



DEPARTAMENTO DE LENGUAJES
Y SISTEMAS INFORMÁTICOS E INGENIERÍA
DEL SOFTWARE

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS INFORMÁTICOS

UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DOCTORAL

EVALUACIÓN PARA EL APRENDIZAJE EN LA ASIGNATURA
SISTEMAS OPERATIVOS EN MODALIDAD ONLINE: UN ESTUDIO
CUALITATIVO BASADO EN LA TAXONOMÍA DE BLOOM

Autora: Sonia Pamplona Roche
Director: Nelson Medinilla Martínez

Madrid, 2014

A mi director, Nelson Medinilla

*y a los adultos del futuro, en especial a
Manuel, Nacho, Beatriz, Valeria,
Nuria, Clara, Raúl, Sofía, Valentina, Pablo, Marcos,
Pablo, Guillermo, Cristina, Sergio,
Palmira, Alvar, Alonso,
Esther, Elba, Myriam, Natalia y Daniel*

Agradecimientos

Esta tesis refleja el trabajo de cinco años de investigación que no hubieran sido posibles sin la ayuda de muchas personas. Desde estas líneas quiero agradecer a todas su colaboración.

En primer lugar, quiero dar las gracias a mi director, Nelson Medinilla, por haber confiado en mí desde los comienzos de esta investigación y por haber sido siempre mi luz en este largo y arduo camino.

También quiero agradecer todo el apoyo recibido por parte de la Universidad a Distancia de Madrid, UDIMA, institución en la que he trabajado como profesora estos últimos cinco años y en la que se ha llevado a cabo esta investigación. En especial, quiero dar las gracias por la atención recibida al presidente D. Roque de las Heras y a la directora de innovación Dña. Ana Landeta.

Agradezco a la directora del laboratorio Decoroso Crespo y a todos sus miembros la ayuda prestada y las aportaciones realizadas para mejorar mi trabajo. Muchas gracias a Angélica de Antonio Jiménez, Pamela Catherine Flores Naranjo, Cristian Moral Martos, Diego Riofrío Luzcando, Óscar Javier Romero López, Juan Luís Gutiérrez Blanco y Graciela Lara López.

Quiero dar las gracias a todos mis compañeros de la Universidad a Distancia de Madrid por procurarme un ambiente multidisciplinar que me enriquece cada día. En especial quiero agradecer sus consejos y su aliento a Ana Peñas, Margarita Garbisu, Giovanna Caprara, Javier Bravo, Patricia Víctor, Yolanda Berdasco, Esther Pascua, Jordi Manel Monferrer, Laura Fernández de Mosteyrín, Isabel Sánchez de Movellán, Agustina Martínez, Sara Fuentes, Esther López, Ángela García Carballo, Olga Martínez Moure y Raquel García Revilla.

Envío mi agradecimiento a todo el personal de las bibliotecas de la Escuela Técnica Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid y de la Universidad a Distancia de Madrid, UDIMA, por haber atendido siempre todas mis peticiones de la mejor forma posible.

Agradezco la amabilidad y la ayuda prestada por José Luis Fuertes, Secretario Técnico del programa de Doctorado en Tecnologías para el Desarrollo de Sistemas Software Complejos.

Muchas gracias por sus consejos y por la ayuda recibida a los profesores Jacinto González Pachón, Xavier Ferré Grau, Jaime Ramírez Rodríguez, Julio García del Real y Ricardo Imbert Paredes.

Doy mi más profundo agradecimiento a los estudiantes que han participado en este estudio y a todos los estudiantes en general, puesto que son el verdadero motivo para la realización de esta tesis.

Finalmente, agradezco a mis familiares y amigos su apoyo incondicional. Muchas gracias a Luisa Roche, David Pamplona, María García, Manuel Pamplona, Cristina Roche, Mabel Alonso, Marcela Vargas, Raquel Escutia, Noemí Borja, Marian Díaz, María Casao, Juana Gómez, Julia Pérez, Eva Sánchez, María Quirós, María Ipas, Cristina Valladares, Nuria Alonso, Laura Concepción, Sergio Veiga y Libertad Troitiño.

Resumen

La asignatura Sistemas Operativos presenta dificultades para su aprendizaje, pero poco se conoce acerca de las mismas, ya que no han sido determinadas ni estudiadas por la literatura. Asimismo, los trabajos existentes sobre la enseñanza y aprendizaje de Sistemas Operativos se limitan a proponer distintos enfoques para impartir la asignatura y en general no evalúan el aprendizaje de los estudiantes para comprobar la eficacia del método propuesto ni usan metodologías de investigación rigurosas. Por otra parte, la impartición de la asignatura Sistemas Operativos en modalidad *online* ha sido escasamente estudiada y podría tener dificultades adicionales a las de la modalidad presencial, ya que el contexto *online* impone una serie de restricciones tanto para el profesor como para el estudiante.

En la presente tesis se ha llevado a cabo una evaluación formativa en la asignatura Sistemas Operativos, perteneciente al Grado de Ingeniería Informática de una universidad *online*. El objetivo inicial de la evaluación era descubrir las dificultades de los estudiantes para la comprensión de los conceptos de la asignatura. Posteriormente y, dada la buena aceptación de la evaluación por parte de los estudiantes, se ampliaron los objetivos del trabajo para explorar los efectos de la evaluación realizada sobre el aprendizaje.

La evaluación formativa diseñada está basada en la taxonomía revisada de Bloom y sus principales objetivos son: (a) promover el aprendizaje significativo y (b) hacer a los estudiantes conscientes de su proceso de aprendizaje. La metodología de investigación utilizada es el estudio de caso cualitativo y la muestra está constituida por 9 estudiantes del total de 13 matriculados en la asignatura. Los datos cualitativos analizados proceden de las pruebas de evaluación formativa llevadas a cabo por los estudiantes durante la impartición de la asignatura.

Los conceptos de sistemas operativos que han resultado más difíciles de comprender en el curso *online* estudiado han sido las *interrupciones* y los *semáforos*. Además, alrededor de estos conceptos se han identificado las dificultades específicas y sus posibles causas. Las dificultades descubiertas acerca de los *semáforos* corroboran las investigaciones existentes en el área de programación concurrente. El resto de las dificultades identificadas no habían sido determinadas por la literatura existente.

En cuanto a los efectos de la evaluación formativa sobre el aprendizaje, la evidencia empírica muestra que ésta ha provocado en los estudiantes una reflexión profunda sobre los conceptos de la asignatura y sobre su propio proceso de aprendizaje. El estudio de caso presentado puede ayudar a los profesores del área de ingeniería a crear evaluaciones formativas en cursos *online*.

La tesis, por tanto, realiza aportaciones relevantes en las áreas de enseñanza y aprendizaje de sistemas operativos, evaluación formativa, metodologías cualitativas y educación *online*.

Abstract

Operating Systems is a difficult subject to learn; however little is known about said difficulties, as they have not been studied nor determined by the relevant literature. Existing studies on teaching and learning the subject of operating systems are limited to presenting different approaches for teaching the subject and generally do not evaluate students' learning to verify the effectiveness of the proposed methods, nor do they use rigorous research methodologies. On the other hand, there are very few studies on teaching operating systems *online*, which may inherently present more difficulties than the in-person format, since an *online* context imposes a series of restrictions on both professors and students, such as not having face-to-face interaction for communications.

This thesis studies a formative assessment of the subject of operating systems, as part of the Degree in Information Technology Engineering for an *online* university. The initial objective of this assessment was to determine the students' difficulties in conceptual comprehension for this subject. Once students had accepted the assessment, the study's objectives were expanded to include an investigation of the effects of the assessment on learning.

The designed formative assessment was based on Revised Bloom's Taxonomy with the following main objectives: (a) to promote meaningful learning and (b) to make students aware of their learning process. The research methodology involves a qualitative case study with a sample consisting of 9 of the total 13 students registered for this course. The qualitative data analyzed comes from the formative assessment tests taken by these students during the course.

The most difficult operating systems concepts for students in the *online* course were *interrupts* and *semaphores*. Additionally, the specific difficulties and their possible causes have been identified. The students' comprehension difficulties with *semaphores* corroborate the existing research in the area of concurrent programming. The other identified difficulties were not discussed in the existing literature.

Regarding the effects of the formative assessment on learning, the empirical evidence shows that it causes students to reflect carefully on the subject's concepts as well as their own learning process. The presented case study can help professors in the area of engineering to create formative assessments for *online* courses.

This thesis, therefore, makes relevant contributions to the areas of teaching and learning operating systems, formative assessment, qualitative methodologies, and *online* education.

Índice breve

1. Introducción.....	1
1.1 Contexto	1
1.2 Problema de investigación	1
1.3 Propósito del estudio	2
1.4 Publicación de resultados.....	3
1.5 Estructura del trabajo.....	4
2. Marco teórico.....	7
2.1 Evaluación formativa <i>versus</i> evaluación sumativa	7
2.2 Aprendizaje significativo.....	9
2.3 Taxonomía revisada de Bloom	9
2.4 La investigación en educación en informática (<i>Computer Science Education Research</i>)	11
2.5 Métodos cualitativos	12
3. Trabajos relacionados.....	13
3.1 Enseñanza y aprendizaje de sistemas operativos.....	13
3.2 Otros trabajos relacionados	47
3.4 Contexto y originalidad del presente trabajo.....	52
4. Estudio empírico.....	55
4.1 Definición del problema	55
4.2 Metodología	58
4.3 Diseño de las pruebas de evaluación formativa	58
4.4 Recogida de datos	72
4.5 Análisis de datos	72
4.6 Validez, confiabilidad y ética	77
5. Resultados y discusión.....	81
5.1 Dificultades en la comprensión del concepto <i>interrupción</i>	81
5.2. Dificultades en la comprensión del concepto <i>semáforo</i>	132
5.3 Efectos de la evaluación formativa sobre el aprendizaje	136
5.4 Resumen de la sección resultados y discusión	143
6. Conclusiones y trabajo futuro	147
Bibliografía	151

Anexos.....	163
Anexo A. Análisis de los objetivos de aprendizaje de Sistemas Operativos.....	165
Anexo B. Prueba de evaluación: Cuestionario de conocimientos previos	169
Anexo C. Prueba de evaluación: Cuestionario I.....	171
Anexo D. Prueba de evaluación: Cuestionario II	175
Anexo E. Prueba de evaluación: Cuestionario III.....	179
Anexo F. Prueba de evaluación: Ejercicios y problemas I.....	185
Anexo G. Prueba de evaluación: Ejercicios y problemas II	189
Anexo H. Prueba de evaluación: Prácticas con un simulador	193
Anexo I. Consentimiento informado	205
Anexo J. Códigos acerca de las dificultades de aprendizaje del concepto <i>interrupción</i>	207

Índice general

1. Introducción.....	1
1.1 Contexto	1
1.2 Problema de investigación	1
1.3 Propósito del estudio	2
1.4 Publicación de resultados.....	3
1.5 Estructura del trabajo.....	4
2. Marco teórico.....	7
2.1 Evaluación formativa <i>versus</i> evaluación sumativa	7
2.2 Aprendizaje significativo.....	9
2.3 Taxonomía revisada de Bloom	9
2.4 La investigación en educación en informática (<i>Computer Science Education Research</i>)	11
2.5 Métodos cualitativos	12
3. Trabajos relacionados.....	13
3.1 Enseñanza y aprendizaje de sistemas operativos.....	13
3.1.1 Análisis del contenido de la asignatura Sistemas Operativos	15
3.1.2 Análisis de las fechas de publicación de los trabajos	15
3.1.3 Análisis del propósito principal de los trabajos.....	16
3.1.4 Proyectos de desarrollo de un sistema operativo real.....	18
3.1.4.1 Descripción	18
3.1.4.2 Análisis.....	19
3.1.5 Proyectos de desarrollo de un sistema operativo instruccional	23
3.1.5.1 Descripción	23
3.1.5.2 Análisis.....	25
3.1.6 Otros proyectos de desarrollo.....	27
3.1.6.1 Descripción	28
3.1.6.2 Análisis.....	29
3.1.7 Uso de simuladores de sistemas operativos	31
3.1.7.1 Descripción	32
3.1.7.2 Análisis.....	33
3.1.8 Uso de aplicaciones web	35
3.1.8.1 Descripción	35
3.1.8.2 Análisis.....	36

3.1.9 Utilización de técnicas de aprendizaje activo.....	37
3.1.9.1 Descripción	37
3.1.9.2 Análisis.....	38
3.1.10 Uso de juegos para motivar al aprendizaje de la asignatura	39
3.1.10.1 Descripción	39
3.1.10.2 Análisis.....	40
3.1.11 Estudiar la comprensión de la asignatura Sistemas Operativos.....	41
3.1.11.1 Descripción	41
3.1.11.2 Análisis.....	42
3.1.12 Resumen de la revisión de literatura	43
3.2 Otros trabajos relacionados	47
3.2.1 Trabajos acerca de la comprensión de conceptos	47
3.2.1.1 Bibliografía de Seymour Papert acerca del aprendizaje con computadoras	47
3.2.1.2 Estudios acerca de las <i>misconceptions</i> en ciencias.....	47
3.2.2 Trabajos acerca de la evaluación formativa <i>online</i>	49
3.4 Contexto y originalidad del presente trabajo.....	52
4. Estudio empírico.....	55
4.1 Definición del problema	55
4.1.1 El contexto <i>online</i>	55
4.1.2 La asignatura Sistemas Operativos.....	57
4.2 Metodología	58
4.3 Diseño de las pruebas de evaluación formativa.....	58
4.3.1 Objetivos del curso.....	59
4.3.2. Pruebas de evaluación	61
4.3.2.1 Selección de las pruebas de los principales textos de sistemas operativos.....	62
4.3.2.2 Modificación de las pruebas seleccionadas.....	67
4.3.2.3 Descripción de las pruebas de evaluación llevadas a cabo.....	69
4.4 Recogida de datos	72
4.4.1 Muestra	72
4.4.2. Documentos	72
4.5 Análisis de datos.....	72
4.5.1 Análisis de datos para descubrir conceptos que resultan difíciles.....	73
4.5.2. Análisis de datos para descubrir las dificultades que hay alrededor de cada concepto..	75
4.5.3 Análisis de datos para descubrir efectos de la evaluación formativa	75
4.5.4 Esfuerzo dedicado al proceso de análisis	76
4.6 Validez, confiabilidad y ética	77
4.6.1 Validez interna	77
4.6.3 Validez externa.....	79
4.6.2 Confiabilidad	79
4.6.4 Ética.....	80
5. Resultados y discusión.....	81
5.1 Dificultades en la comprensión del concepto <i>interrupción</i>	81
5.1.1 Importancia del concepto interrupción en la asignatura Sistemas Operativos	81
5.1.2 Primeros resultados	82

5.1.3	Significados del término <i>interrupción</i>	83
5.1.3.1	Significados del término <i>interrupción</i> en los diccionarios	84
5.1.3.2	Significados del término <i>interrupción</i> en los libros de texto.....	89
5.1.5	Preguntas acerca de la comprensión del concepto <i>interrupción</i>	102
5.1.5.1	Primera pregunta acerca del concepto interrupción.....	102
5.1.5.2	Segunda pregunta acerca del concepto interrupción.....	111
5.1.5.3	Tercera pregunta acerca del concepto interrupción	115
5.1.5.4	Cuarta pregunta acerca del concepto interrupción	120
5.1.5.5	Quinta pregunta acerca del concepto interrupción.....	124
5.1.5.6	Conclusiones acerca de la comprensión del concepto interrupción	127
5.2.	Dificultades en la comprensión del concepto <i>semáforo</i>	132
5.2.1	Preguntas acerca de la comprensión del concepto <i>semáforo</i>	132
5.2.1.1	Primera pregunta acerca del concepto semáforo	132
5.2.1.2	Segunda pregunta acerca del concepto semáforo	133
5.2.1.3	Tercera pregunta acerca del concepto semáforo.....	134
5.2.2	Conclusiones acerca de la comprensión del concepto <i>semáforo</i>	135
5.3	Efectos de la evaluación formativa sobre el aprendizaje	136
5.3.1	Reflexión acerca de los conceptos de la asignatura.....	136
5.3.2	Desconcierto en la resolución de las pruebas de evaluación.....	139
5.3.3	Toma de conciencia de los progresos y las dificultades en el aprendizaje	141
5.4	Resumen de la sección resultados y discusión	143
6.	Conclusiones y trabajo futuro	147
	Bibliografía	151
	Anexos.....	163
	Anexo A. Análisis de los objetivos de aprendizaje de Sistemas Operativos	165
	Anexo B. Prueba de evaluación: Cuestionario de conocimientos previos	169
	Anexo C. Prueba de evaluación: Cuestionario I.....	171
	Anexo D. Prueba de evaluación: Cuestionario II	175
	Anexo E. Prueba de evaluación: Cuestionario III.....	179
	Anexo F. Prueba de evaluación: Ejercicios y problemas I.....	185
	Anexo G. Prueba de evaluación: Ejercicios y problemas II	189
	Anexo H. Prueba de evaluación: Prácticas con un simulador	193
	Anexo I. Consentimiento informado	205
	Anexo J. Códigos acerca de las dificultades de aprendizaje del concepto <i>interrupción</i>	207

Índice de tablas

Tabla 3.1. Análisis de los trabajos acerca de proyectos de desarrollo de un sistema operativo real.....	22
Tabla 3.2. Análisis de los trabajos acerca de sistemas operativos instruccionales.....	26
Tabla 3.3. Análisis de los trabajos acerca de otros proyectos de desarrollo	30
Tabla 3.4. Análisis de los trabajos acerca de simuladores de sistemas operativos	34
Tabla 3.5. Análisis de los trabajos acerca de aplicaciones web.....	37
Tabla 3.6. Análisis de los trabajos acerca de técnicas de aprendizaje activo	39
Tabla 3.7. Análisis de los trabajos acerca de juegos.....	40
Tabla 3.8. Análisis de los trabajos acerca de la comprensión de conceptos	42
Tabla 3.9. Resumen de la revisión de literatura	46
Tabla 4.1. Número y porcentaje de objetivos por categoría de procesos cognitivos.....	61
Tabla 4.2. Número y porcentaje de objetivos por tipo de conocimiento.....	61
Tabla 4.3. Objetivos instruccionales de la pregunta 8 de la prueba de evaluación <i>Cuestionario I</i>	63
Tabla 4.4. Objetivos instruccionales de la pregunta 6 de la prueba de evaluación <i>Cuestionario II</i>	64
Tabla 4.5. Objetivos instruccionales de la pregunta 1 de la prueba de evaluación <i>Ejercicios y problemas I</i>	65
Tabla 4.6. Objetivos instruccionales y objetivos educativos del ejercicio 1 de la prueba <i>Ejercicios y problemas II</i>	66
Tabla 5.1. Apariciones del término <i>interrupt</i> en el libro de texto <i>Operating System Concepts with Java</i>	90
Tabla 5.2. Apariciones del término <i>interrupt</i> en el libro <i>Operating Systems: Internal and Design Principles</i>	94
Tabla 5.3. Tipos de interrupciones	94
Tabla 5.4. Mecanismos para interrumpir la ejecución de un proceso	95
Tabla 5.5. Apariciones del término <i>interrupción</i> en el libro <i>Sistemas Operativos Una visión aplicada</i>	97
Tabla 5.6. Significados de <i>interrupción</i> en la pregunta I	109
Tabla 5.7. Dificultades y significados de <i>interrupción</i> asociados (Primera pregunta)	110
Tabla 5.8. Resumen por estudiante: significados de <i>interrupción</i> y dificultades (Primera pregunta).....	110
Tabla 5.9. Significados de <i>interrupción</i> en la primera pregunta	114
Tabla 5.10. Significados de <i>interrupción</i> . Primera y segunda pregunta.	114
Tabla 5.11. Resumen de la comprensión del concepto <i>interrupción</i>	130

1. Introducción

El presente capítulo tiene como objeto introducir el tema de estudio de la tesis doctoral. Para ello, en primer lugar se precisa el área de investigación, después se describe el problema que ha originado el trabajo, y finalmente se enumeran las publicaciones a las que ha dado lugar el estudio realizado.

1.1 Contexto

El presente trabajo se enmarca en un área de investigación de reciente aparición que se ha consolidado en los últimos años: la educación en informática, cuyo objetivo es estudiar los procesos de enseñanza y aprendizaje en las disciplinas relacionadas con la computación.

1.2 Problema de investigación

La asignatura de Sistemas Operativos, perteneciente a los estudios superiores de Ingeniería en Informática, es considerada difícil tanto en su impartición como en su aprendizaje por la literatura, los profesores y los estudiantes (Downey, 1999; Jong, Lai, Hsia, Lin, & Lu, 2013; Luiz Paulo Maia, Machado, & Pacheco, 2005).

Sin embargo, poco se conoce sobre las dificultades que presenta y sus causas. La mayoría de los trabajos sobre enseñanza y aprendizaje de sistemas operativos se limitan a describir experiencias prácticas y carecen de una base teórica que permita explicar por qué determinadas herramientas o técnicas facilitan más el aprendizaje que otras.

Por otra parte, el aprendizaje de la asignatura Sistemas Operativos mediante un curso *online* puede ser más difícil que en una clase presencial porque este contexto impone una serie de restricciones tanto al profesor como al

estudiante, tales como la pérdida de la interacción cara a cara en las comunicaciones.

La primera motivación de este trabajo fue el estudio las dificultades en el aprendizaje de la asignatura Sistemas Operativos en un contexto *online*. Para llevar a cabo esta tarea, se diseñó un estudio empírico con el único propósito de observar el proceso de aprendizaje de los estudiantes.

Para detectar las dificultades de los estudiantes en el proceso de aprendizaje, fue necesario determinar qué debían aprender los estudiantes y diseñar pruebas de evaluación para comprobar si los estudiantes habían aprendido lo que estaba previsto.

Durante las primeras semanas de curso se observó, de forma inesperada, que la evaluación puesta en marcha tenía efectos positivos sobre el aprendizaje de los estudiantes. Una explicación de este resultado reside en que la evaluación diseñada para detectar las dificultades comparte estrategias con la evaluación formativa, cuyo propósito es precisamente facilitar el aprendizaje.

Como consecuencia de los efectos positivos manifestados, se ampliaron los objetivos del trabajo al estudio y análisis de dos aspectos distintos dentro de la evaluación formativa. El primero centrado en las dificultades que existen durante el aprendizaje de la asignatura Sistemas Operativos, y el segundo en los efectos de la evaluación formativa sobre este proceso de aprendizaje.

1.3 Propósito del estudio

Las siguientes preguntas de investigación han servido de guía para realizar el estudio:

1. ¿Qué conceptos de sistemas operativos resultan más difíciles de comprender?
2. ¿Qué dificultades tienen los estudiantes alrededor de estos conceptos?
3. ¿Cómo influye la evaluación formativa diseñada en las experiencias de aprendizaje de los estudiantes?

1.4 Publicación de resultados

Los resultados principales de la tesis son dos: la participación y publicación de resultados en un congreso internacional y la publicación de un artículo en una revista incluida en el índice Journal Citation Reports (JCR).

- **Participación en el congreso *Koli Calling* en noviembre de 2013.**

Sonia Pamplona, Nelson Medinilla y Pamela Flores (2013). Exploring misconceptions of operating systems in an *online* course. En *Proceedings of the 13th Koli Calling International Conference on Computing Education Research - Koli Calling '13* (pp. 77–86). ACM Press.

El congreso anual denominado *Koli Calling International Conference on Computing Education Research* se caracteriza por publicar trabajos con una sólida base teórica y con métodos de investigación rigurosos. Está organizado por la Universidad de Newcastle de Australia y la Universidad Aalto de Finlandia. Ambas universidades cuentan con programas de doctorado de reconocido prestigio en el área de investigación en educación informática (Computer Science Education Research).

El trabajo publicado en el congreso Koli Calling presenta los resultados preliminares del trabajo sobre las dificultades encontradas en el proceso de aprendizaje de la asignatura Sistemas Operativos en un contexto *online*.

- **Publicación de un artículo en el número monográfico denominado “Current Trends of E-Learning in Engineering Education” de la revista *International Journal of Engineering Education*.**

Sonia Pamplona, Nelson Medinilla y Pamela Flores (2014). Assessment for Learning: A Case Study of an *Online* Course in Operating Systems. *International Journal of Engineering Education* (aceptado el 22 de mayo de 2014, en proceso de publicación).

La revista *International Journal of Engineering Education* es una de las pocas revistas dedicadas a la educación en ingeniería que están incluidas en el índice Journal Citations Report (JCR).

El trabajo publicado en esta revista presenta los resultados definitivos del trabajo acerca de los efectos sobre el aprendizaje de la evaluación formativa diseñada para la asignatura Sistemas Operativos en un contexto *online*.

1.5 Estructura del trabajo

A continuación se describe brevemente el contenido de los siguientes capítulos de la presente tesis:

- **Capítulo 2: Marco teórico.** En este capítulo se hace una breve referencia al marco teórico utilizado en el estudio. En primer lugar se definen los conceptos de evaluación formativa y aprendizaje significativo y después se hace un breve resumen de la taxonomía de objetivos de educación utilizada en este trabajo: la taxonomía revisada de Bloom. Después se proporciona información sobre el área de investigación a la que pertenece esta tesis, educación en informática. Por último, se precisa el propósito de los métodos cualitativos y los resultados que se esperan de un estudio de este tipo.
- **Capítulo 3. Trabajos relacionados.** Este capítulo refleja el resultado del trabajo bibliográfico dedicado a estudiar aquellos trabajos que otorguen significado a la presente investigación. En primer lugar, se ofrece un análisis exhaustivo de los trabajos acerca de la enseñanza y aprendizaje de sistemas operativos, el tema central de esta tesis. Este análisis constituye la mayor parte del capítulo. Después, se discuten otros trabajos que constituyen una referencia para este estudio por compartir objetivos y/o metodología, aunque no pertenezcan al área de sistemas operativos.
- **Capítulo 4. Estudio empírico.** En este capítulo se describe el estudio empírico llevado a cabo. En primer lugar se precisa el contexto de la investigación: un aula virtual de la asignatura Sistemas Operativos en una universidad *online*. En segundo lugar se justifica la metodología de investigación utilizada. En tercer lugar se discute el diseño de las pruebas de evaluación realizadas por los estudiantes durante el curso. Por último,

se detallan los procesos de recogida y análisis de datos y las estrategias adoptadas respecto a la validez, confiabilidad y ética de la investigación.

- **Capítulo 5. Resultados y discusión.** En este capítulo se comunican los principales resultados de la investigación. En primer lugar se discuten las dificultades de comprensión de los conceptos *interrupción* y *semáforo*. Posteriormente se describen los efectos de la evaluación formativa sobre el aprendizaje. Para finalizar, se sintetizan las respuestas a las preguntas de investigación.
- **Capítulo 6. Conclusiones y trabajo futuro.** Este capítulo resume los hallazgos de la investigación, hace explícitas las contribuciones del estudio y plantea las líneas de trabajo futuro.
- **Anexos.** Se incluyen diez anexos que contienen la siguiente información: el análisis de los objetivos de aprendizaje de la asignatura Sistemas Operativos de acuerdo con la taxonomía revisada de Bloom, las pruebas de evaluación llevadas a cabo por los estudiantes en el curso, el consentimiento informado que han firmado los estudiantes y el conjunto de códigos que se ha creado en este estudio para descubrir las dificultades de aprendizaje del concepto interrupción.

2. Marco teórico

En este capítulo se hace una breve referencia al marco teórico utilizado en el estudio. En primer lugar se definen los conceptos de evaluación formativa y aprendizaje significativo y después se hace un breve resumen de la taxonomía de objetivos de educación utilizada en este trabajo: la taxonomía revisada de Bloom. Después se proporciona información sobre el área de investigación a la que pertenece esta tesis, educación en informática. Por último, se precisa el propósito de los métodos cualitativos y los resultados que se esperan de un estudio de este tipo.

2.1 Evaluación formativa *versus* evaluación sumativa

Los términos evaluación formativa y evaluación sumativa fueron utilizados por primera vez para explicar los distintos roles ejercidos por la evaluación de un curriculum (Scriven, 1967).

Más adelante, Benjamin Bloom y sus colaboradores aplicaron la misma distinción a la evaluación del aprendizaje de los estudiantes. La evaluación sumativa es el tipo de evaluación utilizada al final de un periodo, curso o programa con el propósito de calificar, certificar, evaluar el progreso o investigar la eficacia de un curriculum o plan educativo. La evaluación formativa es el uso de la evaluación sistemática con el propósito de mejorar uno de estos tres procesos: la construcción del curriculum, la enseñanza y el aprendizaje. Bloom declara que el valor más importante de la evaluación formativa es la ayuda que puede proporcionar a los estudiantes en su proceso de aprendizaje (Bloom, Madaus, & Hastings, 1981).

Bloom distingue la evaluación formativa de la sumativa de acuerdo al propósito que persigue. Es decir, los términos formativa y sumativa no se aplican a las evaluaciones en sí mismas, sino a las funciones que llevan a cabo. De este modo, una prueba de evaluación puede ser utilizada con fines formativos o con fines sumativos. Recientemente, la literatura se refiere a la evaluación sumativa como evaluación del aprendizaje y a la evaluación formativa como evaluación para el aprendizaje (Gikandi, Morrow, & Davis, 2011).

Las recomendaciones de Bloom para la creación de la evaluación formativa han sido ampliadas y enriquecidas con las últimas investigaciones sobre aprendizaje. Con el fin de proporcionar un marco teórico más sólido a la evaluación formativa Wiliam & Thompson distinguen tres procesos fundamentales en la enseñanza y el aprendizaje (Wiliam & Thompson, 2007):

- Establecer en qué punto del aprendizaje están los estudiantes.
- Establecer hacia dónde deben ir.
- Establecer qué se necesita hacer para dirigirlos a ese punto.

A partir de los mencionados procesos, los autores conceptualizan la evaluación formativa a través de cinco estrategias:

1. Clarificar y compartir los objetivos de aprendizaje.
2. Obtener evidencias de la comprensión de los estudiantes.
3. Proporcionar a los estudiantes una retroalimentación para ayudar a conseguir los objetivos de aprendizaje.
4. Fomentar la colaboración entre los estudiantes.
5. Hacer a los estudiantes responsables de su propio aprendizaje.

Las estrategias mencionadas proporcionan el marco teórico de la evaluación formativa que se ha llevado a cabo en esta investigación.

2.2 Aprendizaje significativo

El aprendizaje significativo está basado en la transferencia, que es la habilidad para usar lo aprendido para resolver nuevos problemas, responder nuevas preguntas o facilitar el aprendizaje de una nueva materia. Por el contrario, el aprendizaje memorístico está basado en la retención, que es la habilidad de recordar el material de una forma similar a como fue presentado durante la instrucción (Mayer, 2002).

El concepto de aprendizaje significativo aparece de forma frecuente en los estudios de evaluación formativa (Dopper & Sjoer, 2004) (Chung, Shel, & Kaiser, 2006) (Crisp & Ward, 2008). Este concepto puede adoptar distintos nombres tales como deep learning, learning with Comperending o problem solving en función del marco teórico utilizado por el estudio.

2.3 Taxonomía revisada de Bloom

Las finalidades de la evaluación formativa diseñada para la asignatura objeto de estudio son dos (a) promover el aprendizaje significativo y (b) hacer a los estudiantes conscientes de su proceso de aprendizaje. Para cumplir estos fines se ha usado la taxonomía revisada de Bloom (L. W. Anderson et al., 2000) que permite crear objetivos y pruebas de evaluación alineadas con el propósito de la evaluación.

La taxonomía revisada de Bloom tiene dos dimensiones: procesos cognitivos y tipo de conocimiento. La dimensión de los procesos cognitivos está constituida por seis categorías: *recordar*, *comprender*, *aplicar*, *analizar*, *evaluar* y *crear*. Cada una de las categorías está asociada con dos o más procesos cognitivos específicos. Por ejemplo la categoría *recordar* está asociada con los procesos cognitivos *reconocer* y *recordar*. El aprendizaje memorístico está relacionado con la categoría *recordar*, y el aprendizaje significativo está relacionado con las cinco categorías restantes.

La primera categoría, *recordar*, consiste en recuperar conocimiento de la memoria a largo plazo. La categoría *comprender* consiste en construir significado de los mensajes instruccionales comunicados de forma oral, escrita o

gráfica. Uno de los procesos cognitivos más importantes incluidos dentro de esta categoría es *inferir*, que significa relacionar conocimientos para extraer conclusiones que no se han hecho explícitas durante la instrucción. La categoría *aplicar* consiste en llevar a cabo o usar un procedimiento en una situación determinada. La categoría *analizar* consiste en dividir un material en sus partes constituyentes y determinar cómo estas partes se relacionan una con otra y con la estructura general. La categoría *evaluar* consiste en hacer juicios basados en criterios y estándares. Por último, la categoría *crear* reorganiza elementos en un nuevo patrón o estructura.

Las categorías de los procesos cognitivos son muy importantes en este estudio, ya que permiten diferenciar el aprendizaje significativo del aprendizaje memorístico.

La dimensión del conocimiento incluye cuatro categorías: *factual*, *conceptual*, *procedural* y *metacognitivo*. El conocimiento *factual* consiste principalmente en la terminología usada en una determinada disciplina. Además, también se considera conocimiento *factual* el conocimiento sobre personas, fechas y fuentes de información. El conocimiento *conceptual* se refiere al conocimiento de cuerpos de contenidos extensos y organizados, como conceptos, principios, modelos y teorías. El conocimiento *procedural* es el conocimiento sobre cómo hacer algo. El conocimiento *metacognitivo* es el conocimiento sobre la cognición en general, así como la conciencia y el conocimiento sobre la propia cognición (L. W. Anderson et al., 2000).

El conocimiento *metacognitivo* tiene una presencia muy importante en este estudio, así como en otras investigaciones sobre evaluación formativa (Vonderwell, 2007) (Crisp & Ward, 2008). Los trabajos existentes han estudiado el conocimiento *metacognitivo* de diversas formas como las discusiones *online* (Vonderwell, 2007) o la medición de la confianza de las respuestas de los estudiantes (Crisp & Ward, 2008). En este estudio se han utilizado cuestionarios de preguntas cuya única finalidad es valorar el conocimiento *metacognitivo*.

2.4 La investigación en educación en informática (*Computer Science Education Research*)

Los fundamentos de la investigación en educación en informática proceden de la informática y de las ciencias sociales como la educación, la psicología o la sociología. Las disciplinas de ciencias sociales han utilizado con éxito un conjunto de metodologías de investigación que son poco conocidas en el área de computación. En concreto, esta tesis utiliza una metodología cualitativa, que se utiliza desde la segunda mitad del siglo XIX en disciplinas relacionadas con el comportamiento humano como la educación.

La consolidación de esta área de investigación ha sido posible gracias al apoyo de las principales sociedades profesionales en informática: ACM y IEEE y a la aparición de diversos congresos y revistas científicas.

La sociedad ACM creó en 1968 el *Special Interest Group in Computer Science Education (SIGCSE)* que celebra anualmente su congreso, también conocido como SIGCSE y cuenta con la revista *ACM Transactions on Computing Education*, creada en 2001. Por otra parte, la sociedad de IEEE contribuye a la investigación en educación en informática con su congreso anual *IEEE Frontiers in Education*, que se celebra desde el año 2002, y con la revista *IEEE Transactions on Education*.

Existe otro importante congreso anual, denominado *Koli Calling International Conference on Computing Education Research*, que se caracteriza por publicar trabajos con una sólida base teórica y con métodos de investigación rigurosos. El congreso *Koli Calling* está organizado por la Universidad de Newcastle de Australia y la Universidad Aalto de Finlandia. Ambas universidades cuentan con programas de doctorado de reconocido prestigio en *Computer Science Education Research*.

Entre las revistas científicas más importantes del área de investigación, se encuentran *International Journal of Engineering Education* y *Computers & Education*. Ambas están indexadas en ISI Web of Knowledge y forman parte de la base de datos JCR (*Journal Citation Reports*).

2.5 Métodos cualitativos

La presente tesis utiliza un método de investigación cualitativo: el estudio de caso. Los métodos cualitativos ayudan a comprender y explicar el significado de los fenómenos sociales con la menor modificación posible del entorno natural en el que se producen (Merriam, 1998).

El estudio de caso se define como la investigación intensiva de un único objeto de indagación social, tal y como un aula (Stake, 1978). La ventaja principal de un estudio de caso es que el investigador puede descubrir hechos o procesos que probablemente pasaría por alto si utilizara otros métodos de investigación más superficiales (Wittrock, 1989).

Los informes de los estudios de caso realizan algún tipo de aportación cuando generan proponen nuevos conceptos o sugieren explicaciones innovadoras de los acontecimientos y cuando esas ideas están respaldadas por pruebas presentadas en forma de narraciones (Wittrock, 1989).

3. Trabajos relacionados

Este capítulo refleja el resultado del trabajo bibliográfico dedicado a estudiar aquellos trabajos que otorguen significado a la presente investigación. En primer lugar, se ofrece un análisis exhaustivo de los trabajos acerca de la enseñanza y aprendizaje de sistemas operativos, el tema central de esta tesis. Este análisis constituye la mayor parte del capítulo. Después, se discuten otros trabajos que constituyen una referencia para este estudio por compartir objetivos y/o metodología, aunque no pertenezcan al área de sistemas operativos.

3.1 Enseñanza y aprendizaje de sistemas operativos

La recopilación de los trabajos acerca de la enseñanza y el aprendizaje de sistemas operativos se ha realizado en dos etapas: (1) antes del diseño de la investigación y (2) durante el desarrollo de la investigación y al finalizar, para actualizar la información y contrastar resultados.

Los trabajos estudiados antes del desarrollo de la presente investigación (anteriores al año 2010) permitieron conformar el problema y las preguntas de investigación. Los trabajos estudiados durante y después de la investigación (comprendidos entre los años 2010 y 2014) han respaldado la presente tesis.

El trabajo bibliográfico comenzó con la realización de búsquedas en las principales bases de datos electrónicas:

- ACM Digital Library
- ERIC (Educational Research Information Center)

- IEEE Xplore
- JSTOR
- PLoS Journals (Public Library of Science)
- Rebiun
- SciELO - Scientific Electronic Library *Online*
- ScienceDirect (Elsevier)
- Springer
- Web of Science (Thomson Scientific / ISI Web Services)

En primer lugar, se buscaron artículos que contuvieran la cadena *operating systems* en el título y además tuvieran como palabras clave alguna de las siguientes: *teaching, learning, education, course*. Los resultados obtenidos ayudaron a obtener otros términos para realizar búsquedas adicionales. Una vez agotadas todas las búsquedas, se procedió a examinar los artículos referenciados por cada uno de los trabajos encontrados y a añadir a los resultados aquellos trabajos que no estuvieran ya incluidos. El proceso de búsqueda continuó hasta que no se encontraron nuevos artículos relevantes. La gestión de las referencias encontradas se realizó con la herramienta *Mendeley*.

En cuanto a los criterios de selección de los trabajos, sólo se consideraron fuentes con revisión entre iguales para asegurar la calidad de las publicaciones. El resultado de la búsqueda fueron sesenta y seis trabajos acerca de la enseñanza y aprendizaje de la asignatura Sistemas Operativos.

Una vez recopilados los trabajos se procedió a realizar los siguientes pasos:

1. Analizar los contenidos de las asignaturas de Sistemas Operativos mencionadas en los trabajos para comprobar su grado de similitud.
2. Analizar las fechas de publicación de los artículos para valorar el interés en la enseñanza y aprendizaje de sistemas operativos.

3. Analizar el propósito principal de cada uno de los trabajos para agruparlos y estudiarlos de una forma sistemática.

En los siguientes apartados se exponen los resultados de cada uno de los pasos realizados.

3.1.1 Análisis del contenido de la asignatura Sistemas Operativos

El análisis de los contenidos de las asignaturas de Sistemas Operativos tratadas en todos los trabajos reveló que éstas son homogéneas entre sí y, están de acuerdo con los objetivos y contenidos del curriculum recomendado por ACM/IEEE-CS Joint Task Force (ACM/IEEE-CS Joint Interim Review Task Force, 2008).

Los temas principales de la asignatura de Sistemas Operativos son: gestión, planificación y sincronización de procesos, gestión de memoria, gestión de E/S y gestión de ficheros. Además, en los trabajos analizados se otorga una especial importancia a los siguientes conceptos: procesos, algoritmos de planificación de procesos, llamadas al sistema, interbloqueo, semáforos, memoria virtual y algoritmos de planificación de disco.

3.1.2 Análisis de las fechas de publicación de los trabajos

En cuanto al análisis temporal de los trabajos, se observa que los años de publicación de los artículos analizados comienzan en el año 1983 y terminan en el presente año 2014. La Figura 3.1 muestra el número de artículos publicados en cada año.

La media de artículos publicados por año es de 2,3. El número máximo de artículos publicados en un año ha sido 6 y ocurrió en el año 2005. En los años 1985, 1986, 1988 y 1994 no se publicó ninguno de los trabajos analizados. Sin embargo, la falta de publicación en estos años no supone una discontinuidad en la investigación ya que se durante estos años se ha podido trabajar en propuestas que hayan sido publicadas en años posteriores.

En resumen, el análisis de la dimensión temporal muestra que las publicaciones han sido continuas en el tiempo y que se mantiene vigente la preocupación sobre la enseñanza y el aprendizaje de sistemas operativos.

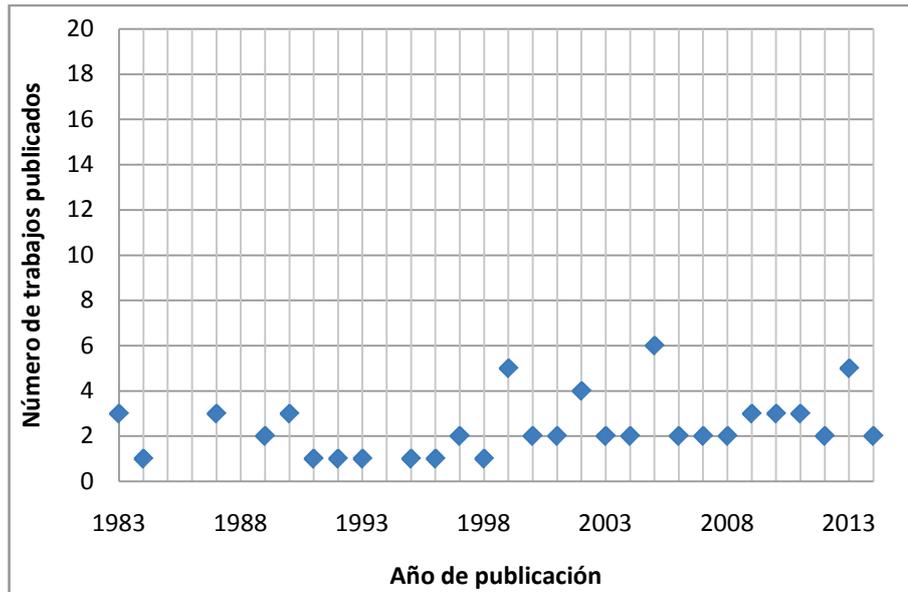


Figura 3.1. Número de trabajos publicados acerca de la enseñanza y aprendizaje de *Sistemas Operativos*

3.1.3 Análisis del propósito principal de los trabajos

En cuanto al propósito de los trabajos, todos comparten el objetivo general de facilitar la enseñanza y/o el aprendizaje, pero lo hacen a través de distintas propuestas. Estas propuestas se han agrupado de acuerdo a las siguientes categorías:

- Presentar un determinado tipo de **actividades prácticas** para la asignatura sistemas operativos. Dentro de esta categoría se distinguen las siguientes opciones:
 - **Proyectos de desarrollo del kernel de un sistema operativo real.** Este tipo de proyectos tienen como objetivo el desarrollo o la modificación de un parte del kernel de un sistema operativo real.
 - **Proyectos de desarrollo del kernel de un sistema operativo instruccional.** Estos proyectos consisten, al igual que en la categoría anterior, en el desarrollo o la modificación del kernel de un sistema operativo. La diferencia clave es que el desarrollo se realiza con un sistema operativo creado únicamente con propósitos docentes, con el objetivo de reducir la complejidad del proyecto.

- **Proyectos de desarrollo que no supongan la modificación de un sistema operativo.** En esta opción tienen cabida distintos tipos de proyectos de desarrollo que no suponen la modificación del kernel de un sistema operativo. Un ejemplo de este tipo de proyectos pueden ser programas que usen las llamadas al sistema de un determinado sistema operativo.
- **Simuladores.** Dentro de esta categoría se consideran todos aquellos programas que permiten explorar de una forma visual el funcionamiento de un sistema operativo.
- **Uso de aplicaciones web.** En esta categoría se agrupan distintas aplicaciones para la enseñanza y el aprendizaje de Sistemas Operativos a través de Internet. Estas propuestas son indicadas para el aprendizaje autónomo del estudiante, ya sea en cursos presenciales o a distancia.
- Usar **técnicas de aprendizaje activo.** Estas técnicas propician una actitud activa del estudiante en clase, en contraste con lo que ocurre en el método expositivo clásico, en el que el estudiante se limita a escuchar y tomar notas. Las propuestas están relacionadas con distintas actividades prácticas que pueden coincidir con las mencionadas en la categoría anterior.
- Usar **juegos para motivar a los estudiantes** al aprendizaje de la asignatura Sistemas Operativos. Este tipo de trabajos presenta juegos desarrollados especialmente para la enseñanza y el aprendizaje de sistemas operativos. El propósito principal de estos juegos es motivar al estudiante.
- Estudiar la **comprensión de los conceptos** de la asignatura Sistemas Operativos. Esta categoría agrupa aquellos trabajos cuyo propósito principal ha sido estudiar las dificultades de los estudiantes para la comprensión de la asignatura.

En los apartados siguientes se describen y analizan los trabajos de cada una de las categorías mencionadas. El análisis ha sido realizado con la herramienta

ATLAS.ti y ha tenido en cuenta los siguientes aspectos de cada uno de los trabajos:

- **Dificultades en la enseñanza y/o el aprendizaje de la asignatura.** Se determina si el trabajo hace referencia a la existencia de dificultades en la enseñanza y/o el aprendizaje de la asignatura Sistemas Operativos.
- **Objetivos de aprendizaje de la asignatura.** Se analiza si el trabajo hace explícitos los objetivos de aprendizaje de la asignatura Sistemas Operativos.
- **Contexto del curso estudiado.** Se especifica si el trabajo ha sido diseñado para un curso con modalidad presencial o con modalidad *online*.
- **Evaluación del aprendizaje de los estudiantes.** En el análisis realizado se señala si el trabajo ha realizado o ha propuesto una forma de evaluar el aprendizaje de los estudiantes. Puesto que el objetivo común de todos los trabajos analizados es mejorar el aprendizaje o la enseñanza de la asignatura Sistemas Operativos, es conveniente establecer una forma de evaluar el aprendizaje para poder comprobar que la mejora ha tenido lugar.
- **Metodología de investigación.** En caso de que los trabajos presentados incluyan un estudio empírico, se examina la metodología de investigación utilizada.

3.1.4 Proyectos de desarrollo de un sistema operativo real

En primer lugar se describen por orden cronológico las propuestas de desarrollo del kernel de sistemas operativos reales y después se realiza un análisis de las mismas.

