

# *Sistemas en Tiempo Real: Introducción*

---

***Dr. Pedro Mejía Alvarez***  
***Cinvestav-IPN. Sección de Computación***  
***pmejia@cs.cinvestav.mx***  
***<http://delta.cs.cinvestav.mx/~pmejia>***

# *Contenido*

---

- ◆ Que es un Sistema en Tiempo Real.
- ◆ Predecibilidad.
- ◆ Tipo de de tareas.
- ◆ Planificación de Sistemas de Tiempo Real
- ◆ Aplicaciones de Sistemas en Tiempo Real

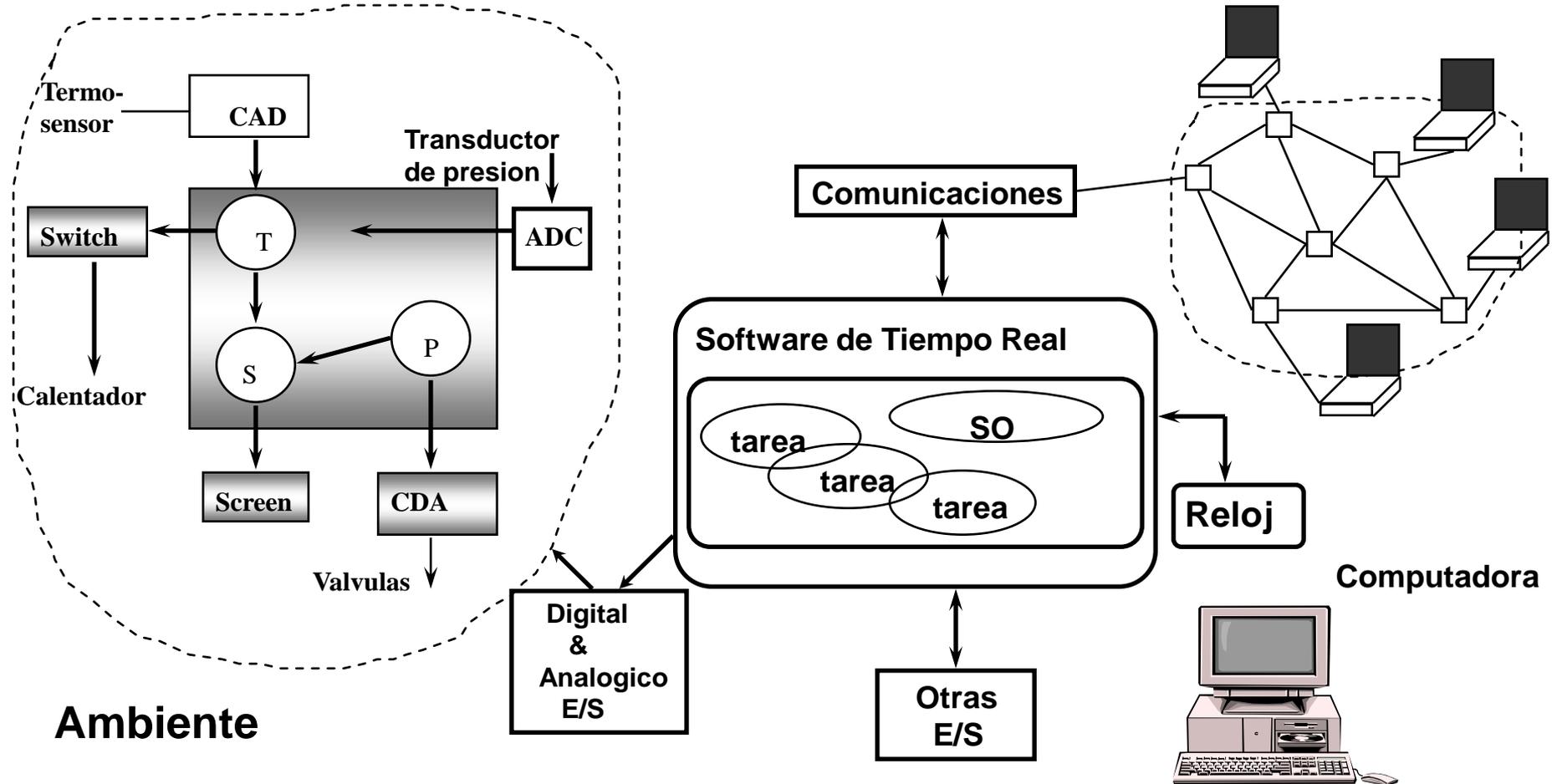
# Que es un Sistema en Tiempo real ?

---

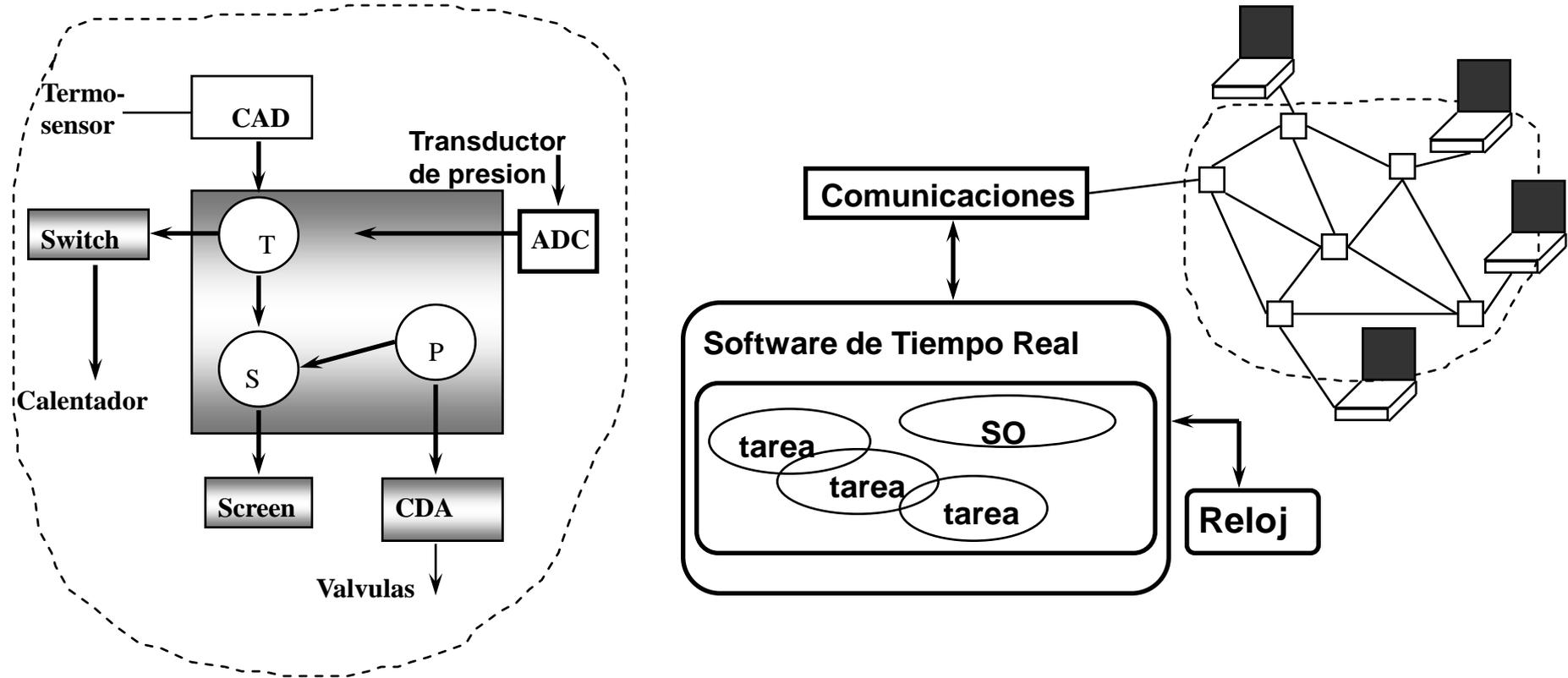
*Un sistema de tiempo real es un sistema de procesamiento de información el cual tiene que responder a estímulos de entrada generados externamente en un período finito y específico.*

- ◆ Las respuestas correctas dependen no solo de los resultados lógicos sino también del tiempo en que son entregadas.
- ◆ Las fallas para responder a tiempo son tan malas como una mala respuesta!.

# Elementos de un Sistema de Tiempo Real



# Elementos de un Sistema de Tiempo Real



# *Características de un Sistema de Tiempo Real*

---

## ◆ Tiempo.

- ◆ Administración y control del tiempo.
- ◆ Tareas deben ser asignadas y terminadas antes de su plazo.
- ◆ La ejecución correcta no solo considera la lógica sino también el tiempo en que se producen los resultados.

## ◆ Confiabilidad.

- ◆ Predecibilidad
- ◆ Tolerancia a Fallos.
- ◆ Seguridad.

## ◆ Ambiente.

- ◆ Características dinámicas del ambiente.

# *Características de un Sistema en Tiempo Real*

---

Un sistema en tiempo real es una combinación de computadoras, dispositivos de E/S, hardware y software de propósito específico en donde:

- ◆ existe una fuerte interacción con el ambiente.
- ◆ el ambiente cambia con el tiempo
- ◆ el sistema debe controlar y/o reaccionar a diferentes aspectos del ambiente.

## **Como resultado:**

- ◆ Se imponen restricciones de tiempos al software.
- ◆ El software es naturalmente concurrente.
- ◆ Se exige una alta confiabilidad.

# *Restricciones de un sistema en tiempo real*

---

- ◆ **Restricciones de tiempos:** (Computo, Periodo, Plazos).
- ◆ **Restricciones de predecibilidad.**
- ◆ **Restricciones de recursos:** una tarea puede requerir acceso a ciertos recursos, además del procesador, como dispositivos de E/S, redes de comunicación, estructuras de datos, archivos y bases de datos.
- ◆ **Restricciones de precedencia:** una tarea puede requerir resultados de una u otra tarea antes de comenzar su ejecución.
- ◆ **Restricciones de confiabilidad y desempeño:** una tarea podría tener que cumplir con ciertas restricciones de confiabilidad, disponibilidad o desempeño.

# *Predecibilidad*

---

Una característica distintiva de un sistema en tiempo real es la predecibilidad. La cual implica que debe ser posible demostrar o comprobar a priori que los requerimientos de tiempos se cumplen en cualquier circunstancia.

Como consecuencia, la predecibilidad implica:

- ◆ una cuidadosa planificación de tareas y recursos.
- ◆ cumplimiento predecible de requisitos temporales: determinismo.
- ◆ anticipación a fallos, y sus requerimientos temporales.
- ◆ consideraciones de sobrecargas: degradación controlada.
- ◆ consideraciones de elementos de impredecibilidad.
- ◆ dotar al sistema con capacidades de monitorización y control de tiempos (hardware, software, sistema operativo, lenguaje, líneas y protocolos de comunicaciones).

# *Tipo de Tareas*

---

## ◆ De Tiempo Crítico

- ◆ Las tareas deben completarse antes de su plazo de respuesta

## ◆ Acriticas

- ◆ Las tareas deben completarse tan pronto como sea posible

## ◆ Tareas Periódicas

- ◆ Reiniciación periódica de tareas, cada instancia debe completar antes de su plazo. Ejemplo: Señal de 60 Hz

## ◆ Tareas Aperiódicas

- ◆ Se activan una sola vez, p.ej, dispara un misil y verifica el blanco.

# *Tipos de Garantías*

---

- ◆ Los sistemas de TR requieren una garantía del cumplimiento de que las tareas cumplirán sus plazos de respuesta.
- ◆ Tipos de Garantías
  - ◆ Determinística - Tarea  $i$  siempre terminará antes de su plazo
  - ◆ Estadística - Tarea  $i$  terminará antes de su plazo la mayor parte del tiempo. terminará lo mas pronto posible.

