

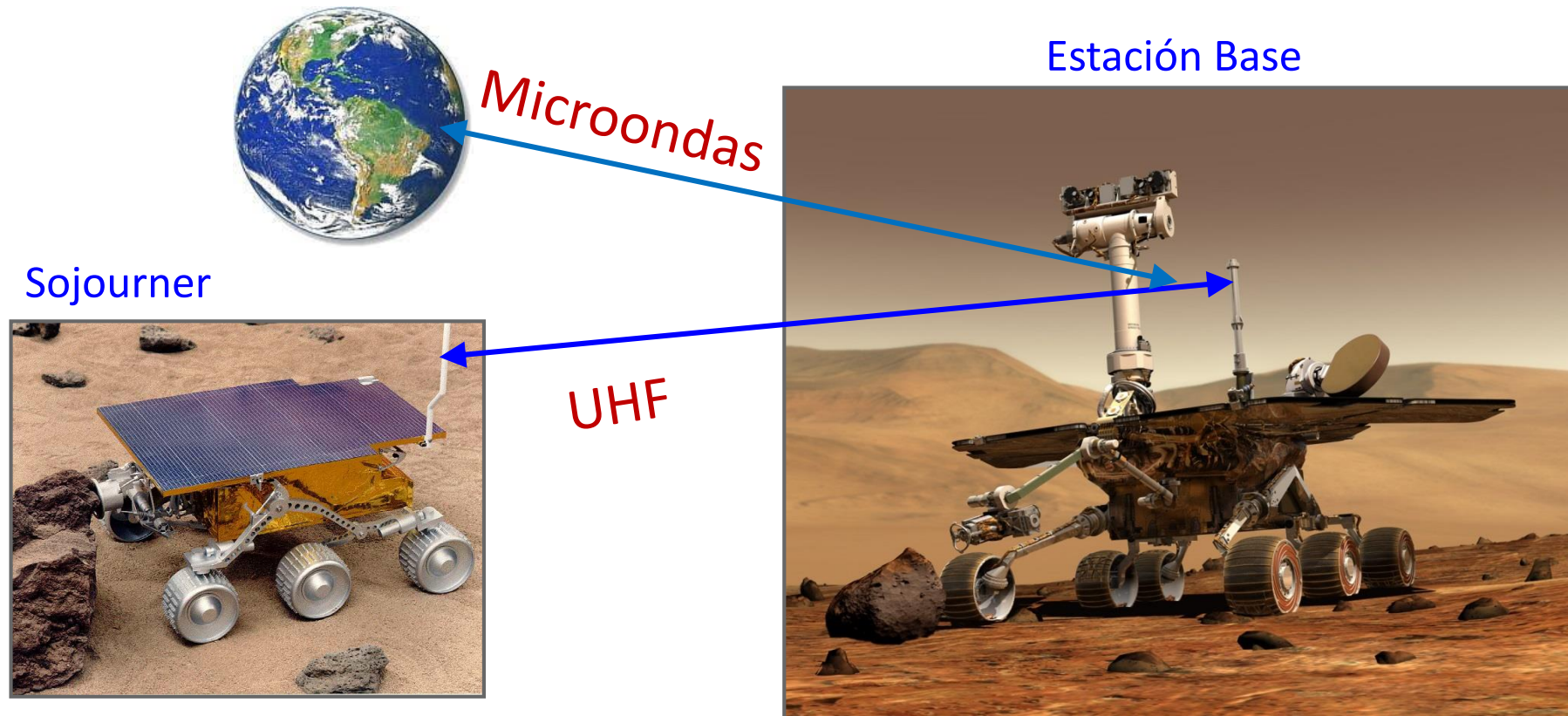
Programación de Aplicaciones Real Time (RT)



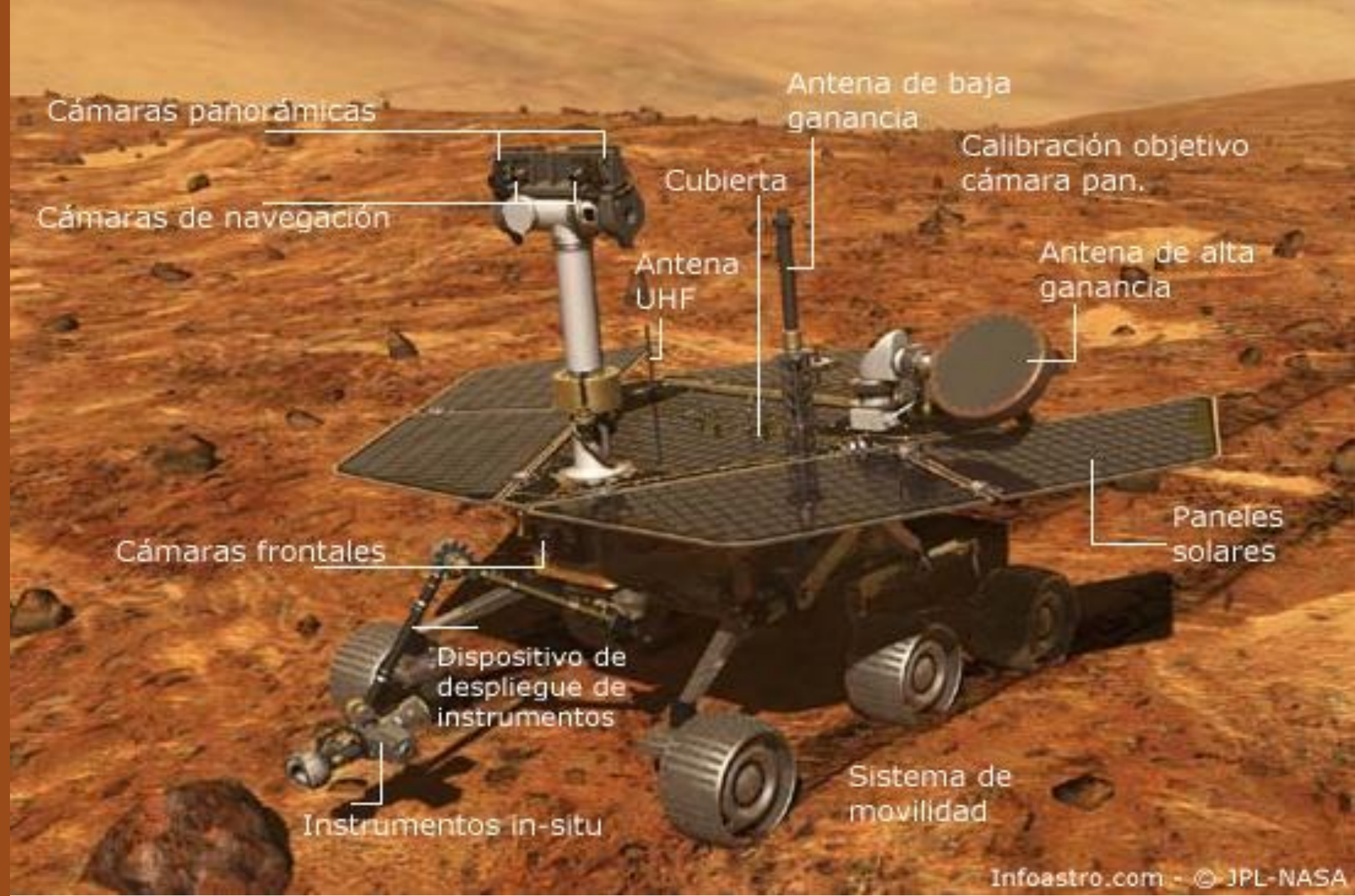
Ing. Eddie Sobrado
sobrado.ea@pucp.edu.pe

Ejemplo 1: Mars Pathfinder

- El primer carro robot en descender a la superficie y desplazarse sobre Marte fue el **Sojourner**. Caminó sobre la superficie de Marte el 6 de julio de 1997.



Gobernada por un procesador con sistema operativo en tiempo real **VXWorks**.



Cámaras panorámicas

Cámaras de navegación

Cámaras frontales

Instrumentos in-situ

Dispositivo de
despliegue de
instrumentos

Cubierta

Antena
UHF

Antena de baja
ganancia

Calibración objetivo
cámara pan.