3.1.4.1 Descripción

En 1995, se presentan varios proyectos sobre el sistema UNIX que incluyen la modificación de los algoritmos de planificación de procesos y de los algoritmos de planificación de disco (Pérez-Dávila, 1995).

Diez años más tarde, se presenta otra experiencia de desarrollo del kernel de un sistema operativo real: Linux (Nieh & Vaill, 2005). A pesar de que han pasado diez años y han aparecido nuevos enfoques para la enseñanza de la asignatura de sistemas operativos, los autores mantienen que los estudiantes aprenden más con proyectos de implementación del kernel de un sistema operativo real.

En los años sucesivos, se proponen proyectos de desarrollo de kernel sobre los siguientes sistemas operativos: iPodLinux, un sistema operativo diseñado para los dispositivos iPod del fabricante Apple (Lawson & Barnett, 2008) y Microsoft Windows (Hwang, 2009).

En el año 2010 se publican de nuevo dos proyectos de desarrollo del kernel de Linux (Hess & Paulson, 2010) (Laadan, Nieh, & Viennot, 2010). Este último trabajo citado propone una mejora con respecto a los trabajos anteriores: el uso de un entorno de desarrollo virtual en el que los sistemas operativos pueden desarrollados, depurados y arrancados sin afectar al resto de los usuarios.

En los años 2011 (Laadan, Nieh, & Viennot, 2011) y 2012 (Andrus & Nieh, 2012) se presentan proyectos sobre los sistemas operativos Linux y Android.

El último de los trabajos presenta una herramienta para el trabajo con proyectos de desarrollo del kernel de sistemas operativos reales. La herramienta se llama *GradeBoard* y tiene dos propósitos principales: (1) facilitar la calificación a los profesores y (2) facilitar a los estudiantes la comprensión y aprendizaje de sus errores (Dall & Nieh, 2014). Este trabajo se diferencia del resto de trabajos de esta categoría en que es el único que establece una forma de evaluar el aprendizaje de los estudiantes.

3.1.4.2 Análisis

Los sistemas operativos reales propuestos para realizar proyectos de desarrollo del kernel han sido *UNIX*, *Linux*, *iPodLinux* y *Android*. Se han analizado un total de 9 trabajos en esta categoría. El primer trabajo de este tipo fue presentado en 1995 y el último en 2012. Por tanto, se puede afirmar que este tipo de propuesta se sigue considerando una opción válida para la enseñanza de sistemas operativos en el momento de escritura de esta tesis (2014). La Tabla 3.1 recoge una síntesis de los distintos aspectos analizados sobre este tipo de propuestas.

Los valores reflejados en la tabla son los siguientes:

- Columna *Autor(es) y año*. Se cita el trabajo en formato APA sexta edición.
- Columna *Menciona dificultades*: Puede tener los siguientes valores:
 - *No*, cuando el trabajo no menciona que existan dificultades para enseñar o aprender Sistemas Operativos.
 - *En la enseñanza*, cuando el trabajo menciona que existen dificultades en la enseñanza de Sistemas Operativos.
 - *En el aprendizaje*, cuando el trabajo menciona que existen dificultades en el aprendizaje de Sistemas Operativos.
 - *En la enseñanza y el aprendizaje*, cuando el trabajo menciona que existen dificultades en el aprendizaje de Sistemas Operativos.
- Columna *Establece objetivos de aprendizaje*: Puede tener dos valores:
 - *Sí*, cuando el trabajo establece de forma explícita los objetivos de aprendizaje.
 - *No*, cuando el trabajo no establece de forma explícita los objetivos de aprendizaje.
- Columna *Evalúa el aprendizaje*: Puede tener tres valores:
 - *No*, cuando el trabajo no evalúa el aprendizaje de los estudiantes.
 - *Encuesta de satisfacción*, cuando el aprendizaje de los estudiantes es evaluado mediante una encuesta de satisfacción.
 - *Prueba de evaluación*, cuando el aprendizaje de los estudiantes es evaluado mediante una prueba de evaluación.
- Columna *Metodología de investigación*: Puede tener tres valores:
 - *Vacío*. Cuando no el trabajo no realiza un estudio empírico.

- *Metodología cuantitativa.* Cuando el trabajo utiliza una metodología de investigación cuantitativa.
- *Metodología cualitativa.* Cuando el trabajo utiliza una metodología de investigación cualitativa.
- *Columna Modalidad curso.* Puede tener tres valores:
 - *Presencial.* Cuando el estudio se ha realizado en un entorno presencial.
 - *Online.* Cuando el estudio se ha realizado en un entorno *online*.
 - *Presencial/online.* Cuando el estudio se ha realizado en ambos entornos, *online* y presencial.

En cuanto a las dificultades mencionadas sobre la docencia en sistemas operativos, cuatro de los nueve trabajos mencionan que hay dificultades. Dos de ellos afirman que hay dificultades para enseñar la asignatura sin precisar el tipo de dificultades. Los dos restantes destacan que hay dificultades tanto en la enseñanza como en el aprendizaje de la asignatura. A continuación se describen las dificultades mencionadas por cada uno de estos dos trabajos.

En (Laadan et al., 2010) se señala que los sistemas operativos en producción son un tipo de software muy complejo y con muchas líneas de código y que estas características pueden dificultar la comprensión de los mismos. Es decir, exponen los problemas asociados con el tipo de solución que propones. De forma consecuente con los problemas expuestos, los autores proponen proyectos con una complejidad adecuada para los estudiantes.

En (Dall & Nieh, 2014) se abordan otras dos dificultades del desarrollo del kernel de sistemas operativos reales. La primera es la dificultad para calificar las actividades, que afecta a los profesores. La segunda es la dificultad para la comprensión de los errores cometidos en la programación, que afecta a los estudiantes.

Tabla 3.1. Análisis de los trabajos acerca de proyectos de desarrollo de un sistema operativo real

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Pérez-Dávila, 1995)	En la enseñanza	No	No	—	Presencial
(Nieh & Vaill, 2005)	No	Sí	No	—	Presencial/ <i>online</i>
(Lawson & Barnett, 2008)	No	No	No	—	Presencial
(Hwang, 2009)	No	No	No	—	Presencial
(Hess & Paulson, 2010)	En la enseñanza	No	Encuesta de satisfacción.	Metodología cuantitativa	Presencial
(Laadan et al., 2010)	No	No	No	—	Presencial
(Laadan et al., 2011)	En la enseñanza y en el aprendizaje	No	No	—	Presencial
(Andrus & Nieh, 2012)	No	No	Encuesta de satisfacción.	Metodología cuantitativa	Presencial
(Dall & Nieh, 2014)	En la enseñanza y en el aprendizaje	No	Prueba de evaluación	Metodología cuantitativa	Presencial/ <i>online</i>

En cuanto a los objetivos de aprendizaje que se intentan facilitar con la propuesta, sólo hay un trabajo que los hace explícitos. El trabajo de (Nieh & Vaill, 2005) deja claro que su objetivo final es la comprensión de los principios de los sistemas operativos y no el aprendizaje del desarrollo de un kernel de un sistema operativo. El resto de trabajos no establece ningún objetivo de aprendizaje. Por tanto, sin un objetivo claro, se podría interpretar que el objetivo del curso es aprender a programar sistemas operativos.

Con referencia a la evaluación del aprendizaje, sólo hay un trabajo que se preocupa de este punto, proponiendo el uso de la herramienta *Gradeboard* (Dall & Nieh, 2014), que tiene dos objetivos: (1) que el profesor pueda verificar si lo realizado por el estudiante es correcto y (2) que el estudiante aprenda de sus errores. Otros dos trabajos no evalúan exactamente el aprendizaje, sino la

satisfacción de los estudiantes y lo hacen a través de una encuesta. Este enfoque tiene el problema de que las percepciones del estudiante pueden no coincidir con su aprendizaje real.

Sólo tres de los nueve trabajos han realizado algún tipo de estudio empírico. Los tres han adoptado un enfoque exclusivamente cuantitativo.

Por último, dos de los nueve trabajos presentan propuestas diseñadas tanto para entornos presenciales como para entornos *online*.

3.1.5 Proyectos de desarrollo de un sistema operativo instruccional

En este apartado se ha tomado como referencia la siguiente definición de sistema operativo instruccional: es aquel diseñado especialmente para la enseñanza y/o aprendizaje (C. L. Anderson & Nguyen, 2005). En primer lugar se describen por orden cronológico las propuestas de desarrollo del kernel de sistemas operativos instruccionales y después se realiza un análisis de las mismas.

3.1.5.1 Descripción

Los primeros trabajos (Shub, 1983) (Fornaro, Garrard, & Uzzle, 1983) (Hayashi, 1983) (Schiper, Dalang, Egli, Mattini, & Simon, 1984) están centrados en sistemas operativos instruccionales que permitan programar un sistema operativo en un lenguaje de alto nivel. Estas propuestas aparecen como respuesta a la complejidad de la programación de sistemas operativos en ese momento, que debe hacerse en ensamblador y debe probarse directamente con el hardware.

En 1987 Tanenbaum publica un trabajo sobre MINIX, un sistema operativo instruccional creado por él mismo (Tanenbaum, 1987). MINIX tiene las mismas llamadas al sistema que la versión 7 de UNIX y corre sobre un hardware real. En este mismo año surge otra propuesta de sistema operativo instruccional, en este caso sobre hardware simulado: las máquinas HAL y OGRE (Wolfe, 1987).

En el año 1989 aparecen dos trabajos sobre los sistemas operativos MPX-PC (Lane & Ghosal, 1989) y XINU WU (Hummel, 1989), una nueva versión del sistema operativo Xinu creado en 1984 y todavía vigente en la actualidad (Comer, 2011).

En 1990 se publican dos experiencias de proyectos de programación de sistemas operativos en lenguaje C (Reek, 1990) (Donaldson, 1990).

En el siguiente año se propone un entorno de programación concurrente basado en semáforos que simula las llamadas al sistema de UNIX System V (Higginbotham & Morelli, 1991).

En 1993 aparece el sistema operativo Nachos que incluye simuladores de CPU y dispositivos y corre como un proceso UNIX (Christopher, Procter, & Anderson, 1993).

Otra propuesta educativa interesante es la creación del entorno de programación concurrente BACI que constituye una alternativa al uso de las llamadas al sistema de un sistema operativo real (Bynum & Camp, 1996).

En 1999, se publica el Proyecto TINA, un proyecto dedicado al desarrollo de experimental de kernel. TINA es un sistema operativo que corre sobre el simulador de hardware MPS (Morsiani & Davoli, 1999).

Los siguientes sistemas operativos instruccionales presentados fueron OS/161 (Holland, Lim, & Seltzer, 2002), PortOS (Atkin & Sirer, 2002) y GeekOS (Hovemeyer, Hollingsworth, & Bhattacharjee, 2004).

En 2007, se publica un trabajo acerca de un nuevo sistema operativo instruccional: BabyOS (Liu, Chen, & Gong, 2007). Este proyecto se diferencia de los anteriores en que incluye conceptos de diseño de sistemas operativos embebidos.

En 2009, se presentan dos proyectos. El primero es sobre la programación completa de un sistema operativo sencillo que corra en un hardware real (Black, 2009). El segundo es acerca de Pintos, un sistema operativo instruccional con una nueva característica: puede funcionar tanto en un hardware real como en un hardware simulado (Pfaff, Romano, & Back, 2009).

En 2012 se publica un trabajo sobre PennOS (Aviv et al., 2012). Este sistema operativo instruccional corre como una aplicación de usuario, no está preparado para ejecutarse sobre un hardware real o simulado. Sin embargo permite realizar proyectos tales como la creación de un shell y un planificador de procesos.

Por último, en 2012 surge una propuesta para desarrollar un kernel basada en threads. El entorno creado permite que los estudiantes desarrollen el kernel de un sistema operativo de forma incremental (Manacero & Lobato, 2013).

Entre los trabajos sobre enseñanza de sistemas operativos, hay dos publicaciones especiales dedicadas al estudio de los sistemas operativos instruccionales disponibles. La primera es acerca de una discusión realizada sobre este tema en un panel celebrado en la edición 30 del importante congreso SIGCSE (Special Interest Group on Computer Science Education). En este trabajo se analizan algunos sistemas operativos instruccionales que no han sido mencionados hasta ahora en este apartado: CHIP/OCA, OSP, Topsy, KMOS, AcadOS, TUNIS (Goldweber, Barr, Camp, Grahm, & Hartley, 1999).

La segunda publicación (C. L. Anderson & Nguyen, 2005) es un estudio sobre sistemas operativos instruccionales. Una característica importante del trabajo es que no se limita a una búsqueda bibliográfica, sino que también analiza el enfoque de enseñanza de sistemas operativos utilizado por noventa y ocho universidades. El estudio concluye que no hay un sistema operativo instruccional dominante. Se ofrecen a continuación los sistemas operativos mencionados en el estudio citado que no han sido mencionados anteriormente en este apartado: Bosch y JOS.

3.1.5.2 Análisis

De acuerdo a los trabajos expuestos en este apartado existen dos tipos de sistemas operativos instruccionales:

- Los que son programados totalmente por el estudiante. En este caso los trabajos proponen un entorno de desarrollo lo más sencillo posible para llevar a cabo esta tarea.
- Los que han sido ya programados total o parcialmente y se ofrecen a los estudiantes para su estudio o modificación. Dentro de este tipo de sistemas operativos instruccionales podemos distinguir tres tipos: los que corren sobre hardware real, los que corren sobre hardware simulado y los que son capaces de correr tanto en hardware real como en hardware simulado. Los sistemas operativos que corren sobre hardware simulado presentan menos dificultades para los estudiantes, puesto que ocultan la complejidad del hardware real.

En este apartado se han presentado 32 sistemas operativos instruccionales en el periodo comprendido desde el año 1983 hasta el año 2013. Por tanto, es posible afirmar que el enfoque de desarrollar un sistema operativo instruccional sigue vigente en la enseñanza de sistemas operativos.

La Tabla 3.2 recoge los distintos aspectos analizados sobre las propuestas de sistemas operativos instruccionales.

Tabla 3.2. Análisis de los trabajos acerca de sistemas operativos instruccionales

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Shub, 1983)	No	No	No	___	Presencial
(Fornaro et al., 1983)	No	Sí	No	___	Presencial
(Hayashi, 1983)	En la enseñanza	No	No	___	Presencial
(Schiper et al., 1984)	En la enseñanza	Sí	No	___	Presencial
(Tanenbaum, 1987)	En la enseñanza	No	No	___	Presencial
(Wolfe, 1987)	No	No	No	___	Presencial
(Lane & Ghosal, 1989)	No	No	No	___	Presencial
(Hummel, 1989)	En la enseñanza	No	No	___	Presencial
(Reek, 1990)	En la enseñanza	Sí	No	___	Presencial
(Donaldson, 1990)	En la enseñanza	Sí	No	___	Presencial
(Higginbotham & Morelli, 1991)	En la enseñanza	No	No	___	Presencial
(Christopher et al., 1993)	No	No	No	___	Presencial
(Bynum & Camp, 1996)	En la enseñanza	No	No	___	Presencial
(Morsiani & Davoli, 1999)	No	No	No	___	Presencial
(Holland et al., 2002)	No	No	No	___	Presencial
(Atkin & Sirer, 2002)	No	No	No	___	Presencial

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Hovemeyer et al., 2004)	No	Sí	No	—	Presencial
(Liu et al., 2007)	En la enseñanza	No	No	—	Presencial
(Black, 2009)	En la enseñanza	No	No	—	Presencial
(Pfaff et al., 2009)	En la enseñanza	No	No	—	Presencial
(Aviv et al., 2012)	En la enseñanza	No	No	—	Presencial
(Manacero & Lobato, 2013)	En la enseñanza	No	No	—	Presencial

Las dificultades mencionadas por estos trabajos están centradas en la enseñanza, en la búsqueda de herramientas que permitan a los estudiantes practicar con un sistema operativo, pero sin exponerlos a la complejidad de los sistemas operativos reales.

En cuanto a los objetivos de aprendizaje, sólo cuatro de los trabajos los mencionan de forma explícita. Todos ellos coinciden en que el objetivo principal del curso es la comprensión de los conceptos de sistemas operativos. Como en el caso anterior, en los trabajos que no mencionan los objetivos se podría interpretar que el objetivo del curso es aprender a programar sistemas operativos.

Como se puede comprobar en la tabla, en ninguna de las propuestas de sistemas operativos instruccionales se evalúa el aprendizaje de los estudiantes ni se realiza ningún tipo de estudio empírico. Tampoco se menciona en ninguno la posibilidad de utilizar esta propuesta en cursos *online*.

Los trabajos sobre sistemas operativos instruccionales parecen estar centrados en la presentación técnica del sistema operativo, están centrados en la tecnología más que en su uso educativo.

3.1.6 Otros proyectos de desarrollo

En este apartado se agrupan proyectos de desarrollo que no suponen la programación del kernel de un sistema operativo, ya sea uno real o uno creado

a propósito de la instrucción. A continuación se describen por orden cronológico las propuestas asociadas con este enfoque y se realiza un análisis de las mismas.

3.1.6.1 Descripción

En 1993, 1997 y 1998 se publican tres propuestas muy similares. Los trabajos describen proyectos de desarrollo sobre sistemas operativos que incluyen el uso de llamadas al sistema y el acceso a estructuras de datos de un sistema operativo (Ramakrishnan & Lancaster, 1993) (Wagner & Ressler, 1997) (O’Gorman, 1998).

En 1997 se presenta también otro proyecto del mismo tipo (Holliday, 1997). El factor que lo diferencia de los trabajos anteriores es que intenta dar una visión global del funcionamiento de un sistema operativo. Para ello pone el énfasis en dos conceptos: las llamadas al sistema y las interrupciones. En cuanto a las llamadas al sistema, el objetivo del autor es que los estudiantes comprendan la abstracción definida por la interfaz de llamadas al sistema e implementada por el kernel. Por otra parte, el objetivo del autor en cuanto a las interrupciones es que los estudiantes comprendan la secuencia concreta de eventos que ocurre cuando se produce una interrupción y que es la esencia del funcionamiento del kernel.

Otro enfoque relacionado con programación, que no supone la modificación de un sistema operativo es el análisis del código de programas cortos con un resultado inesperado para el estudiante (Ziegler, 1999). El efecto sorpresa motiva al estudiante a descubrir las causas del comportamiento de los programas y profundizar en los conceptos de la asignatura.

En 1999 se presentan varios proyectos de programación que tienen como finalidad la realización de experimentos para medir el rendimiento de un sistema operativo (Downey, 1999). Esta propuesta introduce el diseño experimental como parte del material del curso al igual que la propuesta de simuladores de Robbins (Robbins & Robbins, 1999) que se describe en un apartado posterior. Los estudiantes realizan pequeños proyectos de programación con el propósito de llevar a cabo una serie de experimentos para medir el rendimiento de los servicios del sistema e inferir información sobre su implementación a partir de los resultados.

Por último, en 2011 se publica una propuesta (Desnoyers, 2011) que también pone el énfasis en el funcionamiento global de un sistema operativo como el trabajo descrito en los párrafos anteriores (Holliday, 1997). El propósito del curso no es comprender cómo está estructurado un sistema operativo ni cuáles son sus principales conceptos. El curso intenta enseñar cómo funciona un computador, es decir, qué eventos ocurren en respuesta a determinados estímulos.

Un ejemplo de objetivos de aprendizaje de esta última propuesta sería que los estudiantes sepan explicar los eventos que suceden cuando un usuario ejecuta el comando `ls` en un computador. En particular, los estudiantes deben comprender el funcionamiento del driver del teclado y del planificador para entregar la entrada al shell, las operaciones de la memoria virtual necesarias para traer a memoria el ejecutable del comando `ls` y las acciones necesarias para localizar un determinado directorio y mostrar sus entradas.

3.1.6.2 Análisis

La filosofía de este enfoque se basa en que los conceptos son la parte más importante del curso de sistemas operativos y la práctica debe estar destinada a reforzar esta teoría y no a programar un sistema operativo.

Los trabajos de este tipo comienzan en el año 1993 y finalizan en el año 2011. Por tanto, es posible afirmar que este enfoque está todavía vigente en la enseñanza de sistemas operativos.

La Tabla 3.3 recoge una síntesis de los distintos aspectos analizados sobre las siete propuestas de esta categoría.

Las dificultades referidas por estos trabajos están relacionadas tanto con la enseñanza como con el aprendizaje. Incluso el último trabajo, menciona las dificultades que existen para la evaluación de los estudiantes. De acuerdo con los autores, las propuestas presentadas entrañan menor dificultad para los estudiantes que los proyectos en los que hay que desarrollar el kernel de un sistema operativo. Por tanto, siempre que el objetivo del curso sea la comprensión de conceptos, este tipo de propuestas pueden suponer un aprendizaje más eficiente que las propuestas comentadas anteriormente.

Tabla 3.3. Análisis de los trabajos acerca de otros proyectos de desarrollo

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Ramakrishnan & Lancaster, 1993)	En el aprendizaje	Sí	No	—	Presencial
(Wagner & Ressler, 1997)	En la enseñanza	Sí	No	—	Presencial
(O’Gorman, 1998)	No	No	No	—	Presencial
(Holliday, 1997)	En el aprendizaje	Sí	No	—	Presencial
(Ziegler, 1999)	En la enseñanza y en el aprendizaje	No	No	—	Presencial
(Downey, 1999)	No	Sí	No	—	Presencial
(Desnoyers, 2011)	En la enseñanza	Sí	Prueba de evaluación	—	Presencial

En cuanto a los objetivos de aprendizaje, la mayoría de los trabajos los hacen explícitos. En los párrafos siguientes se describen estos objetivos de aprendizaje:

- Los autores de una de las (Ramakrishnan & Lancaster, 1993) pretenden que los estudiantes, a través de la experiencia de programación, comprendan los conceptos tratados en las clases, perciban la complejidad de un sistema operativo y comprendan mejor su experiencia como usuarios de sistemas operativos.
- En (Wagner & Ressler, 1997) se afirma que la filosofía que soporta el enfoque propuesto es que los conceptos de sistemas operativos son la parte más importante del curso.
- Downey (Downey, 1999) afirma que idealmente el trabajo que los estudiantes hacen fuera de clase debería ayudarlos a entender los conceptos tratados en el curso, exponerlos a sistemas operativos y entornos de desarrollo que vayan a encontrar en el futuro y permitirles desarrollar habilidades que necesiten en el futuro para trabajar o investigar en informática.

- La propuesta de Desnoyers insiste en la idea de que su curso de sistemas operativos no explica cómo se estructura un sistema operativo ni cuáles son sus principales conceptos, lo que intenta enseñar es cómo funciona un sistema operativo (Desnoyers, 2011). Por ejemplo, el propósito de una clase puede ser que los conozcan los pasos involucrados en la carga de un programa, la invocación de las funciones del sistema operativo y el cambio de contexto

Por tanto, la mayoría de los trabajos coinciden en que el objetivo del curso es la comprensión de conceptos. Dos de los trabajos van incluso un paso más allá, ya que su objetivo es que el estudiante comprenda el funcionamiento global de un sistema operativo.

En cuanto a la evaluación, sólo hay un trabajo que se preocupa de este punto (Desnoyers, 2011). El autor advierte que la evaluación de la comprensión de los estudiantes no es sencilla ya que si un estudiante realiza un programa que funciona de modo esperado no demuestra que el estudiante haya comprendido los conceptos subyacentes. Lo mismo ocurre con la realización de los problemas, que el estudiante llegue a la solución correcta no implica la comprensión.

Por último, todas las propuestas se han realizado para entornos presenciales, no hay ninguna experiencia *online* dentro de esta categoría.

3.1.7 Uso de simuladores de sistemas operativos

En este apartado se agrupan todas aquellas soluciones que usan herramientas que permiten visualizar de forma gráfica, los detalles implicados en el funcionamiento de un sistema operativo, y no sólo los resultados finales. Estas herramientas son referidas en la literatura como simuladores y ponen el énfasis en la interfaz de usuario de la aplicación, puesto que su objetivo principal es la visualización. A continuación se describen por orden cronológico las propuestas de enseñanza y aprendizaje de sistemas operativos con simuladores y se analiza cada una de ellas.

3.1.7.1 Descripción

Antes de comenzar con la descripción de los principales simuladores, se advierte al lector de que en algunas ocasiones la literatura llama simuladores a los sistemas operativos instruccionales, ya que su intención es “simular” el comportamiento de un sistema operativo e incluso “simular” un hardware determinado. Este tipo de herramientas han sido excluidas de este apartado, ya que han sido tratadas en un apartado anterior (3.1.5 Proyectos de desarrollo de un sistema operativo instruccional).

El primer simulador que se encuentra en la literatura analizada es GRAPHOS (Cañas, 1987). Su creación se basó en la idea de que una representación gráfica del flujo de la información podría mostrar de forma clara a los estudiantes qué está pasando en el sistema operativo en cada momento. Los conceptos ilustrados por el simulador son: procesos concurrentes, planificación de procesos y gestión de memoria.

En 1990, se publica un trabajo acerca de PRMS, un simulador gráfico que fue desarrollado para ilustrar conceptos importantes de sistemas operativos tales como la creación de procesos, la transición de estado de los procesos, el cambio de contexto, la planificación de procesos, la planificación de recursos, la gestión de interrupciones y la detección y recuperación de interbloqueos (Hays, Miller, Othmer, & Saeed, 1990).

En 2002, aparece otro simulador llamado RCOS.java (Jones, 1996). Su característica diferenciadora es que está realizado con lenguaje Java y corre en un navegador web.

El conjunto más numeroso de simuladores ha sido creado por Steve Robbins. Sus ocho simuladores corren sobre un navegador web y permiten realizar experimentos y explorar el comportamiento de distintas partes de un sistema operativo. Cada uno de los simuladores diseñados aborda un importante concepto de sistemas operativos: planificación de procesos (Robbins & Robbins, 1999), comunicación y sincronización de procesos (Robbins, 2000, 2001, 2002), planificación de disco (Robbins, 2004), memoria virtual (Robbins, 2005), entrada/salida (Robbins, 2006) y ejecución de programas (Robbins, 2007).

En 2003 se desarrolla un nuevo simulador llamado SOsim para el aprendizaje de sistemas operativos con una característica que lo diferencia de los anteriores: su facilidad de uso (L P Maia & Pacheco Jr., 2003) (Luiz Paulo Maia, Machado, & Pacheco Jr., 2005). Los simuladores realizados hasta el momento requerían conocimientos de UNIX y/o programación, o tenían interfaces de usuario muy complejas. El objetivo principal de SOsim es presentar los conceptos y las técnicas de los sistemas operativos modernos, tales como multiprogramación, procesos, planificación y gestión de memoria.

Uno de los simuladores más importantes en la actualidad es YASS (Mustafa, 2011, 2013). Este simulador sirve para la enseñanza de dos disciplinas estrechamente relacionadas entre sí: sistemas operativos y estructura de computadores. El simulador corre únicamente sobre los sistemas operativos Windows XP, Windows Vista y Windows 7.

Por último, en 2013 se publica un trabajo sobre un simulador dedicado exclusivamente a la gestión de memoria: Alg_OS (Garpis, 2013). El simulador corre sobre un navegador web y está especialmente dedicado a mostrar el funcionamiento de los algoritmos de reemplazo de páginas.

3.1.7.2 Análisis

Los simuladores intentan ofrecer una forma de aprender sistemas operativos más sencilla que las soluciones presentadas anteriormente, centradas en el desarrollo de software.

Los trabajos acerca de simuladores comienzan en el año 1987 y finalizan en el año 2013. Por tanto, es posible afirmar que este enfoque está todavía vigente en la enseñanza de sistemas operativos.

La siguiente tabla recoge una síntesis de los distintos aspectos analizados sobre las dieciséis propuestas de esta categoría.

Con referencia a las dificultades mencionadas, la casi totalidad de los trabajos indican que existen dificultades en la enseñanza y aprendizaje de sistemas operativos. Los trabajos establecen que una posible causa de las dificultades de aprendizaje es que los estudiantes no pueden visualizar el comportamiento de un sistema operativo, no pueden saber qué está sucediendo en cada momento.

Tabla 3.4. Análisis de los trabajos acerca de simuladores de sistemas operativos

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Cañas, 1987)	En el aprendizaje	Sí	No	—	Presencial
(Hays et al., 1990)	En el aprendizaje	No	No	—	Presencial
(Jones, D., 2002)	En la enseñanza y en el aprendizaje	No	No	—	Presencial
(Robbins & Robbins, 1999)	No	Sí	No	—	Presencial
(Robbins, 2000)	En el aprendizaje	Sí	No	—	Presencial
(Robbins, 2001)	En el aprendizaje	Sí	No	—	Presencial
(Robbins, 2002)	En la enseñanza	Sí	No	—	Presencial
(Robbins, 2004)	No	No	No	—	Presencial
(Robbins, 2005)	En la enseñanza	Sí	Prueba de evaluación	—	Presencial
(Robbins, 2006)	En la enseñanza	Sí	No	—	Presencial
(Robbins, 2007)	En la enseñanza	Sí	No	—	Presencial
(L P Maia & Pacheco Jr., 2003)	En la enseñanza y en el aprendizaje	Sí	Encuesta de satisfacción.	Metodología cuantitativa	Presencial/ <i>online</i>
(Luiz Paulo Maia, Machado, & Pacheco Jr., 2005)	En la enseñanza	Sí	Encuesta de satisfacción.	Metodología cuantitativa	Presencial/ <i>online</i>
(Mustafa, 2011)	En la enseñanza	Sí	Encuesta de satisfacción	Metodología cuantitativa	Presencial
(Mustafa, 2013)	No	No	No	—	Presencial
(Garpmpis, 2013)	En el aprendizaje	Sí	Encuesta de satisfacción	Metodología cuantitativa	Presencial

En cuanto a los objetivos de aprendizaje, la gran mayoría de los trabajos los hacen explícitos y afirman que el objetivo del curso es la comprensión de conceptos. Por ejemplo, la experimentación con el simulador de planificación de

procesos creado por Robbins facilita la comprensión del planificación de procesos incluyendo el funcionamiento de los diferentes algoritmos y los parámetros que influyen en su rendimiento (Robbins & Robbins, 1999).

En cuanto a la evaluación, sólo hay cuatro trabajos que hayan evaluado el aprendizaje de los estudiantes. Todos ellos lo han realizado a través de una encuesta de satisfacción que se ha analizado desde un punto de vista cuantitativo. Como ya se ha mencionado los resultados de estas encuestas podrían no coincidir con el aprendizaje real.

Por último, en dos trabajos se menciona que los simuladores presentados son adecuados también para la educación *online*.

3.1.8 Uso de aplicaciones web

A continuación se describen y analizan distintas propuestas de aplicaciones web para la enseñanza y el aprendizaje de sistemas operativos. Estas propuestas tienen en común que son adecuadas para el aprendizaje autónomo del estudiante, ya sea en cursos presenciales o a distancia.

3.1.8.1 Descripción

En 1996 se presenta una herramienta web destinada al aprendizaje de sistemas operativos: CALOS (*Computer-Aided Learning Operating Systems*). El material del curso consiste en ejercicios interactivos, simulaciones interactivas, mecanismos de comunicación entre profesor y estudiante, mecanismos de comunicación entre estudiantes, evaluaciones para el estudiante, un glosario y una bibliografía (Goldberg, 1996).

Un año más tarde, se comunican los resultados de un experimento para evaluar la mencionada herramienta web. El propósito del experimento fue evaluar la herramienta en términos de rendimiento académico y de aceptación del estudiante. Los estudiantes del curso fueron divididos en tres grupos: 1) los que sólo realizaron el curso web, 2) los que sólo asistieron a las clases sin usar el curso web y 3) los que asistieron a las clases y además usaron el curso web. Los resultados del experimento señalaron que el curso web era equivalente al curso presencial en términos de rendimiento académico y de aceptación del estudiante. Por otra parte, la media del rendimiento académico de los

estudiantes del grupo que utilizó ambos métodos (clases presenciales y curso web) fue superior a la de los otros grupos (Goldberg, 1997a).

El creador de esta aplicación web, Murray Goldberg (Universidad de British Columbia, Canadá), alentado por el éxito del curso de sistemas operativos diseñó un entorno de autor para cursos basados en la web llamado WebCT (Goldberg, 1997b). WebCT se convirtió en el primer sistema de gestión de cursos para educación superior y todavía es una de las plataformas más usadas en e-learning en la actualidad. Ahora la herramienta se conoce con el nombre de Blackboard Learn.

En 2005 aparece otro trabajo (Félix Buendía, Cano, & Sahuquillo, 2005) que describe dos herramientas que permiten la puesta en práctica de conceptos de sistemas operativos a partir del uso de aplicaciones Web. La primera representa un simulador utilizado para mostrar el funcionamiento de los principales aspectos relacionados con la gestión de memoria en un sistema operativo, de forma que el alumno pueda interactuar con el simulador, introduciendo ejemplos de carga y configurando parámetros del sistema. La segunda herramienta permite el acceso remoto a funciones de un sistema operativo real a través de un interfaz Web.

En 2006, se publican los resultados del uso de la herramienta WebgeneOS, un sistema web que permite a los estudiantes enviar sus programas usando formularios web, ejecutarlos en un sistema operativo nativo y recibir los resultados de la ejecución (Felix Buendía & Cano, 2006). Los beneficios de la herramienta fueron evaluados mediante una comparación de las calificaciones de los estudiantes. Los estudiantes que usaron WebgeneOS mejoraron sus calificaciones en un 13%.

3.1.8.2 Análisis

Se han analizado cuatro trabajos acerca de aplicaciones web. La Tabla 3.5 recoge los distintos aspectos analizados sobre los mismos.

Dos de los trabajos exponen que existen dificultades con la enseñanza de sistemas operativos y uno de ellos hace especial mención de la falta de aplicaciones para enseñar y aprender sistemas operativos en cursos *online* (Felix Buendía & Cano, 2006).

Los autores de la herramienta web WebgeneOS (Felix Buendía & Cano, 2006) establecen en su artículo lo que los estudiantes deben aprender en la asignatura Sistemas Operativos. Específicamente, los estudiantes tienen que aprender los principios de funcionamiento de un sistema operativo, cuáles son los principales componentes de un sistema operativo, cómo el sistema operativo actúa de intermediario entre la máquina y el usuario, cómo son gestionados los recursos del sistema (CPU, memoria, dispositivos de almacenamiento) y cómo el sistema operativo proporciona comandos para interactuar con los computadores. El resto de los trabajos no hacen explícitos los objetivos del curso estudiado.

En cuanto a la metodología de investigación utilizada, los dos trabajos que han realizado un estudio empírico han realizado un experimento controlado.

Por último, en todos los trabajos se indica que las aplicaciones web presentadas pueden utilizarse tanto en cursos presenciales como en cursos *online*.

Tabla 3.5. Análisis de los trabajos acerca de aplicaciones web

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Goldberg, 1996)	No	No	No	—	Presencial/ <i>online</i>
(Goldberg, 1997a)	No	No	Prueba de evaluación	Metodología cuantitativa	Presencial/ <i>online</i>
(Félix Buendía et al., 2005)	En la enseñanza	No	No	—	Presencial/ <i>online</i>
(Felix Buendía & Cano, 2006)	En la enseñanza <i>online</i>	Sí	Prueba de evaluación	Metodología cuantitativa	Presencial/ <i>online</i>

3.1.9 Utilización de técnicas de aprendizaje activo

En este apartado se describen y analizan los trabajos que han presentado soluciones relacionadas con distintas técnicas de aprendizaje activo. El aprendizaje activo consiste en que los estudiantes realicen actividades durante las clases en lugar de escuchar al profesor y tomar notas (Lincke, 2005).

3.1.9.1 Descripción

En 2005 se presenta una propuesta de pequeños proyectos de programación junto con la técnica de aprendizaje activo (Lincke, 2005).

En 2008, se publica un trabajo acerca de la puesta en marcha de técnicas de cooperative learning (jigsaw) para realizar prácticas con los simuladores de Robbins, mencionados en un apartado anterior (Jorge E. Pérez, García Martín, & Muñoz Fernández, 2008).

En 2010, se publican los resultados de un importante trabajo sobre el uso de técnicas de aprendizaje activo en la asignatura Sistemas Operativos (Jorge Enrique Pérez, García, Muñoz, & Sierra, 2010). El estudio tuvo como objetivo evaluar los efectos de las metodologías de aprendizaje activo en la adquisición de competencias genéricas. En concreto, se pusieron en práctica dos tipos de metodologías (aprendizaje cooperativo y aprendizaje basado en problemas), y se estudió su efecto en el rendimiento académico y en dos competencias genéricas (trabajo en grupo y gestión del tiempo).

El enfoque original del mencionado trabajo consistió en medir el nivel de competencias adquiridas mediante test psicométricos en lugar de encuestas como suele hacerse de forma habitual. Los resultados mostraron que las metodologías activas utilizadas tuvieron un efecto positivo en el rendimiento académico de los estudiantes en comparación con las clases tradicionales. Se observó también que las metodologías activas tuvieron un efecto positivo en la competencia de trabajo en grupo. En cuanto a la competencia de planificación y gestión del tiempo, no se observó ningún efecto sobre ella.

3.1.9.2 Análisis

Se han analizado tres propuestas acerca de técnicas de aprendizaje activo. La Tabla 3.6 recoge una síntesis de los distintos aspectos analizados.

Los primeros trabajos sobre la enseñanza de sistemas operativos aparecen en la década de los ochenta. Sin embargo, las propuestas relacionadas con las técnicas de aprendizaje activo no aparecen hasta 2005.

Los trabajos exponen que los estudiantes tienen dificultades para aprender en las clases que consisten en una charla unidireccional por parte del profesor. Por este motivo, proponen distintas actividades durante las clases presenciales para que los estudiantes adopten un papel más activo en su aprendizaje.

Ninguno de los tres trabajos hace explícitos los objetivos del curso de sistemas operativos. En el primero de ellos no se realiza ningún estudio empírico ni se propone una forma para evaluar el aprendizaje de los estudiantes. El segundo trabajo (Jorge E. Pérez et al., 2008) utiliza una encuesta de satisfacción para evaluar su propuesta. El tercero (Jorge Enrique Pérez et al., 2010) lleva a cabo un experimento controlado en el que se miden tanto las competencias genéricas como el rendimiento académico de los estudiantes.

Por último, todas las propuestas han sido diseñadas exclusivamente para cursos presenciales. No hay ninguna mención al entorno *online*.

Tabla 3.6. Análisis de los trabajos acerca de técnicas de aprendizaje activo

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Lincke, 2005)	En la enseñanza y en el aprendizaje	No	No	—	Presencial
(Jorge E. Pérez et al., 2008)	En el aprendizaje	No	Encuesta de satisfacción	Metodología cuantitativa	Presencial
(Jorge Enrique Pérez et al., 2010)	En el aprendizaje	No	Prueba de evaluación	Metodología cuantitativa	Presencial

3.1.10 Uso de juegos para motivar al aprendizaje de la asignatura

En este apartado se describen y analizan los trabajos que hacen uso de juegos para mejorar el aprendizaje en la asignatura.

3.1.10.1 Descripción

El primer trabajo aparece en el año 2003 y presenta dos juegos para reforzar los objetivos de aprendizaje de la asignatura sistemas operativos (Hill, Ray, Blair, & Carver Jr., 2003). El primero de ellos es una adaptación del conocido juego de los barcos y está destinado a conocer las diferencias entre procesos e hilos. El

segundo es un ocurrente juego para comprender la transición entre los distintos estados de un proceso. El juego requiere de equipos de 4 a 6 personas en los que una de ellas tiene el papel de sistema operativo, otro de temporizador y el resto son programas que deben estar pendientes de un determinado número de procesos. Cada jugador debe realizar las acciones asociadas con su rol.

En el año 2013 se publican dos trabajos más sobre aprendizaje de Sistemas Operativos basado en juegos. Uno de ellos tiene por objeto el estudio de los efectos de un juego sobre el aprendizaje cooperativo (Jong et al., 2013). El trabajo estudia las diferencias en motivación y rendimiento entre el aprendizaje cooperativo basado en juegos y el aprendizaje cooperativo en clases presenciales. Los resultados muestran que el enfoque del aprendizaje cooperativo basado en juegos supera al enfoque tradicional.

Otra solución basada en juegos y también publicada en 2013 es el uso de juegos basados en redes sociales (She, Lin, Jong, & Hsia, 2013). Los resultados del estudio muestran que el juego aumenta la motivación de los estudiantes.

3.1.10.2 Análisis

Se han analizado tres propuestas acerca de la utilización de juegos en la asignatura Sistemas Operativos. La Tabla 3.7 muestra una síntesis de los resultados.

Tabla 3.7. Análisis de los trabajos acerca de juegos

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Hill et al., 2003)	No	Sí	Prueba de evaluación	Metodología cuantitativa	Presencial
(Jong et al., 2013)	En el aprendizaje	Sí	Prueba de evaluación	Metodología cuantitativa	Presencial/ <i>online</i>
(She et al., 2013)	En el aprendizaje	Sí	No	Metodología cuantitativa	Presencial/ <i>online</i>

Todos los trabajos analizados acerca del uso de juegos en la asignatura Sistemas Operativos han sido publicados en los diez últimos años.

En cuanto a las dificultades sobre el aprendizaje de la asignatura Sistemas Operativos, dos de los trabajos exponen que los estudiantes pierden la motivación para dedicar tiempo a la asignatura debido a su dificultad. Como posible solución a este problema, los trabajos proponen distintos tipos de juegos.

Por otra parte, los tres trabajos analizados coinciden en que un objetivo principal de la asignatura Sistemas Operativos es la comprensión de conceptos.

Con referencia a los estudios empíricos, los tres trabajos presentan los resultados de la puesta en marcha de la propuesta realizada. El primer trabajo (Hill et al., 2003) evalúa el aprendizaje de los estudiantes mediante un test que se realiza después de las experiencias propuestas. El segundo trabajo (Jong et al., 2013) realiza un experimento controlado y mide el aprendizaje de los estudiantes con la ayuda de un pre-test y un post-test. El tercer trabajo (She et al., 2013) usa también un experimento controlado, pero no mide el aprendizaje de los estudiantes, sino su motivación antes y después del juego.

Por último, y con relación a la modalidad de los cursos, la primera propuesta se realiza para un entorno presencial y las dos últimas están pensadas para realizarse a través de Internet. Por tanto, las dos últimas son adecuadas tanto para cursos presenciales como para cursos online.

3.1.11 Estudiar la comprensión de la asignatura Sistemas Operativos

En este apartado se describen y analizan los trabajos que tienen como principal objetivo el estudio de la comprensión de la asignatura Sistemas Operativos.

3.1.11.1 Descripción

Sólo existen dos trabajos en esta categoría. El propósito del primer trabajo es el estudio de los modelos mentales que tienen los estudiantes acerca de los semáforos. Este estudio constituye un hito en la investigación en la enseñanza y el aprendizaje de Sistemas Operativos porque es el primero que utiliza la metodología cualitativa (Kolikant, Ben-Ari, & Pollack, 2000). De acuerdo con los autores, la metodología cualitativa puede aportar información útil para la docencia que la metodología cuantitativa no puede proporcionar. La metodología cuantitativa, generalmente basada en un experimento controlado proporciona información sobre qué enfoque es mejor, pero no sobre cómo ni por qué ese enfoque mejora el aprendizaje o la docencia.

El segundo trabajo (Webb & Taylor, 2014) tiene como objetivo explorar las dificultades en el aprendizaje de la asignatura Sistemas Operativos. Para ello, usan como herramienta una prueba de evaluación que realizan los estudiantes antes y después de cursar la asignatura. Los autores llaman a esta prueba *Concept Inventory*, dado que la primera prueba diseñada con éste mismo propósito en el campo de la física se llamó *Force Concept Inventory* (Hestenes, Wells, & Swackhamer, 1992).

3.1.11.2 Análisis

Los dos trabajos analizados tienen su motivación en las dificultades que presenta la asignatura Sistemas Operativos para su aprendizaje. El objetivo de ambos es común y consiste en explorar las dificultades que tienen los estudiantes para comprender la asignatura.

En cuanto al establecimiento y evaluación de los objetivos de aprendizaje, ambos trabajos los establecen y los evalúan. El primero usa una metodología cualitativa para analizar una sesión de trabajo de los estudiantes. El segundo tiene un enfoque cuantitativo y está basado en los resultados de la prueba de evaluación. La Tabla 3.8 recoge una síntesis del análisis de estos trabajos.

Tabla 3.8. Análisis de los trabajos acerca de la comprensión de conceptos

Autor (es) y año	Menciona dificultades	Establece objetivos de aprendizaje	Evalúa el aprendizaje	Metodología de investigación	Modalidad curso
(Kolikant et al., 2000)	En el aprendizaje	Sí	Prueba de evaluación	Metodología cualitativa	Presencial
(Webb & Taylor, 2014)	En el aprendizaje	Sí	Prueba de evaluación	Metodología cuantitativa	Presencial/ <i>online</i>

3.1.12 Resumen de la revisión de literatura

En este apartado se presentan las conclusiones de la revisión de literatura realizada. Además, con objeto de comprobar los datos mencionados en dichas conclusiones, se proporciona también la Tabla 3.9 que realiza una síntesis del análisis llevado a cabo acerca de cada uno de los trabajos.

La revisión de literatura se resume en los siguientes puntos:

- **Confirmación de la importancia del problema abordado.** Un 66% de los trabajos analizados mencionan que existen dificultades en la enseñanza y el aprendizaje de la asignatura Sistemas Operativos. Asimismo, los trabajos sobre enseñanza y aprendizaje de Sistemas Operativos se han publicado de manera continua desde 1983 hasta la actualidad. Por tanto, se puede afirmar que el problema abordado es importante y se mantiene vigente en la fecha de publicación de la presente tesis.
- **Curriculum de la asignatura** Sistemas Operativos. Las asignaturas con el nombre Sistemas Operativos referidas en los trabajos analizados son homogéneas entre sí y están de acuerdo con los objetivos y contenidos del curriculum recomendado por ACM/IEEE (ACM/IEEE-CS Joint Interim Review Task Force, 2008). Los temas principales de la asignatura Sistemas Operativos son: gestión, planificación y sincronización de procesos, gestión de memoria, gestión de E/S y gestión de ficheros.
- **Objetivo principal de los trabajos analizados.** El objetivo principal de los trabajos analizados se puede agrupar en las siguientes categorías:
 - a. Presentar un determinado tipo de actividades prácticas: proyectos de desarrollo, simuladores o aplicaciones web para el aprendizaje autónomo de la asignatura Sistemas Operativos.
 - b. Usar técnicas de aprendizaje activo.
 - c. Usar juegos para motivar a los estudiantes.
 - d. Estudiar la comprensión de los conceptos de la asignatura Sistemas Operativos.

- **Exploración de las dificultades en aprendizaje de la asignatura Sistemas Operativos.**

En el área de la educación Sistemas Operativos, sólo existen dos trabajos cuyo objetivo ha sido la exploración de las dificultades para el aprendizaje de esta asignatura. Como ya se han mencionado, el estudio de las *misconceptions* es un área importante de la investigación en educación. Este tipo de trabajos son frecuentes en el área de programación, que es el área que más se ha desarrollado en la investigación en educación en informática (*Computer Science Education Research*).