# *Utilidad y Efectividad de un Sistema en Tiempo Real*

---

- ◆ Si el sistema pierde un plazo:
  - ◆ De ninguna utilidad
  - ◆ Parcialmente útil
  - ◆ No solo inútil, sino peligroso

# *Planificación = Tecnología que permite dar garantías de ejecución al STR*

## ◆ Problema

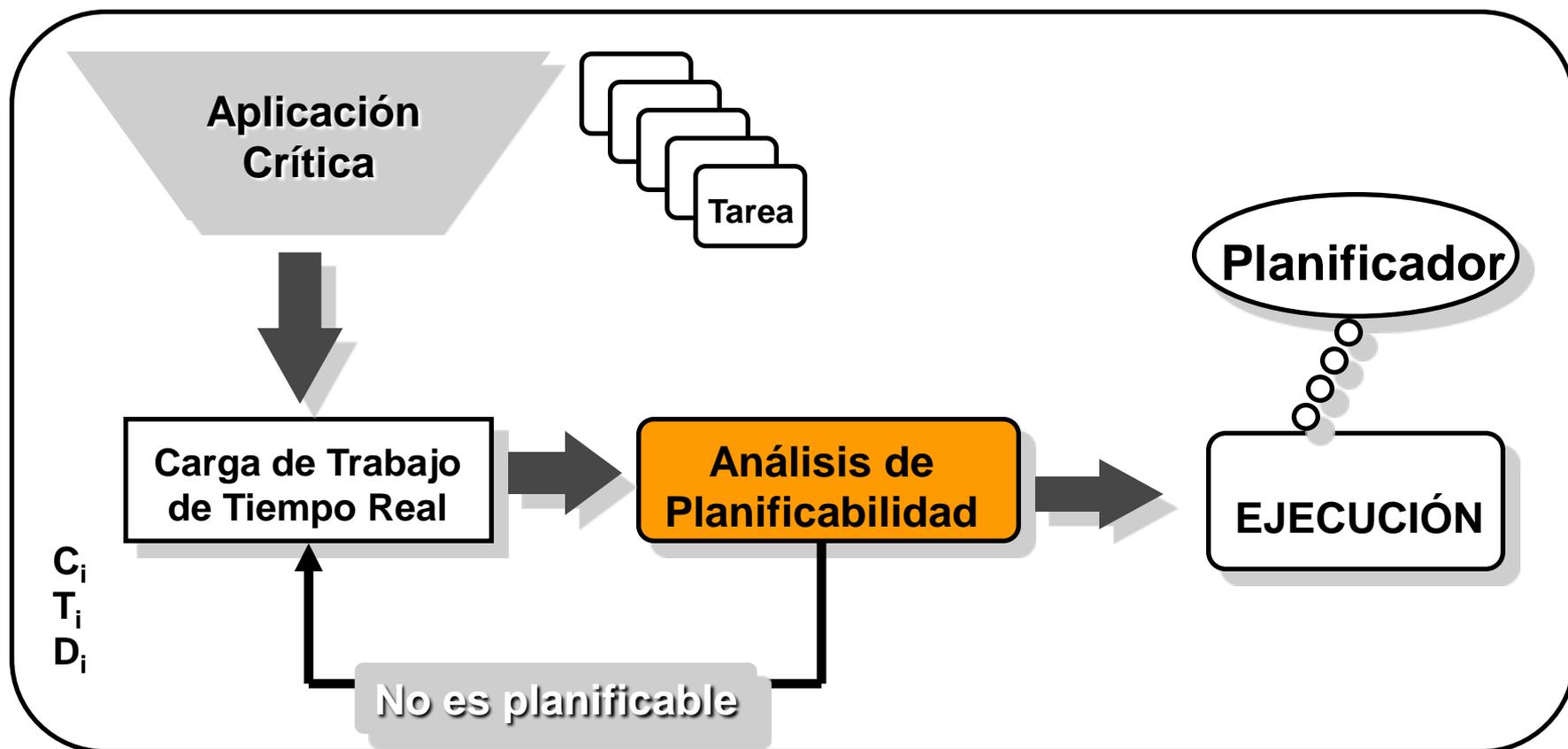
- ◆ Un conjunto de procesos requieren el acceso a un recurso compartido: Procesador, medio de comunicación, dispositivos.
- ◆ Si no existe contención suficiente:
  - ◆ Todos los procesos comparten adecuadamente el recurso
- ◆ Si existe algún grado de contención en el sistema:
  - ◆ Diseño - quien obtiene que prioridades, y cuando ?
  - ◆ Cómo se asigna el acceso a los recursos compartidos ?
  - ◆ Problemas de concurrencia

# *Planificación vs. Planificabilidad*

---

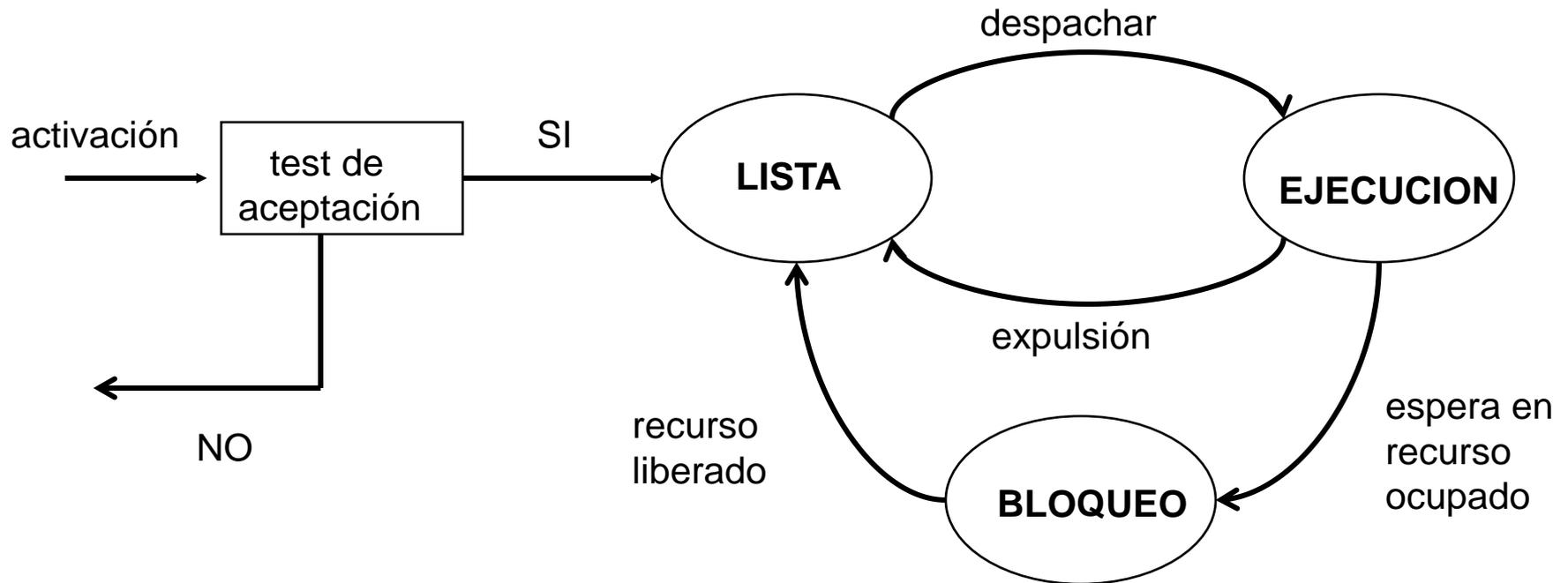
- ◆ Planificación = enfoque para dar prioridad a una tarea sobre otra.
- ◆ Planificabilidad = análisis formal que permite garantizar que un conjunto de tarea periódicas pueden (o no) ser asignadas a un recurso compartido (p.ej. el procesador) tal que cada una de las tareas cumpla con su plazo de respuesta.
- ◆ Planificabilidad = Control de admisión

# Proceso de Planificación

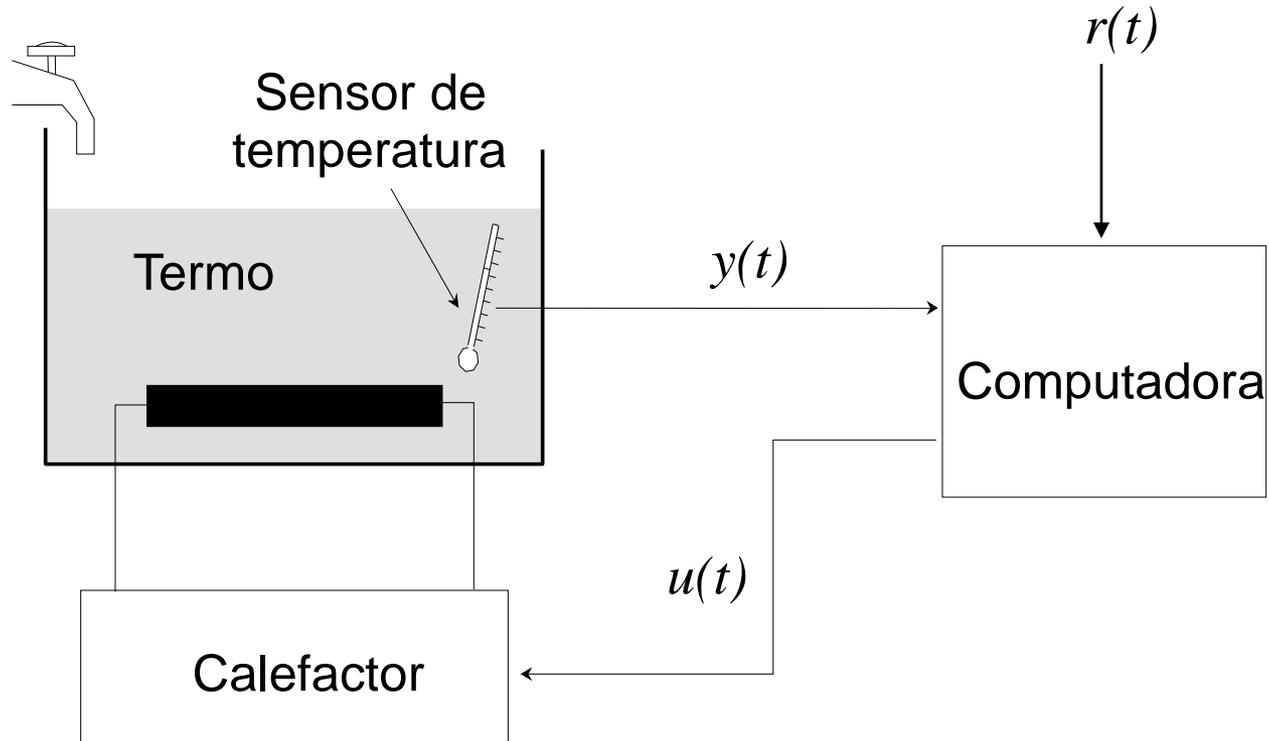


# Proceso de Planificación (estados)

---



# Caracterización de la Aplicación



- **Componentes:** tanque, sensores, actuadores, interfaces A/D y computadora.

# *Caracterización de la Aplicación*

---

## **Restricciones de Funcionamiento.**

- ◆ Componentes: tanque, sensores, actuadores, interfaces A/D y computadora.
- ◆ En el sistema (planta), dada una  $u(t)$ , se produce una  $y(t)$ .
- ◆ El sistema recibe una señal de referencia  $r(t)$ , el cual indica la temperatura deseada.
- ◆ El objetivo del sistema de control es que  $y(t)$ , sea lo mas parecida a  $r(t)$ .
- ◆ Para lograr el objetivo, es preciso aplicar a la planta la señal  $u(t)$
- ◆  $y(t)$  muestra el valor real de la temperatura del termo.

# Caracterización de la Aplicación

---

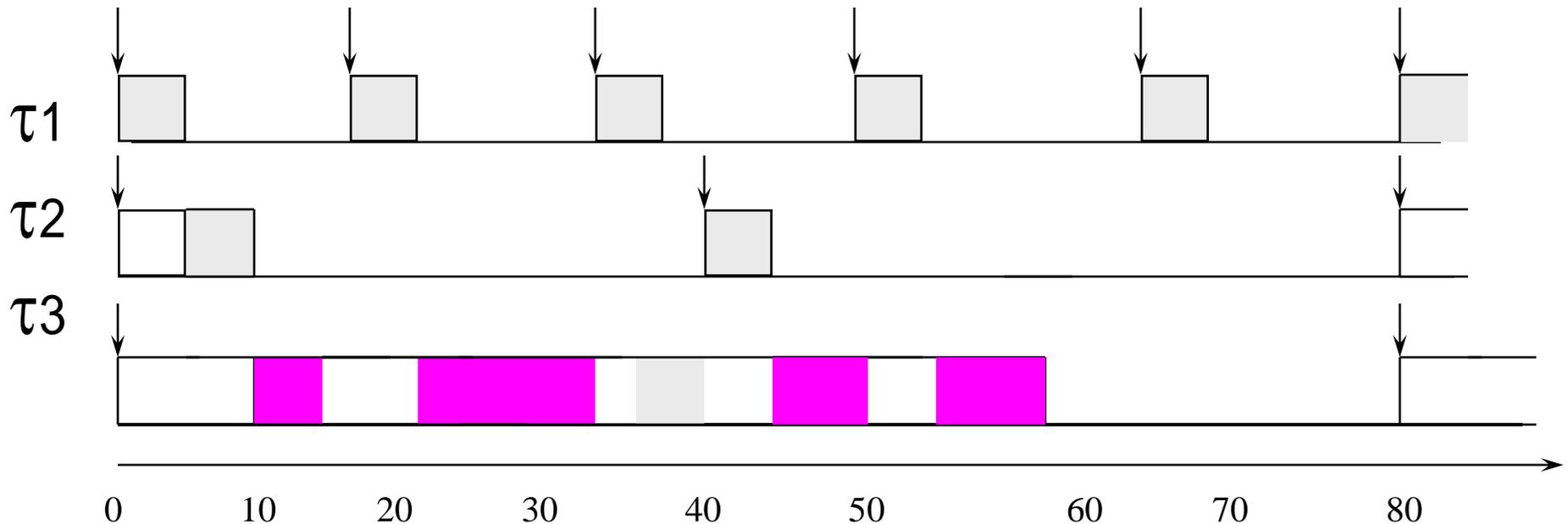
## Restricciones de Tiempos:

- Tiempos de cómputo.
  - Periodos de muestreo.
  - Tiempos de acceso a los dispositivos (sensores actuadores)
  - Plazos de Respuesta.
  - Análisis de Planificación.
  - Planificar el sistema con un Sistema Operativo.
- Los tiempos hay que obtenerlos, ya sea mediante mediciones instrucción por instrucción, mediante analizadores lógicos o osciloscopios digitales.*
- El periodo de muestreo y el plazo de respuesta dependen de la aplicación.*
- Se trata de asegurar que:*

$$R_j \leq D_j$$

# Ejemplo

<i>Tarea</i>	<i>T</i>	<i>C</i>	<i>P</i>	<i>U</i>
$\tau_1$	16	4	1	0.250
$\tau_2$	40	5	2	0.125
$\tau_3$	80	32	3	0.400
				0.775



# *Tipos de Planificación*

---

## ◆ Planificación Estática

- ◆ Estimación a-priori de las restricciones temporales de todas las tareas del sistema
- ◆ Opera sobre un conjunto de tareas no necesariamente estáticas
- ◆ Asignación estática de prioridades:
  - ◆ Tabla estática.
  - ◆ Asignación de prioridades estáticas.

