

# Aprendizaje guiado con simuladores y tutorización mediante mensajería instantánea

Martínez,M.O., Piñol,P., López,O.

Dpto. Física y Arquitectura de Computadores  
Universidad Miguel Hernández

## Resumen

En este artículo se pretende recoger la experiencia realizada en la asignatura ‘Sistemas Operativos’ de segundo curso de Ingeniería Técnica en Informática de Gestión, impartida en la Escuela Politécnica Superior de Orihuela de la Universidad Miguel Hernández. En concreto la experiencia se ha realizado sobre la parte del temario de la asignatura que cubre los conceptos de concurrencia, exclusión mutua y sincronización de procesos.

Con este ejemplo proponemos una metodología para la inclusión de las nuevas tecnologías en el aula y en las tutorías, cuyos objetivos principales son intentar garantizar un buen rendimiento en las prácticas de la asignatura así como obtener una mayor motivación del alumnado, una mejor comprensión de los conceptos teóricos y un uso planificado y guiado en el proceso de enseñanza-aprendizaje.

Para conseguir dichos objetivos se ha puesto en marcha, siguiendo la planificación de dicha metodología, el uso en el aula de una herramienta que simula el efecto de los *semáforos*, los *monitores* y la sincronización de procesos llamada JBACI (Java Ben-Ari Concurrent Interpreter)[11].

## 1. Introducción

Al inicio del curso se realizó una encuesta a los alumnos sobre los conocimientos previos necesarios para abordar sin dificultad ciertos aspectos prácticos de la asignatura. Esta encuesta mostró un perfil en el alumnado con conocimientos demasiado superficiales de programación en C, tanto en entornos Unix-like como en entornos Windows . También se descubrió que el porcentaje de alumnos que había instalado Linux en casa alguna vez era bajísimo y mucho menor el que lo utilizaba habitualmente.

En las aulas disponibles para prácticas se encontraba instalado únicamente Windows. Además, cierta parte del alumnado estaba más interesado en aprender Linux como simples usuarios que en ‘perder tiempo’ con la teoría de los Sistemas Operativos, ya que a su modo de ver ‘nunca’ se iban a encontrar con la necesidad de poner en práctica dichos conocimientos, bien como diseñadores o como programadores de sistemas. La desmotivación del alumnado estaba bien presente.

Nuestra intención inicial era desarrollar las prácticas de la asignatura relativas a concurrencia y sincronización de procesos en lenguaje C bajo Linux y pensamos que dicha aproximación es recomendable para esta asignatura si el perfil del alumnado lo permite. A parte de las carencias técnicas, se observó cierta pasividad en los alumnos respecto a la participación en el aula y la motivación en general; este comportamiento provoca la falta de ‘feedback’ hacia el profesor.

Intuimos que siguiendo la aproximación docente basada en Linux o Windows seguramente no cumpliríamos con las expectativas cuantitativas y/o cualitativas que entendemos necesarias para estos temas. Supusimos que perderían más tiempo con las problemáticas típicas de la inexperiencia en el lenguaje de programación que en centrarse en la comprensión de dichos conceptos. Pretendíamos centrar su atención en la resolución del problema y no en la sintaxis. Igualmente pensamos que para que los conceptos quedaran suficientemente claros, los ejercicios que deberían hacer en las prácticas debían tener un nivel de complejidad medio-alto y esto era necesario mantenerlo.

Se trata de una asignatura cuatrimestral, con un contenido denso en conceptos que deben quedar bien afianzados en el alumno. Lo ideal es realizar las prácticas de esta parte de la asignatura utilizando los mecanismos que un sistema operativo real pone a disposición del programador de sistemas (véase [9][1][2][3][10]). Pero como decimos, había que buscar una alternativa que asegurara, en cualquier caso, que el tiempo dedicado a llevarla a cabo fuese el mismo que el que se emplearía sin ella, pues el tiempo de cada bloque ya estaba fijado en la programación de la asignatura.

La alternativa que hemos encontrado se basa en el uso de JBACI, un simulador de concurrencia, ejecutable en distintos entornos y con una sintaxis muy sencilla y reducida. Es necesario adaptar los problemas típicos de concurrencia y sincronización a dicha herramienta. Hemos desarrollado una metodología para la docencia de los ejercicios y prácticas basándonos en dicha herramienta.

