

# PROYECTO FIN DE CARRERA



Universidad  
Carlos III de Madrid

## Diseño e implementación de un sistema operativo para fines didácticos

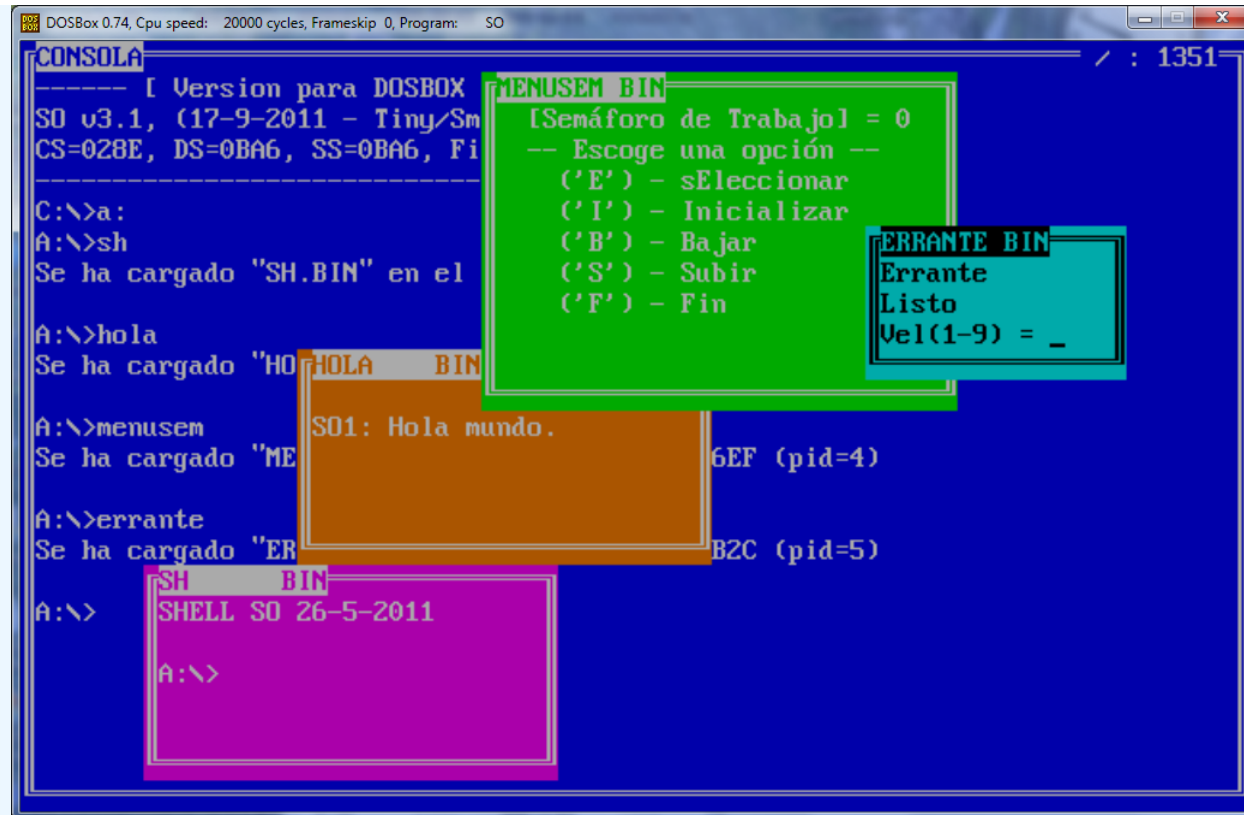
**ESCUELA POLITÉCNICA SUPERIOR  
INGENIERÍA EN INFORMÁTICA**

Autor: Aníbal Ramírez García

Tutor: Francisco Javier García Blas

# Índice

1. Introducción
2. Estado de la cuestión
3. Gestión del proyecto
4. Requisitos
5. Implementación
6. Resultados
7. Conclusiones



```
DOSBox 0.74, Cpu speed: 20000 cycles, Frameskip 0, Program: SO
CONSOLA
----- [ Version para DOSBOX
SO v3.1, (17-9-2011 - Tiny/Sm
CS=028E, DS=0BA6, SS=0BA6, Fi
-----
C:\>a:
A:\>sh
Se ha cargado "SH.BIN" en el
A:\>hola
Se ha cargado "HOLA.BIN"
A:\>menusem
Se ha cargado "MENUSEM.BIN"
A:\>errante
Se ha cargado "ERRANTE.BIN"
A:\>
SH BIN
SHELL SO 26-5-2011
A:\>

MENUSEM BIN
[Semáforo de Trabajo] = 0
-- Escoge una opción --
('E') - sEleccionar
('I') - Inicializar
('B') - Bajar
('S') - Subir
('F') - Fin

ERRANTE BIN
Errante
Listo
Vel(1-9) = _

HOLA BIN
SO1: Hola mundo.
6EF (pid=4)
B2C (pid=5)
```

## MOTIVACIÓN

### 1. Introducción

### 2. Estado de la cuestión

### 3. Gestión del proyecto

### 4. Requisitos

### 5. Desarrollo

### 6. Resultados

### 7. Conclusiones

- ☐ Necesidad de contar con un S.O. de **diseño propio**, **sencillo** en los detalles y con capacidad suficiente, para ser usado como modelo y **herramienta** de **apoyo** para la **enseñanza** de los fundamentos de los sistemas operativos.
- ☐ Los sistemas disponibles en la actualidad **no son adecuados**. En general son **demasiado complejos**.
- ☐ Para aquellos más sencillos, la realización de prácticas **requiere** un **amplio conocimiento** del **sistema** y de las herramientas empleadas en su diseño y construcción.

## OBJETIVOS BASICOS

### 1. Introducción

### 2. Estado de la cuestión

- ☐ Un diseño **simple**, de tamaño **reducido**, pero **capaz**.

### 3. Gestión del proyecto

- ☐ Correr tanto sobre máquina física como virtual.

### 4. Requisitos

- ☐ Poder ser desarrollado en un **entorno sencillo**, con facilidades de compilación, ejecución y **depuración integradas**.

### 5. Desarrollo

### 6. Resultados

### 7. Conclusiones

- ☐ Facilitar la elaboración de “**prácticas**” sobre el mismo para su realización por el estudiante.

## SSOO PARA LA ENSEÑANZA

1. Introducción



### MikeOS:

2. Estado de la cuestión

- Interfaz de menús y diálogo modo texto.

3. Gestión del proyecto

- Arranque desde disquete, CD-ROM y USB.

4. Requisitos

- Sobre unas 60 llamadas al sistema.

5 Desarrollo

- Gestor ficheros, editor textos, visor imágenes, juegos...

6. Resultados

- Incluye intérprete BASIC.

7. Conclusiones

- Conexión terminal serie

- Escrito en ASM.

## SSOO PARA LA ENSEÑANZA

1. Introducción

2. Estado de la cuestión

3. Gestión del proyecto

4. Requisitos

5. Desarrollo

6. Resultados

7. Conclusiones



### GeekOS:

- **Objetivo básico:** Ser herramienta de aprendizaje en el diseño y construcción de un núcleo de S.O.
- Incluye proyectos para la enseñanza.
- Gestión de memoria dinámica.
- Hilos sobre kernel por rodajas con planificación estática
- *Mutexes* para sincronización de hilos.
- Modo usuario con protección de memoria por segmentación.
- Controladores teclado y VGA modo texto

## SSOO PARA LA ENSEÑANZA

1. Introducción
2. **Estado de la cuestión**
3. Gestión del proyecto
4. Requisitos
5. Desarrollo
6. Resultados
7. Conclusiones

- ☐ **MenuetOS:** Su objetivo, eliminar capas entre las partes del sistema. Escrito en **ensamblador 32/64 bits**. Diseño propio. Multiprocesador, multihilo, protección anillo 3...
- ☐ **GnuFiwix:** Objetivo educacional. Basado en UNIX con núcleo compatible LINUX. POSIX. Modo protegido 32 bits. Multitarea, Memoria virtual con paginación, señales, pipes. Ficheros ext2, abstracción VFS, etc.
- ☐ **Minirighi:** Objetivo educacional. Escrito en 'C' y 'asm' desde 0 Multihilo, 32 bits, POSIX, núcleo ligero y de fácil lectura. Fácilmente adaptable a nuevas tecnologías.

**\* Estos sistemas son mas complejos que los anteriores**

## SSOO PARA LA ENSEÑANZA

### □ MINIX:

- Similar a Unix V7. Llamadas POSIX.
- Sacrifica eficiencia por comprensión y modularidad.
- Multiproceso y multiusuario.
- Para PC compatible sin uso de BIOS.
- “Drivers” video: CGA, monocromo, EGA...
- Controladores disco duro tipo XT, AT y BIOS
- Manejo red TCP
- Escrito en ‘C’ ANSI.

