

# PROYECTO DE SISTEMAS OPERATIVOS

---

*Juanjo Costa y Alex Pajuelo*

*Curso Q2 2020-2021*

<b>1.- Introducción</b>	<b>2</b>
<b>2.- Objetivo del proyecto</b>	<b>2</b>
<b>3.- Proyectos</b>	<b>2</b>
<b>4.- Fases del proyecto</b>	<b>2</b>
4.1.- Fase 1: Asignación de parejas y proyectos	3
4.2.- Fase 2: Diseño del proyecto	3
4.3.- Fase 3: Implementación del 90% del proyecto	3
4.4.- Fase 4: Entrega Final	4
<b>5.- Evaluación del proyecto</b>	<b>4</b>
<b>Apéndice A: Descripción de los Documentos de las entregas</b>	<b>5</b>
A.1 Documento de Diseño	5
A.2 Documento de Seguimiento	6
A.3 Documento Final	7
<b>Apéndice B: Rúbricas</b>	<b>8</b>
B.1 Rúbrica de Diseño	8
B.2 Rúbrica de Seguimiento	10
B.3 Rúbrica Final	11

## 1.- Introducción

Este documento explica todas las fases para la realización del proyecto, así como su evaluación.

Ante cualquier duda en este documento o del trabajo a realizar, dirígete al Coordinador de la Asignatura.

## 2.- Objetivo del proyecto

El objetivo del proyecto es que un estudiante demuestre haber adquirido los conocimientos suficientes del kernel de ZeOS, extendiéndolo con la implementación de una característica real presente en un sistema operativo actual.

## 3.- Proyectos

En el Racó se pueden encontrar todas las propuestas de proyectos actuales. También se aceptarán propuestas de los estudiantes si éstas han sido validadas por el Coordinador de la asignatura.

La descripción de cada proyecto tiene los siguientes campos:

- Título del proyecto.
- Máxima puntuación: puntuación que se obtendrá en caso de implementar todo lo que se pide en el enunciado.
- Descripción: descripción breve del estado actual de ZeOS relativo al proyecto y de lo que se pretende implementar.
- Niveles: pueden existir un máximo de 3. Para cada nivel se describe el trabajo a realizar para alcanzar la nota máxima de ese nivel. Los niveles son incrementales: no se obtendrá la nota del nivel 2 si el nivel 1 no se ha realizado correctamente.

## 4.- Fases del proyecto

Para realizar el proyecto, se tienen que seguir una serie de fases. Durante estas fases, descritas a continuación, los estudiantes podrán consultar cualquier tema, relativo al proyecto, al profesor. La asistencia al laboratorio no es obligatoria, pero sí recomendada, para realizar un correcto seguimiento del proyecto.

Todas las entregas se realizarán a través del Racó. En el caso del código, este tiene que entregarse dentro de un fichero .tar.gz y se tiene que haber realizado un *make clean* previamente para evitar entregar ficheros objeto o ejecutables. En el caso de tener que

entregar también documentos, éstos siempre tendrán que estar en **formato pdf**. NO SE ACEPTARÁN ficheros que no cumplan estas reglas.

Consultad la planificación de la asignatura en la página web para saber las fechas de inicio/fin de cada una de estas fases.

Consultad, también, la Guía Docente de la FIB y/o la página web de la asignatura para saber cuál es la metodología de evaluación y cuánto representa este proyecto en la nota de la asignatura.

#### **4.1.- Fase 1: Asignación de parejas**

Los proyectos se realizarán en parejas. Esta primera fase consiste en que los estudiantes notifiquen al profesor quienes son los componentes de la pareja. Salvo contadas excepciones, y siempre justificadas, no se permitirá a un estudiante realizar el proyecto en solitario.

Esta notificación se realizará a través del fórum de la asignatura en el Racó de la FIB. Consistirá en publicar un mensaje cuyo contenido será:

- Componentes de la pareja (nombre y apellidos).

#### **4.2.- Fase 2: Diseño del proyecto**

Una vez cada pareja tiene asignado un proyecto, ya puede empezar la Fase 2. Durante esta fase tiene que decidir a qué nivel quiere llegar y cómo lo va a realizar. El resultado de esta fase es un documento, llamado Documento de Diseño, que se entregará por el Racó de la asignatura.

El contenido del Documento de Diseño se puede consultar en el apéndice A.1 Documento de Diseño.

Una vez entregado este documento, el profesor de la asignatura lo revisará, evaluará y enviará feedback a la pareja, bien presencialmente o a través de correo electrónico.