- **Evaluación del aprendizaje de los estudiantes.**

Con referencia a la evaluación del aprendizaje de los estudiantes, los trabajos analizados presentan dos carencias. La primera es acerca de los objetivos de aprendizaje y la segunda sobre la forma de evaluar el aprendizaje.

- Por una parte, tan sólo un 44% de los trabajos estudiados, establecen de forma explícita los objetivos de aprendizaje que desean alcanzar. Además, en general los objetivos se establecen de una forma muy poco precisa. Por ejemplo, la mayor parte de los trabajos indican que el objetivo es “comprender conceptos”. Pero los autores no definen el término comprender ni hacen referencia a una taxonomía de objetivos de la educación. Por tanto, no queda claro el significado de “comprender”. La falta de precisión de los objetivos de aprendizaje tiene como consecuencia que no se pueda evaluar si se han alcanzado estos objetivos, y por tanto no se pueda conocer si la propuesta ha tenido éxito. Además impide conocer si una determinada propuesta es aplicable a un caso particular con unos determinados objetivos de aprendizaje. También dificulta la comparación de dos propuestas, ya que no se puede comprobar si comparten los mismos objetivos.
- Por otra parte, sólo un 23% de los trabajos analizados establecen una forma de evaluar el aprendizaje de los estudiantes. Por tanto, la gran mayoría de los trabajos, un 77%, presentan un método

para la mejorar de la enseñanza o el aprendizaje de Sistemas Operativos, pero no comprueban su eficacia. Además, de los 15 trabajos que evalúan el aprendizaje, 7 evalúan el aprendizaje únicamente a través de una encuesta de satisfacción y sólo 8 trabajos evalúan el aprendizaje de una forma objetiva, a través de una prueba que permita comprobar que se han alcanzado los objetivos de aprendizaje establecidos.

- **Metodologías de investigación.** Un 79% de los trabajos analizados se han limitado a presentar propuestas para la enseñanza de la asignatura Sistemas Operativos, y no han realizado un estudio empírico. Por otra parte, de los 16 estudios empíricos publicados, 15 aplican una metodología cuantitativa y sólo uno de ellos aplica una metodología cualitativa.
- **Modalidad *online*.** Sólo un 17% de los trabajos analizados presentan soluciones aplicables para un curso *online* de Sistemas Operativos. En parte, la escasez de trabajos para esta modalidad se debe a que la aparición de los entornos *online* en el ámbito universitario se ha producido en los últimos diez años.
- **Evaluación formativa.** No hay ningún trabajo en el área de Sistemas Operativos que estudie la evaluación formativa, ya sea en entornos *online* o presenciales.
- **Madurez de la investigación en el área de educación en Sistemas Operativos.** A la vista de los datos presentados en la Tabla 3.9, la mayoría de los trabajos están enfocados únicamente en la presentación técnica de la herramienta que proponen para la enseñanza o el aprendizaje. Esto hecho ocurre especialmente con los trabajos dedicados a los sistemas operativos instruccionales. La falta de estudios empíricos y de métodos para la evaluación del aprendizaje de los estudiantes sitúa al área de educación en Sistemas Operativos en una fase anterior con respecto a otras materias del área *Computer Science Education Research* como la programación.

Tabla 3.9. Resumen de la revisión de literatura

Propósito principal del trabajo	Trabajos que mencionan dificultades	Trabajos que establecen objetivos de aprendizaje	Trabajos que evalúan el aprendizaje	Trabajos que han realizado un estudio empírico	Trabajos realizados para una modalidad <i>online</i>
Proyectos de desarrollo de un sistema operativo real	4/9	1/9	3/9	3/9	2/9
Proyectos de desarrollo de un sistema operativo instruccional	13/22	5/22	0/22	0/22	0/22
Otros proyectos de desarrollo	5/7	5/7	1/7	0/7	0/7
Uso de simuladores de sistemas operativos	13/16	12/16	5/16	4/16	2/16
Uso de aplicaciones web	2/4	1/4	2/4	2/4	4/4
Utilización de técnicas de aprendizaje activo	3/3	0/3	2/3	2/3	0/3
Uso de juegos para motivar al aprendizaje de la asignatura	2/3	3/3	2/3	3/3	2/3
Estudiar la comprensión de la asignatura Sistemas Operativos	2/2	2/2	2/2	2/2	1/2
TOTAL	44/66	29/66	15/66	14/66	11/66

3.2 Otros trabajos relacionados

Este apartado se organiza alrededor de los dos temáticas principales de la tesis: la comprensión de conceptos y la evaluación formativa *online*.

3.2.1 Trabajos acerca de la comprensión de conceptos

A continuación se citan los trabajos relacionados con la comprensión de conceptos y los motivos por los que están relacionados con la presente tesis.

3.2.1.1 Bibliografía de Seymour Papert acerca del aprendizaje con computadoras

El origen de esta tesis fueron dos trabajos de Seymour Papert acerca del aprendizaje con computadoras (Papert, 1980, 1993). En ellos, Papert relata la forma en que las computadoras ayudan a los niños al aprendizaje de conceptos de distinta naturaleza (matemática, ingeniería, etc.).

Los trabajos de Papert centran su atención en *cómo* la relación entre niños y computadoras afecta al aprendizaje. Este enfoque contrasta con la finalidad de los experimentos controlados, que proporcionan información sobre *qué* métodos o herramientas usar, pero no pueden explicar *cómo* esos métodos o herramientas mejoran el aprendizaje.

La idea de investigar *cómo* tiene lugar el aprendizaje motivó a la autora de esta tesis al estudio de la metodología de investigación cualitativa y a la posterior aplicación de esta metodología en el presente trabajo.

3.2.1.2 Estudios acerca de las *misconceptions* en ciencias

Otros trabajos que han inspirado esta tesis son los dedicados a estudio de las *misconceptions*. Este tipo de trabajos conforman un área importante de la investigación en educación de las ciencias. Las *misconceptions* son concepciones claramente incompatibles con las concepciones aceptadas y que se mantienen de forma persistente, incluso después de la instrucción (Abimbola, 1988).

Uno de los trabajos más significativos en el campo de las *misconceptions* estudiado por la autora en el momento de génesis de esta investigación es el *Force Concept Inventory* (Hestenes et al., 1992). Este trabajo diseña un test que

permite comprobar las *misconceptions* que tienen los estudiantes acerca del concepto de fuerza, un concepto central en la mecánica Newtoniana.

Hoy en día abundan las referencias sobre *misconceptions* en diversas materias (Bransford, Brown, & Cocking, 1999). En el campo de Computer Science Education Research existen numerosos trabajos, que se han dedicado principalmente a la comprensión de conceptos relacionados con la programación tales como: asignación y secuencia (Simon, 2011), sentencias lógicas (VanDeGrift et al., 2010), programación concurrente (Lönnberg & Berglund, 2007), paso de parámetros (Madison, 1995).

Entre los estudios citados acerca de programación, la tesis doctoral de Madison (Madison, 1995) ha constituido una referencia importante en el inicio de este trabajo por compartir objetivos y haber usado la misma metodología de investigación. Madison estudió la comprensión del concepto de *paso por parámetros* en un curso introductorio de programación. Los datos del estudio proceden de ocho estudiantes y están compuestos por entrevistas, observaciones realizadas en el aula y trabajos llevados a cabo por los estudiantes durante el curso.

Otros estudios en estrecha relación con esta tesis son los dedicados a la comprensión de conceptos acerca de la *Entrada/Salida* (Larraza-Mendiluze & Garay-Vitoria, 2013a, 2013b). Este es un concepto estudiado en el campo de Arquitectura de Computadores, pero también está estrechamente relacionado con la asignatura Sistemas Operativos.

Uno de los trabajos sobre la Entrada/Salida (Larraza-Mendiluze & Garay-Vitoria, 2013a) analiza los objetivos de aprendizaje de 152 preguntas de exámenes procedentes de 12 universidades. Los resultados del análisis muestran que la mayoría de las preguntas se corresponden con los niveles superiores de la taxonomía de Bloom (aplicación, análisis, síntesis y evaluación) y abarcan un amplio rango de conceptos.

El segundo de los trabajos citados sobre Entrada/Salida (Larraza-Mendiluze & Garay-Vitoria, 2013b) identifica *misconceptions* utilizando un original método para elicitar la comprensión de los estudiantes: los mapas conceptuales. Dicha

investigación analiza los mapas conceptuales contruidos por 39 estudiantes después de cursar la asignatura Estructura de Computadores. El análisis se lleva a cabo mezclando todos los mapas conceptuales y analizando las relaciones existentes entre los distintos conceptos creados en los mapas. Los resultados muestran que en las mentes de los estudiantes hay tres conceptos que no están conectados entre sí: el *controlador de E/S*, la *sincronización* y el *DMA*.

Como se ha se ha mencionado en el apartado anterior, sólo hay dos trabajos dedicados al estudio de las *misconceptions* en el área de Sistemas Operativos. Uno de ellos es una investigación cualitativa que estudia la comprensión del concepto *semáforo* (Kolikant et al., 2000). El otro trabajo (Webb & Taylor, 2014) está dedicado al diseño de un test que permite detectar *misconceptions*, similar al *Force Concept Inventory*, mencionado en este mismo apartado.

3.2.2 Trabajos acerca de la evaluación formativa *online*

Los trabajos en el área de evaluación formativa en entornos *online* son muy escasos a pesar de la relevancia del tema. Se exponen a continuación, por orden cronológico, ocho trabajos que están relacionados con la presente tesis, y que han sido publicados en los últimos diez años (2004 – 2014).

Los dos primeros estudios están realizados con el área de ingeniería. En el año 2004 (Dopper & Sjoer, 2004), se realiza un estudio sobre los beneficios que se pueden obtener con un sistema de evaluación formativa *online* llamado *Etude*, basado en un banco de preguntas. En este momento se considera que una de las principales aportaciones de la tecnología a la evaluación es la gestión de un mayor número de estudiantes y la posibilidad de otorgar calificaciones de forma automática. El sistema *Etude* permite un análisis y presentación de los resultados rápido y sistemático, un *feedback* inmediato a través del computador, la generación automática del test en función del nivel del estudiante y la disponibilidad de los test en cualquier momento y lugar. La disciplina contemplada en el estudio es la escritura de informes de forma colaborativa dentro de un programa de estudios en ingeniería.

En el año 2006 (Chung et al., 2006), se publica otro estudio sobre evaluación formativa en el área de ingeniería, en concreto sobre resolución de problemas de circuitos. En esta ocasión se utiliza un software llamado *Discourse* que

permite al profesor realizar sesiones síncronas con los estudiantes para contestar sus preguntas y monitorizar sus resultados. La percepción de los estudiantes acerca del uso de la herramienta es positiva, aunque se concluye el trabajo afirmando que la tecnología por sí sola no facilita el aprendizaje, son las acciones llevadas a cabo por el profesor lo más valorado por los estudiantes.

Más adelante, en la primavera de 2007 hace su aparición un importante estudio centrado en la evaluación de las discusiones asíncronas *online* (Vonderwell, 2007). La herramienta informática utilizada en este caso es *WebCT*, pero a diferencia de los dos estudios anteriores no se pone el énfasis en la tecnología, sino en comunicar las dimensiones que influyen en el aprendizaje y la evaluación en las discusiones *online*. Para ello se utiliza un riguroso estudio de caso cualitativo. La disciplina objeto del estudio es la educación (*Educational Research and Planning for Technology*).

En el verano de 2007 (Smith, 2007), sale a la luz un estudio que avala la correlación entre las evaluaciones formativas y los exámenes finales. El estudio se realiza sobre materias de *geociencia* (*Earth History and Environmental Geology*) y la herramienta utilizada es la misma que en el estudio anterior: *WebCT*. Se utiliza la taxonomía de Bloom para categorizar las preguntas de la evaluación formativa y la sumativa.

En el año 2008, se lleva a cabo un estudio sobre una herramienta desarrollada para potenciar los efectos de la evaluación formativa (PePCAA Project) en el área de psicología pedagógica (Crisp & Ward, 2008). Este proyecto incluye dos innovaciones importantes: preguntas basadas en el método del caso y medición de la confianza de las respuestas proporcionadas por los estudiantes. Los resultados confirman los beneficios de la evaluación formativa: motivación, aprendizaje significativo y responsabilidad sobre el propio aprendizaje.

A continuación, existen dos estudios importantes acerca de los *e-portfolios* usados como herramienta de evaluación formativa en cursos de tecnología para profesores. El primero (2008) afirma que los *e-portfolios* permiten a los estudiantes realizar conexiones con el aprendizaje pasado y transformarlo en conocimiento activo y auténtico (Terrell & Lin, 2008). El segundo (2009) evalúa los efectos de la colaboración entre estudiantes en la construcción de un *e-*

porfolio. Sus resultados revelan que la colaboración produce mejores e-
porfolios, crea experiencias de aprendizaje positivas y mejora el manejo de la
tecnología (Wang, 2009).

El siguiente trabajo, publicado en 2010, tuvo como objetivo identificar teorías y
prácticas vigentes en relación con la evaluación formativa en las que las
tecnologías jugaban un papel clave (Underwood et al., 2010). Sus resultados
sugieren que existen atributos tecnológicos clave que pueden suponer una
diferencia en el potencial de los estudiantes para el aprendizaje. Sin embargo,
afirman que la tecnología no garantiza el aprendizaje, éste depende del
conjunto de respuestas humanas, factores motivaciones y contextos socio-
interactivos que existen alrededor de las herramientas electrónicas.

Todos estos trabajos proporcionan importantes contribuciones al conocimiento
de la evaluación formativa. Sin embargo, la mayoría no ofrecen unas guías claras
para el diseño y puesta en marcha de la misma. Sólo los trabajos (Vonderwell,
2007) y (Crisp & Ward, 2008) proporcionan indicaciones que pueden ayudar a
los profesores a crear evaluaciones formativas válidas. Por otra parte, aunque
todos los trabajos descritos ofrecen evidencia empírica, sólo los trabajos
(Vonderwell, 2007) y (Smith, 2007) usan una metodología de investigación
rigurosa. Por último, sólo uno de los trabajos está relacionado directamente con
una materia de ingeniería: análisis y resolución de circuitos (Chung et al., 2006).

3.4 Contexto y originalidad del presente trabajo

Este trabajo se enmarca dentro del área de investigación en *Computer Science Education Research*, que tiene como propósito estudiar los procesos de enseñanza y aprendizaje en las disciplinas relacionadas con la computación.

La temática principal de la presente tesis es la enseñanza y el aprendizaje de sistemas operativos. Las características que diferencian esta investigación de los trabajos existentes en esta materia son las siguientes:

- Realiza un estudio exploratorio de las dificultades de aprendizaje de la asignatura Sistemas Operativos, que no había sido realizado hasta el momento. Sólo existe un trabajo que se ha realizado de forma paralela al presente estudio y cuyos resultados preliminares han sido publicados en 2014 (Webb & Taylor, 2014). El presente trabajo se diferencia del trabajo existente en que utiliza un marco teórico en evaluación basado en la evaluación formativa y en la taxonomía revisada de Bloom. Por el contrario, la evaluación realizada por el trabajo publicado no está sustentada de forma teórica. Además, la presente tesis *descubre* las dificultades de aprendizaje a través de un estudio cualitativo. En contraste, el trabajo publicado construye un test que usa distractores basados en los problemas observados por los profesores en la docencia de la asignatura. Por tanto, no hay descubrimiento, sino confirmación de las dificultades observadas previamente por los profesores.
- Utiliza como herramienta de aprendizaje la evaluación formativa, que no ha sido utilizada por ningún trabajo de educación en Sistemas Operativos hasta el momento. Además, en la presente tesis se describe cómo se realizó el diseño de la evaluación formativa con el fin de que pueda servir como referencia. Estas indicaciones sobre el diseño y puesta en marcha de la evaluación formativa no son ofrecidas por la mayor parte de los trabajos sobre evaluación formativa *online*, y por tanto aportan originalidad a esta investigación.
- Hace explícitos los objetivos de aprendizaje del curso impartido, cuestión poco frecuente en los trabajos revisados. Además, analiza los objetivos de aprendizaje de la asignatura de acuerdo con la taxonomía de Bloom, lo que permite precisar lo que se pretende conseguir con el curso.

- Establece una forma de evaluar el aprendizaje de los estudiantes, cuestión también poco frecuente en los trabajos revisados. La evaluación efectuada está alineada con los objetivos de aprendizaje establecidos y tiene como finalidad principal promover el aprendizaje significativo.
- Realiza un estudio empírico y utiliza una metodología de investigación rigurosa, lo que lo diferencia de los estudios existentes. Además, utiliza una metodología cualitativa, muy poco aplicada hasta el momento en el área de Sistemas Operativos y en general en el área de *Computer Science Education Research*.
- Estudia un curso *online* de Sistemas Operativos, modalidad que ha sido escasamente estudiada hasta el momento.

4. Estudio empírico

En este capítulo se describe el estudio empírico llevado a cabo. En primer lugar se precisa el contexto de la investigación: un aula virtual de la asignatura Sistemas Operativos en una universidad *online*. En segundo lugar se justifica la metodología de investigación utilizada. En tercer lugar se discute el diseño de las pruebas de evaluación realizadas por los estudiantes durante el curso. Por último, se detallan los procesos de recogida y análisis de datos y las estrategias adoptadas respecto a la validez, confiabilidad y ética de la investigación.

4.1 Definición del problema

El estudio se realizó en la asignatura Sistemas Operativos perteneciente al segundo curso del Grado en Ingeniería Informática en una universidad *online* (Universidad a Distancia de Madrid, UDIMA). La duración del curso fue de catorce semanas comprendidas entre los meses de octubre de 2012 y febrero de 2013.

A continuación se describe el contexto del curso y la asignatura objeto de estudio: Sistemas Operativos.

4.1.1 El contexto *online*

En la literatura no existe acuerdo acerca del significado de los términos *e-learning* y *online learning* (Nachmias, Moore, Dickson-Deane, & Galyen, 2011) (Sangrà, Vlachopoulos, Cabrera, & Bravo, 2011). Se precisan a continuación las definiciones que se han tomado como referencia en esta investigación.

E-learning es una modalidad de enseñanza y aprendizaje, que puede representar todo o una parte del modelo educativo en el que se aplica, que explota los medios y dispositivos electrónicos para facilitar el acceso, la evolución y la mejora de la calidad de la educación y la formación (Sangrà et al., 2011).

Se llama aprendizaje online a un tipo de educación con las siguientes características: está dirigida por un profesor, tiene lugar a través de internet, el profesor y el estudiante están separados geográficamente y usa un software que proporciona un entorno estructurado de aprendizaje. El aprendizaje online puede ser síncrono (interacción con los estudiantes en tiempo real, como una videoconferencia), o asíncrono (comunicación separada en el tiempo, como el email o los foros de discusión online) (Watson, Murin, Vashaw, Gemin, & Rapp, 2012).

En este estudio se considera *e-learning* a cualquier aprendizaje que tiene lugar a través de medios electrónicos y aprendizaje *online* a una modalidad específica de *e-learning* en la que la comunicación se realiza a través de internet y el estudiante y el profesor están separados geográficamente.

La mayoría de los estudios publicados acerca de evaluación formativa *online* se refieren en realidad a experiencias de evaluación en entornos presenciales en las que se ha usado computadoras (Dopper & Sjoer, 2004) (Chung et al., 2006) (Crisp & Ward, 2008). Este estudio describe una situación distinta: el desarrollo de una evaluación formativa en un curso que se desarrolla por completo de forma *online*, en el que profesores y estudiantes están separados geográficamente. En el aprendizaje *online*, la evaluación formativa puede tener más importancia que en un curso presencial porque puede constituir una guía para el estudiante, un camino a seguir en el que el estudiante puede valorar su aprendizaje en todo momento.

Además, la condición *online* imprime unas características especiales al estudio. Por una parte, el investigador tiene más problemas para percibir el proceso de aprendizaje, puesto que no es posible la observación directa del estudiante. Por otra parte, el estudio es más aséptico respecto a la influencia del profesor en los resultados.

Las condiciones en las que se imparte el curso *online* son las siguientes:

- Los estudiantes disponen de un aula virtual realizada con el sistema de gestión del aprendizaje *moodle* que contiene todos los materiales y actividades del curso.

- El aula virtual proporciona mecanismos para interactuar con el profesor y con el resto de estudiantes de la asignatura (foros).
- El curso se imparte exclusivamente de forma *online*, no existen clases presenciales.
- Las actividades con las que se pretende que los estudiantes lleven a cabo su aprendizaje son la lectura del manual y las actividades de evaluación continua.

4.1.2 La asignatura Sistemas Operativos

Los Sistemas Operativos son un área de conocimiento incluida en la mayoría de los estudios superiores relacionados con la informática en todo el mundo. Su importancia está reconocida en el curriculum que publican desde hace más de 40 años las dos principales sociedades profesionales en informática: ACM y IEEE (ACM/IEEE-CS Joint Interim Review Task Force, 2008).

El manual utilizado en el curso se ha creado expresamente para esta asignatura y cuenta con las siguientes características:

- El manual contiene la información necesaria para que los estudiantes realicen las actividades y pruebas de evaluación del curso. Sin embargo, dentro de esta información no se encuentran las respuestas a las actividades planteadas de forma directa, para llegar a ellas es necesario llevar a cabo procesos cognitivos superiores tales como interpretar, inferir, comparar, explicar, etc. En definitiva, para superar las pruebas de evaluación se debe realizar un aprendizaje significativo, que vaya más allá de los procesos cognitivos *reconocer* y *recordar*.
- Puesto que el objetivo del curso es la comprensión de los conceptos esenciales de la asignatura y no el recuerdo de los mismos, se permite la utilización del manual en todas las pruebas de la evaluación formativa. De este modo, se puede constatar con mayor probabilidad que si un estudiante no contesta de forma adecuada a una pregunta, no se deba a un fallo en los procesos cognitivos de reconocer o recordar.

- El manual está confeccionado de acuerdo con el currículum recomendado por ACM/IEEE-CS Joint Task Force (ACM/IEEE-CS Joint Interim Review Task Force, 2008) y contiene los siguientes capítulos.
 1. Introducción a los sistemas operativos
 2. Gestión de procesos
 3. Planificación de procesos
 4. Comunicación y sincronización de procesos
 5. Gestión de memoria
 6. Gestión de entrada/salida
 7. Gestión de ficheros
 8. Protección y seguridad
 9. Caso de estudio 1: UNIX/Linux
 10. Caso de estudio 2: Windows 7

4.2 Metodología

El propósito del estudio es estudiar el proceso aprendizaje de los estudiantes con el fin de descubrir dificultades e indagar acerca de los efectos de la evaluación formativa. Dado el carácter exploratorio de la investigación, se ha usado como metodología el estudio de caso cualitativo (Stake, 1978).

4.3 Diseño de las pruebas de evaluación formativa

A continuación se describe cómo se realizó el diseño de la evaluación formativa llevada a cabo en el estudio con la intención de que pueda servir como referencia a otros profesores del área de informática. Esto supone una contribución del presente trabajo, ya que la mayoría de los estudios empíricos sobre evaluación formativa no proporcionan una descripción detallada del diseño e implementación de la evaluación llevada a cabo.

En este estudio, además de hacer explícitos los objetivos de aprendizaje, se realizó un análisis de los mismos con el objetivo de conocer la naturaleza de la asignatura Sistemas Operativos. Para ello se ha utilizado la taxonomía revisada de Bloom.

El proceso de diseño de la evaluación formativa constó de dos fases:

1. Establecer los objetivos del curso.
2. Diseñar las pruebas de evaluación.

En los siguientes epígrafes se detallan los pasos seguidos en cada una de las fases:

4.3.1 Objetivos del curso

El establecimiento de los objetivos de aprendizaje de una determinada materia es conveniente, puesto que el proceso de evaluación debe comprobar si se han alcanzado estos objetivos. A pesar de este hecho, son pocos los trabajos sobre educación en Sistemas Operativos que han hecho explícitos los objetivos del curso estudiado.

Para establecer los objetivos del curso *online* objeto del estudio se tomaron como punto de partida los objetivos de aprendizaje recomendados por el ACM/IEEE Computer Science Curriculum sobre la materia Sistemas Operativos (ACM/IEEE-CS Joint Interim Review Task Force, 2008). En particular, se seleccionaron los objetivos de los temas que constituyen el núcleo de la asignatura: visión general de los sistemas operativos, principios de los sistemas operativos, concurrencia, planificación, gestión de memoria y seguridad y protección.

En primer lugar se clasificaron los objetivos seleccionados de acuerdo a la taxonomía revisada de Bloom (L. W. Anderson et al., 2000). Dicha clasificación presenta dos dificultades que se describen a continuación.

La primera dificultad es que la clasificación de los objetivos de aprendizaje no es una tarea automática como pueda parecer. Requiere un conocimiento profundo de los procesos cognitivos recogidos en la taxonomía revisada de Bloom. Por ejemplo, en muchas ocasiones el verbo utilizado en la redacción del objetivo no

coincide con el proceso cognitivo. En el objetivo “Explain the benefits of building abstract layers in hierarchical fashion”, el proceso cognitivo es interpretar y no explicar como cabría esperar, ya que explicar está definido en esta taxonomía como “Contruir un modelo causa-efecto de un sistema”.

La segunda dificultad que se presenta en la clasificación de objetivos educativos es la necesidad de saber o suponer la naturaleza de las experiencias educativas que los estudiantes han tenido durante el curso. Así, por ejemplo, un problema presentado en una prueba requerirá procesos cognitivos complejos si la situación con la que se enfrenta el estudiante es nueva, mientras que no exigirá más que una simple evocación si ya ha pasado por una experiencia de aprendizaje en la cual un problema semejante fue analizado y discutido.

Esta dificultad se ha resuelto en el curso estudiado suponiendo que las experiencias educativas que los estudiantes tienen durante el curso se limitan a la realización de la evaluación formativa propuesta y al estudio del manual de la asignatura. Por ejemplo, el objetivo “Describir la necesidad de concurrencia en un sistema operativo” está relacionado con el proceso cognitivo inferir, ya que la información que hay que describir no se encuentra de forma explícita en el manual, debe ser inferida por los estudiantes.

El resultado del proceso de clasificación de los objetivos de la asignatura de acuerdo con la taxonomía de Bloom se puede encontrar en el anexo A. La Tabla 4.1 muestra los resultados de la clasificación de objetivos en la dimensión de procesos cognitivos. La Tabla 4.2 muestra los resultados en la dimensión del tipo de conocimiento.

Los resultados muestran que el 88% de los objetivos están relacionados con procesos cognitivos de la categoría *comprensión* y el 90% de los objetivos se refieren al conocimiento conceptual. Por tanto, se podría afirmar el objetivo principal del curso de sistemas operativos propuesto por ACM/IEEE es comprender conceptos. De acuerdo con la taxonomía revisada de Bloom, la categoría *comprensión* está formada por los procesos cognitivos *interpretar, ejemplificar, clasificar, resumir, inferir, comparar* and *explicar*. Dichos procesos suponen un aprendizaje significativo.

Tabla 4.1. Número y porcentaje de objetivos por categoría de procesos cognitivos

	Número de objetivos	Porcentaje
Recordar	2	5%
Comprender	37	88%
Aplicar	3	7%
Analyze	0	0%
Evaluate	0	0%
Create	0	0%
TOTAL	42	100%

Tabla 4.2. Número y porcentaje de objetivos por tipo de conocimiento

	Número de objetivos	Porcentaje
Factual	2	5%
Conceptual	38	90%
Procedural	2	5%
Metacognitivo	0	0%
TOTAL	42	100%

En la Tabla 4.2 se puede comprobar que el documento de ACM/IEEE no contiene ningún objetivo de aprendizaje sobre el conocimiento metacognitivo. Sin embargo, el conocimiento de los estudiantes sobre su propia cognición juega un papel muy importante en el aprendizaje (Bransford et al., 1999). Por tanto, además de los objetivos de aprendizaje propuestos por ACM/IEEE se incluyó como objetivo del curso estudiado que los estudiantes reflexionen sobre su aprendizaje.

4.3.2. Pruebas de evaluación

El diseño de las pruebas ha constado de las dos fases que se describen a continuación: (1) seleccionar las pruebas de los principales textos de sistemas operativos y (2) modificar dichas pruebas.

4.3.2.1 Selección de las pruebas de los principales textos de sistemas operativos

El objetivo de esta fase fue seleccionar pruebas que promuevan el aprendizaje significativo. Las pruebas de evaluación del curso se han construido a partir de preguntas y problemas incluidos en los principales textos de la materia Sistemas Operativos (Candela, García, Quesada, Santana, & Santos, 2007; Carretero, García, De Miguel, & Pérez, 2007; Casillas & Iglesias, 2004; F. Pérez, Carretero, & García, 2003; Silberschatz, Galvin, & Gagne, 2011; Stallings, 2011). El proceso de selección de las pruebas constó de las siguientes fases:

- Hacer explícitos los objetivos de aprendizaje de las pruebas de evaluación de los libros de texto.
- Clasificar dichos objetivos de aprendizaje de acuerdo con la taxonomía revisada de Bloom.
- Decidir si la prueba pasa a formar parte de la evaluación formativa. El criterio de decisión fue que los objetivos de aprendizaje de la pregunta se correspondan con los objetivos de aprendizaje establecidos para el curso y supongan además un aprendizaje significativo.

Para poder hacer explícitos de aprendizaje de las pruebas hay que tener en cuenta que hay varios tipos de objetivos de aprendizaje. Los objetivos recomendados por ACM/IEEE se denominan objetivos educativos, el tiempo necesario para alcanzarlos son semanas o meses, y su función principal es diseñar el curriculum de una determinada materia. Por otra parte, los objetivos de aprendizaje de cada una de las pruebas de evaluación se denominan objetivos instruccionales, el tiempo necesario para alcanzarlos son horas o días, y sirven para preparar clases y pruebas de evaluación (Airasian, 1994; Sosniak, 1994).

A continuación se exponen algunos ejemplos de los problemas seleccionados, y se detallan sus objetivos instruccionales y el objetivo educativo de ACM/IEEE con el que están relacionados.

Pregunta 8 (Prueba de evaluación: Cuestionario I. Anexo B)

¿Qué código crees que necesita ejecutarse con las interrupciones inhibidas?

- A. Ninguno, ya que podrían perderse interrupciones.
- B. Todo el código del sistema operativo.
- C. Ciertas partes críticas del código del sistema operativo como por ejemplo el cambio de contexto.

Justifica tu respuesta

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N1, pregunta 20.

Tabla 4.3. Objetivos instruccionales de la pregunta 8 de la prueba de evaluación *Cuestionario I*

	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
Objetivos intruccionales			
Inferir las situaciones en las que deben se inhibidas las interrupciones	Conceptual	Comprender	Inferir
Objetivo educativo de ACM/IEEE relacionado			
6. Describir razones para usar interrupciones, planificación de procesos y cambio de contexto para soportar la concurrencia en un sistema operativo. (Apartado Concurrencia)	Conceptual	Comprender	Interpretar

Pregunta 6 (Prueba de evaluación: Cuestionario II. Anexo C)

El cambio de proceso (también llamado cambio de contexto):

- A. Lo realiza el planificador.
- B. Modifica la entrada en la tabla de procesos del proceso desalojado.
- C. Siempre se origina por una *interrupción* de reloj.
- D. Se produce siempre que el proceso abandona la cola de procesos en espera y pasa a la cola de procesos preparados.

Justifica tu respuesta

Fuente: adaptado de (Casillas & Iglesias, 2004), página 9, cuestión 1.2.4.

Tabla 4.4. Objetivos instruccionales de la pregunta 6 de la prueba de evaluación *Cuestionario II*

	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
Objetivos intruccionales			
Inferir las circunstancias y eventos que pueden causar un cambio de contexto.	Conceptual	Comprender	Inferir
Objetivo educativo de ACM/IEEE relacionado			
3. Sintetizar el rango de mecanismos que pueden ser empleados en un sistema operativo para implementar la concurrencia. Describir los beneficios de cada uno de ellos (Apartado Concurrencia).	Conceptual	Comprender	Interpretar

Pregunta 1 (Prueba de evaluación: Ejercicios y problemas I. Anexo E)

Supongamos un sistema operativo con una planificación round-robin con un tiempo de 0,25 μ s entre dos rodajas de tiempo (independientemente de si ocurre o no un cambio de proceso). Si se ejecutan tres trabajos de 6 μ s, 3 μ s y 8 μ s.

Contesta las siguientes preguntas:

- ¿Cuál es el rendimiento si la rodaja de tiempo es de 1 μ s?
- ¿Cuál es el rendimiento si la rodaja de tiempo es de 0,5 μ s?
- ¿En cuál de los dos casos presentados anteriormente el rendimiento es mayor? ¿Cómo lo explicarías?

Fuente: adaptado de (Stallings, 2011), página 124, ejercicio 2.4.

Tabla 4.5. Objetivos instruccionales de la pregunta 1 de la prueba de evaluación *Ejercicios y problemas I*

	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
Objetivos instruccionales			
Interpretar el algoritmo de planificación round-robin.	Conceptual	Comprender	Interpretar
Comparar dos posibles implementaciones del algoritmo round-robin para comprender la influencia del tamaño de la rodaja de tiempo en el rendimiento.	Conceptual	Comprender	Comparar
Objetivo educativo de ACM/IEEE relacionado			
1. Comparar los algoritmos de planificación de procesos más comunes (Apartado Planificación).	Conceptual	Comprender	Comparar

Ejercicio 1 (Prueba de evaluación: Ejercicios y problemas II. Anexo E)

Una CPU ejecuta 50 instrucciones máquina por microsegundo (μs). Supongamos que un programa procesa un fichero de registros y que realiza las siguientes operaciones para cada registro:

- Lectura del registro (Duración: 11 μs)
- Proceso de la información del registro (Duración: ejecución de 150 instrucciones máquina)
- Escritura del registro (Duración: 11 μs)

¿Cuál será el porcentaje de utilización de la CPU durante la ejecución de este programa? Justifica los pasos seguidos para realizar el cálculo.

Fuente: adaptado de (Stallings, 2011), página 124, ejercicio 2.3.

Tabla 4.6. Objetivos instruccionales y objetivos educativos del ejercicio 1 de la prueba Ejercicios y problemas II

	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
Objetivos intruccionales			
Interpretar el concepto de utilización de la CPU.	Conceptual	Comprender	Interpretar
Inferir que durante el tiempo de E/S la CPU queda libre.	Conceptual	Comprender	Inferir
Objetivo educativo de ACM/IEEE relacionado			
1. Describir la necesidad de concurrencia en un sistema operativo (Apartado Concurrencia).	Conceptual	Comprender	Inferir

4.3.2.2 Modificación de las pruebas seleccionadas

A continuación se describen las modificaciones de las pruebas seleccionadas en relación con cada una de las finalidades deseadas:

- Obtener evidencias del aprendizaje de los estudiantes
- Potenciar el conocimiento metacognitivo

Obtener evidencias del aprendizaje de los estudiantes

La evaluación formativa requiere comprobar que los estudiantes han alcanzado los objetivos de aprendizaje establecidos (William & Thompson, 2007). Esta comprobación no es sencilla, ya que es posible que un estudiante conteste de forma correcta a una pregunta multi-opción a causa del azar, o que un estudiante resuelva un problema de forma correcta aplicando un procedimiento conocido, sin entender realmente por qué se resuelve de esa forma. Para aumentar la seguridad de que han tenido lugar los procesos cognitivos previstos, se han modificado las preguntas multi-opción y los problemas de la siguiente forma:

- En las preguntas multi-opción se ha añadido una pregunta adicional en la que se pide al estudiante que justifique su respuesta, es decir, que explique el motivo por el que ha seleccionado su respuesta.
- En los problemas se han añadido preguntas con la misma intención mencionada, para el estudiante haga explícito su proceso de pensamiento.

A continuación se muestran varias preguntas a modo de ejemplo donde se resalta en negrita la pregunta que se ha añadido a la prueba original:

Pregunta 8 (Prueba de evaluación: Cuestionario I. Anexo B)

¿Qué código crees que necesita ejecutarse con las interrupciones inhibidas?

- A. Ninguno, ya que podrían perderse interrupciones.
- B. Todo el código del sistema operativo.
- C. Ciertas partes críticas del código del sistema operativo como por ejemplo el cambio de contexto.

Justifica tu respuesta

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N1, pregunta 20.

Pregunta 6 (Prueba de evaluación: Cuestionario II. Anexo C)

El cambio de proceso (también llamado cambio de contexto):

- A. Lo realiza el planificador.
- B. Modifica la entrada en la tabla de procesos del proceso desalojado.
- C. Siempre se origina por una *interrupción* de reloj.
- D. Se produce siempre que el proceso abandona la cola de procesos en espera y pasa a la cola de procesos preparados.

Justifica tu respuesta

Fuente: adaptado de (Casillas & Iglesias, 2004), página 9, cuestión 1.2.4.

Ejercicio 1 (Prueba de evaluación: Ejercicios y problemas I. Anexo E)

Supongamos un sistema operativo con una planificación round-robin con un tiempo de 0,25 μs entre dos rodajas de tiempo (independientemente de si ocurre o no un cambio de proceso). Si se ejecutan tres trabajos de 6 μs , 3 μs y 8 μs .

Contesta las siguientes preguntas:

- ¿Cuál es el rendimiento si la rodaja de tiempo es de 1 μs ?
- ¿Cuál es el rendimiento si la rodaja de tiempo es de 0,5 μs ?
- **¿En cuál de los dos casos presentados anteriormente el rendimiento es mayor? ¿Cómo lo explicarías?**

Fuente: adaptado de (Stallings, 2011), página 124, ejercicio 2.4.

Ejercicio 1 (Prueba de evaluación: Ejercicios y problemas II. Anexo F)

Una CPU ejecuta 50 instrucciones máquina por microsegundo (μs). Supongamos que un programa procesa un fichero de registros y que realiza las siguientes operaciones para cada registro:

- Lectura del registro (Duración: 11 μs)
- Proceso de la información del registro (Duración: ejecución de 150 instrucciones máquina)
- Escritura del registro (Duración: 11 μs)

¿Cuál será el porcentaje de utilización de la CPU durante la ejecución de este programa? **Justifica los pasos seguidos para realizar el cálculo.**

Fuente: adaptado de (Stallings, 2011), página 124, ejercicio 2.3.

Potenciar el conocimiento metacognitivo

Uno de los objetivos del curso fue que los estudiantes reflexionaran sobre su aprendizaje. Para facilitar este objetivo se añadieron preguntas que exigen expresamente que el estudiante tenga este tipo de reflexión, dicho en términos de Flavell (Flavell, 1979) que propicien experiencias metacognitivas. El conocimiento metacognitivo es un elemento clave en el diseño de la evaluación formativa llevada a cabo en la asignatura.

A continuación se muestran algunas de las preguntas que se han utilizado para propiciar la reflexión sobre el proceso de aprendizaje particular de cada estudiante. Estas preguntas están inspiradas por las utilizadas en una de las primeras investigaciones en educación en informática (Madison, 1995).

Ejercicio 4 (Prueba de evaluación: Ejercicios y problemas I. Anexo E)

- a). ¿Qué opinas sobre lo que hemos realizado en la asignatura hasta el momento? ¿Ha sido fácil? ¿Difícil? ¿Interesante? ¿Aburrido? ¿Puedes explicar por qué?
- b). ¿Qué ejercicio de esta actividad te ha parecido más complicado? ¿Puedes explicar por qué?
- c) ¿Qué ejercicio de esta actividad te ha parecido más interesante? ¿Puedes explicar por qué?
- d) ¿Qué conceptos o aspectos de la asignatura te han parecido más complicados hasta el momento? ¿Sabrías explicar por qué?
- e) ¿Te han ayudado estos ejercicios a entender mejor algún aspecto de la asignatura? ¿Cuál? ¿Sabrías explicar por qué?
- f) ¿Has aprendido algo nuevo con esta actividad? ¿Qué? ¿Sabrías explicar cómo?

4.3.2.3 Descripción de las pruebas de evaluación llevadas a cabo

Las pruebas realizadas son las siguientes:

- Un cuestionario de conocimientos previos.
- Tres cuestionarios con preguntas multiopción.
- Dos actividades consistentes en la resolución de ejercicios y problemas de sistemas operativos.

- Una actividad que implica el uso de un simulador de sistemas operativos (Mustafa, 2011).

A continuación se describe brevemente cada una de las pruebas mencionadas.

Cuestionario de conocimientos previos

Este cuestionario fue diseñado con los siguientes objetivos:

- Que el profesor conozca los conocimientos previos sobre sistemas operativos de cada estudiante para poder adecuar la asignatura al grupo.
- Que el estudiante tome conciencia de sus conocimientos previos sobre la asignatura.

Para poder alcanzar estos dos objetivos, se proporcionaron las siguientes instrucciones a los estudiantes:

- Las preguntas deben ser contestadas con tus propias palabras. No consultes Internet ni ningún otro tipo de documentación para realizar este cuestionario, no copies textualmente de ninguna fuente.
- Reflexiona cada respuesta antes de contestar.
- Se valorará el tiempo y reflexión dedicados a contestar el cuestionario, y no el número de respuestas “correctas”.

En el anexo B se puede encontrar esta prueba de evaluación.

Cuestionarios con preguntas multiopción

Se diseñaron tres pruebas de evaluación formativa con 10 preguntas de opción múltiple. Las pruebas se corresponden con los contenidos del manual de la asignatura de la siguiente forma:

- **Cuestionario I.** Introducción a los sistemas operativos y gestión de procesos.
- **Cuestionario II.** Planificación de procesos. Comunicación y sincronización de procesos.

- **Cuestionario III.** Gestión de memoria. Gestión de E/S. Gestión de ficheros.

Después de contestar a cada una de las preguntas de opción múltiple, el estudiante debe justificar su respuesta, exponer las razones de su elección. La intención de que el estudiante incluya la justificación es que haga explícito su proceso de pensamiento.

Los cuestionarios se han realizado con la herramienta *moodle* del mismo nombre y su objetivo principal es el aprendizaje. Los estudiantes dispusieron de un plazo de dos semanas y de dos intentos para realizarlos. En el caso de que la respuesta seleccionada por el estudiante no sea la correcta se recibe una retroalimentación una vez que se ha finalizado el intento. En los anexos C, D y E se encuentran los tres cuestionarios usados en el curso estudiado.

Actividades con ejercicios y problemas

Se diseñaron dos actividades con ejercicios y problemas de sistemas operativos. Algunos de los ejercicios se podían resolver con lápiz y papel y para otros era necesario el uso del sistema operativo Linux. En el caso de que el estudiante tuviera dificultades con alguno de los ejercicios y problemas, se proporcionó una segunda oportunidad para realizarla sin proporcionarle la solución, sólo unas indicaciones acerca de los errores cometidos.

Estas actividades se pusieron a disposición de los estudiantes a través de la herramienta *Tarea* del sistema de gestión de aprendizaje *moodle*. En los anexos F y G se pueden encontrar las dos actividades de este tipo que fueron usadas en el curso.

Actividad que usa un simulador de Sistemas Operativos

Se diseñó una actividad que implica el uso de un simulador de sistemas operativos. La actividad está basada en un tutorial proporcionado por el autor (Mustafa, 2013) que fue adaptado para la realización en el curso estudiado.

Esta actividad se puso a disposición de los estudiantes a través de la herramienta *Tarea* del sistema de gestión de aprendizaje *moodle*. En el anexo H se puede encontrar el detalle de esta actividad.

4.4 Recogida de datos

A continuación se describe el número de estudiantes de la muestra, la forma de realizar la selección de los participantes del estudio y los documentos que se analizan en esta investigación.

4.4.1 Muestra

La muestra está constituida por 9 estudiantes del total de 13 matriculados en la asignatura. La selección de los estudiantes comprende a todos aquellos que han realizado las actividades previstas en la asignatura, es decir, que no han abandonado el curso.

4.4.2. Documentos

Para dar respuesta a las preguntas de investigación planteadas se han analizado todas las pruebas de evaluación formativa realizadas durante el curso por cada uno de los estudiantes. Se recuerdan las pruebas realizadas:

- Un cuestionario de conocimientos previos.
- Tres cuestionarios con preguntas multiopción.
- Dos actividades consistentes en la resolución de problemas de sistemas operativos.
- Una actividad consistente en el uso de un simulador de sistemas operativos (Mustafa, 2011).

4.5 Análisis de datos

El fin de una investigación cualitativa es el descubrimiento y éste tiene lugar durante el proceso de análisis (Wittrock, 1989). La técnica que se emplea para facilitar el descubrimiento es la codificación, que consiste en la creación de códigos durante el estudio de los datos. Un código en investigación cualitativa es una palabra o frase corta que se asigna a un fragmento de texto con el objetivo de capturar su esencia (Saldaña, 2012).

La mayor dificultad en la realización de una investigación cualitativa es el proceso de análisis porque no existen unas pautas generales que se puedan aplicar para cualquier caso. Por el contrario, en una investigación cualitativa existen un consenso sobre los tipos de análisis que el investigador puede usar (Miles & Huberman, 1994).

Saldaña, en su manual sobre análisis, describe con detalle 29 métodos de codificación (Saldaña, 2012). Sin embargo, ninguno de ellos se ha podido aplicar de forma directa para realizar este estudio. Como consecuencia, ha sido necesario crear un procedimiento de análisis para dar respuesta las preguntas de investigación de este estudio.

Además de difícil, el análisis cualitativo es también muy costoso en términos de tiempo porque es un proceso iterativo. Cada iteración produce un nuevo conjunto de códigos en el que se añaden códigos nuevos y se borran o reformulan los códigos antiguos. Una nueva iteración puede incluso ignorar los códigos creados con anterioridad y adoptar un procedimiento totalmente distintos.

A continuación se detalla el proceso de análisis que se ha seguido en este estudio con el fin de que se pueda conocer con detalle el proceso de descubrimiento de los resultados. Dada la naturaleza iterativa del estudio, se advierte de que para llegar al proceso descrito en los siguientes apartados se han tenido que analizar los datos del estudio en numerosas ocasiones. El análisis ha sido realizado con la ayuda del software de análisis cualitativo ATLAS.ti ("ATLAS.ti (Version 7) [Computer Software]," 2013).

Los siguientes apartados describen el proceso de análisis, que ha sido diferente para cada una de las tres preguntas de investigación planteadas. Por último, se describen algunas estadísticas que permiten valorar el esfuerzo dedicado al proceso de análisis.

4.5.1 Análisis de datos para descubrir conceptos que resultan difíciles

El análisis de datos para descubrir los conceptos que resultan difíciles se ha llevado a cabo en dos fases (Miles & Huberman, 1994). En la primera fase, se analizaron todas las pruebas de evaluación y se crearon códigos para cada una de las afirmaciones de los estudiantes que no son totalmente correctas.