**\* Muy difundido y utilizado en la enseñanza**



## INTRODUCCIÓN

1. Introducción

2. Estado de la cuestión

3. **Gestión del proyecto**

4. Requisitos

5. Desarrollo

6. Resultados

7. Conclusiones

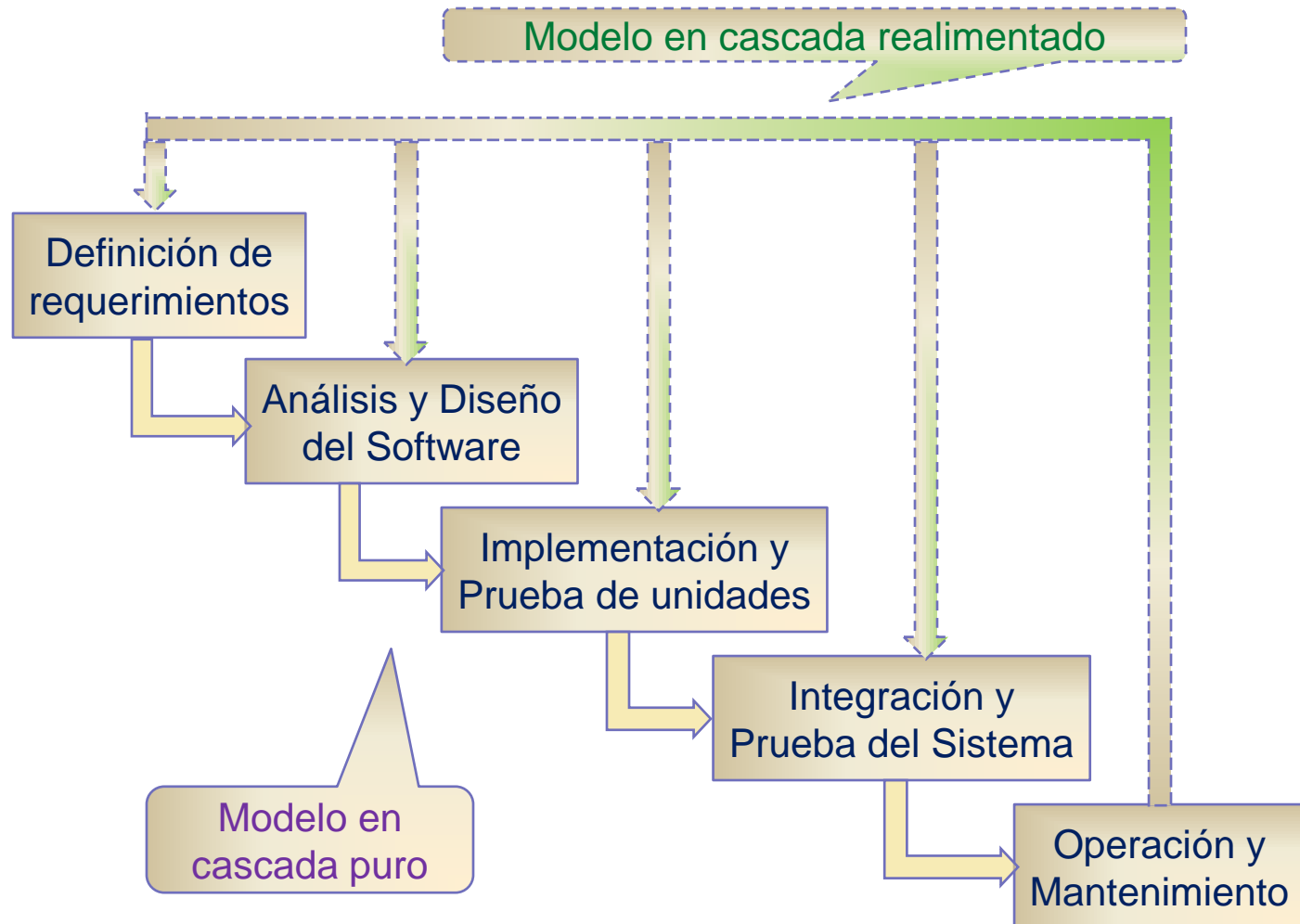
### ☐ **Etapas mínimas:**

- Captura, análisis y especificación de requisitos.
- Diseño y codificación.
- Pruebas, instalación y mantenimiento.

### ☐ **Modelos del proceso (ciclo de vida):**

- En cascada (secuencial, clásico, tradicional).
- En cascada realimentado.
- *Evolutivo incremental.*
- Espiral.

## Modelos en cascada



1. Introducción
2. Estado de la cuestión
3. **Gestión del proyecto**
4. Requisitos
5. Desarrollo
6. Resultados
7. Conclusiones

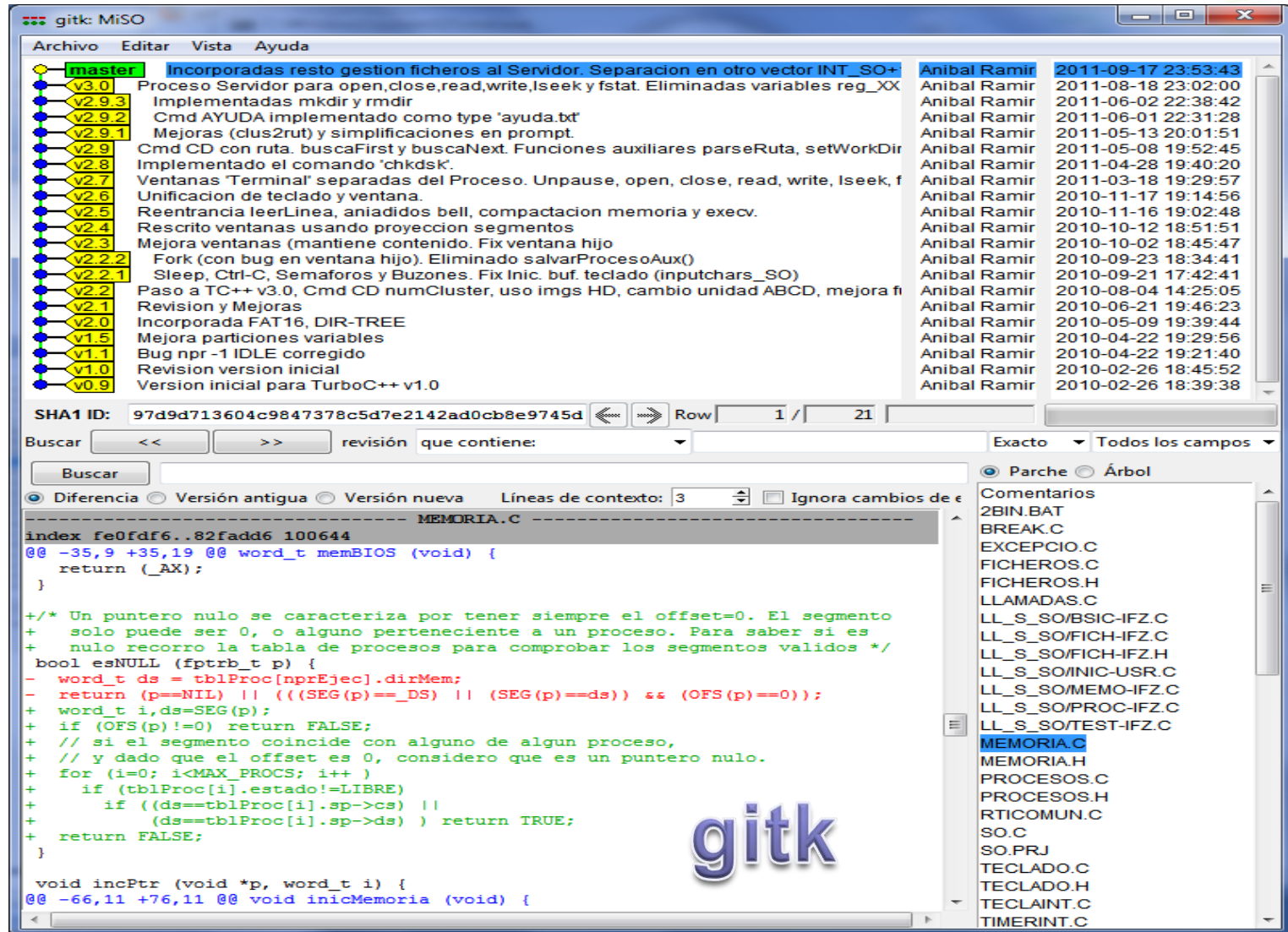
## Modelo evolutivo incremental

1. Introducción
2. Estado de la cuestión
3. **Gestión del proyecto**
4. Requisitos
5. Desarrollo
6. Resultados
7. Conclusiones