Antena de alta
ganancia

Paneles
solares

Sistema de
movilidad

Ejemplo 1: Mars Pathfinder

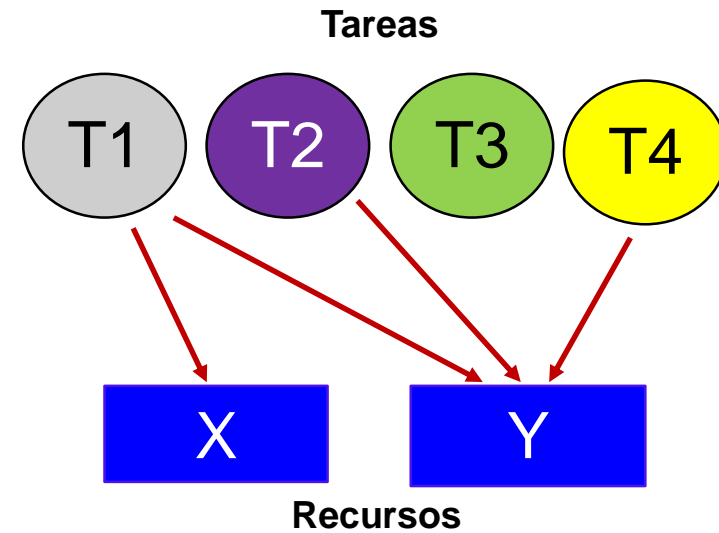
Un único procesador ejecuta **varias tareas**:

- a. **Tarea de alta prioridad**: que tiene que comprobar **periódicamente** que todo el **sistema** funciona correctamente.
- b. **Tareas de media prioridad**: adquisición de imágenes, compresión de información, otros subsistemas.
- c. **Tarea de baja prioridad** manejando una **estación meteorológica** de la que hay que recabar datos en intervalos relativamente grandes.

Tiene una **región crítica** con la tarea de alta **prioridad** para el acceso a **datos comunes**, que causó el fallo debido a una **inversión de prioridad**.

Problema: Interacción entre tareas

La **multitarea** es imprescindible en sistemas en tiempo real



2. Control de vuelo

- **Hacer cada ciclo de 1/180 segundos:**

- ✓ Validar datos de sensores y seleccionar fuente de datos; si hay error reconfigurar el sistema.
- ✓ Realizar cálculos de control externo a 30Hz, una vez cada 6 ciclos: circuitos externos de inclinación, balanceo y guiñada
- ✓ Realizar los cálculos de control interno a 90Hz una vez cada 2 ciclos, de los circuitos internos de balanceo usando como entrada las salidas del paso anterior



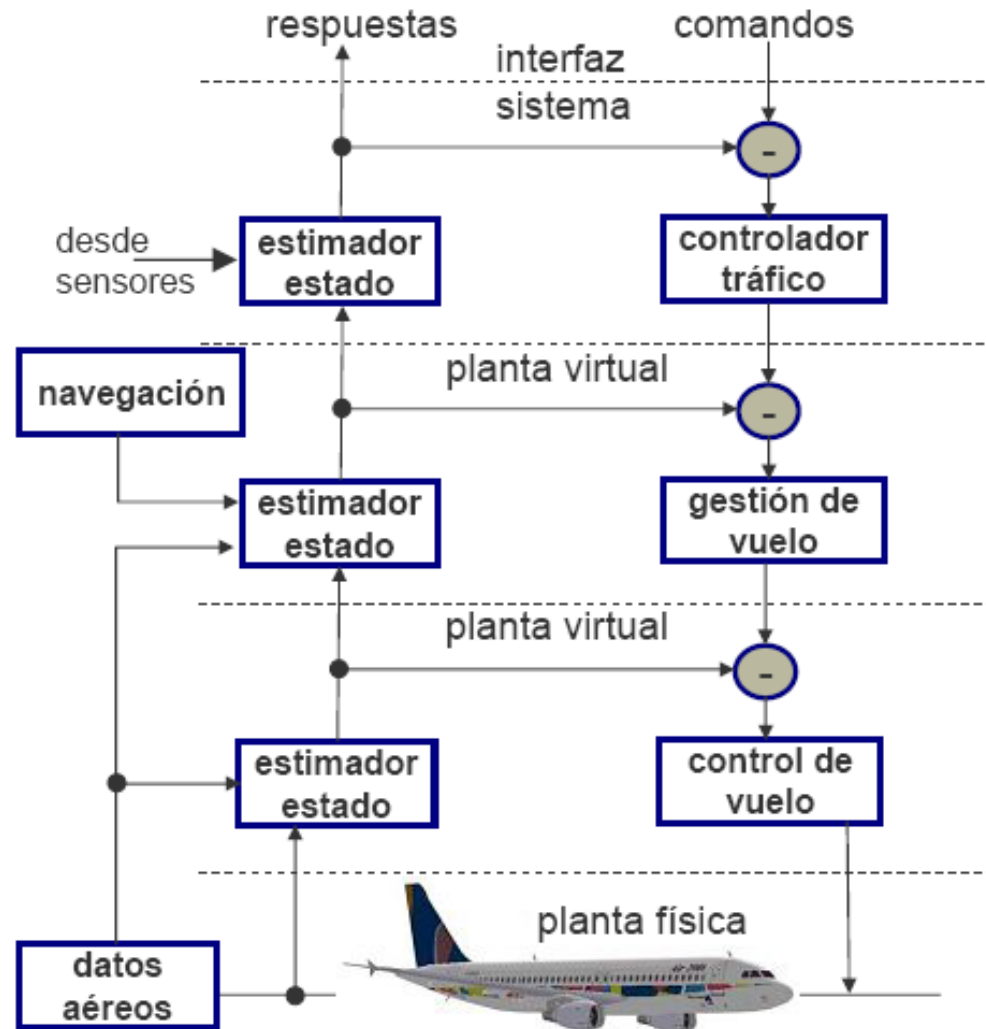
2. Control de vuelo

- ✓ Realizar los cálculos de los circuitos internos de: **guiñada** (depende de control interno anterior) a 20Hz (1x9 ciclos).
- ✓ Transmitir los resultados de salida a los comandos
- ✓ Ejecutar los comandos
- ✓ Esperar al principio del siguiente ciclo



3. Control de vuelo de alto nivel

Jerarquía de control de un controlador de tráfico aéreo





Definición de Tiempo Real

Conceptos básicos de Tiempo Real

Definición para servidores, entornos de comunicación

- a. Proceso en el que el usuario no percibe el tiempo empleado en su ejecución
- b. Tipo de procesamiento en cual una transacción es ejecutada y procesada sin espera alguna

Definición para procesamiento digital de señales

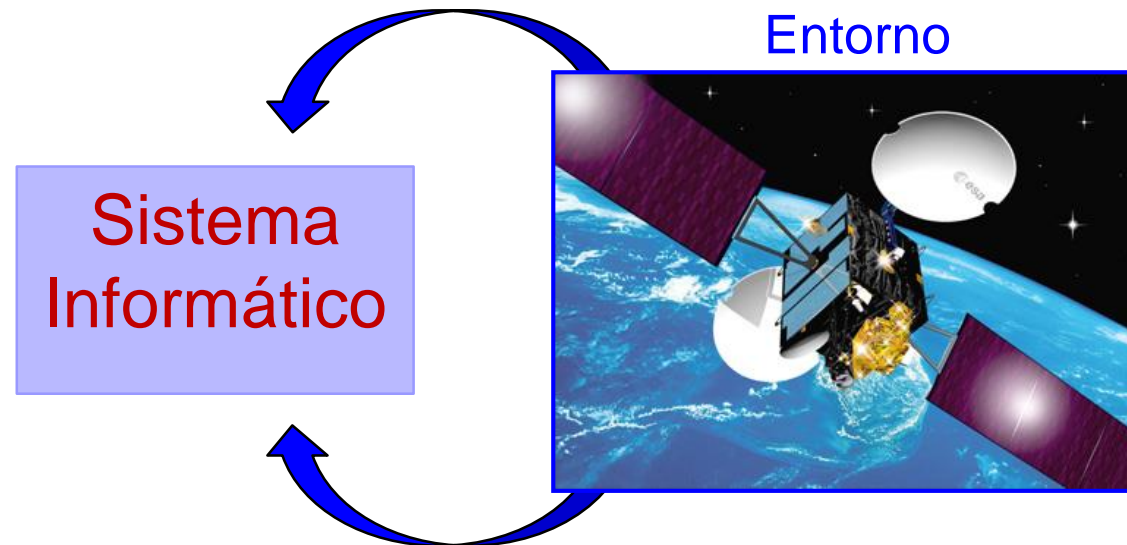
- a. Rápida transmisión y proceso de datos orientados a eventos y transacciones a medida que se producen