En el caso de los cuestionarios con preguntas multiopción el análisis resultó algo más complejo. Se recuerda que en estos cuestionarios cada pregunta de opción múltiple consta de dos partes: la selección de la respuesta y el razonamiento que ha efectuado el estudiante para responder a la pregunta. A continuación se describe la forma de codificación utilizada para cada una de las partes mencionadas:

Selección de la respuesta:

- Si la respuesta es correcta, no se crea ningún código, ya que no se detectan dificultades.
- Si la pregunta no es correcta, se crea un código que represente la respuesta del estudiante, el distractor seleccionado. Ejemplo: “El cambio de proceso siempre se origina por una *interrupción* de reloj”.
- Si la pregunta está en blanco, se crea un código que informe de qué pregunta no ha contestado el estudiante. Ejemplo: “No contesta qué transición entre estados de un proceso no puede tener lugar en un algoritmo de planificación no preventivo”.

Justificación de la respuesta:

- Si la justificación es correcta, no se crea ningún código, ya que no se detectan dificultades.
- Si la justificación no es correcta, se crear un código redactando la justificación en conjunto con la respuesta ofrecida por el estudiante. Ejemplo: “Un mismo proceso no puede ejecutar dos operaciones wait sobre un semáforo porque esta operación es indivisible o atómica”.
- Si no hay justificación se crea un código que informe de qué respuesta no se ha justificado. Ejemplo: “No justifica por qué es posible ejecutar dos operaciones wait consecutivas sobre un semáforo”.

La segunda fase de análisis tuvo como objetivo la codificación de patrones (Miles & Huberman, 1994). En esta fase se identificaron los conceptos subyacentes en cada una de las respuestas incorrectas y se agruparon los

códigos relacionados con el mismo concepto. Estas agrupaciones dieron lugar a la respuesta de la primera pregunta de investigación: los conceptos de sistemas operativos que resultan más difíciles de comprender.

4.5.2. Análisis de datos para descubrir las dificultades que hay alrededor de cada concepto

Una vez finalizado el primer análisis, se descubrieron dos conceptos que presentaban dificultades: las *interrupciones* y los *semáforos*.

Se hicieron dos nuevos análisis de las pruebas de evaluación, uno para cada uno de los conceptos descubiertos. En esta ocasión se crearon códigos para todas las afirmaciones que los estudiantes habían realizado sobre estos conceptos, ya fueran correctas o incorrectas. El objetivo de estos análisis fue conocer qué concepciones tenían los estudiantes sobre estos conceptos para poder realizar un estudio de las causas que producían las dificultades descubiertas. En el apartado de resultados se dan detalles adicionales sobre cómo se ha realizado este análisis.

Estos análisis dieron lugar a la respuesta de la segunda pregunta de investigación de este estudio: las dificultades concretas que los estudiantes tienen alrededor de los conceptos que se resultan más difíciles de comprender.

4.5.3 Análisis de datos para descubrir efectos de la evaluación formativa

El análisis de datos para descubrir los efectos de la evaluación formativa se ha llevado a cabo en dos fases (Miles & Huberman, 1994). Durante la primera fase, se codificaron los datos identificando las afirmaciones de los estudiantes relacionadas con los efectos de la evaluación formativa. Una vez finalizado el primer análisis para todos los documentos del estudio de caso, se estudiaron todos los códigos creados para identificar patrones en las respuestas de los estudiantes.

En la segunda fase de análisis, se analizaron de nuevo todos los datos en busca de más información sobre cada uno de los patrones identificados. El objetivo de esta fase fue obtener un conocimiento más profundo del fenómeno estudiado y triangular los datos para verificar los resultados.

4.5.4 Esfuerzo dedicado al proceso de análisis

A continuación se detallan algunas estadísticas sobre el proceso de análisis que pueden dar una idea de su complejidad. Los datos proporcionados se pueden comprobar en la Figura 4.1, una captura de pantalla de ATLAS.ti, el software utilizado para la realización del análisis.

- **Tiempo dedicado a realizar el análisis.** El proceso de análisis tuvo una duración de 6 meses y comenzó cuando se obtuvieron los primeros datos sobre el estudio, en noviembre de 2012. Los primeros análisis se realizaron con el programa *Microsoft Excel* que resultó poco adecuado para las tareas de análisis. Por este motivo se decidió usar un software especialmente diseñado para el análisis cualitativo y se seleccionó *ATLAS.ti*. En la Figura 4.1 se puede comprobar que el proyecto *ATLAS.ti* comenzó el 22 de enero de 2013 y los últimos cambios fueron realizados el 2 de junio de 2014.
- **Número de documentos analizados.** El número de documentos analizados es 99. Los documentos analizados en este estudio son las pruebas de evaluación realizadas por 9 estudiantes. Cada estudiante ha realizado al menos siete pruebas. La mayoría de ellos ha realizado más de 7 pruebas, puesto que cada prueba se podía realizar dos veces. Cada una de las pruebas tiene una media de 10 preguntas o problemas. Los anexos B, C, D, E, F, G y H contienen cada una de las pruebas de evaluación realizadas por los estudiantes.
- **Número de citas.** Una *cita (quotation)* es un fragmento de texto al que se le asignado un código. El número total de citas de este estudio es 1787.
- **Número de códigos creados.** Un *código (code)* en investigación cualitativa es una palabra o frase corta que se asigna a un fragmento de texto. El número de códigos creados a lo largo de todo el proceso de análisis es de 640.

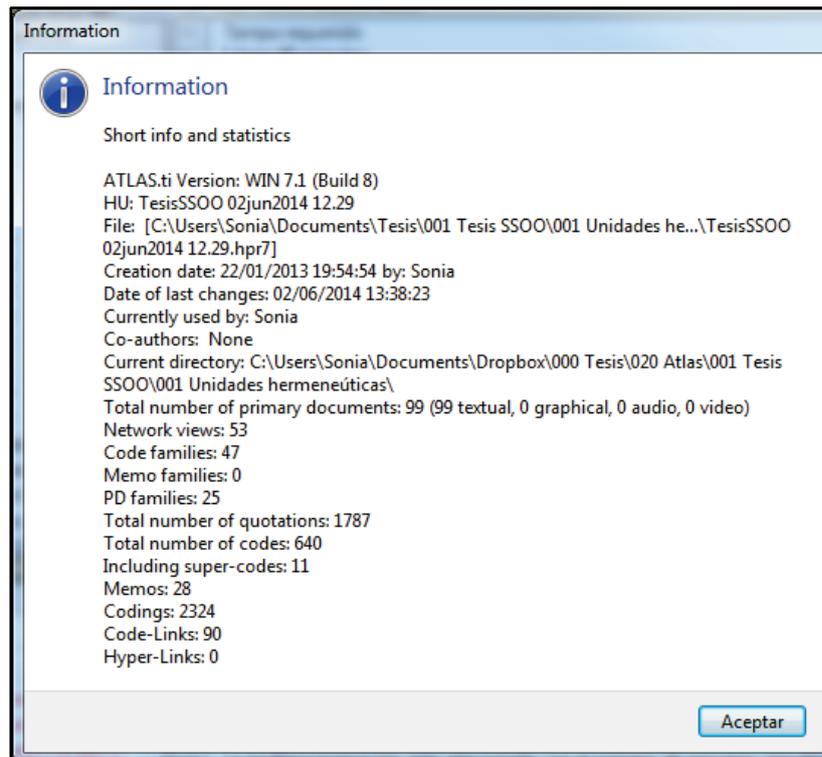


Figura 4.1. Estadísticas del proceso de análisis con ATLAS.ti

4.6 Validez, confiabilidad y ética

Toda investigación científica se ocupa de producir conocimiento válido y confiable de una forma ética. La naturaleza de la investigación cualitativa hace que estas cuestiones adopten una forma diferente que en la investigación cualitativa (Merriam, 1998).

A continuación se tratan la validez interna, la validez externa y la confiabilidad de la investigación. Finalmente, se concluye el apartado con las cuestiones éticas que se han tenido en cuenta en este estudio.

4.6.1 Validez interna

La validez interna de una investigación trata acerca de la coherencia entre los resultados de la investigación y la realidad (Merriam, 1998).

Uno de los supuestos de la investigación cualitativa es que la realidad es holística, multidimensional y está siempre cambiando. No es un fenómeno único esperando a ser descubierto, observado y medido como en la

investigación cuantitativa. Por tanto, no se puede determinar la validez valorando el isomorfismo entre datos recogidos y realidad (Merriam, 1998). La validez interna de un estudio cualitativo está relacionada con la credibilidad y adecuación de sus resultados (Miles & Huberman, 1994).

Las estrategias aplicadas en este estudio para aumentar la validez interna son las siguientes:

1. **Triangulación.** Se ha realizado una triangulación de los resultados con distintos orígenes de datos. En este caso, los distintos orígenes de datos están constituidos por las distintas preguntas de las distintas pruebas de evaluación.
2. **Puesta en común de los resultados.** La puesta en común de los resultados de este estudio se ha realizado en tres situaciones distintas. En primer lugar, se ha llevado a cabo una puesta en común de los resultados de la investigación según han ido surgiendo. En particular, esta puesta en común ha sido realizada con dos miembros del grupo de investigación al que pertenece la autora de este trabajo. La primera es el director de esta tesis, Ingeniero Electricista y Doctor en Informática, con 40 años de experiencia en docencia y sin conocimientos avanzados en la materia de Sistemas Operativos. La segunda, es una Ingeniera Informática, estudiante de doctorado, con 4 años de experiencia como docente y que ha cursado la asignatura Sistemas Operativos. Otra situación de puesta en común de los resultados ha sido durante la participación en el congreso internacional Koli Calling (Pamplona, Medinilla, & Flores, 2013). Por último, la publicación de los resultados en la revista International Journal of Engineering Education confirma que los revisores del artículo han considerado válidos los resultados.
3. **Declaración de los sesgos del investigador.** La investigadora de este estudio es Ingeniera en Informática, tiene 25 de experiencia como docente y es profesora de la asignatura que ha sido objeto de estudio, Sistemas Operativos, desde hace 5 años.

4.6.3 Validez externa

La validez externa de una investigación se refiere a la posibilidad de aplicar los resultados del estudio a otras situaciones (Merriam, 1998). Es decir, a la posibilidad de generalizar los resultados.

Con respecto a este punto, hay que tener en cuenta que la finalidad del estudio de caso no es la generalización de los resultados, sino el estudio en profundidad de un contexto concreto con el objetivo de aprender del mismo (Flyvbjerg, 2006). No obstante, los resultados de este estudio se pueden considerar hipótesis de partida para trabajos futuros posteriores.

4.6.2 Confiabilidad

La confiabilidad de una investigación se refiere a la medida en que los resultados de investigación pueden ser replicados (Merriam, 1998).

La confiabilidad es problemática en las ciencias sociales porque el comportamiento humano no es estático. Por tanto, si se repitiera un mismo estudio de caso, difícilmente se obtendrían los mismos resultados.

Por este motivo, el término confiabilidad en el sentido tradicional no tiene sentido cuando se habla de investigación cualitativa y se sugiere que en su lugar se estudie la consistencia de los resultados obtenidos con los datos recogidos (Lincoln & Guba, 1985).

Las técnicas utilizadas en este estudio para asegurar esta consistencia son las siguientes:

- **Seguimiento para auditoría.** Esta técnica consiste en que el investigador deje un rastro de cómo ha llegado hasta los resultados para que otra persona pueda juzgar la consistencia de éstos con los datos de origen (Lincoln & Guba, 1985). Para que esta auditoría pueda ser realizada este estudio incluye la siguiente información:
 - Una descripción detallada del contexto del curso (ver 4.1 Definición del problema).
 - Cómo fueron recogidos los datos (ver 4.4 Recogida de datos).

- Cómo fue realizado el análisis para obtener los distintos códigos (ver 4.5 Análisis de datos).
- Dada la complejidad del resultado relacionado con la comprensión del concepto interrupción se incluye un anexo con los códigos creados alrededor de este resultado (Anexo J). En este anexo se detalla el nombre de cada uno de los códigos, una definición del código y una lista de las citas a las que ha sido asignado el código.

4.6.4 Ética

En los estudios cualitativos, pueden surgir dilemas éticos acerca de la recogida de datos y de la difusión de los resultados (Merriam, 1998).

Con relación a estos aspectos éticos, en esta investigación se ha elaborado un documento con un consentimiento informado que han firmado los participantes del estudio. Este documento se puede encontrar en el Anexo I, y tiene en cuenta las siguientes cuestiones:

- Presenta la investigación que se va a realizar junto con sus objetivos.
- Solicita permiso para analizar las distintas pruebas de evaluación de la asignatura Sistemas Operativos que el estudiante ha realizado durante el primer semestre del curso 2012-2013.
- Garantiza al estudiante que se tomarán las medidas oportunas para asegurar el anonimato y la confidencialidad de los datos recogidos. En ningún caso se asociará ningún nombre propio con los datos.
- Aclara que la participación en el estudio es totalmente voluntaria. La decisión de no participar no implicará ningún perjuicio por parte de la Universidad a Distancia de Madrid, UDIMA.
- Ofrece la posibilidad de compartir los resultados con los estudiantes.

5. Resultados y discusión

En este capítulo se comunican los principales resultados de la investigación. En primer lugar se discuten las dificultades de comprensión de los conceptos *interrupción* y *semáforo*. Posteriormente se describen los efectos de la evaluación formativa sobre el aprendizaje. Para finalizar, se sintetizan las respuestas a las preguntas de investigación.

5.1 Dificultades en la comprensión del concepto *interrupción*

Los estudiantes muestran problemas con la comprensión del concepto *interrupción*. Los resultados de la investigación sugieren que estos problemas pueden estar provocados por la influencia del significado del término coloquial *interrupción* en el aprendizaje de la materia Sistemas Operativos.

5.1.1 Importancia del concepto *interrupción* en la asignatura Sistemas Operativos

El concepto *interrupción* es clave en la asignatura de sistemas operativos, ya que un sistema operativo es un software dirigido por *interrupciones* (Silberschatz et al., 2011). El sistema operativo no se está ejecutando de forma continua, sino que interviene en determinadas situaciones: cuando pulsamos una tecla, cuando un programa necesita comunicarse con el hardware, etc. Cada vez que el sistema operativo se ejecuta, interrumpe la tarea que está realizando el computador en ese momento.

Por otra parte, el procesador está provisto de mecanismos que permiten interrumpir el programa que se está ejecutando en un momento determinado.

Por ejemplo, cuando un dispositivo de E/S completa una operación, ocurre la siguiente secuencia de eventos (Stallings, 2011):

1. El dispositivo emite una señal de *interrupción* hacia el procesador.
2. El procesador termina de ejecutar la instrucción actual.
3. Al finalizar de ejecutar la instrucción el procesador comprueba si hay alguna petición de *interrupción* pendiente, determina que hay una, y envía un reconocimiento de señal al dispositivo que ha emitido la *interrupción*. Este reconocimiento permite al dispositivo retirar la señal de *interrupción*.
4. El procesador transfiere el control a la rutina de tratamiento de la *interrupción*. En primer lugar, guarda la información necesaria para continuar el programa en el punto en el que se ha interrumpido.
5. El procesador carga el contador de programa con la dirección en la que se encuentra la rutina de tratamiento de la *interrupción*.

En resumen, sería difícil explicar el funcionamiento de un sistema operativo sin usar las palabras *interrupción* o interrumpir. Las interrupciones son inherentes al funcionamiento de un sistema operativo.

5.1.2 Primeros resultados

Durante las primeras fases del análisis de las respuestas de los estudiantes, se percibió que el concepto *interrupción* no se había comprendido de la forma esperada, pero no se encontró una explicación clara acerca de estas dificultades. A continuación se muestran algunas de las respuestas de los estudiantes:

“Las interrupciones no sirven para mejorar la utilización del procesador puesto que una *interrupción* es una suspensión temporal de un programa y si el grado de *interrupción* es alto puede provocar que sea más lenta la ejecución de procesos ya que el procesador estaría dedicado a atender dichas interrupciones”

“Una *interrupción* es una llamada al sistema”

Una primera explicación a los hechos fue que los estudiantes *desconocían* la existencia de las interrupciones hardware, y sólo tenían en cuenta las interrupciones software para responder a las preguntas. Este hecho implicaba que los estudiantes conocían bien un tipo de interrupciones y desconocían otro, pero no se descubrió ninguna causa probable que explicara este comportamiento.

Después de realizar varias veces el análisis de los datos, surgió otra posible explicación de las respuestas. Los estudiantes comprendían el concepto *interrupción* de acuerdo a su significado coloquial y no a su significado técnico. Es decir, los estudiantes comprendían el concepto *interrupción* como el efecto de una acción que interrumpe, y no como el mecanismo del procesador que permite interrumpir un programa. Para comprobar si esta interpretación podía explicar los resultados de la investigación se llevo a cabo un estudio de los significados del término *interrupción*, tanto en los principales diccionarios como en los libros de texto. Los resultados de este estudio se encuentran en los dos apartados siguientes.

5.1.3 Significados del término *interrupción*

El estudio de los significados del término *interrupción* se ha realizado en dos lenguas: español e inglés. Aunque la lengua en la que se ha impartido la asignatura es la lengua española, la lengua inglesa puede tener una influencia muy importante ya que la mayoría de los textos de referencia de la asignatura Sistemas Operativos están escritos en inglés o son traducciones de libros de texto cuya lengua original es el inglés.

La lengua inglesa tiene dos términos diferentes para referirse al concepto de *interrupción*: el término coloquial *interruption* y el término técnico *interrupt*. Ambos se traducen al español por el mismo término: *interrupción*.

Por este motivo, cada vez que aparece el término *interrupción*, puede tratarse de:

- El término de la lengua española *interrupción*.
- La traducción del término inglés *interruption*.
- La traducción del término inglés *interrupt*.

Se muestra a continuación que cada uno de estos términos tiene varios significados. En primer lugar se estudia el significado de los términos en los principales diccionarios de cada lengua. En segundo lugar, se analizan los significados de los términos en los libros de texto de Sistemas Operativos.

5.1.3.1 Significados del término *interrupción* en los diccionarios

El estudio del significado del término *interrupción* es importante porque la mayoría de los libros de sistemas operativos no proporcionan una definición clara del término. Además, el significado del término *interrupción* en el lenguaje coloquial puede influir en la comprensión de los estudiantes (Madison, 1995).

Significados del término español *interrupción*

Para estudiar los significados del término en español se han seleccionado dos diccionarios de referencia: el Diccionario de la Real Academia de la Lengua española (Real Academia de la Lengua Española, 2000) y el diccionario de uso María Moliner (Moliner, 2003). Los significados del término *interrupción* en estos diccionarios son los siguientes:

Diccionario de la lengua española

interrupción

(Del lat. interruptio, -ōnis).

1. f. Acción y efecto de interrumpir.

Diccionario de uso del español María Moliner

interrupción

- f. Acción y efecto de interrumpir[se].

Ambos diccionarios coinciden y proporcionan dos significados distintos para el término *interrupción*:

1. Acción de interrumpir.
2. Efecto de interrumpir.

Por tanto, el término *interrupción* en español es polisémico y tiene, al menos, los dos significados anteriores. Consideremos como ejemplo la siguiente situación: estamos escribiendo un correo electrónico y mientras tanto recibimos una llamada. Según ambos diccionarios, la llamada en sí es una *interrupción* (acción) que produce a su vez la *interrupción* de la escritura (efecto).

Significados del término inglés *interruption*

Para estudiar los significados de los términos en inglés se han seleccionado dos diccionarios de referencia: el diccionario Collins y el diccionario Merriam-Webster. A continuación se encuentran los significados del término proporcionados por el diccionario Collins, en versión inglesa y americana. El término *interruption* no aparece en el diccionario Merriam-Webster.

Diccionario Collins (British English)

Interruption. (n.d). En Collins online dictionary. Recuperado de <http://www.collinsdictionary.com/dictionary/english/interruption>

1. something that interrupts, such as a comment, question, or action
2. an interval or intermission
3. the act of interrupting or the state of being interrupted

Diccionario Collins (American English)

Interruption. (n.d). En Collins online dictionary. Recuperado de <http://www.collinsdictionary.com/dictionary/american/interruption>

1. an interrupting or being interrupted
2. anything that interrupts
3. the interval during which something is interrupted; intermission

De acuerdo con estos diccionarios, el término *interruption* tiene los siguientes significados:

1. Acción de interrumpir
2. Efecto de interrumpir
3. Algo que interrumpe
4. El intervalo durante el cual algo es interrumpido

Los dos primeros significados coinciden con el significado proporcionado por los diccionarios españoles. Los otros dos significados no están recogidos de forma

explícita en los diccionarios españoles, pero podemos encontrar ejemplos en los que el término español *interrupción* se usa con dichos significados:

- Algo que interrumpe. Si una barrera interrumpe el paso, podemos decir que la barrera es una *interrupción*.
- El intervalo durante el cual algo es interrumpido. En el ejemplo de la llamada que interrumpe la escritura de un correo electrónico, podríamos decir que hemos sufrido una *interrupción* de un tiempo equivalente a la duración de la llamada.

Significados del término inglés *interrupt*

El término *interrupt* es un sustantivo técnico que no se utiliza en lenguaje coloquial, sólo tiene significado dentro del contexto de la informática. Se escribe de la misma forma que el verbo interrumpir (*interrupt*).

A continuación se encuentran los significados del sustantivo proporcionados por la versión inglesa del diccionario Collins y por el diccionario Merriam-Webster. El sustantivo *interrupt* no aparece en la versión americana del diccionario Collins.

Diccionario Collins (British English)

Interrupt. (n.d). En Collins online dictionary. Recuperado de <http://www.collinsdictionary.com/dictionary/english/interrupt>

the signal to initiate the stopping of the running of one computer program in order to run another, after which the running of the original program is usually continued.

El término técnico *interrupt* tiene un único significado según el diccionario Collins: una señal que produce una parada de un programa informático para ejecutar otro, después del cual el programa original continúa de nuevo.

El término *interrupt* es un término técnico que sólo se aplica al campo de la informática. Pero si se establece una analogía con el ejemplo de la llamada, se llamaría *interrupción* a la señal de llamada, al sonido que produce el teléfono cuando hay una llamada.

Diccionario Merriam-Webster

Interrupt. (n.d). En Merriam-Webster online dictionary. Recuperado de <http://www.merriam-webster.com/dictionary/interrupt>

a feature of a computer that permits the temporary interruption of one activity (as the execution of a program) in order to perform another; also the interruption itself.

El diccionario Merriam-Webster proporciona dos significados para el término *interrupt*:

1. La característica del computador que permite la *interrupción* temporal de una actividad
2. La *interrupción* en sí misma.

De acuerdo con el primer significado proporcionado por este diccionario, si realizamos de nuevo una analogía con la situación de la llamada, la *interrupción* sería el mecanismo del teléfono que permite que suene el timbre. Aquí se puede apreciar que el significado del término *interrupción* se complica.

En el segundo significado del diccionario Merriam-Webster para el término *interrupt* ("the *interruption* itself") se usa el término *interruption*. Por tanto, el término *interrupt* tiene además los cuatro significados analizados anteriormente para el término no técnico.

Conclusiones acerca de los significados del término *interrupt* en los diccionarios

En resumen, de acuerdo con los diccionarios consultados, el término *interrupt* tiene los siguientes significados (Figura 5.1):

1. Acción de interrumpir.
2. Efecto de interrumpir.
3. Algo que interrumpe.
4. El intervalo durante el cual algo es interrumpido.
5. Señal que inicia la parada de un programa de computador para ejecutar otro.
6. Característica de un computador que permite la *interrupción* temporal de una actividad para realizar otra.

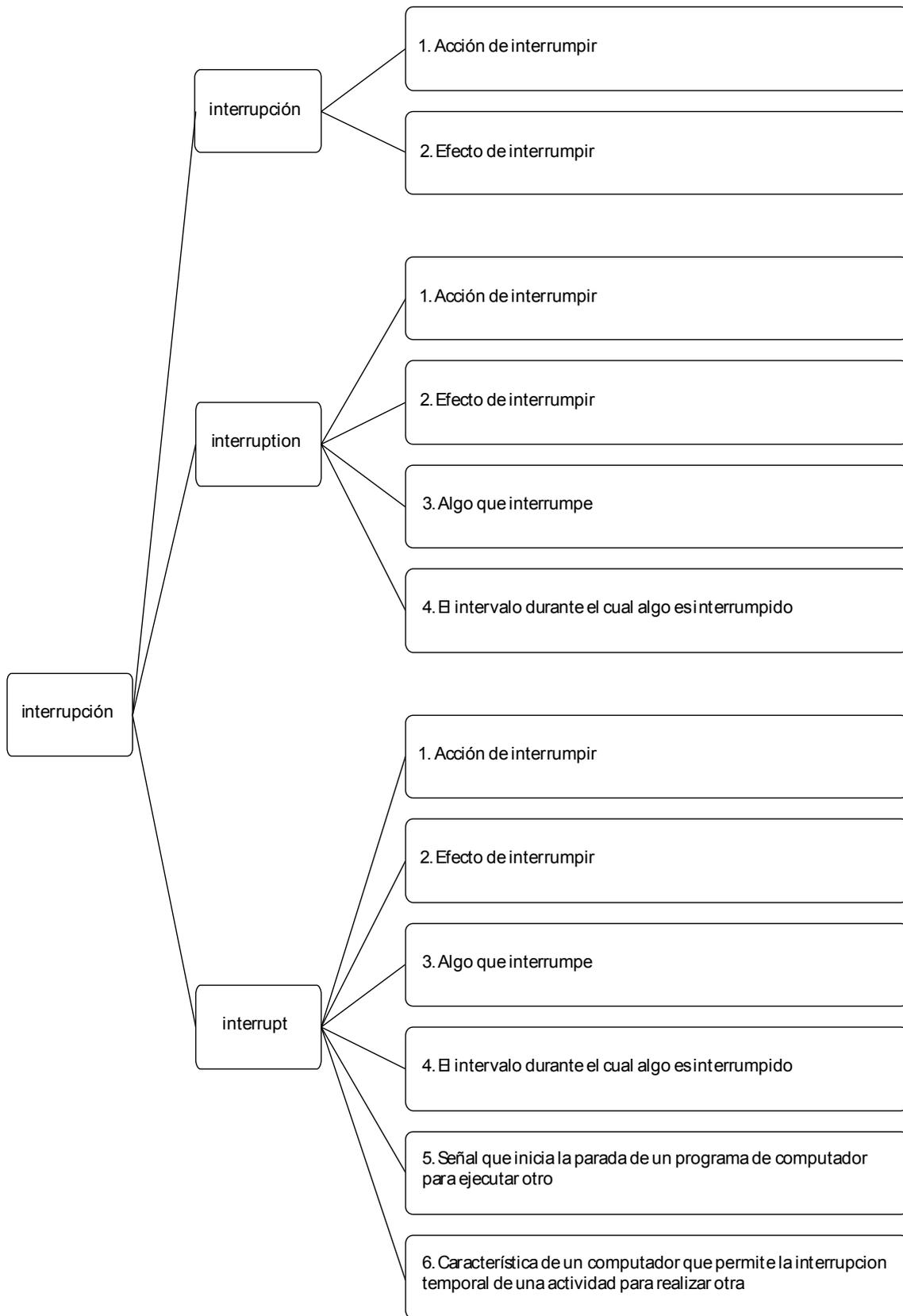


Figura 5.1. Significados del término interrupción

5.1.3.2 Significados del término *interrupción* en los libros de texto

El estudio de los significados del término *interrupción* se ha realizado en tres libros de texto representativos:

- *Operating Systems with Java* (Silberschatz et al., 2011)
- *Operating Systems: Internal and Design Principles* (Stallings, 2011)
- *Sistemas Operativos. Una visión aplicada* (Carretero et al., 2007)

La versión original de los dos primeros libros está escrita en inglés, y la del tercero en español.

Con cada uno de estos libros se ha seguido el siguiente procedimiento:

- Se han buscado en el índice alfabético:
 - los términos *interrupt* e *interruption* en los libros que están escritos en inglés.
 - el término *interrupción* en el libro que está escrito en español.
- Se ha analizado el significado de todas las apariciones de dichos términos. Para realizar esta tarea se han tenido en cuenta:
 - Las definiciones del concepto *interrupción* proporcionadas por el autor. Se han encontrado definiciones explícitas, en las que el autor ha definido el concepto *interrupción*, e implícitas, cuando el significado del término *interrupción* se podía deducir del contexto.
 - Los distintos tipos de interrupciones considerados por el autor.

Libro: *Operating System Concepts with Java*

El libro *Operating System Concepts with Java* (Silberschatz et al., 2011) está escrito en inglés, y por tanto los términos que se han buscado en el índice alfabético son *interrupt* y *interruption*. Como cabía esperar, el término *interruption* no aparece en el índice alfabético puesto que no es un término técnico.

A continuación se muestran los capítulos, epígrafes y páginas en los que se encuentra el término *interrupt*:

Tabla 5.1. Apariciones del término *interrupt* en el libro de texto *Operating System Concepts with Java*

Capítulo y epígrafe	Páginas
Chapter 1. Introduction 1.2 Computer-System Organization	8-9
Chapter 1. Introduction 1.4 Computer-System Operations	20-21
Chapter 13. I/O Systems 13.2 I/O Hardware	605-609

El término *interrupt* aparece por primera vez en este libro en la página 8, en el apartado *Computer-System Organization*. Es de especial importancia señalar que no hay una definición previa del término, lo que sugiere que los autores consideran que los lectores están ya familiarizados con el término *interrupt*.

“The occurrence of an event is usually signaled by an *interrupt* from either the hardware or the software. Hardware may trigger an *interrupt* at any time, by sending a signal to the CPU, usually by way of the system bus. Software may trigger an *interrupt* by Ejecutar a special operation called a system call.” (Página 8)

En este primer párrafo, el término *interrupt* aparece tres veces, pero no hay ninguna definición del término, ni implícita ni explícita. El texto dice que una *interrupción* puede proceder del hardware o del hardware, y que indica que ha ocurrido un evento. En el caso del hardware, una *interrupción* se pone en marcha enviando una señal a la CPU, y en el caso del software, se ejecuta una llamada al sistema. A pesar de la información proporcionada, no se determina la naturaleza de la *interrupción*, qué es exactamente una *interrupción*.

Más adelante, dentro del mismo apartado, *Computer-System Organization*, se encuentra un párrafo que aclara el significado que el autor proporciona al término:

“Interrupts are an important part of a computer architecture. Each computer design has its own interrupt mechanism, but several functions are

common. The interrupt must transfer control to the appropriate interrupt service routine. (...) The interrupt architecture must also save the address of the interrupted instruction. (...) After the interrupt is serviced, the saved return address is loaded into the program counter, and the interrupted computation resumes as though the interrupt had not occurred” (Página 8)

En este párrafo, el autor afirma que las interrupciones son una parte importante de la arquitectura del computador. Además, se explica que cada computador tiene su propio mecanismo para tratar las interrupciones, pero todos tienen varias funciones en común. Estas funciones son: (1) transferir el control a una rutina de tratamiento de la *interrupción*, (2) guardar la dirección de la instrucción que se ha interrumpido y (3) continuar con la instrucción que se había interrumpido. De estas afirmaciones se podría inferir que los autores denominan *interrupción* a la esencia del mecanismo de *interrupción* que comparten todos los computadores. Esta interpretación del término *interrupción* coincide con el significado proporcionado con el diccionario Merriam-Webster: característica de un computador que permite la *interrupción* temporal de una *interrupción* para realizar otra.

En el apartado *Operating-System Operations* aparece de nuevo el concepto *interrupción* con un significado coherente con el deducido de la aparición anterior:

“As mentioned earlier, modern operating systems are interrupt driven. If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen. Events are almost always signaled by the occurrence of an interrupt or a trap. A trap (or an exception) is a software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed.” (Página 21, apartado 1.5 Operating-System Operations)

Los autores señalan que la ejecución del sistema operativo tiene lugar cuando ocurre una *interrupción*. Es decir, el mecanismo de *interrupción* permite al sistema operativo ejecutarse cuando es necesario. Además en este párrafo los autores distinguen entre *interrupt* y *trap*. Los autores definen el término *trap* como una *interrupción* generada por software que está causada por un error o por una llamada al sistema.

Más adelante, en la página 605 y dentro del epígrafe 13.2 I/O Hardware se encuentra una definición explícita del término *interrupción*:

“The hardware mechanism that enables a device to notify the CPU is called an *interrupt*.”

En esta frase, los autores denominan *interrupción* al mecanismo hardware que permite a un dispositivo notificar a la CPU. Este significado está de acuerdo con el significado que se ha deducido anteriormente.

En los párrafos siguientes hay varias frases que corroboran el significado deducido:

“A modern operating system interacts with the interrupt mechanism in several ways.”

“The events that trigger interrupts have a common property: they are occurrences that induce the CPU to execute an urgent, self-contained routine”.

“An operating system has other good uses for an efficient hardware and software mechanism that saves a small amount of processor state and then calls a privileged routine in the kernel”

El análisis realizado hasta el momento sugiere que las interrupciones son un mecanismo del computador que consiste principalmente en salvar el estado del procesador y ejecutar una rutina urgente del kernel.

Por último, los autores describen varios usos del mecanismo de *interrupción* en un sistema operativo.

“During I/O, the various device controllers raise interrupts when they are ready for service”

“The interrupt mechanism is also used to handle a wide variety of exceptions, such as dividing by zero, accessing a protected or nonexistent memory address, or attempting to execute a privileged instruction from user mode.”

“In summary, interrupts are used throughout modern operating systems to handle asynchronous events and to trap to supervisor-mode routines in the kernel”

En resumen, en este libro de texto:

- Hay una única definición explícita del término *interrupción*: “el mecanismo hardware que permita a un dispositivo notificar a la CPU se llama interrupción”. Sin embargo la definición no es completa, dice que el mecanismo hardware se llama interrupción. Pero podría haber otras definiciones adicionales para este término.
- En cuanto a las definiciones implícitas que se pueden deducir del texto:
 - El significado del término *interrupt* se podría asociar con primer significado proporcionado por el diccionario Merriam-Webster: “característica de un computador que permite la *interrupción* temporal de una actividad para realizar otra”.
 - El análisis del texto permite deducir la siguiente definición, más precisa: mecanismo hardware del computador que consiste principalmente en salvar el estado del procesador y ejecutar una rutina urgente del kernel.
 - Los autores distinguen los siguientes tipos de interrupciones:
 - *Interrupt*
 - *Trap* (una interrupción generada por software y que está causada por un error o por una llamada al sistema)

Libro: *Operating Systems: Internals and Design Principles*

El libro *Operating Systems: Internals and Design Principles* (Stallings, 2011), está escrito en inglés, y por tanto los términos que se han buscado en el índice alfabético son *interrupt* y *interruption*. Al igual que en el libro anterior, el término *interruption* no aparece en el índice alfabético.

A continuación se muestran los capítulos, epígrafes y páginas en los que se encuentra el término *interrupt* (Tabla 5.2).

Tabla 5.2. Apariciones del término *interrupt* en el libro *Operating Systems: Internal and Design Principles*

Capítulo y epígrafe	Páginas
Chapter 1. Computer System Overview 1.4 Interrupts	34-43
Chapter 3 Process Description and Control 3.4 Process Control	157-159
Chapter 5 Concurrency: Mutual Exclusion and Synchronization 5.2 Mutual exclusion: Hardware Support	230

La primera aparición del término *interrupt* se encuentra en la página 34, en el título del apartado 1.4. *Interrupts*, que pertenece al capítulo *Computer System Overview*.

“Virtually all computers provide a mechanism by which other modules (I/O, memory) may interrupt the normal sequencing of the processor. Table lists the most common classes of interrupts.”

Tabla 5.3. Tipos de interrupciones

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user’s allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
Hardware failure	Generate by failure, such as power failure or memory parity error.

Fuente: (Stallings, 2011)

En esta primera aparición del término, se establecen cuatro tipos de interrupciones de acuerdo a su procedencia: de programa, de reloj, de E/S y de fallo de hardware.

La siguiente aparición del término se encuentra a continuación de la anterior:

“Interrupts are provided primarily as a way to improve processor utilization”

El autor sostiene que las interrupciones constituyen una manera de mejorar la utilización del procesador. A continuación, ilustra su afirmación con un ejemplo acerca de las interrupciones de E/S. Estas interrupciones mejoran la utilización del procesador porque permiten que el procesador esté libre mientras se está llevando a cabo la operación de E/S. Cuando el dispositivo requiera atención del procesador, será él mismo el que avise a éste mediante una *interrupción*.

El autor sigue utilizando el ejemplo de la E/S en los apartados siguientes del epígrafe dedicado a las interrupciones: *Interrupciones y ciclo de instrucción*, *Procesamiento de interrupciones* y *Múltiples interrupciones*.

En el apartado 3.4 *Process Control* aparece el concepto *interrupción* en el siguiente párrafo y la siguiente tabla:

“A process switch may occur any time that the OS has gained control from the currently running process. The following table suggests the possible events that may give control to the OS.”

Tabla 5.4. Mecanismos para interrumpir la ejecución de un proceso

Mechanism	Cause	Use
Interrupt	External to the execution of the current instruction	Reaction to an asynchronous external event
Trap	Associated with the execution of the current instruction	Handling of an error or an exception condition
Supervisor call	Explicit request	Call to an operating system function

Fuente: (Stallings, 2011)

En la tabla, el autor presenta una clasificación de los distintos mecanismos que existen para interrumpir la ejecución de un proceso. Son tres: *interrupciones*, *trap* y *llamadas al sistema*. Por tanto, la tabla muestra que los las interrupciones (*interrupts*) son uno de los mecanismos que permiten interrumpir un programa.

Los autores indican que las interrupciones, los *trap* y las *llamadas al sistema* son tres formas distintas de interrumpir un proceso. Por tanto, una llamada al sistema produce una *interrupción* en el proceso, un *trap* produce también una *interrupción* en un proceso y por último una *interrupción* produce una *interrupción* en el proceso. En esta última frase estamos utilizando el término

interrupción con dos significados distintos, uno para referirse a la causa, y otro para referirse al efecto.

El siguiente párrafo aclara las diferencias entre los términos *interrupt* y *trap*:

“Actually, we can distinguish, as many systems do, two kinds of systems interrupts, one of which is simply referred to as an interrupt, and the other as a trap. The former is due to some sort of event that is external to and independent of the currently running process, such as the completion of an I/O operation. The latter relates to an error or exception condition generated within the currently running process, such as an illegal file access attempt.”

Este párrafo, junto con la tabla anterior, aclara que el autor considera interrupciones también a los trap. Sin embargo, no llama interrupciones a las llamadas al sistema. Esto produce cierta confusión, ya que las llamadas al sistema interrumpen la ejecución del programa que se está ejecutando en ese momento para ejecutar el sistema operativo y por tanto atendiendo a las definiciones del término podrían considerarse una *interrupción*.

En resumen, en este libro de texto:

- No hay ninguna definición explícita del término interrupción.
- En cuanto a las definiciones implícitas que se pueden deducir del texto:
 - El significado del término *interrupt* del libro se podría asociar con primer significado proporcionado por el diccionario Merriam-Webster: característica de un computador que permite la interrupción temporal de una actividad para realizar otra.
 - El análisis del texto permite deducir la siguiente definición, más precisa: mecanismo para interrumpir la ejecución de un proceso.
- El autor proporciona dos clasificaciones distintas acerca de las interrupciones:
 - En primer lugar, el autor distingue entre interrupciones de reloj, de programa, de E/S y de fallo de hardware. A continuación se explica cada tipo de interrupción.

- Una **interrupción de programa** es generada como resultado de la ejecución de una instrucción por alguno de los siguientes motivos: desbordamiento aritmético, división por cero, intento de ejecutar una instrucción máquina ilegal y referencia a una zona de memoria no permitida para el usuario.
- Una **interrupción de reloj** es generada por un temporizador del procesador. Permite al sistema operativo realizar ciertas funciones de una forma regular.
- Una **interrupción de E/S** es generada por un controlador de E/S para indicar la finalización normal de una operación o para indicar varias situaciones de error.
- Una **interrupción de fallo de hardware** es generada por un fallo del hardware, tal como un fallo de alimentación o un error de paridad de memoria.
- En segundo lugar, el autor distingue dos tipos de interrupciones: interrupciones y trap. Las primeras están provocadas por un evento hardware y las segundas son consecuencia de la ejecución de las instrucciones del programa. Hay que notar que, a diferencia de los autores anteriores (Silberschatz et al., 2011), Stallings (Stallings, 2011) no considera que una llamada al sistema sea un trap.

Libro: *Sistemas operativos. Una visión aplicada*

El libro *Sistemas operativos Una visión aplicada* está escrito en español, y por tanto el término que se ha buscado en el índice alfabético es *interrupción*.

A continuación se muestran los capítulos, epígrafes y páginas en los que se encuentra el término *interrupt*.

Tabla 5.5. Apariciones del término *interrupción* en el libro *Sistemas Operativos Una visión aplicada*

Capítulo	Páginas
1. Conceptos arquitectónicos del computador	6-9
3. Procesos	88, 105-116

En este libro el término *interrupción* aparece por primera vez en la página 6, dentro del apartado Conceptos arquitectónicos del computador:

“Una *interrupción* se solicita activando una señal que llega a la unidad de control. El agente generador o solicitante de la *interrupción* ha de activar la mencionada señal cuando necesite que se le atienda, es decir, que se ejecute un programa que le atienda. Ante la solicitud de una *interrupción*, siempre y cuando esté habilitada este tipo de *interrupción*, la unidad de control realiza un ciclo de aceptación de *interrupción*.”

Después de esta introducción, los autores describen los pasos que realiza la unidad de control para atender a la *interrupción*, que conforman lo que los autores llaman ciclo de aceptación de introducción:

- Salvar registros del procesador.
- Cambiar el modo de ejecución a núcleo.
- Cargar un nuevo valor en el contador de programa.
- Inhibir las interrupciones (en muchos procesadores).

Más adelante los autores realizan una distinción entre interrupciones síncronas e interrupciones asíncronas.

“Se dice que la *interrupción* es síncrona cuando es consecuencia directa de las instrucciones máquinas que se están ejecutando. En el resto de los casos se dice que es asíncrona.”

A continuación, se realiza una clasificación de las interrupciones atendiendo a las distintas causas que las producen:

“Las *interrupciones* se pueden generar por diversas causas que clasificaremos de la siguiente forma:

- Excepciones hardware síncronas. [...]
- Excepciones hardware asíncronas. [...]
- *Interrupciones* externas. Se trata de interrupciones asíncronas producidas por elementos externos al procesador como son: a) El

reloj, que se analizará con detalle en la sección siguiente, b) los controladores de los dispositivos de E/S, que necesitan interrumpir para indicar que han terminado una operación o un conjunto de ellas, o que necesitan atención, y c) otros procesadores.

- Instrucciones máquina de llamada al sistema. [...]"

En esta clasificación se hace evidente el uso del término *interrupción* con dos significados distintos. Cuando los autores dicen "Las interrupciones se pueden generar por distintas causas" el término *interrupción* tiene el significado de "efecto de interrumpir". Más adelante, se da distintos nombres a las acciones que han provocado la *interrupción*: excepciones, interrupciones externas e instrucciones máquina de llamada al sistema. En esta segunda parte el término "*interrupción externa*" designa a una de las posibles causas de una *interrupción*.

Después de esta clasificación, los autores dedican un apartado a explicar el concepto de prioridad en las interrupciones y el mecanismo de inhibición de interrupciones.

La última aparición del término *interrupción* en el apartado de conceptos arquitectónicos del computador está relacionada con las interrupciones de reloj.

"El oscilador que gobierna las fases de ejecución de las instrucciones máquina se denomina reloj. Cuando se dice que un procesador es de 5 GHz, se está especificando que el oscilador que gobierna su ritmo de funcionamiento interno produce una onda cuadrada con una frecuencia de % GHz. La señal producida por el oscilador anterior, o por otro oscilador, se divide mediante un divisor de frecuencia para generar una *interrupción externa* cada cierto intervalo de tiempo. Estas interrupciones, que se están produciendo constantemente, se denominan interrupciones de reloj o tics. El objetivo de estas interrupciones es, como veremos más adelante, hacer que el sistema operativo entre a ejecutar de manera periódica."

El siguiente apartado en el que aparece el término *interrupción* es el apartado 3. *Procesos*, subapartado 3.4. *Vida de un proceso*:

"Mientras está ejecutando un proceso puede ocurrir interrupciones. Veamos detalladamente los pasos involucrados:

- Un proceso está en ejecución, por lo tanto, parte de su información reside en los registros de la máquina, que están siendo constantemente modificados por la ejecución de sus instrucciones máquina.
- Bien sea porque llega una *interrupción* externa, una excepción hardware o porque el proceso solicita un servicio del sistema operativo, el proceso para su ejecución.
- Inmediatamente entra a ejecutar el sistema operativo ya sea para atender a la *interrupción* o para atender al servicio demandado.
- La ejecución del sistema operativo, como la de todo programa, modifica los contenidos de los registros de la máquina, destruyendo sus valores anteriores”

En este fragmento se usa el término *interrupción* con dos significados distintos, de una forma similar al párrafo comentado con anterioridad. En la expresión “Mientras está ejecutando un proceso puede ocurrir interrupciones”, el término *interrupción* tiene el significado de “efecto de interrumpir”. En la frase “Bien sea porque llega una *interrupción* externa, una excepción hardware o porque el proceso solicita un servicio del sistema operativo, el proceso para su ejecución”, el término *interrupción* tiene el significado de “acción de interrumpir”, hace referencia a la causa que ha provocado la *interrupción*.

Más adelante, dentro del mismo apartado “3. Procesos”, se encuentra el subapartado “3.11 Tratamiento de interrupciones” en el que se dedican diez páginas completas a la forma en la que se tratan los distintos tipos de interrupciones en la distintas situaciones que se pueden producir.

En resumen, en este libro de texto:

- No hay ninguna definición explícita del término *interrupción*.
- En cuanto a las definiciones implícitas que se pueden deducir del texto, existen dos distintas:
 - El significado que predomina en el libro para el término *interrupción* es: *efecto de interrumpir*.

- Otro significado que aparece en el libro para éste mismo término es: acción de *interrumpir*. Es decir, el término interrupción también designa la causa de la interrupción. Los autores hablan de que la causa de una *interrupción* puede ser una *interrupción externa*.
- Los autores realizan dos clasificaciones distintas de las interrupciones:
 - Interrupciones síncronas y asíncronas.
 - Interrupciones provocadas por excepciones hardware, interrupciones externas o instrucciones máquina de llamada al sistema.

Conclusiones acerca de los significados del término *interrupción* en los libros de texto

Después de realizar el análisis de estos tres libros de texto se puede afirmar que:

- Los autores no realizan una definición explícita del término interrupción.
- Se han encontrado tres significados distintos del término interrupción en los libros de texto:
 - Acción de interrumpir
 - Efecto de interrumpir
 - Característica de un computador que permite la interrupción temporal de una actividad para realizar otra.
- Se han encontrado tres clasificaciones distintas de las *interrupciones*.

Por tanto, se puede concluir que no existe un consenso en el uso del término *interrupción*. Cada autor hace un uso distinto del mismo. Y lo que es más importante, los autores no parecen ser conscientes de que los distintos significados del término pueden causar problemas. Si lo fueran, harían una definición explícita antes de comenzar a tratar el tema de las interrupciones, y/o realizarían una discusión sobre los distintos usos del término.

5.1.5 Preguntas acerca de la comprensión del concepto *interrupción*

A continuación se detallan las cinco preguntas que han ayudado a detectar los problemas acerca del concepto *interrupción*.