## Realización del proyecto

1. Introducción
2. Estado de la cuestión
3. **Gestión del proyecto**
4. Requisitos
5. Desarrollo
6. Resultados
7. Conclusiones



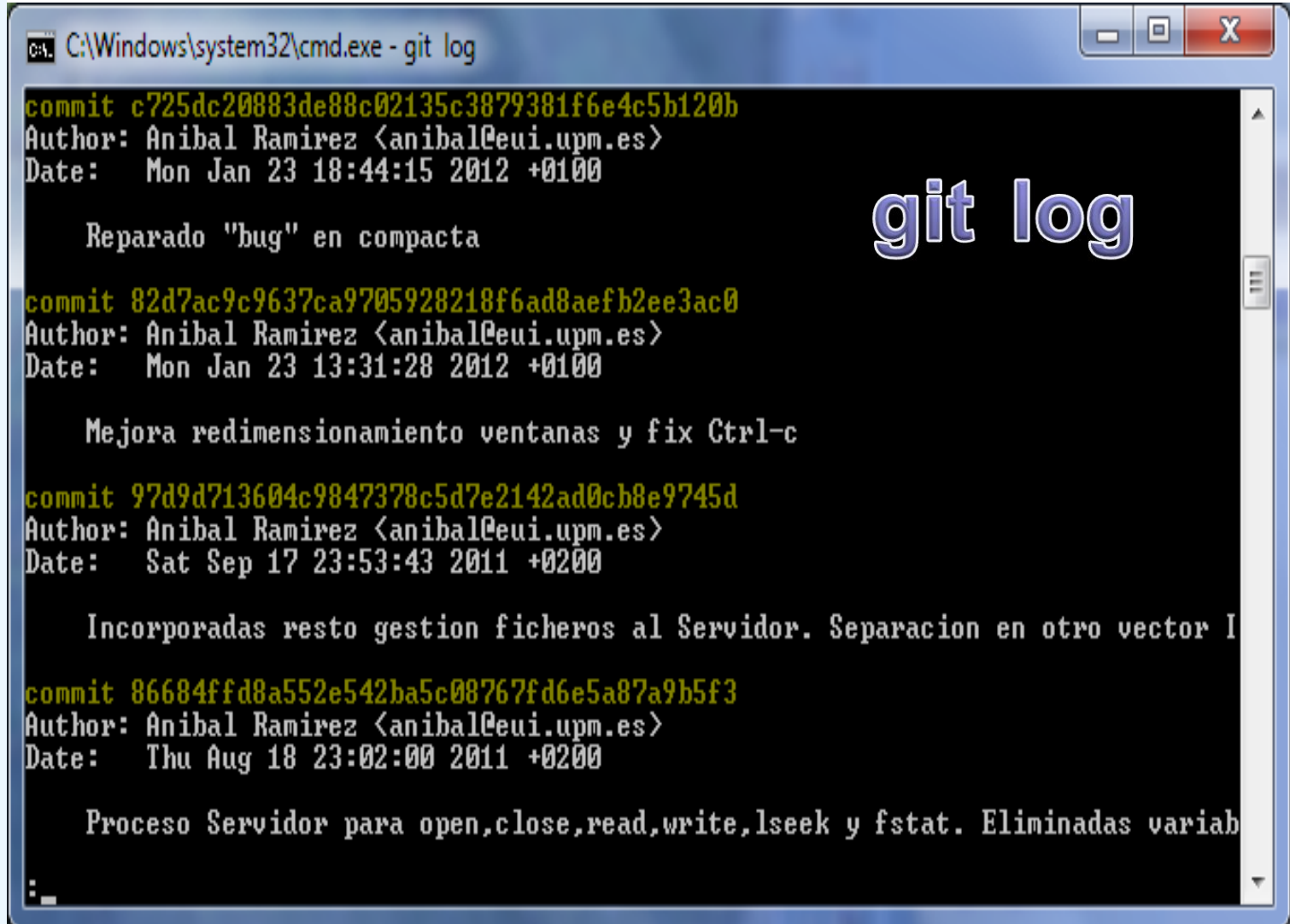
The screenshot shows the gitk GUI with the following components:

- Commit History (Left Panel):** A list of commits starting from v0.9 to master. The master commit is highlighted with a green dot and the message "Incorporadas resto gestion ficheros al Servidor. Separacion en otro vector INT\_SO+".
- Commit Details (Right Panel):** A table showing the commit hash, author (Anibal Ramir), and date for each commit.
- Diff View (Main Panel):** Shows the difference between the current commit and the previous one. The diff is for the file `MEMORIA.C` and shows changes to the `index` and `memBIOS` functions.
- File List (Bottom Right Panel):** A list of files in the repository, including `2BIN.BAT`, `BREAK.C`, `EXCEPCIO.C`, `FICHEROS.C`, `FICHEROS.H`, `LLAMADAS.C`, `LL_S_SO/BSIC-IFZ.C`, `LL_S_SO/FICH-IFZ.C`, `LL_S_SO/FICH-IFZ.H`, `LL_S_SO/NIC-USR.C`, `LL_S_SO/MEMO-IFZ.C`, `LL_S_SO/PROC-IFZ.C`, `LL_S_SO/TEST-IFZ.C`, `MEMORIA.C`, `MEMORIA.H`, `PROCESOS.C`, `PROCESOS.H`, `RTICOMUN.C`, `SO.C`, `SO.PRJ`, `TECLADO.C`, `TECLADO.H`, `TECLAINT.C`, and `TIMERINT.C`.

gitk

## Realización del proyecto

1. Introducción
2. Estado de la cuestión
3. **Gestión del proyecto**
4. Requisitos
5. Desarrollo
6. Resultados
7. Conclusiones



```
C:\Windows\system32\cmd.exe - git log

commit c725dc20883de88c02135c3879381f6e4c5b120b
Author: Anibal Ramirez <anibal@eu1.upm.es>
Date:   Mon Jan 23 18:44:15 2012 +0100

    Reparado "bug" en compacta

commit 82d7ac9c9637ca9705928218f6ad8aefb2ee3ac0
Author: Anibal Ramirez <anibal@eu1.upm.es>
Date:   Mon Jan 23 13:31:28 2012 +0100

    Mejora redimensionamiento ventanas y fix Ctrl-c

commit 97d9d713604c9847378c5d7e2142ad0cb8e9745d
Author: Anibal Ramirez <anibal@eu1.upm.es>
Date:   Sat Sep 17 23:53:43 2011 +0200

    Incorporadas resto gestion ficheros al Servidor. Separacion en otro vector I

commit 86684ffd8a552e542ba5c08767fd6e5a87a9b5f3
Author: Anibal Ramirez <anibal@eu1.upm.es>
Date:   Thu Aug 18 23:02:00 2011 +0200

    Proceso Servidor para open,close,read,write,lseek y fstat. Eliminadas variab
```

