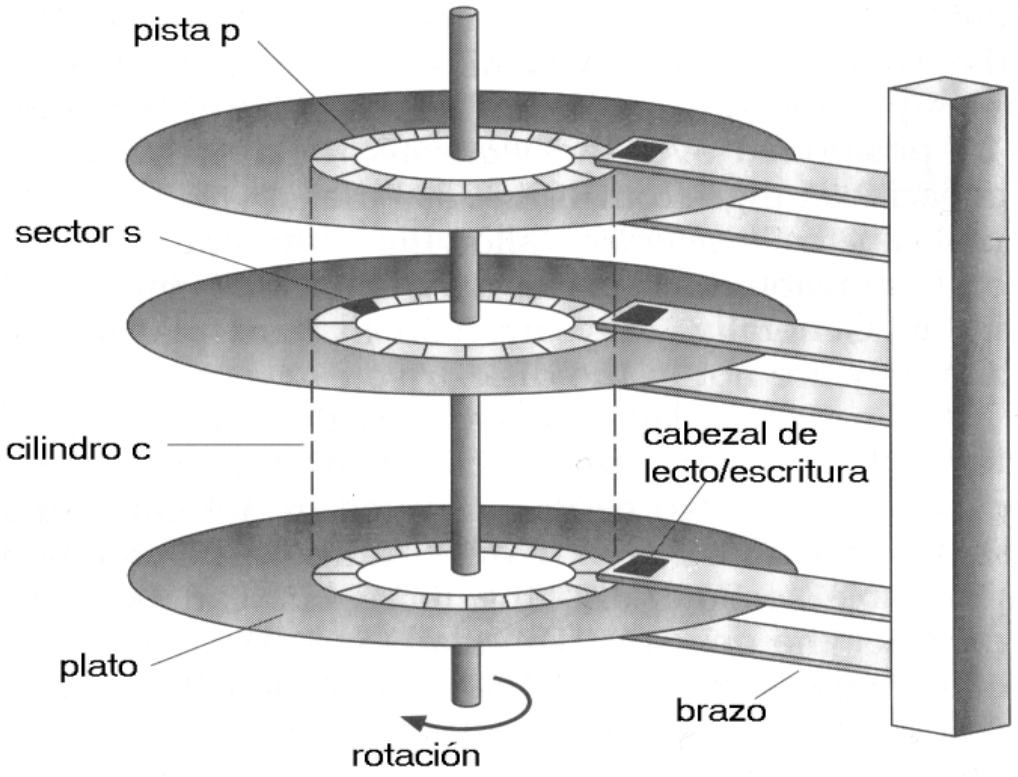
Simulación

Clase XV: Método de las tres fases - Caso de estudio

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Descripción del problema
Planificación de discos

- ***Estructura física de un disco para almacenamiento secundario:***

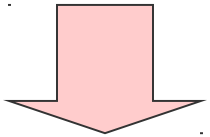


Descripción del problema
Planificación de discos

- El acceso a un bloque en un disco se compone de tres etapas:
  - ***mover la cabeza*** lecto/escritora a la pista/cilindro especificada → **Tiempo de búsqueda**
  - ***esperar a que el bloque se ubique*** sobre la cabeza lecto/escritora → **Tiempo de latencia**
  - ***leer el bloque***

Descripción del problema
Planificación de discos

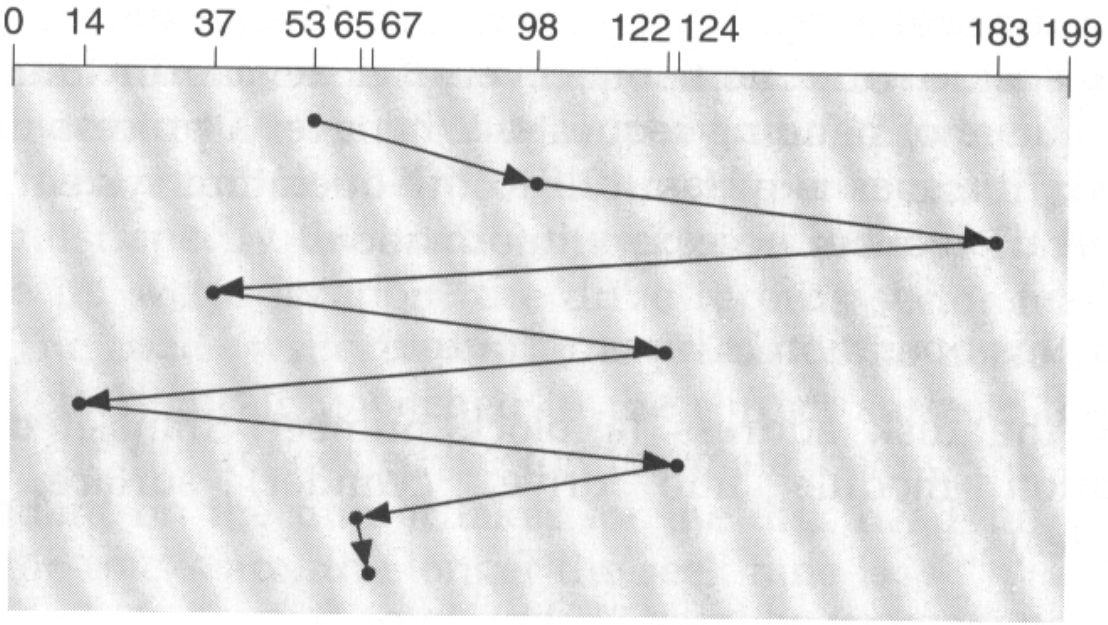
- El ***tiempo total*** de acceso a un bloque de disco lo vamos a considerar compuesto por:
  - ***tiempo de búsqueda (TB):*** proporcional a la distancia entre la ubicación actual y destino de la cabeza
  - ***tiempo de latencia (TL):*** 1/2 * tiempo de una rotación del disco (en promedio)
  - ***tiempo de lectura del bloque (TLB):*** unidad fija (incluye la lectura y la transferencia del bloque a memoria principal)



Tiempo Total = TB + TL + TLB

Descripción del problema
Planificación de discos

- *Esquema de planificación **FCFS** (primero en llegar-primero en ser atendido)*
- *Ejemplo: secuencia ⇨ 98,183,37,122,14,124,65,67 **



*** Por simplicidad se utiliza una notación lineal para la dirección de los bloques**

Descripción del problema
Planificación de discos

- Cada proceso en un S.O. que desea realizar una lectura o escritura en el disco debe hacerlo a través de un ***system call***.
- En cada requerimiento se debe indicar:
  - ***tipo de operación*** (*lectura/escritura*)
  - ***dirección en el disco*** (*pista+sector*)
  - ***dirección de memoria principal*** (donde se almacenarán los datos leídos o donde se contiene la información a escribir)

Descripción del problema
Planificación de discos

- El ***planificador de discos*** en un sistema operativo establece la secuencia de acceso para los bloques de datos que deben ser leídos o escritos.
- Cada disco posee una ***lista de requerimientos pendientes*** cuando éstos no pueden ser atendidos inmediatamente.
- La política elegida para la planificación de discos tiene un gran impacto en el desempeño de los procesos con un alto grado de utilización de almacenamiento secundario.

