



SISTEMAS EMBEBIDOS

SOFTWARE EMBEBIDO

- Algunos sistemas computacionales se utilizan para controlar dispositivos, como artefactos domésticos, controladores de juego, líneas de montaje, etc. interactuando directamente con el hardware.
- El software debe reaccionar a eventos generados por el hardware y emitir señales de control en respuesta a esos eventos.
- El software en estos sistemas esta embebido en el hardware generalmente en memorias de solo lectura y debe responder en tiempo real a los eventos.



RESPUESTA DE LOS SISTEMAS EMBEBIDOS

- La respuesta en tiempo real es la diferencia entre los sistemas embebidos y otros sistemas de software, tales como los sistemas de información.
- Para los sistemas que no son de tiempo real se debe comprobar que en respuesta a una entrada el sistema genera la salida correcta.
- En un sistema de tiempo real, la salida correcta depende tanto de la respuesta a la entrada como del tiempo que tarda en generar dicha respuesta



DEFINICIÓN

Un sistema de software de tiempo real es un sistema cuya correcta operación depende tanto de los resultados producidos por el sistema como del tiempo en que se producen dichos resultados.

Si los resultados no se producen según la especificación de tiempo se considera una falla del sistema.



CARACTERÍSTICAS:

- Operan de manera continua, es decir, su operación no tiene fin.
- Las interacciones con el entorno del sistema son incontrolables e impredecibles.
- Hay limitaciones físicas que afectan el diseño de un sistema. (limitaciones de energía o espacio físico)
- Se requiere interacción directa con el hardware.
- La protección y fiabilidad dominan el diseño del sistema.



DISEÑO DE SISTEMAS EMBEBIDOS

- En el diseño del software se deben considerar el diseño y el rendimiento del hardware.
- Parte del proceso de diseño comprende decidir de cuáles funcionalidades del sistema se implementaran por software y cuáles por hardware.
- Las decisiones del hardware, software de soporte y temporización del sistema deben considerarse al inicio del proceso.
- El enfoque más utilizado se basa en un modelo estímulo-respuesta.



SISTEMAS REACTIVOS

- Dado un estímulo, el sistema debe reaccionar generando una respuesta correcta en el tiempo correcto.
- **Estímulos periódicos:** Ocurren a intervalos predecibles.
 - El sistema examina un sensor cada 50 milisegundos y responde según el valor de dicho sensor.
- **Estímulos no periódicos:** Ocurren de manera irregular e impredecible, y se señalan mediante el mecanismo de interrupción.
 - Una interrupción que indique que se completó una transferencia I/O .

