

EXERCITA: UNA BASE DE DATOS DE MATERIAL PARA PRÁCTICAS DE PROGRAMACIÓN[§]

C. Gregorio Rodríguez¹, L. F. Llana Díaz¹, R. Martínez Unanue¹, P. Palao
Gostanza¹, C. Pareja Flores² y J. Á. Velázquez Iturbide³

¹*Facultad de Informática,
Depto. Sistemas Informáticos y Programación, U. Complutense de Madrid
e-mail: cgregorio@sip.ucm.es, llana@sip.ucm.es, raquel@eucmos.sim.ucm.es,
ecceso@eucmos.sim.ucm.es*

²*Escuela Universitaria de Estadística,
Depto. Sistemas Informáticos y Programación, U. Complutense de Madrid
e-mail: cpareja@sip.ucm.es*

³*Escuela Superior de Ciencias Experimentales y Tecnología,
Universidad Rey Juan Carlos de Madrid
e-mail: a.velazquez@escet.urjc.es*

RESUMEN: Presentamos en este artículo el diseño y realización de una base de datos de materiales y herramientas para la práctica de la programación. El núcleo de dicha base de datos es el lenguaje *eXercita*, cuya descripción, implementación y uso se explican en estas páginas. Se expone igualmente la organización del contenido de la base de datos (principalmente ejercicios y prácticas), que da cuenta de sus posibilidades, flexibilidad y amplitud. Estas herramientas permiten distribuir el material en papel o en formato electrónico, ya sea mediante CD-Rom o a través de Internet.

1.- INTRODUCCIÓN: HISTORIA Y MOTIVACIÓN

Este trabajo nació como respuesta a una necesidad de varios profesores que han venido enseñando distintas asignaturas de programación en los últimos años. Dicha necesidad no es nueva (como recogen un buen número de referencias y depósitos de materiales; véase la bibliografía): consiste esencialmente en preparar y renovar materiales educativos, atractivos y originales; y la solución más simple era reciclar el material propio y el de los demás. El primer beneficio es tener más tiempo, ya sea para ampliar la colección o para otras de nuestras múltiples tareas. Además, tal compilación de recursos abundantes, frescos, variados y de calidad supone para el alumno un estímulo hacia una tarea tan bonita como es la programación.

[§]Este trabajo ha sido financiado por la Universidad Complutense de Madrid, mediante el proyecto de innovación educativa *Elaboración de material educativo para prácticas de programación*, ref. PIE 99/32, 2000.

Pero los múltiples colaboradores dispuestos a aportar material quisieran poder usar toda la colección a su medida: con distintos estilos de presentación, con la posibilidad de incluir o no pistas, todos o parte de los apartados, soluciones, ejemplos, juegos de datos, indicaciones de otros materiales en que se basan, lecturas complementarias, grado de dificultad previsto, etc. En resumen, la estructura de los ejercicios debía ser capaz de abarcar una casuística notablemente compleja debido a la diversidad de la información que puede interesar al plantear un ejercicio o una práctica.

Así pues, cada ejercicio individual se compone de múltiples piezas que se han de identificar para su procesamiento, y cada colección que se desea producir también. Para que toda esta información se pueda manejar de forma automática, ha habido que empezar por diseñar un lenguaje de marcado a la medida de estas necesidades y otras parecidas, y para que todo ello funcione, se han desarrollado compiladores y herramientas apropiadas.

Este artículo presenta el resumen de dicho lenguaje y herramientas. El manual de uso de eXercita [GLMPPV00] da cuenta de un sinfín de pormenores técnicos y de uso que no tienen cabida aquí.

En los siguientes apartados describimos el lenguaje de marcas usado (apartado 2), así como los compiladores de este lenguaje hacia PostScript y HTML, en que se presentará el material a los usuarios finales: alumnos y profesores (apartado 3). En el apartado 4 hablamos del contenido de dicha base de datos, y de la organización del mismo. Finalmente, en el apartado 5 resumimos nuestras conclusiones.

2.- ESPECIFICACIÓN: LENGUAJE Y CONCEPTOS

En este apartado describimos el lenguaje eXercita, que esencialmente es una notación para almacenar y posteriormente gestionar enunciados y soluciones de ejercicios, exámenes y prácticas.

El lenguaje de elaboración tipográfica es LaTeX; esto quiere decir que todas las piezas de texto dentro de un ejercicio se escriben con la notación de ese lenguaje. En cierto sentido, eXercita es sólo una capa de marcado que envuelve LaTeX, dando lugar a un lenguaje orientado hacia un dominio específico, que es el problema que nos atañe. En nuestra opinión, las alternativas más razonables son LaTeX, HTML y XML; el resto de los lenguajes no son legibles o están demasiado orientados a su generación y tratamiento automatizado (PDF, RTF), o no permiten manejar la estructura de los documentos, sino sólo su presentación. Hemos elegido LaTeX frente a HTML porque es más conciso, es más usual generarlo a mano, está orientado a la producción de documentos impresos en alta calidad y existen buenos traductores de LaTeX a HTML. En cuanto a XML, entra de lleno en nuestros planes.

