago digiriendo los bizcochos a razón de 1 cada 2 horas. Si ella termina el proyecto en 8 horas. ¿Cuántos bizcochos tendrá en el estómago?

#### Solución:

Bizcochos Consumidos por hora = 1

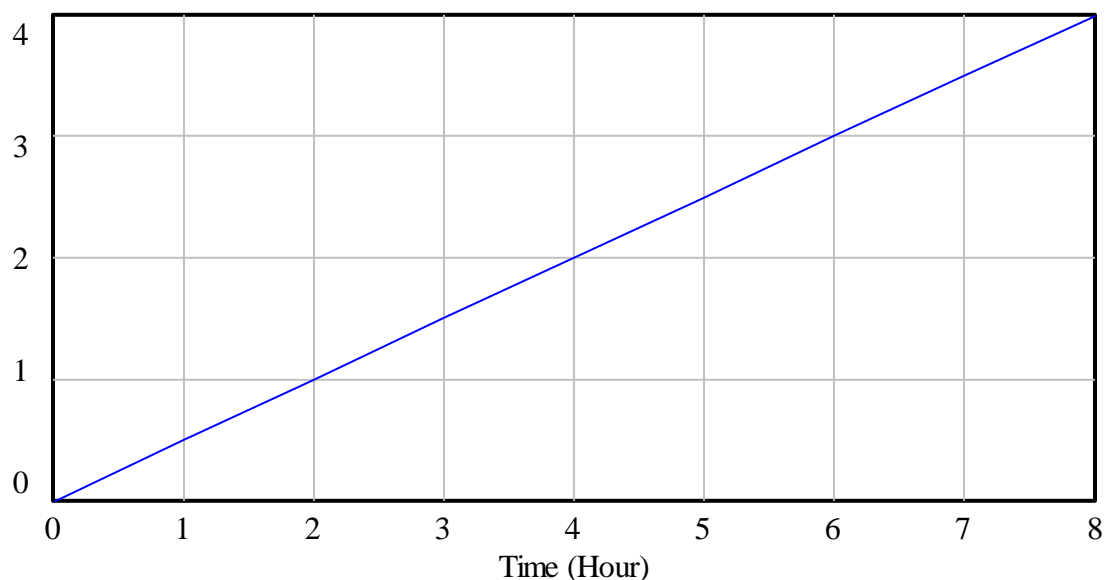
Bizcochos digeridos por hora = 0.5

Bizcochos en el estómago inicialmente = 0



Tiempo (Hora)	Bizcochos en el estómago
0	0
1	0.5
2	1
3	1.5
4	2
5	2.5
6	3
7	3.5
8	4

### Bizcochos en el estómago



Bizcochos en el estómago : Corrida1 — Bizcochos

Con estos ejercicios se observa la aplicación de un lenguaje de simulación como VENSIM, ya que proveen resultados rápidos y precisos que nos ayudan a comprender el sistema que se simula.

### Apéndice A

Siglas O Nombres	Significado de las Siglas o Nombres	Comentarios
ASTRAL	Traductor de esquemáticos analógico a lenguaje algebraico	Instrucciones de entrada compiladas que modelaban la computadora analógica.
DAS	Simulador analógico digital	Algoritmo de integración rectangular utilizado en la IBM 7090, que se modificó quedando en un algoritmo predictor-corrector de quinto orden.
MIDAS	DAS modificado	Escrito para las IBM 7090/7094; usaba un esquema de ordenamiento similar al de ASTRAL y un algoritmo de integración mejorado.

PACTOLUS	El rey Midas se bañó en el río Pactolo, para quitarse la maldición de que todo lo que tocaba se convertía en oro	Escrito para la IBM 1620, pequeña computadora que los usuarios podían operar en forma individual. El PACTOLUS fue precursor de DSL-90 y CSMP.
MIMIC	Relacionado ("mímico") con MIDAS	Mejora sobre MIDAS, que cambió el algoritmo predictor-corrector de integración de quinto orden por el algoritmo de Runge-Kutta de cuarto orden de tamaño de incremento variable.
DSL-90	Lenguaje de simulación digital para la IBM 7090/7094	DSL-90 traduce diagramas de bloques de simulaciones de computadora analógica o ecuaciones diferenciales en una subrutina de FORTRAN IV que se compila y ejecuta con uno de cuatro algoritmos de integración.
CSMP	Programa de modelación de sistemas continuos	Creado para la serie IBM 1130 e IBM 360, CSMP incorpora características de DSL-90 y PACTOLUS, y agrega la opción de interacción por consola. Ofrece ocho algoritmos de integración.
CSSL	Programa de modelación de sistemas continuos	CSSL es un preprocesador que convierte la descripción del sistema por medio de ecuaciones diferenciales en un lenguaje procesal.
GPSS		GPSS modela transacciones discretas que se desplazan por el sistema, como las personas que hacen fila (cola) para ser atendidos en un banco. Resuelve problemas de atención a filas (colas) mediante el uso de almacenamiento, retardo, interruptores, álgebra booleana y álgebra ordinaria.