#### **4.3.- Fase 3: Implementación del 90% del proyecto**

Durante esta tercera fase, los componentes de la pareja empiezan a desarrollar el proyecto, siguiendo el Documento de Diseño, junto con los juegos de prueba necesarios. El resultado de esta tercera fase es un fichero .tar.gz con el código del proyecto, que no tiene que estar completo ni funcionar, y un documento, llamado Documento de Seguimiento, en el que se detalle el progreso alcanzado.

El contenido de este documento se puede consultar en el apéndice A.2 Documento de Seguimiento.

Una vez entregado este documento, el profesor de la asignatura lo revisará, evaluará y enviará feedback a la pareja, bien presencialmente o a través de correo electrónico.

#### 4.4.- Fase 4: Entrega Final

En esta última fase, los estudiantes tienen que acabar de implementar y probar todo el proyecto (hasta el nivel que quieran alcanzar). El proyecto entregado tiene que funcionar y tiene que ser validable con juegos de prueba que los estudiantes realizarán y, también entregarán.

Además, para esta fase, se tiene que entregar un documento, llamado Documento Final, semejante al documento de la fase anterior, cuyo contenido se puede consultar en el apéndice A.3 Documento Final

Una vez entregado este documento, el profesor de la asignatura lo revisará, evaluará y enviará feedback a la pareja, bien presencialmente o a través de correo electrónico.

### 5.- Evaluación del proyecto

Cada una de las fases del proyecto, excepto la primera, se evaluarán y su nota se publicará en el Racó, de forma separada o conjuntamente.

Para evaluar cada una de las fases, existen, en el Apéndice B, un conjunto de rúbricas en las que se exponen qué conceptos se evaluarán y cuál es el peso de ese apartado respecto a la nota total que se puede obtener en esa fase.

Existen 3 rúbricas:

- Fase 2: Diseño.
- Fase 3: Implementación del 75% del proyecto.
- Fase 4: Entrega Final.

Cada una de estas rúbricas tienen un listado de conceptos que se evaluarán junto la descripción del trabajo a realizar en cada uno de estos conceptos para obtener una calificación de A, B, C y/o D (nivel alcanzado) en ese concepto. Además, cada concepto tiene un peso asignado sobre el total de la rúbrica. El nivel alcanzado es un factor multiplicador de esos pesos de tal forma que:

- Una A es el 100% del peso.
- Una B es el 50% del peso.
- Una C es el 25% del peso.
- Una D es el 0% del peso.

Para obtener la nota de esa fase, se sumarán todos los pesos de los conceptos, después de aplicarles, a cada uno de ellos, el factor multiplicador del nivel alcanzado.

En caso de que un concepto de una rúbrica no sea aplicable a un proyecto en concreto, los pesos de los otros conceptos, antes de aplicar el factor multiplicador de nivel alcanzado, se aumentarán de forma proporcional, hasta que sumen el 100% de la rúbrica.

Finalmente, la nota publicada puede variar ligeramente de la rúbrica debido a redondeos o a criterio del profesor.

## Apéndice A: Descripción de los Documentos de las entregas

En este apéndice se describen los documentos necesarios en cada entrega:

- Documento de Diseño.
- Documento de Seguimiento.
- Documento Final.

### A.1 Documento de Diseño

La primera entrega del proyecto consiste en un documento, de como mucho 4 páginas donde se describan las modificaciones que se tienen que realizar a ZeOS para implementar el mecanismo descrito. Los apartados obligatorios de este documento son:

1.- Nivel al que se quiere optar en el proyecto (ver descripciones de los proyectos)

2.- Identificación de todos los temas de la siguiente lista aplicables al proyecto y que por lo tanto implementará:

- Interrupciones y excepciones.
- Mecanismos de sincronización.
- Medición tiempo (estadísticas).
- Planificación de procesos.
- Gestión de memoria nivel usuario.
- Gestión de memoria nivel sistema.
- Llamadas a sistema.
- Signals.
- Sistema de ficheros virtual (VFS).
- Gestión de dispositivos de bloques.
- Comunicación entre procesos.

3.- Breve descripción de la implementación que se realizará junto con las decisiones de diseño/implementación tomadas.

4.- Descripción de las estructuras nuevas que se tienen que añadir a ZeOS para implementar este proyecto. Esta descripción tiene que incluir:

- El nombre de la estructura.
- La justificación de por qué se necesita esta estructura.
- Los campos de la estructura junto con una breve descripción de cada uno de ellos.

5.- Descripción de las estructuras que se existen actualmente en ZeOS y que se tienen que modificar. Esta descripción tiene que incluir:

- El nombre de la estructura.
- Justificación de por qué se tiene que modificar esta estructura.
- Descripción de la estructura a modificar.