## Requerimientos más importantes del S.O.

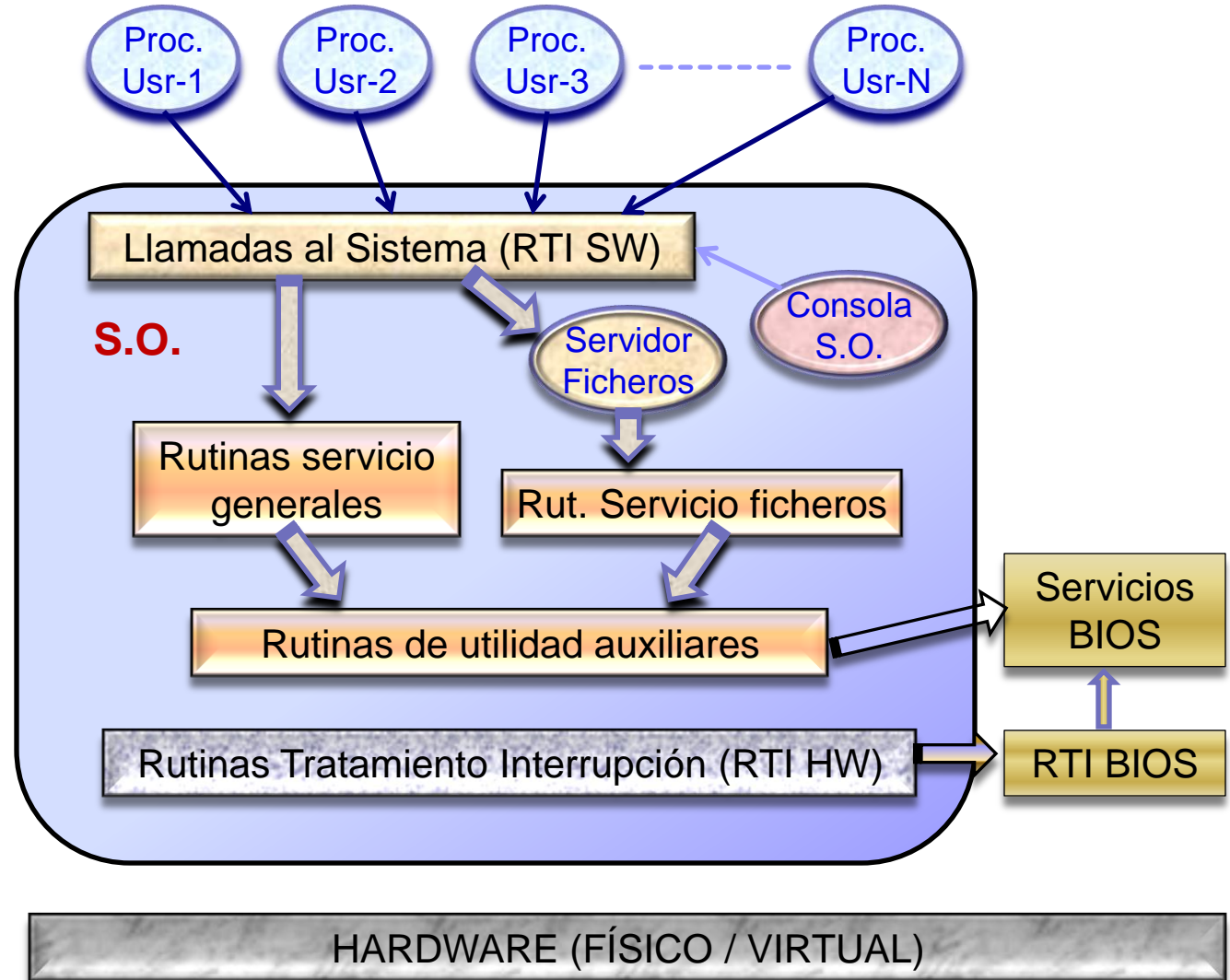
- Multitarea expulsora.
- Comunicación indirecta mediante buzones.
- Primitivas de sincronización mediante Semáforos
- Gestión memoria dinámica (con posibilidad intercambio a disco)
- Múltiples terminales virtuales mediante ventanas tipo texto.
- Sistema de ficheros tipo FAT ( al menos 12 y 16 bits).

### ----- Otros -----

- Correr en máquina desnuda (PC compatible, Intel x86).
- Correr bajo MS-DOS, pero sin sus servicios.
- Código en ficheros fuente separados según funcionalidad.
- Aceptar compilación, ejecución y depuración en un entorno integrado tipo Turbo-C.

1. Introducción
2. Estado de la cuestión
3. Gestión del proyecto
4. **Requisitos**
5. Desarrollo
6. Resultados
7. Conclusiones

## Arquitectura del sistema





## Procesos

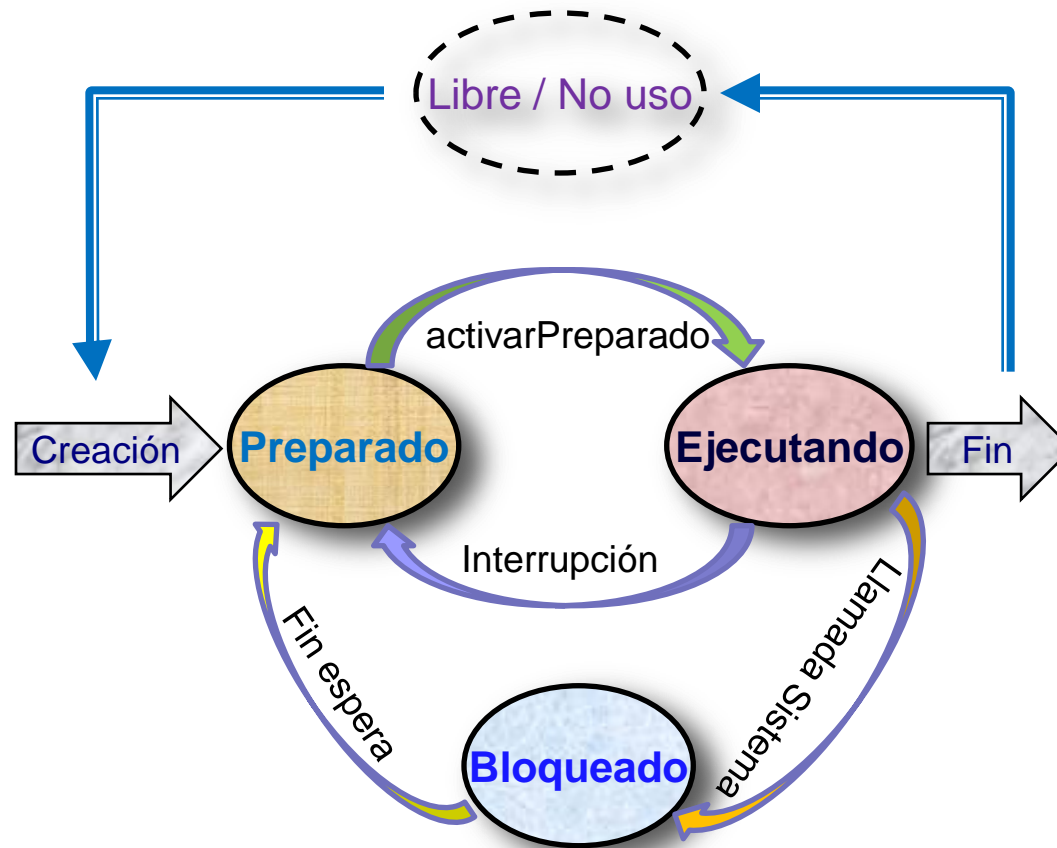
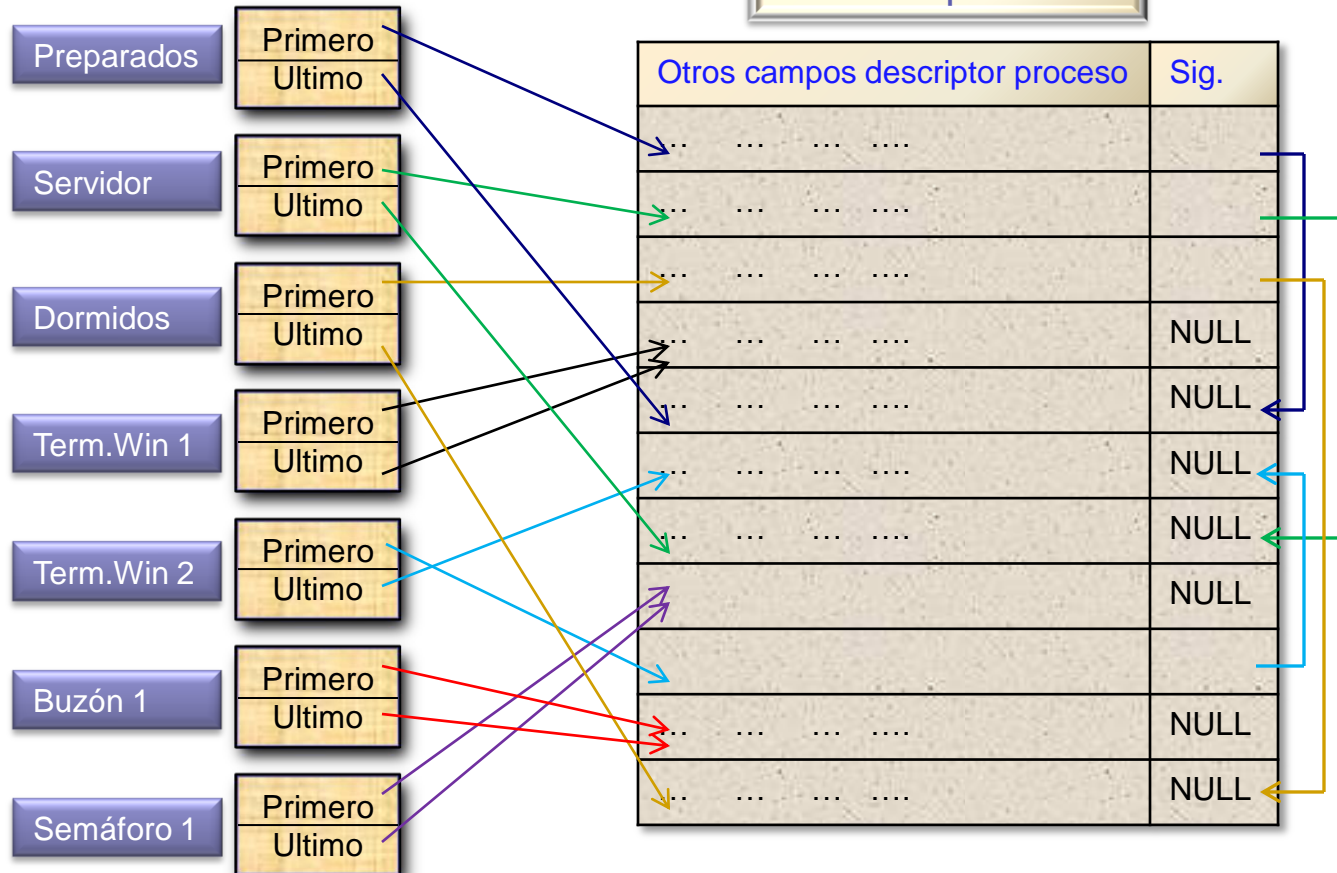