## ◆ Planificación Dinámica

- ◆ Los parámetros de las tareas (C,T,D) no son conocidas a priori
- ◆ Las tareas pueden arribar en forma dinámica. De la misma forma las prioridades de las tareas se asigna durante la ejecución.
- ◆ No puede dar garantía de cumplimiento de plazos a priori

# *Sistemas de Tiempo Real Críticos*

---

## **Que tipos de Sistemas de tiempo real son críticos ?**

- Tiempo Real no es cómputo rápido.
- Como comprobar/verificar que siempre se cumplen los plazos en un sistema de tiempo real crítico ?
- Implementar mecanismos para monitorizar el estricto cumplimiento de tiempos
- Que consecuencia trae la pérdida de plazos en este tipo de Sistemas.

# *Sistemas de Tiempo Real No Críticos*

## **Que tipos de Sistemas de Tiempo Real son no Críticos ?**

- **Restricciones de Tiempo Tolerantes:** algunos plazos se permite que se pierdan:
- **n de m plazos perdidos.** (% por ciento de plazos perdidos)
- **Tareas con Cómputo obligatorio y computo opcional.**
- **Calidad de servicio:** (obtener 70% de los mensajes de la red).
- **Mejor esfuerzo:** que se cumplan el mayor número de plazos posibles.

### **(a) Requerimientos de tiempos sin restricciones:**

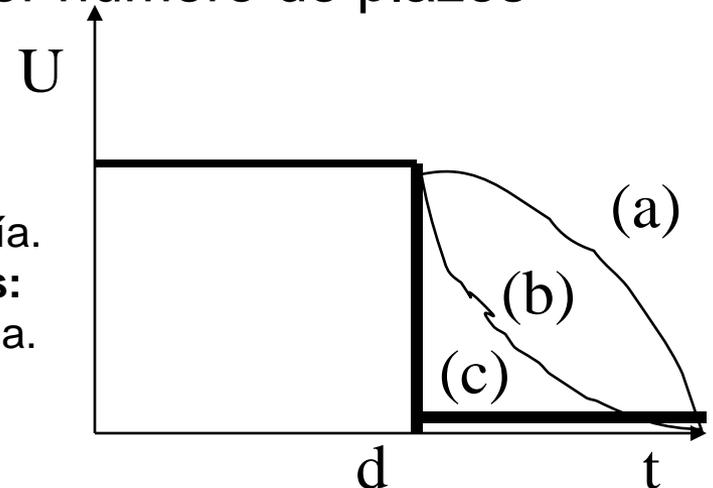
- Sistemas de transacciones en línea, Switches de telefonía.

### **(b) Requerimientos de plazos con pocas restricciones:**

- Sistema de obtención de precios de cotizaciones en bolsa.

### **(c) Requerimientos de plazos con restricciones:**

- Multimedia

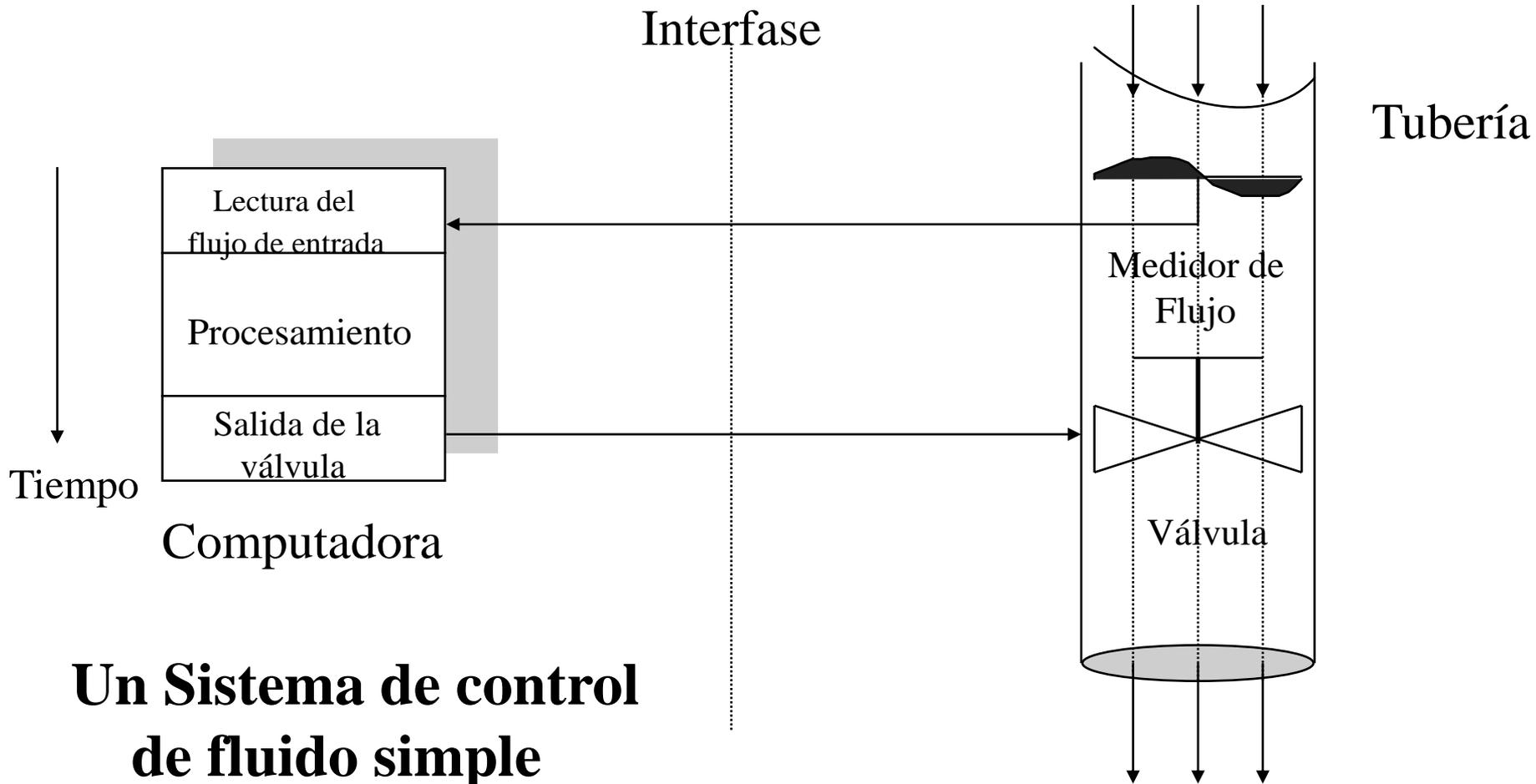


# *Aplicaciones de los Sistema de Tiempo Real*

---

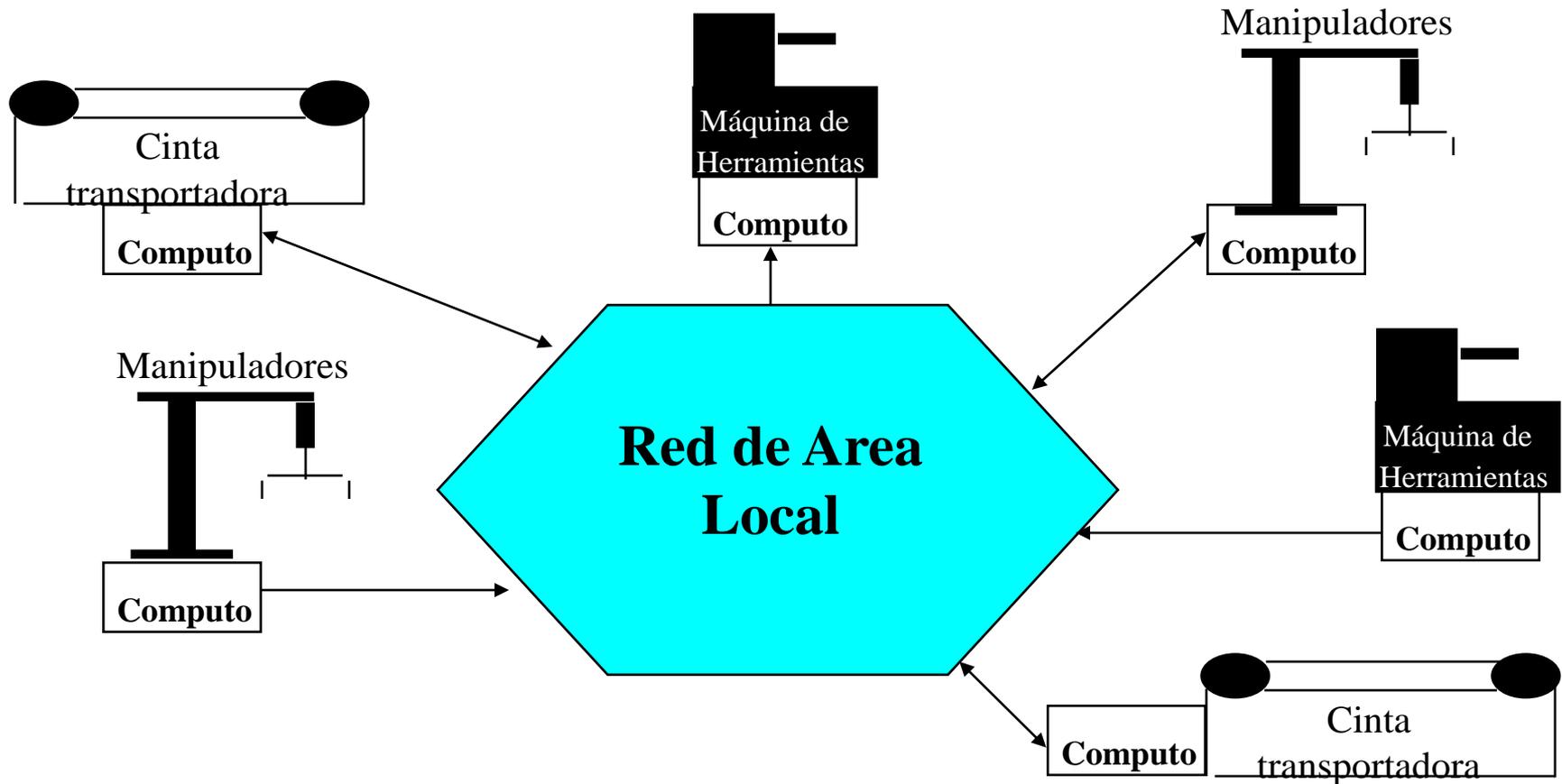
- ◆ **Dominio Industrial**
  - ◆ Controlador de la planta
  - ◆ Robot para tratamiento de material peligroso
- ◆ **Uso militar**
  - ◆ Sistema de reconocimiento de blancos automático
  - ◆ Sistema de guiado de misiles y navegación
- ◆ **Sistemas altamente críticos**
  - ◆ Plantas nucleares
  - ◆ Sistemas de aviónica
- ◆ **Sistemas de Telecomunicaciones**
  - ◆ Sistemas Multimedia
  - ◆ Sistemas de Telefonía Mobil

# Aplicaciones



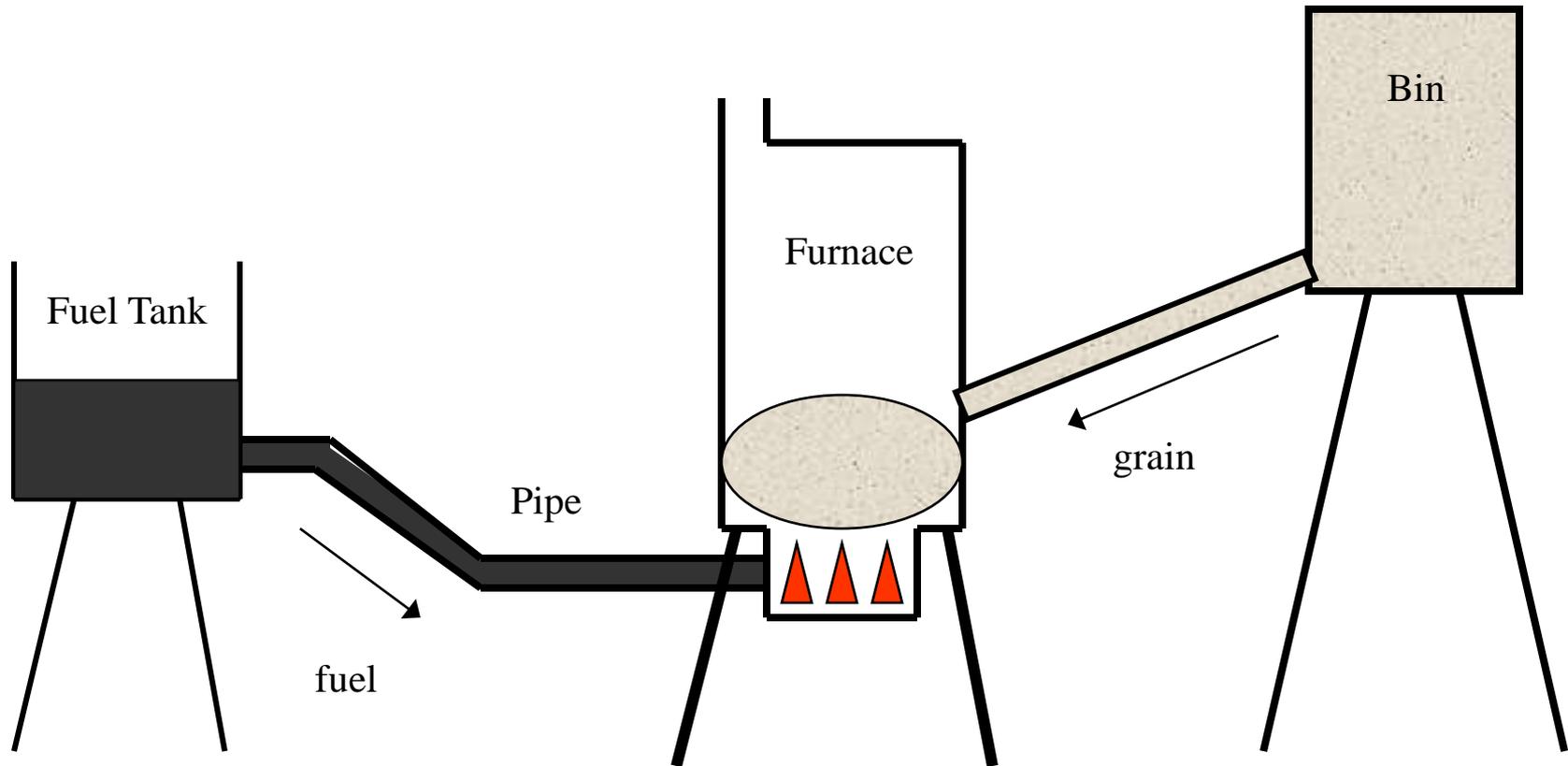
# Sistema de computo embebido de control de produccion

---

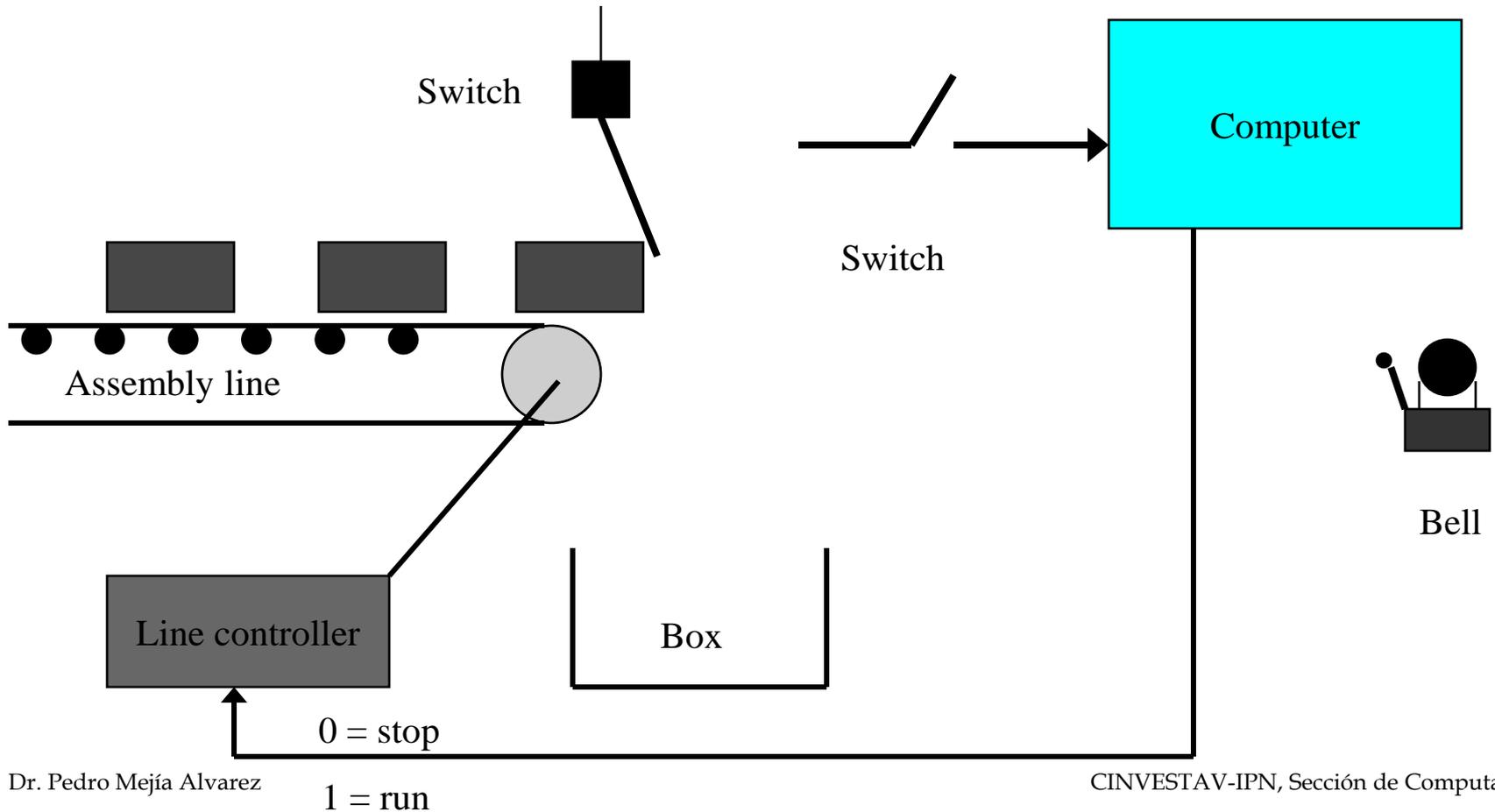


# *Aplicaciones: Control de temperatura en un Tanque*

---



# Aplicaciones: Control de Manufactura



# Sistemas Embebidos

- ◆ Sistemas de computo basados en microprocesadores
  - ◆ Funciones especificas, pre-definidas
  - ◆ Recursos limitados (memoria, potencia)
  - ◆ La aplicacion se ejecuta desde ROM
- ◆ Existen...
  - ◆ Los Sistemas Embebidos estan en el 90% de los dispositivos electronicos y de computo en todo el mundo
  - ◆ Existen mas sistemas embebidos que computadoras (>10).



Alvarez



# Ejemplos

<b>Sistema de oficina y equipo móviles</b>	<b>Sistemas en Edificios</b>	<b>Fabricación y control de procesos</b>
<p>Maquinas contestadoras  Copiadoras  Faxes  Laptops y notebooks  Telefonos móviles  PDAs, Organizadores personales  Still and video cameras  Sistemas de Telefonía  Grabación de tiempo  Impresoras  Hornos de Microondas</p>	<p>Aire acondicionado  Respaldo y generadores de luz  Administración de edificios  Televisión por cable  Controladores de Fuego  Calefacción y Ventilación  Elevadores, Escaleras  Iluminación  Seguridad  Cámaras de Seguridad  Extintores</p>	<p>Fábricas automatizadas  Plantas embotelladoras  Sist. Control de energía  Plantas de manufactura  Estaciones nucleares  Refinerías de Petróleo  Sistemas de Energía  Estaciones de Poder  Robots  Sistemas de switching  Sist. Agua y alcantarillado</p>

# Ejemplos

<b>Transporte</b>	<b>Comunicaciones</b>	<b>Otros equipos</b>
Aereoplanos Trenes Autobuses Barcos Muelles Automobiles Cntr.de Trafico aereo Sist. de senalizacion Radars Luces de Trafico Maquinas de boletos Camaras de velocidad Detects.de velocidad	Telefonia Cable Switches de telefonos Satelites Posicionamiento Global	Maquinas contestadoras Sist. Tarjetas de credito Equipos de Imagenes Medicas Ctrl. De calefaccion domestica VCRs

# *Características de un sistema embebido*

---

## **Operacion en Real-Time**

- Reactivo: Calculos que deben ocurrir en respuesta a eventos externos
- Exactitud es parcialmente una funcion del tiempo

## **Pequeño y de bajo peso**

- Dispositivos manuales y aplicaciones de trasportacion

## **Bajo consumo de energia**

- Bateria con duracion de 8 horas (laptops comunmente menso de 2 horas)

## **Ambientes asperos**

- Calor, vibracion, golpes, fluctuaciones de corriente, interferencia, luz, corrosion

## **Operación de seguridad critica**

- Deben funcionar correctamente y no deben funcionar correctamente

## **Alto costo**

# Anatomía de un Proyecto Embebido

1 Requerimientos de Aplicación

2 Procesador y Arquitectura



3 SO-TR y Arq. de Soft.

pSOS+

VxWorks

Neutrino

lynxOS

nucleus

4 Herramientas de Desarrollo - (compilador, depurador, simulador)



MetroWerks



# *Dificultades en el Diseño*

---

- ◆ Diseño de la aplicación en tiempo real.
- ◆ Control de la concurrencia de procesos.
- ◆ Selección de la arquitectura de hardware que mejor responda a la aplicación.
- ◆ Obtención de tiempos: *Caracterización*
  - ◆ especificar los tiempos a los que las acciones deben llevarse a cabo.
  - ◆ especificar en cuanto tiempo debe completarse cada acción.
  - ◆ responder a situaciones en las cuales no todos los tiempos se cumplen
  - ◆ responder a situaciones en las que los requisitos de tiempos cambian dinámicamente.

# *Consideraciones de Diseño*

---

A quien afecta la introducción de Tiempo Real ?.

- ◆ A los lenguajes, compiladores.
- ◆ Al sistema operativo.
- ◆ A la arquitectura de hardware.
- ◆ A la metodología de diseño.
- ◆ A la arquitectura del sistema (HW/SW/Ambiente).

# *Programacion de Sistemas de Tiempo Real.*

---

Actividades en una computadora de automovil.

Control de  
Velocidad

**C=4ms.**  
**T=20ms.**  
**D=5ms.**

Control de  
Frenado

**C=10ms.**  
**T=40ms.**  
**D=40ms.**

Control de  
combustible

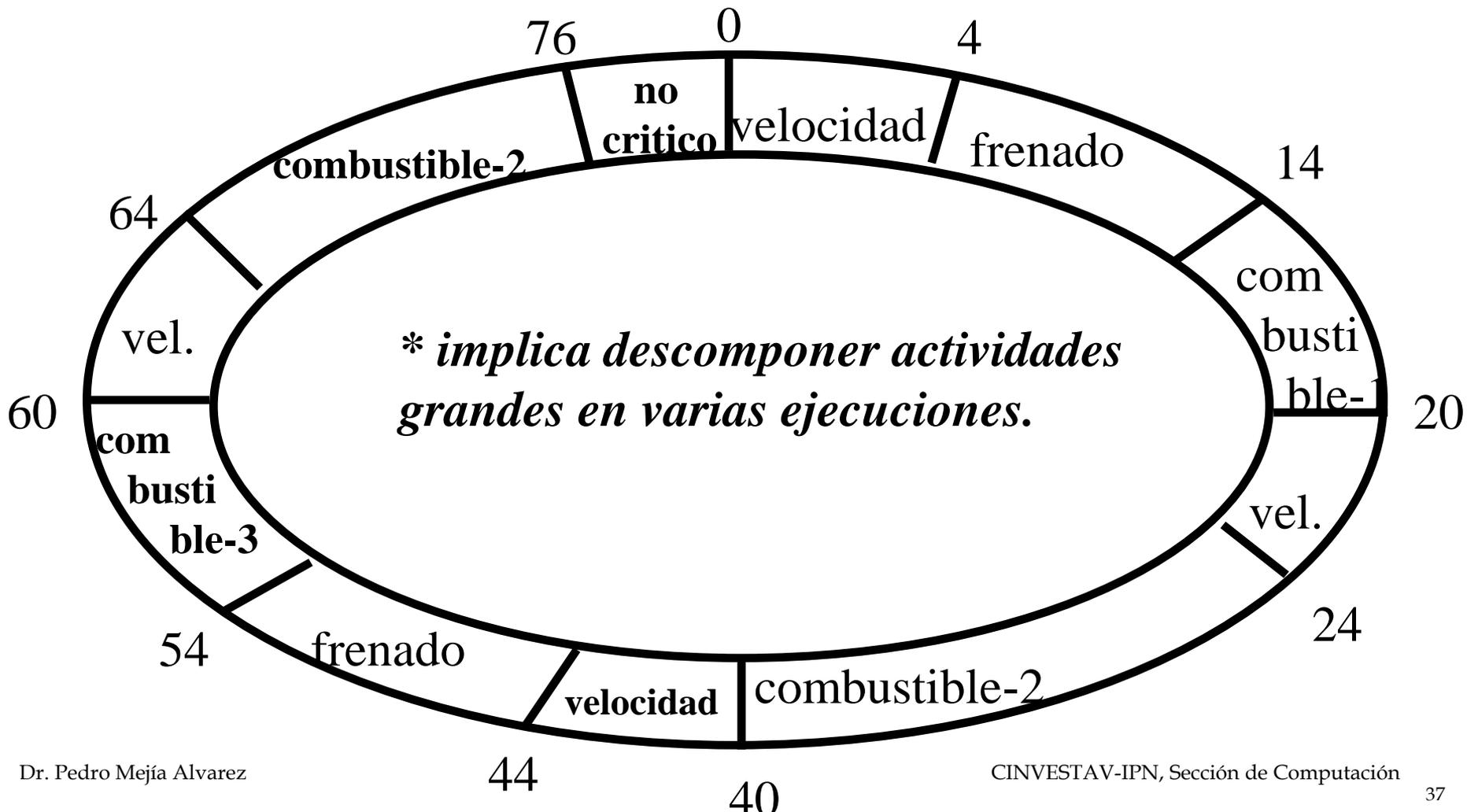
**C=40ms.**  
**T=80ms.**  
**D=80ms.**

Otro software  
no-critico

**C=10ms.**  
**T=40ms.**

**C=tiempo de computo (peor caso), T=Periodo de ejecucion, D=Plazo de respuesta**

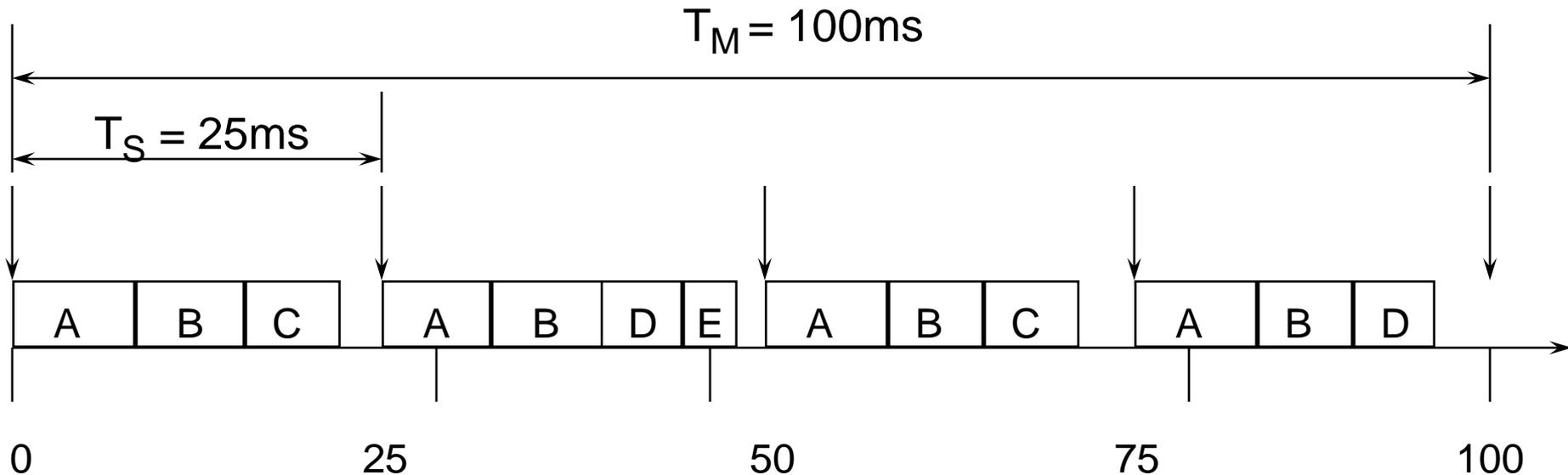
# Solucion Ciclica.



# Solucion Ciclica: Ejemplo

Tarea	T	C
A	25	10
B	25	8
C	50	5
D	50	4
E	100	2

- El ciclo principal dura 100 ms
- Se compone de 4 ciclos secundarios de 25 ms cada uno



# *Programación del Ejecutivo cíclico*

---

```
procedure Cyclic_Executive is  
  type Cycle is mod 4;  
  Frame ; Cycle := 0;  
begin  
  loop  
    Wait_for_Interrupt;  
    case Frame is  
      when 0 => A; B; C;  
      when 1 => A; B; D; E;  
      when 2 => A; B; C;  
      when 3 => A; B; D;  
    end case;  
    Frame := Frame + 1;  
  end loop;  
end Cyclic_Executive;
```

# Solucion Concurrente.

---

*La solucion concurrente es mas simple de disenar y modificar.*

```
task body speed is
begin
loop
  speed_measurement;
  next:=next+0.2
  sleep_until_next;
end loop
end speed
```

```
task body brake is
begin
loop
  control_brakes;
  next:=next+0.2
  sleep_until_next;
end loop
end speed
```

```
task body fuel is
begin
loop
  fuel_injection;
  next:=next+0.2
  sleep_until_next;
end loop
end speed
```

```
task body non_critical is
begin
loop
  perform_computation;
end loop
end speed
```

# *Lenguajes de Programacion de Sistemas de Tiempo Real*

---

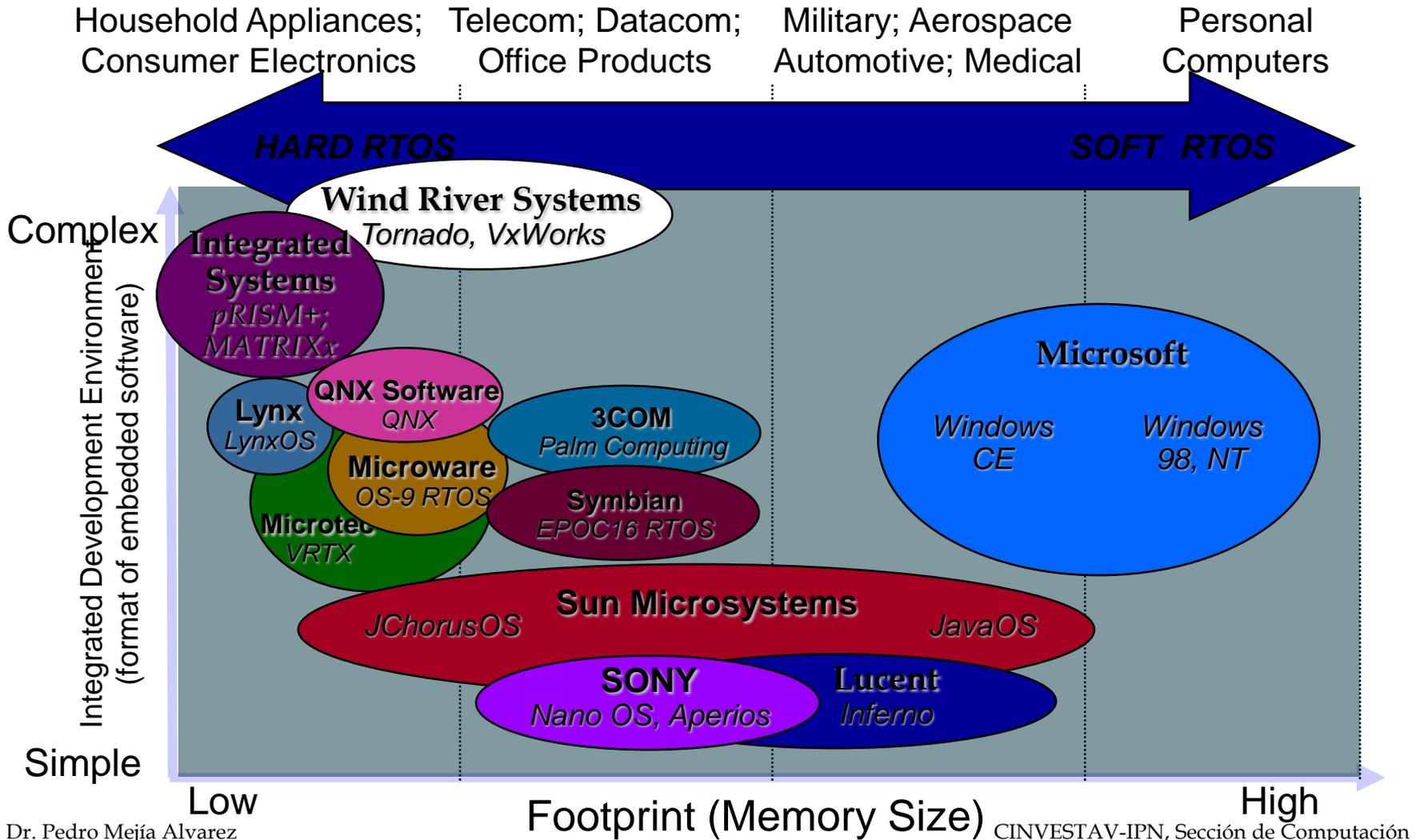
- ◆ ADA
- ◆ Modula-2
- ◆ Java-RT
- ◆ Ocamm.
- ◆ C
- ◆ C++

# *Ambientes y Herramientas de Diseño de Sistemas de Tiempo Real*

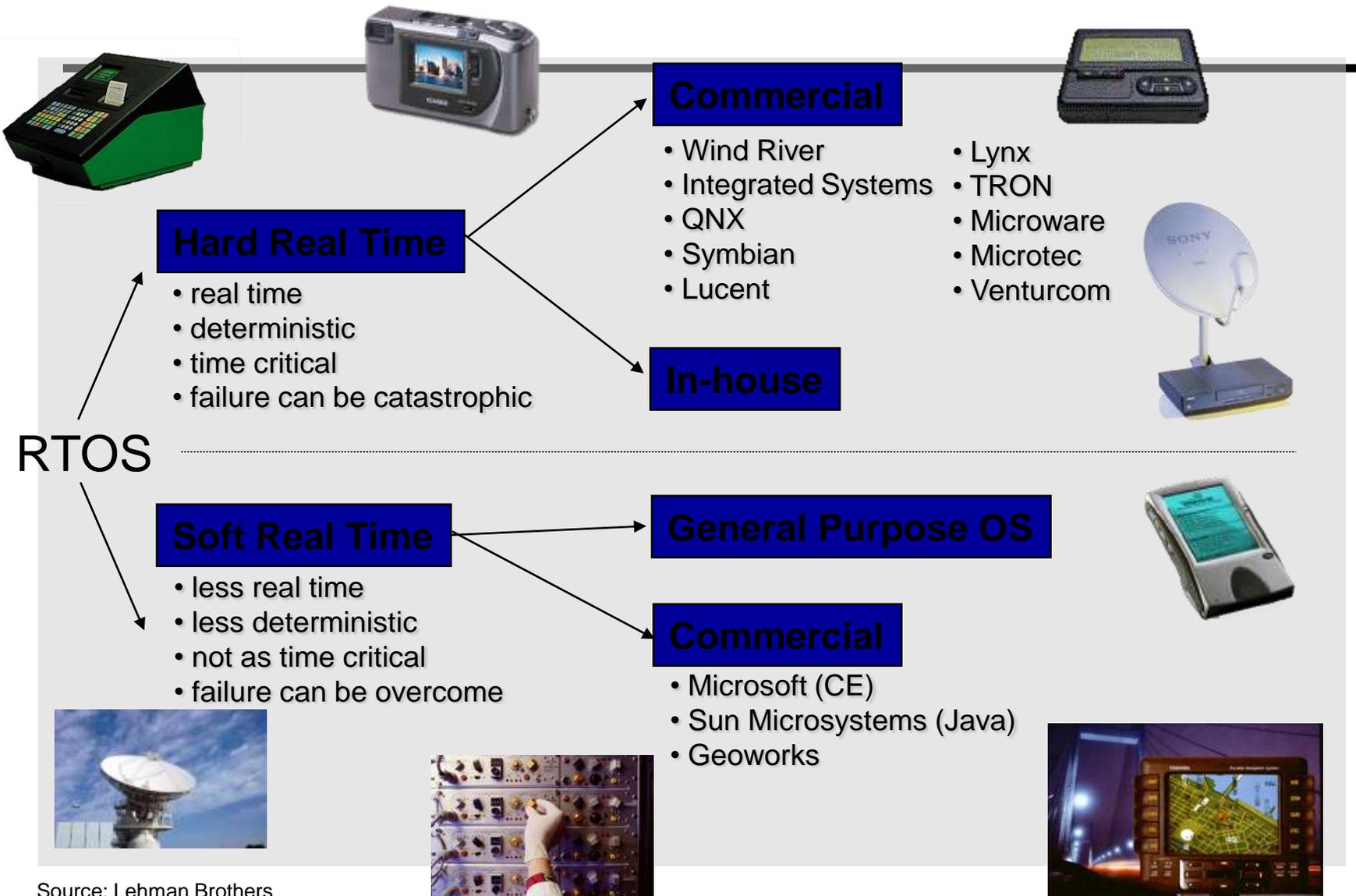
---

- ◆ Tornado
- ◆ UML-RT
- ◆ Rational Rose - RT
- ◆ Object Time Tool Set
- ◆ PERTS (TriPac)
- ◆ TimeWiz (TimeSys)
- ◆ QNX

# Sistemas Operativos de Tiempo Real



# SO's: Hard Real-Time vs. Soft Real-Time

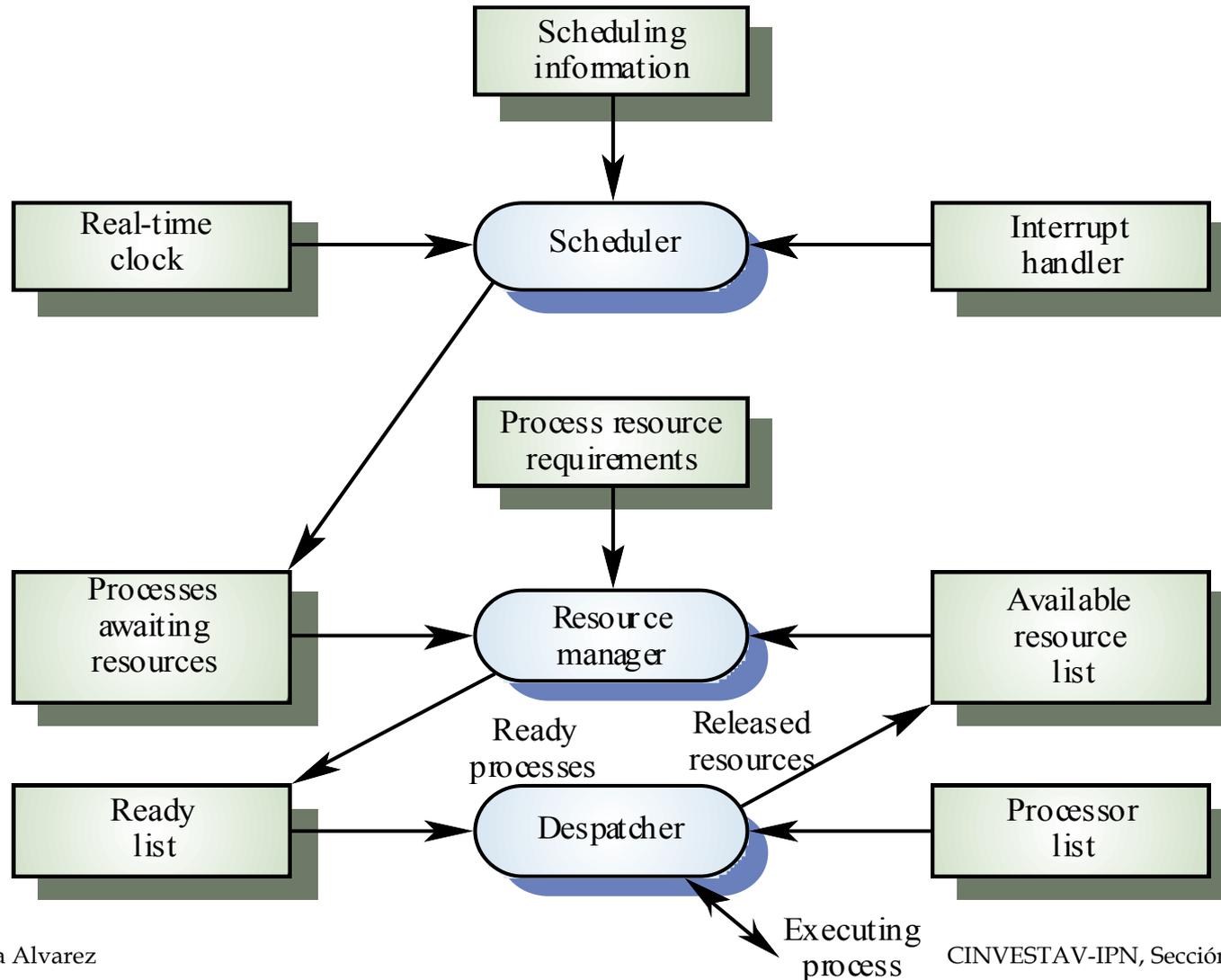


# *Sistemas Operativos de Tiempo Real*

---

- ◆ Estructura del sistema operativo.
- ◆ Modelos de tareas, estados de las tareas, servicios y transacciones.
- ◆ Procesos y threads. Cambio de contexto.
- ◆ Algoritmos de Planificación: Ciclica, RMS, EDF.
- ◆ Inversión de prioridad y protocolo de techo de prioridad.
- ◆ Tareas aperiodicas y Servidor esporadico.
- ◆ Sincronización de procesos.
- ◆ Comunicación entre procesos.
- ◆ Relojes y timers.
- ◆ Memoria compartida, locking, reserva.
- ◆ E/S sincrónica y asincrónica.
- ◆ Transacciones en tiempo real y manejo de archivos.
- ◆ Manejo de interrupciones y device drivers.

# Componentes del Ejecutivo de Tiempo Real



# Componentes Principales

---

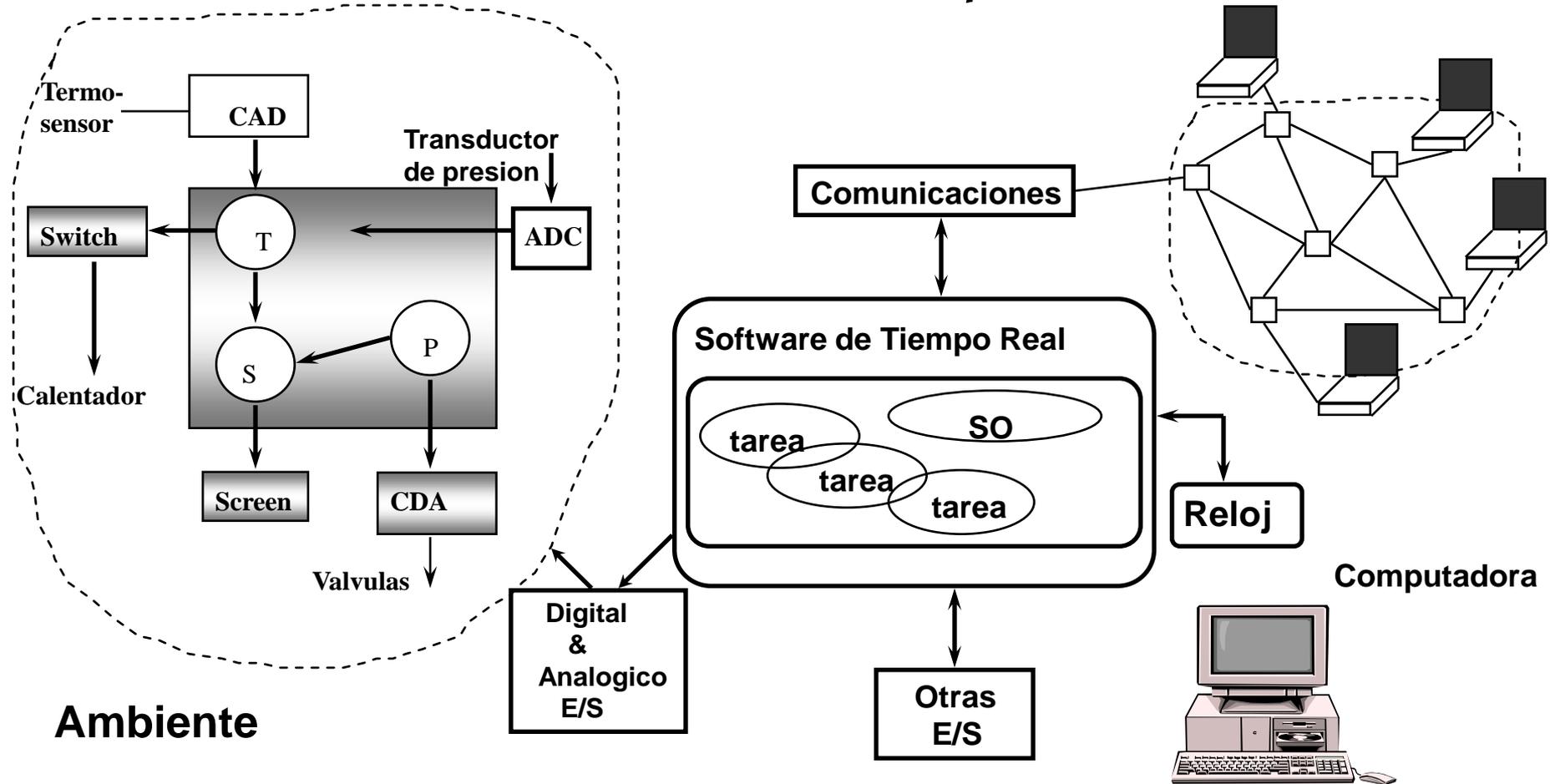
- Reloj de tiempo real
  - Provee información para el calendario de procesos (process scheduling).
- Manejador de interrupciones
  - Controla el servicio a los requerimientos instantáneos.
- Scheduler
  - Selecciona el próximo proceso a correr.
- Administrador de recursos
  - Asigna recursos de memoria ,dispositivos de E/S, Red.
- Despachador
  - Inicia la ejecución de procesos.

# *Características del SO-RT*

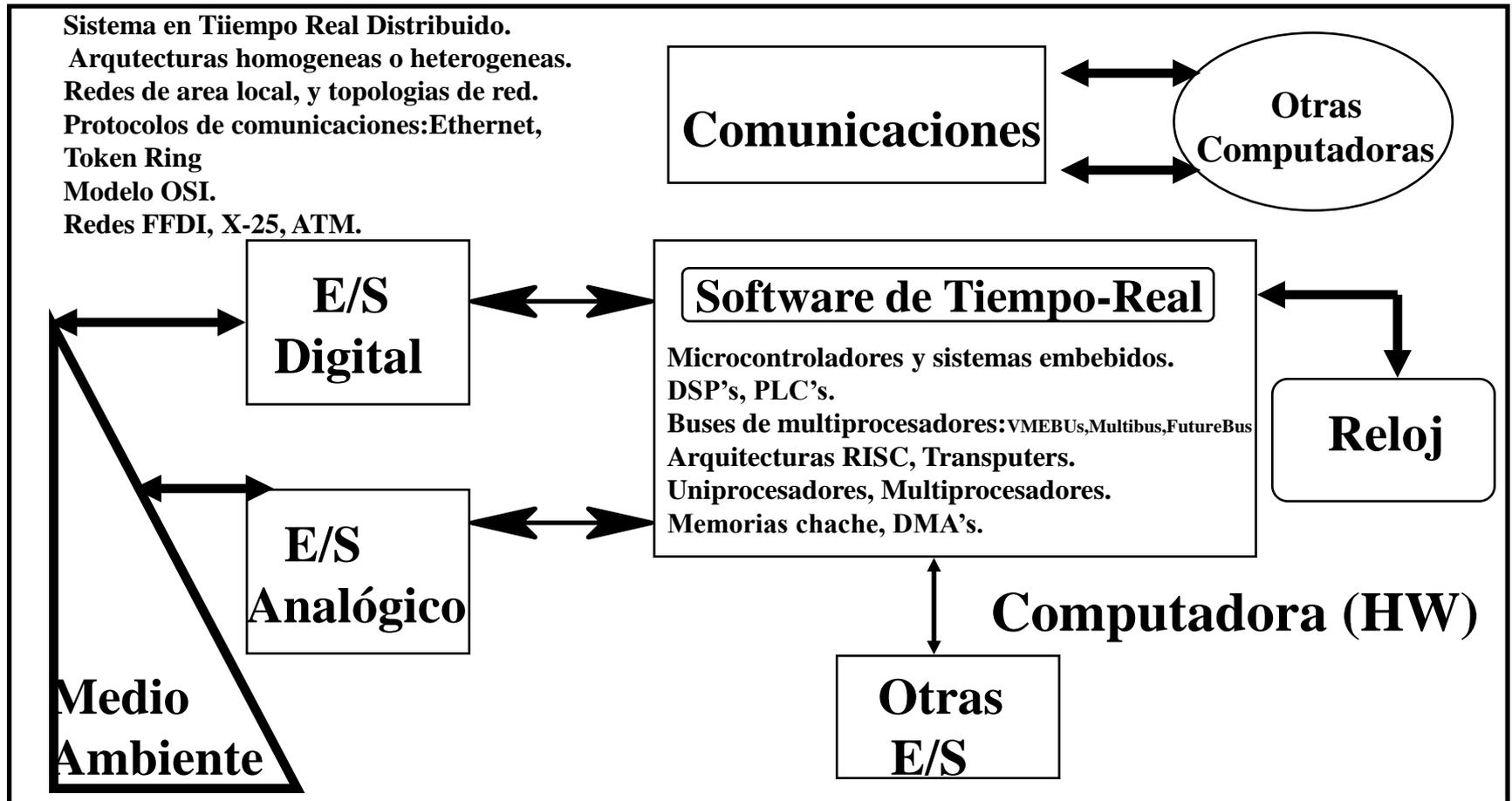
---

- ◆ Velocidad y Eficiencia
- ◆ Bajo Tamano.
- ◆ Ejecutable desde ROM.
- ◆ Predecibilidad: Pocos accesos a disco, DMA, cache, etc.
- ◆ Control completo del tiempo y de los recursos.
- ◆ Configurable para aplicaciones específicas.
- ◆ Manejo de diversos dispositivos de E/S.
- ◆ Tolerante a Fallas

# Arquitectura de un Sistema de Tiempo Real



# Arquitecturas de Tiempo Real



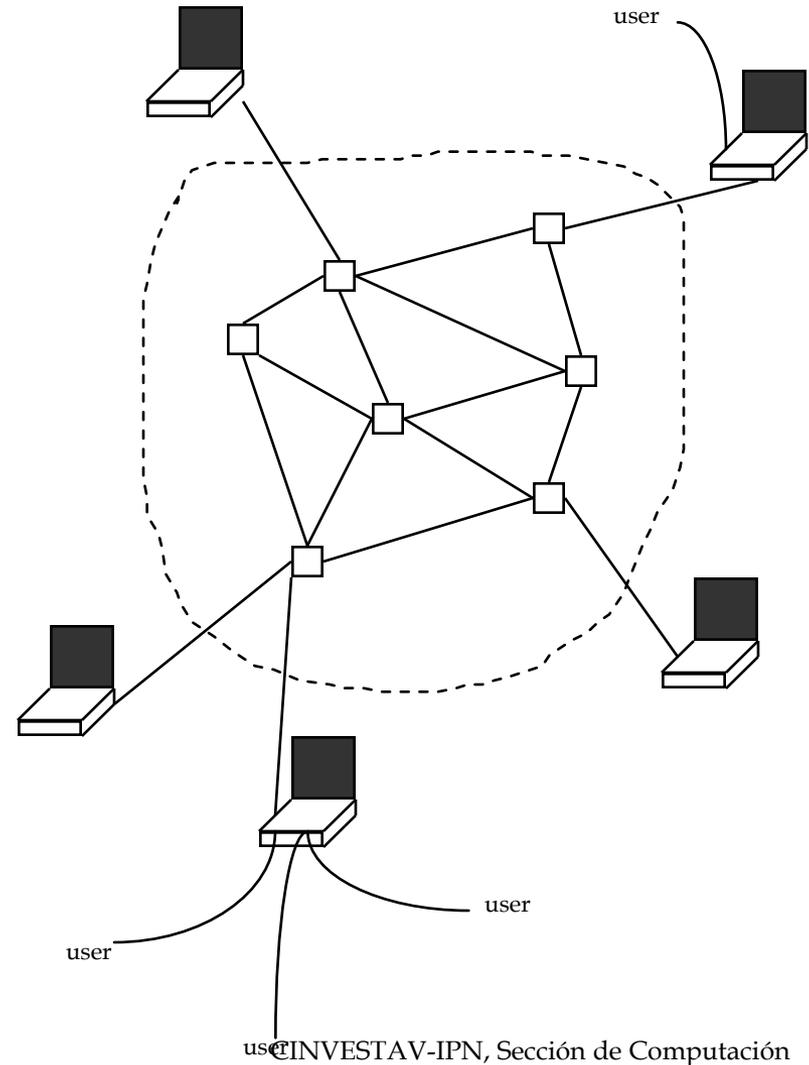
# *Bases de Datos en Tiempo Real*

---

- ◆ Uno de los principales problemas con las bases de datos es su naturaleza impredecible sobre sus tiempos de respuesta.
- ◆ Esta impredecibilidad es debida a:
  - ◆ la naturaleza de los queries.
  - ◆ accesos a disco.
  - ◆ contension de recursos.
  - ◆ la inability de conocer cuantos objetos debe acceder un query y cuanto tiempo le tomara.
- ◆ Restricciones importantes:
  - ◆ planificacion de transacciones que cumplan plazos de respuesta.
  - ◆ semantica explicita que permita especificar manejo de transacciones con tiempos.
  - ◆ chequeo del sistema de base de datos de cumplir con requerimientos de tiempos en sus transacciones.

# Comunicaciones en Tiempo Real:

- ◆ A conjunto de mensajes generados desde un nodo fuente hacia un nodo destino y conectados a través de una red de topología single/multihop
- ◆ Los mensajes deben de cumplir las restricciones de TR:end-to-end-scheduling
  - ◆ **Deadline, Periodicidad, Modo, Guarantee type, TR Hard/Soft,...**



# *Problemas específicos de un Sistema Distribuido de Tiempo Real*

---

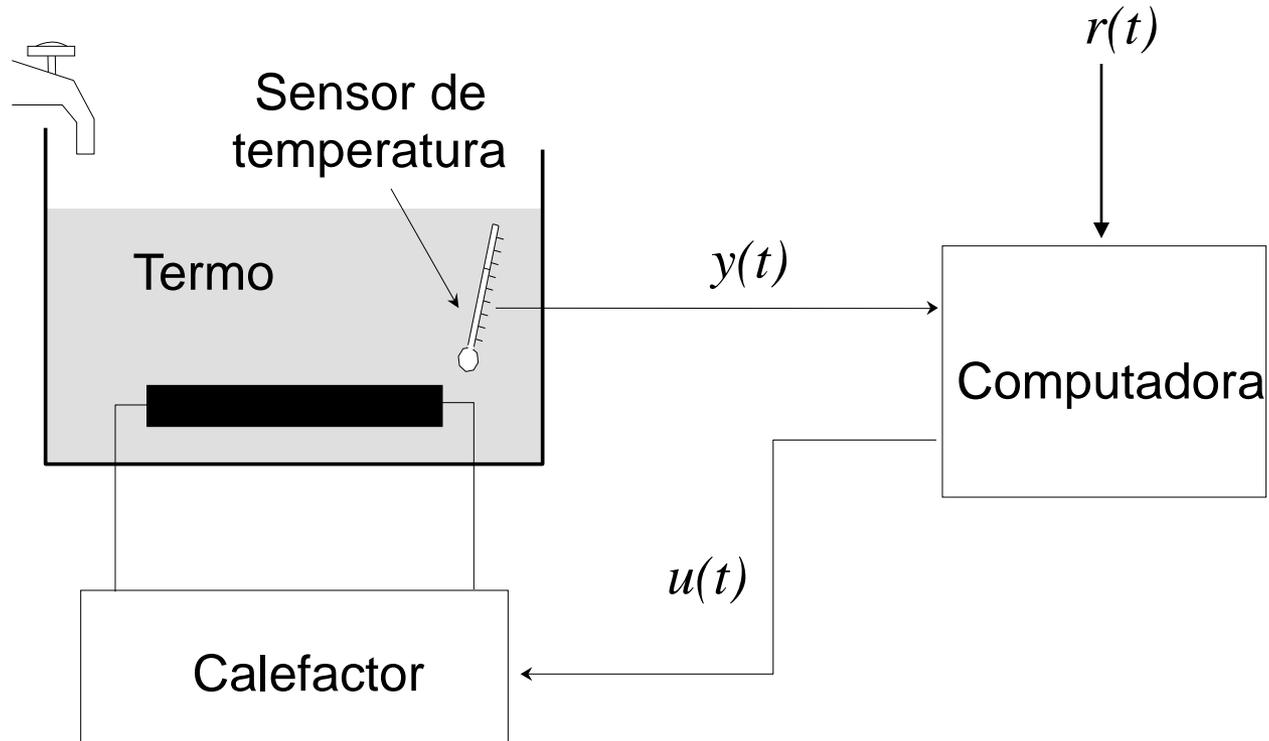
- ◆ Planificación del medio de comunicación.
- ◆ Asignación de tareas a procesadores.
- ◆ Plazos de respuesta globales (end-to-end).
- ◆ Sincronización de relojes.
- ◆ Tolerancia a fallos.

# *Sistemas de Control de Procesos en Tiempo Real*

---

- ◆ Teoria de Control:
  - ◆ Proporcionar garantías de estabilidad y robustes en el sistema.
- ◆ Teoria de Planificacion:
  - ◆ Minimiza Perdida de plazos de respuesta.
  - ◆ Maximizar la calidad del servicio.

# Caracterización de la Aplicación



- **Componentes:** tanque, sensores, actuadores, interfaces A/D y computadora.

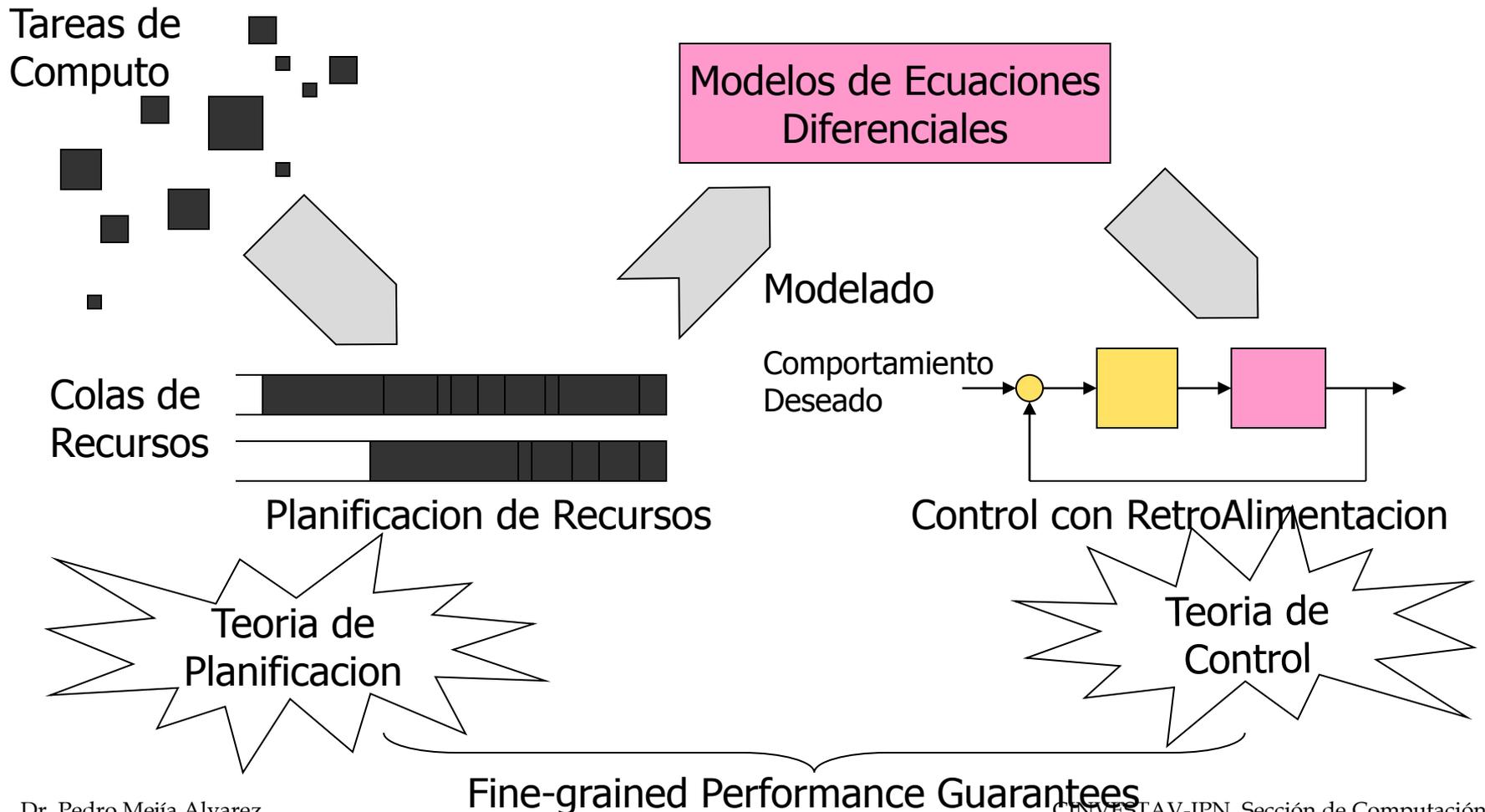
# *Caracterización de la Aplicación*

---

## **Restricciones de Funcionamiento.**

- ◆ Componentes: tanque, sensores, actuadores, interfaces A/D y computadora.
- ◆ En el sistema (planta), dada una  $u(t)$ , se produce una  $y(t)$ .
- ◆ El sistema recibe una señal de referencia  $r(t)$ , el cual indica la temperatura deseada.
- ◆ El objetivo del sistema de control es que  $y(t)$ , sea lo mas parecida a  $r(t)$ .
- ◆ Para lograr el objetivo, es preciso aplicar a la planta la señal  $u(t)$
- ◆  $y(t)$  muestra el valor real de la temperatura del termo.

# *Elementos Teoricos de un Sistema de Control de Tiempo Real*



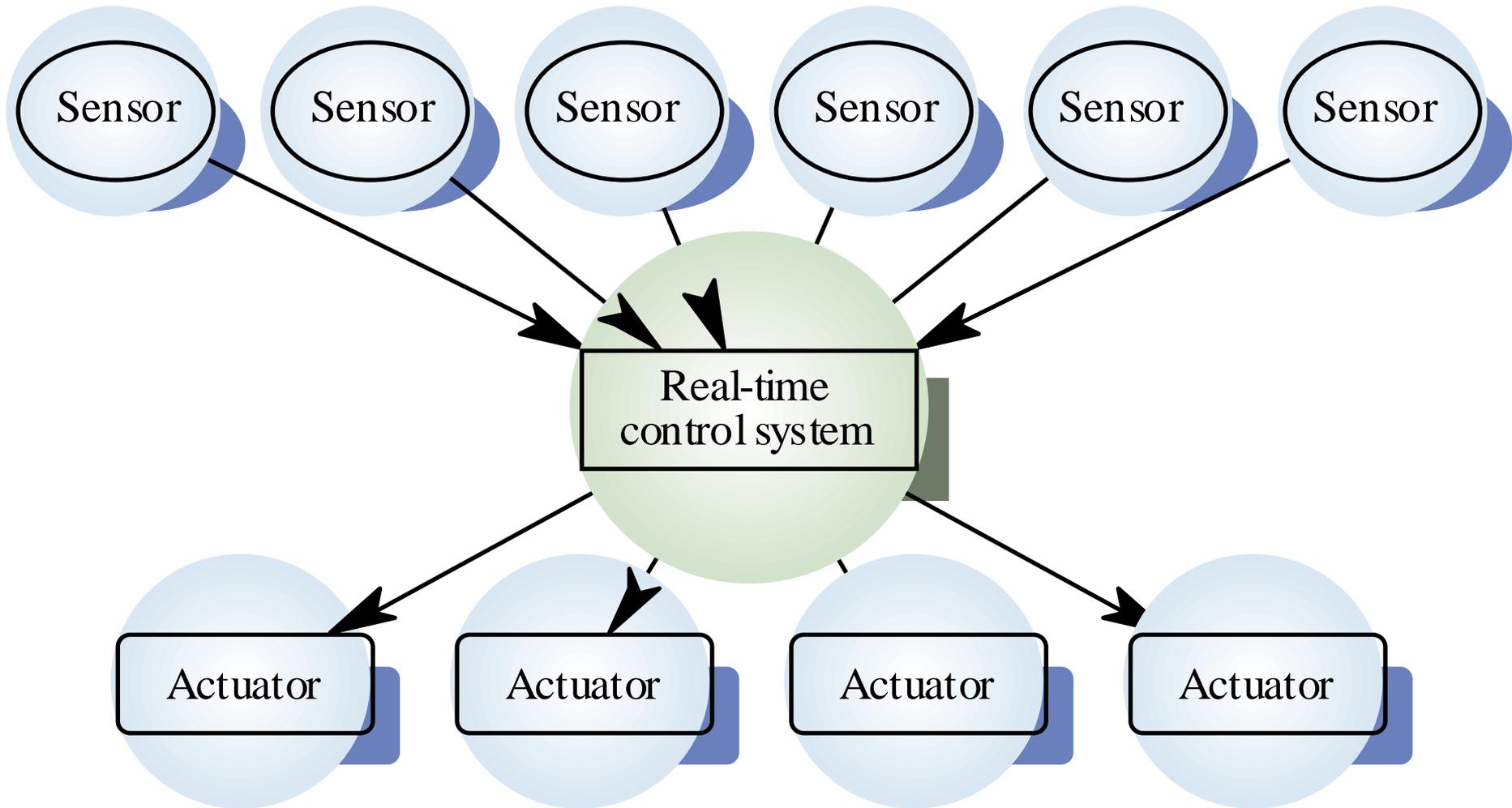
# *Proceso de Diseño de Sistemas de Tiempo Real*

