

Conceptos avanzados de Programación Orientada a Objetos aplicados a la implementación de Sistemas de Tiempo Real utilizando plataformas móviles Android y iOS

**Carlos Matias Tripode**

Universidad Nacional del Sur

Departamento de Ingeniería Eléctrica y Computadoras

Bahía Blanca, Argentina

2014

Conceptos avanzados de Programación Orientada a Objetos aplicados a la implementación de Sistemas de Tiempo Real utilizando plataformas móviles Android y iOS

**Carlos Matias Tripode**

Tesis de investigación presentada como requisito parcial para optar al título de:

**Magister en Ingeniería**

Director:

Dr Rodrigo Santos

Codirector:

Dr Javier Orozco

Línea de Investigación:

Sistemas de Tiempo Real

Grupo de Investigación:

Laboratorio de Sistemas Digitales

Universidad Nacional del Sur

Departamento de Ingeniería Eléctrica y de Computadoras

Bahía Blanca, Argentina

2014

*(Dedicatoria o lema)*

*Su uso es opcional y cada autor podrá determinar la distribución del texto en la página, se sugiere esta presentación. En ella el autor del trabajo dedica su trabajo en forma especial a personas y/o entidades.*

*Por ejemplo:*

*A mis padres*

*o*

*La preocupación por el hombre y su destino siempre debe ser el interés primordial de todo esfuerzo técnico. Nunca olvides esto entre tus diagramas y ecuaciones.*

*Albert Einstein*

**Agradecimientos**

Esta sección es opcional, en ella el autor agradece a las personas o instituciones que colaboraron en la realización del trabajo. Si se incluye esta sección, deben aparecer los nombres completos, los cargos y su aporte al trabajo.

A continuación se presenta la tabla de contenido la cual se actualiza automáticamente. Para los textos editados en Microsoft Word se debe hacer *click* en el botón derecho del mouse sobre la tabla de contenido y aparecerá el icono Actualizar Campos (), luego aparecerá una ventana en la cual debe seleccionar la opción Actualizar toda la tabla.

Resumen

El resumen es una presentación abreviada y precisa (la NTC 1486 de 2008 recomienda revisar la norma ISO 214 de 1976). Se debe usar una extensión máxima de 12 renglones. Se recomienda que este resumen sea analítico, es decir, que sea completo, con información cuantitativa y cualitativa, generalmente incluyendo los siguientes aspectos: objetivos, diseño, lugar y circunstancias, pacientes (u objetivo del estudio), intervención, mediciones y principales resultados, y conclusiones. Al final del resumen se deben usar palabras claves tomadas del texto (mínimo 3 y máximo 7 palabras), las cuales permiten la recuperación de la información.

**Palabras clave:).** Sistemas de Tiempo Real, aplicaciones móviles, Android, iOS, programación orientada a objetos.

**Abstract**

Es el mismo resumen pero traducido al inglés. Se debe usar una extensión máxima de 12 renglones. Al final del Abstract se deben traducir las anteriores palabras claves tomadas del texto (mínimo 3 y máximo 7 palabras), llamadas keywords. Es posible incluir el resumen en otro idioma diferente al español o al inglés, si se considera como importante dentro del tema tratado en la investigación, por ejemplo: un trabajo dedicado a problemas lingüísticos del mandarín seguramente estaría mejor con un resumen en mandarín.

**Keywords:** Real-time systems, mobile applications, Android, iOS, object oriented programing**.**

**Contenido**

Pág.

Resumen IX

Lista de figuras XII

Lista de tablas XIII

Lista de Símbolos y abreviaturas XIV

Introducción 1

1. Sistema de Tiempo Real (STR) 5

2. Patrones de diseño y principios SOLID 15

3. Patrones de disenio para Sistemas de Tiempo Real 19

4. Patrones Arquitecturales para Sistemas de Tiempo Real 23

5. Plataforma móvil Android 23

6. Plataforma móvil iOS 23

7. Conclusiones y recomendaciones 27

Bibliografía 39

Lista de figuras

Pág.

**Figura 1‑1:** Ejecución de una tarea de tiempo real 5

Nota: si es necesario es posible incluir una lista de ilustraciones y/o fotografías, en caso que se utilicen en el desarrollo de la tesis o trabajo de investigación.

Lista de tablas

Pág.

**Tabla 2‑1**: Participación de las energías renovables primaria [14]. 17

Nota: si es necesario es posible incluir una lista de cuadros, en caso que se utilicen en el desarrollo de la tesis o trabajo de investigación.