Para cada una de las preguntas se proporciona la siguiente información.

- Prueba a la que pertenece. Objetivos y ámbito de la prueba.
- Enunciado de la pregunta.
- Objetivos de aprendizaje de la pregunta y clasificación de los mismos de acuerdo a la taxonomía revisada de Bloom.
- Importancia de la pregunta en la asignatura Sistemas Operativos.
- Significado o significados de *interrupción* que es necesario tener en cuenta para comprender la pregunta. Dado que el término *interrupción*, tiene múltiples significados, puede ocurrir que la pregunta no tenga sentido para alguno de los significados del término.
- Respuestas de los estudiantes. En este apartado se describen y se discuten las respuestas que los estudiantes han proporcionado para cada una de las preguntas.

5.1.5.1 Primera pregunta acerca del concepto *interrupción*

Prueba a la que pertenece. Objetivos y ámbito de la prueba

Es la pregunta número 13 del cuestionario de conocimientos iniciales. Este cuestionario fue cumplimentado al inicio del curso, antes de comenzar con el estudio del manual. Su objetivo fue obtener información acerca de los conocimientos previos de los estudiantes acerca de los conceptos más importantes de la asignatura Sistemas Operativos. Se puede encontrar una información más detallada sobre el mismo en el Anexo B.

Enunciado de la pregunta

¿Sabes qué es una *interrupción*? En el caso de que lo sepas, explícalo con tus palabras.

Categoría: Comprender

Proceso cognitivo: Interpretar

Fuente: Elaboración propia

Objetivo de aprendizaje de la pregunta

En esta pregunta se pretende que el estudiante defina el concepto de *interrupción* con sus propias palabras, que haga explícito el concepto que tiene del término *interrupción*. En la taxonomía revisada de Bloom, éste objetivo se corresponde con el proceso cognitivo interpretar, que se encuentra en la categoría comprender.

El concepto *interrupción*, de acuerdo con las recomendaciones del curriculum ACM/IEEE (ACM/IEEE-CS Joint Interim Review Task Force, 2008), se suele tratar por primera vez en una asignatura que trata sobre la organización y arquitectura de un computador y que se cursa antes de la asignatura Sistemas Operativos. Por tanto, se espera que el estudiante conozca el término informático *interrupción* antes de comenzar el curso.

Importancia de la pregunta en la asignatura Sistemas Operativos

La pregunta se realiza porque las concepciones iniciales del estudiante pueden influir en el aprendizaje (Bransford et al., 1999).

Significado del término *interrupción* en la pregunta

No se necesita otorgar un significado concreto al término *interrupción* para contestar a la pregunta, ya que lo que se desea es descubrir qué concepción de *interrupción* tiene el estudiante antes de comenzar el curso.

Respuestas de los estudiantes

En las respuestas de los estudiantes a esta pregunta se han estudiado dos aspectos:

- Los distintos significados que los estudiantes han otorgado al término *interrupción*.
- Las dificultades para la comprensión del concepto *interrupción*.

En primer lugar se muestran los distintos significados del término *interrupción* que aparecen en las respuestas de las estudiantes.

Las respuestas de los estudiantes muestran 4 significados distintos del término *interrupción*, de los seis existentes:

- Significado 2. Efecto de interrumpir
- Significado 3. Algo que interrumpe
- Significado 4. Señal
- Significado 6. Característica de un computador

Los significados del término *interrupción* que no aparecen en las respuestas de los estudiantes son los siguientes:

- Significado 1. Acción de interrumpir
- Significado 4. El intervalo durante el cual algo es interrumpido

En los siguientes epígrafes se describen las respuestas de los estudiantes que están relacionadas con cada uno de los significados.

Significado 2. Efecto

A continuación se muestran las respuestas de los estudiantes que hacen referencia al siguiente significado del término *interrupción*:

“Efecto de interrumpir”

El estudiante E01 se refiere a una *interrupción* como a la consecuencia de la ejecución de una llamada al sistema. Por tanto, se está refiriendo a la *interrupción* como un efecto.

E01: “Es un evento generado por una llamada al sistema.”

El estudiante E07 afirma que las interrupciones son detenciones de un proceso. Por tanto, se refiere también al efecto de interrumpir.

E07: “La verdad es que aún no conozco bien la diferenciación conceptual entre *interrupción* e interbloqueo, pero vagamente podría barruntar que son detenciones que provoca el SO a la ejecución de un proceso cuando otro proceso, al que se le da prioridad o turno, ejecute su propio código por o bien un tiempo determinado o bien hasta que finalice si es que ese proceso

era más importante que el anterior. De ahí que haya las que veníamos a tratar en Tecnología y Estructura de Computadores como "peticiones de *interrupción*" donde un proceso, al solicitar auxilio o servicios del SO demanda que otro proceso deponga su actividad y le ceda el procesador para hacer su labor. Como ejemplo, creo que hay una *interrupción* cada vez que escribo en el teclado o mueva (aunque si quiera sea un milímetro) mi ratón ya que estos le dicen al procesador que, en tiempo real, se vea el resultado en pantalla y, si algo estaba usando el procesador debe digamos *echarse a un lado*."

La respuesta del estudiante E10 hace referencia también al efecto de interrumpir, con la expresión "parar la ejecución de un programa". Para este estudiante, la *interrupción* es la parada en la ejecución del programa.

E10: "Creo que consiste en parar la ejecución de un programa, para resolver un evento."

Significado 3. Algo que interrumpe

A continuación se muestran las respuestas de los estudiantes que hacen referencia al siguiente significado del término *interrupción*:

"Algo que interrumpe"

El estudiante E04 define *interrupción*, como una clase de llamada al sistema que permite interrumpir. Es decir, para el estudiante, una *interrupción* es algo que interrumpe.

E04: "Creo que una *interrupción* es una clase de llamada al sistema que permite interrumpir un proceso o el uso de un dispositivo"

Significado 4. Señal

A continuación se muestran las respuestas de los estudiantes que hacen referencia al siguiente significado del término *interrupción*:

"Señal que inicia la parada de un programa de computador para ejecutar otro"

El estudiante E05, se refiere a una *interrupción* mediante las palabras "petición de atención". En estas palabras está implícito el significado de aviso, señal.

E05: “Petición de atención que se realiza al sistema, para poder utilizar desde el que invoca, los recursos que tiene disponible.”

El estudiante E06 define en primer lugar una *interrupción* como una señal hardware, lo que se corresponde con el significado de señal. En segundo lugar, proporciona otra definición de *interrupción* que se explicará en un apartado posterior.

E06: “Es una señal hardware. La forma que tiene el procesador para aceptar peticiones, interrumpir lo que está haciendo y pasar a ejecutar algo o controlar cierto dispositivo o periférico. Luego volverá a ejecutar lo que estaba haciendo. Pueden ser de distinto tipo según quien las gestione (procesador, bus o periférico) o si toma el control total de la CPU o sólo rodajas de tiempo de esta.”

El estudiante E08 usa la expresión “envía una *interrupción* al procesador para que éste atienda al nuevo proceso más prioritario”. Por tanto, para este estudiante, una *interrupción* es una señal, un aviso.

E08: “Cuando un proceso se está ejecutando en el procesador y llega uno más prioritario el planificador (PLC) de turno envía una *interrupción* al procesador para que ésta atienda al nuevo proceso más prioritario.”

Significado 6. Característica

A continuación se muestran las respuestas de los estudiantes que hacen referencia al siguiente significado del término *interrupción*:

“Característica de un computador que permite la *interrupción* temporal de una actividad para realizar otra”

Para el estudiante E02 las interrupciones son un medio para que el procesador atienda a peticiones mientras está desarrollando otras tareas.

E02: “Un *interrupción*, es una llamada al sistema, el S.O tiene una serie de interrupciones numeradas y con prioridad, esta idea fue desarrollada para que el procesador pudiese atender a peticiones que entran por estas IRQ mientras está gestionando otras tareas, es decir, el procesador tras cada ciclo de ejecución comprueba que los valores en las entradas de IRQ si están o activas, en tal caso, atendería o asignaría ciclos de CPU a dicha *interrupción*, retardando el acceso a la aplicación que le tocara el turno de

CPU. Por ejemplo las entradas de teclado, pasan por un IRQ siempre, se considera que el teclado debe tener una prioridad sobre el sistema.”

El estudiante E03 define *interrupción* como un mecanismo que permite que un programa sea interrumpido temporalmente, lo que se corresponde con la definición de “característica”.

E03: “Es un mecanismo que permite que un programa sea interrumpido temporalmente para atender una llamada del procesador, para luego seguir con la tarea como si nada.”

El estudiante E06 proporciona en su respuesta dos definiciones acerca del concepto de *interrupción*. La primera ha sido ya explicada en un apartado anterior. En la segunda, el estudiante afirma que una *interrupción* es la forma que tiene el procesador para interrumpir lo que está haciendo para pasar a hacer otra cosa. Esta segunda definición del estudiante se corresponde con el significado de “característica de un computador que permite la *interrupción* temporal de una actividad para realizar otra” (significado 6).

E06: “Es una señal hardware. La forma que tiene el procesador para aceptar peticiones, interrumpir lo que está haciendo y pasar a ejecutar algo o controlar cierto dispositivo o periférico. Luego volverá a ejecutar lo que estaba haciendo. Pueden ser de distinto tipo según quien las gestione (procesador, bus o periférico) o si toma el control total de la CPU o sólo rodajas de tiempo de esta.”

A continuación se describen las dificultades que se han descubierto en relación con la comprensión del concepto *interrupción*. Para detectar estas dificultades se han identificado las respuestas que no son totalmente correctas, que son incompatibles claramente con las concepciones aceptadas.

Se han encontrado dos tipos de dificultades:

- Dificultad 1. Definición incompleta de *interrupción*.
- Dificultad 2. Confusión con el origen de las interrupciones, cómo se producen.

En los siguientes epígrafes se describen las respuestas de los estudiantes que están relacionadas con cada una de las dificultades.

Dificultad 1. Definición incompleta de *interrupción*

El estudiante E01 afirma que una *interrupción* es un evento generado por una llamada al sistema. Esta afirmación es correcta. Sin embargo, una *interrupción* se puede producir por otras causas además de la llamada al sistema (fallo hardware, aviso del reloj, aviso de un dispositivo E/S, error en la ejecución de un programa, etc.). Por tanto, la definición que proporciona este estudiante es incompleta y puede producir problemas como se verá más adelante.

E01: “Es un evento generado por una llamada al sistema.”

Dificultad 2. Confusión acerca del origen de las interrupciones

Esta dificultad está relacionada con conocer cómo se ha producido la *interrupción*, qué ha ocurrido exactamente para provocar la *interrupción*.

Una *interrupción* se puede producir por distintas causas. A continuación se estudia cómo se produce la *interrupción* en cada uno de los posibles casos:

- *Interrupción* de reloj. El reloj envía la *interrupción* al procesador cada cierto intervalo de tiempo.
- *Interrupción* de E/S. El dispositivo de E/S envía la *interrupción* al procesador en el momento que requiera la comunicación con el mismo.
- *Interrupción* por error hardware. El hardware envía la *interrupción* al procesador.
- *Interrupción* por llamada al sistema. Cuando en un programa se ejecuta una llamada al sistema, se produce una *interrupción* mediante una instrucción ejecutada por el procesador.
- *Interrupción* por error en la ejecución del programa. (Ejemplo: división por cero). En este caso, el propio programa es el que provoca el error.

Como se ha podido comprobar, en todos los casos el que recibe la *interrupción* es el procesador. La parte del computador que envía o provoca la *interrupción* varía en cada caso: reloj, dispositivo de E/S, hardware, programa.

En los párrafos siguientes se muestran las respuestas de los estudiantes que presentan problemas relacionados con la procedencia de las interrupciones:

El estudiante E03 habla de atender una llamada del procesador. Esto indica que el estudiante cree que las interrupciones proceden del procesador, que es el procesador el que interrumpe. Sin embargo, como se ha explicado anteriormente el procesador es el que recibe las interrupciones y no el que las envía.

E03: “Es un mecanismo que permite que un programa sea interrumpido temporalmente para atender una llamada del procesador, para luego seguir con la tarea como si nada.”

El estudiante E08 afirma que el planificador envía una *interrupción* al procesador. Sin embargo, lo que ocurre en realidad es que se produce una *interrupción* de reloj cada cierto tiempo y es en la rutina de tratamiento de esta *interrupción* donde se comprueba si hay un proceso más prioritario.

E08: “Cuando un proceso se está ejecutando en el procesador y llega uno más prioritario el planificador (PLC) de turno envía una *interrupción* al procesador para que ésta atienda al nuevo proceso más prioritario.”

En resumen, se ha mostrado que antes de comenzar el curso, las respuestas de los estudiantes muestran cuatro significados distintos del término *interrupción* (Tabla 8).

Tabla 5.6. Significados de *interrupción* en la pregunta I

Significados	Estudiantes
2. Efecto de interrumpir	E01, E07, E10
3. Algo que interrumpe	E04
4. Señal	E05, E06, E08
5. Característica	E02, E03, E06

Por otra parte, se han encontrado dos dificultades en relación con el concepto *interrupción*. Los significados del término *interrupción* podrían estar asociados con las dificultades que se han encontrado. Para poder analizar esta relación, se ha construido una tabla que muestra las dificultades de los estudiantes junto con los significados que los estudiantes han otorgado al término *interrupción*.

Tabla 5.7. Dificultades y significados de *interrupción* asociados (Primera pregunta)

Estudiante	Dificultad	Significado con el que aparece
E01	Definición incompleta de <i>interrupción</i> : Una <i>interrupción</i> es evento generado por una llamada al sistema	2. Efecto de interrumpir
E03	Confusión con el origen de las interrupciones: Una <i>interrupción</i> es una llamada del procesador	5. Característica
E08	Confusión con el origen de las interrupciones: El planificador envía una <i>interrupción</i> al procesador	4. Señal

En los datos mostrados en la tabla no se percibe ninguna relación entre dificultades y significados del término *interrupción*.

En la tabla siguiente se ofrece un resumen general de los datos que se han obtenido sobre cada estudiante.

Tabla 5.8. Resumen por estudiante: significados de *interrupción* y dificultades (Primera pregunta)

Estudiante	Significados	Dificultades
E01	Efecto de <i>interrupción</i>	Definición incompleta de <i>interrupción</i> : Una <i>interrupción</i> es evento generado por una llamada al sistema
E02	Característica de un computador	
E03	Característica de un computador	Confusión con el origen de las interrupciones: Una <i>interrupción</i> es una llamada del procesador
E04	Algo que interrumpe	
E05	Señal	
E06	Señal Característica	
E07	Efecto de interrumpir	
E08	Señal	Confusión con el origen de las interrupciones: El planificador envía una <i>interrupción</i> al procesador
E10	Efecto de interrumpir	

5.1.5.2 Segunda pregunta acerca del concepto interrupción

Prueba a la que pertenece. Objetivos y ámbito de la prueba

Es la pregunta número 1 del Cuestionario I. Este cuestionario fue realizado durante la semana 7 del curso. Las preguntas del cuestionario pertenecen a los temas siguientes: introducción a los sistemas operativos y gestión de procesos. El objetivo de la prueba es valorar la comprensión de los estudiantes acerca de los conceptos tratados en el manual de la asignatura. Se puede encontrar una información más detallada sobre el cuestionario I en el Anexo C.

Enunciado de la pregunta

Pregunta 1. El controlador del dispositivo de E/S y la memoria intercambian datos directamente, sin la intervención del procesador, cuando se tiene:

- a) E/S programada
- b) E/S por interrupciones
- c) E/S mediante DMA

Categoría: Comprender

Proceso cognitivo: Comparar

Fuente: Pregunta 1.33, Página 53 (Candela et al., 2007).

Objetivo de aprendizaje de la pregunta

El objetivo principal de la pregunta es comparar distintos tipos de E/S. Su análisis puede resultar interesante para el estudio del concepto *interrupción*, puesto que dicho concepto aparece en las posibles respuestas.

En la E/S programada el procesador está ejecutando un programa que se comunica de forma continuada con el controlador del dispositivo, por tanto el procesador interviene en todo momento. En la E/S por interrupciones, el controlador del dispositivo envía una señal de *interrupción* al procesador cuando los datos están listos, y es el procesador el que se encarga de realizar la transferencia de los datos. Por último, en la E/S mediante DMA, el controlador del dispositivo intercambia datos con la memoria sin la intervención del procesador. Cuando la transferencia de datos finaliza, se envía una señal de *interrupción* al procesador para comunicar que la operación ha finalizado. Por tanto, la respuesta correcta es la respuesta c), E/S mediante DMA.

Importancia de la pregunta en la asignatura Sistemas Operativos

La pregunta es importante dentro de la asignatura Sistemas Operativos porque relaciona las interrupciones con las operaciones E/S. Una de las ideas más importantes a este respecto es que con la ayuda de las interrupciones, el procesador puede ejecutar otras instrucciones mientras una operación de E/S está en marcha (Stallings, 2011).

Significado del término *interrupción* en la pregunta

No se necesita otorgar un significado concreto al término *interrupción* para contestar a la pregunta. El estudiante debe conocer en qué consiste cada uno de los mecanismos de E/S mencionados en las respuestas. Tal y como se ha explicado anteriormente, el concepto *interrupción* está relacionado con la E/S por *interrupciones* y también con la E/S con *DMA*.

Respuestas de los estudiantes

En las respuestas de los estudiantes a esta pregunta se han estudiado los mismos aspectos que en la pregunta anterior:

- Los distintos significados que los estudiantes han otorgado al término *interrupción*.
- Las dificultades para la comprensión del concepto *interrupción*.

Como ya se ha mencionado el objetivo principal de la pregunta es comparar los distintos tipos de E/S. Por tanto, hay estudiantes que no hacen explícito en su respuesta lo que entienden por *interrupción*. El análisis que se ofrece a continuación sólo muestra las respuestas de los estudiantes que hacen referencia al concepto *interrupción*.

En cuanto a los distintos significados del término *interrupción*, sólo aparece uno, el de señal:

“Señal que inicia la parada de un programa de computador para ejecutar otro”

A continuación se muestran las respuestas de los estudiantes que están relacionadas con el significado mencionado.

Significado 4. Señal

El estudiante E01 menciona que el periférico avisa al procesador con una *interrupción*. Esta afirmación está relacionada con el significado de “señal”. Este estudiante no había mencionado este significado en el cuestionario de conocimientos iniciales, aparece por primera vez en esta pregunta.

E01: “Mediante DMA, el controlador del dispositivo se encarga de transferir los datos entre periférico y memoria, porque en la programada, se necesita que se esté ejecutando un programa que use E/S, y por interrupciones es el periférico el que avisa al procesador con una *interrupción*.”

El estudiante E06 indica en su respuesta que una *interrupción* avisa del requerimiento de E/S, lo que sugiere que el significado está relacionado con señal. Este significado ya había sido mencionado antes por este mismo estudiante en la pregunta anterior.

E06: “E/S mediante DMA significa direct acces memory, por tanto los dispositivos de entrada-salida pueden acceder directamente a la memoria sin pasar por el procesador, ni requerir interrumpir al procesador con interrupciones. Esto acelera el intercambio de información con la memoria y no ralentiza al procesador, pues mientras tanto sigue haciendo su trabajo, esto es PERMITE CONCURRENCIA entre el procesador y la entrada o salida. Las otras opciones, programada y por *interrupción* si requieren intervención del procesador, ya sea por que tiene el procesador un programa en ejecución para acceder al dispositivo de entrada-salida (NO HAY CONCURRENCIA entre procesador y entrada o salida), o por que una *interrupción* le avisa del requerimiento de entrada-salida y debe decidir si darle paso y cuando (SI PERMITE CONCURRENCIA entre procesador y entrada o salida).”

El estudiante E08 afirma que el dispositivo avisa al procesador a través de una *interrupción*, lo que sugiere que una *interrupción* es una forma de avisar, una señal. Este significado ya había sido mencionado antes por este mismo estudiante en la pregunta anterior.

E08: “En la E/S mediante DMA (Acceso Directo a Memoria) es el controlador del dispositivo el que se encarga directamente de transferir los datos entre el periférico y la memoria sin interrumpir al procesador. Esto permite que el procesador pueda dedicarse a ejecutar otro programa,

mientras dure el intercambio. Una E/S programada no permite concurrencia y el procesador está dedicado a la ejecución del programa de E/S. En el caso de E/S por interrupciones es el propio dispositivo de E/S el que avisará al procesador, aunque también permite concurrencia como ya he comentado el propio dispositivo avisa al procesador a través de una *interrupción*.”

En cuanto a las dificultades con el concepto *interrupción*, no se ha encontrado ninguna en las respuestas de esta pregunta.

En resumen, no todos los estudiantes han hecho referencia al concepto *interrupción* en su respuesta a esta pregunta. Los tres estudiantes que lo han hecho han mencionado los términos “aviso” o “avisar”, que están relacionados con el concepto de señal. A continuación se muestra una tabla con el resumen de los significados del término *interrupción* en esta pregunta.

Tabla 5.9. Significados de interrupción en la primera pregunta

Significados	Estudiantes
4. Señal	E01, E06, E08

En la siguiente tabla se pueden encontrar los significados del término *interrupción* que han mostrado los estudiantes en las dos preguntas analizadas hasta el momento.

Tabla 5.10. Significados de *interrupción*. Primera y segunda pregunta.

Significados	Primera Pregunta I	Segunda Pregunta
2. Efecto de interrumpir	E01, E07, E10	
3. Algo que interrumpe	E04	
4. Señal	E05, E06, E08	E01, E06, E08
5. Característica	E02, E03, E06	

En la tabla se puede apreciar que sólo dos estudiantes mantienen el mismo significado del término *interrupción* en las dos preguntas.

5.1.5.3 Tercera pregunta acerca del concepto interrupción

Prueba a la que pertenece. Objetivos y ámbito de la prueba

Es la pregunta número 5 del Cuestionario I. Este cuestionario fue realizado durante la semana 7 del curso. Las preguntas del cuestionario pertenecen a los temas siguientes: introducción a los sistemas operativos y gestión de procesos. El objetivo de la prueba es valorar la comprensión de los estudiantes acerca de los conceptos tratados en el manual de la asignatura. Se puede encontrar una información más detallada sobre el cuestionario en el Anexo C.

Enunciado de la pregunta

Pregunta 5. Las interrupciones sirven para mejorar la utilización del procesador.

- a) Verdadero
- b) Falso

Categoría: Comprender

Proceso cognitivo: Inferir

Fuente: Testbank. Chapter 1. TRUE/FALSE Questions. Question 8 (Stallings, 2011)

Objetivo de aprendizaje de la pregunta

En esta pregunta, el estudiante debe inferir si las interrupciones sirven para mejorar la utilización del procesador. Dado que el libro utilizado en la asignatura no realiza de forma explícita la afirmación, los estudiantes necesitan inferir la misma a partir de otra información contenida en el mismo.

La información se puede inferir de los conceptos de E/S programada y E/S por interrupciones. Como ya se ha mencionado en la pregunta anterior, con la ayuda de las interrupciones el procesador puede ejecutar otras instrucciones mientras una operación de E/S está en marcha. Por tanto, se puede decir que las interrupciones mejoran la utilización del procesador porque permiten aprovechar el tiempo de espera de las operaciones de E/S. La respuesta correcta es a) Verdadero.

Importancia de la pregunta en la asignatura Sistemas Operativos

La pregunta es importante dentro de la asignatura Sistemas Operativos porque hace reflexionar sobre las ventajas del mecanismo de interrupciones.

Significado del término *interrupción* en la pregunta

En esta pregunta existe un problema con uno de los significados del término *interrupción*: “efecto de interrumpir”. Si el estudiante interpreta el término “*interrupción*” de esta forma, puede pensar que las interrupciones son algo negativo para el procesador puesto que producen paradas en su actividad.

Para que la pregunta tenga sentido, hay que considerar otros significados del término como el de señal, o el de característica, o incluso el de acción de interrumpir. La clave para comprender la pregunta es pensar en cómo suceden las interrupciones, y no en el efecto que tienen. Las interrupciones mejoran la utilización del procesador porque liberan al procesador de estar ejecutando un programa que pregunta continuamente si una operación de E/S ha terminado o si existe algún problema con el hardware.

Respuestas de los estudiantes

En las respuestas de los estudiantes a esta pregunta se han estudiado los mismos aspectos que en las preguntas anteriores:

- Los distintos significados que los estudiantes han otorgado al término *interrupción*.
- Las dificultades para la comprensión del concepto *interrupción*.

En primer lugar se muestran los distintos significados del término *interrupción* que aparecen en las respuestas de los estudiantes.

Las respuestas de los estudiantes muestran 2 significados del término *interrupción* de los 6 existentes:

- Significado 2. Efecto de interrumpir
- Significado 6. Característica de un computador

En cuanto a las dificultades, se ha descubierto una única dificultad: los estudiantes que piensan que las interrupciones no sirven para mejorar la utilización del procesador.

En esta ocasión, los significados de las respuestas y las dificultades se discuten de forma conjunta, puesto que están directamente relacionados.

Dificultad 3. Las interrupciones no sirven para mejorar la utilización del procesador y Significado 2. Efecto.

El estudiante E01 dice que las interrupciones no sirven para mejorar la utilización porque sirven para cambiar al modo privilegiado del sistema operativo. Es cierto que cuando ocurre una *interrupción*, se cambia al modo privilegiado del sistema operativo. Sin embargo, este hecho no implica que las interrupciones no sirvan para mejorar la utilización del procesador. En la respuesta del estudiante se aprecia que el estudiante asocia el término *interrupción* con el efecto provocado por una *interrupción* y no con el mecanismo con el que se produce la *interrupción*. Como el estudiante no piensa en el mecanismo, no es capaz de inferir que las interrupciones mejoran la utilización del procesador.

E01: “Sirven para cambiar al modo privilegiado del sistema operativo.”

El estudiante E08 dice que si se producen muchas interrupciones la ejecución de procesos se hace más lenta. En esta respuesta, al igual que en la anterior, el estudiante asocia el término *interrupción* con el efecto de interrumpir y no con el mecanismo que permite interrumpir.

E08: “Entiendo que es falso puesto que una *interrupción* es una suspensión temporal de un programa, no hace que se mejore el uso del procesador sino que lo interrumpe y si el grado de *interrupción* es alto podría provocar que incluso fuera más lenta la ejecución de procesos puesto que tendríamos al procesador dedicado a atender dichas interrupciones.”

En resumen, la respuesta incorrecta de estos dos estudiantes parece estar provocada por el significado que han asociado los estudiantes con el término *interrupción*. La asociación del término *interrupción* con el efecto de interrumpir y no con el mecanismo que produce la *interrupción* les ha impedido inferir la respuesta correcta a la pregunta.

Significado 2. Efecto

A continuación se muestran las respuestas de otros estudiantes que también han asociado el término *interrupción* con el significado “efecto de interrumpir” y sin embargo han contestado que las interrupciones sirven para mejorar la utilización del procesador.

E02: “Las interrupciones mejoran la utilización del procesador, mediante el uso de prioridades aplicadas en las distintas líneas de IRQ para ser procesadas antes por el procesador.”

E04: “Creemos que la respuesta es sí, pero debe ser matizada, en tanto que existen interrupciones que son externas al propio procesador. De todos modos, atendiendo al funcionamiento interno, consideramos que las interrupciones permiten una mejor utilización del procesador en tanto que permiten un cambio de proceso, es decir, que un proceso que está en ejecución pase a otro estado, como bloqueado (p.ej. esperando a que termine una operación de E/S) o listo (esperando que se le asigne tiempo para poder utilizar el procesador) Precisamente al "ordenar" la utilización del procesador por medio del citado mecanismo del cambio de estado de proceso, las interrupciones sirven para mejorar la utilización de aquél.”

E05: “Efectivamente, en función del tipo de *interrupción* que se esté produciendo se puede mejorar el uso eficiente del procesador. Un ejemplo claro sería en el uso de la planificación ROUND-ROBIN, donde cuando a un proceso en ejecución, se le termina la rodaja de tiempo asignada, se produce una *interrupción*, que permite que otros procesos en espera puedan hacer uso de la CPU. Esas interrupciones facilitan que todos los procesos que requieren hacer uso de la CPU, estén en igualdad de condiciones para poder utilizarlo.”

E07: “Efectivamente, merced a las interrupciones un proceso puede tener el control del procesador como mínimo cuando le llegue su turno programado con anterioridad (si se emplea un algoritmo de planificación de tiempo compartido o de tipo round robin) y puede llevarse a término. De no existir las interrupciones, un proceso no se podría ejecutar hasta que finalizase el anterior o anteriores que estuviesen en la cola de tareas como ocurre con el procesamiento tipo batch. En los procesos interactivos, a diferencia de los de proceso por lotes, es necesario que el procesador atienda a interrupciones cuando el usuario u otro proceso demande la atención del procesador para otra tarea. Si no fuera porque las interrupciones invoquen

al procesador, éste sólo podría trabajar en lotes y eso siempre que no hubiese errores pues, de nuevo, si tampoco hubiese interrupciones por errores, el ordenador no podría responder a ellos resolviéndolos saltando a otro proceso de subsanación del error o bien dándole el turno a otro proceso alternativo que, de otra manera no podría haber sido atendido.”

Las respuestas anteriores afirman que las interrupciones mejoran la utilización del procesador porque permiten cambiar de proceso, y de esta forma se puede repartir el tiempo del procesador o ejecutar procesos prioritarios. El problema con estas respuestas es que han interpretado el concepto “utilización de la CPU” de una forma distinta de la que se pretendía en la pregunta. Ellos entienden el término utilización como uso, en sentido coloquial. Y por tanto, consideran que poder interrumpir procesos constituye un mejor uso del procesador. Sin embargo el significado del término “utilización del procesador” que indica el libro de texto utilizado en el curso es muy distinto:

“La utilización de CPU es el porcentaje del tiempo que está ocupada la CPU, bien ejecutando programas de usuario o bien código del sistema operativo, y puede oscilar entre 0% y 100%.”

Significado 6. Característica

Por último se presentan las respuestas que han contestado y justificado la pregunta de acuerdo a lo esperado. En todas ellas los estudiantes han interpretado el término *interrupción* como un mecanismo, lo que coincide con la definición 6:

“Característica de un computador que permite la *interrupción* temporal de una actividad para realizar otra”

E06: “Creo que la respuesta sería verdadero, pues si se produce una excepción síncrona (por el propio programa) o asíncronas (fallo de algún dispositivo), si no hubiese forma de avisar al procesador con los bits del registro de su estado, para dar paso a la *interrupción*, el procesador podría quedar inutilizable, o ejecutando procesos que no llevarían a ninguna parte (bucles, espera eterna etc). Además, los periféricos que se comunican por interrupciones hardware con el procesador lo hacen de forma más eficiente que si fuese por programa, consumiendo menos ciclos de reloj, permitiendo que otros procesos sigan en estado de ejecución mientras se atiende a las interrupciones. Las interrupciones permiten avisar al procesador de por

ejemplo una entrada de un dispositivo, evitando así el tener que controlar periódicamente la entrada para ver si ya está disponible el dato o no, haciendo más útil al procesador que delega esta función de control al propio dispositivo o al bus del sistema.”

E10: “Se evita que un proceso siga ejecutándose mientras espera una entrada por parte del usuario, por ejemplo.”

En resumen, el problema principal con esta pregunta ha sido que la mayoría de los estudiantes (7 de 9) han interpretado el término *interrupción* como *efecto de interrumpir* y no cómo el mecanismo que ha producido la *interrupción*. Esta interpretación parece ser la causa de las respuestas incorrectas, por haber impedido pensar en las ventajas del mecanismo de interrupciones, que era el objetivo de la pregunta.

5.1.5.4 Cuarta pregunta acerca del concepto interrupción

Prueba a la que pertenece. Objetivos y ámbito de la prueba

Es la pregunta número 8 del Cuestionario I. Este cuestionario fue realizado durante la semana 7 del curso. Las preguntas del cuestionario pertenecen a los temas siguientes: “Planificación de procesos” y “Sincronización y comunicación de procesos”. El objetivo de la prueba es valorar la comprensión de los estudiantes acerca de los conceptos tratados en el manual de la asignatura. Se puede encontrar una información más detallada sobre el cuestionario en el Anexo C.

Enunciado de la pregunta

Pregunta 8. ¿Qué código crees que necesita ejecutarse con las interrupciones inhibidas?

- a) Ninguno, ya que podrían perderse interrupciones.
- b) Todo el código del sistema operativo.
- c) Ciertas partes críticas del código del sistema operativo, como por ejemplo el cambio de contexto.

Categoría: Comprender

Proceso cognitivo: Inferir

Fuente: Recursos para el instructor TEST N1 Pregunta 20 (Carretero et al., 2007).

Objetivo de aprendizaje de la pregunta

En esta pregunta, el estudiante debe inferir qué código necesita ejecutarse con las interrupciones inhibidas. De acuerdo con la taxonomía revisada de Bloom, la pregunta se clasificaría en la categoría *comprender* y el proceso cognitivo al que hace referencia es *inferir*.

El libro utilizado en la asignatura no discute este punto. Por tanto, es necesario que el estudiante infiera la respuesta a partir de otra información contenida en el libro. En este caso, la respuesta se puede inferir de la siguiente forma:

- Durante un cambio de contexto, se guarda el contenido del registro contador del programa y de otros registros para que el proceso puede continuar más tarde en el mismo punto en el que se paró.
- Si durante estas operaciones ocurriera una *interrupción*, la copia del registro contador de programa y otros registros podría fallar.
- Si la copia del registro contador de programa y otros registros fallara, el proceso no podría continuar en el mismo punto en el que se paró.
- Por tanto, es necesario inhibir las interrupciones durante un cambio de contexto.

Importancia de la pregunta en la asignatura Sistemas Operativos

La pregunta es importante dentro de la asignatura Sistemas Operativos porque hace reflexionar al estudiante sobre los inconvenientes del mecanismo de interrupciones y sobre cómo debe hacerse la programación de un sistema operativo para evitar los problemas derivados del uso de interrupciones.

El problema reside en que una *interrupción* puede ocurrir en cualquier momento. Esto supone que cualquier código que esté ejecutando el procesador se puede parar en cualquier momento para continuarse más tarde. Pero hay ciertas operaciones que son críticas, que deben ser completadas antes de que se atienda a cualquier *interrupción*.

Significado del término *interrupción* en la pregunta

De antemano, no existe ningún conflicto con los significados del término *interrupción* en esta pregunta. Se pregunta sobre el código que no se puede

interrumpir. Por tanto, no es necesario discernir entre mecanismo y efecto de interrumpir. Se puede entender en ambos sentidos.

Respuestas de los estudiantes

En las respuestas de los estudiantes a esta pregunta se han estudiado dos aspectos:

- Los distintos significados que los estudiantes han otorgado al término *interrupción*.
- Las dificultades para la comprensión del concepto *interrupción*.

En esta pregunta, todos los estudiantes han contestado de forma correcta a la pregunta y por tanto afirman que ciertas partes críticas del código del sistema operativo, como por ejemplo el cambio de contexto, necesitan ejecutarse con las interrupciones inhibidas. Sin embargo, la dificultad que se encontrado es que ningún estudiante ha justificado de forma apropiada su respuesta, como se puede apreciar a continuación.

Dificultad 4. No se justifica por qué motivo ciertas partes críticas del sistema operativo se deben ejecutar con las interrupciones inhibidas

Los estudiantes E01 y E02 han dejado la segunda parte de la pregunta (Justifica tu respuesta) en blanco.

Los estudiantes E03, E07 y E08 han respondido a la parte de la pregunta "Justifica tu respuesta", pero no aportan argumentos que expliquen por qué no se puede interrumpir este tipo de código.

E03: "Para que el propio computador pueda interrumpir un proceso para devolver el control al sistema operativo."

E07: "Juzgo que la última opción es bastante cercana a lo que debería ocurrir, esto es, que cuando ciertas operaciones críticas del S.O. se estén ejecutando como el cambio de contexto o el proceso de subsanar un error grave, el procesador no debería ser interrumpido bajo ningún concepto. De hecho, eso es lo que me ha parecido observar como usuario cuando mi sistema operativo "cae" en uno de esos "errores graves" y no responde a absolutamente nada. En caso de que dos interrupciones concurren, se debe atender a la de mayor prioridad y, de ser relativa a una parte crítica de

código, la atención que el procesador le debe conceder debe ser exclusiva.”

E08: “Aquellas que se ejecuta código del sistema operativo, es decir en modo privilegiado. Es el propio sistema operativo quien decide sobre el tratamiento de este tipo de interrupciones, es quien tiene que valorar su procesamiento.”

Los estudiantes E04, E05 y E06 aportan más información que los estudiantes anteriores sobre los posibles efectos de una *interrupción*, pero no explican qué podría ocurrir exactamente si se interrumpen ciertas partes críticas del código del sistema operativo.

E04: “Evidentemente, el apartado a) no es correcto porque necesariamente se ejecuta código, pero no todo el código del sistema operativo, por lo que el apartado b) tampoco es correcto. Con las interrupciones inhibidas se ejecutan ciertas partes críticas del código del sistema operativo, como el cambio de contexto, en tanto que en éste se contienen los datos que están presentes en los registros del procesador.”

E05: “Entiendo que en el diseño de los sistemas operativos hay partes de código crítico, que cuando se ejecutan, no deben permitir la atención de ningún tipo de operación de *interrupción*, pues poner en riesgo la integridad del sistema.”

E06: “Hay situaciones críticas en las que una *interrupción* podría provocar situaciones inesperadas, fallas de seguridad etc. Si se está cambiando de contexto, hasta que el cambio no se produzca no se atenderán interrupciones pues el sistema operativo no estaría preparado para atenderlas. En otras muchas situaciones pasaría lo mismo, pues hay procesos o microoperaciones que una vez iniciados deben completarse sin ninguna *interrupción*. Sistemas en tiempo real podrían ser otro ejemplo, pues cuando llega el momento de ejecutar un proceso, este no se puede demorar más de un tiempo límite, o no sería un sistema operativo en tiempo real. Entonces ciertas interrupciones seguro estarán inhibidas temporalmente.”

En resumen, la dificultad descubierta con esta pregunta es que la mayoría de los estudiantes no han justificado por qué motivo ciertas partes críticas del código del sistema operativo necesitan ejecutarse con las interrupciones inhibidas.

Sobre los significados del término *interrupción* que los estudiantes han usado para responder la pregunta no podemos decir nada, puesto que las respuestas de los estudiantes no proporcionan información al respecto.

Por tanto, no tenemos ninguna indicación sobre el motivo de la dificultad detectada.

5.1.5.5 Quinta pregunta acerca del concepto interrupción

Prueba a la que pertenece. Objetivos y ámbito de la prueba

Es la pregunta número 6 del Cuestionario II. Este cuestionario fue realizado durante la semana 11 del curso. Las preguntas del cuestionario pertenecen a los temas siguientes: “Planificación de procesos” y “Sincronización y comunicación de procesos”. El objetivo de la prueba es valorar la comprensión de los estudiantes acerca de los conceptos tratados en el manual de la asignatura. Se puede encontrar una información más detallada sobre el cuestionario en el Anexo D.

Enunciado de la pregunta

Pregunta 6. El cambio de proceso (también llamado cambio de contexto):

- a) Lo realiza el planificador.
- b) Modifica la entrada en la tabla de procesos del proceso desalojado.
- c) Siempre se origina por una *interrupción* de reloj.
- d) Se produce siempre que el proceso abandona la cola de procesos en espera y pasa a la cola de procesos preparados.

Categoría: Comprender

Proceso cognitivo: Inferir

Fuente: Página 9. Pregunta 1.2.4 (Casillas & Iglesias, 2004)

Objetivo de aprendizaje de la pregunta

En esta pregunta, el estudiante debe inferir qué información es cierta acerca del cambio de contexto. De acuerdo con la taxonomía revisada de Bloom, la pregunta se clasificaría en la categoría *comprender* y el proceso cognitivo al que hace referencia es *inferir*.

El objetivo principal de la pregunta es la comprensión del mecanismo de cambio de contexto. Su análisis puede resultar interesante para el estudio del concepto *interrupción*, puesto que dicho concepto aparece en una de las posibles respuestas.

Importancia de la pregunta en la asignatura Sistemas Operativos

La pregunta es importante dentro de la asignatura Sistemas Operativos porque relaciona los conceptos de *interrupción* y *cambio de contexto*.

Significado del término *interrupción* en la pregunta

En esta pregunta aparece un tipo concreto de interrupciones, las asociadas con el reloj del sistema. En principio no se prevé ninguna confusión acerca de su significado.

Respuestas de los estudiantes

En las respuestas de los estudiantes a esta pregunta se han estudiado dos aspectos:

- Los distintos significados del término *interrupción*.
- Dificultades con la comprensión del concepto *interrupción*.

En cuanto a los significados que los estudiantes han otorgado al concepto *interrupción*, no se puede deducir nada de las respuestas proporcionadas por los estudiantes.

Con referencia a las dificultades, los estudiantes E01, E03 y E10 creen que el cambio de proceso se origina siempre por una *interrupción* de reloj.

El estudiante E01 justifica su respuesta diciendo que una *interrupción* sirve para realizar un cambio de contexto.

E01: "Puesto que una *interrupción* sirve, entre otras cosas, para realizar un cambio de contexto."

Sin embargo, una *interrupción* no siempre origina un cambio de contexto. Por ejemplo, la mayoría de las veces que ocurre una *interrupción* de reloj no se produce un cambio de contexto. Una *interrupción* de reloj implicará un cambio de contexto sólo si se realiza alguna decisión de planificación dentro de la rutina de tratamiento. Si consideramos otros tipos de *interrupción*, la llamada al sistema tampoco produce un cambio de contexto.

El estudiante E03 no aporta argumentos para justificar su respuesta:

E03: "Excepto la respuesta a) creo que las demás las respuestas participan del cambio de contexto."

Por último, el estudiante E10 justifica su respuesta haciendo una afirmación que tampoco es cierta.

E10: "Se realiza un cambio de contexto, siempre que hay una *interrupción* de reloj."

En este estudiante se observa una dificultad adicional. Piensa que siempre que hay una *interrupción* de reloj, se origina un cambio de contexto.

En resumen, en esta pregunta se observan tres dificultades que están relacionadas entre sí:

- Dificultad 5. El cambio de contexto siempre se origina por una *interrupción* de reloj.
- Dificultad 6. Siempre que hay una *interrupción* de reloj, se origina un cambio de contexto.
- Dificultad 7. Una *interrupción* sirve para realizar un cambio de contexto.

Es decir, lo que ocurre es que los estudiantes asocian las interrupciones con los cambios de contexto. El origen posible de esta confusión es el significado del término coloquial *interrupción*, asociado con el efecto del interrumpir y no con su causa.

5.1.5.6 Conclusiones acerca de la comprensión del concepto interrupción

La Tabla 5.11 resume los hallazgos que se han realizado a través de las preguntas mencionadas en los apartados anteriores.

Las conclusiones sobre los datos analizados en este estudio son las siguientes:

- Las respuestas de los estudiantes muestran 4 significados distintos del término *interrupción*, de los seis existentes:
 - Significado 2. Efecto de interrumpir
 - Significado 3. Algo que interrumpe
 - Significado 4. Señal
 - Significado 6. Característica de un computador

Los significados del término *interrupción* que no aparecen en las respuestas de los estudiantes son los siguientes:

- Significado 1. Acción de interrumpir
- Significado 4. El intervalo durante el cual algo es interrumpido

En resumen, en este apartado se ha mostrado que el término *interrupción* tiene distintos significados y además se ha comprobado que los estudiantes interpretan el término de cuatro formas distintas. Por otra parte, el análisis de los libros de texto sugiere que los profesores no son conscientes de la posibilidad de que un estudiante pueda interpretar el término de distintas formas.

- Los significados que los estudiantes otorgan a las distintas apariciones del término *interrupción* varían en cada una de las preguntas, no se mantienen constantes en el tiempo. Este hecho tiene dos implicaciones. La primera es que los estudiantes varían su interpretación del término *interrupción* en función del contexto y no siempre lo hacen de forma correcta, ya que existen ocasiones en que la interpretación que seleccionan no les permite resolver el problema propuesto. La segunda es que los estudiantes en ningún momento han detectado ninguna

situación de ambigüedad con el término, ya que no han realizado ninguna pregunta al respecto en ningún momento del curso ni tampoco han argumentado sus respuestas teniendo en cuenta los distintos significados del término. Por tanto, los estudiantes creen comprender a qué se refiere exactamente el término interrupción en cada momento, no son conscientes de los múltiples significados del término ni de las dificultades que puede ocasionar este hecho.

- Las respuestas de los estudiantes muestran siete tipos de dificultades:
 - Dificultad 1. Definen de forma incompleta el término interrupción
 - Dificultad 2. Confunden el origen de las interrupciones.
 - Dificultad 3. Afirman que las interrupciones no sirven para mejorar la utilización del procesador.
 - Dificultad 4. No justifican por qué motivo ciertas partes críticas del sistema operativo se deben ejecutar con las interrupciones inhibidas.
 - Dificultad 5. Creen que el cambio de proceso se origina siempre por una interrupción de reloj.
 - Dificultad 6. Creen que una interrupción sirve para realizar un cambio de contexto.
 - Dificultad 7. Creen que siempre que hay una interrupción, se origina un cambio de contexto.

- En cuanto a las posibles causas de las dificultades mencionadas:
 - La definición de forma incompleta del término interrupción (dificultad 1) se podría explicar por el hecho de que los estudiantes no son conscientes de la polisemia del término interrupción y realizan sus razonamientos con la primera interpretación que realizan del término.

- La confusión acerca del origen de las interrupciones (dificultad 2) podría tener como causa el hecho de que los estudiantes asocien el término interrupción en la mayoría de las ocasiones con el significado de *efecto*. De este modo, conocen los efectos que las interrupciones causan, pero ignoran *cómo* se producen estas interrupciones, sólo piensan en las consecuencias que originan.
- El hecho de que los estudiantes piensen que las *interrupciones* no sirven para mejorar la utilización del procesador (dificultad 3) está relacionado con su interpretación del término *interrupción* como efecto. La asociación del término interrupción con el efecto de interrumpir y no con el mecanismo que produce la interrupción les ha impedido realizar la inferencia necesaria para contestar a la pregunta. Por ejemplo, un estudiante dice que si el procesador es interrumpido en muchas ocasiones, la ejecución de procesos se hará más lenta.
- La causa de que los estudiantes no justifiquen por qué ciertas partes críticas del sistema operativo se deben ejecutar con las interrupciones inhibidas (dificultad 4) no ha quedado clara en este estudio. Los datos no permiten deducir una causa posible porque las justificaciones de esta pregunta están en blanco, los estudiantes no han contestado nada.
- Por último, la causa de que los estudiantes asocien las *interrupciones* con los *cambios de contexto* (dificultades 5, 6 y 7) parece ser la interpretación del término interrupción como un *efecto*. Ellos consideran que un *cambio de contexto* producen una *interrupción* en la ejecución del proceso, e interpretan de esta forma el término *interrupción*.

