

Simulación a eventos discretos

Clase nro 3.

Curso 2010.

Repaso clase 2

- Tiempo
- Entidades y recursos
- Eventos
- Estados y colas
- Actividades
- Diagramas
- Ejecutivo y calendario

S.E.D. - Curso 2010

Clase 3

- Temas de hoy:
 - Métodos de estructuración
 - Pseudocódigos
 - Ejemplos

S.E.D. - Curso 2010

Métodos de estructuración

El ejecutivo se estructura según tres métodos de estructuración de la simulación o "enfoques del mundo"

1. Tres fases. (orientado a eventos)
2. Dos fases. (orientado a eventos)
3. A procesos. (orientado a procesos)

Tres fases: eventos condicionados y fijos se programan como procedimientos separados.

Dos fases: los procedimientos que describen los eventos fijos incluyen todos los eventos condicionados que van a suceder como resultado de los propios eventos agendados o fijos.

S.E.D. - Curso 2010

Ejecutivos

Ejecutivo del método de las tres fases:

1. Avanzar el reloj al tiempo del próximo evento fijo.
2. Ejecutar todos los eventos fijos que ocurren a esa hora.
3. Verificar todos los eventos condicionados y ejecutar aquellos cuyas condiciones se satisfagan.

Ejecutivo del método de las dos fases:

1. Avanzar el reloj al tiempo del próximo evento.
 2. Ejecutar el próximo evento agendado (fijo) a suceder.
- Observar que se omite la 3era fase solamente.

S.E.D. - Curso 2010

Simulación en Pascal/C++

Determinada la lógica de la simulación, debemos representarla mediante un programa.

Existen varios paquetes específicos de lenguajes de programación de simulación (SPL).

Ventajas de un SPL:

- impone una estructura (basada en algunos de los métodos vistos).
- provee de facilidades en sorteos de v.a., manejo de listas, descripción de entidades, tratamiento estadístico y escritura de informes sobre resultados.

Alternativa: Lenguaje genérico (escribir rutinas de SPL).

S.E.D. - Curso 2010

Pseudocódigo

Próximo paso en el diseño del programa de la simulación es escribir un pseudocódigo.

Usaremos algo parecido al utilizado en análisis estructurado y algo de Pascal.

S.E.D. - Curso 2010

Pseudocódigo

El que usamos es similar al inglés estructurado descrito por De Marco (1978) y usado en análisis de sistemas, es un subconjunto del lenguaje Inglés escrito en formato rígido sin calificativos. El lenguaje es menos preciso y más transparente que un programa en Pascal pero es fácil de traducir a cualquier lenguaje de programación estructurado.

Como el libro usa Pascal, el pseudocódigo que presenta está escrito e indentado en forma similar al Pascal, usando palabras específicas del Pascal como ser *begin*, *end*, *while*. Las oraciones que no son propias de Pascal se delimitan con llaves y paréntesis angulados.

S.E.D. - Curso 2010

Pseudocódigo

Las figuras 2.4 y 2.5 muestran un pseudocódigo del problema del hospital simple usando el método de las tres fases y el método de eventos. Los apéndices 2.A y 2.B muestran pseudocódigo para los otros casos de estudio.

S.E.D. - Curso 2010

Figura 2.4 Hospital simple - 3 fases

procedure arribo-paciente; {B1}

begin

<poner al recién llegado en Q1 para admisión>;

<crear al nuevo paciente>;

<setear el tiempo de arribo de este nuevo paciente>;

end;

S.E.D. - Curso 2010

Figura 2.4

Hospital simple - 3 fases

```
procedure inicio-internación; {C1}  
begin  
  while <Q1 no vacía y cama disponible>  
  begin  
    <admitir el próximo paciente en Q1>;  
    <ocupar una cama>;  
    <setear el tiempo del alta de este paciente>;  
  end  
end;
```

S.E.D. - Curso 2010

Figura 2.4

Hospital simple - 3 fases

```
procedure fin-internación; {B2}  
begin  
  <eliminar paciente>;  
  <devolver la cama>;  
end;
```