Diagrama de estados de un proceso



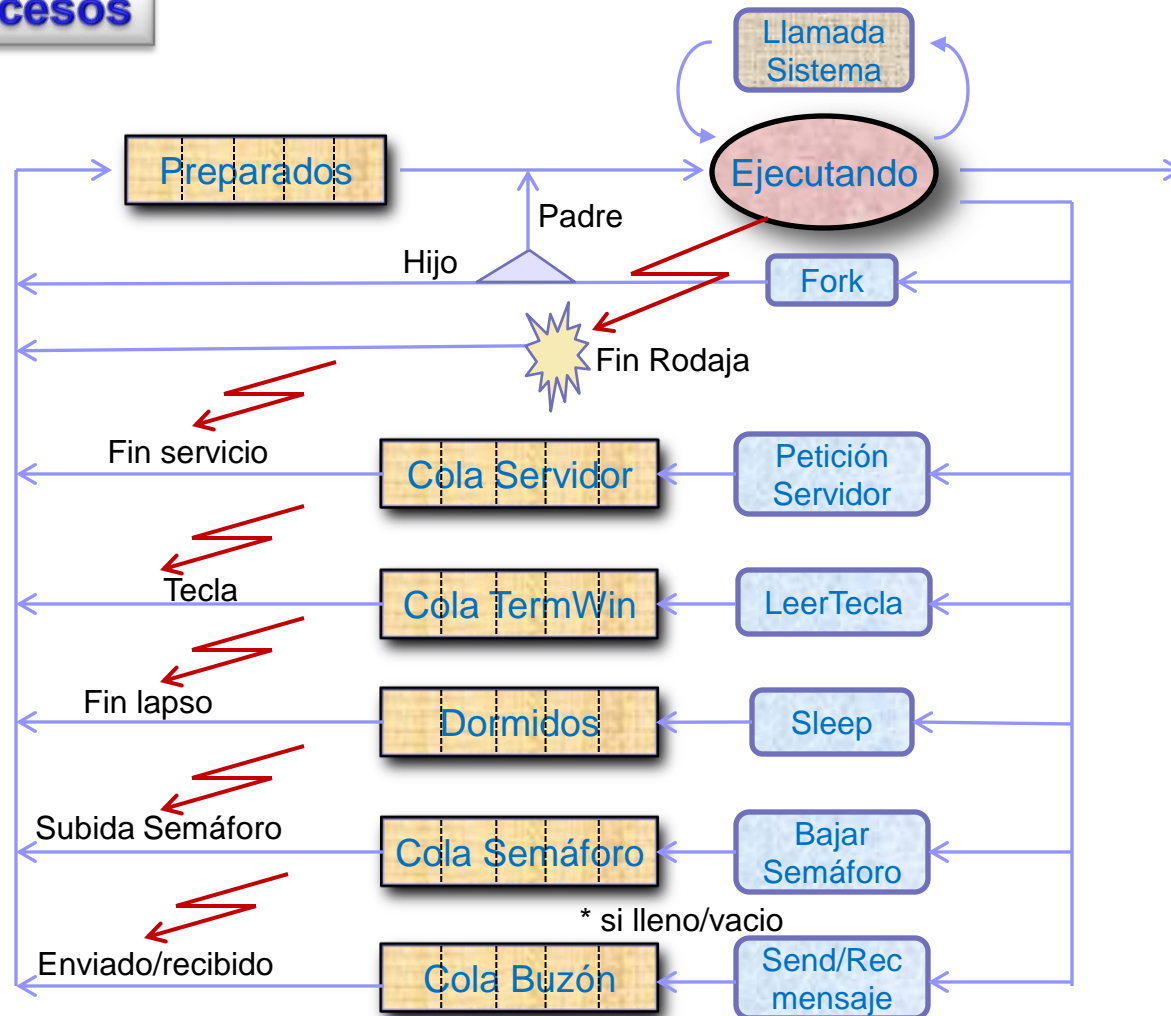
## Procesos

Tabla de procesos



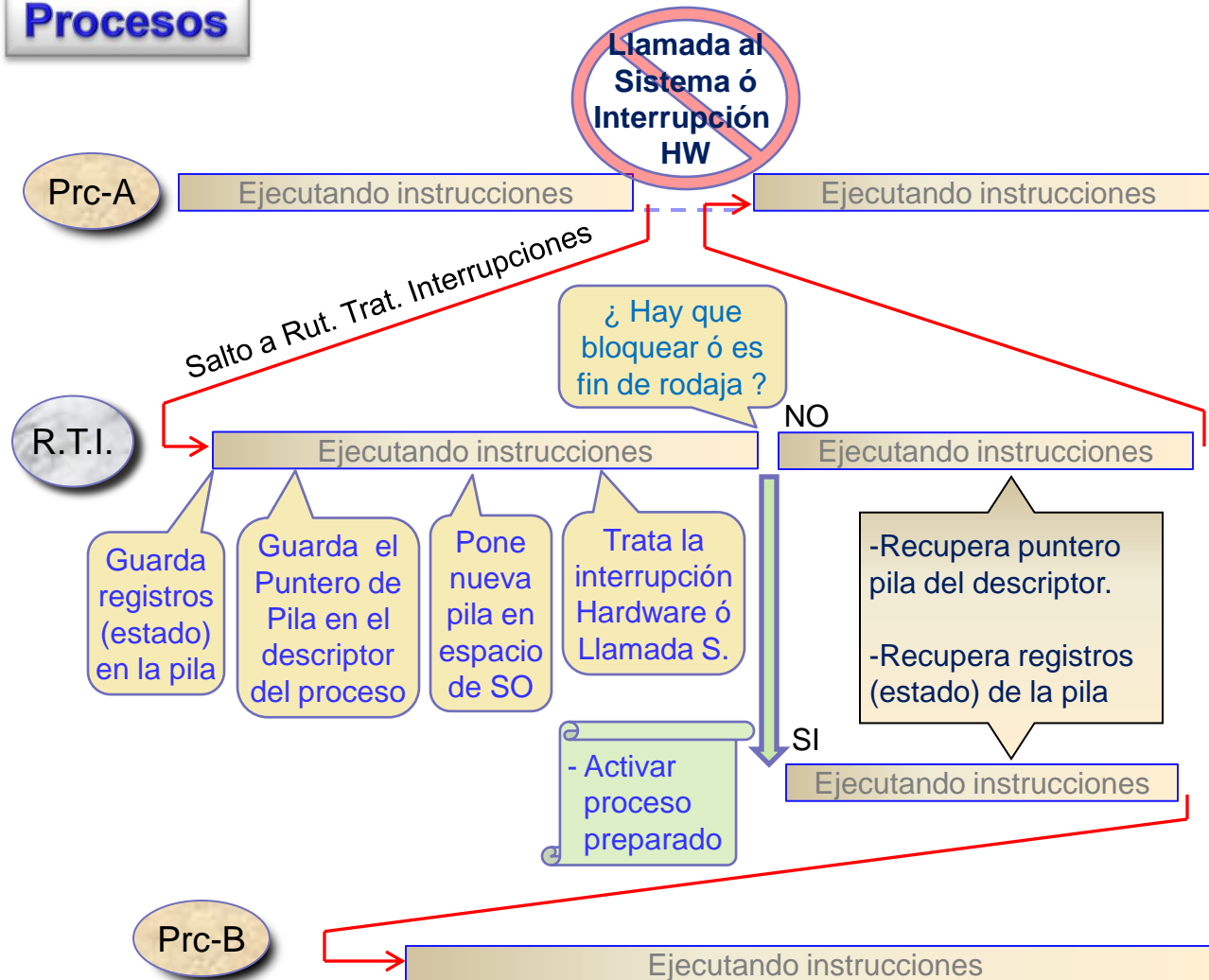
Colas de espera de varios recursos

## Procesos



## Transición de un proceso entre colas del sistema

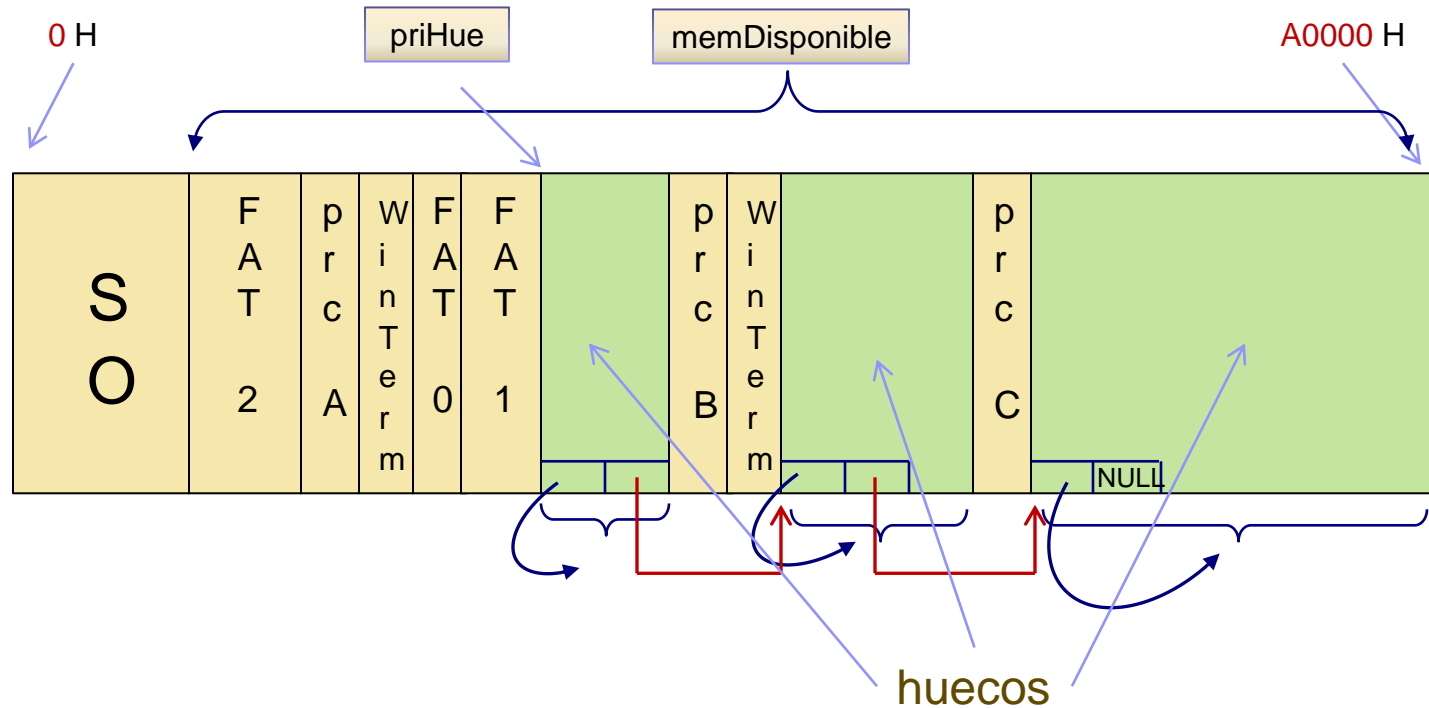