Definición para usuarios “comunes”

- a. Un sistema de tiempo real es aquel capaz de procesar una muestra de señal antes de que ingrese al sistema la siguiente muestra

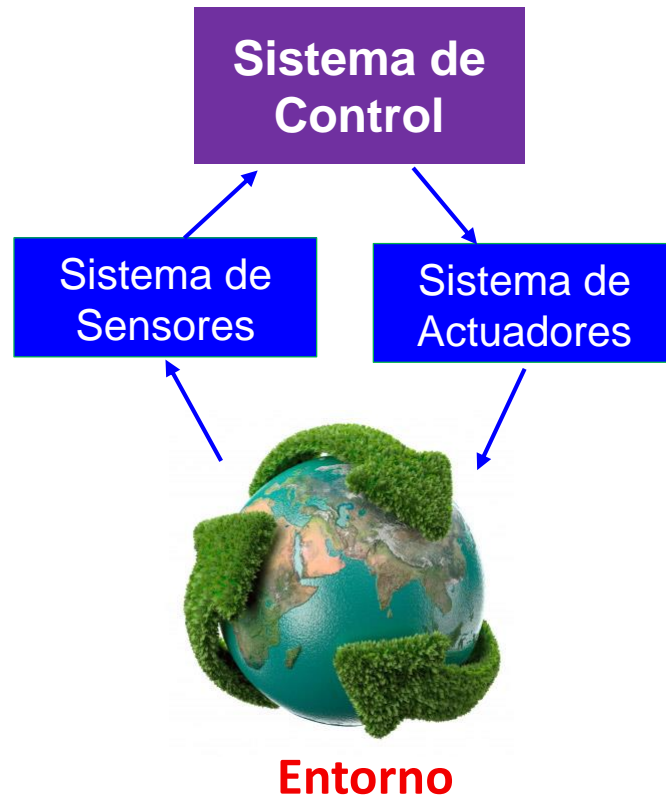
¿Que es un Sistema de Tiempo real ?

- Un sistema de tiempo real es un sistema de procesamiento de información el cual tiene que responder **(en un período finito y específico)** a estímulos de entrada generados externamente
- **NO** es sinónimo de **inmediatez**.



¿Que es un Sistema de Tiempo real ?

- Un **sistema de tiempo real** es un sistema informático que:
 - ✓ **Interacciona repetidamente** con su entorno físico
 - ✓ **Responde** a los estímulos que recibe de dicho entorno en un **plazo de tiempo** determinado

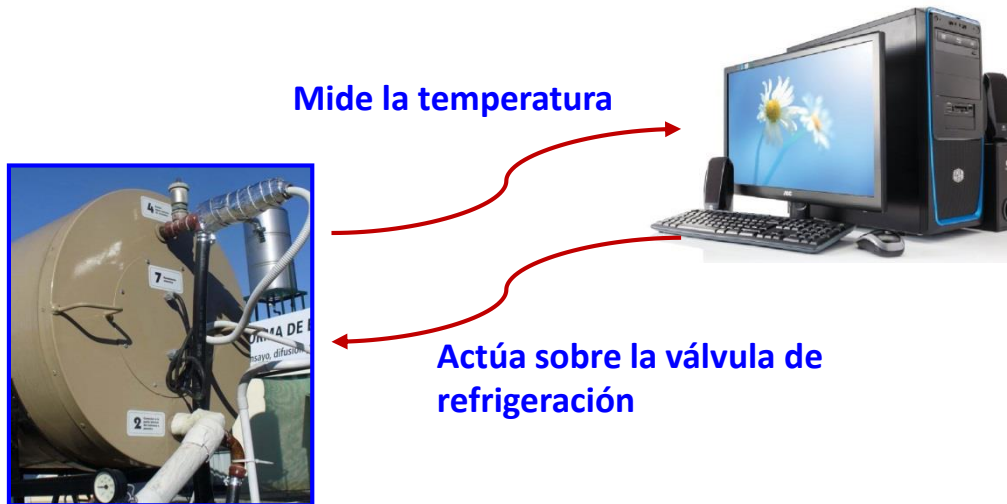


*Para que el funcionamiento del sistema sea correcto no basta con que las **acciones** sean **correctas lógicamente**, sino que además, deben producirse dentro de un intervalo de **tiempo** especificado*



¿Que es un Sistema de Tiempo real ?

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



Requerimiento temporal:

La medida de la temperatura y el control de la válvula debe realizarse con una periodicidad máxima de 10ms



Significado de Tiempo real

- La **reacción** de los sistemas a los **eventos externos** debe ocurrir **durante su evolución**. Como consecuencia, el tiempo del sistema (tiempo interno) debe medirse usando la misma escala usada para medir el tiempo en el entorno controlado (tiempo externo)

Sistema de tiempo real \neq Sistema rápido

- Un STR está definido por:
 - ✓ Los **eventos externos** que debe atender.
 - ✓ La **respuesta** que debe producir ante estos eventos.
 - ✓ Los **requerimientos de temporización** de esas respuestas.

Significado de Tiempo real

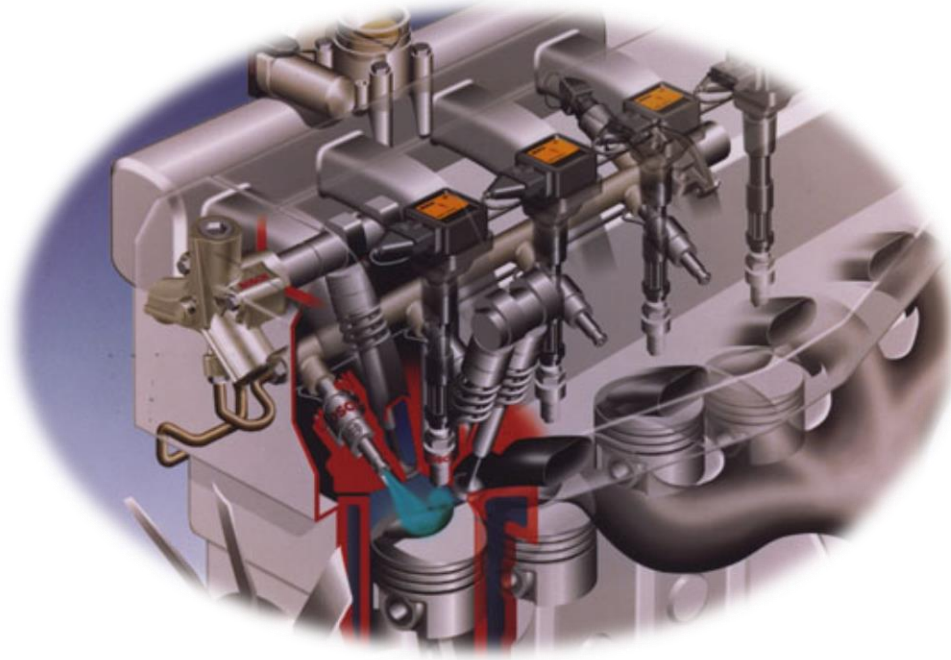
- Si en el sistema se incorporan **eventos que evolucionan más rápidamente** que las acciones que pueden manejarlos, entonces dichas acciones **ya no serán efectivas**.