Construcción de los Diagramas de Ciclo de Actividades

- Identificación de las clases de ***entidades más relevantes***:
  - ***Cabeza lecto/escritora*** (servidor)
  - ***System calls*** pidiendo una operación de lectura/escritura (clientes)

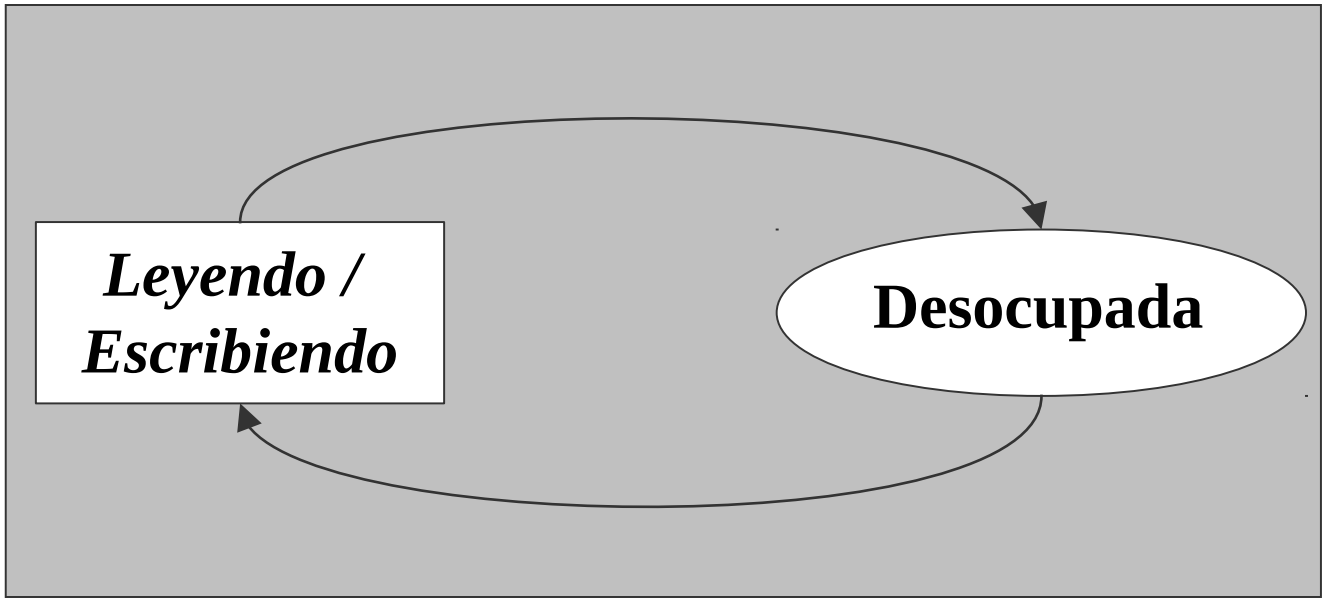
Construcción de los Diagramas de Ciclo de Actividades

- Identificación de los estados ***activos*** e ***inactivos*** por los que pasan las entidades identificadas:
  - ***Cabeza lecto/escritora*** (servidor)
    - La única actividad consiste en realizar la lectura/escritura

      por lo que identificaremos el ***estado activo***:
      - ***Leyendo/escribiendo*** (un sector en el disco).
    - Consideraremos además un ***estado inactivo***:
      - ***Desocupado*** para indicar los momentos en que el disco no realiza ninguna operación.

Construcción de los Diagramas de Ciclo de Actividades

- Ciclo de actividades de la ***cabeza lecto/escritora***

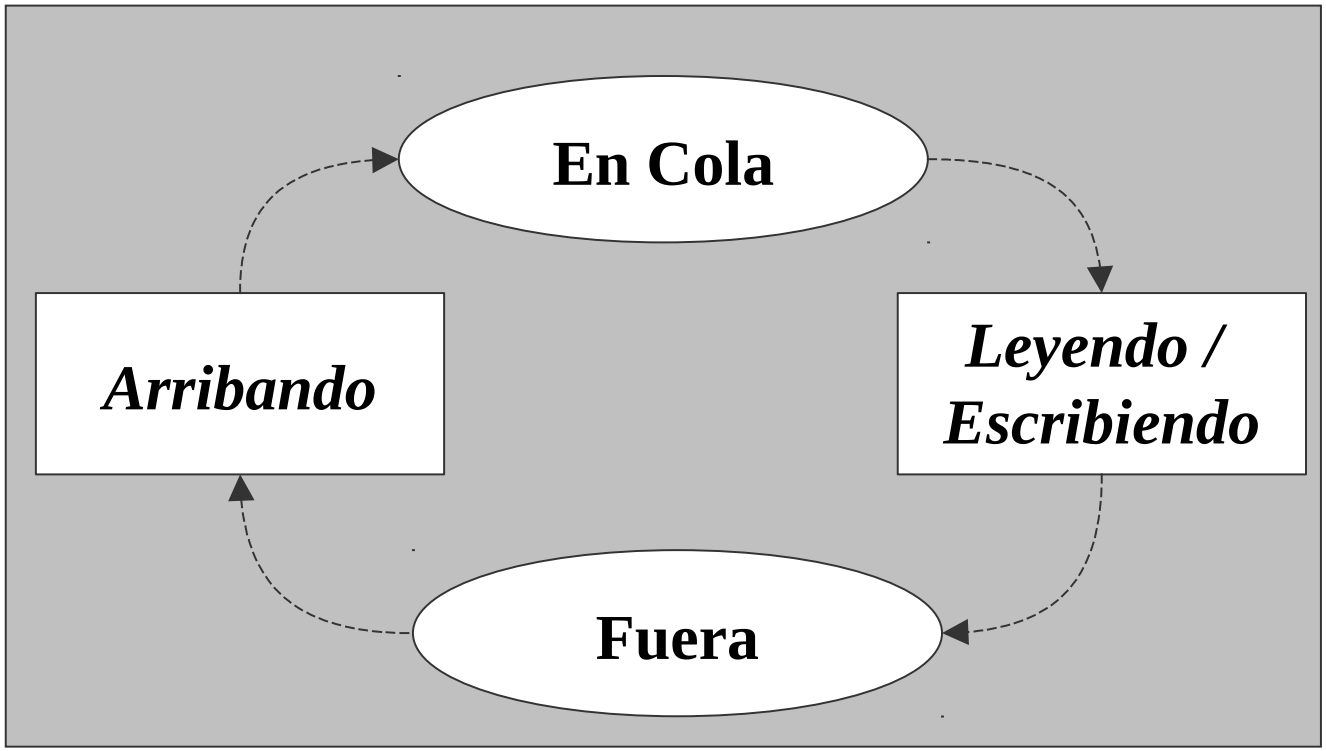


Construcción de los Diagramas de Ciclo de Actividades

- Identificación de los estados ***activos*** e ***inactivos*** por los que pasan las entidades identificadas:
  - ***System Call***
    - Una ***System call*** tendrá por otro lado dos estados activos:
      - ***Leyendo/escribiendo*** (un sector en el disco).
      - ***Arribando***
    - Y por lo tanto también pasará por dos estados **inactivos**:
      - ***Espera en la cola*** para ser atendida por la cabeza lecto-escritora y
      - ***Fuera*** para representar que aun no llegó o que ya salió del sistema.