Lista de Símbolos y abreviaturas

**Esta sección es opcional, dado que existen disciplinas que no manejan símbolos y/o abreviaturas. Se incluyen símbolos generales (con letras latinas y griegas), subíndices, superíndices y abreviaturas (incluir sólo las clases de símbolos que se utilicen). Cada una de estas listas debe estar ubicada en orden alfabético de acuerdo con la primera letra del símbolo (en esta plantilla, el título del tipo de símbolo esta en letra Arial de 14 puntos y en negrilla). Para escribir la definición en las tablas, se puede usar la herramienta de referencia cruzada (para textos editados en Microsoft Word). A continuación se presentan algunos ejemplos.**

**Símbolos con letras latinas**

| **Símbolo** | **Término** | **Unidad SI** | **Definición** |
| --- | --- | --- | --- |
| *A* | Área | m2 |  |
| *ABET* | Área interna del sólido |  | ver DIN ISO 9277 |
| *Ag* | Área transversal de la fase gaseosa | m2 | Ec. 3.2 |
| *As* | Área transversal de la carga a granel | m2 | Ec. 3.6 |
| *a* | Coeficiente | 1 | Tabla 3-1 |
| *a* | Contenido de ceniza | 1 |  |
| *c* | Contenido de carbono | 1 |  |
| *c* | Longitud de la cuerda | m | Figura 3-4 |
| *c* | Concentración de la cantidad de materia |  |  |
| *D* | Diámetro | m |  |
| *EA* | Energía de activación |  | Ec. 3.7 |
| *F* | Fracción de material volátil | 1 | ver DIN 51720 |
| *Fr* | Número de Froude | 1 |  |
|  | Aceleración de la gravedad |  |  |
| *H* | Entalpía | J | *U + PV* |
| *H0* | Poder calorífico superior |  | ver DIN 51857 |
| *h* | Coeficiente de hidrógeno | 1 |  |
| *K* | Coeficiente de equilibrio | 1 | Ec. 2.5 |
| *L* | Longitud | m | *DF* |
| *L* | Longitud del reactor | m | Figura 5-4 |
| *Q* | Calor | kJ | 1. *LT* |
| *T* | Temperatura | K | *DF* |
| *t* | tiempo | s | *DF* |
| *xi* | Fracción de la cantidad de materia | 1 |  |
| *V* | Volumen | m3 |  |
|  | Velocidad |  |  |
| *wi* | Fracción en masa del componente i | 1 |  |
| *WW,i* | Contenido de humedad de la sustancia i | 1 |  |
| *Z* | Factor de gases reales | 1 |  |

**Símbolos con letras griegas**

| **Símbolo** | **Término** | **Unidad SI** | **Definición** |
| --- | --- | --- | --- |
| *α* | Factor de superficie |  | (*w*F,waf)(*A*BET) |
| *β* | Grado de formación del componente i | 1 |  |
| ** | Wandhafreibwinkel (Stahlblech) | 1 | Sección 3.2 |
|  | Porosidad de la partícula | 1 |  |
| *η* | mittlere Bettneigunswinkel (Stürzen) | 1 | Figura 3-1 |
| *θ* | Ángulo de inclinación | 1 | Figura 4-1 |
| ** | Velocidad de calentamiento |  |  |
| ** | Tiempo adimensional | 1 | Ec. 2.6 |
| *Φ*V | Flujo volumétrico |  |  |

**Subíndices**

| **Subíndice** | **Término** |
| --- | --- |
| bm | Materia orgánica |
| DR | Dubinin-Radushkevich |
| E | Experimental |
| g | Fase gaseosa |
| k | Condensado |
| Ma | Macroporos |
| P | Partícula |
| p | pirolizado |
| R | Reacción |
| t | Total |
| wf | Libre de agua |
| waf | Libre de agua y ceniza |
| 0 | Estado de referencia |

**Superíndices**

| **Superíndice** | **Término** |
| --- | --- |
| n | Exponente, potencia |

**Abreviaturas**

| **Abreviatura** | **Término** |
| --- | --- |
| 1.*LT* | Primera ley de la termodinámica |
| *DF* | Dimension fundamental |
| *RFF* | Racimos de fruta fresca |

Introducción

**En la introducción, el autor presenta y señala la importancia, el origen (los antecedentes teóricos y prácticos), los objetivos, los alcances, las limitaciones, la metodología empleada, el significado que el estudio tiene en el avance del campo respectivo y su aplicación en el área investigada. No debe confundirse con el resumen y se recomienda que la introducción tenga una extensión de mínimo 2 páginas y máximo de 4 páginas.**

La presente plantilla maneja la fuente Arial para el texto de los párrafos y para los títulos y subtítulos. Sin embargo, es posible sugerir otras fuentes tales como Garomond, Calibri, Cambria o Times New Roman, que por claridad y forma, son adecuadas para la edición de textos académicos.

Esta sección se **encabeza con la palabra introducción, escrita con minúscula (en la primera línea), con un espaciado anterior de 100 puntos y posterior de 24 puntos, interlineado sencillo y en letra negrilla de 20 puntos (en este caso se usa Arial).**

La presente plantilla tiene en cuenta aspectos importantes de la Norma Técnica Colombiana - NTC 1486, con el fin que sea usada para la presentación final de las tesis de maestría y doctorado y especializaciones y especialidades en el área de la salud, desarrolladas en la Universidad Nacional de Colombia.