El tiempo en que se ejecutan las acciones del sistema es significativo

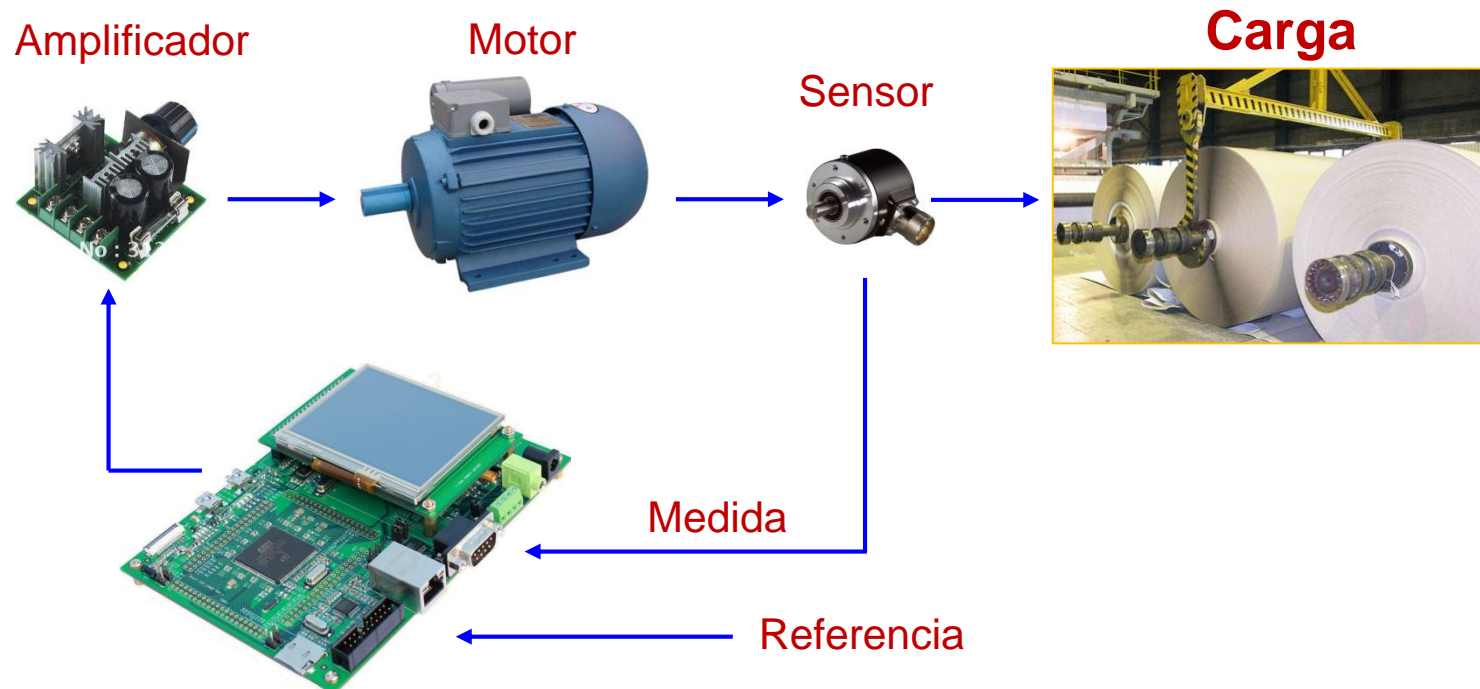
Ejemplo: Inyección de Combustible

- El sistema de control de **inyección de combustible** en un motor alternativo (presentes en autos) debe realizar la inyección de la mezcla dentro del **intervalo de tiempo marcado** por la rotación del motor, de otro modo el motor no funcionará correctamente.

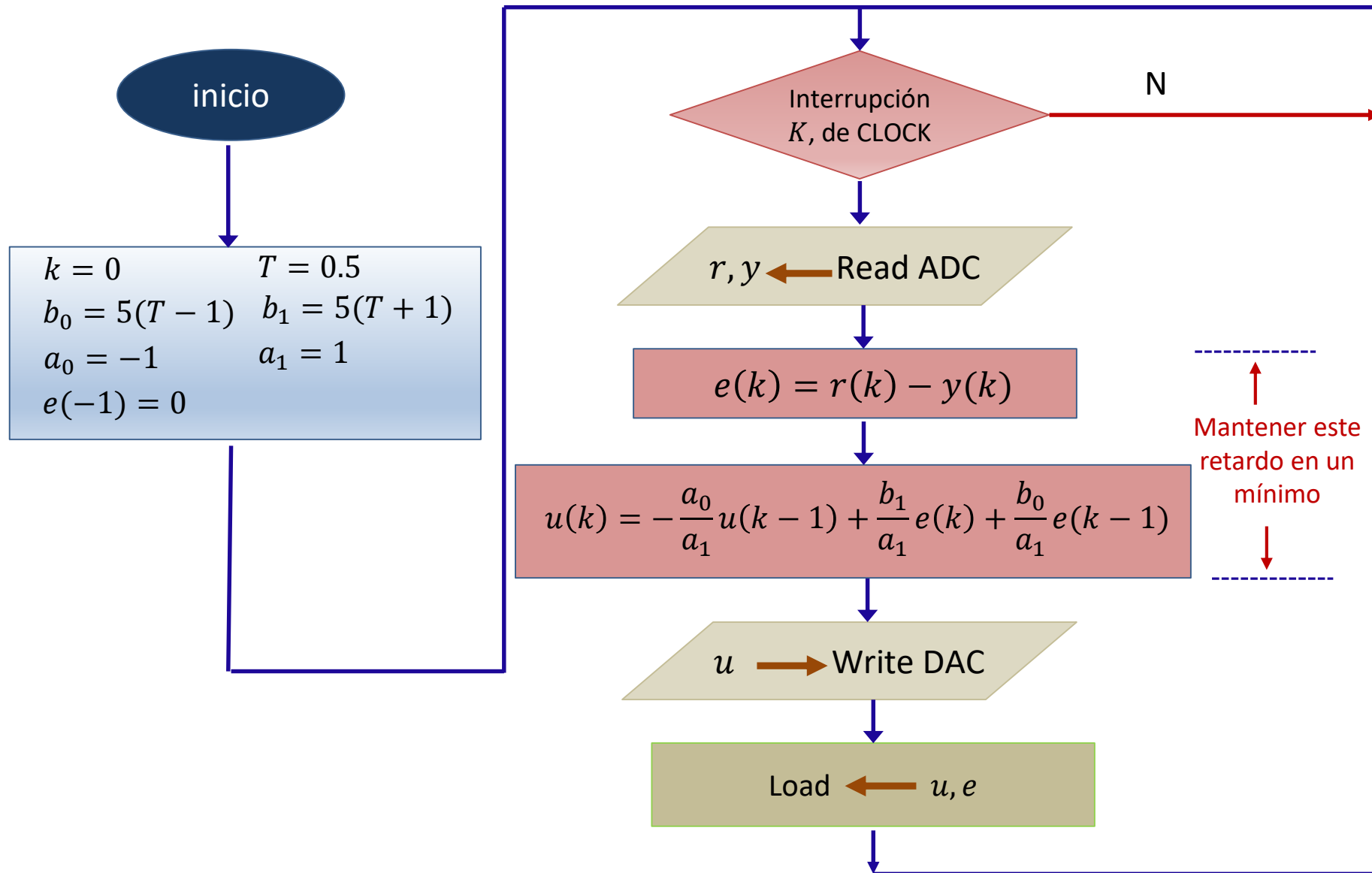


Ejemplo: control de un motor DC o AC

- Si queremos **controlar la velocidad** de un motor sometido a una carga variable utilizando un control **PID**.
- El control **PID** es una función a la que se le pasan unos parámetros (velocidad actual del motor) y retorna el valor de la señal de control que hay que aplicar al motor (la tensión con la que hay que alimentar el motor)



Ejemplo: control de un motor DC o AC

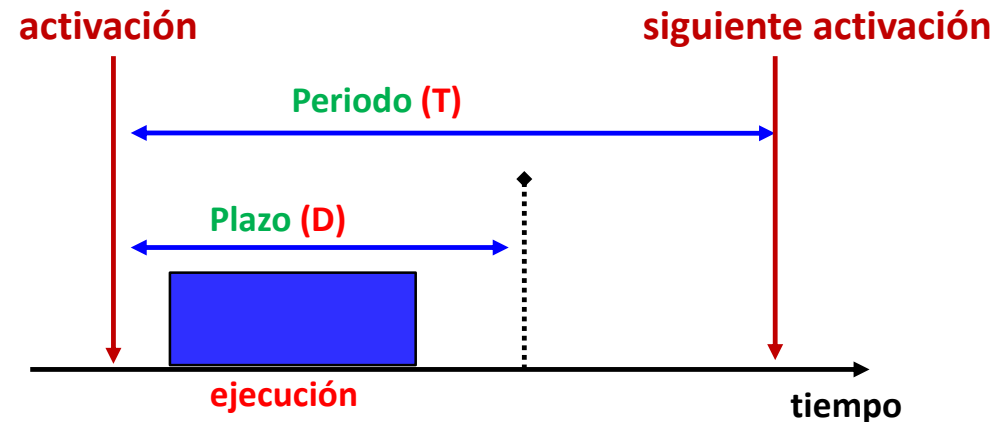


Ejemplo: control de un motor DC o AC

- La teoría supone que el **tiempo de cómputo** es despreciable, esto es, desde que se lee la velocidad del motor hasta que se actúa, pasa muy poco tiempo.
- Normalmente se puede **tolerar un cierto retraso**. Una característica es que el algoritmo PID se tiene que ejecutar periódicamente. Si entre **dos invocaciones consecutivas** a la función PID pasa demasiado tiempo, entonces el motor **puede alcanzar una velocidad no deseada**.