Debido a que el profesorado de la asignatura dista físicamente de los alumnos por el carácter multicampus de la Universidad Miguel Hernández, hace necesario buscar alternativas para evitar desplazamientos a profesores y alumnos. Para evitar esto, el seguimiento, tutorización y soporte al grupo de alumnos se realizó mediante la mensajería instantánea MSN Messenger de la que se utilizan sus características de compartición de aplicaciones, voz.

## 2. Presentación de la Asignatura

La asignatura ‘Sistemas Operativos’ es una asignatura cuatrimestral de 7,5 créditos distribuidos en 4,5 créditos teóricos y 3 créditos prácticos. Se imparte en el primer cuatrimestre a razón de tres horas de teoría semanales y dos de prácticas. La experiencia debía garantizar no consumir más tiempo que el inicialmente programado 7,5 horas, distribuidas en cinco sesiones de hora y media, de las cuales 2 sesiones son de teoría y el resto de resolución de ejercicios.

## 3. Motivación metodológica

Nuestra propuesta de desarrollo de esta parte de la asignatura la basamos en las estrategias para desarrollar expectativas positivas de aprendizaje descritas por J.L. Castillejo en [4]:

- *Es necesario que el alumno tenga continuadas experiencias de éxito.* Para ello el profesor plantea problemas muy sencillos utilizando JBACI, pero con errores fácilmente detectables por el alumno, lo que provoca pequeños debates en el aula para resolverlos. Como efecto añadido, el alumno va asimilando la sintaxis de JBACI.
- *En el diseño de instrucción deben aparecer con especificidad y claridad los requisitos para lograr los objetivos propuestos.* Puesto que el objetivo era que realizaran prácticas complejas para los conceptos de exclusión mutua y sincronización, se presentaron las prácticas obligatorias que debían realizar.
- *Se eleva el nivel de expectativas en la medida que se permite un autocontrol de los resultados y se usan refuerzos de atribución que permitan conectar el éxito con su habilidad y esfuerzo personal.* Durante las sesiones teóricas en las que se presentan ejercicios a resolver se crean grupos de alumnos que deben solucionar dichos ejercicios; de esta forma se fomenta la interrelación grupal en el aula y el aprendizaje cooperativo.

Para potenciar las estrategias expuestas anteriormente, nos hemos basado en algunos elementos didácticos que sugieren C. Coll y otros en [5] y, J. Gimeno en [8]. Estos autores argumentan que el diseño y la planificación de la enseñanza deberían prestar atención simultáneamente a cuatro dimensiones:

- *Los contenidos a enseñar.* Son los contenidos propios del temario de esta parte de la asignatura como son la exclusión mutua y sincronización de procesos usando *semáforos* y *monitores* entre otros mecanismos.
- *Los métodos de enseñanza.* Basándonos en las tres estrategias metodológicas descritas anteriormente, hemos puesto en práctica la planificación metodológica que se detalla en el próximo apartado, haciendo uso de herramientas tecnológicas en el aula para potenciar el proceso de enseñanza-aprendizaje.
- *La secuenciación de los contenidos.* La secuenciación de los contenidos se desarrolla en la metodología propuesta introduciendo primeramente ejemplos sencillos e incrementando posteriormente el nivel de dificultad y contenidos progresivamente.
- *La organización social de las actividades de aprendizaje.* Se ha fomentado la participación en grupos dentro del aula, resolviendo los ejercicios planteados con el fin de aumentar la motivación y la participación.

## 4. Planificación metodológica

La planificación metodológica realizada se muestra en la figura 1.

En primer lugar se realiza la presentación teórica de los conceptos de concurrencia, exclusión mutua, sincronización, *semáforos*, *semáforos binarios* y *monitores*. Esta exposición se realiza de la forma más concisa y clara posible, introduciendo algún ejemplo, pero sin dedicar excesivo esfuerzo en que el alumno sea capaz, con dichas exposiciones, de resolver cualquier problema. Seguidamente, se pone en conocimiento de los alumnos los enunciados de las prácticas obligatorias que deben realizar en el laboratorio, cuya