La descripción del lenguaje eXercita en su totalidad excede ampliamente el espacio disponible en estas páginas. No obstante, enumeramos algunas de las marcas que lo definen para dar una idea de sus posibilidades expresivas:

- Parámetros de particularización: fecha (con distintos estilos), asignatura, curso, grupo, año académico, titulación, centro, institución, lenguaje de programación, etc.
- Piezas: título, propósito, resumen, texto, prólogo, enunciado, apartado, solución, pista, ejemplo, juego de datos de prueba, epílogo, procedencia, historia del problema, bibliografía, etc.
- Elementos diversos: para incluir logotipos, parámetros, referencias cruzadas a otros ejercicios, tratamientos condicionales diversos, etc.

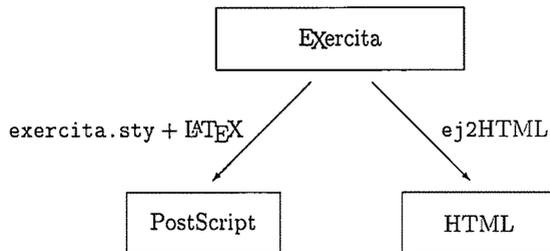
- Requisitos de uso: dependencias entre ejercicios, estilos, formatos y paquetes de LaTeX, etc.
- Etiquetas de catalogación: modelo de programación, lenguaje, conceptos del lenguaje usados, grado de dificultad previsto, envergadura del ejercicio, etc.

En el manual de uso de eXercita se puede consultar el detalle de todas las marcas mencionadas, así como su significado y manejo

3.- IMPLEMENTACIÓN: HERRAMIENTAS

En este apartado se describen dos de las herramientas que hemos construido para transformar ejercicios en algo legible: o un documento impreso en papel, o un lugar interactivo en la red. Estas herramientas cubren nichos completamente distintos, uno tradicional y otro muy moderno, pero ambos reales y hoyados en la práctica. También muestra dos extremos en la relación entre ejercicio y estudiante: la primera estática, decidida por los docentes, y la otra dinámica y elegible por quien estudia para ejercitar sus carencias o afinidades.

Aunque no es éste el momento de entrar en detalles técnicos, las herramientas desarrolladas operan como sigue



donde el paquete `exercita.sty` (escrito en TEX) permite empotrar eXercita en LaTeX y, por tanto, generar PostScript, mientras que `ej2HTML` (escrito en Perl) compila a HTML. En el manual de eXercita se explica detalladamente el uso de estas herramientas. Ahora, repasemos brevemente estos dos modos de publicación.

a) Publicación de ejercicios con LaTeX.

El uso más convencional de eXercita es generar un documento impreso. Lo más fácil habría sido construir un compilador que transformara código eXercita a código LaTeX, de donde se podría generar un documento en una segunda fase. Pero es mejor construir una serie de paquetes LaTeX que traduzcan directamente texto escrito en eXercita en un documento. De esta forma evitamos tener que hacer un procesamiento en dos o más pasos y se gana en portabilidad, porque TeX es igual en toda máquina.

El procesamiento de eXercita con LaTeX se divide en dos niveles:

- El nivel inferior está gobernado por un motor de *análisis de las marcas* de eXercita. Aunque este motor sólo analiza estas marcas, sin saber qué hacer con ellas, se puede configurar completamente, indicando qué marcas se han de capturar y qué debe hacerse con ellas.

Aunque se puede configurar esta capa con muy poco trabajo, exige unos conocimientos mínimos de programación con LaTeX (e incluso TeX) que no todo usuario posterior tendrá obviamente, además de ser necesario repetir ese esfuerzo para cada documento.

- Pero como, salvo en contadas ocasiones, siempre se quiere cargar las mismas partes y hacer lo mismo con ellas, sobre este nivel se pueden desarrollar paquetes que lo configuren de cualquiera de estas formas usuales. El superior puede ser ocupado por infinidad de *paquetes* y *clases* (estilos). Aunque actualmente sólo se ha desarrollado una clase (uniexer), ésta incluye parámetros que permiten configurarla para propósitos muy variados.

b) Publicación de ejercicios en Internet.

El lenguaje destinatario para la publicación en la red es HTML. También usamos JavaScript para hacer las presentaciones más cómodas, activas y comunicativas. El aspecto final es fácilmente modificable gracias a las hojas de estilo.