Ejemplo: control de un motor DC o AC

- En **resumen** podemos considerar la función PID como un programa que se ha de **ejecutar de forma periódica (Ti)**; desde que la ponemos en ejecución hasta que acaba, además ha de estar acotado por un **tiempo máximo (Di)** que viene impuesto por la teoría de diseño del PID, y que el código PID emplea una **cantidad de tiempo (Ci)** en función de la velocidad del procesador



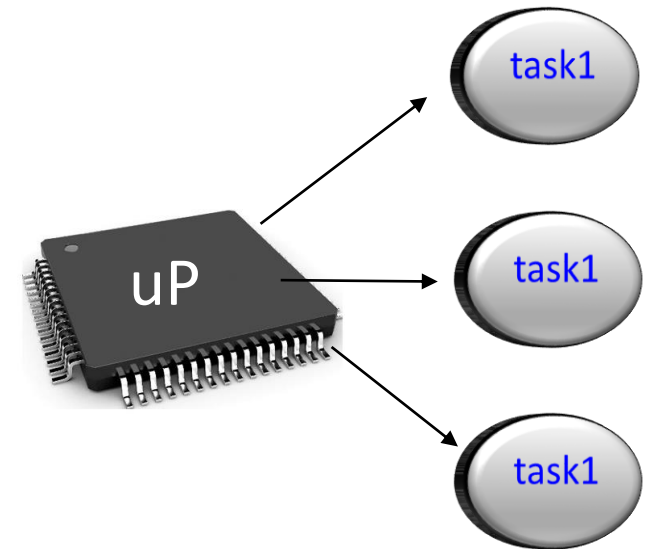
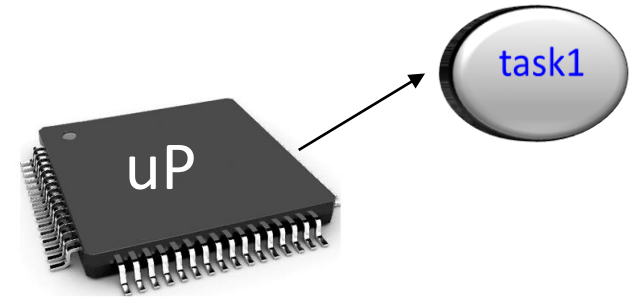
Ti: Periodo de la tarea *i*.

Di: Plazo de ejecución (deadline) de la tarea *i*.

Ci: Tiempo de computo en el **peor de los casos** de la tarea *i*.

Restricción Temporal

- Si el sistema está compuesto por **una sola tarea**, entonces no hay problema de tiempo real. En caso de que el procesador no es lo suficientemente rápido entonces simplemente se cambia de procesador.
- Los problemas de Tiempo Real aparecen cuando el sistema está **compuesto por varias tareas**, y hay que repartir el **procesador (o procesadores)** entre todas ellas.



Restricción Temporal

- Para ello no podemos utilizar un sistema clásico de tiempo compartido como puede ser el utilizado por [Linux](#) con los procesos normales, ni programas que requieran tiempo real en [Windows](#).



Restricción Temporal

- Se debe de usar un sistema operativo de tiempo real para garantizar que su **tarea crítica** se ejecute de manera precisa en el horario establecido.
- El sistema operativo de **Windows** esta diseñado para ofrecer o compartir tiempo del procesador a tareas que corren en **el fondo (background)**.
- Como en este caso **el usuario tiene poco control** sobre cómo el procesador comparte su tiempo, Windows no es considerado un sistema operativo de tiempo real. De esta manera, tareas no críticas (*como un protector de pantalla o software de protección de virus*) pueden **interferir** con un proceso de tiempo crítico en LabVIEW por ejemplo, como una **aplicación de adquisición de datos o un lazo de control**.

Restricción Temporal

- Dado que el **sistema operativo** es el elemento que más **restringe** la **capacidad de reacción** de un sistema de tiempo real, resulta de vital importancia conocer el **comportamiento temporal de sus servicios**. Esta información facilita el análisis y la evaluación del sistema operativo en su fase de desarrollo, lo que permite detectar puntos débiles en el diseño.
- Entre las medidas que **caracterizan temporalmente** a los sistemas operativos de tiempo real se puede nombrar a:

*El cambio de
contexto*

*La latencia de
interrupción*

*La sincronización
de procesos*

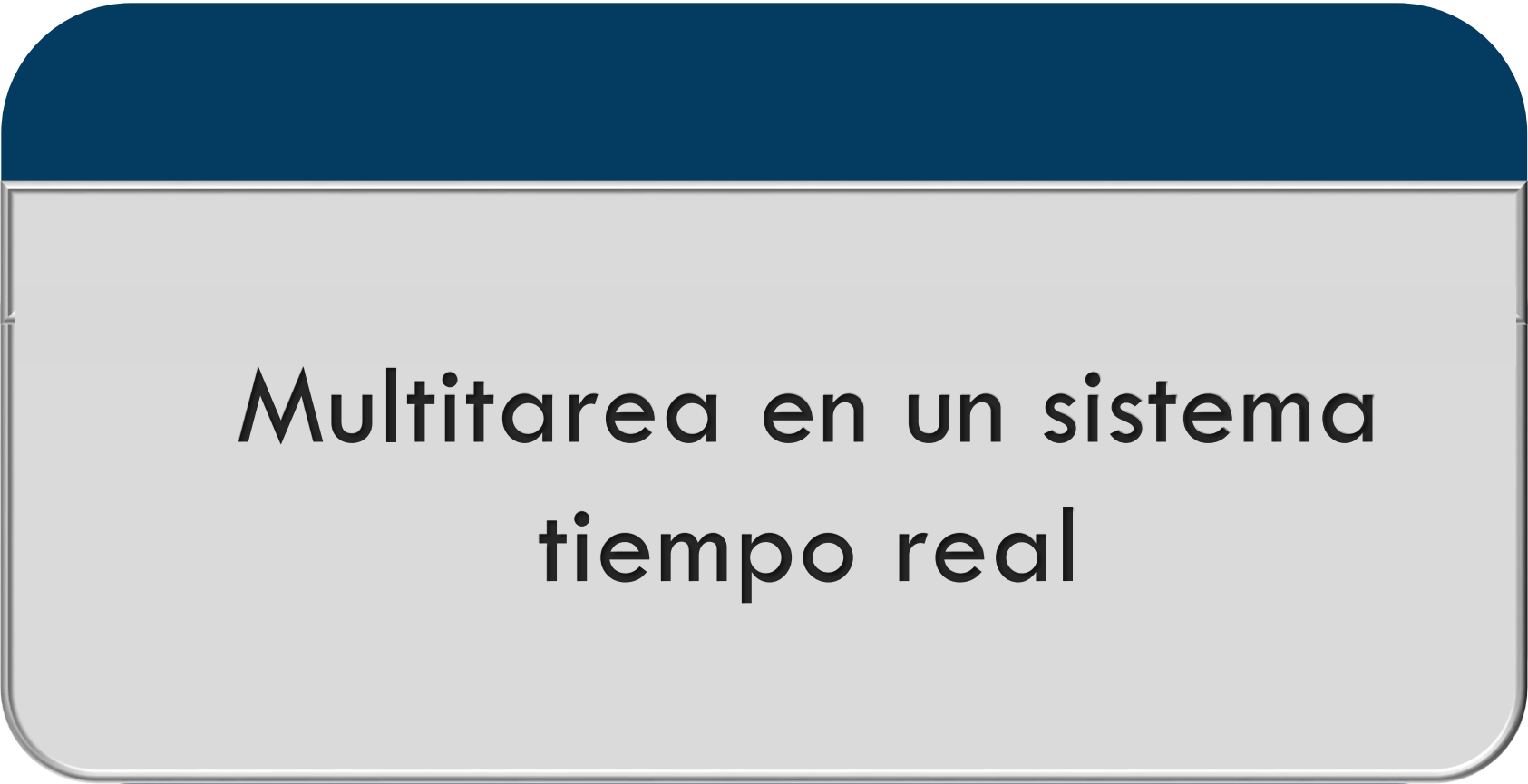
*La comunicación
entre procesos*

Consideraciones importantes

¿Entonces para implementar sistemas Real time que considero?