---

- ◆ **Identificar** la estimulación a ser procesada y las respuestas requeridas de esa estimulación
- ◆ Por cada estímulo una respuesta, **identificar** los límites de tiempo
- ◆ **Unir** el procesamiento de estímulos/respuestas en procesos concurrentes. Un proceso debe estar asociado con cada tipo de estímulo y respuesta
- ◆ **Diseñar** algoritmos para procesar cada tipo de estímulo y respuesta. Estos deben satisfacer los requerimientos de tiempo involucrados
- ◆ **Diseñar** un planificador del sistema, en el cual se mostrará la secuencia de los procesos sus tiempos y sus plazos

# *Un Modelo de Sistema de Tiempo Real*

---



# *Sistemas de Estímulo/Respuesta*

---

- ◆ Dando un estímulo, el sistema debe producir una respuesta dentro de un tiempo específico
- ◆ Estimulación periódica. Es la que se da en intervalos de tiempo predecibles
  - ◆ *Por ejemplo, un sensor de temperatura debe estar midiendo 10 veces por segundo*
- ◆ Estimulación no periódica. Es la cual ocurre en instantes impredecibles
  - ◆ *Por ejemplo, una falla de energía en el sistema debe provocar una interrupción la cual debe ser procesada por el sistema*

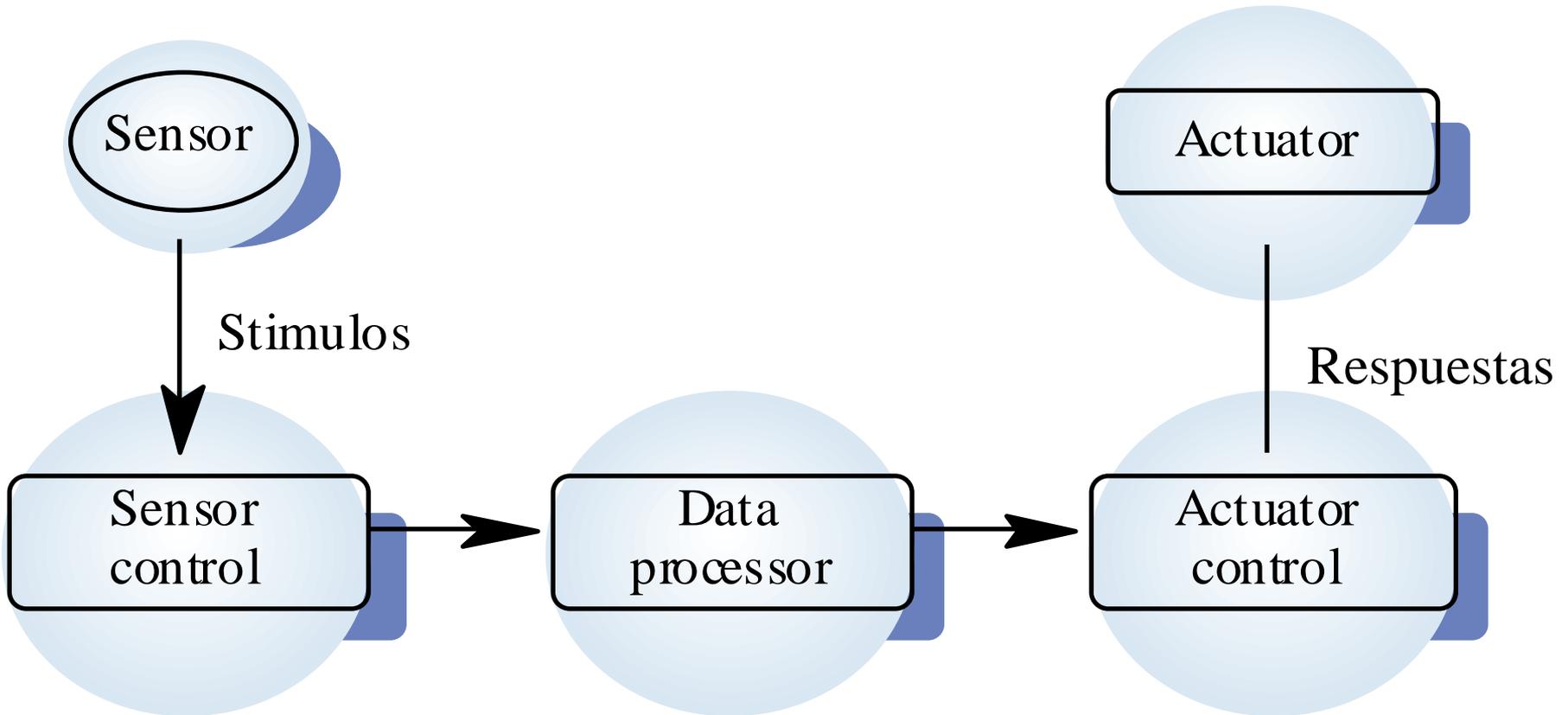
# *Elementos del Sistema*

---

- ◆ Procesos de Control de Sensores
  - ◆ Recolectan información de los sensores. Deben almacenarla temporalmente y mandarla como respuesta a un sensor de estímulos
- ◆ Procesador de Datos
  - ◆ Lleva a cabo el procesamiento de la información recolectada y computa la respuesta del sistema
- ◆ Control del Activador
  - ◆ Produce señales de control para el activador

# *Procesos del Sensor/Activador*

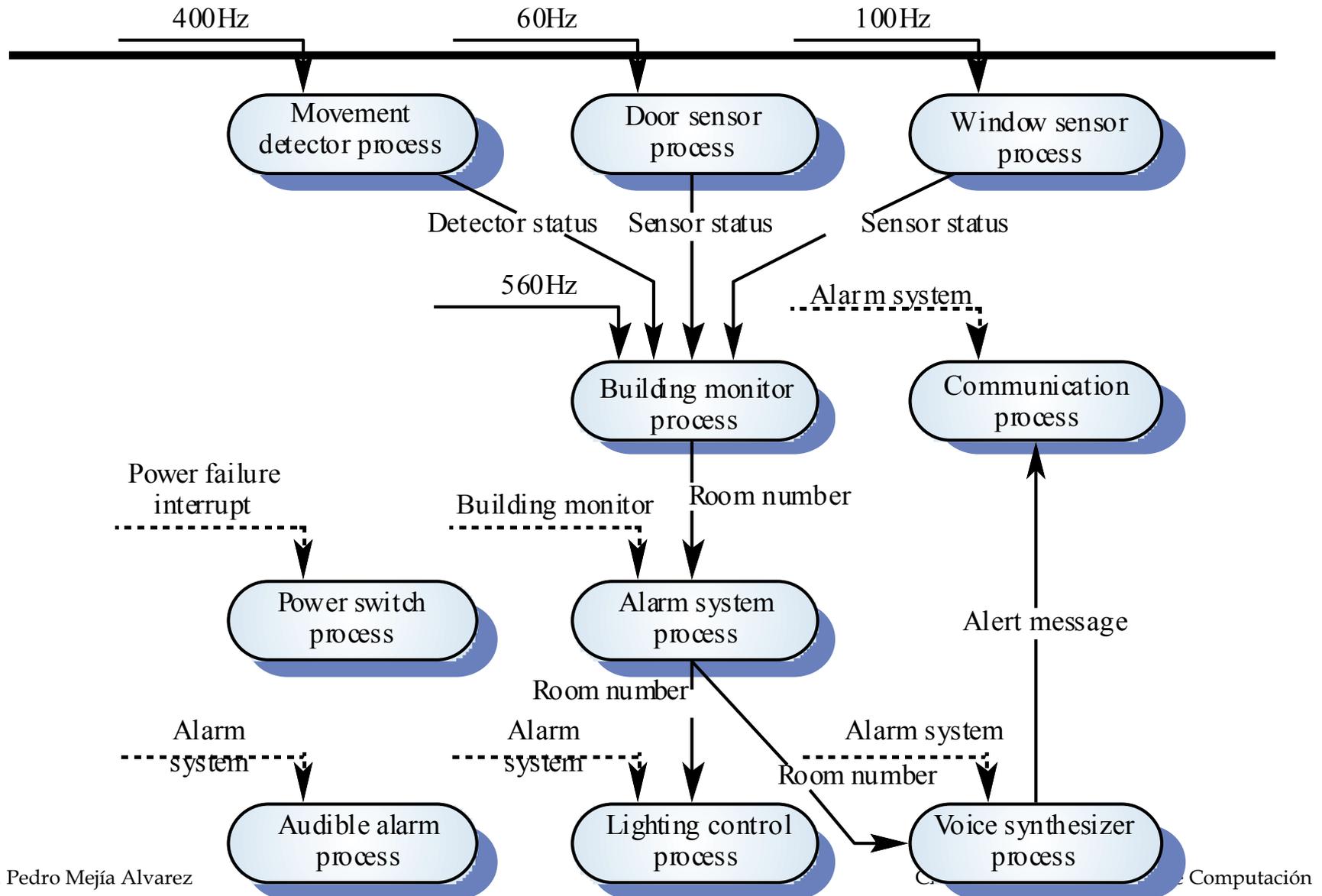
---



# Requerimientos de tiempo

<b>Estímulo/Respuesta</b>	<b>Requerimientos de tiempo</b>
Interrupción por falla de energía	El encendido de la energía de emergencia debe estar listo en un tope de 50 ms.
Alarma de puerta	Cada alarma de puerta debe activarse dos veces por segundo.
Alarma de ventana	Cada alarma de ventana debe activarse dos veces por segundo.
Detector de movimiento	Cada detector de movimiento debe activarse dos por segundo.
Alarma sonora	La alarma sonora debe activarse dentro de 1/2 segundo de que una alarma se active por un sensor
Encendido de luces	Las luces deben encenderse dentro de 1/2 segundo de que una alarma se active por un sensor.
Comunicaciones	La llamada a la policía se activa dentro de 2 seg. de la activación de una alarma por un sensor.
Sintetizador de voz	Un mensaje sintetizado debe activarse dentro de 4 seg. de que una alarma sea activada por un sensor

# Arquitectura del Proceso



# *Tareas de monitoreo del edificio*

---

```
task Building_monitor is  
  entry Initialise ;  
  entry Test ;  
  entry Monitor ;  
end Building_monitor ;
```

```
task body Building_monitor is  
  type ROOMS is array (NATURAL range <>) of ROOM_NUMBER ;  
  Move_sensor, Window_sensor, Door_sensor : SENSOR ;  
  Move_sensor_locations: ROOMS (0..Number_of_move_sensors-1) ;  
  Window_sensor_locations: ROOMS (0.. Number_of_window_sensors -1) ;  
  Corridor_sensor_locations : ROOMS (0..Number_of_corridor_sensors-1) ;  
  Next_movement_sensor, Next_window_sensor,  
  Next_door_sensor: NATURAL := 0;  
begin  
  select  
  accept Initialise do  
    -- code here to read sensor locations from a file and  
    -- initialise all location arrays  
  end Initialise ;  
  or  
  accept Test do  
    -- code here to activate a sensor test routine  
  end Test ;  
  or  
  accept Monitor do  
    -- the main processing loop
```

# *Tareas de monitoreo del edificio*

---

```
accept Monitor do
  -- the main processing loop
  loop
    -- TIMING: Each movement sensor twice/second
    Next_move_sensor :=
      Next_move_sensor + 1 rem Number_of_move_sensors ;
    -- rendezvous with Movement detector process
    Movement_detector.Interrogate (Move_sensor) ;
    if Move_sensor /= OK then
      Alarm_system.Initiate
        (Move_sensor_locations (Next_move_sensor)) ;
    end if ;
    -- TIMING: Each window sensor twice/second
    -- rendezvous with Window sensor process
    Next_window_sensor :=
      Next_window_sensor + 1 rem Number_of_window_sensors ;
    Window_sensor.Interrogate (Window_sensor) ;
    if Window_sensor /= OK then
      Alarm_system.Initiate (Window_sensor_locations
        (Next_move_sensor)) ;
    end if ;
    -- TIMING: Each door sensor twice/second
    -- rendezvous with Door sensor process
    -- Comparable code to the above here
  end loop ;
end select ;
end Building_monitor ;
```

# *Trabajo en Desarrollo*

---

- ◆ Control de Procesos en Tiempo Real.
- ◆ Planificabilidad.
- ◆ Metodos de Diseno de Sistemas de Tiempo Real.
- ◆ Ahorro de Energia en Sistemas de Computo Portatil de TR
- ◆ Tolerancia a Fallas en Sistemas de Tiempo Real.
- ◆ Sistemas Multimedia.
- ◆ Sistemas Operativos de Tiempo Real Embebidos para Sistemas de Computo Mobil

Seccion de Computacion, CINVESTAV-IPN.

[pmejia@cs.cinvestav.mx](mailto:pmejia@cs.cinvestav.mx)

<http://delta.cs.cinvestav.mx/~pmejia>

# *Tolerancia a Fallos en STR*

---

## **Objetivo: Integrar en los Sistemas de Tiempo Real la atención a Fallos**

- ◆ Correcto funcionamiento                      FUNCIONALIDAD
- ◆ Cumplimiento de plazos críticos            TIEMPO REAL
- ◆ Confiabilidad                                    TOLERANCIA A FALLOS

## **Características:**

- ◆ Listo siempre que se le necesita            DISPONIBLE
- ◆ Proporciona una operación estable        FIABLE
- ◆ Protege de situaciones catastróficas      SEGURO