Las márgenes deben ser de 2,54 centímetros (1 pulgada) en la parte superior, inferior y exterior y de 3,6 centímetros en la margen interna (a partir de márgenes simétricos). La plantilla esta diseñada para imprimir por lado y lado en hojas tamaño carta. Se sugiere que los encabezados cambien según la sección del documento (para lo cual esta plantilla esta construida por secciones). El número de página se ubica en la parte superior derecha en las páginas impares y en la superior izquierda en las páginas pares (en letra Arial de 11 puntos, de acuerdo al formato presentado en esta plantilla). El título de cada capítulo debe estar numerado y comenzar en una hoja independiente (página impar) y con el mismo formato del título Introducción (**escrita con minúscula, en la primera línea, con un espaciado anterior de 100 puntos y posterior de 24 puntos e interlineado sencillo y en letra de 20 puntos y negrilla; en este caso se usa Arial).** El texto debe llegar hasta la margen inferior establecida. Se debe evitar títulos o subtítulos solos al final de la página o renglones sueltos.

Si se requiere ampliar la información sobre normas adicionales para la escritura se puede consultar la norma NTC 1486 en la Base de datos del ICONTEC (Normas Técnicas Colombianas) disponible en el portal del SINAB de la Universidad Nacional de Colombia [www.sinab.unal.edu.co](http://www.sinab.unal.edu.co), en la sección “Recursos bibliográficos” opción “Bases de datos”. Este portal también brinda la posibilidad de acceder a un instructivo para la utilización de Microsoft Word y Acrobat Professional, el cual está disponible en la sección “Servicios”, opción “Trámites” y enlace “Entrega de tesis”

La tesis o trabajo de investigación se debe escribir con interlineado sencillo y después de punto aparte a dos interlíneas sencillas (dos veces la tecla Enter). La redacción debe ser impersonal y genérica. La numeración de las hojas sugiere que las páginas preliminares se realicen en números romanos en mayúscula y las demás en números arábigos, en forma consecutiva a partir de la introducción que comenzará con el número 1. La cubierta y la portada no se numeran pero si se cuentan como páginas.

El tamaño de letra sugerido y teniendo en cuenta la familia fuente Arial de 11 puntos para el texto de estilo “Párrafo”, Arial para los títulos, de 20 puntos (estilo “Título Primer nivel”) y de 16 y 14 para los subtítulos (estilos “Título segundo nivel” y “Título tercer nivel”, respectivamente).

Para trabajos muy extensos se recomienda publicar más de un volumen. Se debe tener en cuenta que algunas facultades tienen reglamentada la extensión máxima de las tesis o trabajos de investigación; en caso que no sea así, se sugiere que el documento no supere 120 páginas.

**No se debe utilizar numeración compuesta como 13A, 14B ó 17 bis, entre otros, que indican superposición de texto en el documento. Para resaltar, puede usarse letra cursiva o negrilla. Los términos de otras lenguas que aparezcan dentro del texto se escriben en cursiva.**

# 

# Sistema de Tiempo Real (STR)

Un sistema de Tiempo Real es un sistema informático que:

* Interacciona con su entorno físico
* Responde a los estímulos del entorno dentro de un plazo de tiempo determinado

No basta con que las acciones del sistema sean correctas, sino que, además, tienen que ejecutarse dentro de un intervalo de tiempo determinado. Los sistemas tiempo real suelen estar integrados en un sistema de ingeniería más general, en el que realizan funciones de control y/o monitorización.