Figura 1: Esquema metodológico

complejidad es media-alta. Entendidos los enunciados, se pasa una encuesta a los alumnos cuyo objetivo es establecer un nivel de referencia para poder medir de alguna forma las mejoras obtenidas por la aplicación de la metodología. Tras evaluar los resultados de la encuesta, vimos que la mayoría de los alumnos pensaban que les resultaría muy difícil realizar la práctica, tal y como habíamos previsto. Posteriormente, se hace una presentación del entorno gráfico de programación concurrente JBACI aprovechando que se muestra el primer ejemplo de programación concurrente, de manera que se les explica los menús y las diferentes opciones del programa. Tras esta presentación del entorno, se presentan problemas resueltos con JBACI en el aula de teoría mediante el uso de un cañón conectado al ordenador donde el profesor utiliza el simulador y transparencias para ir explicando los conceptos teóricos. Se empiezan las simulaciones con problemas muy sencillos implementados con *semáforos*, *semáforos binarios* y *monitores* cuyo único objetivo es afianzar los conocimientos teóricos y llamar la atención del alumno para conseguir que piense que no es tan difícil como parecía a priori y que se sienta capaz de ir participando en los siguientes ejercicios planteados. Para cada problema presentado se discute en clase el enunciado y una vez entendido el problema se presenta el código en JBACI. En unos casos este código corresponde a la solución y en otros se presenta una solución incompleta o con errores que se resuelven ‘en directo’, preferiblemente como consecuencia de un debate provocado acerca de cómo solucionarlo.

Estos ejercicios planteados por el profesor han sido desarrollados por un grupo de alumnos del curso anterior en prácticas tutorizadas. Para estas prácticas utilizamos las tutorías virtuales con la herramienta MSN Messenger. Dado el buen funcionamiento de la herramienta para este fin (véase [6]) y el carácter multicampus de nuestra universidad, hemos extendido este tipo de tutorías a todos los alumnos de este curso.

Tras la realización de las prácticas, se vuelve a realizar la encuesta al alumnado con el objetivo de poder medir las diferencias respecto a la encuesta inicial y analizar sus resultados. La mayoría de los alumnos coinciden en que les ha resultado menos difícil de lo que inicialmente habían supuesto. También han respondido de forma satisfactoria en cuanto al uso del simulador JBACI.

## 5. Herramientas

### 5.1. JBACI

JBACI es una opción para obtener experiencia en programación concurrente, que facilita el entendimiento de los conceptos de sección crítica, concurrencia y sincronización. JBACI viene de Java Ben-Ari Concurrent Interpreter. Se trata de un compilador simplificado de lenguaje C++ que simula la ejecución de procesos concurrentes y soporta *semáforos enteros*, *semáforos binarios* y *monitores*.

Como hemos comentado, el nivel de programación en lenguaje C de los alumnos era suficientemente bajo como para cuestionarnos el uso de Linux en las prácticas de la asignatura, ya que pensamos que perderían más tiempo resolviendo problemas relacionados con la inexperiencia en el lenguaje que centrándose en los contenidos. Por este motivo y los siguientes decidimos utilizar el simulador JBACI:

- *Simplicidad en la sintaxis de JBACI.* Un subconjunto muy limitado de C++.
- *Entorno multiplataforma.* Fácil instalación independientemente del sistema operativo de que se disponga, pues existen las versiones en JAVA (Windows), Linux/Unix y MS-DOS.
- *Se trata de un software de distribución gratuita y se distribuye con el código fuente.*
- *El entorno JBACI utilizado es un entorno gráfico muy sencillo de utilizar.* Permite la ejecución paso a paso, pudiendo realizar operaciones de depuración y obtener el valor de las variables tanto globales como las internas de los *semáforos y monitores*.
- *JBACI actualmente es una herramienta muy extendida y usada en otras universidades [11].*

## 5.2. Tutorías con MSN Messenger

MSN Messenger es una aplicación de libre distribución que permite la conexión directa entre dos o más ordenadores de forma que dos o más personas pueden mantener una conversación en tiempo real, bien de forma escrita, oral e incluso visual. El único medio técnico que se necesita es una conexión a Internet.

La aplicación MSN Messenger resultó de una utilidad fundamental en el desarrollo de las tutorías virtuales. Se trata de un software de libre distribución, muy intuitivo y sobre todo ampliamente utilizado por la comunidad estudiantil, lo que implica que no es necesario ningún tipo de explicación sobre la instalación o el manejo de dicha aplicación. Se ha utilizado la opción ‘Compartir aplicación’ que permite al profesor ver lo que está realizando el alumno en el entorno JBACI, así como tomar el control de dicha aplicación de forma remota y solucionar las dudas que plantea. Además se dispone de otros tres medios que permiten que la comunicación sea completa. La ventana de texto, que permite el intercambio de preguntas y respuestas de forma escrita, el micrófono, que permite comunicarse de forma verbal y la pizarra, que esporádicamente permite realizar esquemas o gráficos sencillos que añadan claridad a la explicación. Tanto las conversaciones en texto como las explicaciones en la pizarra pueden ser guardadas para consultas posteriores por los estudiantes.

Este sistema permite una reunión ‘virtual’ en la que cada persona puede estar en campus distintos (o incluso en su propia casa) y participar de forma conjunta. Además de evitar los desplazamientos de los alumnos, éstos no están sujetos a las horas de tutorías pues se les ha ofrecido la posibilidad de contactar con el profesor en cualquier momento en el que éste esté conectado (siempre que éste disponga de tiempo).

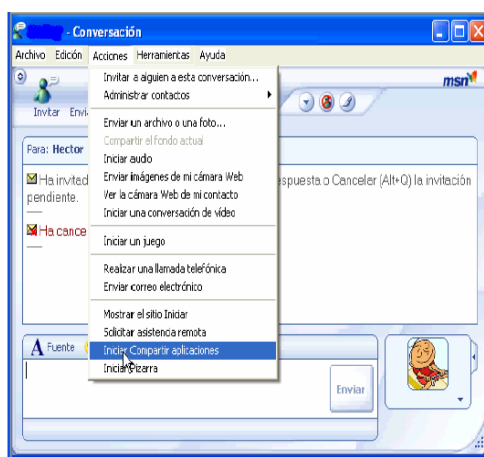


Figura 2: Pantalla de conversación del Messenger

## 6. Ejemplo de ejercicio

Para poner un ejemplo de cómo se ha realizado la experiencia, se ha capturado una imagen de JBACI. En la página web de BACI [11] se puede acceder a este entorno.

En la figura 3 se muestra la ventana de edición del entorno de desarrollo JBACI. El código de la figura corresponde al primer problema propuesto, donde se utilizan dos procesos cooperantes para conseguir un objetivo, sumar 20, donde cada proceso aporta 10 a la suma total. Se aprovecha el ejemplo para presentar la forma de inicializar en JBACI los *arrays*, la sintaxis del bucle *for*, la sentencia *cobegin*, cómo imprimir en la consola con *cout* y cómo obtener el identificador de un proceso con la función *wich-proc*. Pero sobre todo se

```

1 int total;
2 //semaphore s=1;
3
4 int arr[5];
5
6 void add10() {
7     int i;
8     for (i = 1; i <= 10;i++)
9     {
10        //wait(s);
11        cout << which_proc();
12        total = total + 1;
13        //signal(s);
14    }
15 }
16
17 void inicializar ()
18 {
19     int count;
20     for (count=0; count<5; count++)
21     {
22        arr[count]=count;
23    }
24 } //inicializar//
25
26
27 void main() {
28     total = 0;
29     inicializar();
30     cobegin {
31        add10();
32        add10();
33    }
34     cout << "Sum = " << total << endl;
35 }

```

Figura 3: Editor código JBACI

llama la atención de los alumnos en cómo se puede dividir una tarea en la ejecución de procesos cooperantes para realizarla. En el ejercicio se pretende demostrar cómo, si no se protege mediante exclusión mutua la variable *total*, que es utilizada por los procesos cooperantes, se pueden producir resultados erróneos por no garantizar la exclusión mutua en la lectura y actualización de la variable *total*, es decir, la instrucción  $total = total + 1$  debería ejecutarse sin interrupción. En la imagen aparecen comentadas las sentencias de declaración del *semáforo* que se usa para establecer la sección crítica y las sentencias *wait* y *signal* que la implementan. En el ejemplo que se muestra a los alumnos, estas instrucciones comentadas, no aparecen. Se ejecuta el programa varias veces y se observa que cada vez da un resultado distinto y que normalmente no se obtiene el resultado esperado, 20.

En la figura 4 podemos ver el proceso principal *main* y los dos procesos cooperantes. Este código se

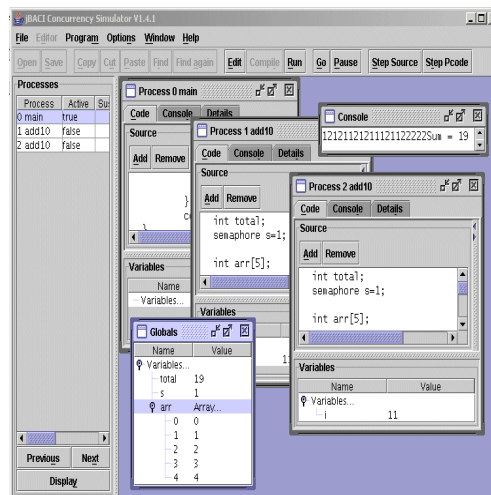


Figura 4: Pantalla ejecución JBACI

puede ejecutar paso a paso y observar que la ejecución se alterna entre los dos procesos. En la ventana de la consola se van mostrando resultados y en la ventana de variables el valor interno de los *semáforos*. La alternancia de procesos se puede observar en la lista de procesos(a la izquierda de la imagen).