STRESS	Dispositivo de resolución de sistemas de ingeniería estructural	Se utiliza para modelar estructuras armadas elásticas y cargadas en forma estática. Desarrollado en el MIT.
STRUDL I y STRUDL II	Lenguaje de diseño estructural	Perfeccionamiento de STRESS, STRUDL II maneja cargas dinámicas de sistemas discretos o continuos. STRUDL fue creado por Robert D. Logcher en los MIT Civil Engineering Systems Labs en 1968.
COGO	Geometría de coordenadas	Disponible para la mayor parte de las computadoras. Lo utilizan investigadores para producir mapas y evaluar áreas de propiedad.
SAAM/27	Simulación, análisis y modelación	Creado por National Institutes of Health y utilizado para la simulación biomédica, SAAM se está promoviendo para todas las aplicaciones de simulación.
DYNAMO	Modelos dinámicos	Creado por Jay Forrester y Alexander Pugh en el MIT para modelar la producción económica y la dinámica mundial. Se usa un conjunto de símbolos únicos, pero que básicamente describen coeficientes e índices de acoplamiento que se integran.
SIMSCRIPTII. 5		Extensión de GPSS. Maneja simulaciones de filas de espera (colas) y planificación. En SIMSCRIPT, el sistema es descrito por "entidades" que tienen "atributos" que interactúan con actividades descritas por "rutinas de eventos".

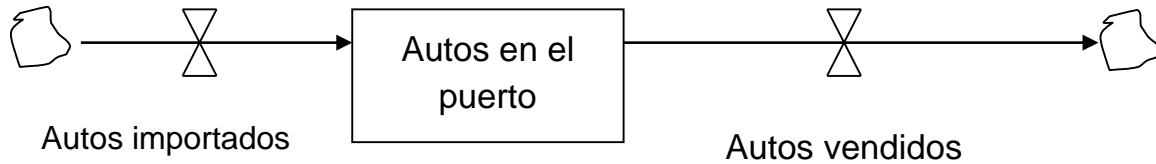
Referencia: ENCICLOPEDIA DE LA ELECTRONICA

MIDAS	→	PACTOLUS	Influence
PACTOLUS	→	CSMP	Influence
PACTOLUS	→	DIANA	Enhancement of
PACTOLUS	→	DSL/90	Based on
PACTOLUS	→	HYPAC	Extension of
PACTOLUS	→	LEANS	Implementation
PACTOLUS	→	Schrader simulation Algol	Incorporated some features of