## Procesos



Esquema sobre el cambio de contexto de procesos

## Memoria

1. Introducción
2. Estado de la cuestión
3. Gestión del proyecto
4. Requisitos
5. **Desarrollo**
6. Resultados
7. Conclusiones

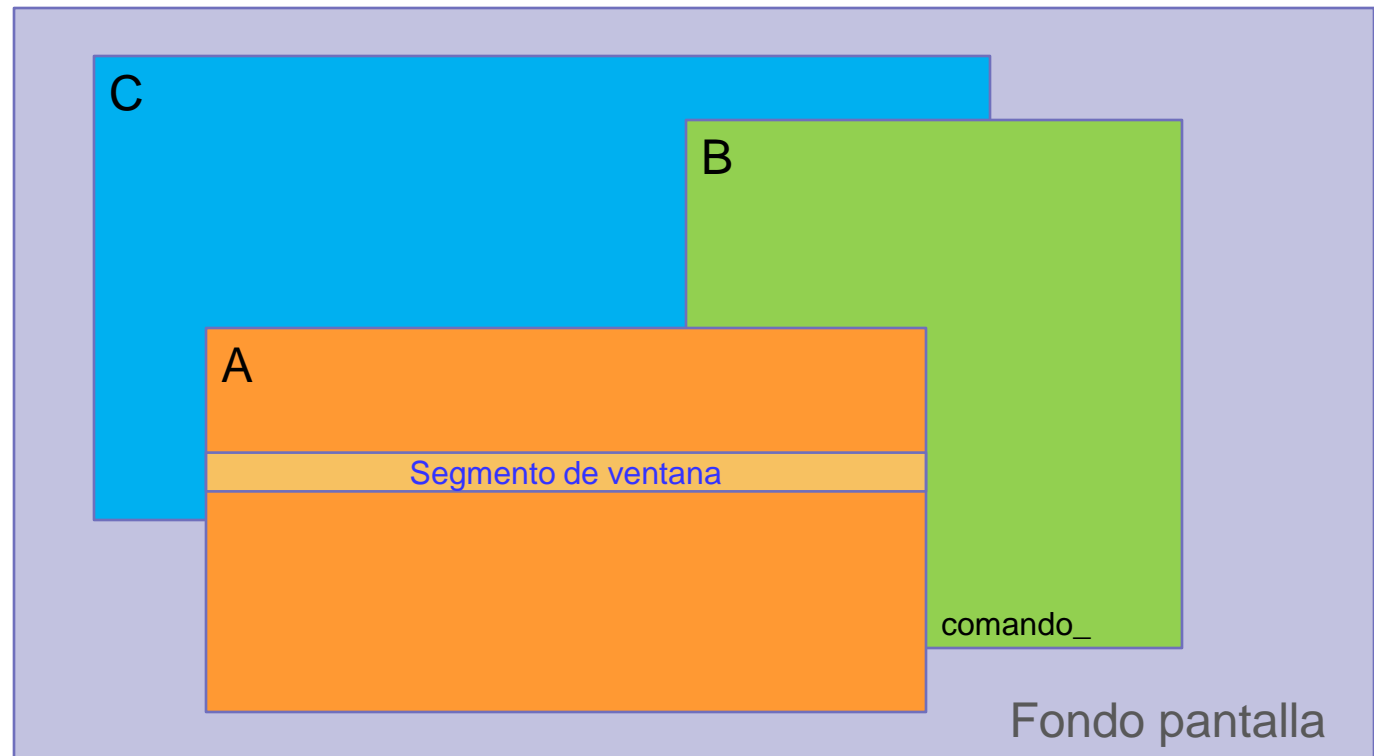
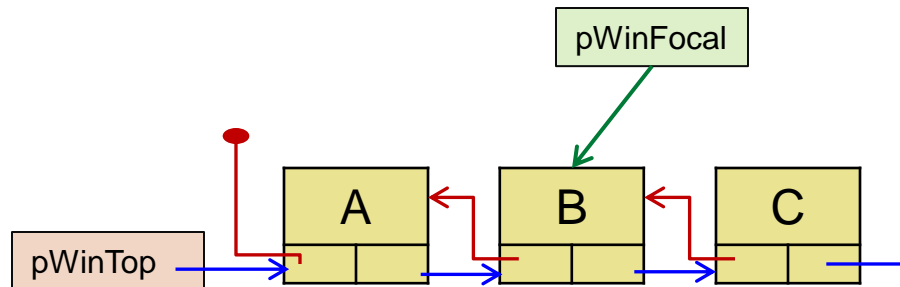