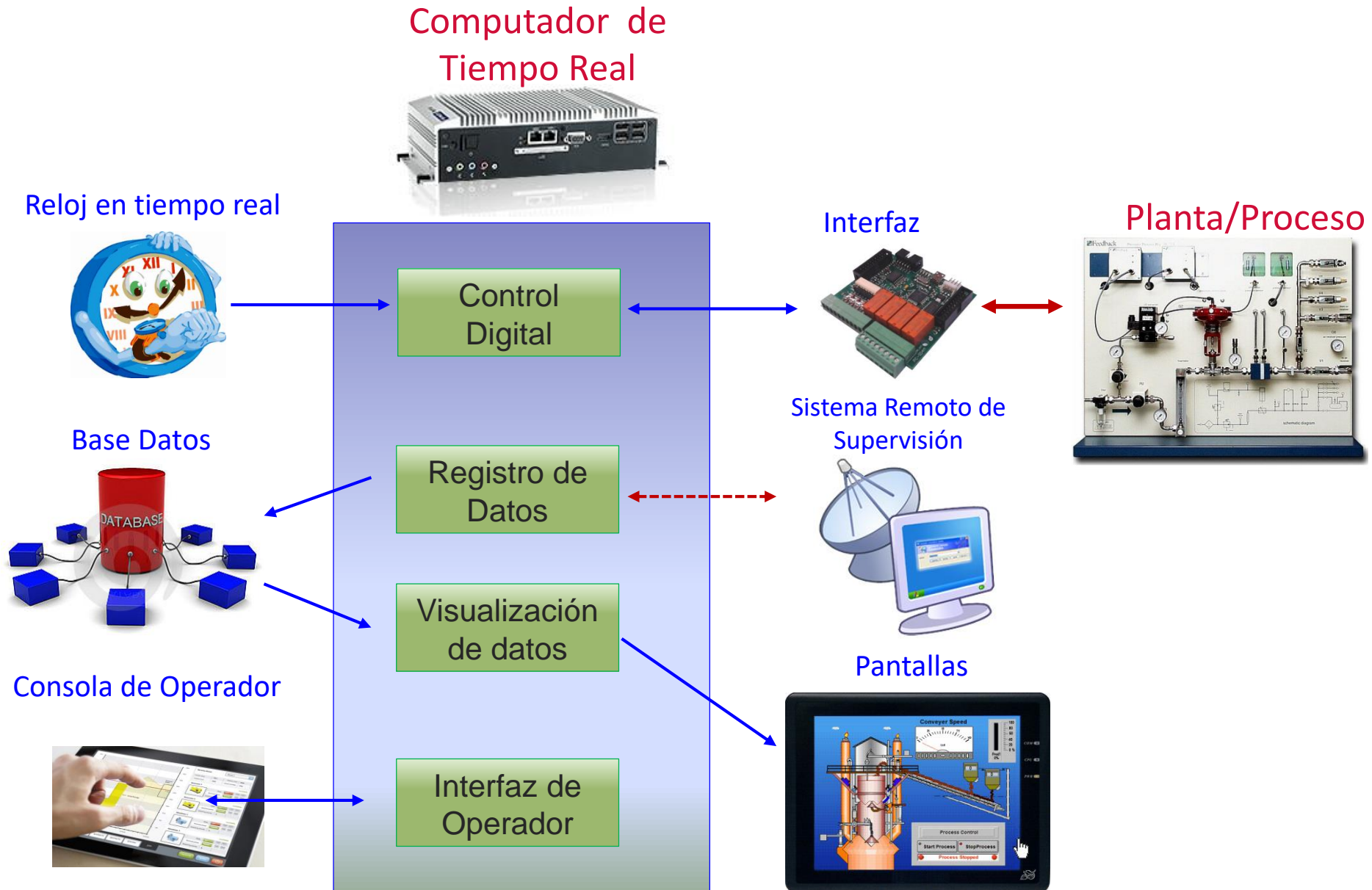


- ✓ Procesador
- ✓ Componentes hardware
- ✓ Sistema operativo
- ✓ Lenguaje de programación
- ✓ Herramientas de desarrollo de software
- ✓ Herramientas de depuración de software
- ✓ Componentes software



Multitarea en un sistema tiempo real

Estructura de un STR típico



Necesidad de Multitarea

- Lo habitual es que un **sistema de tiempo real** tenga que **realizar varias actividades** de forma **concurrente**
- Por ejemplo, tareas en el controlador de un auto:
 - ✓ Medir la velocidad cada 200 ms
 - ✓ Controlar la inyección cada 40 ms
 - ✓ Control del ABS cada 80 ms
- Para el programador resulta más sencillo trabajar sobre cada una de las tareas de **forma independiente**
- El sistema operativo o el lenguaje de programación proporcionará el soporte para que las tareas se **ejecuten de forma concurrente**

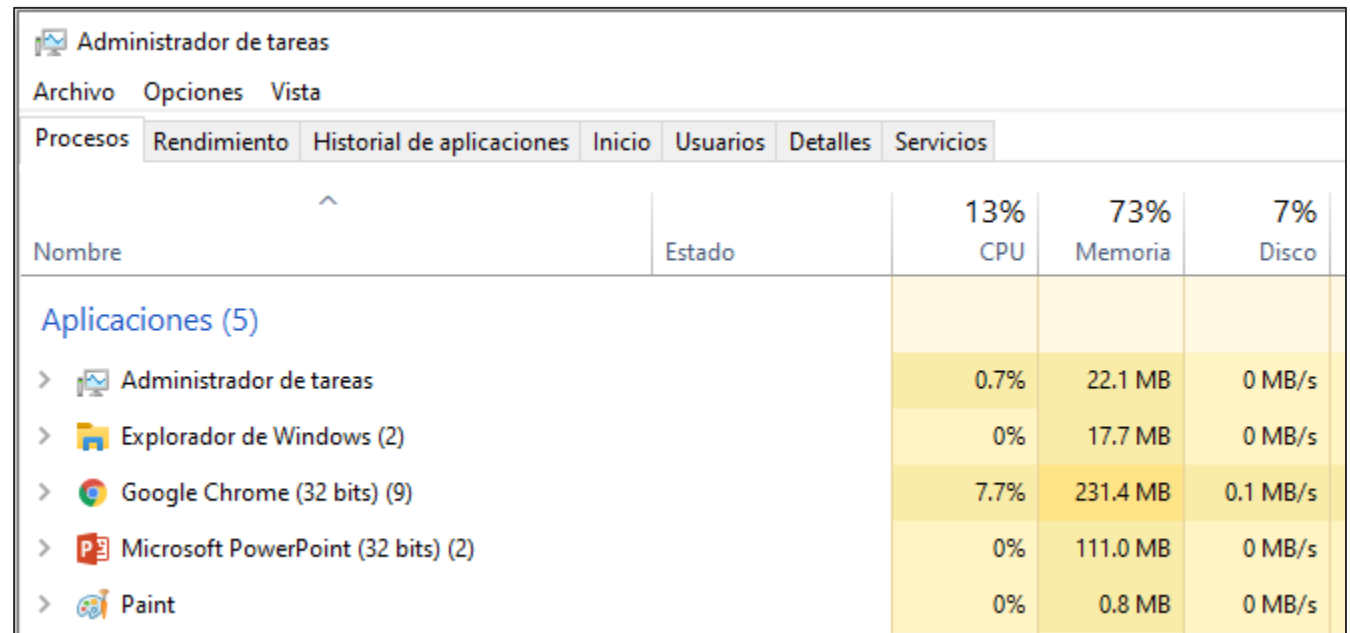


Necesidad de Multitarea

- Tarea Idle

Cuando ninguna de las tareas requiere del procesador, el sistema ejecuta una tarea llamada **idle** u **ociosa**.