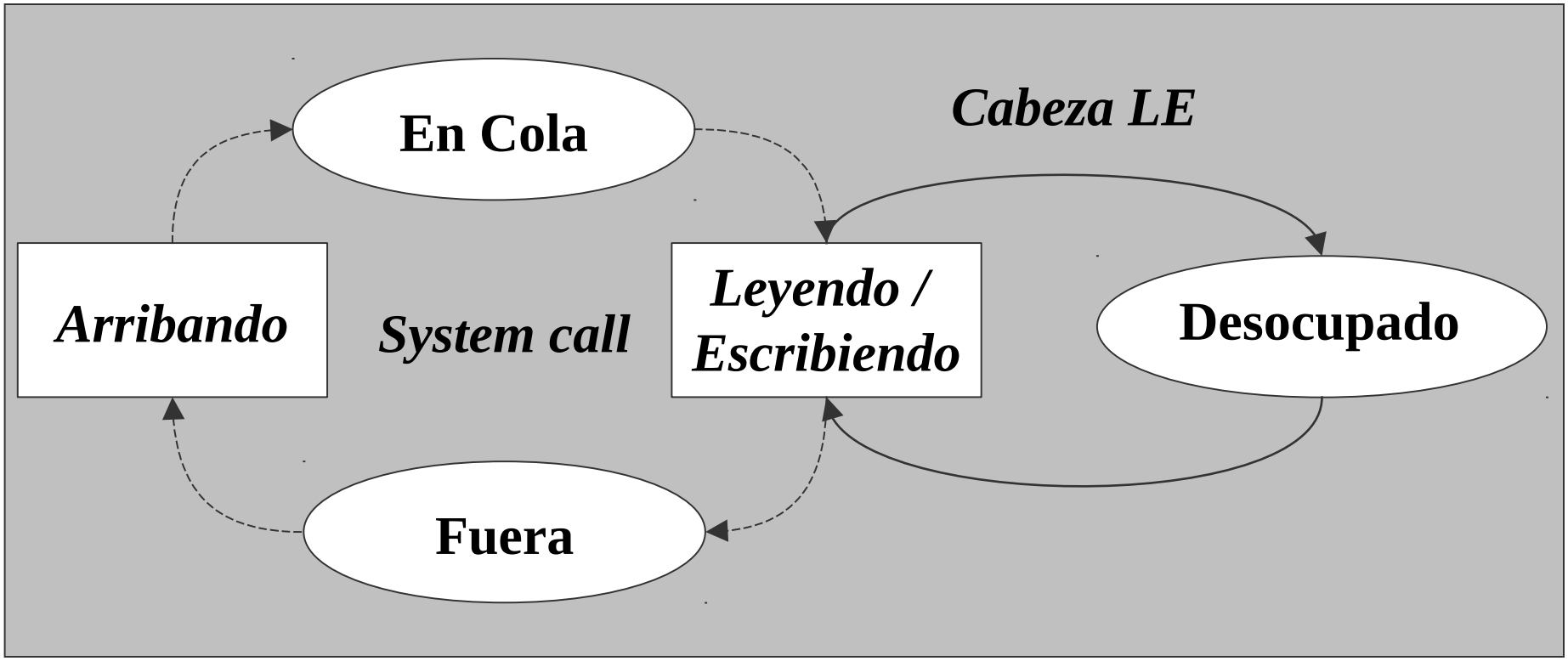
Construcción de los Diagramas de Ciclo de Actividades

- *Ciclo de actividades de los arribos de **system calls***



Construcción de los Diagramas de Ciclo de Actividades

- *Diagrama del Ciclo de actividades completo:*

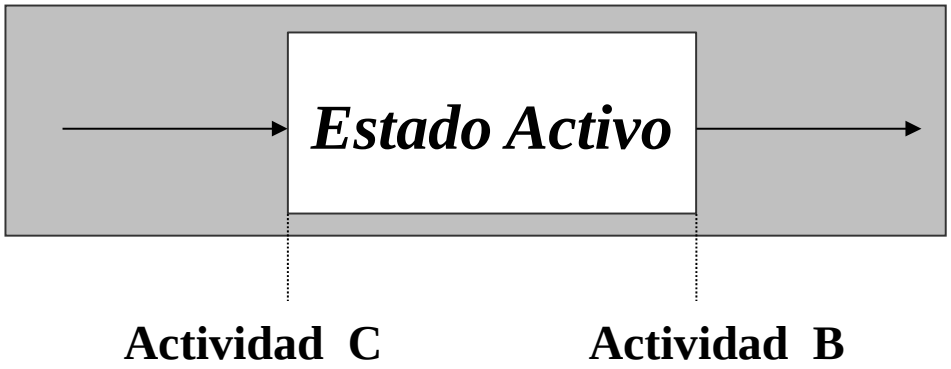


Enfoque de las tres fases
Identificación de Actividades

- Según el método de las tres fases se identifican dos tipos de actividades:
  - ***Actividades B:*** son los ***eventos que pueden ser planificados*** con antelación.
    - Se dan en actividades con tiempo de inicio o finalización que se conoce de forma anticipada.
    - Se ejecutará cuando el reloj de simulación llegue al tiempo planificado.
  - ***Actividades C:*** son los ***eventos que dependen de cierta Condición*** o Cooperación.

Enfoque de las tres fases
Identificación de Actividades

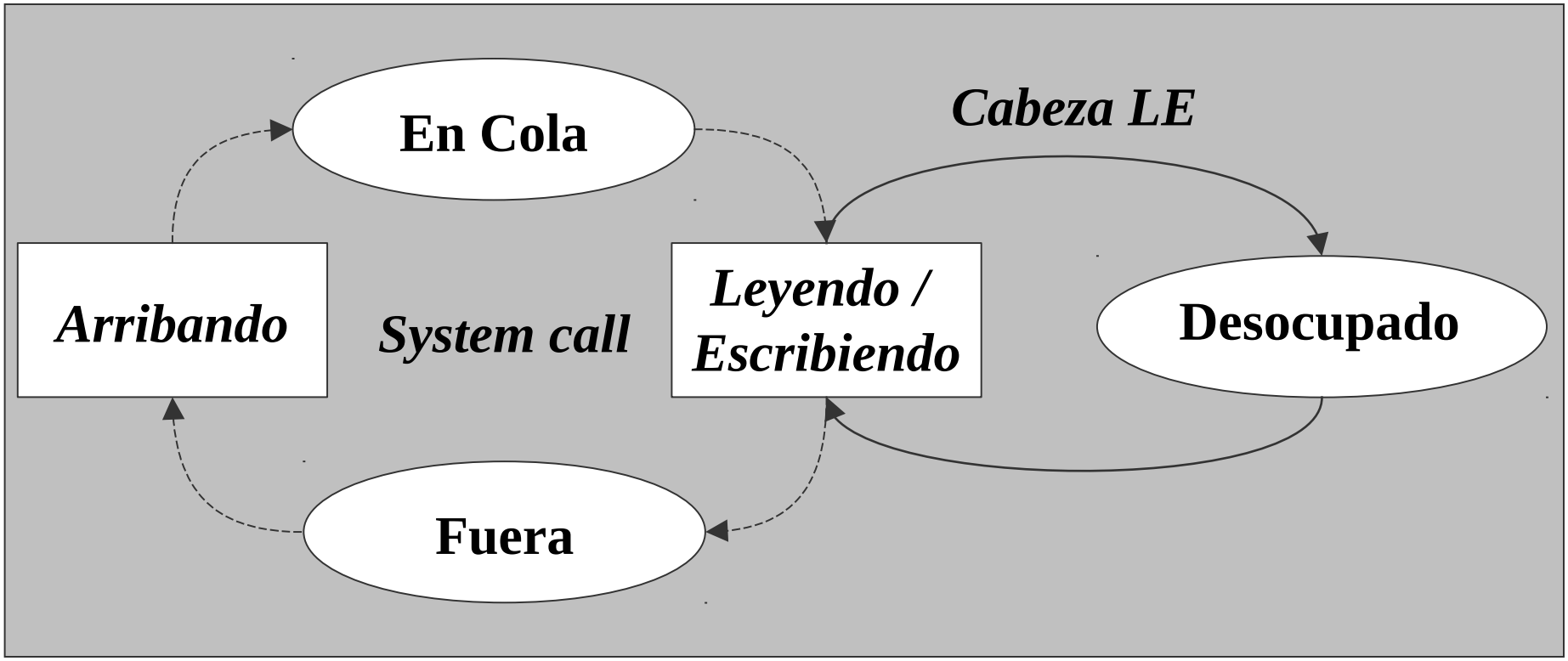
- Regla general:



- Excepción:
  - Estados Activos que generan nuevas entidades
  - En estos casos sólo modelamos la *Actividad B*

Enfoque de las tres fases
Identificación de Entidades

- Identificación de las Actividades B y C:

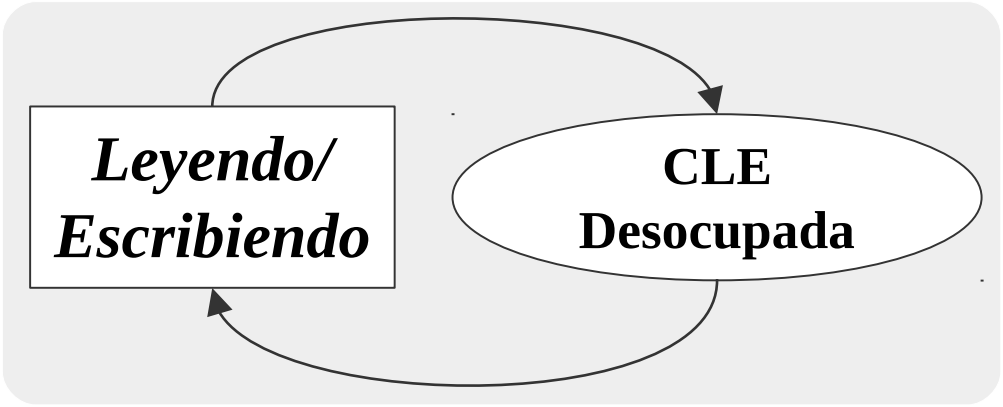


Enfoque de las tres fases
Identificación de las Entidades

- Las entidades identificadas en el DCA fueron:
  - ***Cabeza lecto/escritora*** (servidor)
  - ***System calls*** pidiendo una operación de lectura/escritura (clientes)
- Para llevar a cabo la implementación del método de las tres fases:
  - *Cabeza lecto/escritora* → ***ENTIDAD***
  - *System calls* → ***RECURSO***
  - *Máquina Generadora de Arribos* → ***ENTIDAD***

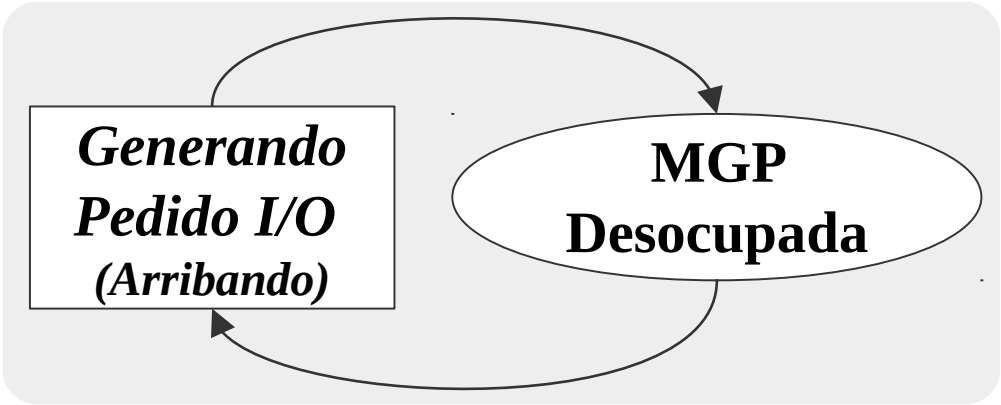
Enfoque de las tres fases

Identificación de las Entidades



Nuevo Diagrama

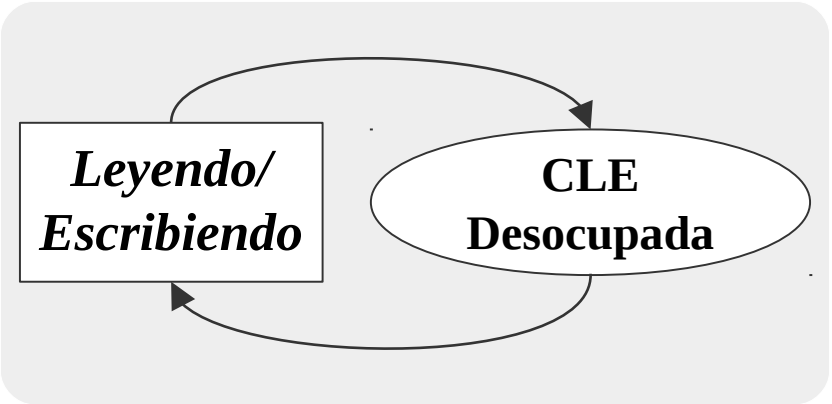
Cabeza L/E



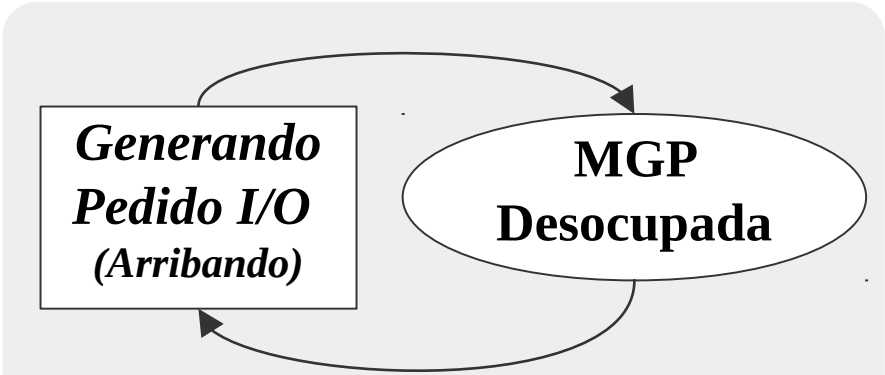
Máquina generadora de pedidos de I/O

Enfoque de las tres fases
Identificación de Actividades

- Identificación de las Actividades B y C:
  - ***Actividades B:***
    - *B1: Arribo de pedido de I/O*
    - *B2: Fin de lectura/escritura del sector*