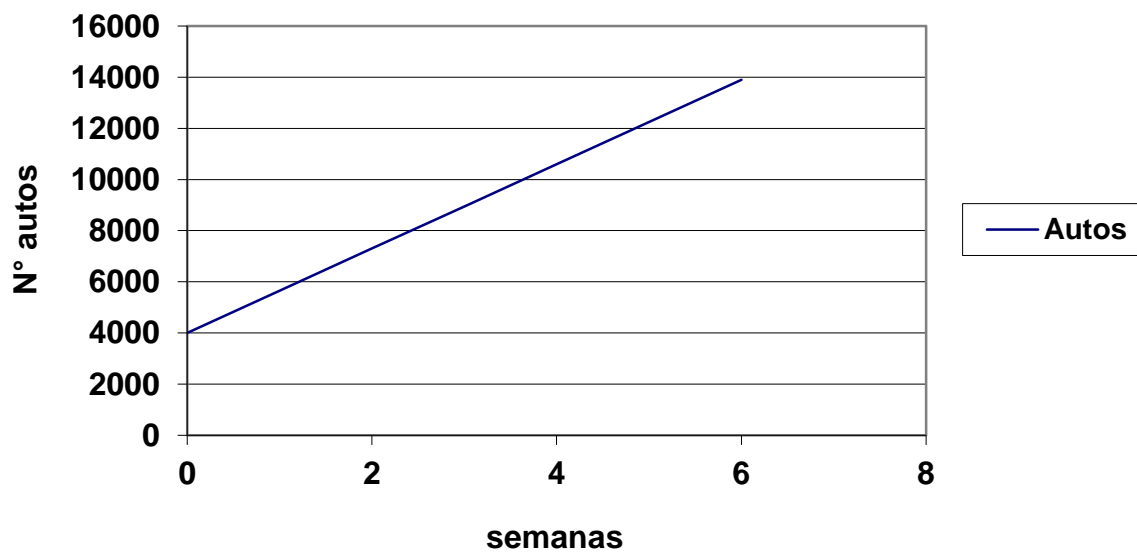


**Ejercicios (por revisar)**

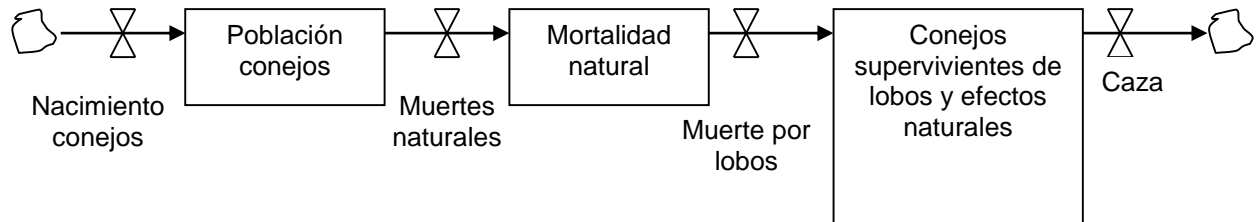
1. Una compañía naviera exportadora e importadora de autos trae semanalmente al puerto 4000 autos, de estos, 2350 son vendidos a las concesionarias por semana. Si ya en el puerto se encontraba una población de 4000 autos ¿Cuántos autos quedarán en el puerto después de 6 semanas?



Semanas	Total de autos en el puerto
0	4000
1	5650
2	7300
3	8950
4	10600
5	12250
6	13900



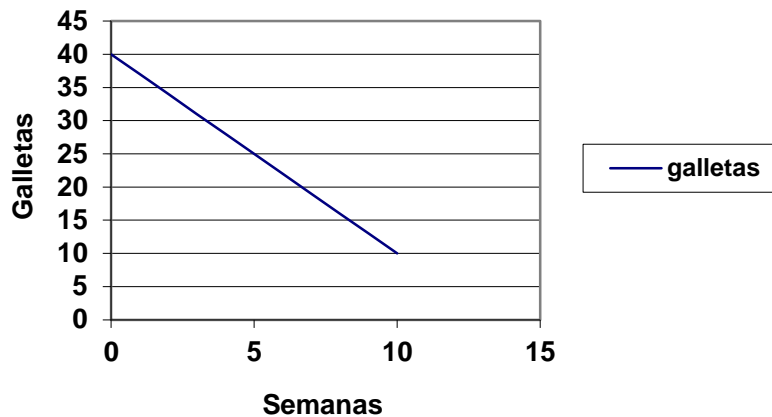
2. En las praderas del norte se encuentra una población de 25000 conejos que se mantiene constante, pero realizando un estudio se ha descubierto que esta población ha sufrido un cambio repentino debido a que tiene solamente una tasa de nacimiento de 25% de la población total y que cada año muere un total de 15 % por muertes naturales, otro 8% por los lobos y un 6 % por la caza. Con esto datos se busca averiguar si la población de conejos estará en peligro de extinción en las praderas del norte en 25 años.



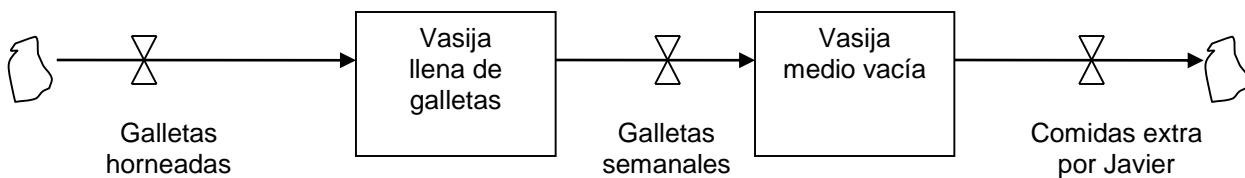
años	Población
0	25000
1	25250
2	25503
3	25758
4	26015
5	26275
6	26538
7	26803
8	27071
9	27342
10	27616
11	27892
12	28171
13	28452
14	28737
15	29024
16	29314
17	29608
18	29904
19	30203
20	30505
21	30810
22	31118
23	31429
24	31743
25	32061

3. A la madre de Javier le encantan las galletas y tiene una vasija llena con aproximadamente 40 galletas. Cada semana se comen en la casa una cantidad de 20 galletas y estas son repuestas por su madre cada sábado. Pero Javier fue sorprendido después de un tiempo de que se estaba comiendo una cantidad de galletas extra cada semana. Si Javier se comía 3 galletas extra cada semana ¿En cuántas semanas su mamá se dio cuenta que alguien estaba comiendo más galletas?