Esto me permite fácilmente contabilizar el **nivel de ocupación del CPU**, poner al mismo en modo de bajo consumo o correr cualquier tarea que pudiera ser de utilidad para el sistema cuando no debe atender ninguno de sus eventos.



Administrador de tareas

Archivo Opciones Vista

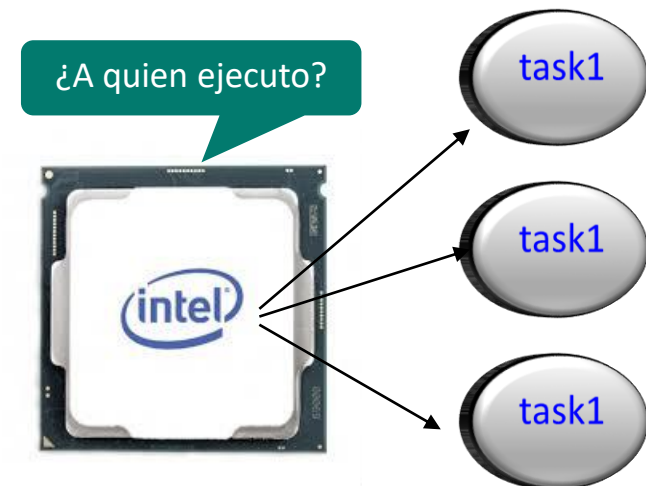
Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios

Nombre	Estado	13% CPU	73% Memoria	7% Disco
Aplicaciones (5)				
> Administrador de tareas		0.7%	22.1 MB	0 MB/s
> Explorador de Windows (2)		0%	17.7 MB	0 MB/s
> Google Chrome (32 bits) (9)		7.7%	231.4 MB	0.1 MB/s
> Microsoft PowerPoint (32 bits) (2)		0%	111.0 MB	0 MB/s
> Paint		0%	0.8 MB	0 MB/s

Consideraciones en aplicaciones Multitarea

- También, debe hacerse un análisis de tiempos, eventos y respuestas más cuidadoso. Al usar un RTOS ya no es el programador quién decide **cuándo ejecutar cada tarea**, sino el **SCHEDULER** (parte del SO).
- Cometer un error en el uso de las reglas de ejecución de tareas puede llevar a que los eventos se procesen fuera del tiempo especificado o que no se procesen en lo absoluto.

El CPU **consume tiempo** en determinar en todo momento qué tarea debe estar corriendo. Si el sistema debe **manejar eventos** que ocurren **demasiado rápido** tal vez **no** haya tiempo para esto.



Consideraciones en aplicaciones Multitarea

- El CPU **consume tiempo** cada vez que debe cambiarse la **tarea en ejecución**.
- Se **consume espacio** de **memoria de código** para implementar la funcionalidad del RTOS.
- Se consume espacio de **memoria de datos** en mantener una pila y un TCB (bloque de control de tarea) por cada tarea
 - El tamaño de estas pilas suele ser configurable POR TAREA, lo cual mitiga este impacto.





Requisitos Funcionales

1. Adquisición de datos

a. Medida de variables

- Cada variable pertenece a la etapa de control de un subsistema
- La imagen de una variable en el STR tiene un intervalo de validez
- Las medidas pueden estar dirigidas por tiempo o por sucesos

b. Acondicionamiento de señales

- Los datos brutos se filtran y convierten a unidades de ingeniería
- Luego se analizan para obtener datos validados

c. Supervisión

- Cuando algún dato tiene valores incorrectos se generan alarmas. Es importante identificar el suceso original de una serie de alarmas encadenadas



1. Adquisición de datos



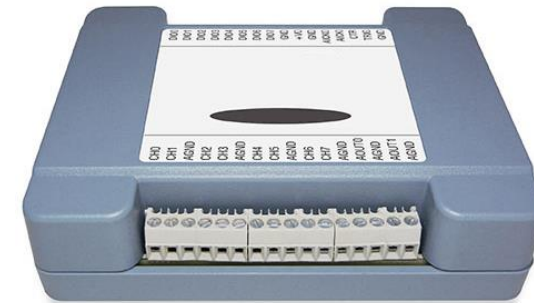
Tarjetas de adquisición PCI



Tarjetas de Red CAN



Tarjetas de adquisición
modulares

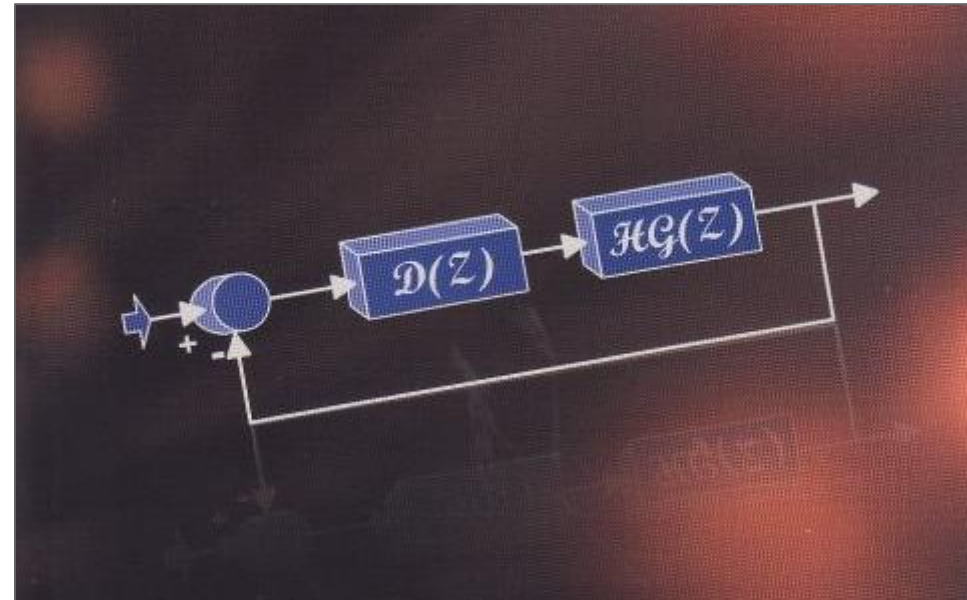


Tarjetas de adquisición
USB

2. Control Digital

- El sistema de tiempo real puede actuar sobre el sistema controlado para conseguir que tenga un comportamiento determinado.
- La **acción de control** es función de la desviación entre los valores de **referencia** y los valores medidos de las variables
- El diseño de los **algoritmos de control** es un problema importante

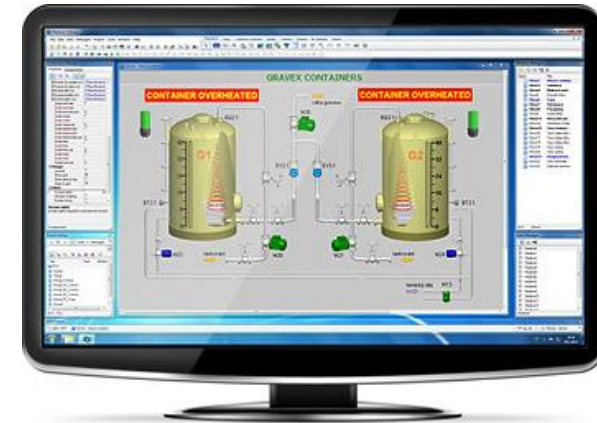
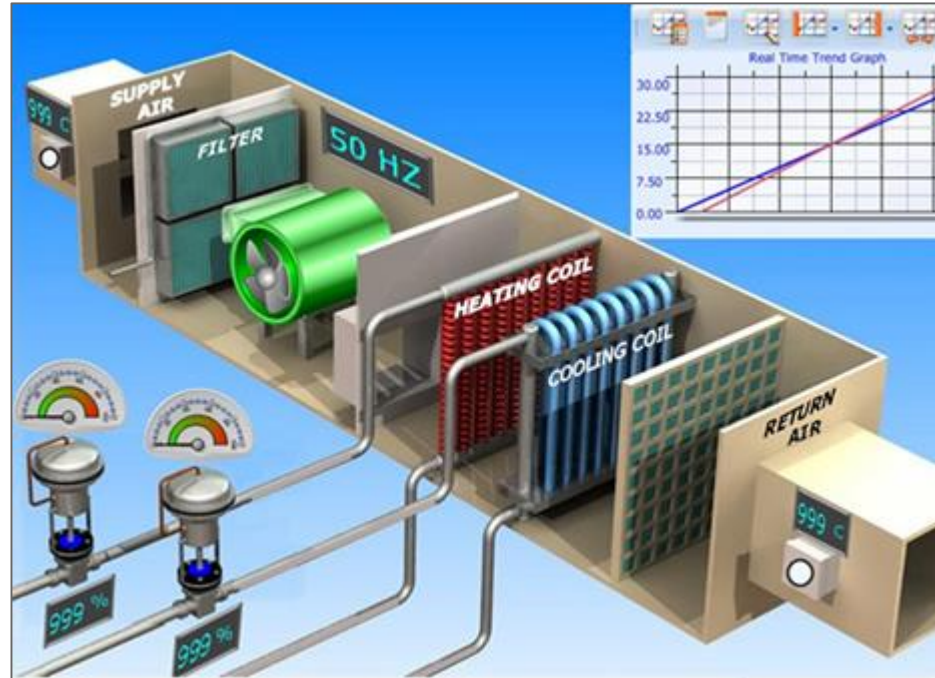
- El comportamiento es muy regular (**muestreo periódico**)