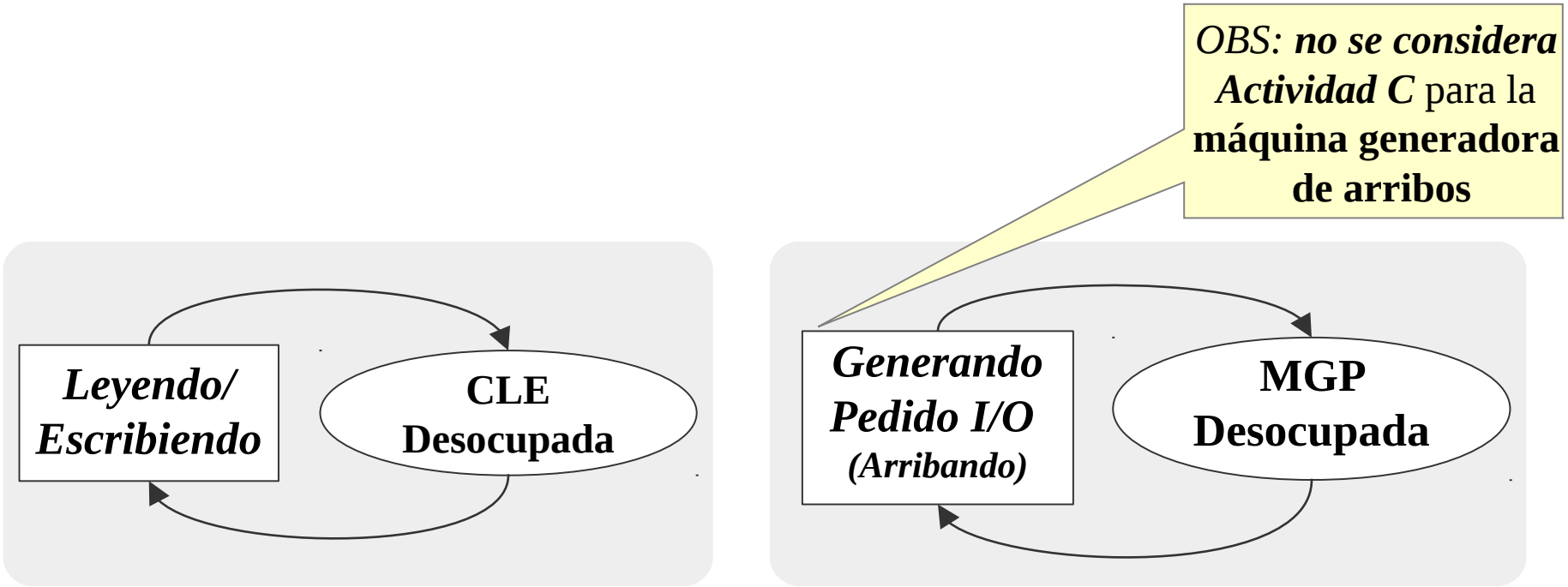
Cabeza L/E



Máquina generadora de pedidos de I/O

Enfoque de las tres fases
Identificación de Actividades

- Identificación de las Actividades B y C:
  - ***Actividades C:***
    - *C1: Comienzo de la lectura/escritura del sector*



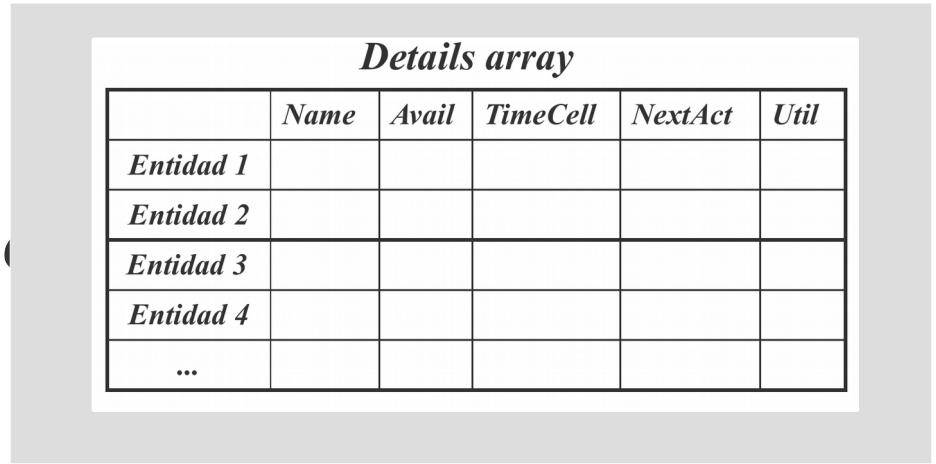
Cabeza L/E ***Máquina generadora de pedidos de I/O***

Enfoque de las tres fases
Estado de las Entidades

- Las entidades consideradas para la ***implementación***:
  - *Cabeza lecto/escritora*
  - *Máquina Generadora de Arribos*
- Para cada una de ellas mantendremos un registro con los siguientes datos:
  - ***Celda de tiempo:*** *t* de la próxima *B*.
  - ***Disponibilidad*.**
  - ***Próxima Actividad:*** *nombre* de la próxima *B*.

Enfoque de las tres fases
Estado de las Entidades

- Las entidades consideradas para la ***implementación***:
  - *Cabeza lecto/escritora*
  - *Máquina Generadora de Arribos*
- Para cada una de ellas mantendremos un registro con los siguientes datos:
  - ***Celda de tiempo:*** *t* de
  - ***Disponibilidad*****.**
  - ***Próxima Actividad:*** *n*



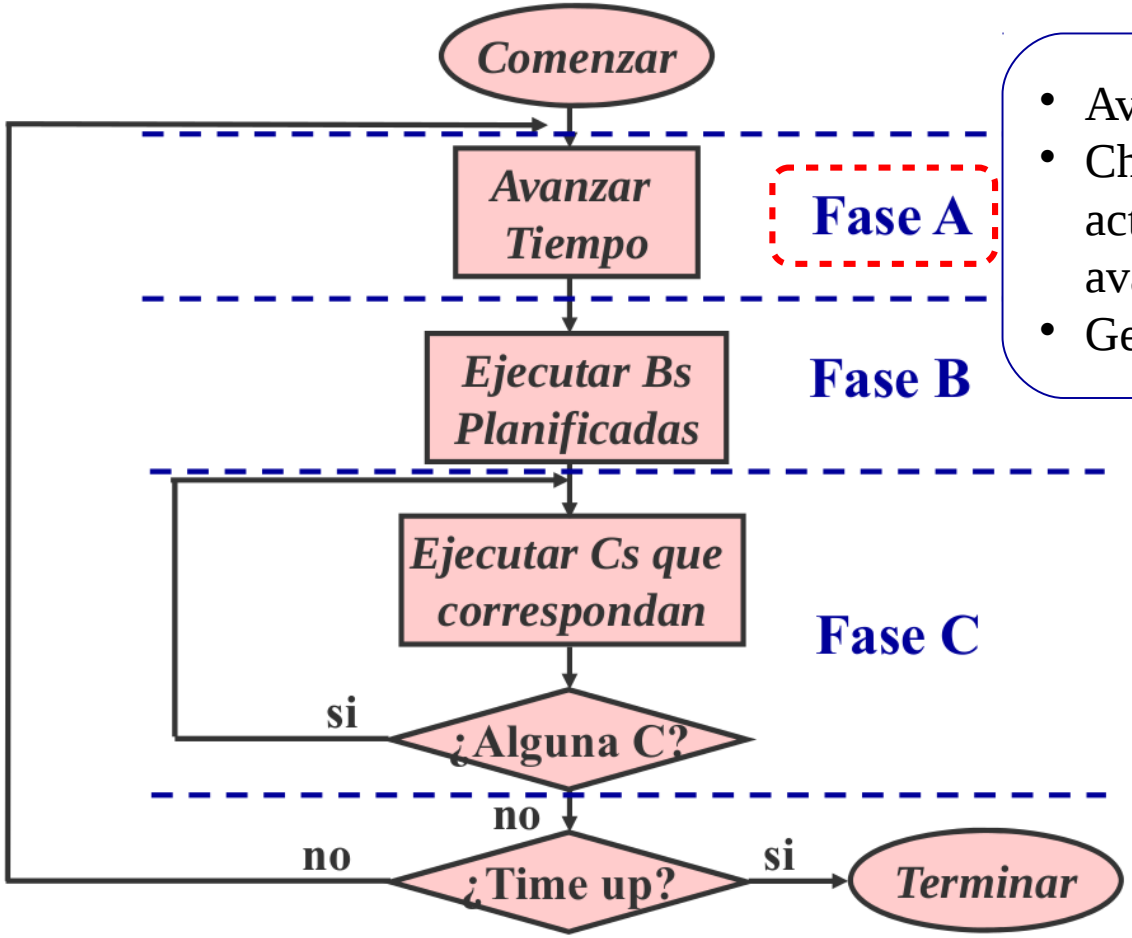
Enfoque de las tres fases
Estado de las Entidades