Tabla 5.11. Resumen de la comprensión del concepto *interrupción*

Primera pregunta		Segunda pregunta		Tercera pregunta		Cuarta pregunta		Quinta pregunta			
Significados	Dificultades	Significados	Dificultades	Significados	Dificultades	Significados	Dificultades	Significados	Dificultades		
E01	Efecto	Definición incompleta de interrupción: Una interrupción es evento generado por una llamada al sistema	—	Señal	—	Efecto	Las interrupciones no sirven para mejorar la utilización del procesador.	—	No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas	—	El cambio de proceso siempre se origina por una interrupción de reloj Una interrupción sirve para realizar un cambio de contexto
E02	Característica	—	—	—	—	Efecto	—	—	No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas	—	—
E03	Característica	Confusión con el origen de las interrupciones: Una interrupción es una llamada del procesador	—	—	—	—	—	—	No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas	—	El cambio de contexto siempre se origina por una interrupción de reloj.
E04	Algo que interrumpe	—	—	—	—	Efecto	—	—	—	—	—
E05	Señal	—	—	—	—	Efecto	—	—	—	—	—
E06	Señal Característica	—	—	Señal	—	Característica	—	—	—	—	—

Primera Pregunta		Segunda pregunta		Tercera pregunta		Cuarta pregunta		Quinta pregunta	
Significados	Dificultades	Significados	Dificultades	Significados	Dificultades	Significados	Dificultades	Significados	Dificultades
E07	Efecto	—	—	Efecto	—	—	No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas	—	—
E08	Señal	Confusión con el origen de las interrupciones: El planificador envía una interrupción al procesador	—	Efecto	Las interrupciones no sirven para mejorar la utilización del procesador.	—	No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas	—	—
E10	Efecto	—	—	Característica	—	—	No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas	—	El cambio de proceso siempre se origina por una interrupción de reloj. Siempre que hay una interrupción de reloj, se origina un cambio de contexto.

5.2. Dificultades en la comprensión del concepto *semáforo*

En este estudio se han descubierto dificultades en la comprensión del concepto *semáforo*. Sin embargo, se disponen de muy pocos datos para averiguar las causas de estas dificultades, como se comprobará a continuación. Por este motivo, la discusión de este resultado es mucho más breve que la del concepto *interrupción*.

5.2.1 Preguntas acerca de la comprensión del concepto *semáforo*

A continuación se detallan las tres preguntas que han ayudado a detectar los problemas acerca del concepto *semáforo*.

En este caso todas las preguntas pertenecen a la misma prueba, el cuestionario número II, que se puede encontrar completo en el Anexo D. Las preguntas que se discuten a continuación son las preguntas 3, 4, 7 y del mencionado cuestionario.

5.2.1.1 Primera pregunta acerca del concepto *semáforo*

¿Cuál es el número máximo de procesos que pueden quedarse bloqueados en un semáforo que se inicializó con un valor de 2?

- A. 1
- B. 2
- C. Un número teóricamente ilimitado

Justifica tu respuesta

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N1, pregunta 21.

Objetivo de aprendizaje de la pregunta

El objetivo de aprendizaje de la pregunta es inferir el número máximo de procesos que pueden quedarse bloqueados en un semáforo que se inicializó con un valor de 2. Para ello, el estudiante debe interpretar el código del semáforo. De acuerdo con la taxonomía revisada de Bloom, este objetivo se corresponde con el proceso cognitivo *interpretar* y pertenece a la categoría *comprender*.

Respuestas de los estudiantes

Hay cuatro estudiantes que han respondido de forma incorrecta a esta pregunta. Pero no se dispone ninguna información sobre el razonamiento realizado por el estudiante, puesto que no han justificado su respuesta.

5.2.1.2 Segunda pregunta acerca del concepto semáforo

Cuando un proceso hace una operación *signal* sobre un semáforo:

- A. Siempre se incrementa en una unidad el valor del semáforo.
- B. Se incrementa en una unidad el valor del semáforo cuando este valor es positivo.
- C. Se incrementa en una unidad el valor del semáforo cuando este valor es negativo.

Justifica tu respuesta

Fuente: adaptado de (Casillas & Iglesias, 2004), página 75, cuestión 3.2.7.

Objetivo de aprendizaje de la pregunta

El objetivo de aprendizaje de la pregunta es interpretar la operación *signal* de un *semáforo*. Para ello, el estudiante debe interpretar el código del semáforo. De acuerdo con la taxonomía revisada de Bloom, este objetivo se corresponde con el proceso cognitivo *interpretar* y pertenece a la categoría *comprender*.

Respuestas de los estudiantes

Hay dos estudiantes que han respondido de forma incorrecta a esta pregunta. Pero no se dispone ninguna información sobre el razonamiento realizado por el estudiante, puesto que no han justificado su respuesta.

5.2.1.3 Tercera pregunta acerca del concepto semáforo

¿Es posible ejecutar dos operaciones *wait* consecutivas sobre un semáforo?

A. Sí

B. Sí, siempre que sean procesos diferentes

C. No

Justifica tu respuesta

Fuente: adaptado de (Casillas & Iglesias, 2004), página 75, cuestión 3.2.7.

Objetivo de aprendizaje de la pregunta

El objetivo de aprendizaje de la pregunta es inferir si es posible ejecutar dos operaciones *wait* consecutivas en un *semáforo*. Para ello, el estudiante debe interpretar el código del semáforo. De acuerdo con la taxonomía revisada de Bloom, este objetivo se corresponde con el proceso cognitivo *interpretar* y pertenece a la categoría *comprender*.

Respuestas de los estudiantes

Esta pregunta ha sido contestada de forma incorrecta por todos los estudiantes. La mayoría de las respuestas no han sido justificadas y otras ofrecían explicaciones sin argumentos. Dos de las respuestas sugieren una posible causa para los problemas de comprensión de los semáforos.

A continuación se expone, a modo de ejemplo, una de las respuestas que ofrecen una explicación sin argumentos:

E06: “Los procesos se ejecutan concurrentemente y dos procesos distintos o un mismo proceso pueden ejecutar una operación *wait*. Si el recurso está libre pasa uno de ellos e incluso si un mismo proceso está haciendo uso del recurso puede realizar otra operación *wait*. Pero si no está libre permanecerán bloqueados a la espera de haya una operación *signal* y pueda entrar uno de ellos.”

Este estudiante ofrece una explicación sobre el funcionamiento de los semáforos, pero no dice en que basa su respuesta. Debería haberse apoyado en el código del *semáforo*, puesto que es lo que define su comportamiento.

A continuación se presentan las respuestas que sugieren una posible causa para los problemas de comprensión del concepto *semáforo*.

E07: “En el ejemplo del portero de discoteca, cuando recibe la señal P o bajada de semáforo para que bloquee el acceso a más procesos (personas a la sala) las personas se empiezan a acumular en una cola hasta que otra operación V o subida de semáforo (up) producida por la salida de una persona de la discoteca permita que el primero que se encoló pueda pasar. Esto implica que sólo hace falta un proceso para poner en wait al “proceso” portero para que éste no deje entrar a más procesos-cliente hasta nuevo aviso de localidades-recurso libres (el aforo son los recursos). Creo pues que sólo puede haber una operación wait pues me parece que con una es suficiente y más serían redundantes.”

E07: “Es como si en un establecimiento, los clientes que quisieran usar los sanitarios cerrados tomasen una llave decrementando las que quedan. Inversamente, cuando abandonasen los servicios devolverían dicha llave incrementando los recursos/aseos. Así, pueden, si hay más de un recurso suceder más de una operación wait o toma de llave seguidas pero sólo por distintas personas-proceso.”

Este estudiante ha respondido que no es posible realizar dos operaciones *wait* consecutivas en un semáforo. La lectura del código le hubiera permitido responder de forma afirmativa a la pregunta. Sin embargo, ha establecido un simil con una situación real que le ha llevado a una conclusión incorrecta.

5.2.2 Conclusiones acerca de la comprensión del concepto *semáforo*

En resumen, se han encontrado dificultades con la comprensión de un mecanismo de sincronización clásico: el *semáforo*. Para entender el funcionamiento de un semáforo, se ha comprobado que un estudiante ha realizado una analogía con una situación real.

Esto origina un problema, ya que el semáforo es una creación, un mecanismo inventando para resolver problemas de sincronización entre procesos

concurrentes. Por tanto, sus características están definidas por el código de sus operaciones y no por las similitudes con situaciones reales.

La dificultad encontrada sobre el mecanismo de sincronización semáforo consiste en que los estudiantes no pueden explicar su comportamiento en los distintos supuestos planteados.

5.3 Efectos de la evaluación formativa sobre el aprendizaje

Hemos descubierto tres efectos provocados por la evaluación formativa llevada a cabo en el curso: (1) reflexión profunda acerca de los conceptos de la asignatura, (2) incertidumbre en la resolución de las pruebas de evaluación y (3) toma de conciencia de los progresos y las dificultades en el aprendizaje.

5.3.1 Reflexión acerca de los conceptos de la asignatura

La evidencia empírica sugiere que ha existido una reflexión profunda sobre los conceptos de la asignatura Sistemas Operativos. La reflexión está presente en las respuestas de los estudiantes de dos formas: explícita e implícita.

Forma explícita

A continuación se muestra un conjunto representativo de las respuestas que hacen explícita la reflexión acerca de la asignatura. Las preguntas de la evaluación que han provocado estas respuestas son las preguntas sobre conocimiento metacognitivo. Los códigos E01, E02, E03, etc. identifican las respuestas de cada uno de los estudiantes.

Pregunta: ¿Qué opinas sobre los cuestionarios de la asignatura? ¿Te han parecido difíciles? ¿Fáciles? ¿Interesantes? ¿Aburridos? Explica los motivos de tu respuesta.

Respuesta E01: “Los cuestionarios de preguntas multi-opción no me han parecido muy difíciles, sólo que son de pensar muchísimo las cosas antes de responder, lo cual me parece genial ya que muchas veces se contestan las cosas sin pensar lo suficiente.”

Pregunta: ¿Crees que los cuestionarios te han ayudado a comprender mejor la asignatura? Justifica tu respuesta.

Respuesta E04: “Sin duda los cuestionarios me han ayudado a entender mejor la asignatura, porque para su realización he tenido que pensar y reflexionar sobre los contenidos teóricos, y ello supone la mejor asimilación de éstos.”

Pregunta: ¿Qué opinas sobre lo que hemos realizado en la asignatura hasta el momento? ¿Ha sido fácil? ¿Difícil? ¿Interesante? ¿Aburrido? ¿Puedes explicar por qué?

Respuesta E05: “La asignatura me resulta bastante interesante, requiere realizar un razonamiento de los principales conceptos”.

Pregunta: ¿Qué ejercicio de esta actividad te ha parecido más difícil? ¿Puedes explicar por qué?

Respuesta E05: “El ejercicio segundo, quizá hasta entender que la respuesta no está en los libros, sino que hay que realizar una reflexión de todos los conceptos, para poder aplicarlos en cada caso.”

Pregunta: ¿Te han ayudado estos ejercicios a entender mejor algún aspecto de la asignatura? ¿Cuál? ¿Sabrías explicar por qué?

Respuesta E06: “Cuando resuelves problemas siempre queda más claro y *te obliga a pensar* cuál es el siguiente paso o proceso que va a ejecutar el procesador (dependiendo del algoritmo etc)”

Pregunta: ¿Qué ejercicio de esta actividad te ha parecido más complicado? ¿Puedes explicar por qué?

Respuesta E08: “La mayoría de los ejercicios no suelen plantear problemas, eso sí han sido de mucho meditar”.

Como se ha podido comprobar en las preguntas y respuestas anteriores, los estudiantes han hecho explícita su reflexión sin haberles preguntado de forma directa. Los datos del estudio muestran que la casi totalidad de los estudiantes que han realizado la evaluación formativa han sido conscientes de su proceso reflexivo y lo consideran positivo para el aprendizaje de la asignatura. Estos hechos son indicios de que las preguntas sobre conocimiento metacognitivo han cumplido con el fin para el que estaban diseñadas, hacer a los estudiantes conscientes de su proceso de aprendizaje.

Forma implícita

A continuación se encuentran dos respuestas representativas que muestran de

forma implícita la reflexión del estudiante. Se muestran también las preguntas de las que proceden: una pregunta multi-opción y un problema. Ambas preguntas tienen como objetivo la comprensión de conceptos: la primera está relacionada con el proceso cognitivo inferir y la segunda con el proceso cognitivo comparar.

Pregunta multi-opción

Objetivo de aprendizaje: Inferir las circunstancias y eventos que causan un cambio de proceso.

A process switch:

- a). is performed by the scheduler.
- b). modifies the entry in the process table of the process evicted.
- c). is always caused by a clock interruption.
- d). occurs whenever a process leaves the waiting process queue and enters in the ready process queue.

Justify your Respuesta correcta

Respuesta E04: “La opción "a" es descartable puesto que el cambio de proceso lo realiza el activador.

La opción "c", una *interrupción* de reloj origina un cambio de proceso pero no es la única opción por la que se origina (E/S, fallo de memoria, etc).

La opción "d", es descartable puesto que en realidad un cambio de contexto se produce cuando un proceso está En ejecución y pasa a otro estado bien sea Bloqueado o Listo.

La opción "b", una de las acciones que se realizan en un cambio de contexto es la actualización de la tabla de procesos del proceso que estaba En ejecución, es decir actualiza su BCP, por lo que entiendo es la opción correcta.”

Problema.

Objetivo de aprendizaje: Comparar dos posibles implementaciones del algoritmo Round-Robin

Supongamos un sistema operativo con una planificación round-robin con un tiempo de 0,25 μ s entre dos rodajas de tiempo (independientemente de si ocurre o no un cambio de proceso). Si se ejecutan tres trabajos de 6 μ s, 3 μ s y 8 μ s.

Contesta las siguiente preguntas:

¿Cuál es el rendimiento si la rodaja de tiempo es de 1 μ s?

¿Cuál es el rendimiento si la rodaja de tiempo es de 0,5 μ s?

¿En cuál de los dos casos presentados anteriormente el rendimiento es mayor? ¿Cómo lo explicarías?

Respuesta E02: De ambos casos, como podemos observar cuando más pequeña es la rodaja de tiempo, peor es el rendimiento de la CPU. Por tanto, el peor caso es el 0,5 μ s de rodaja de tiempo, esto es debido a que en los pasos entre rodajas se pierde tiempo y por tanto al incrementar el número de rodajas empeora el rendimiento.

La mayoría de las explicaciones proporcionadas por los estudiantes han permitido comprobar que han tenido lugar los procesos cognitivos previstos por los objetivos de aprendizaje (comparar, inferir, etc). Por tanto, los resultados sugieren que las preguntas han cumplido con el fin para el que el que estaban diseñadas, promover el aprendizaje significativo. Además, las calificaciones obtenidas por los estudiantes en los exámenes finales soportan este resultado, ya que todos los estudiantes que realizaron todas las pruebas de la evaluación formativa aprobaron la asignatura.

5.3.2 Desconcierto en la resolución de las pruebas de evaluación

El análisis de los datos sugiere que los estudiantes han pasado por una fase de confusión, vacilación y duda al enfrentarse a las pruebas de la evaluación formativa. A continuación se muestra un conjunto de respuestas que representan la situación de incomodidad e inseguridad de los estudiantes.

Pregunta: ¿Qué opinas sobre lo que hemos realizado en la asignatura hasta el momento? ¿Ha sido fácil? ¿Difícil? ¿Interesante? ¿Aburrido? ¿Puedes explicar por qué?

Estudiante E01: “Para mí esta asignatura es muy interesante, junto con programación son las que más me gustan. Quizá me ha resultado algo difícil realizar la actividad porque no sabía cómo hacer todo sabiendo que iba a estar bien, es decir, no tengo la certeza de que todo lo que haya pensado pueda ser completamente correcto.”

Respuesta E03: “Ha sido para mí difícil porque he invertido mucho tiempo, en gran parte porque son cosas desconocidas para mí. Además, creo que simplificar demasiado los textos hace más difícil su comprensión, también he echado en falta ejemplos y ejercicios.”

Respuesta E08: “Hasta la resolución de estos ejercicios me ha parecido una asignatura asequible, el manual no me ha planteado problemas, pero las preguntas de estos ejercicios me han resultado difíciles”

Respuesta E10: “El manual es bastante teórico y las actividades son más prácticas, por lo que tienes que buscarte un poco la vida para saber cómo hacerlas. La mayor dificultad ha sido comprender cómo hacer los ejercicios”.

Pregunta: ¿Qué ejercicio de esta actividad te ha parecido más complicado? ¿Puedes explicar por qué?

Estudiante E04: “Me ha resultado más complicada la parte relativa a los problemas, no tanto por su dificultad intrínseca, sino más bien por la preocupación de interpretar correctamente los enunciados propuestos para proporcionarles una solución adecuada.”

Respuesta E07: “El ejercicio 1 me pareció difícil no por no llegar a su solución fácilmente sino por no haber hecho nunca ejercicios del mismo tipo. De todas formas, su resolución era más fácil de lo que parecía”

Las respuestas anteriores sugieren que los estudiantes no estaban habituados a tener este tipo de pensamiento reflexivo para resolver las evaluaciones. Por el contrario, la forma de proceder habitual de los estudiantes para responder a las preguntas y ejercicios parece ser la siguiente:

- Para responder a las preguntas multi-opción, los estudiantes buscan las palabras clave de la pregunta de forma directa en el manual de la asignatura, disponible en formato PDF.
- Para responder a los problemas, los estudiantes buscan un problema exactamente igual al planteado y copian su forma de realización, sin reflexionar por qué es necesario realizar ciertos pasos.

Los estudiantes han encontrado dificultades al resolver las evaluaciones porque las pruebas planteadas en nuestro curso no se pueden resolver con este tipo de comportamiento. Puesto que nuestro objetivo ha sido promover el aprendizaje significativo, las respuestas a las preguntas multi-opción no se pueden encontrar directamente en el libro y no se han realizado problemas exactamente iguales a los propuestos en las evaluaciones.

5.3.3 Toma de conciencia de los progresos y las dificultades en el aprendizaje

Los resultados sugieren que los estudiantes, además de reflexionar sobre los conceptos de la asignatura, han tomado conciencia de su proceso de aprendizaje y han hecho explícitos sus progresos y sus dificultades.

A continuación se muestra un conjunto representativo de respuestas que hacen explícita la toma de conciencia de los progresos en el aprendizaje de los estudiantes:

E04: “Sin duda los cuestionarios me han ayudado a entender mejor la asignatura, porque para su realización he tenido que pensar y reflexionar sobre los contenidos teóricos, y ello supone la mejor asimilación de éstos.”

E05: “Los problemas me han ayudado a entender mejor la estructura global por procesos y como el sistema reparte juego para un óptimo rendimiento.”

E06: “Los problemas ayudan a comprender conceptos como tiempo de residencia, rendimiento.”

En las respuestas anteriores se observa que uno de los estudiantes piensa que los cuestionarios le han obligado a reflexionar sobre los conceptos de la asignatura y esto ha supuesto una mejor comprensión de los mismos. Otros dos estudiantes explican que la evaluación les ha permitido comprender mejor ciertos conceptos de la asignatura Sistemas Operativos como proceso, tiempo de residencia y rendimiento. El análisis de la totalidad de los datos indica que la mayoría de los estudiantes han sido conscientes de sus progresos y explican cómo la evaluación formativa ha promovido el aprendizaje.

A continuación se muestra un conjunto representativo de respuestas que hacen explícita la toma de conciencia de las dificultades en el aprendizaje de los estudiantes.

E10: “Los cuestionarios me han ayudado a comprender mejor la asignatura, ya que he comprobado que en algunos casos no había comprendido los conceptos mediante la lectura del manual”.

E02: “Los ejercicios más complicados han sido los ejercicios 1,3 y 4, todos ellos porque considero que necesito aclarar conceptos. Tengo dificultades

con los conceptos de rendimiento, uso de cpu, uso de I/O y tiempo de respuesta, tengo un problema de entender conceptos para poder calcularlos.”

E03: “La verdad es que todos los problemas al ser tan diferentes me han ayudado a conocer y entender la asignatura, insisto nueva para mí, porque todos han aportado algo diferente. El problema 4 es el que más difícil me ha parecido porque me ha faltado saber colocar cada proceso en su tiempo adecuado.”

En las respuestas anteriores se muestra que uno de los estudiantes afirma que los cuestionarios le han ayudado a detectar conceptos que no había comprendido durante la lectura del manual de la asignatura. Otro de los estudiantes afirma que no puede calcular los conceptos de rendimiento, uso de CPU, uso de I/O y tiempo de respuesta porque no los comprende. Un último estudiante declara que no ha podido colocar los procesos en el tiempo adecuado. El análisis de la totalidad de los datos indica que la mayoría de los estudiantes han sido conscientes de sus dificultades, la evaluación formativa les ha ayudado a detectar las áreas específicas en las que necesitan mejorar.

Además de las cuestiones relacionadas con la asignatura Sistemas Operativos, los estudiantes manifiestan haber aprendido cuestiones relacionadas con su forma de estudiar. Uno de los estudiantes afirma que cambiará la forma de repartir el tiempo de estudio de las asignaturas, aplicará un algoritmo de gestión de proceso Round Robin.

E03: “Por otra parte sí que me ha parecido interesante comprobar cómo funcionan los distintos algoritmos de planificación, cosa que voy a utilizar a la hora de repartir el tiempo de estudio de las asignaturas. He aprendido que en el futuro fijaré un tiempo de dedicación a cualquier asignatura, pasado el cual pasaré a estudiar las otras asignaturas”

Otro estudiante ha aprendido que la lectura comprensiva del manual puede no ser suficiente, que es necesario poner esos conocimientos a prueba, tal y como se ha realizado en la evaluación formativa llevada a cabo en el curso.

E08: “He aprendido a no confiarme de los conocimientos teóricos que llevo. Es necesario llevarlos a la práctica para comprobar que los he comprendido bien”

El descubrimiento de los dos últimos efectos de la evaluación formativa (incertidumbre y toma de conciencia de progresos y dificultades) corrobora la utilidad de las preguntas sobre conocimiento metacognitivo puesto que dichas preguntas han conseguido el fin deseado: que los estudiantes reflexionen sobre su proceso de aprendizaje.

5.4 Resumen de la sección resultados y discusión

Se han cumplido los objetivos del trabajo dando respuesta a las preguntas de investigación planteadas. A continuación se sintetizan los resultados obtenidos para cada pregunta de investigación.

Los conceptos de sistemas operativos que han resultado más difíciles de comprender en el curso *online* estudiado han sido las *interrupciones* y los *semáforos*. Además, alrededor de estos conceptos se han identificado las dificultades específicas y sus posibles causas. Las dificultades descubiertas acerca de los semáforos apoyan los resultados de los trabajos (Kolikant et al., 2000) y (Ben-David Kolikant, 2004). El resto de dificultades identificadas no han sido estudiadas en la literatura existente.

Por otra parte, se han descubierto tres efectos de la evaluación creada para el curso *online* sobre el aprendizaje de los estudiantes.

Uno de los efectos que ha provocado la evaluación formativa es una reflexión profunda acerca de los conceptos de la asignatura. Este hecho supone que la evaluación formativa creada ha cumplido con uno de los fines para los que estaba diseñada: promover el aprendizaje significativo.

Otros estudios sobre evaluación formativa han promovido el aprendizaje significativo en distintos contextos y con distintos medios (Chung et al., 2006; Crisp & Ward, 2008). El primer trabajo citado se realizó en una universidad presencial y consistió en una evaluación formativa asistida por ordenador con preguntas basadas en el método del caso. En este contexto, la tecnología realizó aportaciones tales como un *feedback* automático al estudiante y la disponibilidad de la prueba a través de Internet. En el segundo estudio citado, se usó en las clases presenciales una herramienta que permite celebrar sesiones síncronas con los estudiantes para la resolución de problemas. En este contexto,

la tecnología realizó aportaciones tales como el anonimato para hacer preguntas al profesor y una representación gráfica del problema en la pantalla del ordenador. En el presente estudio, se han utilizado distintos tipos de pruebas de evaluación seleccionadas mediante la taxonomía revisada de Bloom. Tal y como se ha explicado en el apartado de metodología, se han modificado las pruebas para adaptarlas a un entorno *online*. Las modificaciones realizadas han tenido éxito y pueden servir como referencia para diseñar evaluaciones formativas en entornos *online*.

Otro de los efectos de la evaluación formativa es el desconcierto sentido por los estudiantes al enfrentarse a las pruebas de evaluación. Las pruebas de nuestra evaluación planteaban a los estudiantes situaciones nuevas que no se podían resolver únicamente a través del reconocimiento o del recuerdo como los estudiantes hubieran deseado. Este efecto, que puede parecer negativo, es difícil de evitar, puesto que las situaciones nuevas en la evaluación son esenciales para promover el aprendizaje significativo. Si se intenta evitar el desconcierto, se corre el riesgo de que los estudiantes memoricen procedimientos para resolver los problemas y se supere la asignatura sin llegar a comprenderla, tal y como indican investigaciones clave en el campo de la física (Hestenes et al., 1992).

Los dos efectos anteriores, reflexión y desconcierto, están concatenados. Los resultados de nuestro estudio están de acuerdo con la teoría de Dewey que distingue dos subprocesos en todo pensamiento reflexivo: (a) un estado de confusión, vacilación y duda, y (b) un acto de búsqueda de hechos que sirvan para corroborar o invalidar lo que hemos supuesto (Dewey, 2007).

Por último, la evaluación formativa ha provocado un tercer efecto, que los estudiantes tomen conciencia de sus progresos y dificultades. El segundo y el tercer efecto, desconcierto y toma de conciencia de progresos y dificultades, sugieren que la evaluación formativa ha conseguido promover el conocimiento metacognitivo, uno de los fines para los que estaba diseñada.

Las preguntas que se han usado en el estudio para provocar que los estudiantes reflexionen sobre su conocimiento están inspiradas en preguntas de las entrevistas realizadas en una tesis de referencia sobre Educación en Informática

en un contexto presencial (Madison, 1995). En nuestro contexto *online*, no es posible realizar actividades síncronas con todos los estudiantes y por tanto, en lugar de realizar entrevistas se utilizó un cuestionario de preguntas abiertas. En estas condiciones adversas, puede ser más complicado obtener información de los estudiantes. Sin embargo, los resultados sugieren que el cuestionario ha tenido éxito en nuestro estudio y puede servir como referencia para diseñar otras evaluaciones en entornos *online*.

6. Conclusiones y trabajo futuro

Se han cumplido los objetivos del trabajo dando respuesta a las preguntas de investigación planteadas. Los conceptos de sistemas operativos que han resultado más difíciles de comprender en el curso *online* estudiado han sido las *interrupciones* y los *semáforos*. Además, alrededor de estos conceptos se han identificado las dificultades específicas y sus posibles causas.

Se han explorado los efectos de la evaluación formativa diseñada para un curso de Sistemas Operativos impartido de forma *online*, y se han descubierto tres efectos: (1) una reflexión profunda sobre los conceptos de la asignatura, (2) una fase de incertidumbre en la resolución de las pruebas de evaluación y (3) una toma de conciencia de los progresos y las dificultades en el aprendizaje. Estos resultados corroboran los hallazgos de los trabajos existentes en el área.

Las contribuciones principales que realiza el presente trabajo de investigación se sitúan en las áreas de enseñanza y aprendizaje de sistemas operativos, evaluación formativa, metodologías cualitativas y educación *online* y se detallan a continuación:

- **Precisa dificultades de aprendizaje.** Una de las contribuciones de este trabajo es precisar las dificultades que los estudiantes encuentran en el aprendizaje de la asignatura Sistemas Operativos. Este tipo de estudio exploratorio no se ha realizado con anterioridad en esta área.
- **Propone un método para la mejora del aprendizaje de la asignatura Sistemas Operativos.** El trabajo presenta un método, la evaluación formativa, que puede mejorar el aprendizaje de la asignatura Sistemas Operativos. No existe ningún trabajo de enseñanza y aprendizaje de Sistemas Operativos que haya utilizado la evaluación formativa para mejorar el aprendizaje. Además, la presente investigación se distingue de

las existentes porque la evaluación diseñada tiene una base teórica y hace una caracterización del curso impartido de Sistemas Operativos mediante el análisis de sus objetivos de aprendizaje.

- **Proporciona un ejemplo de uso de la investigación cualitativa en *Computer Science Education*.** El trabajo constituye un ejemplo del uso de la investigación cualitativa, todavía escasamente utilizada y conocida en el área de *Computer Science Education*. La importancia de esta metodología radica en que permite aproximarse a comprender un escenario concreto de aprendizaje y explicar cómo y por qué suceden los fenómenos que en él se producen. En este sentido, la metodología cualitativa tiene un propósito de descubrimiento y constituye un valioso complemento a la investigación cuantitativa, más centrada en la confirmación de hipótesis.
- **Sugiere un conjunto de códigos que se pueden utilizar en investigaciones cualitativas que tengan los mismos objetivos que el presente estudio.** Uno de los resultados de la investigación es conjunto final de los códigos creados. Estos se pueden utilizar en futuras investigaciones cualitativas que pretendan obtener más información sobre uno o varios de los resultados obtenidos en este trabajo.
- **Proporciona un ejemplo de una evaluación formativa en un entorno *online*.** El diseño de evaluación formativa propuesto proporciona una forma concreta y eficaz de llevar a cabo las estrategias de la evaluación formativa en un entorno *online*, que puede servir como referencia para crear evaluaciones formativas de otras asignaturas del área informática.

En cuanto a los trabajos futuros, se han identificado diversos puntos en los que sería interesante profundizar:

- **Estudiar la comprensión de los conceptos *interrupción y semáforo*.** Uno de los posibles trabajos futuros consiste en realizar estudios de caso cualitativos para obtener más información sobre las dificultades descubiertas en este estudio. Se podrían utilizar las mismas pruebas de esta evaluación y/o crear otras pruebas de evaluación diferentes sobre la comprensión de los conceptos *interrupción y semáforo*. Estos estudios de caso pueden utilizar el sistema de codificación resultante de este estudio.

- **Descubrir dificultades de aprendizaje adicionales a las encontradas en este estudio.** Otro posible trabajo futuro consiste en realizar estudios de caso cualitativos para descubrir dificultades de aprendizaje adicionales a las encontradas en este estudio.
- **Poner en marcha soluciones para superar las dificultades descubiertas en esta investigación.** Esta línea de investigación consiste en proponer soluciones para superar las dificultades descubiertas en este estudio y comprobar sus efectos en el aprendizaje. Para este propósito se pueden proponer soluciones nuevas o utilizar soluciones ya presentadas en la literatura sobre la enseñanza y aprendizaje de sistemas operativos. Los estudios empíricos se pueden realizar con métodos cuantitativos, cualitativos o una combinación de ambos.
- **Estudiar la evaluación formativa en entornos online para otras asignaturas del área de informática.** Por último, otra posible línea de investigación consiste en utilizar las estrategias utilizadas en este estudio para llevar a cabo una evaluación formativa *online* y aplicarlas a otras asignaturas del área de informática.

Bibliografía

- Abimbola, I. O. (1988). The problem of terminology in the study of student conceptions in science. *Science Education*, 72(2), 175–184. doi:10.1002/sce.3730720206
- ACM/IEEE-CS Joint Interim Review Task Force. (2008). *Computer Science Curriculum 2008: An Interim Revision of CS 2001, Report from the Interim Review Task Force*.
- Airasian, P. W. (1994). Impact on testing and evaluation. In L. W. Anderson & L. A. Sosniak (Eds.), *Bloom's taxonomy : a forty-year retrospective* (pp. 82–102). Chicago: University of Chicago Press.
- Anderson, C. L., & Nguyen, M. (2005). A survey of contemporary instructional operating systems for use in undergraduate courses. *Journal of Computing Sciences in Colleges*, 21(1), 183–190.
- Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., ... Wittrock, M. C. (2000). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition*. Allyn & Bacon.
- Andrus, J., & Nieh, J. (2012). Teaching operating systems using android. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education - SIGCSE '12* (p. 613). New York, New York, USA: ACM Press. doi:10.1145/2157136.2157312
- Atkin, B., & Sirer, E. G. (2002). PortOS: An Educational Operating System for the Post-PC Environment. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education* (pp. 116–120). New York, NY, USA: ACM. doi:10.1145/563340.563384
- ATLAS.ti (Version 7) [Computer Software]. (2013). Berlin, Germany.
- Aviv, A. J., Mannino, V., Owlarn, T., Shannin, S., Xu, K., & Loo, B. T. (2012). Experiences in Teaching an Educational User-level Operating Systems Implementation Project. *SIGOPS Oper. Syst. Rev.*, 46(2), 80–86. doi:10.1145/2331576.2331588

- Ben-David Kolikant, Y. (2004). Learning concurrency: evolution of students' understanding of synchronization. *International Journal of Human-Computer Studies*, 60(2), 243–268. doi:10.1016/j.ijhcs.2003.10.005
- Black, M. D. (2009). Build an Operating System from Scratch: A Project for an Introductory Operating Systems Course. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (pp. 448–452). New York, NY, USA: ACM. doi:10.1145/1508865.1509022
- Bloom, B. S., Madaus, G. F., & Hastings, J. T. (1981). *Evaluation to improve learning*. McGraw-Hill.
- Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (1999). *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. Washington, DC: National Academy Press.
- Buendía, F., & Cano, J.-C. (2006). WebgeneOS: A Generative and Web-Based Learning Architecture to Teach Operating Systems in Undergraduate Courses. *IEEE Transactions on Education*, 49(4), 464–473. doi:10.1109/TE.2006.882368
- Buendía, F., Cano, J.-C., & Sahuquillo, J. (2005). Uso de simuladores y herramientas Web para la enseñanza de Sistemas Operativos. In *Actas del Simposio Nacional de Docencia en la Informática, SINDI2005 (AENUI)* (pp. 121–128).
- Bynum, B., & Camp, T. (1996). After You, Alfonse: A Mutual Exclusion Toolkit. In *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education* (pp. 170–174). New York, NY, USA: ACM. doi:10.1145/236452.236533
- Candela, S., García, C. M., Quesada, A., Santana, F. J., & Santos, J. M. (2007). *Fundamentos de sistemas operativos: teoría y ejercicios resueltos*. Editorial Paraninfo.
- Cañas, D. A. (1987). Graphos: A Graphic Operating System. In *Proceedings of the Eighteenth SIGCSE Technical Symposium on Computer Science Education* (pp. 201–205). New York, NY, USA: ACM. doi:10.1145/31820.31758
- Carretero, J., García, F., De Miguel, P., & Pérez, F. (2007). *Sistemas Operativos. Una visión aplicada*. McGraw-Hill.
- Casillas, A., & Iglesias, L. (2004). *Sistemas Operativos Ejercicios resueltos*. Pearson Education.

- Christopher, W. A., Procter, S. J., & Anderson, T. E. (1993). The Nachos Instructional Operating System. In *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings* (pp. 1–15). Berkeley, CA, USA: USENIX Association.
- Chung, G. K. W. K., Shel, T., & Kaiser, W. J. (2006). An Exploratory Study of a Novel Online Formative Assessment and Instructional Tool to Promote Students' Circuit Problem Solving. *The Journal of Technology, Learning and Assessment*, 5(6), 1–26.
- Comer, D. (2011). *Operating System Design: The Xinu Approach, Linksys Version*. Taylor & Francis.
- Crisp, V., & Ward, C. (2008). The development of a formative scenario-based computer assisted assessment tool in psychology for teachers: The PePCAA project. *Computers & Education*, 50(4), 1509–1526.
- Dall, C., & Nieh, J. (2014). Teaching Operating Systems Using Code Review. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 549–554). New York, NY, USA: ACM. doi:10.1145/2538862.2538894
- Desnoyers, P. J. (2011). Teaching Operating Systems As How Computers Work. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education* (pp. 281–286). New York, NY, USA: ACM. doi:10.1145/1953163.1953249
- Dewey, J. (2007). *How We Think*. Book Jungle.
- Donaldson, J. L. (1990). Operating Systems from Assembler to C. In *Proceedings of the Twenty-first SIGCSE Technical Symposium on Computer Science Education* (pp. 121–124). New York, NY, USA: ACM. doi:10.1145/323410.323428
- Dopper, S. M., & Sjoer, E. (2004). Implementing formative assessment in engineering education: the use of the online assessment system Etude. *European Journal of Engineering Education*, 29(2), 259–266. doi:10.1080/0304379032000157187
- Downey, A. B. (1999). Teaching experimental design in an operating systems class. In *SIGCSE '99 The proceedings of the thirtieth SIGCSE technical symposium on Computer science education* (Vol. 31, pp. 316–320). New York, New York, USA: ACM. doi:10.1145/384266.299796
- Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34(10), 906–911. doi:10.1037/0003-066X.34.10.906

- Flyvbjerg, B. (2006). Five Misunderstandings about Case-Study Research. *Qualitative Inquiry*, 12(2), 219–245.
- Fornaro, R. J., Garrard, K. P., & Uzzle, E. M. (1983). A Structured Approach to Teaching Operating Systems Principles Using a High Level Concurrent Programming Language. In *Proceedings of the Fourteenth SIGCSE Technical Symposium on Computer Science Education* (pp. 41–49). New York, NY, USA: ACM. doi:10.1145/800038.801010
- Garpis, A. (2013). Alg_OS-A web-based software tool to teach page replacement algorithms of operating systems to undergraduate students. *Computer Applications in Engineering Education*, 21(4), 581–585. doi:10.1002/cae.20502
- Gikandi, J. W., Morrow, D., & Davis, N. E. (2011). Online formative assessment in higher education: A review of the literature. *Computers & Education*, 57(4), 2333–2351.
- Goldberg, M. W. (1996). CALOS: An Experiment with Computer-aided Learning for Operating Systems. In *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education* (pp. 175–179). New York, NY, USA: ACM. doi:10.1145/236452.236534
- Goldberg, M. W. (1997a). CALOS: First Results from an Experiment in Computer-aided Learning for Operating Systems. In *Proceedings of the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education* (pp. 48–52). New York, NY, USA: ACM. doi:10.1145/268084.268101
- Goldberg, M. W. (1997b). WebCT and First Year: Student Reaction to and Use of a Web-based Resource in First Year Computer Science. In *Proceedings of the 2Nd Conference on Integrating Technology into Computer Science Education* (pp. 127–129). New York, NY, USA: ACM. doi:10.1145/268819.268856
- Goldweber, M., Barr, J., Camp, T., Grahm, J., & Hartley, S. (1999). A Comparison of Operating Systems Courseware. In *The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education* (pp. 348–349). New York, NY, USA: ACM. doi:10.1145/299649.299805
- Hayashi, T. (1983). An Operating Systems Programming Laboratory Course. In *Proceedings of the Fourteenth SIGCSE Technical Symposium on Computer Science Education* (pp. 31–35). New York, NY, USA: ACM. doi:10.1145/800038.801008
- Hays, J. H., Miller, L., Othmer, B., & Saeed, M. (1990). Simulation of Process and Resource Management in a Multiprogramming Operating System. In *Proceedings of the Twenty-first SIGCSE Technical Symposium on Computer Science Education* (pp. 125–128). New York, NY, USA: ACM. doi:10.1145/323410.323429

- Hess, R., & Paulson, P. (2010). Linux Kernel Projects for an Undergraduate Operating Systems Course. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 485–489). New York, NY, USA: ACM. doi:10.1145/1734263.1734428
- Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force concept inventory. *The Physics Teacher*, 30(3), 141. doi:10.1119/1.2343497
- Higginbotham, C. W., & Morelli, R. (1991). A System for Teaching Concurrent Programming. In *Proceedings of the Twenty-second SIGCSE Technical Symposium on Computer Science Education* (pp. 309–316). New York, NY, USA: ACM. doi:10.1145/107004.107071
- Hill, J. M. D., Ray, C. K., Blair, J. R. S., & Carver Jr., C. A. (2003). Puzzles and Games: Addressing Different Learning Styles in Teaching Operating Systems Concepts. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education* (pp. 182–186). New York, NY, USA: ACM. doi:10.1145/611892.611964
- Holland, D. A., Lim, A. T., & Seltzer, M. I. (2002). A new instructional operating system. *SIGCSE Bull*, 34, 111–115. doi:10.1145/563517.563383
- Holliday, M. A. (1997). System Calls and Interrupt Vectors in an Operating Systems Course. In *Proceedings of the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education* (pp. 53–57). New York, NY, USA: ACM. doi:10.1145/268084.268104
- Hovemeyer, D., Hollingsworth, J. K., & Bhattacharjee, B. (2004). Running on the Bare Metal with GeekOS. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (pp. 315–319). New York, NY, USA: ACM. doi:10.1145/971300.971411
- Hummel, J. (1989). Xinu/WU: An Improved PC-Xinu Clone? In *Proceedings of the Twentieth SIGCSE Technical Symposium on Computer Science Education* (pp. 226–230). New York, NY, USA: ACM. doi:10.1145/65293.71218
- Hwang, S. (2009). Blended Learning for Teaching Operating Systems with Windows. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education* (p. 380). New York, NY, USA: ACM. doi:10.1145/1562877.1563022
- Jones, D. (1996). RCOS.Java: An Animated Operating System for Computer Science Education. In *Proceedings of the 1st Conference on Integrating Technology into Computer Science Education* (p. 233–). New York, NY, USA: ACM. doi:10.1145/237466.237677

- Jones, D., N. A. (2002). A Constructivist-based Tool for Operating Systems Education. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002* (pp. 882–883). Denver, Colorado.
- Jong, B. S., Lai, C. H., Hsia, Y. T., Lin, T. W., & Lu, C. Y. (2013). Using Game-Based Cooperative Learning to Improve Learning Motivation: A Study of Online Game Use in an Operating Systems Course. *IEEE Transactions on Education*, 56(2), 183–190. doi:10.1109/TE.2012.2207959
- Kolikant, Y. B.-D., Ben-Ari, M., & Pollack, S. (2000). The anthropology of semaphores. In *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education - ITiCSE '00* (Vol. 32, pp. 21–24). New York, New York, USA: ACM Press. doi:10.1145/343048.343061
- Laadan, O., Nieh, J., & Viennot, N. (2010). Teaching Operating Systems Using Virtual Appliances and Distributed Version Control. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 480–484). New York, NY, USA: ACM. doi:10.1145/1734263.1734427
- Laadan, O., Nieh, J., & Viennot, N. (2011). Structured linux kernel projects for teaching operating systems concepts. In *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11* (p. 287). New York, New York, USA: ACM Press. doi:10.1145/1953163.1953250
- Lane, M. G., & Ghosal, A. k. (1989). MPX-PC: An Operating System Project for the PC. In *Proceedings of the Twentieth SIGCSE Technical Symposium on Computer Science Education* (pp. 231–235). New York, NY, USA: ACM. doi:10.1145/65293.71219
- Larraza-Mendiluze, E., & Garay-Vitoria, N. (2013a). The Learning Outcomes of the Exam Question in the Input/Output Topic in Computer Architecture. In *Proceedings of the 2013 Learning and Teaching in Computing and Engineering* (pp. 212–215). Washington, DC, USA: IEEE Computer Society. doi:10.1109/LaTiCE.2013.13
- Larraza-Mendiluze, E., & Garay-Vitoria, N. (2013b). Use of Concept Maps to Analyze Students' Understanding of the I/O Subsystem. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research* (pp. 67–76). New York, NY, USA: ACM. doi:10.1145/2526968.2526976
- Lawson, B., & Barnett, L. (2008). Using iPodLinux in an Introductory OS Course. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (pp. 182–186). New York, NY, USA: ACM. doi:10.1145/1352135.1352199
- Lincke, S. J. (2005). Creating Interest in Operating Systems via Active Learning. In *Proceedings Frontiers in Education 35th Annual Conference* (pp. S3C–7–S3C–10). IEEE. doi:10.1109/FIE.2005.1612262

- Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic Inquiry*. SAGE Publications.
- Liu, H., Chen, X., & Gong, Y. (2007). BabyOS: A Fresh Start. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (pp. 566–570). New York, NY, USA: ACM. doi:10.1145/1227310.1227499
- Lönnberg, J., & Berglund, A. (2007). Students' understandings of concurrent programming. In *Koli Calling '07 Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88* (pp. 77–86). Australian Computer Society, Inc.
- Madison, S. K. (1995, November 30). *A Study of College Students' Construct of Parameter Passing Implications for Instruction*. PhD thesis, U. Of Wisconsin.
- Maia, L. P., Machado, F. B., & Pacheco, A. C. (2005). A constructivist framework for operating systems education. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE '05* (Vol. 37, p. 218). New York, New York, USA: ACM Press. doi:10.1145/1067445.1067505
- Maia, L. P., Machado, F. B., & Pacheco Jr., A. C. (2005). A Constructivist Framework for Operating Systems Education: A Pedagogic Proposal Using the SOsim. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 218–222). New York, NY, USA: ACM. doi:10.1145/1067445.1067505
- Maia, L. P., & Pacheco Jr., A. C. (2003). A simulator supporting lectures on operating systems. In *Frontiers in Education 2003 FIE 2003 33rd Annual* (Vol. 2, pp. F2C–13–17 Vol.2).
- Manacero, A., & Lobato, R. S. (2013). Using a threaded framework to enable practical activities in Operating Systems courses. In *2013 IEEE Frontiers in Education Conference (FIE)* (pp. 1789–1794). IEEE. doi:10.1109/FIE.2013.6685146
- Mayer, R. E. (2002). Rote Versus Meaningful Learning. *Theory Into Practice*, 41(4), 226–232. doi:10.1207/s15430421tip4104_4
- Merrian, S. B. (1998). *Qualitative Research and Case Study Applications in Education* (Second Edi.). San Francisco: Jossey-Bass.
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. SAGE Publications.
- Moliner, M. (2003). *Diccionario de uso del español*. Madrid: Gredos.