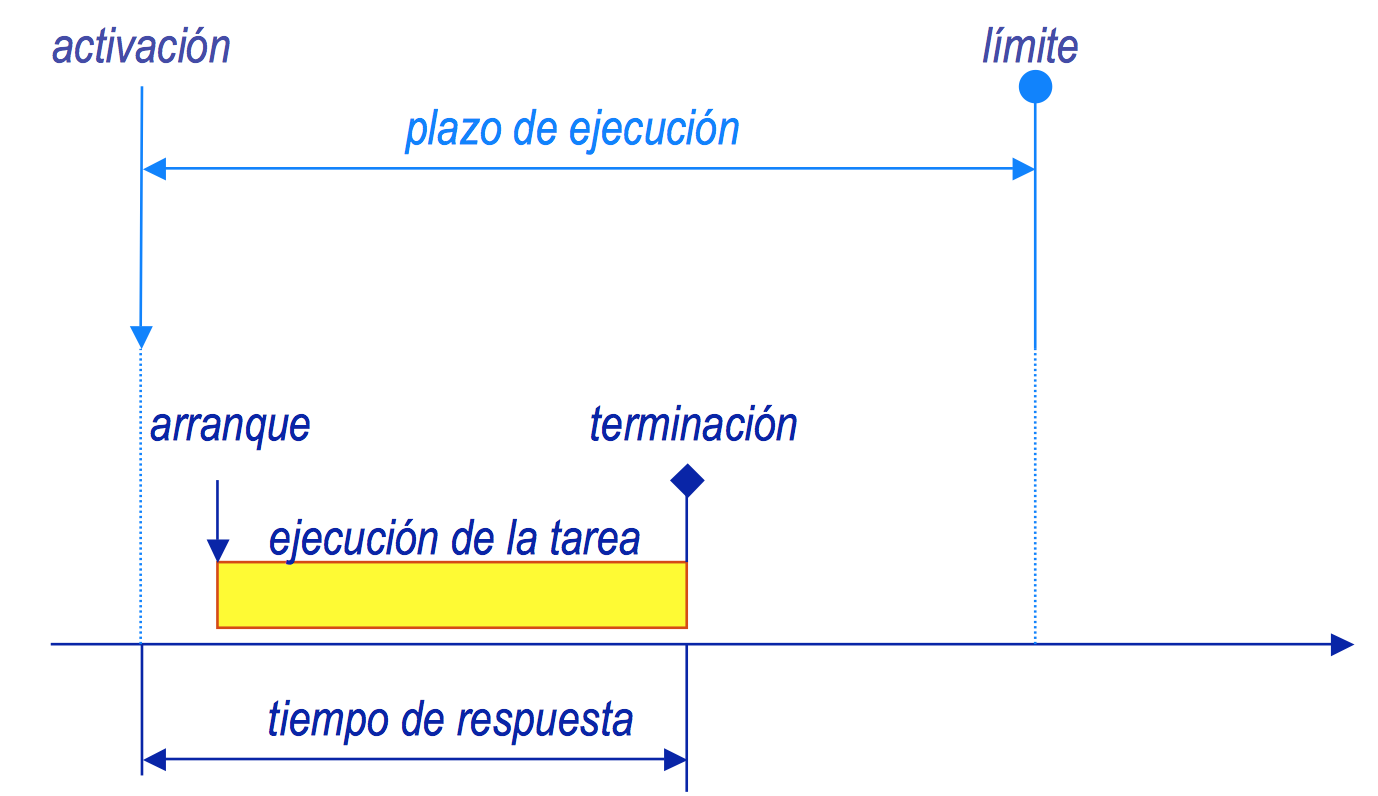
**Los STR se clasifican según sus requerimientos temporales en:**

* Tiempo real estricto *(hard real-time):* todas las acciones deben terminar dentro del plazo especificado. Ejemplo: control de frenado
* Tiempo real flexible *(soft real-time):* se pueden perder plazos de vez en cuando. El valor de la respuesta decrece con el tiempo. Ejemplo: adquisición de datos
* Tiempo real firme *(firm real-time):* se pueden perder plazos ocasionalmente – una respuesta tardía no tiene valor. Ejemplo: sistemas multimedia

Las actividades de un sistema de tiempo real se llaman *tareas.* El comportamiento temporal de las tareas se especifica mediante sus atributos temporales: cuándo se ejecutan (esquema de activación) y qué plazo tienen para ejecutar cada acción (tiempo de respuesta). Los esquemas de activación clasifican las actividades en: periódicas (a intervalos regulares, con período T), aperiódicas (cada vez que ocurre un suceso determinado) y dentro de esta ultima se pueden subclaficar como esporádicas (separación mínima entre activaciones T) y estocásticas (activaciones irregulares)

Los tiempos de respuesta se clasifican en: absoluto (tiempo límite para terminar) y relativo (intervalo desde la activación)

**Figura 1‑1:** Ejecución de una tarea de tiempo real



## Caracteristicas de un STR

* Gran tamaño y complejidad: algunos STR tienen millones de líneas de código. La variedad de funciones aumenta la complejidad incluso en sistemas relativamente pequeños.
* Simultaneidad de acciones (**concurrencia**): los dispositivos físicos controlados funcionan al mismo tiempo (las tareas que los controlan actúan concurrentemente)
* **Dispositivos de entrada y salida** especiales: los manejadores de dispositivos forman parte del software de aplicación
* **Seguridad y confiabilidad**: sistemas críticos (fallos con consecuencias graves como pérdida de vidas humanas, pérdidas económicas, daños medioambientales)
* **Confiabilidad**: Es la probabilidad de proporcionar el servicio especificado. Se mide en: tiempo medio hasta la falla (MTTF). Los sistemas con MTTF > 10 se denominan sistemas ultraconfiables
* **Seguridad**: Existen fallas malignas y benignas. Una falla maligna es aquella que tiene un costo muy superior a la utilidad del sistema.
* **Sistemas críticos**: Estos deben ser ultra confiables respecto a las fallas malignas. En muchos casos se exige una certificación de seguridad efectuada por un organismo independiente
* **Mantenibilidad**: se define como el tiempo medio de reparación (MTTR)  
  Es el tiempo necesario para reparar una fallaa benigna.
* **Disponibilidad**: se define como la fracción de tiempo durante el cual el sistema está disponible
* **Determinismo temporal**: acciones en intervalos de tiempo determinados. Es fundamental que el comportamiento temporal de los STR sea determinista o, al menos, previsible
  + no hay que confundirlo con la necesidad de que sea eficiente
  + el sistema debe responder correctamente en todas las situaciones
  + en los sistemas de tiempo real estricto hay que prever el comportamiento en el peor caso posible
* Las tareas de tiempo real se ejecutan concurrentemente – hilos (threads) o mecanismos similares
* La planificación del uso del procesador debe permitir acotar el tiempo de respuesta: prioridades y otros métodos de planificación
* Es conveniente analizar el comportamiento temporal del sistema antes de probarlo. Las pruebas (tests) no siempre permiten asegurar el comportamiento en el peor caso