6.- Descripción de las modificaciones que se tienen que realizar en el código de ZeOS. Para cada modificación se tiene que indicar:

- Fichero y función a modificar.
- Justificación de la modificación.
- Modificación a realizar.

7.- Descripción de los juegos de prueba que se implementarán para probar el correcto funcionamiento de la implementación. Para cada juego de pruebas se tiene que indicar:

- Objetivo del juego de pruebas (qué se quiere validar).
- Descripción del juego de pruebas (cómo se implementará).
- Comprobación de los resultados obtenidos.

8.- En el caso de que el proyecto pida evaluar el rendimiento del mecanismo propuesto, se tiene que describir cómo se va a realizar esta evaluación. Para ello, se tiene que describir:

- Conjunto de juegos de prueba a utilizar. Para cada uno de ellos se tiene que describir cuál es el objetivo (¿qué se va a medir?) y cómo se va a implementar ese juego de pruebas.
- Métricas que se van a utilizar en la evaluación.
- Cómo se van a realizar las mediciones para obtener las métricas.
- Escenarios a evaluar.

9.- Haz una lista de las tareas que se tienen que realizar para alcanzar los objetivos planteados. Para cada tarea tienes que señalar:

- Nombre de la tarea.
- Descripción breve del trabajo a realizar.

## A.2 Documento de Seguimiento

La entrega de seguimiento del proyecto consiste en:

- El código fuente de ZeOS.
- Los juegos de prueba que habéis utilizado.
- Documento con la lista de tareas actualizada.

El código fuente de ZeOS con el mecanismo implementado. No es necesario que sea 100% funcional, pero sí que debe incluir más del 50% de las tareas finalizadas. Se tiene que entregar un único fichero llamado *proyecto.tar.gz*. Se tiene que haber hecho un make clean previamente.

Los juegos de prueba pueden estar dentro del fichero user.c de ZeOS pero tienen que ser fácilmente diferenciables y se tiene que poder elegir fácilmente cuál de ellos se quiere ejecutar.

La evaluación del rendimiento, **si se pide**, consiste en un documento llamado Evaluación del Rendimiento donde se muestre y explique una gráfica en la que se vea claramente lo que se pide en el enunciado junto con una pequeña conclusión.

El documento de seguimiento tiene una longitud máxima de 4 páginas y tiene que incluir:

1.- Nivel alcanzado.

2.- Explicación de todas las decisiones nuevas tomadas, respecto al documento de diseño, para finalizar el proyecto (hasta el nivel deseado)

3.- Explicación de los problemas detectados respecto a lo descrito en el Documento de Diseño y cómo se han solucionado (si se han solucionado)

4.- Listado de tareas del Documento de Diseño junto con su estado (hecho, en proceso, no hecho). En el caso de que hayan aparecido nuevas tareas, se tienen que incluir también junto con una breve descripción y su estado.

Todos estos ficheros y documentos se tienen que entregar en un único fichero comprimido llamado GrupoXX.tar.gz donde XX es el número del grupo asignado previamente por el profesor.

### A.3 Documento Final

La última entrega del proyecto consiste en:

- El código fuente de ZeOS
- Los juegos de prueba que habéis utilizado
- Evaluación del rendimiento (en el caso de que el enunciado la pidiese)
- Documento de la entrega final

El código fuente de ZeOS con el mecanismo implementado (dependiendo del nivel final alcanzado) tiene que funcionar al 100%. Se tiene que entregar un único fichero llamado *proyecto.tar.gz*. Se tiene que haber hecho un make clean previamente.

Los juegos de prueba pueden estar dentro del fichero user.c de ZeOS pero tienen que ser fácilmente diferenciables y se tiene que poder elegir fácilmente cuál de ellos se quiere ejecutar.

La evaluación del rendimiento, si se pide, consiste en un documento llamado Evaluación del Rendimiento donde se muestre y explique una gráfica en la que se vea claramente lo que se pide en el enunciado junto con una pequeña conclusión.

El documento de entrega final tiene una longitud máxima de 4 páginas y tiene que incluir:

- 1.- Nivel final alcanzado.
- 2.- Explicación de todas las decisiones nuevas tomadas, respecto al documento de diseño, para finalizar el proyecto (hasta el nivel alcanzado)
- 3.- Explicación de los problemas detectados respecto a lo descrito en el Documento de Diseño y cómo se han solucionado (si se han solucionado)
- 4.- Listado de tareas del Documento de Diseño junto con su estado final. En el caso de que hayan aparecido nuevas tareas, se tienen que incluir también junto con una breve descripción y su estado final.

Todos estos ficheros y documentos se tienen que entregar en un único fichero comprimido llamado GrupoXX.tar.gz donde XX es el número del grupo asignado previamente por el profesor.