La herramienta concreta utilizada para generar un documento de red a partir de una colección de ejercicios es ej2HTML, que básicamente consiste en la composición secuencial de nuestra ej2TeX y la conocida LaTeX2HTML de Nikos Dragos.

La estructura del archivo HTML obtenido se basa en el encapsulamiento de la información en capas (*layers*) que son manejables con el lenguaje JavaScript y permite presentar la información procedente de la descripción de los ejercicios de un modo práctico y dinámico, mostrando la información menos relevante sólo si el usuario lo requiere.

Por supuesto, el aspecto del documento generado conserva toda la calidad que permite LaTeX, incluyendo imágenes de las fórmulas matemáticas, figuras, etc.

Junto con el documento final, la presentación final generada integra, en la misma ventana de navegación, el índice de ejercicios generado por ej2TeX y los códigos JavaScript necesarios para que todo funcione correctamente. Se ha incluido un apéndice con sendas figuras para mostrar el aspecto que presenta dicha ventana de navegación, así como una página en papel del documento generado.

4.- CONTENIDO DE LA BASE DE DATOS

a) Criterios de catalogación del contenido.

El contenido de la base de datos está estructurado físicamente por los temas argumentales sobre los que versan los ejercicios. No obstante, la organización física no constituye más que uno entre otros muchos criterios usuales de catalogación: por nivel de dificultad, por extensión, por mecanismos de programación requeridos en su resolución, por modelo(s) de programación recomendados, etc. El lenguaje de marcas de eXercita también contempla estas posibilidades de catalogación, para su posterior selección por los potenciales usuarios. Un par de ejemplos:

Puede interesar a un estudiante de programación en el paradigma imperativo repasar con ejercicios sencillos y no muy largos, sobre bucles.

Un profesor podría estar interesado en buscar enunciados para una práctica final, o desear preparar una hoja de problemas para practicar bucles con ejercicios sobre el bonito tema del número π y las múltiples formas de aproximarse al mismo.

En el manual de eXercita se detallan las posibilidades de catalogación consideradas por el momento.

b) Catalogación y mantenimiento colaborativos.

El planteamiento inicial de eXercita era y sigue siendo la compartición de recursos por distintas personas. Por consiguiente, la misma base de datos ha de permitir a distintas personas la aportación de material nuevo y el mantenimiento, posiblemente en paralelo, evitando las colisiones entre ellos. Este tipo de trabajo cooperativo ya se ha explorado en cientos de proyectos de *software* libre. Sus experiencias han revelado que CVS (*Concurrent Versions System*) [CVS] es una herramienta idónea. Actualmente, nuestro servidor de CVS sólo admite conexiones a los miembros de nuestro grupo, pero en un futuro permitiremos acceso de lectura público.

5.- TRABAJO FUTURO

Los objetivos planteados más a corto plazo para mejorar las herramientas y el material descritos con los siguientes:

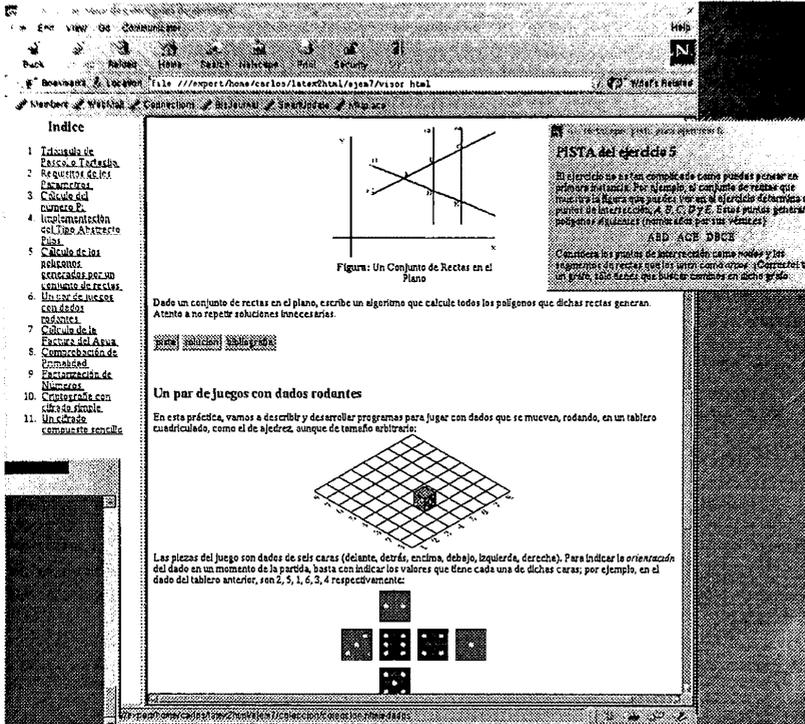
- Preparar el sitio de Internet que facilite el uso libre de estos materiales. Dicho sitio pretende servir a la vez como punto de partida para otros recursos útiles a quien aprende o enseña. Este objetivo debe estar cumplido al principio del curso 2000-2001.
- Construir las herramientas necesarias para flexibilizar la obtención de los materiales de la base de datos. Concretamente, para seleccionar los recursos y el estilo deseados, mediante simple marcado durante su revisión por Internet.
- Difundir la base de datos, asegurando en lo posible la retroalimentación.
- En particular con nuestros alumnos, establecer criterios para medir de un modo fiable la eficacia educativa que tiene el uso de estos materiales.
- Definir en XML un equivalente a nuestras marcas de eXercita, y desarrollar un tercer traductor de eXercita a XML que lleve las marcas normales de LaTeX a marcas normales de HTML, y las marcas de eXercita a las nuevas marcas en XML.

