***Imagen que contiene Logotipo

Descripción generada automáticamente***

Alejandro Ortega Martínez

Grado en Ingeniería Informática

30/5/2021

Tema 3

Diseño

Programación en Tiempo Real

Contenido

[1. Introducción 1](#_Toc73524673)

[2. Diseño 2](#_Toc73524674)

[2.1. Sistemas Operativos 2](#_Toc73524675)

[2.2. Hardware 2](#_Toc73524676)

[2.3. Comunicaciones 2](#_Toc73524677)

[2.4. Lenguajes de Programación 2](#_Toc73524678)

[2. Método de Diseño HRT-HOOD 3](#_Toc73524679)

[2.1. Introducción 3](#_Toc73524680)

[2.2. Visión general de HRT-HOOD 3](#_Toc73524681)

[2.3. Arquitectura Lógica 4](#_Toc73524682)

[2.4. Arquitectura Física 5](#_Toc73524683)

[2.5. Codificación en ADA 5](#_Toc73524684)

# 1. Introducción

El diseño es una de las fases fundamentales en el desarrollo del sistema de tiempo real. Los métodos, las herramientas y la tecnología que se usan para construir otros tipos de sistemas no sirven para los sistemas de tiempo real:

* Sólo contemplan el tiempo de respuesta medio, no el peor.
* No garantizan los requisitos temporales.

El ciclo de vida de un sistema clásico no funciona para los STR.

* Comprobar el comportamiento temporal en las primeras fases.
* Pruebas al final.

# 2. Diseño

## 2.1. Sistemas Operativos

En los Sistemas convencionales no tienen un comportamiento determinista, no permiten garantizar los tiempos de respuesta, y algounos de ellos son poco fiables

En los STR:

* **Concurrencia:** Procesos ligeros (threads)
* **Temporizacion:** Medida de tiempos y ejecución periódica.
* **Planificación determinista:** Gestion del procesador y otros recursos.
* **Dispositivos de E/S:** Acceso a recursos de hardare e interrupciones.

## 2.2. Hardware

En un sistema operativo convencional, puede tener un comportamiento temporal imprevisible. El uso de memoria cache produce diferencias muy grandes en los tiempo, así como los mecanismos de aceleración. **El sw normalmente es independiente del hw**.

En un STR, **el sw manipula el hw**.

## 2.3. Comunicaciones

En los sistemas convencionales ocurren retardos.

En los STR, los **protocolos tienen tiempo de transmisión acotados**.

## 2.4. Lenguajes de Programación

En un Sistema Convencional **no pueden manejar el tiempo durante todo el ciclo de vida**.

En un STR tienen que soportar concurrencia, tiempo real, y acceso al hardware y a las interrupciones, así como tener un **comportamiento temporal analizable**.

C no es muy fiable, y es secuencial; y java es muy inmaduro, por lo que se ha elegido ADA porque esta diseñado exclusivamente para STR.

# 2. Método de Diseño HRT-HOOD

## 2.1. Introducción

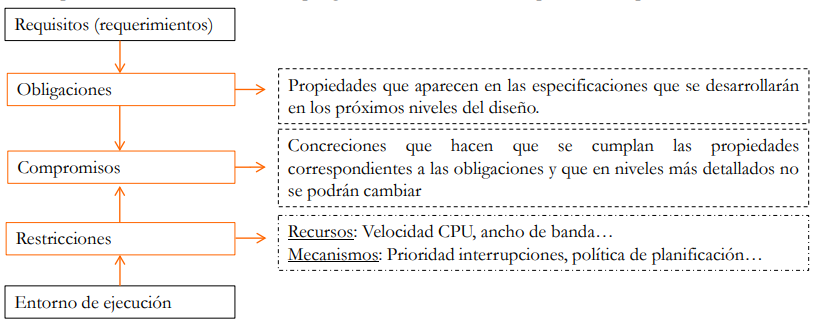
Hard Real Time Hierical Object Oriented Design.

Es un método de diseño **orientado a los STR críticos.**, el sistema se **diseña como un conjunto de objetos abstractos, que se descomponen jerárquicamente, y a los que se les asocia una serie de atributos de tiempo real**.

Es un método específico para ADA.

## 2.2. Visión general de HRT-HOOD

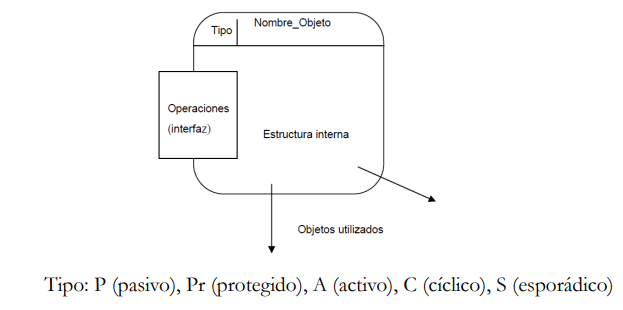
El proceso de diseño es una progresión creciente de compromisos específicos.



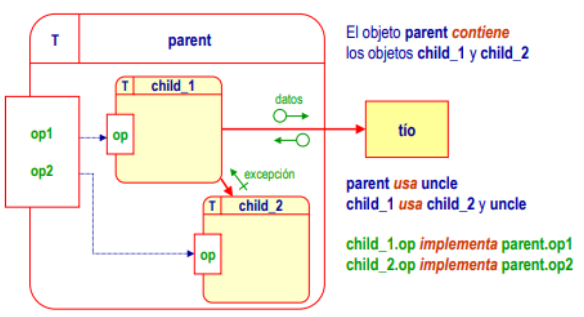
Hay dos niveles de diseño:

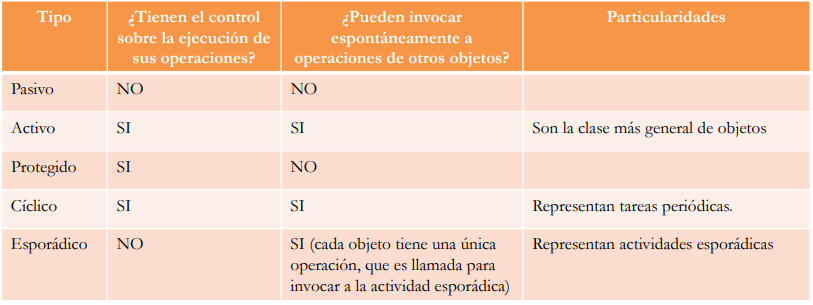
* **Modelo lógico**: Compromisos independientes del entorno.
* **Modelo físico**: Compromisos dependientes del entorno.

## 2.3. Arquitectura Lógica

Tenemos Objetos y operaciones, con sus reglas de descomposición jerárquica y uso.

Relaciones:





## 2.4. Arquitectura Física

Tiene atributos temporales (Deadline, Period, Priority), y de otros tipos.

Su objetivo es relacionar la estructura lógica con recursos de ejecución, y asegurar el cumplimiento de los requisitos no funcionales.

Análisis temporal y otros tipos de análisis.

## 2.5. Codificación en ADA

Cada **objeto** se realiza mediante un **paquete**.

Las **relaciones** de uso se representan mediante clausulas **with**.

Los objetos **hijos** se realizan mediante **paquetes hijos privados**.

La implementación de **operaciones** se realiza mediante **cláusulas de renombrado**.

Los objetos **cíclicos** y **esporádicos** tienen **una tarea** y, posiblemente, un **objeto protegido de sincronización**.