## Sistemas embebidos

Toda división o capítulo, a su vez, puede subdividirse en otros niveles y sólo se enumera hasta el tercer nivel. Los títulos de segundo nivel se escriben con minúscula al margen izquierdo y sin punto final, están separados del texto o contenido por **un interlineado posterior de 10 puntos y anterior de 20 puntos (tal y como se presenta en la plantilla)**.

## Sistemas operativos de tiempo real (SOTR)

Un sistema operativo para tiempo real (SOTR) es un sistema operativo (SO) capaz de garantizar los requisitos temporales de los procesos que controla. Los SO convencionales no son apropiados para la realización de sistemas de tiempo real, debido a que

– no tienen un comportamiento determinista y

– no permiten garantizar los tiempos de respuesta.

Existen algunas características de los sistemas operativos convencionales que impiden su uso como RTOS (*Real-Time Operating System*). Lo veremos en el caso particular de un sistema operativo UNIX. Dichas restricciones se listan a continuación.

**Planificación para tiempo compartido**: Uso de planificadores que aseguran un uso equitativo del tiempo de CPU entre todos los procesos. Es conveniente para un usuario que usa el sistema desde una terminal. No para procesamiento de tiempo real, ya que la ejecución de cualquier proceso depende de forma compleja e impredecible de la carga del sistema y el comportamiento del resto de procesos.

**Baja resolución del temporizador**: Históricamente a los procesos de usuario se les proporcionaban señales de alarma y la llamada al sistema *sleep()* con sólo 1 segundo de resolución ⇒ No es suficiente para procesamiento de tiempo real. Las versiones más modernas proporcionan medios de especificar intervalos con mayor precisión.

**Núcleo no desalojable**: Los procesos que se ejecutan en modo núcleo no pueden ser desalojados. Una llamada al sistema podría tardar demasiado tiempo para poder admitirlo en procesamiento de tiempo real.

**Deshabilitación de interrupciones**: Muy cercano al anterior está el problema de la sincronización. Para proteger los datos que podrían ser accedidos asíncronamente, algunos diseñadores optan por inhibir las interrupciones durante las secciones críticas ⇒ más eficiente que los semáforos. Sin embargo, pone en peligro la posibilidad de responder a eventos externos de forma adecuada.

**Memoria Virtual**: En STR introduce un nivel de impredecibilidad intolerable.

**Soluciones:**

POSIX.1b-1993:

• Estándar para introducir características de tiempo real en UNIX.

• Define planificación con prioridades, bloqueo de páginas de memoria del usuario en memoria, señales para tiempo real, IPC, timers

QNX:

• Cambiar el planificador *round-robin* de tiempo compartido por uno basado en prioridades.

• No hay problemas con la paginación ni el *swapping* (no se usan).

• Es una solución aceptable para sistemas sencillos. 2. POSIX.1b-1993:

• Estándar para introducir características de tiempo real en UNIX.

• Define planificación con prioridades, bloqueo de páginas de memoria del usuario en memoria, señales para tiempo real, IPC, timers . . .

• Cumple el estándar POSIX.1b.

• Arquitectura de microkernel ⇒ el núcleo implementa sólo cuatro servicios: planificación de procesos, comunicación entre procesos, comunicación de red de bajo nivel y manejo de interrupciones.

• El resto de servicios se implementan como procesos de usuario.

### Ejemplo: Real-Time Linux (RTL)

Proporciona un núcleo de tiempo real basado en un planificador con desalojo y prioridades fijas. Está preparado para la gestión de tareas críticas. Es una modificación de código de Linux para gestionar tareas críticas. Está disponible la versión 1.1 para la versión 2.0.36 del kernel de Linux. Por ahora sólo está disponible para la arquitectura del PC. Características:

* basados en Linux.
* Incluye un planificador con desalojo y prioridades fijas, para la ejecución de tareas críticas de tiempo real.
* Las tareas pueden ser periódicas o bien activadas mediante una interrupción (esporádicas o aperiódicas).
* Incorpora mecanismos para la comunicación con los procesos no críticos, que son los de Linux *normal*. Estos mecanismos son colas FIFO.
* Las tareas de tiempo real se ejecutan con la CPU en modo *supervisor* (pueden acceder a puertos E/S, reprogramar interrupciones, etc...).
* Convierte al núcleo de Linux en una tarea más, pero de segundo plano (de prioridad mínima).
* RT-Linux es como un microkernel que realiza operaciones muy básicas: gestionar todas las interrupciones y planificar las tareas de tiempo real.
* Linux deja de disponer de las instrucciones *cli*, *sti* e *iret* para tener en su lugar llamadas al propio microkernel de RT-Linux, que las simula.
* Linux pierde el control del sistema y no se ejecutará si las tareas críticas ocupan toda la CPU ⇒ El sistema puede bloquearse aparentemente.