- Las entidades consideradas para la ***implementación***:
  - *Cabeza lecto/escritora*
  - *Máquina Generadora de Arribos*
- Para cada una de ellas mantendremos un registro con los siguientes datos:
  - ***Celda de tiempo:*** *t* de la próxima *B*.
  - ***Disponibilidad***

	Name	***Avail***	***Time***	***NextAct***	***Util***
Máquina generadora de Pedidos I/O					
Cabeza lecto/escritora					

Enfoque de las tres fases

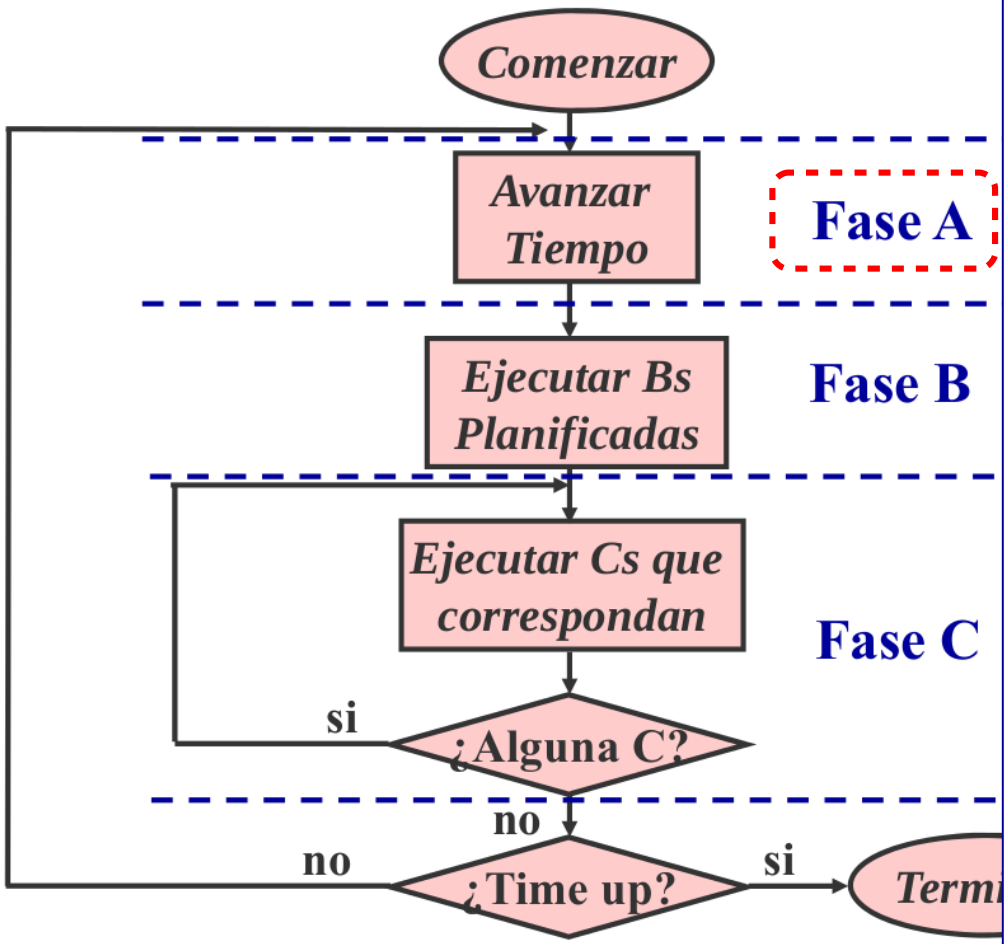
Ejecución de las Fases



- Avanzar T
- Chequear entidades buscando actividades B inminentes (el tiempo avanzará a este valor)
- Generar la lista HacerAhora

Enfoque de las tres fases

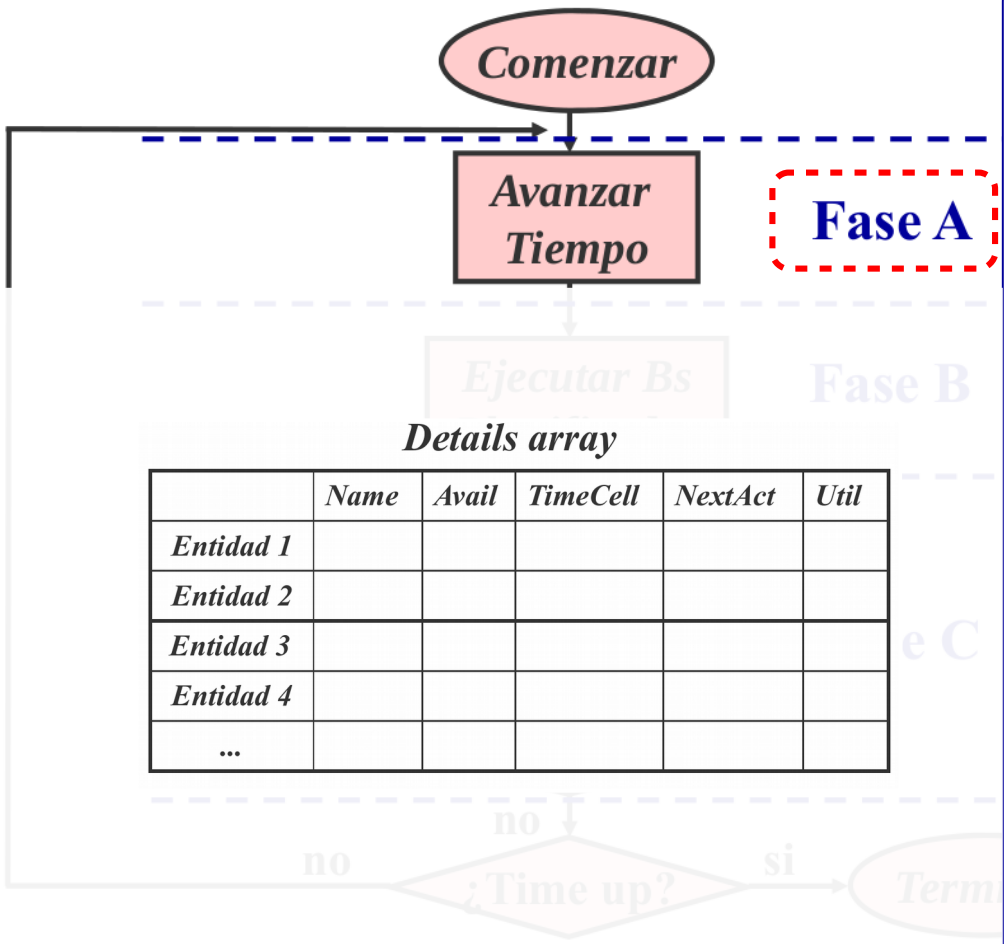
Ejecución de las Fases




```
Procedure Aphase;
Var Ent, mint: Integer;
Begin
 for Ent:= 1 to NumEnts do
  with Details[Ent] do
   If Not Avail then begin
    If Timecell <= Minm then
    begin
     If TimeCell < Minm then
        NumCurrEnts := 1
     Else NumCurrEnts :=
              NumCurrEnts + 1;
     Minm := TimeCell;
     CurrEntArray[NumCurrEnts]
          := Entity;
    End;
   End;
 ....
End;
```