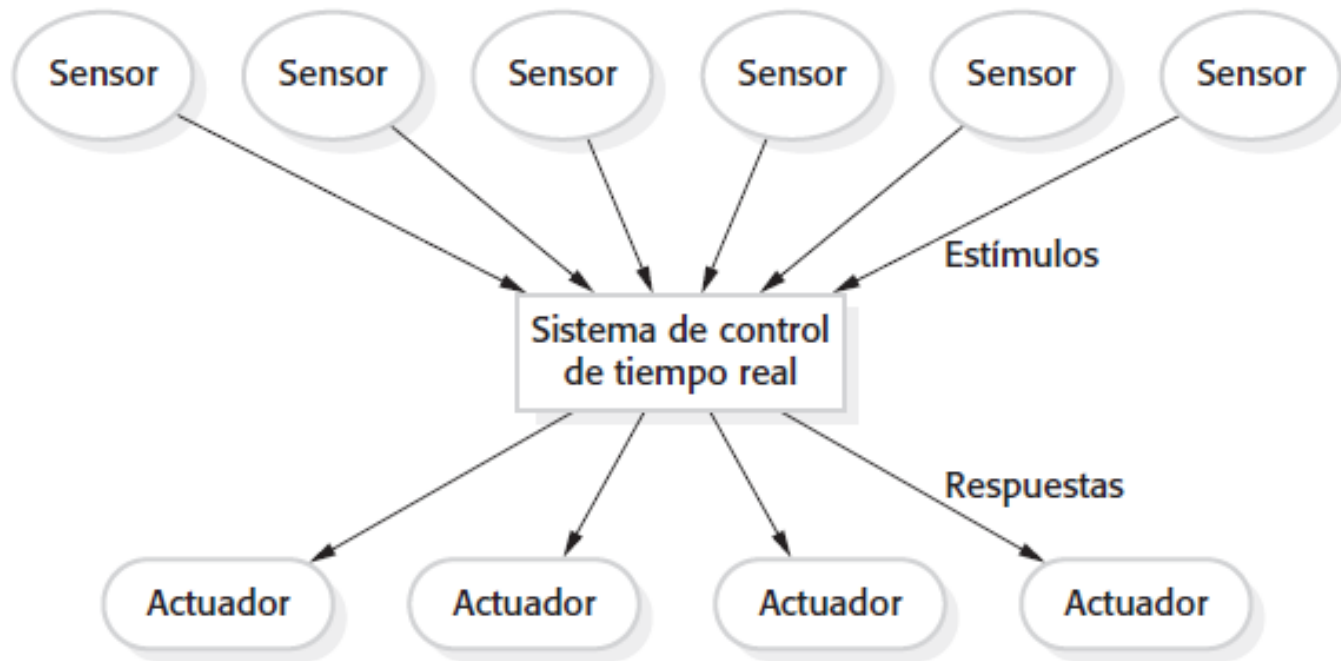


ESTÍMULOS EN UN SISTEMA DE ALARMA

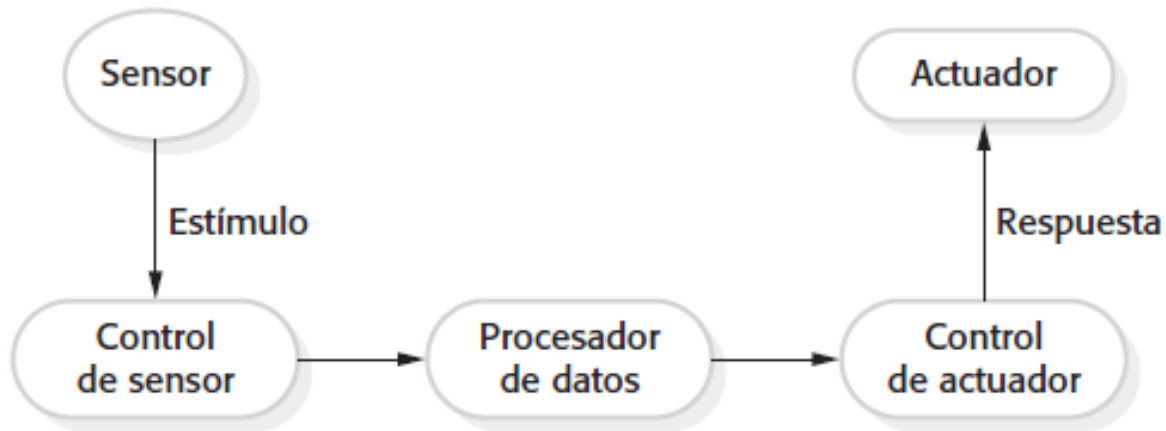
Estímulos	Respuesta
Sensor individual positivo	Iniciar alarma; encender luces alrededor del sitio del sensor positivo.
Dos o más sensores positivos	Iniciar alarma; encender luces alrededor de los sitios de los sensores positivos; llamar a la policía con ubicación de entrada forzada del sospechoso.
Caída de voltaje entre el 10 y 20%	Cambiar a batería de respaldo; efectuar prueba de suministro de energía.
Caída de voltaje de más del 20%	Cambiar a batería de respaldo; iniciar alarma; llamar a la policía; efectuar prueba de suministro de energía.
Falla en el suministro de energía	Llamar a servicio técnico.
Falla de sensor	Llamar a servicio técnico.
Consola de botón de pánico positivo	Iniciar alarma; encender luces alrededor de consola; llamar a la policía.
Inicializar valores de la alarma	Apagar todas las alarmas activas; apagar todas las luces que se hayan encendido.



MODELO GENERAL DE UN SISTEMA EMBEBIDO



SENSORES Y ACTUADORES



CONSIDERACIONES DE ARQUITECTURA

- Cuando se recibe un estímulo, el control se debe transferir al manejador correcto.
- Los sistemas de software de tiempo real se diseñan como un conjunto de procesos cooperativos concurrentes.



PROCESO DE DISEÑO

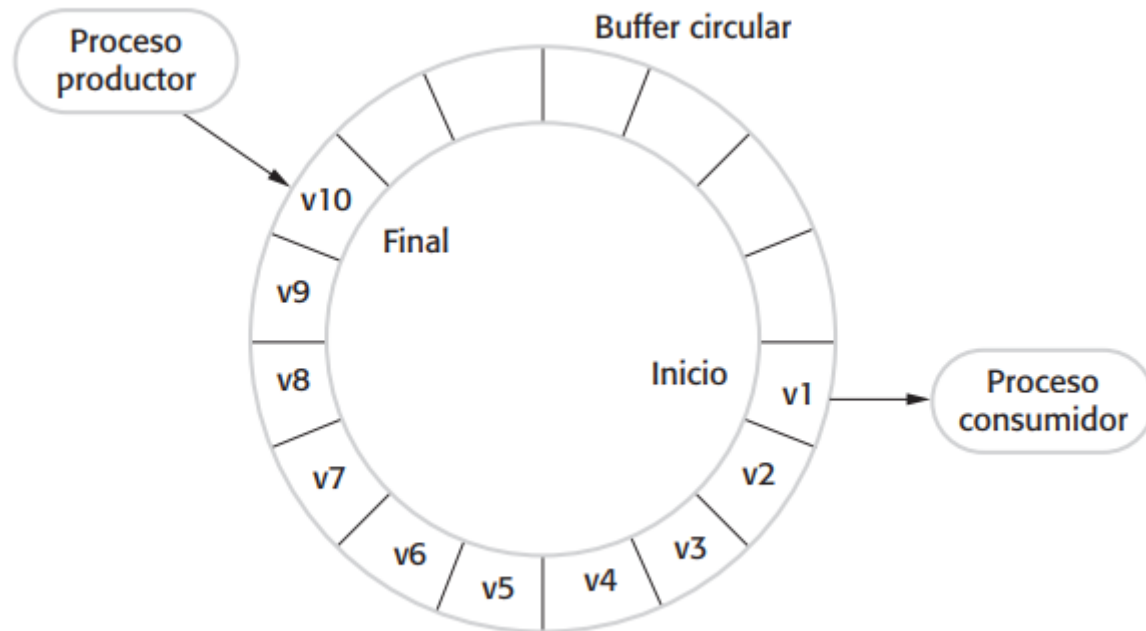
- Selección de plataforma: se elige una plataforma de ejecución para el sistema (hardware y sistema operativo de tiempo real a utilizar).
- Identificación de estímulos/respuestas: los estímulos que debe procesar el sistema y la respuesta o respuestas asociadas para cada estímulo.
- Análisis de temporización: se identifican las restricciones de temporización que se aplican al estímulo y a la respuesta.
- Diseño de procesos: se realiza el diseño de procesos concurrentes.
- Diseño de algoritmo: para cada estímulo y respuesta se diseñan algoritmos que realizan los cálculos requeridos.
- Diseño de datos: Se especifica la información que se intercambia y se diseñan las estructuras de datos. Varios procesos concurrentes pueden compartir estas estructuras de datos.
- Planeación del proceso: Se diseña un sistema de planeación que garantice que los procesos iniciarán a tiempo para cumplir sus plazos.



COORDINACIÓN DE PROCESOS

- Los procesos en un sistema de tiempo real deben coordinarse para compartir información.
- Los mecanismos de coordinación garantizan la exclusión mutua para los recursos compartidos.
- Cuando un proceso modifica un recurso compartido, otros procesos no podrán cambiar dicho recurso.
- El intercambio de información entre procesos requiere que los procesos se ejecuten a diferentes velocidades. Un proceso genera información, otro proceso consume esa información..





EXCLUSIÓN MUTUA

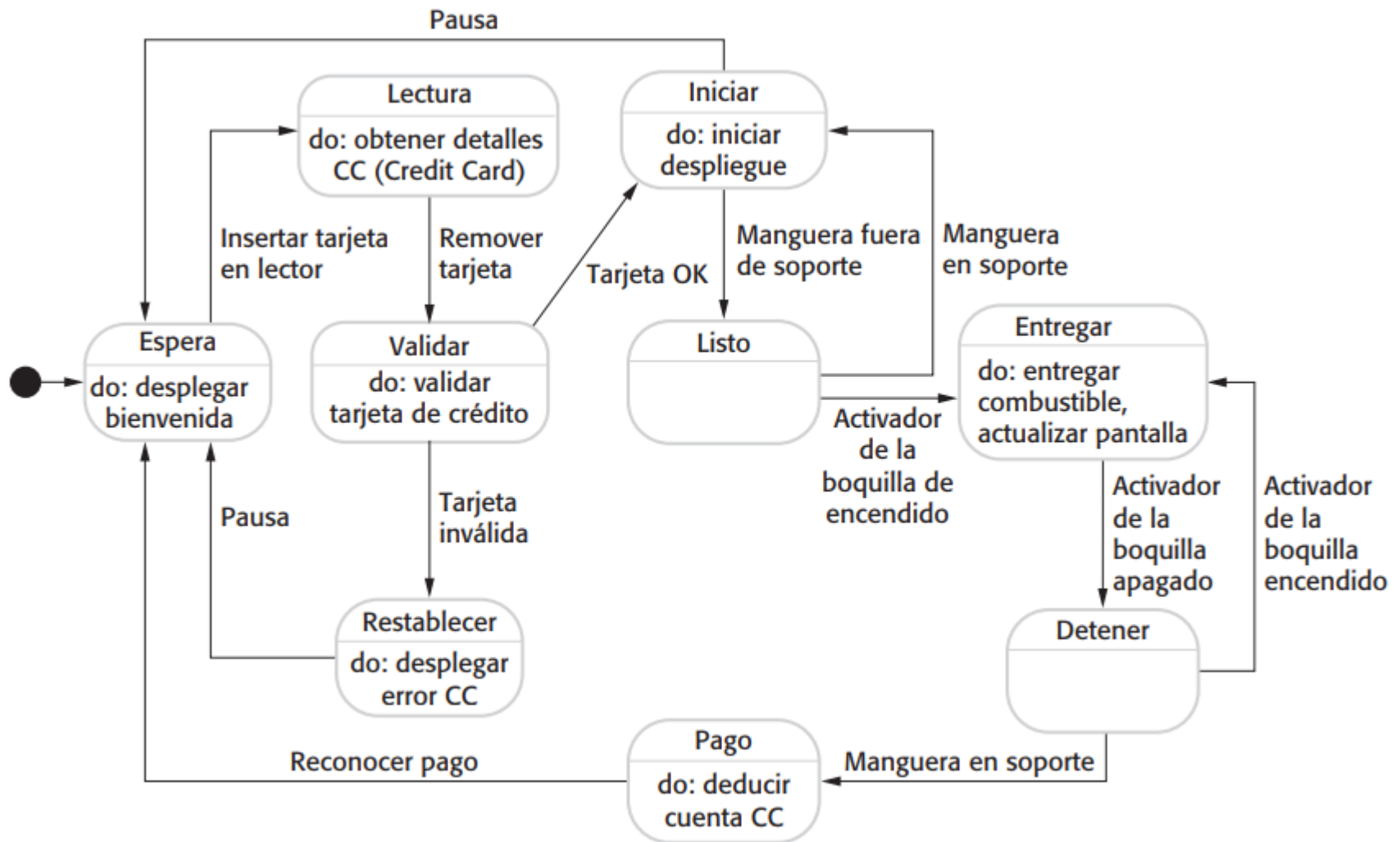
- El proceso productor obtiene los datos y los coloca en el buffer. El proceso consumidor toma los datos desde el buffer .
- Los procesos productor y consumidor debe excluirse mutuamente cuando acceden al mismo elemento.
- El proceso productor no puede agregar datos en un buffer lleno y el proceso consumidor no puede tomar datos de un buffer vacío.