3. Interacción con Personas

a. La interfaz de operador es muy importante

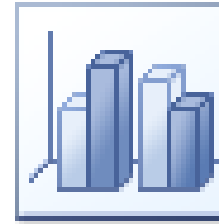
- Una mala interfaz puede causar accidentes



3. Interacción con Personas

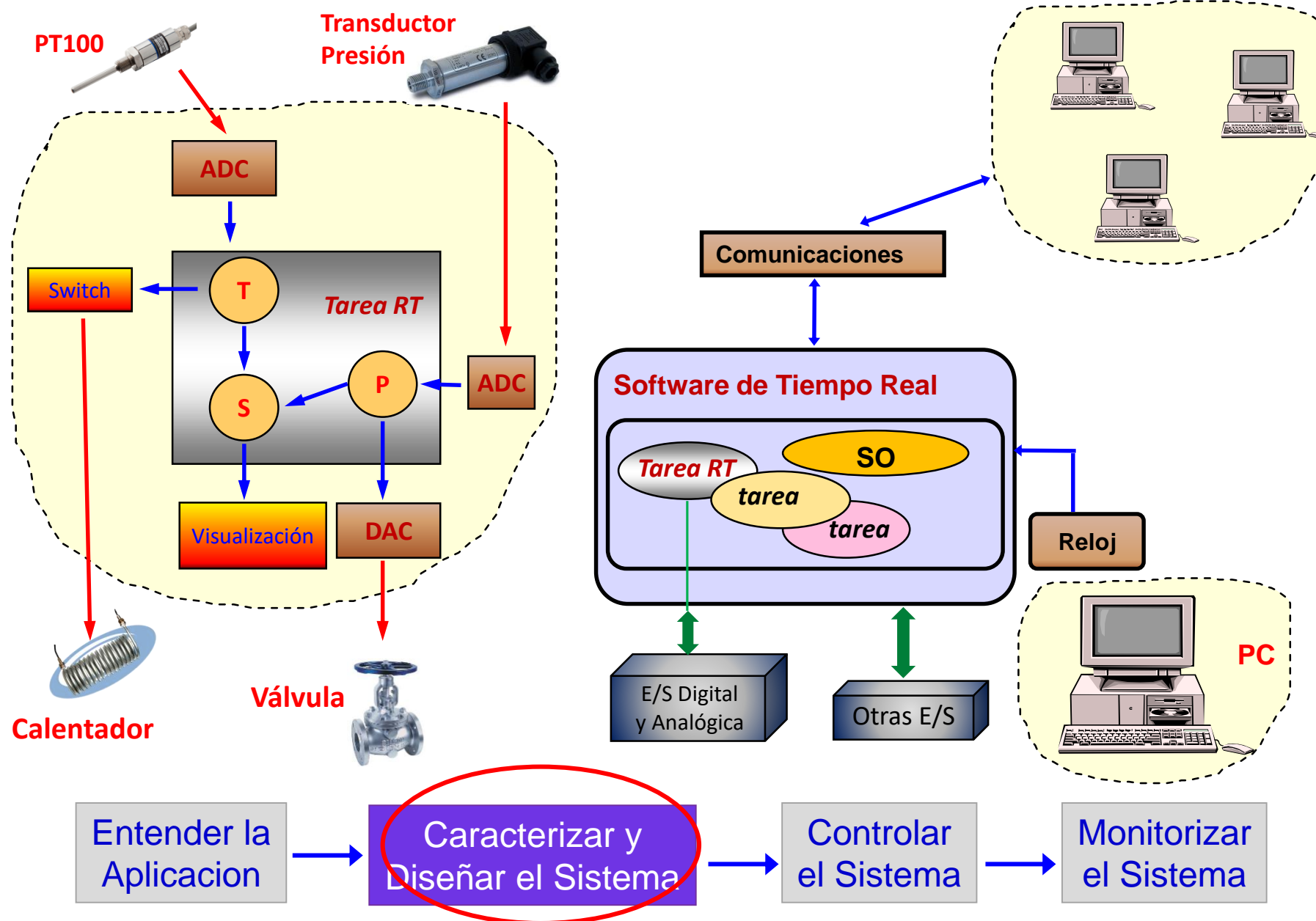
b. Aspectos típicos:

- Presentación de datos
- Presentación de alarmas
- Presentación de tendencias
- Registro de datos
- Generación de informes



  		
TAG	Hora	Texto
Presión	16:35	Presión Caldera >2 bar
Temperatura	16:36	Temperatura Reactor >2000

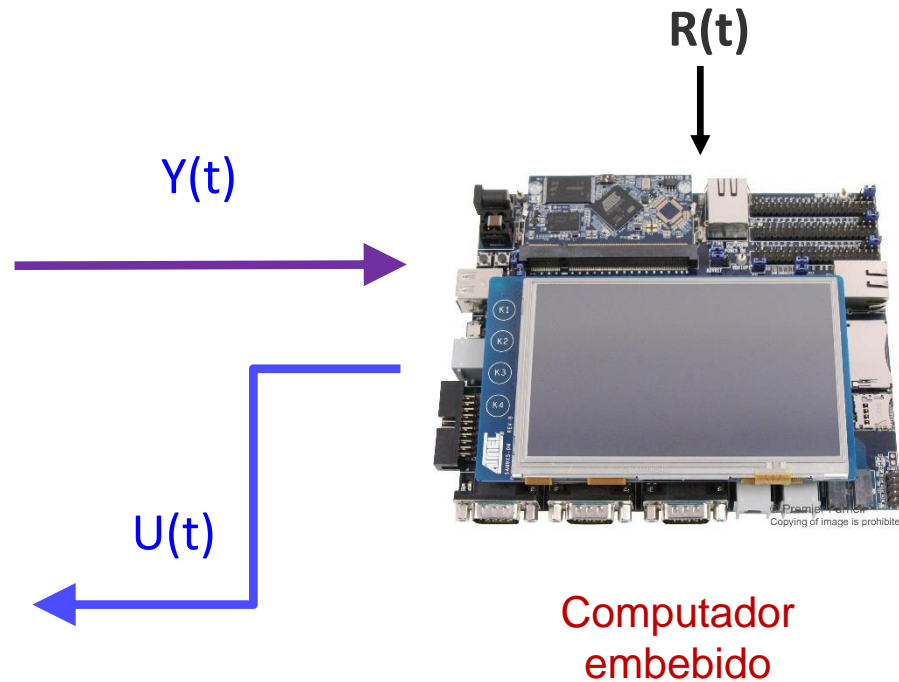
Etapas de desarrollo: **Caracterización**



Caracterización de la Aplicación



Componentes: tanque, sensores, actuadores, interfaces, conversores, adaptadores etc.



Caracterización de la Aplicación

1. 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 sistema.

Caracterización de la Aplicación

2. 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 (D_j) dependen de la aplicación.
- Se trata de asegurar que el tiempo de respuesta (R_j) sea: $R_j \leq D_j$

D_j

Resumen de los requisitos Temporales

Tarea	T	C	P	U
τ_1	16	4	1	0.250
τ_2	40	5	2	0.125
τ_3	80	32	3	0.400
				0.775

Se hace una análisis de cargabilidad del procesador