Ejemplo de mapa de memoria en SO

## E/S - Ventanas

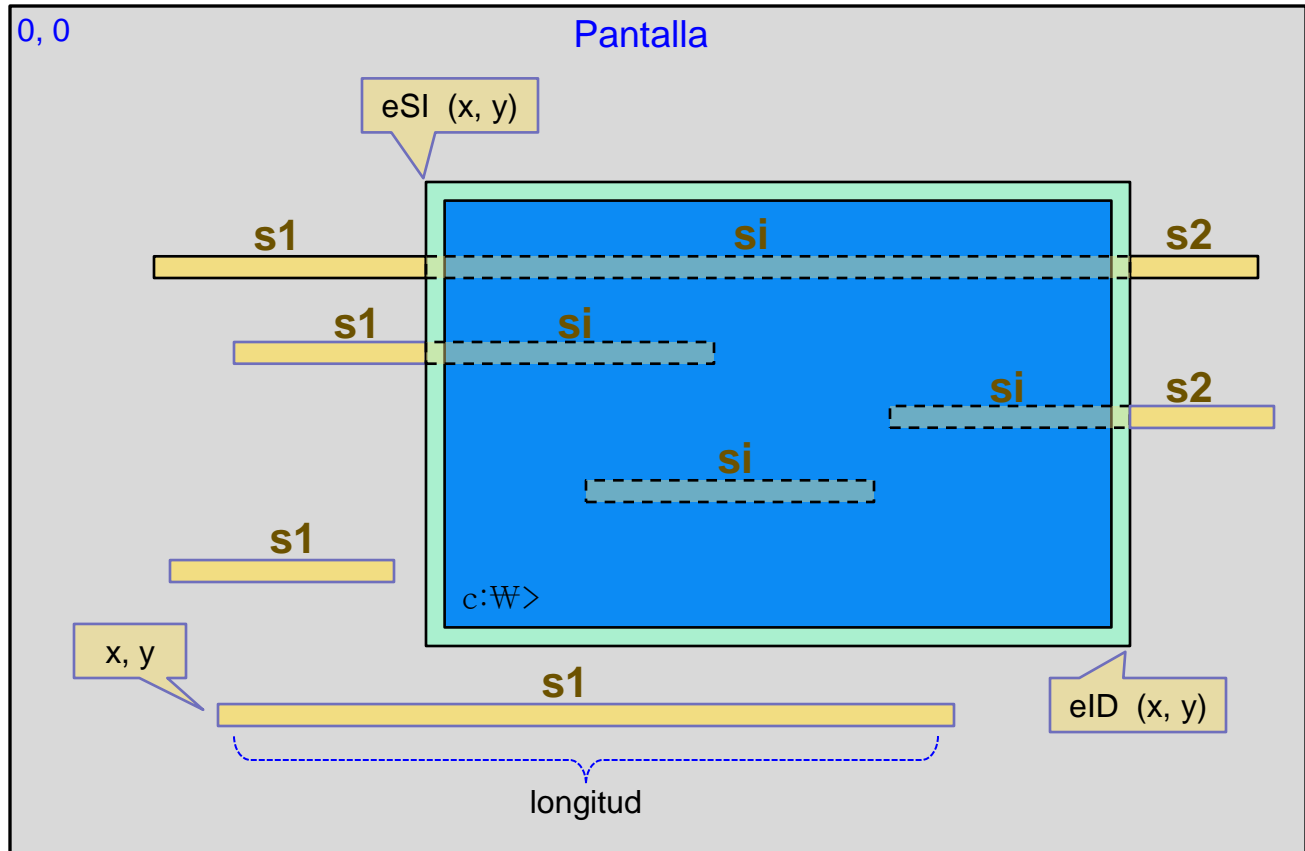
struct win\_t

- nombre
- plano
- eSI, eID, cursor
- atributo color
- pWinUp, pWinDw
- buffTeclas
- nTec, In, Out
- cola procesos



1. Introducción
2. Estado de la cuestión
3. Gestión del proyecto
4. Requisitos
5. **Desarrollo**
6. Resultados
7. Conclusiones

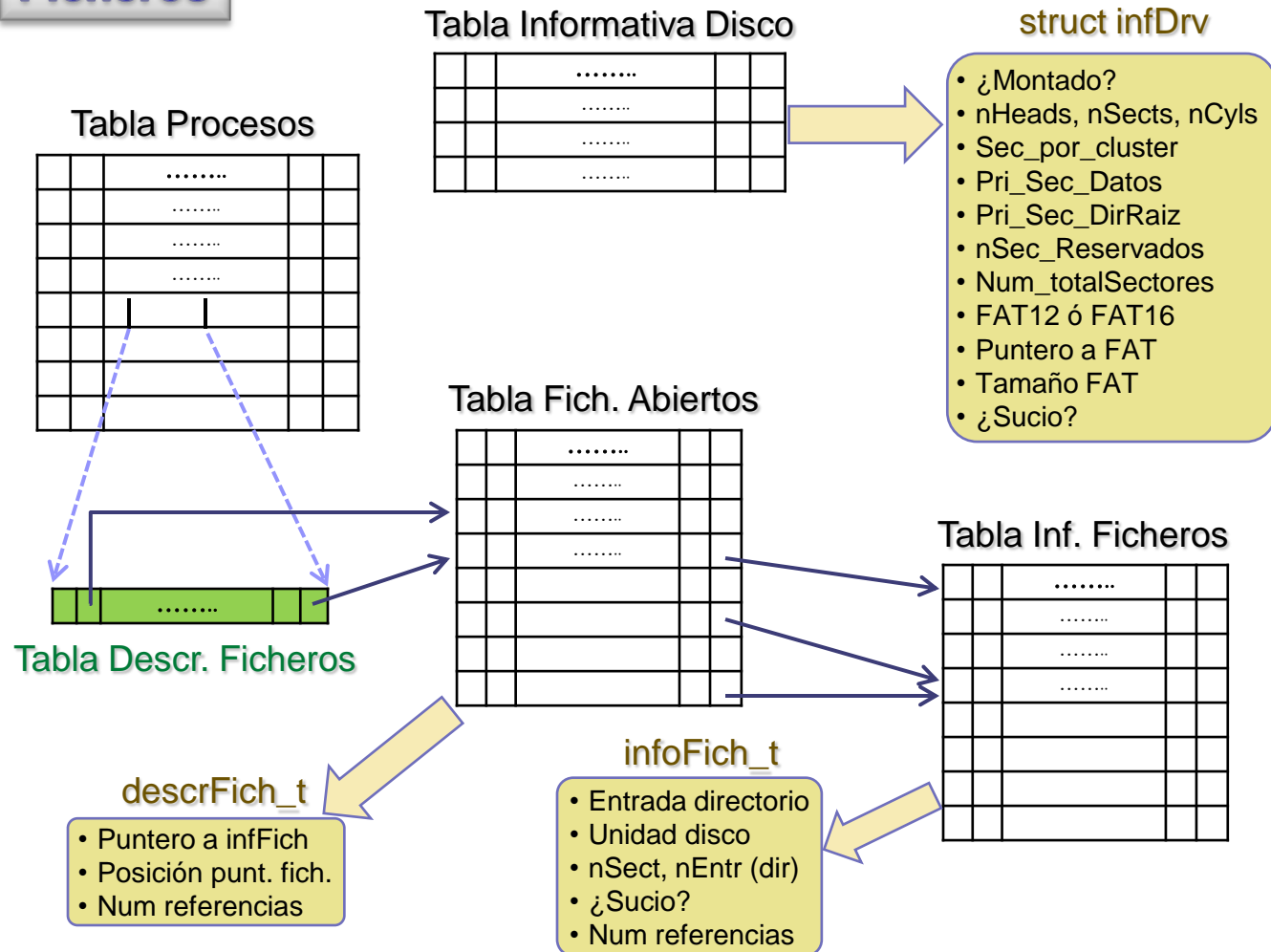
## E/S - Ventanas



Ejemplos de intersección de segmento y ventana

## Ficheros

1. Introducción
2. Estado de la cuestión
3. Gestión del proyecto
4. Requisitos
5. **Desarrollo**
6. Resultados
7. Conclusiones



Principales tablas y estructuras del sistema de ficheros



## 1. Introducción

## 2. Estado de la cuestión

## 3. Gestión del proyecto

## 4. Requisitos

## 5. Desarrollo

## 6. Resultados

## 7. Conclusiones

- Se ha logrado un S.O. simple, sobre máquina desnuda, virtual y en modo invitado DOS, que cubre todos los requisitos pedidos y que se ajusta ampliamente a las necesidades de enseñanza planteadas.
- Inclusión de diversos comandos internos para mostrar y probar determinadas partes internas del S.O.
- Desarrollo de unos programas de usuario “ejemplo o modelo”, para otros y también para realizar pruebas sobre el sistema.
- Realización de unas prácticas a realizar por el estudiante, basadas en este sistema y que muestran algunos de los conceptos más interesantes sobre los fundamentos de los SSOO.



## ➤ COMANDOS INTERNOS :

EXIT, CLS, DIR, MEM, LSV, TFA, PS, KILL, DUMP, MW, CW, CD, DEL, RD, MD, TOUCH, COMPAC, DESMON, TYPE, TST, AYUDA y CHKDSK.