S.E.D. - Curso 2010

Figura 2.5

Hospital simple - 2 fases

```
procedure arribo-paciente; {B1}
begin
  if <hay una cama disponible> then
    begin
      <ocupar una cama>;
      <setear el tiempo del alta del paciente>;
    end
  else
    begin
      <poner al paciente en cola Q1 esperando admisión>;
    end;
  <crear el siguiente paciente>;
  <setear el tiempo de arribo de este paciente>;
end;
```

S.E.D. - Curso 2010

Figura 2.5

Hospital simple - 2 fases

```
procedure fin-internación; {B2}
begin
  <destruir el paciente>;
  if <Q1 no vacía> then
    begin
      <admitir el próximo paciente en Q1>
      <setear el tiempo del alta de ese paciente>
    end
  else
    begin
      <devolver la cama>;
    end
  end;
```

S.E.D. - Curso 2010

Tabla 2.1
Pacientes necesitando internación (4 camas disponibles)

<u>Paciente nro</u>	<u>Día de arribo</u>	<u>Duración estadía</u>
1	0	4
2	1	3
3	1	4
4	1	3
5	2	1
6	4	4
7	4	2
8	4	1
9	4	2
10	5	1
11	5	1
12	6	1
13	8	3

S.E.D. - Curso 2010

Tabla 2.2
Pacientes de tabla 2.1 pasan por el sistema

<u>Día</u>	<u>Arribo</u>	<u>Cola Q1</u>	<u>Internados</u>	<u>De-alta</u>
0	1	-	1	-
1	2,3,4	-	1,2,3,4	-
2	5	5	1,2,3,4	-
3	-	5	1,2,3,4	-
4	6,7,8,9	8,9	5,6,3,7	1,2,4
5	10,11	10,11	8,6,9,7	3,5
6	12	12	10,6,9,11	7,8
7	-	-	12,6	9,10,11
8	13	-	13	6,12

S.E.D. - Curso 2010

Ejecutivo TRES FASES

- A. Avanzar el reloj al tiempo del próximo evento fijo.
- B. Ejecutar todos los eventos fijos que ocurren a esa hora.
- C. Verificar todos los eventos condicionados y ejecutar aquellos cuyas condiciones se satisfagan.

S.E.D. - Curso 2010

Ejercicio 2.13.1 (tres fases)

Ver tabla 2.3 página 30.

S.E.D. - Curso 2010

Ejecutivo DOS FASES

- A. Avanzar el reloj al tiempo del próximo evento.
- B. Ejecutar el próximo evento agendado (fijo) a suceder.

Observar que se omite la 3era fase.

S.E.D. - Curso 2010

Ejercicio 2.13.1 (dos fases)

Ver tabla 2.4 página 32.

S.E.D. - Curso 2010

Ejemplos

Ver pseudocódigos 2.A y 2.B en el libro.

S.E.D. - Curso 2010

Resumen

Un modelo de simulación avanza desde un punto discreto del eje de tiempo (golpe de reloj) al siguiente.

S.E.D. maneja actividades en las que están implicadas entidades individuales. Estas entidades se encuentran en distintos estados a medida que la simulación se desarrolla (ocupadas, ociosas o en fila).

Generalmente las entidades necesitan de recursos para poder realizar alguna actividad.

La disponibilidad de un recurso puede ser una condición para que suceda un evento, el que marca el comienzo de una actividad (evento condicionado).

S.E.D. - Curso 2010

Cont. Resumen

Un evento fijo marca el fin de una actividad y es agendado a ejecutarse en una determinada hora. Un alimentador (*Feeder*), es un evento fijo que genera entidades temporarias.

Un diagrama de actividades muestra el flujo o el ciclo de vida de las entidades. Mediante un pseudocódigo se describen las actividades en lenguaje estructurado.

Existen tres formas (world views) de estructurar y controlar el flujo del tiempo en S.E.D.: 1) a eventos : a) *tres fases*, b) *dos fases*, 2) a procesos; a) y b) se diferencian por el manejo de los eventos condicionados. El método de las tres fases considera a los eventos condicionados como procedimientos separados. En el método de las dos fases, se integran en los eventos fijos.

S.E.D. - Curso 2010

Próxima clase:

**Bibliotecas S.E.D:
PascalSim y EOSimulator.**

S.E.D. - Curso 2010