- Morsiani, M., & Davoli, R. (1999). Learning Operating Systems Structure and Implementation Through the MPS Computer System Simulator. In *The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education* (pp. 63–67). New York, NY, USA: ACM. doi:10.1145/299649.299683
- Mustafa, B. (2011). Visualizing the modern operating system: simulation experiments supporting enhanced learning. In *SIGITE '11 Proceedings of the 2011 conference on Information technology education* (pp. 209–214).
- Mustafa, B. (2013). YASS: A system simulator for operating system and computer architecture teaching and learning. *European Journal of Science and Mathematics Education*, 1(1), 39–42.
- Nachmias, R., Moore, J. L., Dickson-Deane, C., & Galyen, K. (2011). e-Learning, online learning, and distance learning environments: Are they the same? *The Internet and Higher Education*, 14(2), 129–135.
- Nieh, J., & Vaill, C. (2005). Experiences Teaching Operating Systems Using Virtual Platforms and Linux. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (pp. 520–524). New York, NY, USA: ACM. doi:10.1145/1047344.1047508
- O’Gorman, J. (1998). Teaching Operating Systems. *SIGCSE Bull.*, 30(2), 61–63.
- Pamplona, S., Medinilla, N., & Flores, P. (2013). Exploring misconceptions of operating systems in an online course. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research - Koli Calling '13* (pp. 77–86). New York, New York, USA: ACM Press. doi:10.1145/2526968.2526977
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.
- Papert, S. (1993). *The Children’s Machine: Rethinking School in the Age of the Computer*. New York: BasicBooks.
- Pérez, F., Carretero, J., & García, F. (2003). *Problemas de sistemas operativos. De la base al diseño*. McGraw-Hill.
- Pérez, J. E., García, J., Muñoz, I., & Sierra, A. (2010). Active Learning and Generic Competences in an Operating Systems Course. *International Journal of Engineering Education*, 26(6), 1484–1492.
- Pérez, J. E., García Martín, J., & Muñoz Fernández, I. (2008). Cooperative learning in operating systems laboratory. *ACM SIGCSE Bulletin*, 40(3), 323. doi:10.1145/1597849.1384366

- Pérez-Dávila, A. (1995). O.S. bridge between academia and reality. In *Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education - SIGCSE '95* (Vol. 27, pp. 146–148). New York, New York, USA: ACM Press. doi:10.1145/199688.199760
- Pfaff, B., Romano, A., & Back, G. (2009). The pintos instructional operating system kernel. In *SIGCSE '09 Proceedings of the 40th ACM technical symposium on Computer science education* (Vol. 41, pp. 453–457). doi:10.1145/1539024.1509023
- Ramakrishnan, S., & Lancaster, A.-M. (1993). Operating Systems Projects: Linking Theory, Practice and Use. In *Proceedings of the Twenty-fourth SIGCSE Technical Symposium on Computer Science Education* (pp. 256–260). New York, NY, USA: ACM. doi:10.1145/169070.169483
- Real Academia de la Lengua Española. (2000). *Diccionario de la Real Academia de la Lengua española*. Madrid: Espasa-Calpe.
- Reek, M. M. (1990). An Undergraduate Operating Systems Lab Course. In *Proceedings of the Twenty-first SIGCSE Technical Symposium on Computer Science Education* (pp. 171–175). New York, NY, USA: ACM. doi:10.1145/323410.323442
- Robbins, S. (2000). Experimentation with Bounded Buffer Synchronization. In *Proceedings of the Thirty-first SIGCSE Technical Symposium on Computer Science Education* (pp. 330–334). New York, NY, USA: ACM. doi:10.1145/330908.331880
- Robbins, S. (2001). Starving philosophers. In *SIGCSE '01 Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education* (Vol. 33, pp. 317–321). New York, New York, USA: ACM Press. doi:10.1145/364447.364612
- Robbins, S. (2002). Exploration of process interaction in operating systems. In *SIGCSE '02 Proceedings of the 33rd SIGCSE technical symposium on Computer science education* (Vol. 34, pp. 351–355). ACM. doi:10.1145/563517.563476
- Robbins, S. (2004). A disk head scheduling simulator. In *SIGCSE '04 Proceedings of the 35th SIGCSE technical symposium on Computer science education* (Vol. 36, pp. 325–329). ACM. doi:10.1145/1028174.971413
- Robbins, S. (2005). An address translation simulator. In *SIGCSE '05 Proceedings of the 36th SIGCSE technical symposium on Computer science education* (Vol. 37, pp. 515–519). doi:10.1145/1047124.1047507
- Robbins, S. (2006). A UNIX Concurrent I/O Simulator. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (pp. 303–307). New York, NY, USA: ACM. doi:10.1145/1121341.1121435

- Robbins, S. (2007). A Java execution simulator. In *SIGCSE '07 Proceedings of the 38th SIGCSE technical symposium on Computer science education* (pp. 536–540). ACM. doi:10.1145/1227504.1227491
- Robbins, S., & Robbins, K. A. (1999). Empirical Exploration in Undergraduate Operating Systems. In *The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education* (pp. 311–315). New York, NY, USA: ACM. doi:10.1145/299649.299795
- Saldaña, J. (2012). *The Coding Manual for Qualitative Researchers*. SAGE Publications.
- Sangrà, A., Vlachopoulos, D., Cabrera, N., & Bravo, S. (2011). *Hacia una definición inclusiva del e-learning*. Barcelona: e-Learn Center UOC.
- Schipper, A., Dalang, G., Egli, J., Mattini, I., & Simon, R. (1984). A Paged - Operating - System Project. In *Proceedings of the Fifteenth SIGCSE Technical Symposium on Computer Science Education* (pp. 34–41). New York, NY, USA: ACM. doi:10.1145/800039.808618
- Scriven, M. (1967). The methodology of evaluation. In R. W. Tyler, R. M. Gagne, & M. Scriven (Eds.), *Perspectives of curriculum evaluation* (pp. 39–83). Rand McNally College Publishing Company.
- She, Y.-X., Lin, M.-H., Jong, B.-S., & Hsia, Y.-T. (2013). Using Growing Pet Game in Facebook to Enhance Students' Learning Motivation: In Operating System Course. In *2013 Learning and Teaching in Computing and Engineering* (pp. 224–228). IEEE. doi:10.1109/LaTiCE.2013.11
- Shub, C. M. (1983). A Project for a Course in Operating Systems. In *Proceedings of the Fourteenth SIGCSE Technical Symposium on Computer Science Education* (pp. 25–30). New York, NY, USA: ACM. doi:10.1145/800038.801007
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2011). *Operating System Concepts with JAVA*. John Wiley & Sons.
- Simon. (2011). Assignment and sequence. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research - Koli Calling '11* (p. 10). New York, New York, USA: ACM Press. doi:10.1145/2094131.2094134
- Smith, G. (2007). How Does Student Performance on Formative Assessments Relate to Learning Assessed by Exams? *Journal of College Science Teaching*, 36(7), 28 – 34.
- Sosniak, L. A. (1994). The Taxonomy, curriculum and their relations. In L. W. Anderson & L. A. Sosniak (Eds.), *Bloom's taxonomy : a forty-year retrospective* (pp. 103–125). Chicago: University of Chicago Press.

- Stallings, W. (2011). *Operating Systems: Internals and Design Principles (7th Edition)* (p. 816). Prentice Hall.
- Tanenbaum, A. S. (1987). A UNIX Clone with Source Code for Operating Systems Courses. *ACM SIGOPS Operating Systems Review*, 21(1), 20–29.
doi:10.1145/24592.24596
- Terrell, S. R., & Lin, Q. (2008). Preservice teachers' learning experiences of constructing e-portfolios online. *The Internet and Higher Education*, 11(3), 194–200.
- Underwood, J., Loveless, A., Pachler, N., Daly, C., Mor, Y., & Mellar, H. (2010). Formative e-assessment: Practitioner cases. *Computers & Education*, 54(3), 715–721.
- VanDeGrift, T., Bouvier, D., Chen, T.-Y., Lewandowski, G., McCartney, R., & Simon, B. (2010). Commonsense computing (episode 6). In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research - Koli Calling '10* (pp. 76–85). New York, New York, USA: ACM Press.
doi:10.1145/1930464.1930479
- Vonderwell, S. (2007). Asynchronous discussions and assessment in online learning. *Journal of Research on Technology in Education*, 39(3), 309 – 328.
- Wagner, T. D., & Ressler, E. K. (1997). A practical approach to reinforcing concepts in introductory operating systems. In *Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education - SIGCSE '97* (Vol. 29, pp. 44–47). New York, New York, USA: ACM Press. doi:10.1145/268084.268099
- Wang, C. (2009). Comprehensive Assessment of Student Collaboration in Electronic Portfolio Construction: An Evaluation Research. *TechTrends*, 53(1), 58.
- Watson, J., Murin, A., Vashaw, L., Gemin, B., & Rapp, C. (2012). *Keeping Pace with K-12 Online & Blended Learning: An Annual Review of Policy and Practice (2012)* (p. 164). Evergreen Education Group.
- Webb, K. C., & Taylor, C. (2014). Developing a Pre- and Post-course Concept Inventory to Gauge Operating Systems Learning. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 103–108). New York, NY, USA: ACM. doi:10.1145/2538862.2538886
- William, D., & Thompson, M. (2007). Integrating assessment with instruction: what will it take to make it work? In C. A. Dwyer (Ed.), *The future of assessment: shaping teaching and learning* (pp. 53–82). Mahwah, NJ: Lawrence Erlbaum Associates.
- Wittrock, M. C. (1989). *La investigación de la enseñanza, I. Enfoques, teorías y métodos*. Madrid: Ediciones PAIDOS.

- Wolfe, J. L. (1987). Operating System Projects on Two Simulated Machines. In *Proceedings of the Eighteenth SIGCSE Technical Symposium on Computer Science Education* (pp. 212–216). New York, NY, USA: ACM. doi:10.1145/31820.31760
- Ziegler, U. (1999). Discovery learning in introductory operating system courses. In *SIGCSE 99 The proceedings of the thirtieth SIGCSE technical symposium on Computer science education* (pp. 321–325). ACM Press. doi:10.1145/299649.299797

Anexos

Anexo A.

Análisis de los objetivos de aprendizaje de Sistemas Operativos

Objetivos de aprendizaje de Sistemas Operativos propuestos por ACM/IEEE	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
Visión general de Sistemas Operativos			
1. Explicar los objetivos y funciones de los sistemas operativos modernos.	Conceptual	Comprender	Interpretar
2. Describir cómo han evolucionado los sistemas operativos desde los sistemas por lotes hasta los sistemas multiusuario.	Factual	Recordar	Evocar
3. Analizar los compromisos que existen en el diseño de un sistema operativo.	Conceptual	Comprender	Comparar
4. Describir las funciones de un sistema operativo contemporáneo con respecto a la eficiencia y la capacidad para evolucionar.	Conceptual	Comprender	Interpretar
5. Discutir cómo los sistemas operativos de red, cliente-servidor y distribuidos difieren de los sistemas operativos para un único usuario.	Conceptual	Comprender	Comparar
6. Identificar las amenazas potenciales a los sistemas operativos y el diseño de las características de seguridad asociadas.	Factual	Recordar	Evocar
7. Describir cómo el software libre y el uso de Internet están influyendo en el diseño de un sistema operativo.	Conceptual	Comprender	Interpretar

Objetivos de aprendizaje de Sistemas Operativos propuestos por ACM/IEEE	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
Principios de Sistemas Operativos			
1. Explicar el concepto de nivel lógico.	Conceptual	Comprender	Interpretar
2. Explicar los beneficios de construir niveles abstractos de una forma jerárquica.	Conceptual	Comprender	Interpretar
3. Defender la necesidad de las API y el <i>middleware</i> .	Conceptual	Comprender	Interpretar
4. Describir cómo son usados los recursos del ordenador por el software de aplicación y gestionados por el sistema operativo.	Conceptual	Comprender	Interpretar
5. Contrastar el modo de usuario y el modo <i>kernel</i> de un sistema operativo.	Conceptual	Comprender	Comparar
6. Discutir las ventajas y los inconvenientes de usar procesamiento de interrupciones.	Conceptual	Comprender	Comparar
7. Comparar y contrastar las diversas formas de estructurar un sistema operativo tales como orientado a objetos, modular, microkernel y por capas.	Conceptual	Comprender	Comparar
8. Explicar el uso de una lista de dispositivos y una cola de <i>drivers</i> de E/S.	Conceptual	Comprender	Interpretar
Concurrencia			
1. Describir la necesidad de concurrencia en un sistema operativo.	Conceptual	Comprender	Inferir
2. Demostrar los potenciales problemas que pueden surgir en la ejecución de tareas concurrentes.	Conceptual	Comprender	Ejemplificar
3. Sintetizar el rango de mecanismos que pueden ser empleados en un sistema operativo para implementar la concurrencia. Describir los beneficios de cada uno de ellos.	Conceptual	Comprender	Interpretar
4. Explicar los diferentes estados en los que puede estar una tarea y las estructuras de datos necesarias para realizar la gestión de varias tareas.	Conceptual	Comprender	Interpretar
5. Sintetizar los diversos enfoques para resolver el problema de la exclusión mutua en un sistema operativo.	Conceptual	Comprender	Interpretar

Objetivos de aprendizaje de Sistemas Operativos propuestos por ACM/IEEE	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
6. Describir razones para usar interrupciones, planificación de procesos y cambio de contexto para soportar la concurrencia en un sistema operativo.	Conceptual	Comprender	Interpretar
7. Crear diagramas de estado y de transición para problemas simples.	Procedural	Aplicar	Implementar
8. Discutir la utilidad de estructuras de datos, tales como pilas y colas, para la gestión de la concurrencia.	Conceptual	Comprender	Interpretar
9. Explicar las condiciones que provocan un interbloqueo.	Conceptual	Comprender	Explaining
Planificación			
1. Comparar los algoritmos de planificación de procesos más comunes.	Conceptual	Comprender	Comparar
2. Describir las relaciones entre los algoritmos de planificación de procesos y los dominios de aplicación.	Conceptual	Comprender	Inferir
3. Discutir los tipos de planificación del procesador.	Conceptual	Comprender	Interpretar
4. Describir las diferencias entre procesos e hilos.	Conceptual	Comprender	Interpretar
5. Comparar los enfoques estáticos y dinámicos de la planificación de tiempo real.	Conceptual	Comprender	Comparar
6. Discutir la necesidad de la planificación preventiva.	Conceptual	Comprender	Inferir
7. Identificar las formas en la que la lógica de los algoritmos de planificación de procesos se pueden aplicar a otros dominios tales como: planificación de disco, planificación de E/S, planificación de red, planificación de proyectos y otros problemas relacionados con la computación.	Conceptual	Aplicar	Implementar
Gestión de memoria			
1. Explicar la jerarquía de memoria y los compromisos existentes entre rendimiento y coste.	Conceptual	Comprender	Comparar

Objetivos de aprendizaje de Sistemas Operativos propuestos por ACM/IEEE	Tipo de conocimiento	Categoría de procesos cognitivos	Proceso cognitivo
2. Explicar el concepto de memoria virtual y cómo funciona el software y el hardware necesario para implementarla.	Conceptual	Comprender	Interpretar
3. Sintetizar los principios de la memoria virtual referidos a la caché y la paginación.	Conceptual	Comprender	Interpretar
4. Evaluar el compromiso entre tamaño de memoria y velocidad del procesador.	Conceptual	Comprender	Comparar
5. Defender las diferentes formas de reservar memoria a las tareas citando las ventajas de cada una de ellas.	Conceptual	Comprender	Inferir
6. Describir la razón para el uso de la memoria caché.	Conceptual	Comprender	Interpretar
7. Discutir el concepto de <i>thrashing</i> , tanto en términos de las razones por las que ocurre como en términos de las técnicas usadas para reconocer y gestionar el problema.	Conceptual	Comprender	Inferir
Seguridad y Protección			
1. Defender la necesidad de protección y seguridad y las consideraciones éticas en el uso de una computadora.	Conceptual	Comprender	Interpretar
2. Sintetizar las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad.	Conceptual	Comprender	Interpretar
3. Explicar los mecanismos disponibles en un sistema operativo para controlar el acceso a los recursos.	Conceptual	Comprender	Interpretar
4. Llevar a cabo tareas simples de administración de acuerdo a una política de seguridad, por ejemplo la creación de cuentas, concesión de permisos, etc.	Procedural	Aplicar	Ejecutar

Anexo B.

Prueba de evaluación: Cuestionario de conocimientos previos

Este cuestionario tiene los siguientes objetivos:

- Que el profesor conozca los conocimientos previos sobre sistemas operativos de cada estudiante para poder adecuar la asignatura al grupo.
- Que el estudiante tome conciencia de sus conocimientos previos sobre la asignatura.

Para poder alcanzar estos dos objetivos, es necesario que tengas en cuenta los siguientes consejos:

- Las preguntas deben ser contestadas con tus propias palabras. No consultes Internet ni ningún otro tipo de documentación para realizar este cuestionario, no copies textualmente de ninguna fuente.
- Reflexiona cada respuesta antes de contestar.
- Se valorará el tiempo y reflexión dedicados a contestar el cuestionario, y no el número de respuestas “correctas”.

1. Nombre y apellidos

2. Año de nacimiento

3. ¿Tienes o has tenido un trabajo relacionado con la informática? Si es así, describe brevemente tus funciones.

4. Nombra los estudios oficiales y cursos más importantes relacionados con la informática que hayas realizado.

5. Imagina que no existieran los sistemas operativos. ¿Qué implicaciones crees que tendría este hecho?

6. ¿Para qué crees que sirve un sistema operativo?

7. Se dice que los conceptos de sistemas operativos son difíciles de comprender. ¿Tienes alguna idea de por qué sucede esto?
8. Cita los sistemas operativos que hayas usado, ya sea como programador o como usuario. Describe y evalúa brevemente la experiencia que has tenido con cada uno de ellos.
9. ¿Cómo compararías dos sistemas operativos? ¿Qué criterios usarías para decidir entre dos sistemas operativos?
10. ¿Sabes qué es un proceso en el contexto de sistemas operativos? En caso de que lo sepas, explícalo con tus palabras.
11. ¿Sabes qué es una API (interfaz de programación de aplicaciones)? En el caso de que lo sepas ¿Para qué sirve? Cita todas las API que conozcas.
12. ¿Sabes qué es una llamada al sistema en el contexto de sistemas operativos? En el caso de que lo sepas, explícalo con tus palabras.
13. ¿Sabes qué es una interrupción? En el caso de que lo sepas, explícalo con tus palabras.
14. ¿Sabes qué es la multitarea? En el caso de que lo sepas, explícalo con tus palabras.
15. ¿Sabes qué mecanismos usan los computadores para llevar a cabo la multitarea? En el caso de que lo sepas, explícalo con tus palabras.
16. ¿Sabes qué es la memoria caché? En el caso de que lo sepas, explícalo con tus palabras.
17. ¿Sabes qué es la memoria virtual? En el caso de que lo sepas, explícalo con tus palabras. ¿Sabes cómo funciona? Si es así, explícalo de forma breve.
18. ¿Sabes qué ventajas e inconvenientes presenta el uso de la memoria virtual? Si es así, explícalo de forma breve.
19. ¿Qué crees que es más importante en un computador, la velocidad del procesador o el tamaño de memoria RAM? Justifica brevemente tu respuesta.

Anexo C.

Prueba de evaluación: Cuestionario I

Pregunta 1

El controlador del dispositivo de E/S y la memoria intercambian datos directamente, sin la intervención del procesador, cuando se tiene:

- A. E/S programada
- B. E/S por interrupciones
- C. E/S mediante DMA

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Candela et al., 2007), página 53, pregunta 1.33.

Pregunta 2

El modo dual de operación consigue que el sistema informático sea:

- A. Más veloz
- B. Más fácil de usar
- C. Más seguro

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Candela et al., 2007), página 51, pregunta 1.19.

Pregunta 3

¿En qué sentido un sistema operativo es como un gobierno?

- A. Se crea un entorno en el que otros programas pueden hacer un trabajo útil.
- B. Realiza la mayoría de las funciones útiles por sí mismo.
- C. Su principal preocupación son las necesidades individuales.

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Silberschatz et al., 2011), *Instructor Companion Site, Test Bank, Chapter 1, Question 1.*

Pregunta 4

En un sistema monoprocesador, la multiprogramación incrementa el rendimiento del procesador:

- A. Aprovechando el tiempo de espera de la entrada/salida.
- B. Desactivando todas las interrupciones excepto las de más prioridad.
- C. Eliminando todos los ciclos improductivos del procesador.

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Stallings, 2011), *Testbank, Chapter 1, Multiple choice questions, Question 11.*

Pregunta 5

Las interrupciones sirven para mejorar la utilización del procesador

- A. Verdadero
- B. Falso

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Stallings, 2011), *Testbank, Chapter 1, TRUE/FALSE questions, Question 8.*

Pregunta 6

¿Por cuál de las siguientes situaciones puede un proceso pasar al estado “Listo”?

- A. Finalización de una operación de E/S
- B. Está esperando su turno de CPU
- C. Es un proceso recién creado
- D. Todas las anteriores

Justifica tu respuesta

Respuesta correcta: D

Fuente: adaptado de (Silberschatz et al., 2011), *Instructor Companion Site, Test Bank, Chapter 3, Question 7*.

Pregunta 7

¿Qué caracteriza más a un sistema operativo?

- A. El intérprete de comandos.
- B. Las llamadas al sistema.
- C. Los comandos de usuario.
- D. Su interfaz de programación de aplicaciones (API).

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (F. Pérez et al., 2003), página 8, ejercicio 1.4.

Pregunta 8

¿Qué código crees que necesita ejecutarse con las interrupciones inhibidas?

- A. Ninguno, ya que podrían perderse interrupciones.
- B. Todo el código del sistema operativo.
- C. Ciertas partes críticas del código del sistema operativo como por ejemplo el cambio de contexto.

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N1, pregunta 20.

Pregunta 9

Una llamada al sistema es

- A. Un servicio del sistema operativo
- B. Un programa del sistema operativo
- C. Un tipo especial de programa del sistema
- D. Cada uno de los comandos que se ejecuta en el intérprete de comandos

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Candela et al., 2007), página 52, pregunta 1.24.

Pregunta 10

Un cambio de diseño en la estructura o la semántica del bloque de control de proceso puede provocar cambios en varios módulos del sistema operativo.

- A. Verdadero
- B. Falso

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Stallings, 2011), *Testbank, Chapter 3, TRUE/FALSE questions, Question 3.*

Anexo D.

Prueba de evaluación: Cuestionario II

Pregunta 1

¿Cuál de las siguientes políticas de planificación puede producir inanición?

- A. Round Robin
- B. FCFS
- C. SJF
- D. Prioridades con envejecimiento

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N3, pregunta 19.

Pregunta 2

¿Cuál de las siguientes transiciones entre los estados de un proceso no se puede producir en un sistema con un algoritmo de planificación no preventivo?

- A. Bloqueado a Listo.
- B. En ejecución a Listo.
- C: En ejecución a Bloqueado.
- D. Listo a En ejecución.

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N4, pregunta 7.

Pregunta 3

¿Cuál es el número máximo de procesos que pueden quedarse bloqueados en un semáforo que se inicializó con un valor de 2?

- A. 1
- B. 2
- C. Un número teóricamente ilimitado

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N1, pregunta 21.

Pregunta 4

Cuando un proceso hace una operación *signal* sobre un semáforo:

- A. Siempre se incrementa en una unidad el valor del semáforo.
- B. Se incrementa en una unidad el valor del semáforo cuando este valor es positivo.
- C. Se incrementa en una unidad el valor del semáforo cuando este valor es negativo.

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Casillas & Iglesias, 2004), página 75, cuestión 3.2.2.

Pregunta 5

El algoritmo FCFS tiene un rendimiento mejor para procesos cortos que para procesos largos

- A. Verdadero
- B. Falso

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Stallings, 2011), *Testbank, Chapter 9, TRUE/FALSE questions, Question 8.*

Pregunta 6

El cambio de proceso (también llamado cambio de contexto):

- A. Lo realiza el planificador.
- B. Modifica la entrada en la tabla de procesos del proceso desalojado.
- C. Siempre se origina por una *interrupción* de reloj.
- D. Se produce siempre que el proceso abandona la cola de procesos en espera y pasa a la cola de procesos preparados.

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Casillas & Iglesias, 2004), página 9, cuestión 1.2.4.

Pregunta 7

¿Es posible ejecutar dos operaciones *wait* consecutivas sobre un semáforo?

- A. Sí
- B. Sí, siempre que sean procesos diferentes
- C. No

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Casillas & Iglesias, 2004), página 75, cuestión 3.2.7.

Pregunta 8

Los algoritmos de planificación afectan al tiempo total que el proceso pasa en:

- A. La cola de procesos en estado "Listo".
- B. La cola de procesos en estado "Bloqueado".
- C. La cola de procesos en estado "En ejecución".
- D. Cualquiera de las colas anteriores.

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Casillas & Iglesias, 2004), página 8, cuestión 1.2.1.

Pregunta 9

Sea un sistema con 3 procesos tal que cada uno de ellos necesita 3 unidades de un determinado tipo de recurso. ¿Cuántas unidades de ese recurso deben existir como mínimo para evitar el interbloqueo?

- A. 6
- B. 7
- C. 8
- D. 9

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N5, pregunta 21.

Pregunta 10

Una planificación de procesos SJF

- A. Minimiza el tiempo de espera máximo de un proceso
- B. Exige el conocimiento de la duración en CPU de los procesos
- C. Las otras dos opciones son ciertas

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Candela et al., 2007), página 87, pregunta 2.26.

Anexo E.

Prueba de evaluación: Cuestionario III

Pregunta 1

¿Cuál es el algoritmo de planificación del disco más adecuado para un sistema donde el tamaño medio de la cola de espera por el disco es de una petición?

- A. El SSTF por que obtiene el máximo rendimiento.
- B. El FIFO debido a que introduce menor sobrecarga.
- C. El SCAN de N pasos ya que elimina la inanición.
- D. El C-SCAN puesto que uniformiza el tiempo de espera de las peticiones.

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N3, pregunta 32.

Pregunta 2

Cuando se asignan bloques no continuos para almacenar un fichero:

- A. Un tamaño de bloque grande ofrece un mayor rendimiento y aprovechamiento efectivo del disco.
- B. Un tamaño de bloque pequeño ofrece un mayor rendimiento y aprovechamiento efectivo del disco.
- C. El fichero puede cambiar de tamaño sin tener que reubicar todos sus bloques de disco.

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Casillas & Iglesias, 2004), página 165, cuestión 6.2.1.

Pregunta 3

Cuando se utiliza la asignación enlazada como método de asignación de espacio para archivos:

- A. La implementación de la lista de bloques es más compleja que en la asignación indexada.
- B. El número de accesos a disco necesarios para recuperar datos del final de un archivo aumenta con el tamaño del archivo.
- C. El tamaño del bloque es múltiplo de dos.

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Casillas & Iglesias, 2004), página 165, cuestión 6.2.2.

Pregunta 4

Dado un sistema con memoria virtual, ¿cuánto puede avanzar como máximo la aguja del algoritmo de reemplazo de páginas del reloj durante la selección de una página?

- A. Una posición
- B. Una vuelta
- C. Media vuelta
- D. Dos vueltas

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N1, pregunta 23.

Pregunta 5

De las tres técnicas de entrada/salida: E/S programada, E/S por interrupciones y DMA:

- A. La más eficiente de las tres es E/S programada.
- B. La más eficiente de las tres es E/S por interrupciones.
- C. La más eficiente de las tres es DMA.

D. Dos de ellas son las más eficientes por igual: DMA y E/S dirigida.

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Casillas & Iglesias, 2004), página 141, cuestión 5.2.4.

Pregunta 6

En un sistema de memoria virtual paginado por demanda con capacidad para 4 marcos de página en memoria física y con una política de gestión LRU, ¿cuántos fallos de página se producirán para el siguiente patrón de referencia de páginas suponiendo el conjunto residente inicial vacío?: 1, 2, 3, 4, 5, 3, 4, 1, 6

A. 3

B. 5

C. 7

D. 1

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N1, pregunta 28.

Pregunta 7

En un sistema en el que las pistas de las peticiones de disco se distribuyen de forma uniforme, la política de planificación más conveniente de estas cuatro es:

A. C-SCAN

B. SSTF

C. SCAN

Justifica tu respuesta

Respuesta correcta: A

Fuente: adaptado de (Candela et al., 2007), página 267, pregunta 6.31.

Pregunta 8

En un sistema operativo con gestión de memoria virtual, el tamaño de los programas viene limitado por:

- A. El tamaño de la memoria primaria.
- B. El tamaño de la memoria secundaria.
- C. El tamaño de la memoria caché.

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Casillas & Iglesias, 2004), página 33, cuestión 2.2.3.

Pregunta 9

Indica en qué orden se accederá a los bloques 98, 183, 37, 122, 14, 65, 67 de un disco si aplicamos el algoritmo del ascensor con la variante C-SCAN en sentido ascendente. La cabeza está inicialmente en el bloque 53.

- A. 53, 65, 67, 37, 14, 98, 122, 183
- B. 53, 37, 14, 65, 67, 98, 122, 183
- C. 53, 65, 67, 98, 122, 183, 14, 37
- D. 53, 98, 183, 37, 122, 14, 65, 67

Justifica tu respuesta

Respuesta correcta: C

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N2, pregunta 15.

Pregunta 10

¿Para qué se introdujo la memoria virtual en los sistemas operativos?

- A. Para proteger a unos procesos de otros.
- B. Para ejecutar programas grandes en memorias pequeñas.
- C. Para permitir la reubicación dinámica de procesos.

Justifica tu respuesta

Respuesta correcta: B

Fuente: adaptado de (Carretero et al., 2007), recursos del instructor, test N3, pregunta 27.

Anexo F.

Prueba de evaluación: Ejercicios y problemas I

Ejercicio 1

Supongamos un sistema operativo con una planificación round-robin con un tiempo de $0,25 \mu\text{s}$ entre dos rodajas de tiempo (independientemente de si ocurre o no un cambio de proceso). Si se ejecutan tres trabajos de $6 \mu\text{s}$, $3 \mu\text{s}$ y $8 \mu\text{s}$.

Contesta las siguientes preguntas:

¿Cuál es el rendimiento si la rodaja de tiempo es de $1 \mu\text{s}$?

¿Cuál es el rendimiento si la rodaja de tiempo es de $0,5 \mu\text{s}$?

¿En cuál de los dos casos presentados anteriormente el rendimiento es mayor? ¿Cómo lo explicarías?

Fuente: adaptado de (Stallings, 2011), página 124, ejercicio 2.4.

Ejercicio 2

Monitoriza los procesos que hay en un determinado momento en un computador Linux. Utiliza para ello las dos opciones siguientes:

- Con el comando `ps`. Este comando muestra el estado de los procesos y admite una gran multitud de opciones. Puedes encontrar más información sobre ejecutando el comando `man ps`. Por ejemplo, puedes probar el comando `ps axu`.
- Con la herramienta Monitor (o similar) de la interfaz gráfica. Por ejemplo, en ubuntu 10.0.4, el monitor se encuentra en el menú Sistema, dentro de la opción Administración.

Estas herramientas te proporcionan información sobre el número de procesos en el sistema, su estado actual (ejecutando, en espera...), la cantidad de memoria que usan, etc. Puedes probar a ejecutar varios programas al mismo tiempo y ver qué impacto tiene su ejecución en la lista de procesos.

1. Describe brevemente la sesión de trabajo que has realizado para probar las dos opciones de monitorización. Es interesante que acompañes tu explicación con alguna captura de pantalla.

2. ¿Cuál de las dos opciones prefieres para la monitorización de procesos (monitor o ps)? ¿Por qué?

Fuente: Elaboración propia

Ejercicio 3

Monitoriza las llamadas al sistema que tienen lugar durante la ejecución de un determinado programa en un computador Linux. Utiliza para ello el siguiente comando:

a) Comando *strace*. Realiza la traza de ejecución de llamadas al sistema de un determinado programa. Es decir, visualiza el nombre de cada una de las llamadas al sistema que se produce en el programa. Puedes encontrar más información ejecutando el comando `man strace`.

Una forma sencilla de probar *strace* es con los propios comandos de Linux, por ejemplo podéis poner en el terminal (Menú Aplicaciones/Accesorios/Terminal) el comando *strace ls* y obtendréis una salida larga que comienza con algo parecido a esto:

```
execve("/bin/ls", ["ls"], [/* 36 vars */]) = 0
brk(0) = 0x805f000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0xb7f29000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=66530, ...}) = 0
mmap2(NULL, 66530, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f18000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/tls/i686/cmov/librt.so.1", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\220\31"... , 512) = 512
```

```
fstat64(3, {st_mode=S_IFREG|0644, st_size=30624, ...}) = 0
```

```
mmap2(NULL, 33360, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =  
0xb7f0f000
```

En el siguiente enlace puedes encontrar las llamadas al sistema de ubuntu para que puedas consultar el significado de alguna de ellas: [Lista de llamadas al sistema de ubuntu](#)

1. Describe brevemente la sesión de trabajo que has realizado para probar las dos opciones de monitorización. Es interesante que acompañes tu explicación con alguna captura de pantalla.
2. ¿Qué diferencia existe entre un comando y un llamada al sistema? ¿Qué relación existe entre estos dos conceptos?

Fuente: Elaboración propia

Ejercicio 4

Reflexiona sobre tu aprendizaje contestando a las siguientes preguntas:

Es necesario que este apartado se cumplimente después de haber terminado de realizar el resto de ejercicios de la actividad. Se calificará con un punto a los estudiantes que cumplimenten este apartado, no se evalúa su contenido, es un apartado de opinión en el que se agradecerá la sinceridad.

- a). ¿Qué opinas sobre lo que hemos realizado en la asignatura hasta el momento? ¿Ha sido fácil? ¿Difícil? ¿Interesante? ¿Aburrido? ¿Puedes explicar por qué?
- b). ¿Qué ejercicio de esta actividad te ha parecido más complicado? ¿Puedes explicar por qué?
- c) ¿Qué ejercicio de esta actividad te ha parecido más interesante? ¿Puedes explicar por qué?
- d) ¿Qué conceptos o aspectos de la asignatura te han parecido más complicados hasta el momento? ¿Sabrías explicar por qué?
- e) ¿Te han ayudado estos ejercicios a entender mejor algún aspecto de la asignatura? ¿Cuál? ¿Sabrías explicar por qué?
- f) ¿Has aprendido algo nuevo con esta actividad? ¿Qué? ¿Sabrías explicar cómo?

Fuente: Elaboración propia. Basada en las entrevistas realizadas en (Madison, 1995).

Anexo G.

Prueba de evaluación: Ejercicios y problemas II

Ejercicio 1

Una CPU ejecuta 50 instrucciones máquina por microsegundo (μs). Supongamos que un programa procesa un fichero de registros y que realiza las siguientes operaciones para cada registro:

Lectura del registro (Duración: $11 \mu\text{s}$)

Proceso de la información del registro (Duración: ejecución de 150 instrucciones máquina)

Escritura del registro (Duración: $11 \mu\text{s}$)

¿Cuál será el porcentaje de utilización de la CPU durante la ejecución de este programa? Justifica los pasos seguidos para realizar el cálculo.

Fuente: adaptado de (Stallings, 2011), página 124, ejercicio 2.3.

Ejercicio 2

Supongamos un sistema operativo multitarea en el que todos los procesos tienen las mismas características y se cumplen las siguientes condiciones:

En un periodo de computación de duración T , cada proceso pasa la primera mitad realizando operaciones de entrada/salida, y la otra mitad usando el procesador.

- Cada proceso se ejecuta durante un total de N periodos.
- La planificación es round robin.
- Las fases de entrada/salida de unos procesos pueden solaparse con las fases de CPU de otros procesos.

Realiza los siguientes apartados:

a) Calcula los siguientes parámetros para 1, 2 y 4 trabajos simultáneos en el sistema:

- Tiempo de estancia (tiempo que se tarda en completar un trabajo). Justifica los pasos seguidos para realizar el cálculo.

- Rendimiento (número medio de trabajos por unidad de tiempo). Justifica los pasos seguidos para realizar el cálculo.
- Utilización del procesador (porcentaje de tiempo que el procesador está activo). Justifica los pasos seguidos para realizar el cálculo.

b) Explica brevemente los motivos de las variaciones del tiempo de estancia, rendimiento y utilización al ejecutar 1, 2 o 4 trabajos en este sistema operativo. .

Nota: Debes tener en cuenta que los datos del tiempo de estancia y del rendimiento quedarán en función de N y T que son los datos proporcionados.

Fuente: adaptado de (Stallings, 2011), página 124, ejercicio 2.1.

Ejercicio 3

El siguiente conjunto de trabajos va a ser planificado siguiendo el algoritmo FCFS:

Trabajo	Duración ráfaga de CPU	Tiempo bloqueado a la espera de E/S
T0	7	4
T1	1	2
T2	1	2

Escenario 1

- Cada trabajo necesita para su finalización tres ráfagas de CPU. Entre cada dos ráfagas de CPU el proceso solicita una operación de E/S.
- Los trabajos se bloquean después de solicitar operaciones de E/S durante el tiempo especificado en la tabla.
- Las operaciones de E/S se realizan sobre un único dispositivo compartido por todos los trabajos y son atendidas secuencialmente en el mismo orden en el que fueron efectuadas por los procesos.

Escenario 2

Las condiciones uno y dos se mantienen, la condición 3 pasa a ser la siguiente:

- Las operaciones de E/S se realizan sobre tres dispositivos de E/S diferentes.

Escenario 3

Se mantienen las tres condiciones del escenario 1, pero se añade una nueva CPU. Se considera que las dos CPU son idénticas, que cualquier trabajo puede ser ejecutado indistintamente en cualquiera de ellas, y que ambas utilizan la misma cola de trabajos preparados (sólo existe una cola de trabajos preparados en el sistema).

Realiza los siguientes apartados:

1. Representa gráficamente la ejecución de los trabajos a lo largo del tiempo para los escenarios 1, 2 y 3. Puedes utilizar cualquiera de las dos propuestas de representación gráfica o una diferente. Explica y justifica la representación gráfica seleccionada. ¿Por qué esta representación te resulta más clara?
2. Calcula el porcentaje de utilización de la CPU para los escenarios 2 y 3. Justifica los pasos seguidos para realizar el cálculo.
3. Calcula el porcentaje de utilización del dispositivo para los escenarios 2 y 3. Justifica los pasos seguidos para realizar el cálculo.
4. Calcula el porcentaje de tiempo en el que se existe solapamiento (cuando están CPU y dispositivo ocupados de forma simultánea) para los escenarios 2 y 3. Justifica los pasos seguidos para realizar el cálculo.
5. Realiza una breve comparativa sobre las diferencias entre los 3 escenarios.

Fuente: adaptado de (Casillas & Iglesias, 2004), página 14, cuestión 1.4.2.

Ejercicio 4

Supongamos un sistema operativo que emplea una gestión de memoria paginada. Cada página tiene un tamaño de 2.048 bytes y la memoria física disponible para los procesos es de 8 Megabytes. En primer lugar, llega un proceso que necesita 31.566 bytes y, después, llega otro proceso que consume 18.432 bytes. Se pide:

- ¿Qué fragmentación interna provoca cada proceso?
- ¿Qué fragmentación externa provoca cada proceso?
- ¿Ha aclarado este ejercicio los conceptos que tenías sobre fragmentación interna y externa? ¿Ha aclarado algo sobre la gestión de memoria paginada? Explica los motivos de tus respuestas.

Fuente: adaptado de (Casillas & Iglesias, 2004), página 41, cuestión 2.4.5.

Anexo H. Prueba de evaluación: Prácticas con un simulador

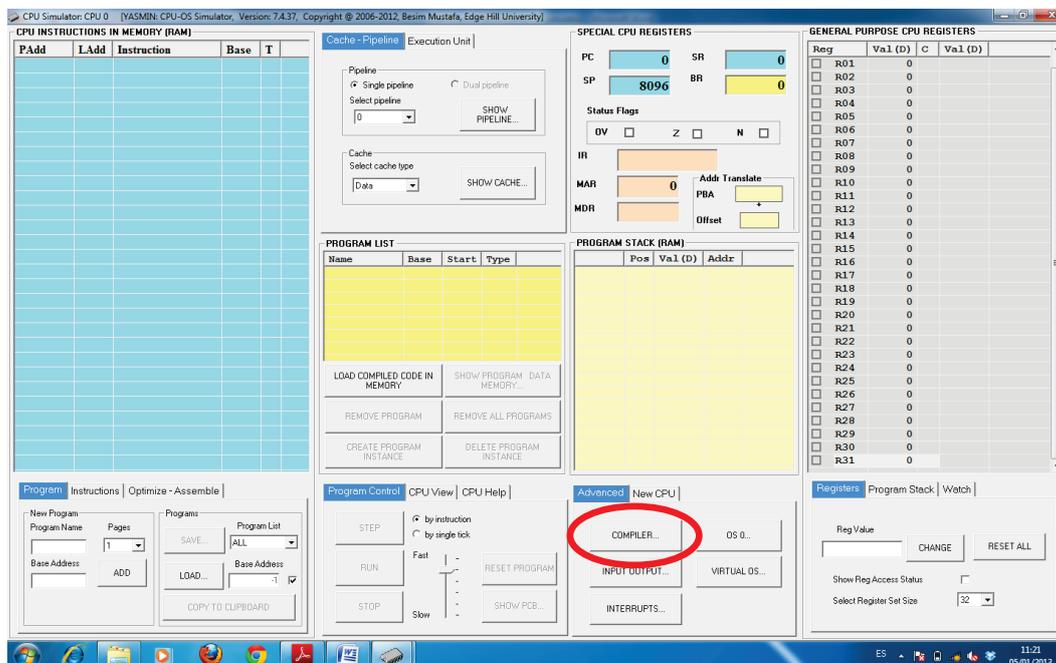
En esta actividad vamos a usar un simulador de sistemas operativos. El simulador hace posible observar el funcionamiento interno de un sistema operativo, hecho que es muy difícil o imposible en un sistema operativo real. Puedes descargar el simulador en la siguiente dirección web (funciona en Windows 7 y Windows XP): <http://www.teach-sim.com/>

Ejercicio 1 (práctica con el simulador de sistemas operativos)

Paso 1

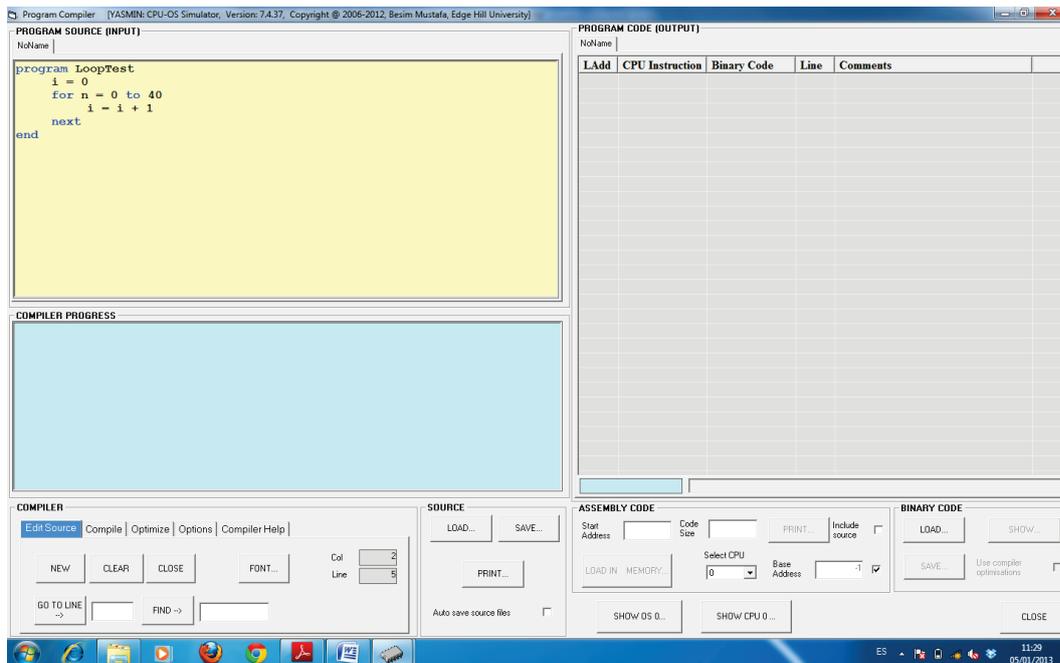
Introducir código fuente en el compilador y generar un programa ejecutable.

Para poder crear procesos, lo primero que debemos tener es un programa. En este paso vamos a crear y compilar un pequeño programa de prueba. Para ello, es necesario abrir la ventana del compilador haciendo clic en el botón **“COMPILER...”**

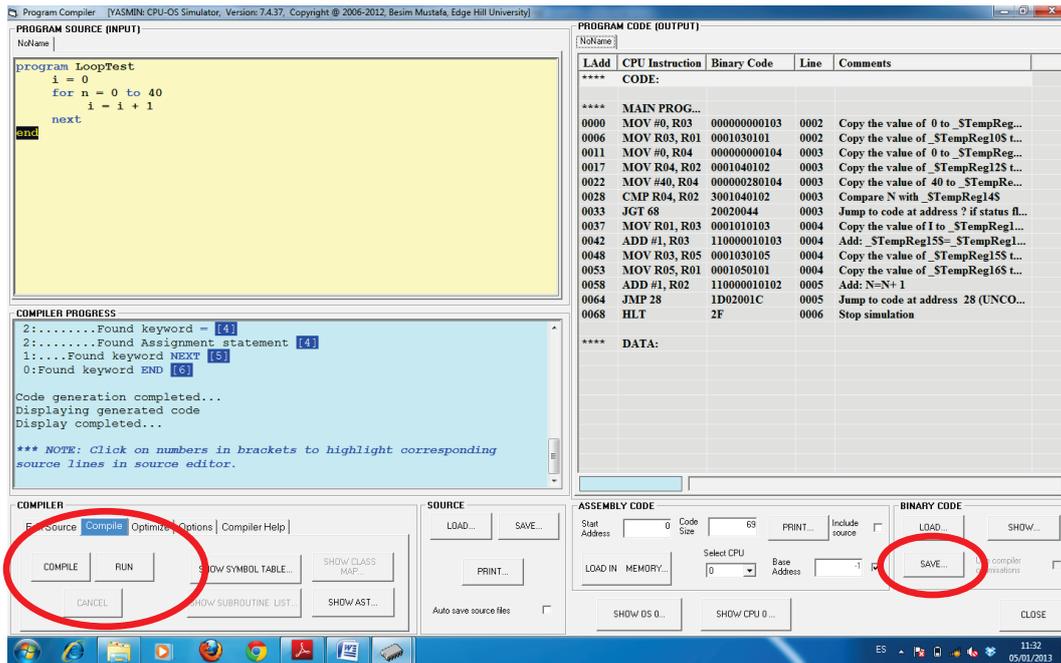


Después de pulsar el botón, aparece la pantalla del compilador. Introduce el siguiente código (lo puedes copiar y pegar) en la sección titulada **“PROGRAM SOURCE (INPUT)”**.

```
program LoopTest
  i = 0
  for n = 0 to 40
    i = i + 1
  next
end
```



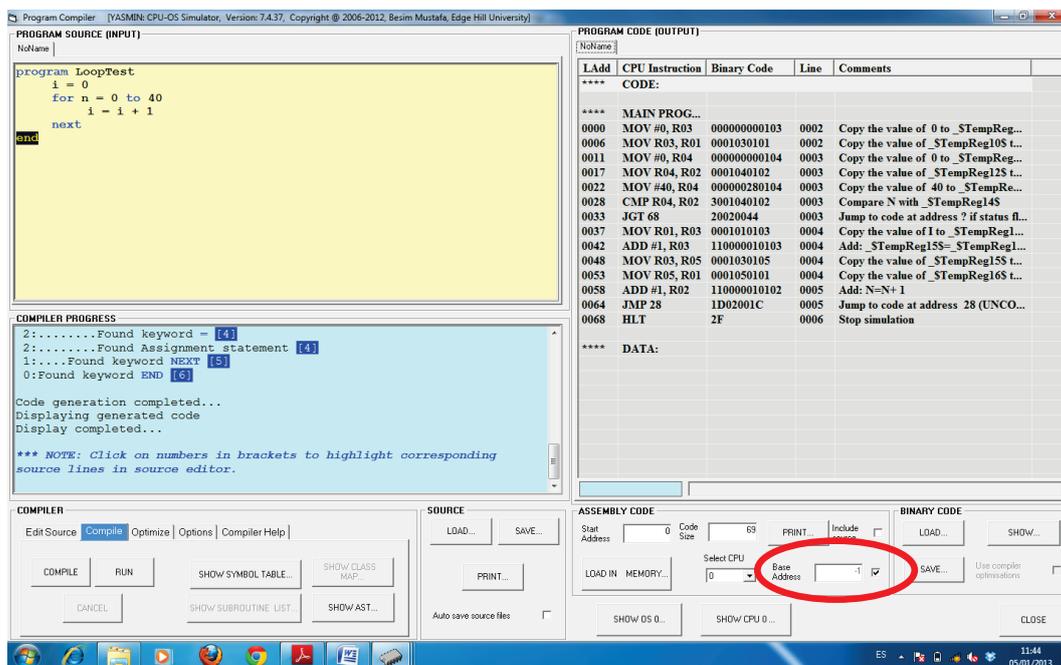
Para compilar el código, haz clic en la pestaña **“Compile”** y después en el botón **“COMPILE”**. El código fuente será traducido a código máquina y mostrado en la parte derecha de la pantalla, en la sección titulada **“PROGRAM CODE (OUTPUT)”**. Guarda el código generado con el botón **“SAVE...”**.