Semanas	Galletas
0	40
1	37
2	34
3	31
4	28
5	25
6	22
7	19
8	16
9	13
10	10



Viendo el análisis se puede decir que la madre se pudo dar cuenta entre las 3 y 5 semanas.



**Bibliografía**

Sistemas de Información Geográfica Modelos y Simulación,  
[http://www.unex.es/geodesia/sig/Tema\\_01.pdf](http://www.unex.es/geodesia/sig/Tema_01.pdf), 2002.

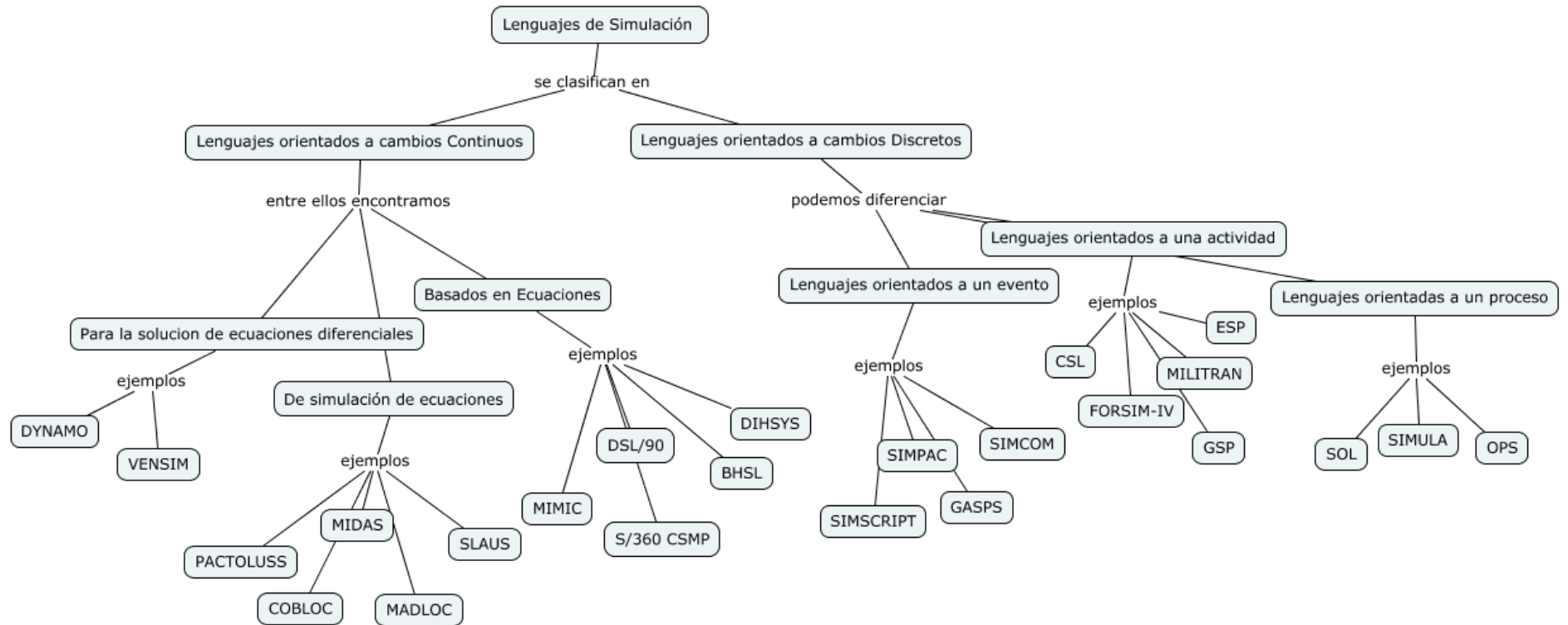
¿Qué es un Modelo?, <http://www.dioptra.es/Explorer/teleformacion/c3/m2/2a.htm>, 2002.

Lenguajes para simulación,  
<http://www.infoab.uclm.es/asignaturas/42539/transparencias/html/simulacion10.htm>, 2002.