## ➤ PROGRAMAS DE USUARIO:

Calibra, errante, erraslp, hola, menuexc, menusem, menumsj, tsfork, sh, sh-frk y test.

## ➤ PRÁCTICAS:

Tratamiento de interrupciones (**Ctrl-C**), manejo de excepciones (**overflow y div. 0**), llamada sleep, semáforos, buzones, compactación memoria, y arranque del sistema (**boot**).

## Comando ayuda

DOSBox 0.74, Cpu speed: 20000 cycles, Frameskip 0, Program: SO

```

CONSOLA
PS      (Muestra la tabla de procesos de SO)
MEM     (Muestra la lista de bloques libres de memoria de SO)
LSV     (Muestra la lista de ventanas)
TFA     (Muestra la tabla de ficheros abiertos)
CLS     (Limpia la pantalla)
EXIT    (Finaliza SO)
COMPAC  (Realiza una compactacion de huecos de memoria)
X:      (Cambia la unidad actual (X puede ser A,B,C o D)
DIR msk (Monta unidad. Lista dir actual (msk admite *))
KILL pid (Mata el proceso 'pid')
CD ruta (Cambia el directorio de trabajo a 'ruta')
DEL ruta (borra el fichero 'ruta')
RD ruta (borra el directorio 'ruta')
MD ruta (crea el directorio 'ruta')
TOUCH ruta [/r]      (pone fecha del sistema en 'ruta'. /r=recursiv.)
TYPE nomFich         (Muestra el contenido de un fichero)
DESMON drv           (Desmonta drive -libera su memoria FAT)
DUMP seg:ofs [bytes] (Volcado de memoria)
CW colorCar colorFondo (Establece los colores de la ventana)
CHKDSK [drv] [f] [n] (Chequea/repara el S. ficheros de la unidad 'drv'
                    'f' graba cambios. 'n'=1 ó 2, uso de Fat)
MW esqSupIzFil esqSupIzCol esqInfDeFil esqInfDeCol (Mueve/redim ventana)
C:\>
    
```

1. Introducción
2. Estado de la cuestión
3. Gestión del proyecto
4. Requisitos
5. Desarrollo
6. **Resultados**
7. Conclusiones

## Programas de usuario

1. Introducción
2. Estado de la cuestión
3. Gestión del proyecto
4. Requisitos
5. Desarrollo
6. **Resultados**
7. Conclusiones

DOSBox 0.74, Cpu speed: 20000 cycles, Frameskip 0, Program: SO

**CONSOLA**

NUEXC SH BIN

4 4 A:\>cd mi\_c0

A:\MI\_C0>dir

5 5

LIBRA	.	2- 6-2011	<DIR>
6 6	..	2- 6-2011	<DIR>

NUMSJ BUILD BAT 0- 0-1980 29

A:\>p COT OBJ 0- 0-1980 263

MI\_COT ASM 0- 0-1980 1225

Cola

Cola A:\MI\_C0>

**MENUEXC BIN** / : 8386

SO: Elija su excepcion:

'T' - Terminar programa

'D' - Division por cero

'O' - Overflow

**MENUMSJ BIN**

SO: ¿ Opcion ?:

('B') - Buzon : 0

('M') - Mensaje:

('E') - Enviar :

('R') - Recibir:

('T') - Terminar

**CALIBRA BIN**

Calibrador del sleep.

Decimas a esperar:

**ERRANTE BIN**

Errante

Listo

Vel(1-9) = \_

np	pd	est	sig	dirM	Mem	CS	I
0	0	Blq	NIL	028E	0000	028E	8
1	1	Eje	NIL	028E	191E	028E	6
2	2	Blq	NIL	2DB1	0328	2DB1	0
3	3	Blq	NIL	31DE	0318	31DE	0
4	4	Blq	NIL	35FB	042E	028E	8
5	5	Blq	NIL	3B2E	0326	3B2E	0021 3B2E 3B2E 323A
6	6	Blq	NIL	3F59	0339	3F59	0021 3F59 3F59 3366

A:\>

## CONCLUSIONES FINALES

### 1. Introducción

### 2. Estado de la cuestión

### 3. Gestión del proyecto

### 4. Requisitos

### 5. Desarrollo

### 6. Resultados

### 7. Conclusiones

- Es posible realizar un S.O. para los fines didácticos requeridos si se mantiene su complejidad dentro de unos márgenes.
- Ha sido posible realizar unas prácticas basadas en dicho S.O. y éstas avalan la idoneidad del mismo para la enseñanza.
- A medida que se intenta *aumentar* las *prestaciones* del sistema, la complejidad del mismo *tiende a dispararse*.
- El diseño monolítico ha resultado ser práctico a la hora de favorecer una gran simplicidad en los detalles de construcción.
- Realizar este proyecto ha contribuido en gran medida a mejorar la comprensión que el autor tenía sobre el diseño y construcción de un S.O.

## Presupuesto

Computing results.

SLOC	Directory	SLOC-by-Language (Sorted)
3108	top_dir	cpp=3108
632	USRS_PRG	cpp=632
333	LL_S_SO	cpp=333
0	MI_CO	(none)

Totals grouped by language (dominant language first):  
cpp: 4073 (100.00%)

Total Physical Source Lines of Code (SLOC) = 4,073

Development Effort Estimate, Person-Years (**Person-Months**) = 0.87 (10.49)  
(Basic COCOMO model, Person-Months =  $2.4 * (KSLOC^{**}1.05)$ )

Schedule Estimate, Years (Months) = 0.51 (6.11)  
(Basic COCOMO model, Months =  $2.5 * (person-months^{**}0.38)$ )

Estimated Average Number of Developers (Effort/Schedule) = 1.72

Total Estimated Cost to Develop = \$ 118,046  
(average salary = \$56,286/year, overhead = 2.40).

SLOCCount, Copyright (C) 2001-2004 David A. Wheeler

**Costes de desarrollo según “sloccount”**

## Presupuesto

USC Unified CodeCount (UCC) - RESULTS SUMMARY  
(c) Copyright 1998 - 2011 University of Southern California  
SLOC COUNT RESULTS - Generated by UCC v.2011.03 on 4 15 2012

Total Lines	Blank Lines	Comments Whole	Comments Embedded	Compiler Direct.	Data Decl.	Exec. Instr.	Number of Files	SLOC	File Type	SLOC Definition
6819	674	1695	1417	336	402	3712	48	4450	CODE	Physical
6819	674	1695	1417	336	353	3866	48	4555	CODE	Logical

### Valoración USC (SLOC lógicas):

✓ Esfuerzo estimado :  $2'4 \times (4555^{1'05}) = 11'79$  meses/hombre.

✓ Si tomamos como coste anual de un analista programador 50.400€:

**Obtenemos un coste total estimado:  $50.400 \times 11'79 / 12 = 49.518 \text{ €}$**

## Presupuesto

### Valoración subjetiva (Consideraciones):

1. Introducción

❖ 3 horas de media sobre una base de 8 horas diarias durante 2,5 años.

2. Estado de la cuestión

❖ Coste empresarial salario anual analista/programador..... = 50.400 €

3. Gestión del proyecto

❖ Coste trabajo desarrollo.....  $2'5 \times 3/8 \times 50.400 = 47.250$  €

4. Requisitos

❖ El beneficio se ha desestimado por ser un proyecto de código abierto.

5. Desarrollo

6. Resultados

7. Conclusiones

Concepto	Importe (€)
Recursos Humanos...	47.250 €
Ordenador Personal..	750 €
Software (abierto)...	0 €
Total presupuesto.....	48.000 €

## TRABAJOS FUTUROS

1. Introducción
  - Nuevas llamadas al sistema y más comandos internos.
2. Estado de la cuestión
  - Comunicación puerto serie, paralelo y ethernet.
3. Gestión del proyecto
  - Manejo de ratón (PS/2 y/o RS232).
4. Requisitos
  - Ficheros especiales de dispositivo y redireccionamiento de E/S.
5. Desarrollo
  - Instalación de nuevos *drivers* estática y/o dinámicamente.
6. Resultados
7. Conclusiones
  - Realizar programas o herramientas de apoyo al S.O. como por ejemplo, un monitor de procesos en ejecución mediante RS232.
  - Adaptar el sistema al modo protegido del i386.



# PROYECTO FIN DE CARRERA



Universidad  
Carlos III de Madrid

¡¡ MUCHAS GRACIAS !!

**ESCUELA POLITÉCNICA SUPERIOR  
INGENIERÍA EN INFORMÁTICA**

Autor: Aníbal Ramírez García

Tutor: Francisco Javier García Blas