Enfoque de las tres fases

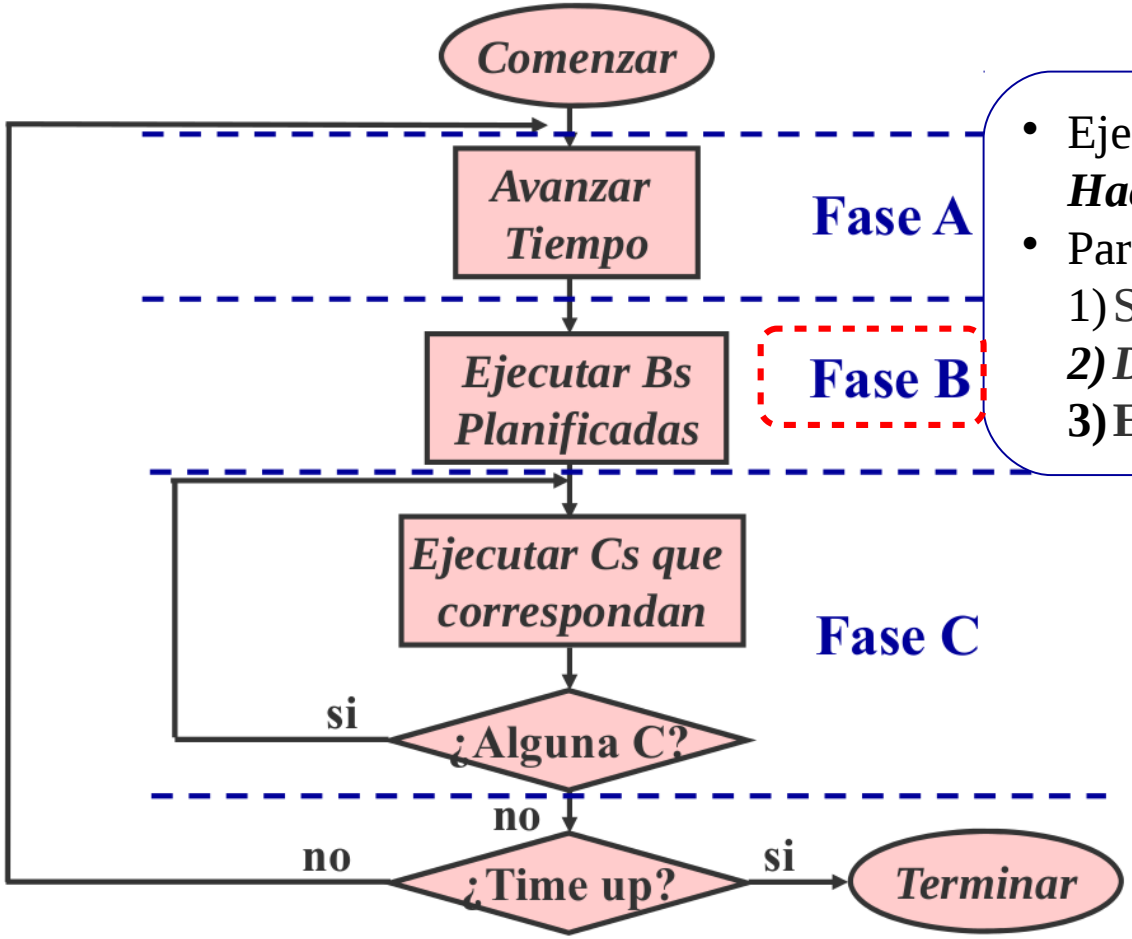
Ejecución de las Fases




```
Procedure Aphase;
Var Ent, mint: Integer;
Begin
 for Ent:= 1 to NumEnts do
  with Details[Ent] do
   If Not Avail then begin
    If Timecell <= Minm then
    begin
     If TimeCell < Minm then
        NumCurrEnts := 1
     Else NumCurrEnts :=
              NumCurrEnts + 1;
     Minm := TimeCell;
     CurrEntArray[NumCurrEnts]
          := Entity;
    End;
   End;
  ....
End;
```

Enfoque de las tres fases

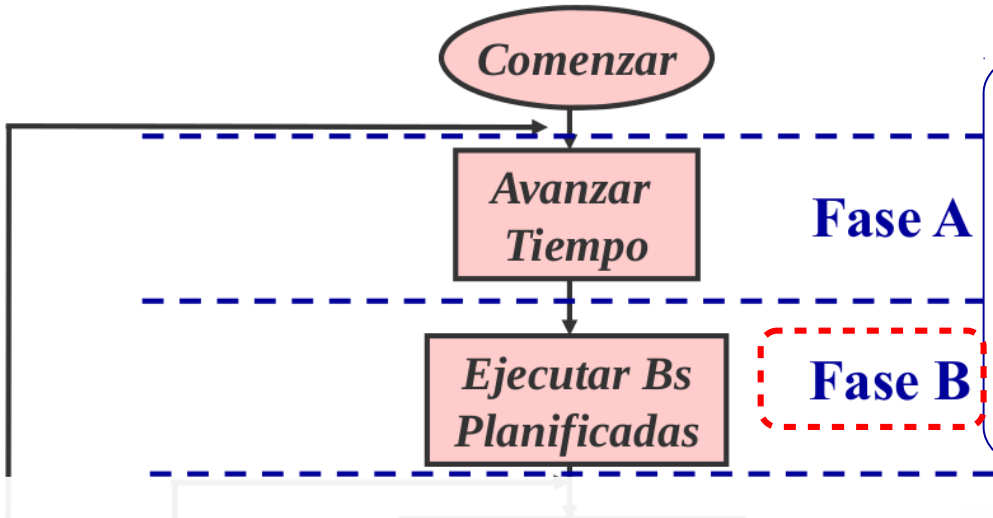
Ejecución de las Fases



- Ejecutar las actividades de la lista ***HacerAhora***
- Para cada entidad (no siempre):
  1) Sacar de ***HacerAhora***
  2) ***Disponibilidad*** ← TRUE
  3) Ejecutar *B*

Enfoque de las tres fases

Ejecución de las Fases



- Ejecutar las actividades de la lista ***HacerAhora***
- Para cada entidad (no siempre):
  1) Sacar de ***HacerAhora***
  2) ***Disponibilidad*** ← TRUE
  3) Ejecutar *B*

Details array

	Name	*Avail*	*TimeCell*	*NextAct*	*Util*
Entidad 1					
Entidad 2					
Entidad 3					
Entidad 4					
...					