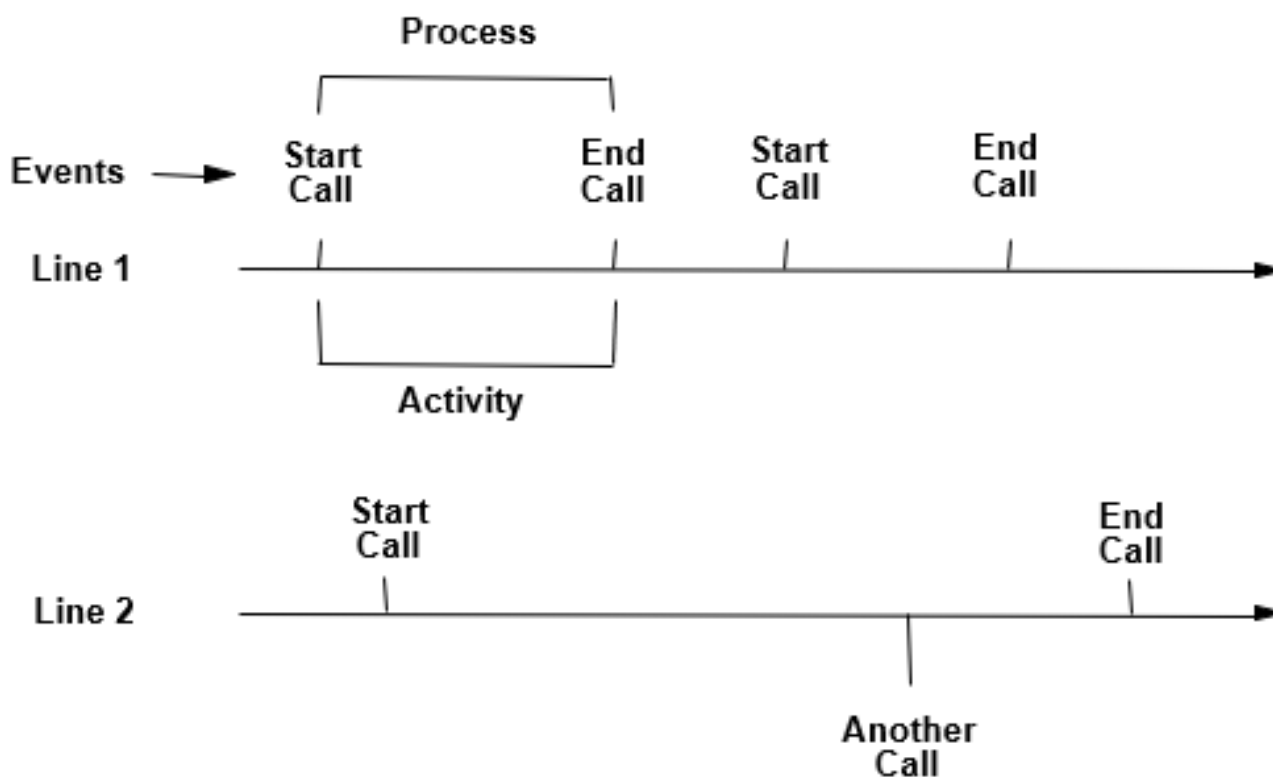
Instrucciones para VENSIM, <http://www.geocities.com/martin3162/vensim.htm>, 2002.

Criterios de selección, <http://www.industrial.uson.mx/materias/m5153/simnetii/sld002.htm>, 2002.

## Apéndice A: Clasificación de los Lenguajes de Simulación



### Apéndice B: ¿Qué es Evento y Proceso?



Indistintamente del lenguaje de simulación que se esté utilizando, todo proceso inicia a partir de un evento. Un **evento** es algo instantáneo y podría definirse como un suceso importante el cual genera cambios en variables o entidades.

Luego de presentarse un evento:

- Variables globales cambian.
- Una entidad se crea, cambia, se archiva o se destruye.
- Otro proceso o evento recibe una alerta para su futura ejecución.

Cuando un evento se presenta se llama a un proceso y, mientras este proceso se ejecuta, el tiempo pasa, pero nada que no esté relacionado con el evento o proceso ocurre. Este tiempo sin acciones es llamado **actividad**.

Por su parte, **proceso** es una secuencia de eventos y actividades que afectan una entidad o variable durante el mientras viaja en nuestro sistema simulado.

En la **línea 1**, de la imagen anterior, un *evento* se hace presente, lo cual da inicio a un *proceso*. El *proceso* se inicia mediante un llamado, este llamado puede desencadenar

una serie de acciones o cambios en las variables o bien, puede convertirse en el llamado a otro proceso, como se muestra en la **línea 2**.