Las tareas de tiempo real en RT-Linux son código que se ejecuta en modo supervisor de la CPU, para tener acceso directo a los dispositivos de E/S.

• Para ejecutar procesos en modo supervisor no basta con hacerlo desde la cuenta *root*, sino que además es necesario que sea parte del código del núcleo.

* El código del núcleo no se pagina ⇒ un tarea no puede ser expulsada a disco.
* Es necesario programar las taras como módulos de carga (el propio RT-Linux

se ha hecho como un módulo).

• Un módulo de carga es un programa que podemos realizar en C en el que:

– Carece de función *main*.

– Tiene una función para iniciar el módulo *init\_module()* ⇒ se ejecuta al cargarlo y desde ella llamaremos a otras.

– Tiene una función para finalizar el módulo *cleanup\_module()* ⇒ se ejecuta al descargar el módulo.

• El núcleo no dispone de salida estándar, por lo que no podremos usar las funciones de E/S por pantalla habituales, en su lugar usaremos *printk*.

## Algoritmos de planificación

Se trata de repartir el tiempo de procesador entre varias tareas de forma que se satisfagan los requisitos temporales. La relación biunívoca entre acciones y procesadores es un plan de ejecución (schedule). El componente del sistema que hace esto es el planificador (scheduler). Para ello utiliza un algoritmo de planificación

Planificación dirigida por tiempo *(time/clock-driven). El* planificador se ejecuta cada vez que llega una señal de reloj. Ejemplo: planificación cíclica

Planificación por turno circular *(round-robin).* Las acciones listas para ejecutarse se agrupan en una cola FIFO. Cada acción se ejecuta durante una rodaja de tiempo y después se pone al final de la cola

Planificación por prioridades. Cada acción tiene una prioridad y se ejecuta siempre la acción de mayor prioridad entre las listas. La planificación está dirigida por sucesos *(event-driven)*

Planificación con desalojo (preemptive scheduling) – se puede desalojar del procesador una acción que se está ejecutando para dar paso a otra. Se usa normalmente con prioridades

Planificación sin desalojo (non-preemptive scheduling) – una acción que ha comenzado a ejecutarse sólo deja el procesador si termina su ejecución. Necesita un recurso que no está disponible » abandona el procesador voluntariamente

### Prioridades fijas

Es el método más corriente en sistemas operativos de tiempo real. La prioridad de las acciones de una misma tarea es siempre la misma.  
Puede variar si hay cambios de modo. Ejemplo: prioridades monótonas en frecuencia (rate-monotonic scheduling) mayor prioridad a la tarea con período más corto

Cada tarea tiene una prioridad fija – planificación estática

Las tareas ejecutables se despachan para su ejecución en orden de prioridad

El despacho puede hacerse – con desalojo  
– sin desalojo  
– con desalojo limitado

En general supondremos prioridades fijas con desalojo – mejor tiempo de respuesta para las tareas de alta prioridad

Cuando se diseña un sistema planificado con prioridades fijas hay dos problemas:

– cómo asignar prioridades a las tareas  
– cómo analizar el sistema para ver si se garantizan los requisitos temporales

Parámetros de planificación

N Número de tareas  
T Período de activación

C Tiempo de ejecución máximo

D Plazo de respuesta  
R Tiempo de respuesta máximo

P Prioridad

Prioridades monótonas en frecuencia

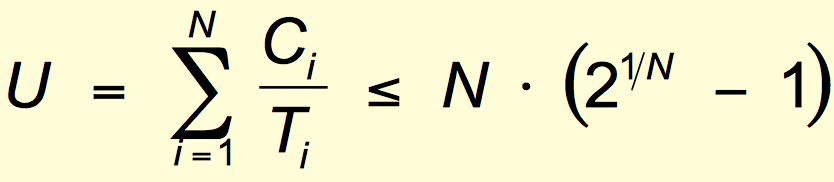
La asignación de mayor prioridad a las tareas de menor período (rate monotonic scheduling) es óptima para un modelo de tareas con

– tareas periódicas,  
– independientes,  
– con plazos iguales a los períodos

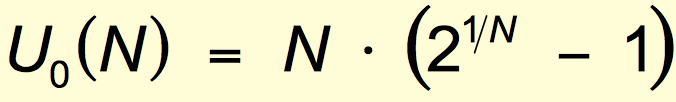
“Si se pueden garantizar los plazos de un sistema de tareas con otra asignación de prioridades, se pueden garantizar con la asignación monótona en frecuencia“ (Liu & Layland, 1973)