CurrEntArray

	Entidad
CurrEnt1	*1*
CurrEnt2	*2*
CurrEnt3	*5*
...	

```
Details[ CurrEnt ]
```

Enfoque de las tres fases
Ejecución de las Fases

- **Identificación de las Actividades B y C:**
  - ***Actividades B:***
    - ***B1: Arribo de pedido de I/O***
    - ***B2: Fin de lectura/escritura del sector***
  - ***Actividades C:***
    - ***C1: Comienzo de la lectura/escritura del sector***

Enfoque de las tres fases
Ejecución de las Fases

- **Identificación de las Actividades B y C:**
  - ***Actividades B:***
    - ***B1: Arribo de pedido de I/O***
    - ***B2: Fin de lectura/escritura del sector***
  - ***Actividades C:***
    - ***C1: Comienzo de la lectura/escritura del sector***

¿Que se debe realizar en cada una de estas actividades?

Enfoque de las tres fases
Ejecución de las Fases

- **Identificación de las Actividades B y C:**
  - ***Actividades B:***
    - ***B1: Arribo de pedido de I/O***
      - Sacar a la Máquina de la lista ***HacerAhora***
      - Actualizar las variables de estado (número de pedidos de I/O, etc)
      - Generar el ***tiempo del próximo arribo***
      - **No** se modifica la ***disponibilidad.***

Enfoque de las tres fases

Ejecución de las Fases

- Identificación de las Actividades B y C:
  - ***Actividades B:***
    - ***B1: Arribo de pedido de I/O***
      - Sacar a la Máquina de la lista ***HacerAhora***
      - Actualizar las variables de estado (número de pedidos de I/O, etc)
      - Generar el ***tiempo del próximo arribo***
    - **No** se modifica la ***disponibilidad.***

```
Procedure Actividad_ArriboPedidoIO;
{Generación de pedidos de I/O}
Var Time : Real;

Begin
 {Incrementa el número de arribos d
 Inc(NumeroArribosLectura);
 Time := NegExp(TiempoArriboLectura
 {Incrementa el número de pedidos q
 Inc(NumeroArribosLectura_EnCola);
 Display('Pedido de I/O No. '+
          Int2Str(NumeroArribosLect
          ',tamaño cola ='+
          Int2Str(NumeroArribosLect
 {Generar el próximo pedido de I/O}
 Schedule(MaquinaArribos,
          Actividad_ArriboPedidoIO,
          Time);
End;
```

Enfoque de las tres fases

Ejecución de las Fases

```
Procedure Actividad_ArriboPedidoIO;
{Generación de pedidos de I/O}
Var Time : Real;

Begin
 {Incrementa el número de arribos de pedidos}
 Inc(NumeroArribosLectura);
 Time := NegExp(TiempoArriboLectura, 1);
 {Incrementa el número de pedidos que están en la cola}
 Inc(NumeroArribosLectura_EnCola);
 Display('Pedido de I/O No. '+
          Int2Str(NumeroArribosLectura,3)+
          ',tamaño cola =‘+
          Int2Str(NumeroArribosLectura_EnCola,3));
 {Generar el próximo pedido de I/O}
 Schedule(MaquinaArribos,
          Actividad_ArriboPedidoIO,
          Time);
End;
```

Enfoque de las tres fases
Ejecución de las Fases

- **Identificación de las Actividades B y C:**
  - ***Actividades B:***
    - ***B2: Fin de lectura/escritura del sector***
      - Sacar a la Máquina de la lista ***HacerAhora***
      - Actualizar las variables de estado (número de pedidos de I/O, etc)

Enfoque de las tres fases

Ejecución de las Fases

- Identificación de las Actividades B y C:
  - ***Actividades B:***
    - ***B2: Fin de lectura/escritura del sector***
      - Sacar a la Máquina de la lista ***HacerAhora***
      - Actualizar las variables de estado (número de pedidos de I/O, etc)

```
Procedure Actividad_FinPedidoIO;
{Fin del servicio de I/O.}
Begin
 {Incrementa el número de servicios
 cabo}
 Inc(NumeroLecturasCompletadas);
 Display('Fin de dervicio de I/O:'
          Int2Str(NumeroLecturasCom
 SectorActual := NuevoSector;
End;
```

Enfoque de las tres fases

Ejecución de las Fases

```
Procedure Actividad_FinPedidoIO;
{Fin del servicio de I/O.}
Begin
 {Incrementa el número de servicios de I/O llevados a
 cabo}
 Inc(NumeroLecturasCompletadas);
 Display('Fin de dervicio de I/O:' +
          Int2Str(NumeroLecturasCompletadas,3));
 SectorActual := NuevoSector;
End;
```

Enfoque de las tres fases
Ejecución de las Fases

- **Identificación de las Actividades B y C:**
  - ***Actividades C:***
    - ***C1: Comienzo de la lectura/escritura del sector***

Test de condición: Si la cabeza lecto/escritora esta
Disponible y hay un pedido de I/O esperando

Acciones:

-Tomar al primer pedido de I/O
- Poner FALSO en la disponibilidad de la entidad
"Cabeza lecto/escritora"
- Calcular un tiempo de finalización de operación de I/O
- Planificar la finalización

Enfoque de las tres fases
Ejecución de las Fases

- **Identificación de las Actividades B y C:**

```
Procedure Actividad_ComienzoIO;
{Ejecutar el primer pedido de lectura en la cola}
Var Time : Real;
    Distancia: Integer;
Begin
 If (NumeroArribosLectura_EnCola > 0) then
  If Details[Server].Avail then Begin
    CStarted := True;
    NuevoSector := Random(cMaximoSector-cMinimoSector+1);
    Distancia := abs(NuevoSector - SectorActual);
    Dec(NumeroArribosLectura_EnCola);
    Time := Distancia*0.2 + cLatenciaRotacional*0.5 + cTiempoLectura;
    TiempoServicioTotal := TiempoServicioTotal + Time;
    Display('Comienzo I/O sector:'+Int2Str(NuevoSector,3));
    Schedule(Server, Actividad_FinPedidoIO, Time);
  End;
End;
```

Otra Actividad B

```
Procedure Actividad_Obervar;
{Obtiene información de la corrida - checkpoint}
Begin
  LecturasObservadas[Obs] := NumeroArribosLectura_EnCola;
  Inc(Obs);
  Schedule(Observador, Actividad_Obervar,
           IntervaloObservacion);    {Proxima observación}
  Display('Ejecución, Cola de pedidos de I/O = ' +
           Int2Str(NumeroArribosLectura_EnCola,3));
End;
```

Inicialización

```
Procedure Initialisation;
Begin
 GetRunPars;                {Lee los parámetros de la corrida}
 InitStreams;               {Inicializa el algoritmo de generación
                             de números aleatorios}
 ZeroLimits;                {Setea los parametros a cero}
 InitCounters;              {Inicializa los contadores}
 InitObservations;          {Inicializa vars. para observación}
 InitEnt(Actividad_ArriboPedidoIO,-12*Ln(Rnd(1)),
         'Pedido I/O ');
 InitEnt(Actividad_Obervar, 20, 'Observador  ');
 {Pone al servidor inicialmente libre}
 InitEntFree('Servidor  ');
 AddC(Actividad_ComienzoIO);
 {Inicializa el SectorActual donde se encuentra la cabeza}
 SectorActual := 0;
 TiempoServicioPromedio := 0;
 TiempoServicioTotal := 0;
 ScreenBkGrnd(True);
End;
```