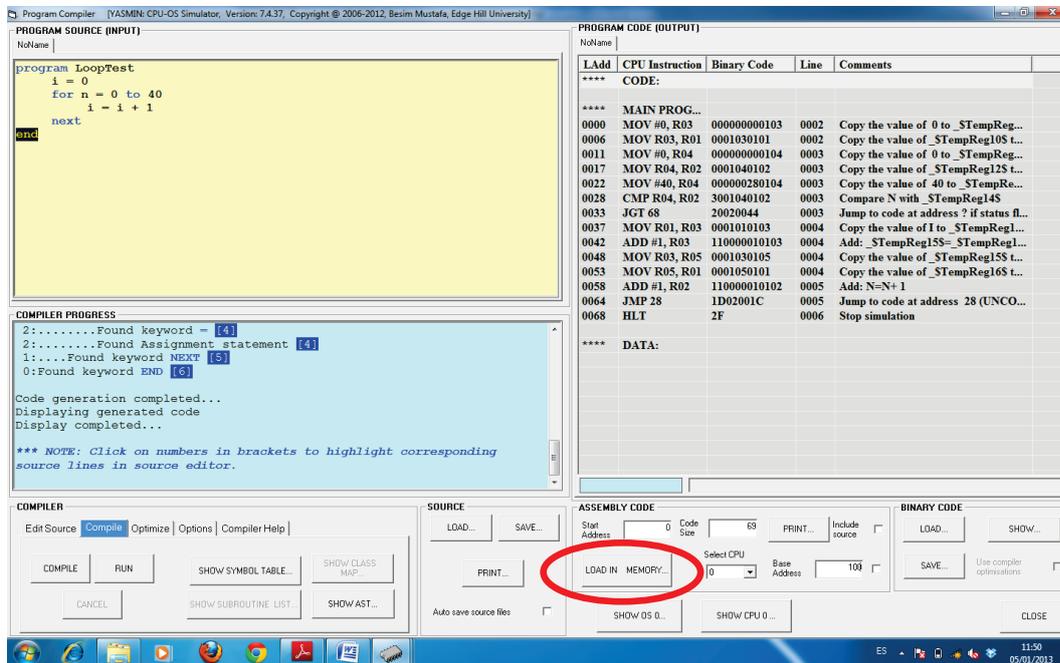
Paso 2

Cargar el programa en la memoria del simulador de CPU.

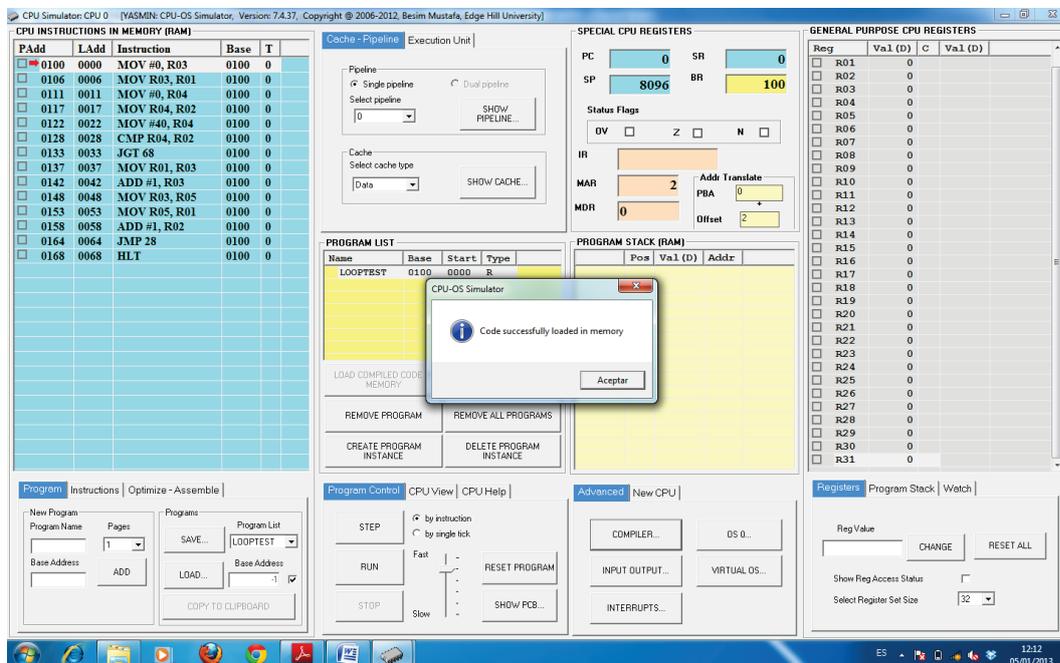
El código necesita cargarse en memoria para que pueda ser ejecutado por la CPU. Lo primero que vas a hacer es especificar una dirección base dentro de la sección "ASSEMBLY CODE". Para ello, desmarca la casilla de verificación que hay junto al campo "Base Address" y después introduce el valor 100 en dicho campo.



Una vez realizado este paso, carga el programa en memoria pulsando el botón “LOAD IN MEMORY”. Esta acción es equivalente a la carga del código del programa en la memoria RAM en los computadores reales.



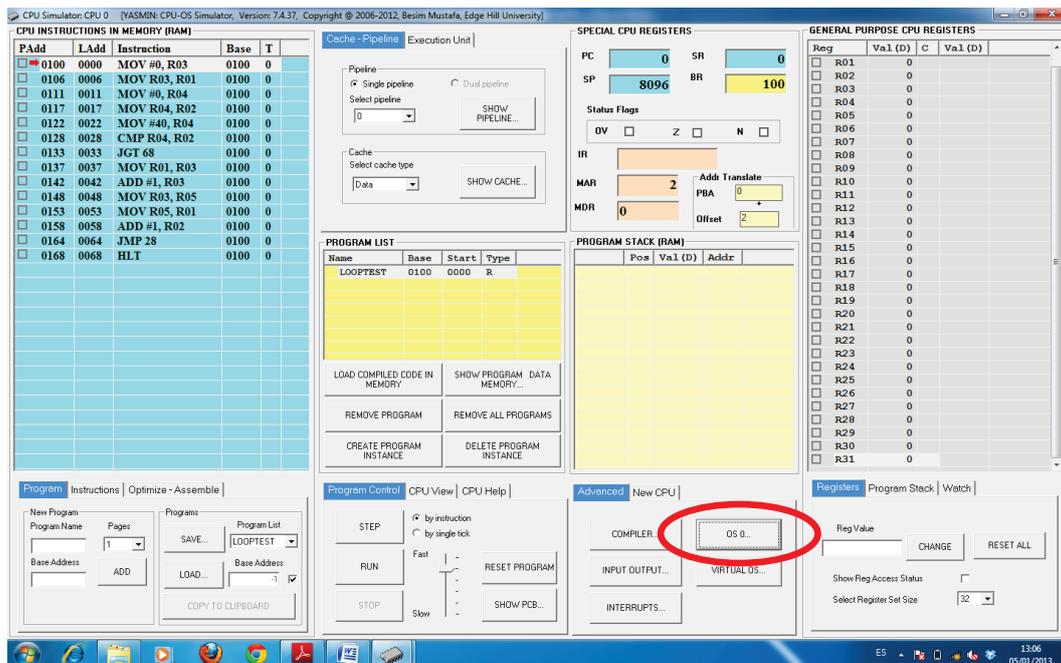
La carga del programa en memoria conduce a la ventana del simulador de CPU.



Paso 3

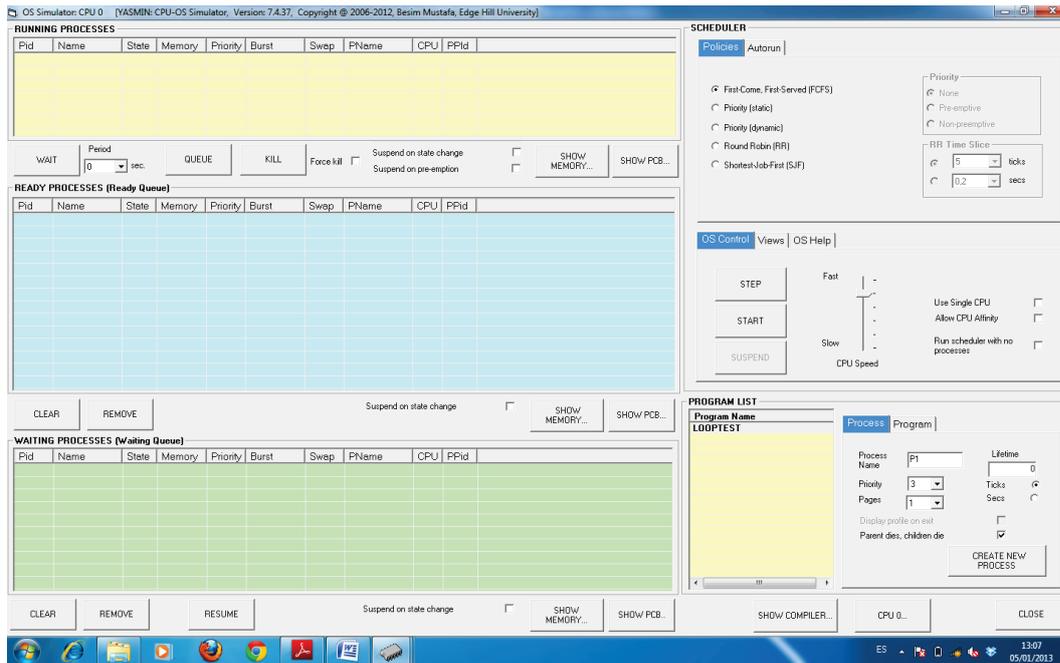
Crear procesos en el simulador del sistema operativo

En este paso vas a utilizar el simulador del sistema operativo para ejecutar el código. Para entrar en el simulador, pulsa el botón **“OS..O”** desde la pantalla del simulador de CPU.

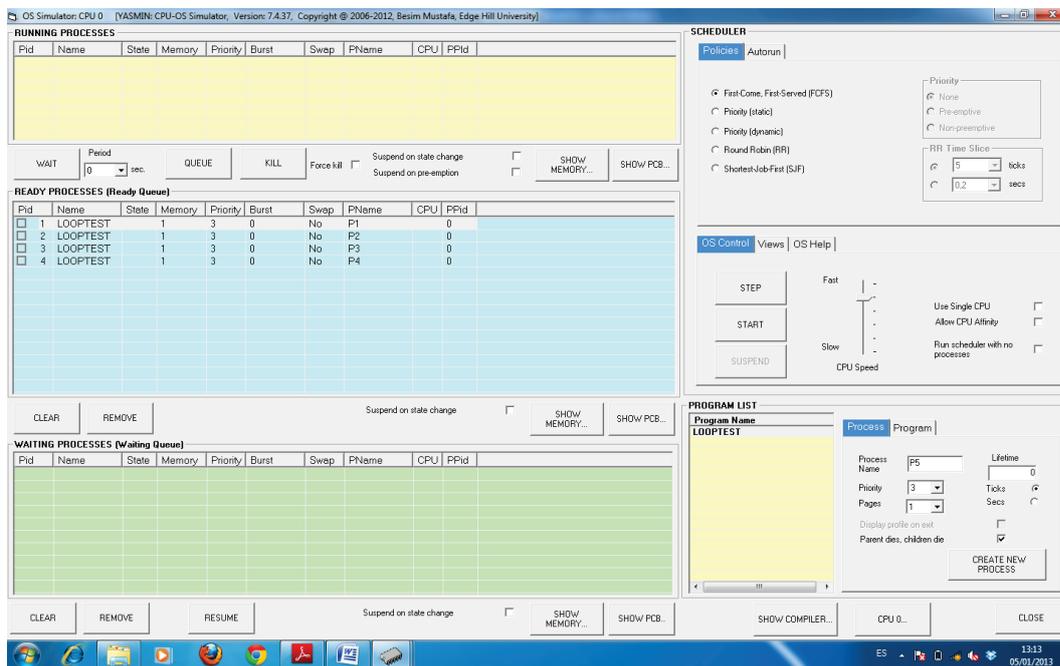


Desde la pantalla del simulador del sistema operativo tenemos acceso a la lista de programas que hay cargados. Si has seguido todos los pasos anteriores, verás el programa **“LOOPTEST”** dentro de la sección **“PROGRAM LIST”**.

Una vez que un programa está cargado se pueden crear tantos procesos del mismo como se desee. Para crear un proceso, sólo hay que pulsar sobre el botón **“CREATE NEW PROCESS”**. Crea cuatro procesos del programa LOOPTEST.



Los procesos recién creados aparecen en la sección “READY PROCESSES (Ready Queue)”.

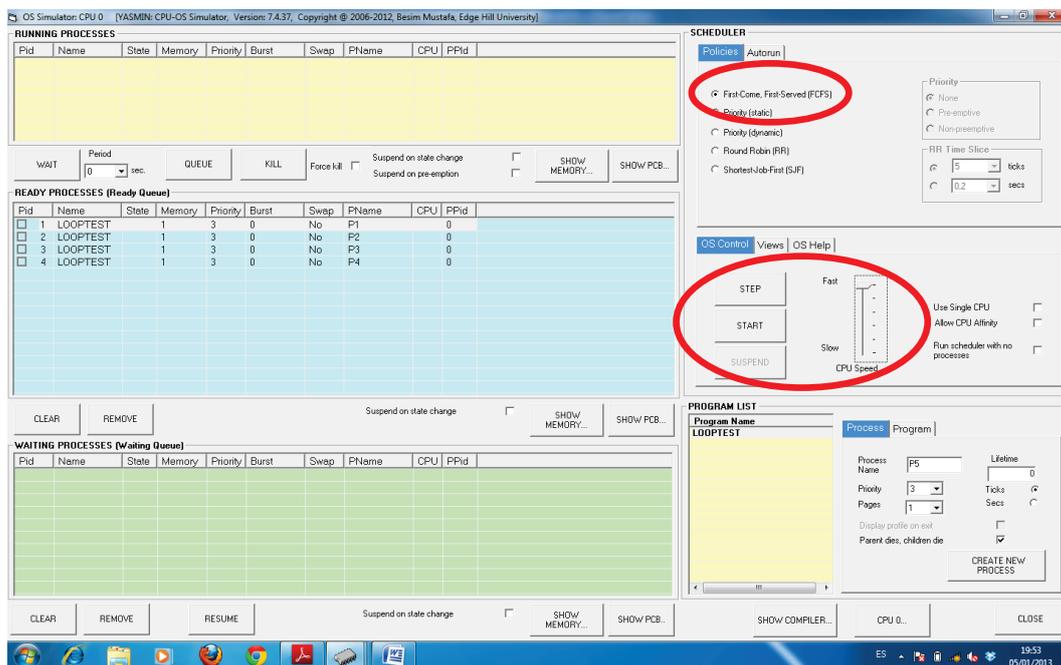


Paso 4

Seleccionar distintas políticas de planificación de procesos y ejecutar los procesos.

Escenario 1.

En primer lugar selecciona la política de planificación de procesos **“First-Come, First-Served FCFS”** en la pestaña **“Policies”** de la sección **“SCHEDULER”**. En este momento el sistema operativo está inactivo. Para activarlo, mueve el deslizador **“CPU Speed”** hasta la posición más rápida y pulsa el botón **“Start”**. Observa lo que ocurre hasta que finalicen los procesos.



Pregunta 1 (FCFS)

Explica lo que ocurre mientras se ejecutan los procesos. Concéntrate en las secciones **“RUNNING PROCESSES”** y **“READY PROCESSES”**.

Escenario 2

Asegúrate de que no hay ningún proceso en el simulador. Selecciona la política de planificación de procesos **“Priority (static)”** en la pestaña **“Policies”** de la sección **“SCHEDULER”**. Después selecciona la opción **“Non-preemptive”**.

Nota: La opción **non-preemptive** quiere decir no preventiva. Es decir, con esta opción una vez que la CPU se ha asignado a un proceso, el proceso se sigue ejecutando hasta que finalice o pase al estado **“Bloqueado”**.

A continuación, crea tres procesos con prioridades 3, 2 y 4 respectivamente. Para ello, usa la lista desplegable “Prioridad” antes del pulsar el botón **“CREATE NEW PROCESS”**. Desliza el selector de velocidad a medio camino entre el máximo y el mínimo y pulsa el botón **“Start”**. Mientras el primer proceso se está ejecutando crea un proceso nuevo con prioridad 1.

Pregunta 2 (Priority static non-preemptive)

Explica lo que ocurre mientras se ejecutan estos cuatro procesos con esta política de planificación.

Escenario 3

Asegúrate de que no hay ningún proceso en el simulador. Selecciona la política de planificación de procesos **“Priority (static)”** en la pestaña **“Policies”** de la sección **“SCHEDULER”**. Después selecciona la opción **“Pre-emptive”**.

Nota: La opción **pre-emptive** quiere decir preventiva. Es decir, con esta opción un proceso puede abandonar el procesador aunque no pase al estado bloqueado ni finalice.

A continuación, crea tres procesos con prioridades 3, 2 y 4 respectivamente. Para ello, usa la lista desplegable “Prioridad” antes del pulsar el botón **“CREATE NEW PROCESS”**. Desliza el selector de velocidad a medio camino entre el máximo y el mínimo y pulsa el botón **“Start”**. Mientras el primer proceso se está ejecutando crea un proceso nuevo con prioridad 1.

Pregunta 3 (Priority static pre-emptive)

Describe lo que observas mientras se ejecutan estos cuatro procesos con esta política de planificación. ¿Cuál es la diferencia con el escenario 2?

Nota: Si en algún momento no quieres esperar a la finalización de los procesos puedes usar el botón **“KILL”** para matar los procesos uno a uno.

Escenario 4

Selecciona la política de planificación de procesos **“Round Robin (RR)”** en la pestaña **“Policies”** de la sección **“SCHEDULER”**. Después selecciona la opción **“Non-preemptive”**. Crea cuatro procesos y pulsa el botón **“Start”**.

Pregunta 4 (Round Robin non-preemptive)

Explica lo que ocurre mientras se ejecutan estos cuatro procesos con esta política de planificación.

Escenario 5

Para este escenario, vas a crear un nuevo programa con un bucle infinito. Para crearlo ve a la ventana del compilador pulsando el botón **“SHOW COMPILER”**. En la sección **“COMPILER”** haz clic en la pestaña **“Edit Source”** y después en el botón **“NEW”**. Compila el programa, grábalo y cárgalo usando 200 como **“Base Address”** en esta ocasión.

```
program LoopForeverTest1
  n = 0
  while true
    n = n + 1
  wend
end
```

Ve la ventana del simulador del sistema operativo pulsando el botón **“OS O...”**.

En este momento verás una entrada adicional en la sección **“PROGRAM LIST”** con el nombre LOOPFOREVERTEST1. Selecciona esta entrada haciendo clic en ella. Ahora vas a crear procesos para este programa, pero en esta ocasión asignarás un tiempo de vida (en segundos) a cada uno para que la duración de los procesos sea distinta. Usando el botón de radio para seleccionar segundos y el cuadro de texto **“Lifetime”** para crear tres procesos con los siguientes tiempos de vida: 10 segundos, 32 segundos y 6 segundos. A continuación, en la pestaña **“Policies”** de la sección **“SCHEDULER”** selecciona Round Robin como algoritmo de planificación y 4 segundos como rodaja de tiempo. Pulsa el botón **“Start”** y observa lo que sucede durante la ejecución de los procesos.

Cuando hayan finalizado los procesos, haz clic en la pestaña **“Views”** de la sección **“SCHEDULER”** y pulsa sobre el botón **“VIEW LOG...”**. Comprueba el orden de ejecución de los procesos y las entradas más relevantes del log.

Pregunta 5 (Round Robin con una rodaja de tiempo de 4 segundos)

Explica lo que ha ocurrido durante la ejecución de los procesos considerando la información más relevante del log.

Crea otros escenarios

En este momento conoces el funcionamiento básico del simulador en cuanto a la planificación de procesos. Ahora es el momento de usar el simulador de forma libre para poder visualizar distintos ejemplos de planificación de procesos.

Pregunta 6

Piensa en **dos** ejemplos concretos de planificación de procesos distintos a los que hemos realizado antes. Ponlos en práctica con el simulador y comprueba si lo que observas coincide con lo que has estudiado en el manual. Describe el proceso que has seguido de una forma similar a las explicaciones que has encontrado aquí y explica los resultados obtenidos.

Fuente: adaptado de (Mustafa, 2013), tutorial descargado de <http://www.teach-sim.com/>

Ejercicio 2

Reflexiona sobre tu aprendizaje contestando a las siguientes preguntas. No se evalúa el contenido, sino el esfuerzo realizado para cumplimentarlas. Es un apartado de opinión en el que se agradecerá la sinceridad.

Sobre el ejercicio con el simulador

- ¿Qué dificultades has encontrado al realizar este ejercicio?
- ¿Has aprendido algo nuevo? ¿Qué? ¿Sabrías explicar cómo lo has aprendido?
- ¿Te ha resultado útil para comprender mejor los algoritmos de planificación de procesos? Justifica tu respuesta.
- ¿Encuentras el uso del simulador sencillo o complicado? Justifica tu respuesta.

- ¿Crees que el uso del simulador es más efectivo para aprender que los libros de texto? Justifica tu respuesta.

Sobre las actividades del curso

- Uno de los objetivos del curso es la comprensión del funcionamiento de los algoritmos de planificación de procesos en distintas situaciones. Para ello hemos utilizado dos aproximaciones distintas (los problemas de las actividades AEC1 y AEC2 y la práctica con el simulador en la AEC3).
- ¿Qué ventajas e inconvenientes tiene para tí cada uno de los métodos (problemas y simulador)? ¿Cuál prefieres? ¿Por qué?
- ¿Qué opinas sobre los controles de la asignatura? ¿Te han parecido difíciles? ¿Fáciles? ¿Interesantes? ¿Aburridos? Explica los motivos de tu respuesta.
- ¿Crees que los controles te han ayudado a comprender mejor la asignatura? Justifica tu respuesta.

Sobre el material didáctico

- ¿Has utilizado algún otro material didáctico además del manual de la asignatura? Si es así, enumera los materiales utilizados y explica por qué los has usado.

Sobre la asignatura en general

- ¿Qué opinas sobre la asignatura hasta el momento? ¿Ha sido fácil? ¿Difícil? ¿Interesante? ¿Aburrida? ¿Puedes explicar por qué?

Fuente: Elaboración propia

Anexo I. Consentimiento informado

Estimado estudiante:

Como profesora del Grado en Ingeniería Informática de la Universidad a Distancia de Madrid, estoy llevando a cabo un estudio sobre la comprensión de los conceptos de la asignatura Sistemas Operativos. El objetivo de la investigación es mejorar el aprendizaje de los estudiantes, adaptando los materiales didácticos y las actividades instruccionales de la asignatura.

Te agradecería mucho tu participación en el estudio, que sin duda será muy valiosa. Para ello, sólo necesito tu permiso para analizar las distintas pruebas de evaluación de las asignatura Sistemas Operativos que has realizado durante el primer semestre del curso 2012-2013 (controles, actividades de aprendizaje, actividades de evaluación continua y examen final presencial).

Te garantizo que se tomarán las medidas oportunas para asegurar el anonimato y la confidencialidad de los datos recogidos. En ningún caso se asociará ningún nombre propio con los datos. Si se publica alguna cita textual de un estudiante, se divulgará usando descripciones generales (un estudiante dijo: "...") de manera que se asegure el anonimato y la confidencialidad. Los resultados de este estudio estarán disponibles para la Universidad a Distancia de Madrid, UDIMA y otras publicaciones interesadas en la enseñanza y el aprendizaje de la Informática.

La participación en el estudio es totalmente voluntaria. La decisión de no participar no implicará ningún perjuicio por parte de la Universidad a Distancia de Madrid, UDIMA.

Una vez que el estudio esté terminado, estaré encantada de compartir los resultados contigo. Para cualquier aclaración adicional, puedes contactar conmigo de forma telefónica (91 856 16 94) o a través de la siguiente dirección de correo electrónico: sonia.pamplona@udima.es.

CONSENTIMIENTO INFORMADO

Yo _____ con DNI _____ comprendo los objetivos del estudio y estoy de acuerdo en participar en las condiciones detalladas por este documento. Entiendo que la participación en este estudio es totalmente voluntaria.

Fecha:

Fecha:

Firma del participante:

Firma de la investigadora:

Nombre del participante:

Nombre de la investigadora:

Sonia Pamplona Roche

Anexo J.

Códigos acerca de las dificultades de aprendizaje del concepto *interrupción*

En este anexo está compuesto por los informes de ATLAS.ti relativos a los códigos sobre el aprendizaje del concepto interrupción y contiene la siguiente información:

- Nombre, definición y número de citas de cada uno de los códigos.
- Citas asociadas al código y comentarios realizados durante el proceso de análisis.

El sistema de codificación presentado consta de 6 códigos relacionados con los significados y 7 códigos relacionados con las dificultades de aprendizaje acerca del concepto interrupción. A continuación se enumeran los distintos códigos creados para esta investigación. Cada uno de los códigos tiene un prefijo que permite organizar los códigos con el programa de análisis cualitativo ATLAS.ti.

Códigos relacionados con los significados del término *interrupción*

- SIG 1 Acción de interrumpir
- SIG 2 Efecto de interrumpir
- SIG 3 Algo que interrumpe
- SIG 4 Intervalo durante el cual algo es interrumpido
- SIG 5 Señal
- SIG 6 Característica

Códigos relacionados con las dificultades de aprendizaje acerca del concepto *interrupción*

- DIF 1 Definición incompleta de interrupción
- DIF 2 Origen de las interrupciones
- DIF 3 Las interrupciones no sirven para mejorar la utilización del procesador
- DIF 4 No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas
- DIF 5 El cambio de proceso siempre se origina por una interrupción de reloj
- DIF 6 Una interrupción sirve para realizar un cambio de contexto
- DIF 7 Siempre que hay una interrupción de reloj, se realiza un cambio de contexto

Código: SIG 1 Acción de interrumpir

Definición: Este código se asigna cuando la respuesta a la prueba de evaluación permite deducir que el significado que el estudiante ha otorgado al término interrupción es: *acción de interrumpir*.

Número de citas: 0

Código: SIG 2 Efecto de interrumpir

Definición: Este código se asigna cuando la respuesta a la prueba de evaluación permite deducir que el significado que el estudiante ha otorgado al término interrupción es: *efecto de interrumpir*.

Número de citas: 11

E07: "La verdad es que aún no conozco bien la diferenciación conceptual entre interrupción e interbloqueo, pero vagamente podría barruntar que son detenciones que provoca el SO a la ejecución de un proceso cuando otro proceso, al que se le da prioridad o turno, ejecute su propio código por o bien un tiempo determinado o bien hasta que finalice si es que ese proceso era más importante que el anterior. De ahí que haya las que veníamos a tratar en Tecnología y Est. de Computadores como "peticiones de interrupción" donde un proceso, al solicitar auxilio o servicios del SO demanda que otro proceso deponga su actividad y le ceda el procesador para hacer su labor. Como ejemplo, creo que hay una interrupción cada vez que escribo en el teclado o mueva (aunque si quiera sea un milímetro) mi ratón ya que estos le dicen al procesador que, en tiempo real, se vea el resultado en pantalla y, si algo estaba usando el procesador debe digamos "echarse a un lado"."

E10: "Creo que consiste en parar la ejecución de un programa, para resolver un evento."

E01: "Es un evento generado por una llamada al sistema."

E04: "Creemos que la respuesta es sí, pero debe ser matizada, en tanto que existen interrupciones que son externas al propio procesador. De todos modos, atendiendo al funcionamiento interno, consideramos que las interrupciones permiten una mejor utilización del procesador en tanto que permiten un cambio de proceso, es decir, que un proceso que está en ejecución pase a otro estado, como bloqueado (p.ej. esperando a que termine una operación de E/S) o listo (esperando que se le asigne tiempo para poder utilizar el procesador) Precisamente al "ordenar" la utilización del procesador por medio del citado mecanismo del cambio de estado de proceso, las interrupciones sirven para mejorar la utilización de aquél.

Comment:

Ocurre lo mismo que en los casos anteriores, el estudiante está pensando en el efecto de la interrupción, que un proceso se detenga y se pase a ejecutar otro y no en la causa.

E07: "Efectivamente, merced a las interrupciones un proceso puede tener el control del procesador como mínimo cuando le llegue su turno programado con anterioridad (si se emplea un algoritmo de planificación de tiempo compartido o de tipo round robin) y puede llevarse a término. De no existir las interrupciones, un proceso no se podría ejecutar hasta que finalizase el anterior o anteriores que estuviesen en la cola de tareas como ocurre con el procesamiento tipo batch. En los procesos interactivos, a diferencia de los de proceso por lotes, es necesario que el procesador atienda a interrupciones cuando el usuario u otro proceso demande la atención del procesador para otra tarea. Si no fuera porque las interrupciones invoquen al procesador, éste sólo podría trabajar en lotes y eso siempre que no hubiese errores pues, de nuevo, si tampoco hubiese interrupciones por errores, el ordenador no podría responder a ellos resolviéndolos saltando a otro proceso de subsanación del error o bien dándole el turno a otro proceso alternativo que, de otra manera no podría haber sido atendido. De esta guisa también, si el proceso que estuviese en ejecución estuviese a la espera de la ocurrencia de un evento para poder seguir trabajando, podría pasar que el procesador llegase a un punto muerto en que no se puede hacer otra cosa mientras ese proceso no termine y le ceda el testigo desaprovechando mucho tiempo. Metafóricamente es como si en una pista de tenis no pudiese jugar nadie durante una hora mientras estuviese reservada a nombre de un jugador por dos horas y éste llegase con el retraso de esa misma hora que se está desaprovechando.

Comment:

Aquí está muy claro que el estudiante no considera la interrupción como un mecanismo, sino como un efecto. Es un estudiante que ha estudiado mucho, pero no viene del mundo de la informática. Por este motivo, suponemos que el significado coloquial del término interrupción ha tenido mucha influencia en él.

E01: "Sirven para cambiar al modo privilegiado del sistema operativo."

Comment:

Esta respuesta parece estar relacionada con la proporcionada en el cuestionario inicial: "Una interrupción es un evento generado por una llamada al sistema". Una interrupción tiene como efecto que el computador pase al modo privilegiado, es cierto. Aquí el estudiante está pensando en otro tipo de interrupción. Y las interrupciones en las que hay que pensar para contestar a la pregunta son las interrupciones de E/S o las interrupciones de reloj.

22/05/2014 17:02:01

Para responder a esta pregunta hay que pensar en cómo se produce una interrupción, en el mecanismo de interrupciones del procesador. El estudiante está pensando en el efecto de la interrupción, más que en cómo se ha provocado.

E02: Las interrupciones mejora la utilización del procesador, mediante el uso de prioridades aplicadas en las distintas líneas de IRQ para ser procesadas antes por el procesador.

Comment:

Aquí el concepto "utilización" se ha interpretado de una forma distinta de como se pretendía en la pregunta.

Sin embargo, creo que también se está pensando en el efecto de interrumpir para contestar a la pregunta. Que se pueda interrumpir es bueno, porque supone que lo prioritario se pueda ejecutar antes.

E05: "Efectivamente, en función del tipo de interrupción que se este produciendo se puede mejorar el uso eficiente del procesador. Un ejemplo claro seria en el uso de la planificación ROUND-ROBIN, donde cuando a un proceso en ejecución, se le termina la rodaja de tiempo asignada, se produce una interrupción, que permite que otros procesos en espera puedan hacer uso de la CPU. Esas interrupciones facilitan que todos los procesos que requieren hacer uso de la CPU, estén en igualdad de condiciones para poder utilizarlo."

Comment:

En la respuesta está presente el efecto de la interrupción, que se pase a ejecutar otra cosa. Y la justificación de la utilización no está en el efecto, sino en la causa de la interrupción, en cómo se ha producido.

Digamos que si se entiende esto desde el punto de vista del efecto hay dos posibilidades:

- Que el estudiante diga que el rendimiento mejora porque se permite cambiar de proceso y eso es bueno, porque se permite ejecutar procesos prioritarios.
- Que el estudiante diga que las interrupciones son perjudiciales, porque suponen una pérdida de tiempo en el procesamiento.

E08: "Entiendo que es falso puesto que una interrupción es una suspensión temporal de un programa, no hace que se mejore el uso del procesador sino que lo interrumpe y si el grado de interrupción es alto podría provocar que incluso fuera más lenta la ejecución de procesos puesto que tendríamos al procesador dedicado a atender dichas interrupciones."

Comment:

Aquí vuelve a estar claro que se llama interrupción al efecto y no al mecanismo. Esto es un problema en esta pregunta.

E07: "Así mismo hay que descartar la opción "c" ya que no necesariamente se produce un cambio de contexto por una interrupción de reloj pues no todos los procesos se alternan por RR o por rodajas de tiempo que se consumen periódicamente sino que el planificador puede haber elegido otro algoritmo para establecer los turnos."

E01: "Puesto que una interrupción sirve, entre otras cosas, para realizar un cambio de contexto."

Código: SIG 3 Algo que interrumpe

Definición: Este código se asigna cuando la respuesta a la prueba de evaluación permite deducir que el significado que el estudiante ha otorgado al término interrupción es: *efecto de interrumpir*.

Número de citas: 1

E04: "Creo que una interrupción es una clase de llamada al sistema que permite interrumpir un proceso o el uso de un dispositivo."

Código: SIG 4 Intervalo durante el cual algo es interrumpido

Definición: Este código se asigna cuando la respuesta a la prueba de evaluación permite deducir que el significado que el estudiante ha otorgado al término interrupción es: *el intervalo durante el cual algo es interrumpido*.

Número de citas: 0

Código: SIG 5 Señal

Definición: Este código se asigna cuando la respuesta a la prueba de evaluación permite deducir que el significado que el estudiante ha otorgado al término interrupción es: *señal que inicia la parada de un computador para ejecutar otro.*

Número de citas: 6

E08: "Cuando un proceso se está ejecutando en el procesador y llega uno más prioritario el planificador (PLC) de turno envía una interrupción al procesador para que ésta atienda al nuevo proceso más prioritario."

E06: "Es una señal hardware. La forma que tiene el procesador para aceptar peticiones, interrumpir lo que esta haciendo y pasar a ejecutar algo o controlar cierto dispositivo o periférico. Luego volverá a ejecutar lo que estaba haciendo. Pueden ser de distinto tipo según quien las gestione (procesador, bus o periférico) o si toma el control total de la CPU o sólo rodajas de tiempo de esta."

E05: "Petición de atención que se realiza al sistema, para poder utilizar desde el que invoca, los recursos que tiene disponible."

E06: "E/S mediante DMA significa direct access memory, por tanto los dispositivos de entrada-salida pueden acceder directamente a la memoria sin pasar por el procesador, ni requerir interrumpir al procesador con interrupciones. Esto acelera el intercambio de información con la memoria y no ralentiza al procesador, pues mientras tanto sigue haciendo su trabajo, esto es PERMITE CONCURRENCIA entre el procesador y la entrada o salida. Las otras opciones, programada y por interrupción si requieren intervención del procesador, ya sea por que tiene el procesador un programa en ejecución para acceder al dispositivo de entrada-salida (NO HAY CONCURRENCIA entre procesador y entrada o salida), o por que una interrupción le avisa del requerimiento de entrada-salida y debe decidir si darle paso y cuando (SI PERMITE CONCURRENCIA entre procesador y entrada o salida)."

E01: "Mediante DMA, el controlador del dispositivo se encarga de transferir los datos entre periférico y memoria, porque en la programada, se necesita que se esté ejecutando un programa que use E/S, y por

interrupciones es el periférico el que avisa al procesador con una interrupción.”

E08: “En la E/S mediante DMA (Acceso Directo a Memoria) es el controlador del dispositivo el que se encarga directamente de transferir los datos entre el periférico y la memoria sin interrumpir al procesador. Esto permite que el procesador pueda dedicarse a ejecutar otro programa, mientras dure el intercambio. Una E/S programada no permite concurrencia y el procesador está dedicado a la ejecución del programa de E/S. En el caso de E/S por interrupciones es el propio dispositivo de E/S el que avisará al procesador, aunque también permite concurrencia como ya he comentado el propio dispositivo avisa al procesador a través de una interrupción.”

Código: SIG 6 Característica

Definición: Este código se asigna cuando la respuesta a la prueba de evaluación permite deducir que el significado que el estudiante ha otorgado al término interrupción es: *característica de un computador que permite la interrupción temporal de una actividad para realizar otra.*

Número de citas: 6

E06: "Es una señal hardware. La forma que tiene el procesador para aceptar peticiones, interrumpir lo que esta haciendo y pasar a ejecutar algo o controlar cierto dispositivo o periférico. Luego volverá a ejecutar lo que estaba haciendo. Pueden ser de distinto tipo según quien las gestione (procesador, bus o periférico) o si toma el control total de la CPU o sólo rodajas de tiempo de esta.

E03: "Es un mecanismo que permite que un programa sea interrumpido temporalmente para atender una llamada del procesador, para luego seguir con la tarea como si nada."

Memos:

Atender una llamada del procesador. Esto indica que las interrupciones proceden del procesador, que es el procesador el que interrumpe y eso no es así.

E02: "Un interrupción, es una llamada al sistema, el S.O tiene una serie de interrupciones numeradas y con prioridad, esta idea fue desarrollada para que el procesador pudiese atender a peticiones que entran por estas IRQ mientras está gestionando otras tareas, es decir, el procesador tras cada ciclo de ejecución comprueba que los valores en las entradas de IRQ si están o activas, en tal caso, atendería o asignaría ciclos de CPU a dicha interrupción, retardando el acceso a la aplicación que le tocara el turno de CPU. Por ejemplo las entradas de teclado, pasan por un IRQ siempre, se considera que el teclado debe tener una prioridad sobre el sistema."

Memos:

Su explicación no es muy clara, pero se refiere a las interrupciones como un medio para que el procesador atienda a peticiones mientras está desarrollando otras tareas.

E06: "Creo que la respuesta sería verdadero, pues si se produce una excepción síncrona (por el propio programa) o asíncronas (fallo de algún dispositivo), si no hubiese forma de avisar al procesador con los bits del registro de su estado, para dar paso a la interrupción, el procesador podría quedar inutilizable, o ejecutando procesos que no llevarían a ninguna parte (bucles, espera eterna etc). Además, los periféricos que se comunican por interrupciones hardware con el procesador lo hacen de forma más eficiente que si fuese por programa, consumiendo menos ciclos de reloj, permitiendo que otros procesos sigan en estado de ejecución mientras se atiende a las interrupciones. Las interrupciones permiten avisar al procesador de por ejemplo una entrada de un dispositivo, evitando así el tener que controlar periódicamente la entrada para ver si ya esta disponible el dato o no, haciendo más útil al procesador que delega esta función de control al propio dispositivo o al bus del sistema."

Comment:

Perfecto, este estudiante realiza un estudio de los distintos tipos de interrupciones, de sus causas, de cómo se han producido. Estudia la afirmación para cada uno de los casos.

E07: "Efectivamente, merced a las interrupciones un proceso puede tener el control del procesador como mínimo cuando le llegue su turno programado con anterioridad (si se emplea un algoritmo de planificación de tiempo compartido o de tipo round robin) y puede llevarse a término. De no existir las interrupciones, un proceso no se podría ejecutar hasta que finalizase el anterior o anteriores que estuviesen en la cola de tareas como ocurre con el procesamiento tipo batch. En los procesos interactivos, a diferencia de los de proceso por lotes, es necesario que el procesador atienda a interrupciones cuando el usuario u otro proceso demande la atención del procesador para otra tarea. Si no fuera porque las interrupciones invoquen al procesador, éste sólo podría trabajar en lotes y eso siempre que no hubiese errores pues, de nuevo, si tampoco hubiese interrupciones por errores, el ordenador no podría responder a ellos resolviéndolos saltando a otro proceso de subsanación del error o bien dándole el turno a otro proceso alternativo que, de otra manera no podría haber sido atendido. De esta guisa también, si el proceso que estuviese en ejecución estuviese a la espera de la ocurrencia de un evento

para poder seguir trabajando, podría pasar que el procesador llegase a un punto muerto en que no se puede hacer otra cosa mientras ese proceso no termine y le ceda el testigo desaprovechando mucho tiempo. Metafóricamente es como si en una pista de tenis no pudiese jugar nadie durante una hora mientras estuviese reservada a nombre de un jugador por dos horas y éste llegase con el retraso de esa misma hora que se está desaprovechando.

Comment:

Aquí está muy claro que el estudiante no considera la interrupción como un mecanismo, sino como un efecto. Es un estudiante que ha estudiado mucho, pero no viene del mundo de la informática. Por este motivo, suponemos que el significado coloquial del término interrupción ha tenido mucha influencia en él.

22/05/2014 17:54:19

OJO, que sí que analiza cada uno de los casos, hay varios conceptos efecto y acción o causa.

E10: "Se evita que un proceso siga ejecutándose mientras espera una entrada por parte del usuario, por ejemplo."

Código: DIF 1 Definición incompleta de interrupción

Definición: Este código se asigna cuando en la respuesta a la prueba de evaluación el estudiante proporciona una definición de interrupción parcial, que no está completa. Entiende por *interrupción* sólo uno de los posibles tipos de *interrupciones*.

Número de citas: 1

E01: "Es un evento generado por una llamada al sistema."

Código: DIF 2 Origen de las interrupciones

Definición: Este código se asigna cuando el estudiante afirma en su respuesta una procedencia de las interrupciones distinta a las concepciones aceptadas.

Número de citas: 2

E08: "Cuando un proceso se está ejecutando en el procesador y llega uno más prioritario el planificador (PLC) de turno envía una interrupción al procesador para que ésta atienda al nuevo proceso más prioritario."

E03: "Es un mecanismo que permite que un programa sea interrumpido temporalmente para atender una llamada del procesador, para luego seguir con la tarea como si nada."

Memos:

Atender una llamada del procesador. Esto indica que las interrupciones proceden del procesador, que es el procesador el que interrumpe y eso no es así.

Código: DIF 3 Las interrupciones no sirven para mejorar la utilización del procesador

Definición: Este código se asigna cuando el estudiante afirma en su respuesta que las interrupciones no sirven para mejorar la utilización del procesador.

Número de citas: 2

E01: "Sirven para cambiar al modo privilegiado del sistema operativo."

Comment:

Esta respuesta parece estar relacionada con la proporcionada en el cuestionario inicial: "Una interrupción es un evento generado por una llamada al sistema". Una interrupción tiene como efecto que el computador pase al modo privilegiado, es cierto. Aquí el estudiante está pensando en otro tipo de interrupción. Y las interrupciones en las que hay que pensar para contestar a la pregunta son las interrupciones de E/S o las interrupciones de reloj.

22/05/2014 17:02:01

Para responder a esta pregunta hay que pensar en cómo se produce una interrupción, en el mecanismo de interrupciones del procesador. Yo creo que el estudiante está pensando en el efecto de la interrupción, más que en cómo se ha provocado.

E08: "Entiendo que es falso puesto que una interrupción es una suspensión temporal de un programa, no hace que se mejore el uso del procesador sino que lo interrumpe y si el grado de interrupción es alto podría provocar que incluso fuera más lenta la ejecución de procesos puesto que tendríamos al procesador dedicado a atender dichas interrupciones."

Comment:

Aquí vuelve a estar claro que se llama interrupción al efecto y no al mecanismo. Esto es un problema en esta pregunta.

Código: DIF 4 No justifica por qué motivo ciertas partes críticas del sistema operativo como por ejemplo el cambio de contexto se deben ejecutar con las interrupciones inhibidas

Definición: Este código se asigna cuando el estudiante ha contestado en una pregunta tipo test que ciertas partes críticas del sistema operativo se deben ejecutar con las interrupciones inhibidas, pero no ofrece explicación plausible para la respuesta proporcionada.

Número de citas: 6

E07: "Juzgo que la última opción es bastante cercana a lo que debería ocurrir, esto es, que cuando ciertas operaciones críticas del S.O. se estén ejecutando como el cambio de contexto o el proceso de subsanar un error grave, el procesador no debería ser interrumpido bajo ningún concepto. De hecho, eso es lo que me ha parecido observar como usuario cuando mi sistema operativo "cae" en uno de esos "errores graves" y no responde a absolutamente nada. En caso de que dos interrupciones concurren, se debe atender a la de mayor prioridad y, de ser relativa a una parte crítica de código, la atención que el procesador le debe conceder debe ser exclusiva."

E01: "" (El estudiante no responde a la pregunta)

E02: "" (El estudiante no responde a la pregunta)

E10: "No deberían interrumpirse las partes vitales del sistema operativo que gestionen las propias interrupciones por ejemplo."

E03: "Para que el propio computador pueda interrumpir un proceso para devolver el control al sistema operativo."

E08: "Aquellas que se ejecuta código del sistema operativo, es decir en modo privilegiado. Es el propio sistema operativo quien decide sobre el tratamiento de este tipo de interrupciones, es quien tiene que valorar su procesamiento."

Código: DIF 5 El cambio de proceso siempre se origina por una interrupción de reloj

Definición: Este código se asigna cuando el estudiante ha contestado en una pregunta tipo test que un cambio de contexto siempre se origina por una interrupción de reloj.

Número de citas: 3

E01: "Siempre se origina por una interrupción de reloj."

E03: "Excepto la respuesta a) creo que las demás las respuestas participan del cambio de contexto."

E10: "Siempre se origina por una interrupción de reloj."

Código: DIF 6 Una interrupción sirve para realizar un cambio de contexto

Definición: Este código se asigna cuando el estudiante ha declarado en su respuesta que una interrupción sirve para realizar un cambio de contexto.

Número de citas: 1

E01: "Puesto que una interrupción sirve, entre otras cosas, para realizar un cambio de contexto."

Código: DIF 7 Siempre que hay una interrupción de reloj, se realiza un cambio de contexto

Definición: Este código se asigna cuando el estudiante ha declarado en su respuesta que siempre que hay una interrupción de reloj, se realiza un cambio de contexto.

Número de citas: 1

E10: "Se realiza un cambio de contexto, siempre que hay una interrupción de reloj."