Condición de garantía de los plazos basada en la utilización

Para este modelo de tareas, con prioridades monótonas en frecuencia, los plazos están garantizados si



utilización mínima garantizada para N tareas



### Prioridades dinámicas

Hay sistemas de tiempo real en los que no se conocen de antemano los tiempos de cómputo ni los esquemas de activación

– las tareas se crean y se destruyen durante la ejecución – hay que hacer el análisis sobre la marcha

El modelo estático que se utiliza para analizar los sistemas planificados con prioridades fijas ya no es adecuado

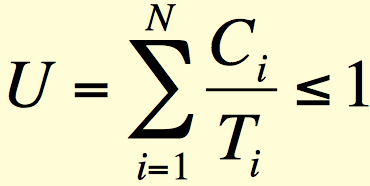
Hay que utilizar métodos de planificación con asignación dinámica de prioridades

**Primero el más urgente (EDF)**

El método de primero el más urgente (EDF) o Earliest Deadline First da mayor prioridad en cada instante a la tarea cuyo plazo absoluto (tiempo límite) vence antes. La primera tarea que se ejecuta es la más urgente – la que tiene que terminar antes. Los tiempos límites se calculan durante la ejecución – planificación dinámica

**Factor de utilización con EDF**

Los plazos están garantizados si y sólo si



Sirve para tareas periódicas y esporádicas (independientes y con Di = Ti). No hace falta calcular el tiempo de respuesta

### Reparto proporcional

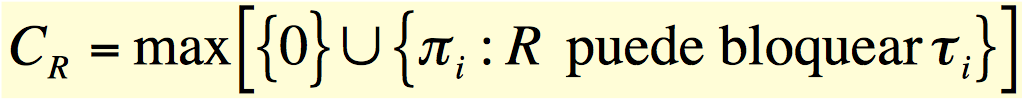
## Protocolos de sincronización

### Protocolos de sincronización para prioridades fijas

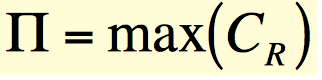
### Protocolos de sincronización para prioridades dinámicas

El protocolo de la pila de recursos (*stack resource policy, SRP*) es una extensión del protocolo de herencia inmediata del techo de prioridad (IPCP) (Baker, 1991). Se asigna a cada tarea un nivel de desalojo, tanto mayor cuanto menor es su plazo relativo Di. Cada recurso R tiene un techo de prioridad dinamico, Cr

El techo del sistema se define como:



Una tarea se ejecuta solo si su prioridad es la mas alta entre las tareas ejecutables



**Propiedades del protocolo SRP**

Tiene las mismas propiedades que el protocolo del techo de prioridad inmediato

– Si una tarea se bloquea, lo hace al principio de su ciclo  
– Una tarea sólo se puede bloquear una vez, como máximo, en cada ciclo  
– No hay interbloqueos

# Patrones de diseño y principios SOLID

Existen varias normas para la citación bibliográfica. Algunas áreas del conocimiento prefieren normas específicas para citar las referencias bibliográficas en el texto y escribir la lista de bibliografía al final de los documentos. Esta plantilla brinda la libertad para que el autor de la tesis utilice la norma bibliográfica común para su disciplina. Sin embargo, se solicita que la norma seleccionada se utilice con rigurosidad**, sin olvidar referenciar “todos” los elementos tomados de otras fuentes (**referencias bibliográficas, patentes consultadas, *softwar*e empleado en el manuscrito, en el tratamiento a los datos y resultados del trabajo, consultas a personas (expertos o público general), entre otros)**.**

## **Patrones GoF**

Existen algunos ejemplos para la citación bibliográfica, por ejemplo, Microsoft Word (versiones posteriores al 2006), en el menú de referencias, se cuenta con la opción de insertar citas bibliográficas utilizando la norma APA (American Psychological Association) u otras normas y con la ayuda para construir automáticamente la lista al final del documento. De la misma manera, existen administradores bibliográficos compatibles con

## Principios SOLID

**Las ilustraciones forman parte del contenido de los capítulos. Se deben colocar en la misma página en que se mencionan o en la siguiente (deben siempre mencionarse en el texto). No se debe emplear la abreviatura "No." ni el signo "#" para su numeración.**

### Unica responsabilidad (single responsability)

### Principio de abierto/cerrado (open/close principle)

### Sustitucion de Liskov (Liskov sustitution)

### Segregacion de Interfaces (interface segregation)

### Inversion de dependencias (dependency inversion)

## Ejemplos prácticos SOLID

## Ejemplo de presentación y citación de tablas y cuadros

**Para la edición de tablas, cada columna debe llevar su título; la primera palabra se debe escribir con mayúscula inicial y preferiblemente sin abreviaturas. En las tablas y cuadros, los títulos y datos se deben ubicar entre líneas horizontales y verticales cerradas.**

**La numeración de las tablas se realiza de la misma manera que las figuras o ilustraciones, a lo largo de todo el texto. Deben llevar un título breve, que concreta el contenido de la tabla; éste se debe escribir en la parte superior de la misma. Para la presentación de cuadros, se deben seguir las indicaciones dadas para las tablas.**