## Apéndice B: Rúbricas

Aquí se describen las rúbricas para la autoevaluación de cada entrega:

- Rúbrica de Diseño
- Rúbrica de Seguimiento
- Rúbrica Final

### B.1 Rúbrica de Diseño

[peso] TAREA	A	B	C	D
[5] Identificación de los temas del proyecto	Identifica TODOS los temas del proyecto	Identifica 50% de los temas del proyecto	Identifica el 25% de los temas del proyecto	**
[5] Breve descripción de la implementación que se realizará junto con las decisiones de diseño/implementación tomadas	La descripción es clara y se remarcan las decisiones de diseño tomadas para implementar la solución.	Hay algún detalle sin aclarar	Hay descripción pero sin decisiones de diseño	*
[20] Descripción de las estructuras nuevas que se tienen que añadir a ZeOS para implementar este proyecto	Identifica TODAS las estructuras nuevas necesarias, con su nombre y justificación y el listado de todos sus campos, con una justificación para cada uno.	Identifica la mayoría de estructuras pero se deja alguna, alguna justificación o algún campo	Aparecen estructuras innecesarias o justificación pobre	*
[20] Descripción de las estructuras que se existen actualmente en ZeOS y que se tienen que modificar	Identifica TODAS las estructuras a modificar necesarias, con su nombre y justificación y el listado de todos sus campos nuevos/modificados, con una justificación para cada uno.	Identifica la mayoría de estructuras, pero se deja alguna estructura, alguna justificación o algún campo	Aparecen estructuras innecesarias o justificación pobre	*
[20] Descripción de las modificaciones que se tienen que realizar en el código de ZeOS	Describe TODAS las funciones a modificar y el motivo, junto con una pequeña descripción informal de lo que debe hacer.	Identifica la mayoría de funciones, pero se deja alguna, algún motivo o algún campo		*

[5] Descripción de los juegos de prueba	Enumera pruebas a nivel funcional sobre todas y cada una de las llamadas a crear. Incluye pruebas de conjunto. Describe para cada prueba lo que quiere evaluar y cómo, así como la comprobación del resultado	Se deja alguna prueba	Hay alguna prueba, pero circunstancial	*
[5 (si aplica)] Evaluar el rendimiento	Describe claramente qué se quiere medir, con qué métrica y como se medirá. Así mismo también describe los escenarios a evaluar.	Se describe que se quiere medir pero las métricas y/o los experimentos no son correctos.	Se describe que se quiere medir pero las métricas y los experimentos no son correctos.	*
[20] Lista de las tareas	Describe un conjunto (>10) de tareas concretas a realizar, agrupadas en tareas y subtareas para una mejor organización. Describe para cada tarea una descripción breve y precisa del trabajo a realizar	Se deja algún detalle	Pocas tareas, o demasiado genéricas	*
Valor	100	50	25	0

\* No hecho, no justificación, no correcto.

\*\* Identifica menos del 25% de los temas.

## B.2 Rúbrica de Seguimiento

[peso] TAREA	A	B	C	D
[5 (Si aplica)] Cambios respecto al diseño inicial	Justificación de los cambios o no respecto al documento anterior	-	-	No justificación
[5 (si aplica)] Problemas detectados	Descripción de los problemas encontrados y como se han solventado	-	-	No justificación
[45] Listado tareas actualizado	>= 75% tareas acabadas	>=50% & <75%	>=25% & <50%	<25%
[45] CÓDIGO	Más del 75% de las tareas funcionan correctamente y existen juegos de pruebas para probarlas	Más del 50% de las tareas funcionan correctamente y existen juegos de pruebas para probarlas	Más del 25% de las tareas funcionan correctamente y existen juegos de pruebas para probarlas	no compila
Valor	100	50	25	0

### B.3 Rúbrica Final

[peso] TAREA	A	B	C	D
[5 (si aplica)] Cambios respecto al diseño seguimiento	Justificación de los cambios o no respecto al documento anterior	-	-	No justificación
[5 (si aplica)] Problemas detectados	Descripción de los problemas encontrados y como se han solventado	-	-	No justificación
[45] Listado tareas actualizado	100% tareas acabadas	<100% & >=75%	<75% & >=50%	<50%
[45] CÓDIGO	El 100% de las tareas están implementadas, funcionan correctamente y existen juegos de pruebas para probarlas	El 75% de las tareas están implementadas, funcionan correctamente y existen juegos de pruebas para probarlas	El 50% de las tareas están implementadas, funcionan correctamente y existen juegos de pruebas para probarlas	Se han completado menos del 50% de las tareas o las que hay no funcionan correctamente .
Valor	100	50	25	0