## 7. Futuras actuaciones

Como futuras actuaciones se habilitarán dos foros, uno técnico, referido a problemas de instalación de la herramienta JBACI y otro teórico-práctico sobre aspectos propios de la materia y la resolución de problemas. La participación en los foros será tomada en cuenta en la evaluación final, de forma que se motive a los grupos a resolver los problemas planteados por los compañeros.

También pretendemos generar unas FAQ's (Frequently Asked Questions) con las dudas más interesantes resueltas durante las tutorías. Estas FAQ's podrían incluir grabaciones de las sesiones de tutorías utilizando software de captura de pantalla. Dicho material multimedia podría servir como material complementario autodidacta para los alumnos (véase [7]).

## 8. Conclusiones

Nuestra experiencia nos ha proporcionado una gran satisfacción al ver que el grado de asimilación de los conceptos presentados ha sido mayor de lo esperado. Su aplicación tuvo un efecto positivo, incrementando la participación y atención del alumno.

Como principales logros obtenidos podemos enumerar que:

- En general el alumnado ha perdido el 'miedo' a la participación en clase y a utilizar las tutorías.
- Se ha mejorado la comprensión de la teoría presentada.
- Los alumnos llegan a las prácticas más preparados y aprovechan mejor el tiempo de la práctica.
- Se han obtenido mejores resultados que en otros temas de la asignatura.
- Se ha eliminado la problemática de la distancia para las tutorías gracias a la tutoría virtual con MSN Messenger.
- Dado el alto grado de participación del alumnado en el uso de las tutorías virtuales, podemos decir que se ha logrado una mayor motivación del alumnado que en otras partes de la asignatura

Como conclusión final queremos remarcar que esta metodología podría ser aplicada a la docencia de cualquier otro área siempre que se disponga de la herramienta software necesaria, ya que dicha metodología se basa en la secuenciación de los contenidos, la sensación de logro por parte del alumnado (auto motivación), la claridad en la exposición de los contenidos y, la participación grupal en el aula y fuera de ella.

## Referencias

- [1] J. Carretero, F. Garcia, P.M. Anasagasti y F. Pérez. *Sistemas Operativos: una visión ampliada*. Mc Graw Hill, 2001.
- [2] J. Carretero, F. Garcia y F. Pérez. *Prácticas de Sistemas Operativos: de la base al diseño*. Mc Graw Hill, 2002.
- [3] J. Carretero, F. Garcia y F. Pérez. *Problemas de Sistemas Operativos: de la base al diseño*. Mc Graw Hill, 2003.
- [4] J.L Castillejo. *Pedagogía Tecnológica*. Ceac, 1987.
- [5] C. Coll, J. Palacios y A. Marchesi. *Desarrollo psicológico y educación*. Alianza, 1991.
- [6] V. Galiano, O. López, M. Martínez, H. Migallón, P. Piñol y D. Ubeda. *Docencia de semáforos y Monitores con BACI: Un ejemplo de aprendizaje guiado*. X Jornadas de enseñanza universitaria. Jenui 2004.
- [7] M.M Galotto, A. Pomares, O. López, P. Piñol, M. Martínez. *Mejora de la Docencia con la creación de material docente complementario autodidáctico*. [www/Internet](http://www/Internet) (CIAWI 2004).
- [8] J. Gimeno Sacristán. *Autoconcepto, Sociabilidad y Rendimiento escolar*. MEC, 1976.
- [9] F.M. Márquez. *Unix : Programación avanzada*. Ra-ma, 1996.
- [10] Kay A. Robbins y S. Robbins. *Unix: Programación práctica. Guía para la concurrencia, la comunicación y los multihilos*. Prentice Hall Hispanoamericana, 1997.
- [11] *Página web principal de BACI: [http : //www.mines.edu/fshome/tcamp/baci/](http://www.mines.edu/fshome/tcamp/baci/)*