

Casos de estudio

Clase nro. 9
CURSO 2010

Modelo de simulación

Producir un modelo de simulación no es solamente escribir código.

La estructura de la simulación y sus distribuciones se derivarán de:

OBJETIVOS
HIPÓTESIS DE TRABAJO
RESPUESTAS
VARIABLES DE DECISIÓN

S.E.D. 2010

Modelo de simulación

El modelo se compone de:

- Especificación (objetivos, hipótesis, variables de decisión, respuestas, duraciones de actividades, diagramas de actividades)
- Pseudocódigos
- Programa (código)

S.E.D. 2010

Modelo de simulación

OBJETIVOS

Deben ser claros, subjetivos o muy detallados, pero determinarán:

- i. Las variables de decisión.
- ii. Cuándo es necesaria una salida visual, un detallado tratamiento estadístico o ambos, y
- iii. Qué salidas son importantes.

S.E.D. 2010

Modelo de simulación

HIPÓTESIS DE TRABAJO

Existen hipótesis implícitas al modelo y otras explícitas. Ambas deben ser documentadas.

Los programas deben ser diseñados de forma de permitir cambios en etapas posteriores del proyecto (reducen la complejidad del modelo).

S.E.D. 2010

Modelo de simulación

RESPUESTAS

Tipos de parámetros y medidas de interés, así como estadísticas y datos a recolectar para el análisis.

VARIABLES DE DECISIÓN

Los objetivos indicarán cuáles factores serán fijos y cuáles cambiables.

S.E.D. 2010

Casos de estudio

- Hospital
- Taller de reparaciones

S.E.D. 2010

Hospital

El *sistema* es una simplificación del problema real.
La *especificación* del problema está dada por el detalle de:

- los objetivos,
- las hipótesis de trabajo,
- las variables de decisión,
- las respuestas y las duraciones de las distintas actividades (Tabla 6.1),
- y el diagrama de actividades (Figura 2.2).

S.E.D. 2010

Hospital

Aclaración de hipótesis:

- a) El arribo de los pacientes se describe mediante un proceso Poisson de tasa constante (aproximación burda pero inicial).
- b) El sistema opera continuamente, cuando en realidad pacientes agendados para operación no arriban por la noche.

Consideraremos el sistema en estado estacionario, por lo tanto investigaremos el efecto de cambiar valores de las variables de decisión en parámetros estables.

Medidas importantes: utilización de camas y tiempos de espera.

S.E.D. 2010

Hospital

Programa

- Se construye según el método elegido.

En PascalSim:

- Sala de operaciones:

```
var theatre : record
  body : entity;
  open, available: boolean;
end;
```

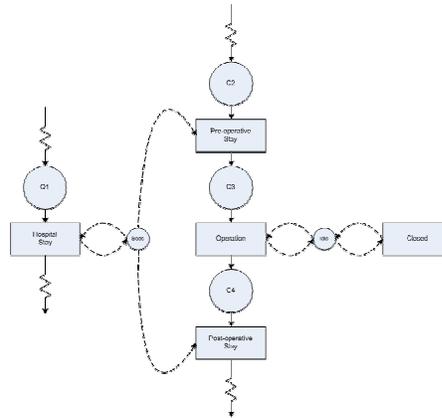
- Las variables de decisión se declaran como constantes globales.

S.E.D. 2010

Hospital

Programa

- La unidad de tiempo de la simulación es la hora.
- q4 es una cola ficticia, ventaja: cada actividad está compuesta por el par de eventos C y B, lo que facilita la modificación posterior del programa.
- Los histogramas se declaran y nuevos valores son ingresados cada vez que haya un cambio en algún tipo de evento C o B.



S.E.D. 2010

Hospital

Estado estacionario

- Simulación del Hospital es de tipo "Estacionaria", debemos determinar cuando comenzar a tomar datos para procesar.
- Utilizamos el método de promedios acumulados (Tabla 6.2). Se agrega código en la fase B del ejecutivo para producir promedios de las respuestas cada 49 hs simuladas. La Fig 6.1 grafica los datos obtenidos.

S.E.D. 2010

Hospital

Estado estacionario

- Observar que la cola de solo internados y el tiempo de espera para operación alcanzan el estado estable rápidamente.
- La estabilidad del sistema se alcanza alrededor de las 720 hs.
- En un proyecto real, se deben obtener un cierto número considerable de promedios acumulados de respuestas, usando diferentes torrentes de números para asegurarse de que realmente se ha alcanzado el estado estacionario.

S.E.D. 2010

Hospital

Resultados

Se simularon 14 días luego de alcanzada la estabilidad. Se utilizaron números aleatorios distintos que los utilizados para determinar el período run-in (resultados en págs. 110 y 111).

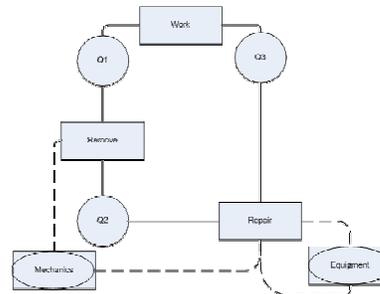
Observar:

- La distribución de las filas de "solo internados" y "pacientes a operar", tienen una varianza grande.
- Las camas han tenido un gran porcentaje de utilización (20 en 318 hs de 336 simuladas)
- 26 pacientes fueron operados y su tiempo de espera fue muy variado.
- Cada corrida con un conjunto de diferentes torrentes producen una replicación. Se necesitan varias replicaciones para obtener datos más acertados. Los datos ameritan reducción de varianza.

S.E.D. 2010

Taller de reparaciones

- Especificación: Tabla 6.3 y Figura 2.3
- Salidas de interés: tiempo medio de máquinas inactivas.
- Simulación terminal (8 horas de una jornada laboral).
- Se toman datos durante toda la simulación.
- Se podrían considerar las máquinas como variables de decisión.
- A tener en cuenta: cómo continuar (reparación de las máquinas) luego de terminada la jornada de trabajo (ver libro, página 130).



S.E.D. 2010

Taller de reparaciones

- Hipótesis adicionales:
 - Probabilidad de falla de todas las máquinas es igual al comienzo de la jornada (aunque algunas fueron recién reparadas)
 - Se ignoran tiempos de desplazamiento de equipamientos

S.E.D. 2010

Taller de reparaciones

Programa

Las máquinas son entidades permanentes, se crean todas al principio del programa

```
procedure initialize;
begin
  <inicializar calendario y torrentes>;
  <inicializar recursos>;
  <inicializar colas>;
  <crear máquinas>
  for <cada máquina m> do <causar falla en máquina m>;
end;
```

S.E.D. 2010

Taller de reparaciones

Programa

Finalización de la reparación de las máquinas una vez finalizada la jornada laboral

```
begin
  initialize;
  run(8*60,2);
  at_work := false;
  run(10000,2);
  report;
end.

procedure end_working;
begin
  if at_work then
    <give machine q1>;
  end;

procedure end_repair;
begin
  <return mechanic>;
  <return equipment>;
  if at_work then
    <cause new breakdown>;
  end;
```

S.E.D. 2010

Taller de reparaciones

- Resultados: el número de máquinas rotas varió entre 0 y 10.
- Quedaron 4 sin reparar al final de la jornada laboral.
- La utilización de mecánicos fue mayor que la de equipos (84.25% vs 68.7%).
- Durante un gran período de tiempo los mecánicos estuvieron todos ocupados.

S.E.D. 2010