MODELADO DE SISTEMAS DE TIEMPO REAL

- Un estímulo produce un cambio en el sistema que lo hace pasar de un estado a otro.
- Se utiliza los modelos de estado de UML





PROGRAMACION EN TIEMPO REAL

Los sistemas de tiempo real duros se programan muchas veces en lenguaje ensamblador, de modo que pueda cumplirse con los plazos ajustados.

También se usan ampliamente los lenguajes a nivel de sistemas, tales como C, que permiten la generación de un código eficiente.



PATRONES

- Observar y reaccionar: Se utiliza cuando un conjunto de sensores se monitorizan manera rutinaria. Cuando un sensor indican que sucedió cierto evento, el sistema reacciona iniciando un proceso para manejar dicho evento.
- Control ambiental: Emplea sensores que proporcionan información sobre el entorno y actuadores que pueden cambiar el entorno. En respuesta a los cambios ambientales detectados por el sensor, se envían señales de control a los actuadores del sistema.
- Segmentación de proceso (pipeline) Se recopilan datos del entorno del sistema que deben transformarse para analizarlos y procesarlos (Analógico-digital)



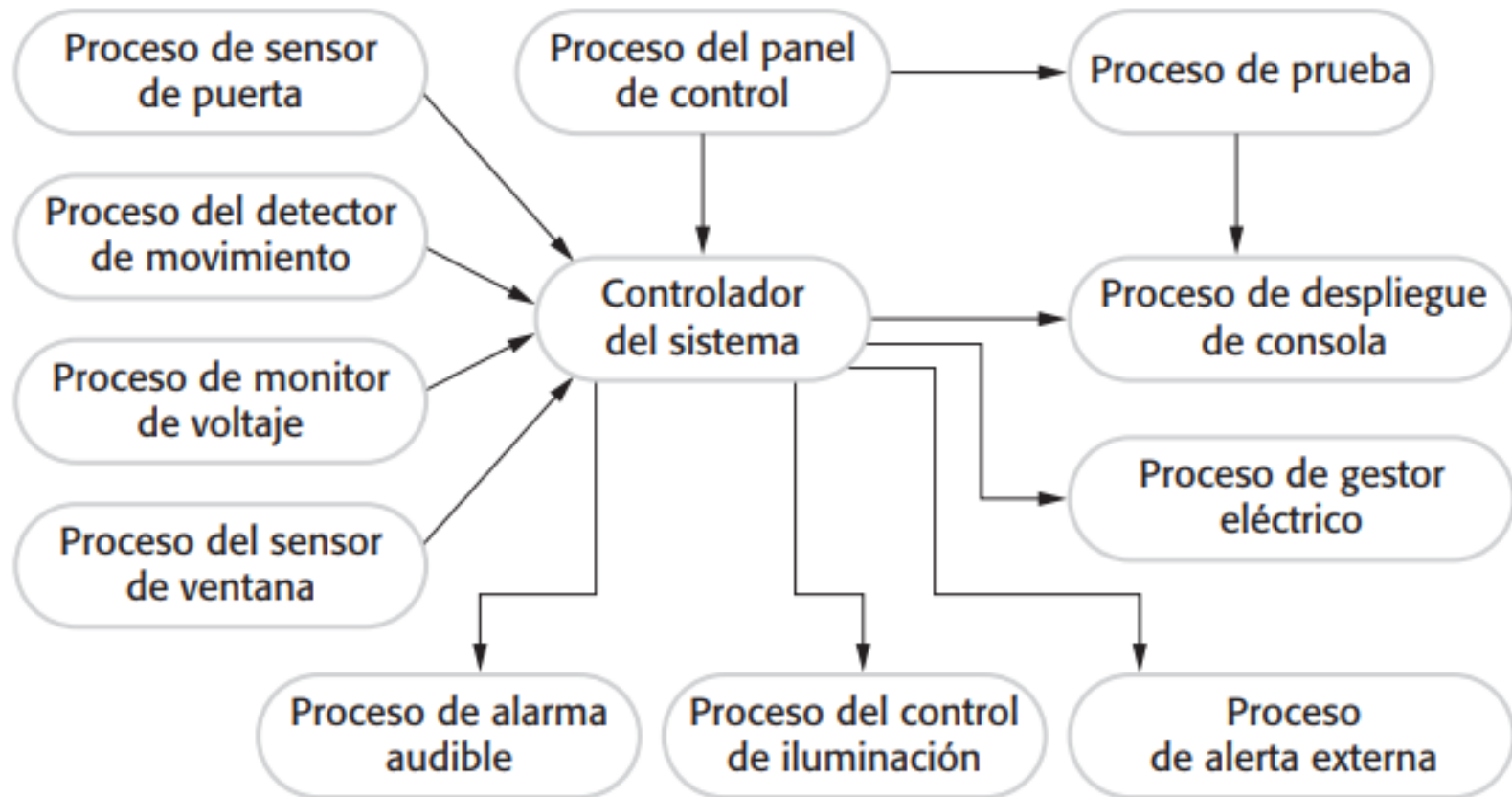
PATRON OBSERVAR Y REACCIONAR

Nombre	Observar y reaccionar
Descripción	Se recopilan y analizan los valores de entrada de un conjunto de sensores de los mismos tipos. Dichos valores se despliegan en alguna forma. Si los valores de sensor indican que surgió alguna condición excepcional, entonces se inician acciones para llamar la atención del operador hacia dicho valor y, en ciertos casos, realizar acciones en respuesta al valor excepcional.
Estímulos	Valores de los sensores unidos al sistema.
Respuestas	Salidas a desplegar, activadores de alarma, señales a sistemas que reaccionan.
Procesos	Observador, Análisis, Despliegue, Alarma, Reactor.
Usado en	Sistemas de monitorización, sistemas de alarma.



PATRON OBSERVAR Y REACCIONAR

SISTEMA DE ALARMA

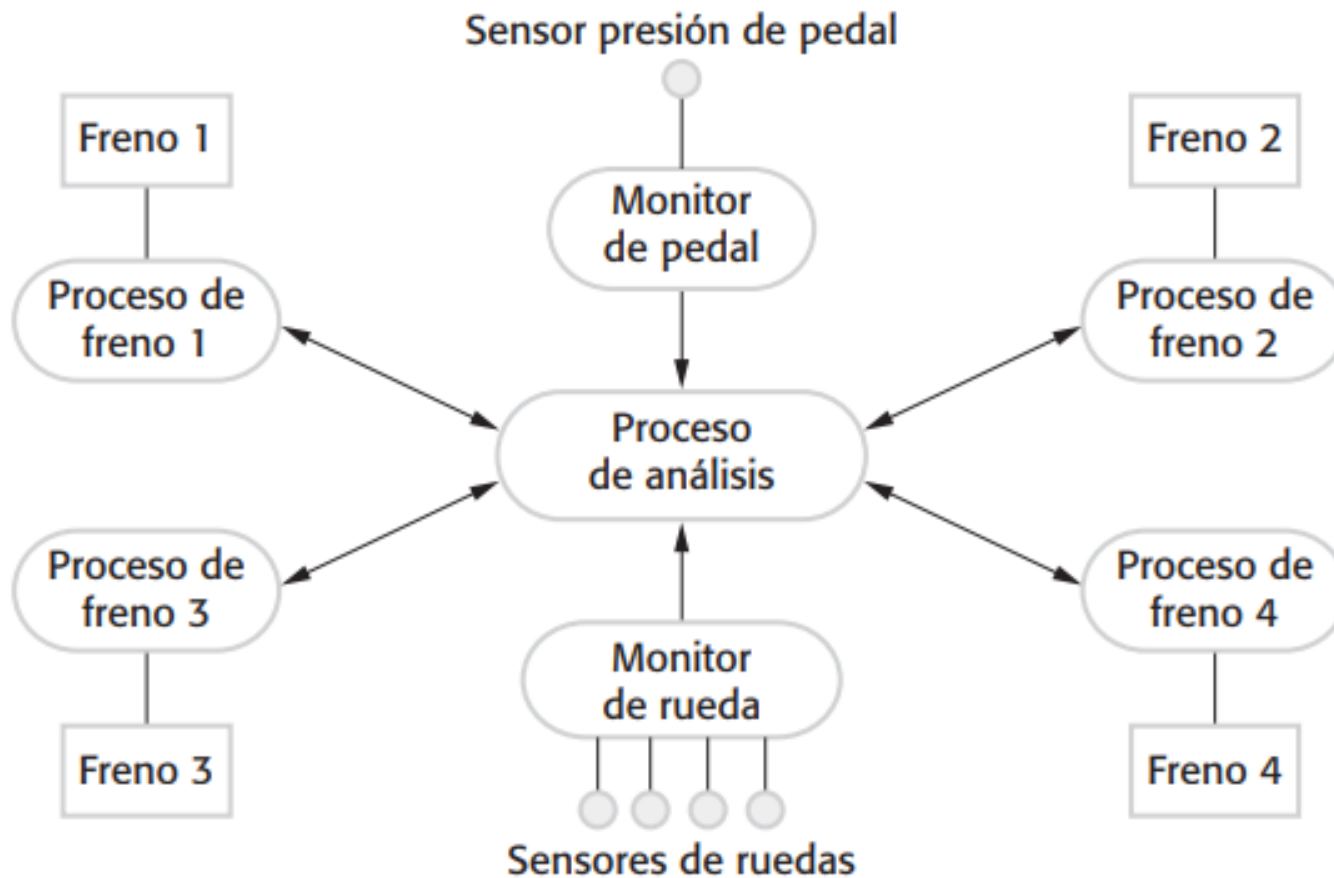


PATRON CONTROL AMBIENTAL

Nombre	Control Ambiental
Descripción	El sistema analiza información de un conjunto de sensores que recopilan datos del entorno del sistema. También se puede recopilar más información del estado de los actuadores que se conectan al sistema. Con base en datos de los sensores y actuadores, se envían señales de control a los actuadores, que en ese momento provocan cambios al entorno del sistema. Puede desplegarse información de los valores del sensor y el estado de los actuadores.
Estímulos	Valores de los sensores unidos al sistema y el estado de los actuadores del sistema.
Respuestas	Señales de control a actuadores, despliegue de información.
Procesos	Monitor, de control, de despliegue, controlador de actuador, monitor de actuador.
Usado en	Sistemas de control.



CONTROL AMBIENTAL SISTEMA DE FRENADO

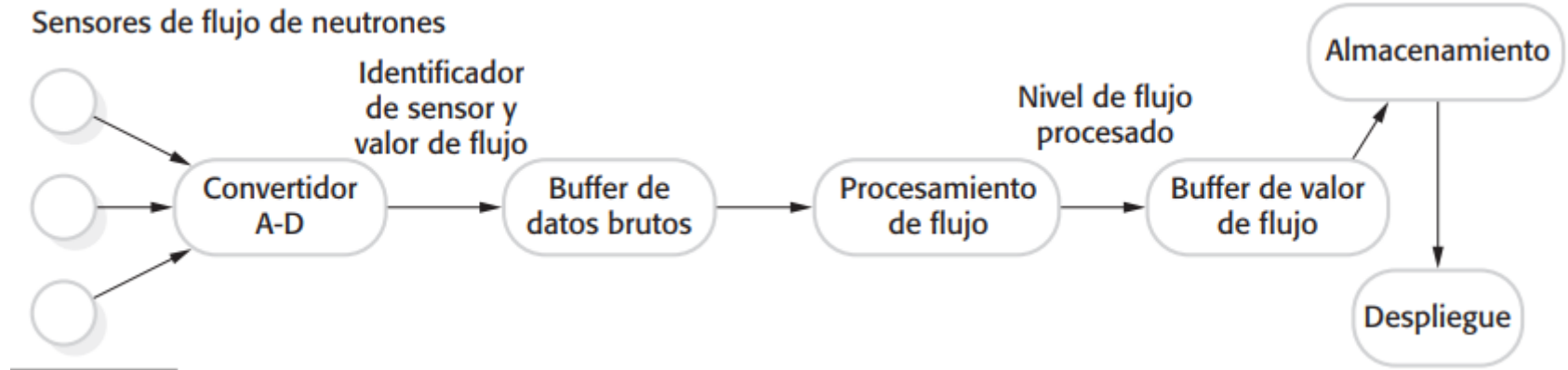
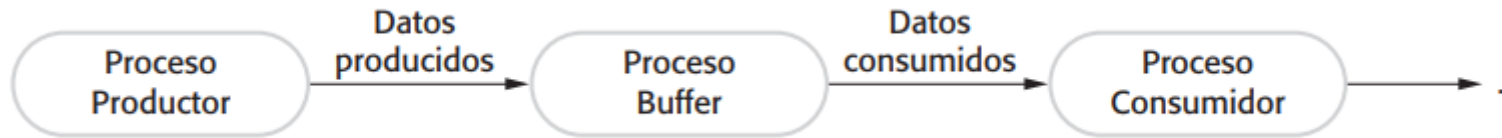


PATRON PIPELINE

Nombre	Segmentación de proceso
Descripción	Una segmentación (<i>pipeline</i>) de procesos se establece con datos que se mueven en secuencia de un extremo de la “tubería” a otro. Con frecuencia, los procesos están vinculados mediante buffers sincronizados para permitir que los procesos productor y consumidor se ejecuten a diferentes velocidades. La culminación de una segmentación puede desplegarse o almacenar los datos, o la “tubería” puede terminar en un actuador.
Estímulos	Valores de entrada del entorno o algún otro proceso
Respuestas	Valores de salida al entorno o un buffer compartido
Procesos	Productor, Buffer, Consumidor
Usado en	Sistemas de adquisición de datos, sistemas multimedia



PATRON PIPELINE



TEMPORIZACION

- **Plazos:** Los tiempos en que deben procesarse los estímulos y producir la salida. Si el sistema no cumple un plazo es una falla de sistema.
- **Frecuencia:** El número de veces por segundo que debe ejecutarse un proceso para tener la seguridad de que siempre puede cumplir los plazos.
- **Tiempo de ejecución:** El tiempo requerido para procesar un estímulo y producir una respuesta. Dos tiempos de ejecución: el tiempo de ejecución promedio de un proceso y el tiempo de ejecución del peor escenario para dicho proceso.

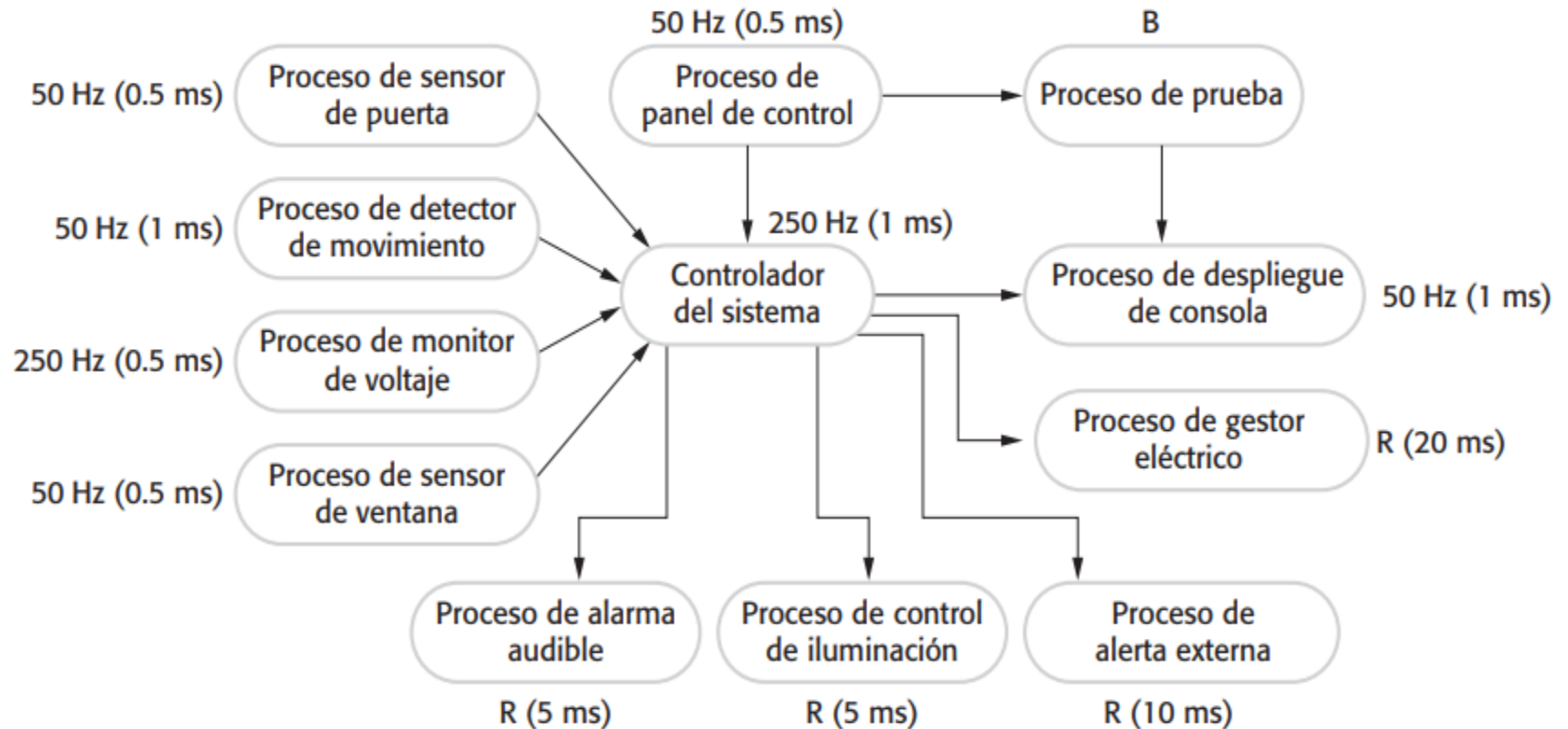


ANALISIS DE TEMPORIZACION

Estímulo/respuesta	Requerimientos de temporización
Falla de energía	El cambio a energía de respaldo debe completarse dentro de un plazo de 50 ms.
Alarma de puerta	Cada alarma de puerta debe revisarse dos veces por segundo.
Alarma de ventana	Cada alarma de ventana debe revisarse dos veces por segundo.
Detector de movimiento	Cada detector de movimiento debe revisarse dos veces por segundo.
Alarma audible	La alarma audible debe encenderse dentro del medio segundo posterior al que un sensor emite una alarma.
Encendido de luces	Las luces deben encenderse dentro del medio segundo posterior al que un sensor emite una alarma.
Comunicaciones	El llamado a la policía debe iniciarse dentro de los 2 segundos posteriores a los que un sensor emite una alarma.
Sintetizador de voz	Un mensaje sintetizado debe estar disponible dentro de los 2 segundos posteriores a los que un sensor emite una alarma.



TEMPORIZACION



SISTEMAS OPERATIVOS DE TIEMPO REAL

- Los sistemas operativos estándar, como Linux y Windows, no se usan como plataforma de ejecución para sistemas de tiempo real.
- Generalmente las aplicaciones embebidas se construyen en la parte superior de un sistema operativo de tiempo real (RTOS).
- Un RTOS es un sistema operativo eficiente que ofrece las características que necesitan los sistemas de tiempo real. (Windows/CE, Vxworks y RTLinux).

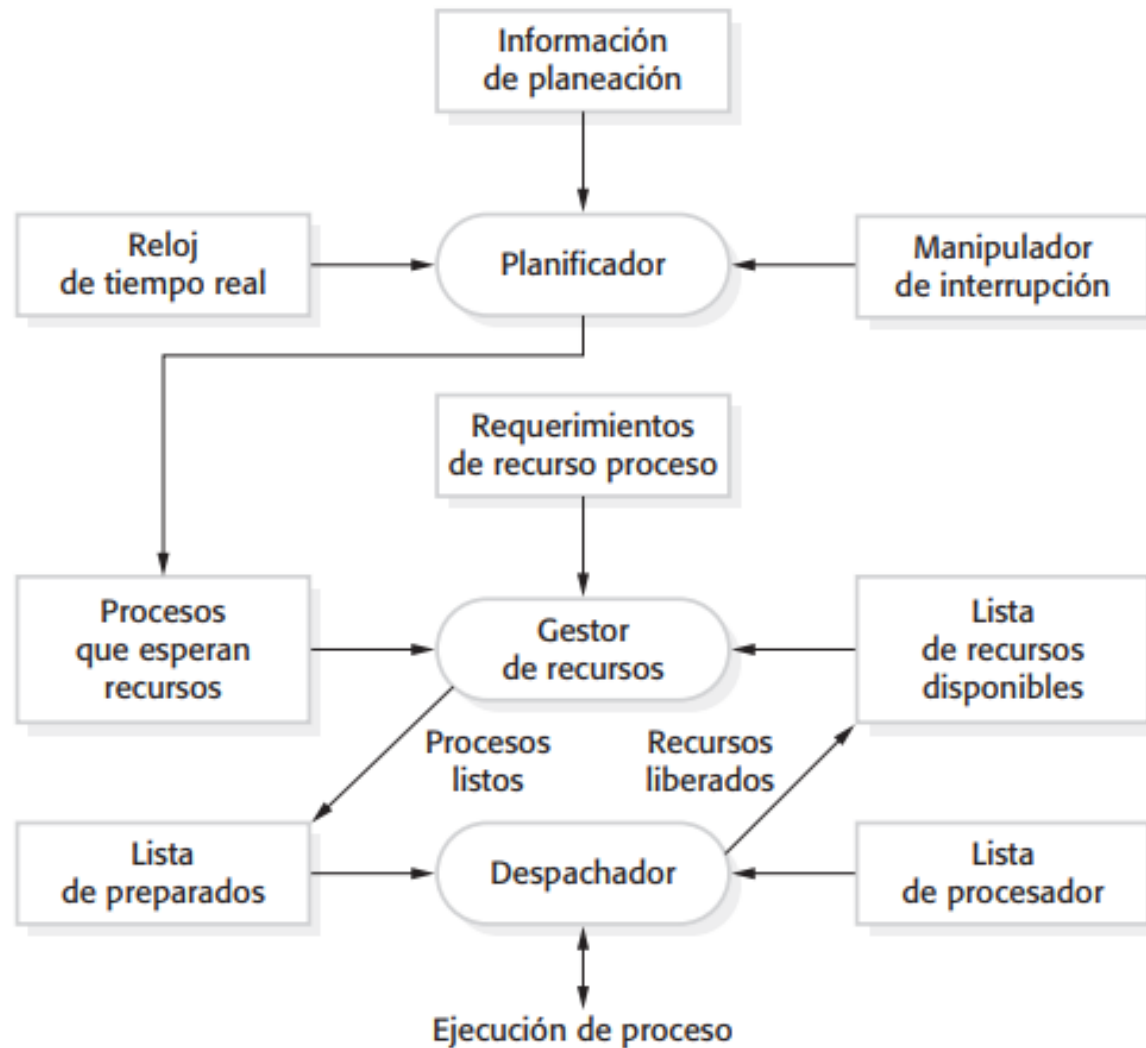


COMPONENTES DE UN RTOS

- Un reloj de tiempo real
 - Proporciona la información requerida para programar los procesos periódicamente.
- Un manipulador de interrupciones
 - Gestiona peticiones no periódicas de servicios.
- Un planificador
 - Examina los procesos que pueden ejecutarse y para elegir uno de ellos para su ejecución.
- Un gestor de recursos.
 - Asigna memoria y recursos de procesador para los procesos programados.
- Un despachador.
 - Responsable de iniciar la ejecución de los procesos



COMPONENTES DE UN RTOS



PRIORIDADES

Un RTOS debe gestionar al menos dos niveles de prioridad para los procesos del sistema:

- **Nivel de interrupción:** Este es el nivel de prioridad más alto. Se asigna a procesos que necesitan una respuesta muy rápida.
- **Nivel de reloj:** Este nivel de prioridad se asigna a los procesos periódicos.
- **Prioridad de procesos en segundo plano:** Procesos que no necesitan cumplir plazos en tiempo real. Se programan para ejecutarse cuando la capacidad de procesador está disponible. (autoverificación)



SERVICIO DE INTERRUPCIONES

- El control se transfiere automáticamente a una dirección de memoria definida.
- Esta dirección de memoria contiene el código de respuesta a la interrupción (interrupt service routine).
- Se deshabilitan las restantes interrupciones hasta finalizar el proceso.
- Interrupt service routines debe ser simple y veloz.



MANEJO DE LOS PROCESOS

- El RTOS debe manejar como se ejecutan procesos concurrente:
- Los procesos periódicos de ejecutan a intervalos regulares.
- El planificador selecciona un proceso a ejecutar de la lista.
- Se selecciona según la prioridad, periodos del proceso, tiempos de ejecución esperados y plazos de finalización.
- Si dos procesos deben ejecutarse en el mismo instante se retrasa el proceso con plazo más largo.