**Un ejemplo para la presentación y citación de tablas (citación indirecta), se presenta a continuación:**

**De esta participación aproximadamente el 60 % proviene de biomasa (Tabla 2‑1).**

****Tabla 2‑1**: Participación de las energías renovables primaria [14].**

|  |  |  |
| --- | --- | --- |
| **Región** | **Participación en el suministro de energía primaria / % (Mtoe)1** | |
| **Energías renovables** | **Participación de la biomasa** |
| **Latinoamérica** | **28,9 (140)** | **62,4 (87,4)** |
| **Colombia** | **27,7 (7,6)** | **54,4 (4,1)** |
| **Alemania** | **3,8 (13,2)** | **65,8 (8,7)** |
| **Mundial** | **13,1 (1401,0)** | **79,4 (1114,8)** |

**1 1 kg oe=10000 kcal=41,868 MJ**

**NOTA: en el caso en que el contenido de la tabla o cuadro sea muy extenso, se puede cambiar el tamaño de la letra, siempre y cuando ésta sea visible por el lector.**

# Patrones de diseño para Sistemas de Tiempo Real

Se deben incluir tantos capítulos como se requieran; sin embargo, **se recomienda que la tesis o trabajo de investigación tenga un mínimo 3 capítulos y máximo de 6 capítulos (incluyendo las conclusiones).**

## Patrón cola de Mensajes

Problema – Obtener una manera simple de solicitar a un servicio a travez de varios niveles de hilos (threads)

Solucion – Encolar en el receptor las suscesivas respuestas por orden de llegada. Cuando puede ejecutarse va a desencolarlas para poder servir

Message Queuing Pattern

• Problem – A simple way to request services be performed across thread boundaries in a thread-robust way

• Solution – Queue incoming services in the receiving thread until that thread runs – then dequeue and service

– This approach is simple and supported by most operating systems – Service responses are delayed until the target thread actually runs, so response may not be timely – No mutual exclusion problem because requests are serialized

# Patrones Arquitecturales para Sistemas de Tiempo Real

Se deben incluir tantos capítulos como se requieran; sin embargo, **se recomienda que la tesis o trabajo de investigación tenga un mínimo 3 capítulos y máximo de 6 capítulos (incluyendo las conclusiones).**

# Plataforma móvil Android

# Plataforma móvil iOS

# Conclusiones y recomendaciones

## Conclusiones

**Las conclusiones constituyen un capítulo independiente y presentan, en forma lógica, los resultados del trabajo. Las conclusiones deben ser la respuesta a los objetivos o propósitos planteados. Se deben titular con la palabra conclusiones en el mismo formato de los títulos de los capítulos anteriores (Títulos primer nivel), precedida por el numeral correspondiente (según la presente plantilla).**

**Las conclusiones deben contemplar las perspectivas de la investigación, las cuales son sugerencias, proyecciones o alternativas que se presentan para modificar, cambiar o incidir sobre una situación específica o una problemática encontrada. Pueden presentarse como un texto con características argumentativas, resultado de una reflexión acerca del trabajo de investigación.**

## Recomendaciones

Se presentan como una serie de aspectos que se podrían realizar en un futuro para emprender investigaciones similares o fortalecer la investigación realizada.

2. Anexo: Programación en Android

**Los Anexos son documentos o elementos que complementan el cuerpo del trabajo y que se relacionan, directa o indirectamente, con la investigación, tales como acetatos, cd, normas, etc. Los anexos deben ir numerados con letras y usando el estilo “Título anexos”.**

1. Anexo: Programación en iOS

**A final del documento es opcional incluir índices o glosarios. Éstos son listas detalladas y especializadas de los términos, nombres, autores, temas, etc., que aparecen en el trabajo. Sirven para facilitar su localización en el texto. Los índices pueden ser alfabéticos, cronológicos, numéricos, analíticos, entre otros. Luego de cada palabra, término, etc., se pone coma y el número de la página donde aparece esta información.**

Bibliografía

**Capitulo 1**

**[1] Giorgio C. Buttazzo. Hard Real-time Computing Systems – Predictable Scheduling Algorithms and Applications**

**[2] Alan Burns y Andy Wellings. Sistemas de Tiempo Real y Lenguajes de Programación (3era edición). http://laurel.datsi.fi.upm.es/~ssoo/STR/**

**Capitulo 2**

**Patrones de Diseño**

**[3] Design Patterns: Elements of Reusable Object-Oriented Software**

**[4] Clean Code a handbook of Agile Software Caftmanship**

**[5] Refactoring: Improving the Design of Existing Code**

**Principios SOLID**

**[6] Design Principles .**

**[7] MARQUEZ DE MELO, José “Comunicación e integración latinoamericana: El papel de ALAIC”. {En línea}. {10 julio de 2008} disponible en: (www.mty.itsem.mx/externos/alaic/texto1html).**