BIBLIOGRAFÍA.

- [B*99] J. Bergin y otros, *Resources for Next Generation Introductory CS Courses*, SIGCSE Bulletin, vol. 31, núm. 4, diciembre de 1999.
- [CVS] Open Source Version Control Software, en la dirección <http://www.cvshome.org/>
- [GLMPPV00] C. Gregorio Rodríguez, L. F. Llana Díaz, R. Martínez Unanue, P. Palao Gostanza, C. Pareja Flores y J. Á. Velázquez Iturbide, *EXercita: manual de uso*, Informe técnico SIP 102-00, Departamento de Sistemas Informáticos y Programación, mayo de 2000.
- [K99] D. Knox, *On-line publication of CS laboratories*, SIGCSE Bulletin, vol. 29, núm. 1, marzo de 1997.
- [K99] D. Knox, *The Computer Science Teaching Center*, disponible en <http://www.cstc.org/>, julio de 1999.
- [K*00] S. Kovourov y otros, "SAIL: A system for generating, archiving and retrieving specialized assignments using LaTeX", Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education, SIGCSE Bulletin, vol. 32, núm. 1, marzo de 2000.
- [JPPV00a] R. Jiménez Peris, C. Pareja Flores, M. Patiño Martínez y J. Á. Velázquez Iturbide, "New technologies in computer science education", en *Computer Science Education in the 21st Century*, Tony Greening (ed.), Springer-Verlag, 2000.
- [M99] R. McCauley, *Computer Science Education Links*, disponible en <http://www.cacs.usl.edu/~mccauley/edlinks/>, julio de 1999.
- [V96] J. Á. Velázquez Iturbide, "Estado del arte en enseñanza universitaria de la informática", *Novática*, 122, julio-agosto de 1996.

APÉNDICE

A) Aspecto de la ventana de navegación.



b) Un fragmento de un documento generado.

Hoja de ejercicios
7 de junio de 2000

Año 1999/2000

- ▷ 1. Un problema que a primera vista puede resultar muy difícil es el de calcular los polígonos que generan un conjunto de rectas en el plano. Considera el ejemplo de la figura 1. En dicho ejemplo puedes apreciar como cuatro rectas r_1, r_2, r_3 y r_4 , han determinado cinco puntos de intersección A, B, C, D y E . Los polígonos que pueden generar dichas rectas son aquellos que tienen por vértices las intersecciones y como lados segmentos incluidos en dichas rectas

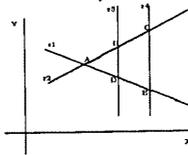
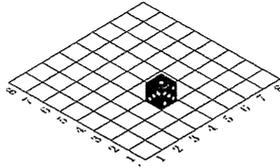


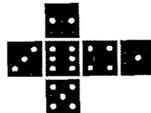
Figura 1: Un Conjunto de Rectas en el Plano

Dado un conjunto de rectas en el plano, escribe un algoritmo que calcule todos los polígonos que dichas rectas generan. Atento a no repetir soluciones innecesarias.

- ▷ 2. En esta práctica, vamos a describir y desarrollar programas para jugar con dados que se mueven, rodando, en un tablero cuadrículado, como el de ajedrez, aunque de tamaño arbitrario:



Las piezas del juego son dados de seis caras (delante, detrás, encima, debajo, izquierda, derecha). Para indicar la *orientación* del dado en un momento de la partida, basta con indicar los valores que tiene cada una de dichas caras; por ejemplo, en el dado del tablero anterior, son 2, 5, 1, 6, 3, 4 respectivamente:



Además, su *posición* en un tablero de $N \times N$ queda registrada en un par de números del conjunto $\{1 \dots N\}$. Por ejemplo, el dado de la figura de arriba está en la posición (4,3). Se pide: