

Actas XVIII JENUI 2012, Ciudad Real, 10-13 de julio 2012
I.S.B.N. 10: 84-615-7157-6 | I.S.B.N. 13:978-84-615-7157-4
Páginas 9-16

Diez años innovando en la enseñanza de los fundamentos de la programación: resultados y conclusiones

Jorge García
Dpto. Lenguajes y
Sistemas Informáticos
Reina Mercedes, s/n
41012 - Sevilla
jorgarcia@us.es

José C. Riquelme
Dpto. Lenguajes y
Sistemas Informáticos
Reina Mercedes, s/n
41012 - Sevilla
riquelme@us.es

Mariano González
Dpto. Lenguajes y
Sistemas Informáticos
Reina Mercedes, s/n
41012 - Sevilla
mariano@us.es

Isabel Nepomuceno
Dpto. Lenguajes y
Sistemas Informáticos
Reina Mercedes, s/n
41012 - Sevilla
inepomuceno@us.es

Resumen

Este artículo presenta los cambios experimentados en las asignaturas relacionadas con la introducción a la programación, impartidas en las titulaciones de Ingeniería Informática de la Universidad de Sevilla durante el período 2001-2011. El artículo hace especial hincapié en las principales innovaciones realizadas a lo largo de dicho período, entre las que destacan: la apuesta por la orientación a objetos en detrimento de la programación estructurada como paradigma de iniciación, el uso de un lenguaje real (Java) y la consiguiente eliminación del pseudocódigo como lenguaje de iniciación, y la introducción de esquemas y librerías (Guava) orientados a facilitar la tarea de programación de los futuros profesionales. Finalmente, mostramos los resultados obtenidos por nuestros alumnos a lo largo de este período y las principales conclusiones que se pueden extraer de su análisis.

Summary

This article presents the changes in the subjects related to the introduction to programming taught in the Computer Engineering degree at University of Seville in the period 2001-2011. The article puts special emphasis on major innovations during this period: object-orientation in lieu of structured programming as a paradigm of initiation, use of real languages (Java) and subsequent elimination of the pseudocode as a language of initiation, and the introduction of patterns and libraries (guava) designed to facilitate the programming tasks of future professionals. Finally, we show the results obtained by our students during this period and show the main conclusions can be drawn from the analysis of the results.

Palabras clave

Fundamentos de programación, ingeniería informática, Java, esquemas, Guava.

1. Introducción

Hace 10 años, algunos profesores de las asignaturas Introducción a la Programación I y II (en adelante, IP1 e IP2) presentaron en las JENUI'00 un nuevo programa para las asignaturas de primer curso de Ingeniería Informática [1] comunes a las tres titulaciones existentes (Ingeniería Superior, Ingeniería Técnica de Sistemas e Ingeniería Técnica de Gestión). Ambas asignaturas comenzaron a impartirse en el curso 1997-98, con la implantación de los nuevos planes de estudio (plan 97) y tenían una duración de un cuatrimestre cada una, con una carga lectiva de 7,5 de los antiguos créditos universitarios LRU (previos a la implantación del Espacio Europeo de Educación Superior, EEES).

La asignatura IP1 desarrollaba un esquema clásico de introducción a la programación estructurada, con el lenguaje C como primer lenguaje para implementar programas, y un lenguaje denominado LEA (Lenguaje de Especificación de Algoritmos) que introducía los conceptos teóricos en pseudocódigo [2]. Los objetivos de IP1 se resumían en:

- Desarrollar una aproximación disciplinada a la especificación, implementación, verificación y documentación de programas.
- Apreiciar el papel central que juega la abstracción en la tarea de programar.
- Proporcionar los fundamentos teóricos y prácticos, básicos para cursar posteriores estudios en programación.
- Estudiar un lenguaje de alto nivel, estructurado, general y extendido.

IP2 continuaba en la misma línea, pero introducía la programación orientada a objetos sustituyendo el lenguaje C por C++. Así, se centraba en la implementación de tipos de datos (listas, pilas, colas, árboles, etc.) manteniendo el lenguaje LEA en las clases de teoría, lo que exigía una adaptación de dicho lenguaje para introducir los conceptos de la orientación a objetos. Teniendo en cuenta lo anterior, los objetivos generales de IP1 se extendían con los siguientes elementos:

- Conocer estructuras de datos, algoritmos y esquemas de uso general.
- Introducir el paradigma de la programación orientada a objetos.
- Aprender un lenguaje de programación orientado a objetos (OO).

Respecto a la organización de las clases, éstas se organizaban en 3 horas de teoría y 2 de prácticas semanales. Los grupos de teoría tenían alrededor de 125 alumnos que se dividían en dos subgrupos para las clases de prácticas. Las clases de teoría se realizaban sobre pizarra y las prácticas se realizaban en aulas de ordenadores (aproximadamente 30 equipos por aula) en las que los alumnos se disponían en parejas dado el elevado número de alumnos por grupo. Las clases de teoría y de prácticas no tenían por qué ser impartidas por los mismos profesores, por lo que el grupo de profesores de la asignatura tenía que hacer un importante esfuerzo de coordinación. Máxime sabiendo que, en el año 2000, el número de matriculados en ambas asignaturas era superior a 1000 alumnos y que continuó progresivamente aumentando en los primeros años de la década debido a los problemas que tenían los alumnos para superarlas.

Por otro lado, los egresados y las empresas que se nutrían de ellos empezaron a llamar la atención sobre la necesidad de cambiar el modelo de aprendizaje, incidiendo menos en aspectos relacionados con la algoritmia y haciendo lo propio en aspectos más ingenieriles como son la reutilización de código, la capacidad del alumno para investigar en nuevas tecnologías e integrarlas en sus desarrollos, o el conocimiento profundo de los lenguajes orientados a objetos más utilizados en el entorno de la Universidad de Sevilla. Para entender el problema que se planteaba, podemos comentar que, por ejemplo, las asignaturas de IP invertían una gran canti-

Didáctica en los estudios de ingeniería informática

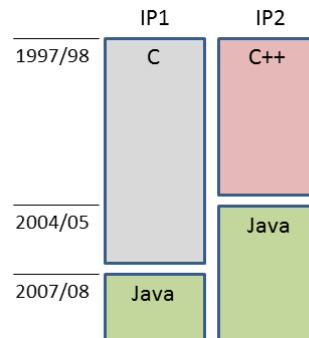


Figura 1: Principales hitos en la transformación de las asignaturas de IP.

dad de tiempo en hacer que los alumnos dominaran algoritmos de ordenación. Sin embargo, Java integra en su API mecanismos de ordenación optimizados basados en las técnicas quicksort y mergesort, tanto para arrays (clase `java.util.Arrays`) como para colecciones de objetos (clase `java.util.Collections`) [3]. Así pues, el nuevo objetivo haría mayor hincapié no tanto en dominar las técnicas de ordenación, sino saber utilizarlas correctamente.

Los nuevos objetivos anteriormente descritos están íntimamente relacionados con la entrada en vigor de los nuevos grados y el EEES, que tiene como objetivo prioritario para los grados, "la adquisición de los conocimientos y capacidades específicas orientadas a su incorporación al ámbito laboral"[4]. Perseguendo este objetivo, se tomó la decisión de cambiar el enfoque de las asignaturas de IP. Esta decisión ha transformado totalmente las asignaturas de introducción a la programación que progresivamente se han adaptado a las nuevas necesidades del alumnado y del entorno profesional en el que principalmente se integran al final de su vida académica. En la Figura 1, se muestran los principales hitos en dicha evolución.

El presente artículo pretende detallar los cambios realizados en los últimos diez años para pasar de una propuesta metodológica *objects-late* a una propuesta *objects-first* [10], analizando los resultados obtenidos y mostrando las principales conclusiones que se pueden extraer de ellos. Más concretamente, la Sección 2 detalla los principales cambios

realizados a lo largo del tiempo, la Sección 3 muestra y analiza los resultados obtenidos en cada etapa, y la Sección 4 resume las principales conclusiones extraídas del estudio junto a las posibles líneas para mejorar los resultados presentados.

2. Evolución metodológica

2.1. Introducción a la Programación

Durante los primeros años de la década, las asignaturas mantuvieron la estructura y el temario visto en la introducción, con ajustes mínimos que, en el curso 2004-05, se convirtieron en cambios notables. Así:

- En IP2, se abandona C++ para escoger Java como el lenguaje idóneo para desarrollar programas orientados a objetos.
- Se elimina la construcción de los tipos de datos, explicando el tipo Collection de Java y sus subtipos.
- Se abandona el pseudocódigo en IP2.
- En IP1, se mantiene el lenguaje C y el pseudocódigo.

El cambio sustancial de pasar de C++ a Java es debido principalmente al hecho de que el desarrollo software en el entorno de la Universidad de Sevilla se realiza mayoritariamente en este lenguaje que, además, es uno de los líderes en el paradigma OO [5, 12]. Este éxito puede ser explicado, entre otras razones, por su independencia de la plataforma y porque es un lenguaje sencillo (por ejemplo, no existe la aritmética de punteros) o todo lo sencillo que puede ser un lenguaje OO, lo que es clave para los no iniciados en las tareas de programación. El resto de cambios son relativamente lógicos puesto que el API de Java provee ya un conjunto de tipos suficientemente amplio para describir colecciones de datos. Por otro lado, se constató que la ampliación de LEA para IP2 estaba demasiado del lenguaje Java, ya que se había diseñado para facilitar a los alumnos la introducción a la algoritmia y no era fácil adaptarlo a la OO. Así, se optó por dar las clases teóricas directamente en Java en lugar de rehacer LEA.

Durante los siguientes tres años, los cambios anteriores se fueron consolidando hasta que en el curso 2007-08, se introdujo un nuevo salto en el programa de ambas asignaturas. Así pues, se establecen los siguientes cambios:

- Se elimina la programación estructurada como paradigma de aprendizaje inicial, desarrollándose desde IP1 la programación orientada a objetos en lenguaje Java (propuesta *objects-firts*). El alumno estudia desde el primer momento los conceptos de interfaz, clase y objeto, realizando sus primeros programas en Java.
- Se abandona completamente LEA, utilizándose Java desde el primer momento.
- Los conceptos básicos de Java se introducen en el temario de IP1 mientras que el temario de IP2 se amplía con nuevos conceptos como los patrones de diseño creacionales (fábricas) o la programación reflexiva.
- Se añaden elementos avanzados del lenguaje Java como los tipos *Iterable* y *Comparator*, nuevos tipos de colecciones del paquete *java.util* como *Map*, etc.
- Se aplica una filosofía docente basada en la formalización de los esquemas básicos de recorridos sobre colecciones de objetos: existe, paratodo, contador, mínimo, máximo, etc.
- El uso de estos esquemas se generaliza mediante el concepto de criterio o expresión booleana para realizar filtros, el de expresión para convertir a diferentes tipos, etc.
- Finalmente se añade un tema de C al final de IP2 para que los alumnos conozcan la sintaxis básica de C, sus principales características diferenciadoras respecto a Java y cómo formalizar un programa en C siguiendo un esquema similar al aprendido para definir tipos en Java.

Los problemas de LEA para ser usado en IP2 se volvieron a repetir al eliminar la programación estructurada de IP1. De nuevo, se tomó la decisión de eliminar LEA del temario. Por otro lado, para explicar la introducción de la orientación a objetos como lenguaje de iniciación es importante tener en cuenta el entorno de la Universidad de Sevilla. En él, la generación de software de gestión es la principal ocupación de nuestros egresados, especialmente en lenguaje Java. Así, el hecho de basar todo el aprendizaje en un lenguaje útil para su futuro y además, más cercano a las tareas de análisis y diseño que los lenguajes de programación estructurada, se debe entender como la adaptación de la oferta académica a las necesidades reales de los alumnos.

Por otra parte, el software de gestión suele trabajar con sistemas de persistencia sobre los que básicamente se realizan cuatro operaciones con variantes (create, retrieve, update, delete; CRUD, en inglés). De esta forma, la aplicación de los esquemas de recorrido va en la línea de preparar al alumno para ser capaz de realizar estas operaciones con una probabilidad de error mínima, puesto que se aplica un patrón conocido desde sus inicios como programador.

Finalmente, C se mantiene con un porcentaje mínimo en el temario completo, principalmente para permitir a los alumnos progresar adecuadamente en otras asignaturas relacionadas con el software de sistema.

2.2. Fundamentos de la Programación

Con la experiencia acumulada, durante el curso 2009-10 se diseñó la asignatura Fundamentos de Programación (en adelante, FP), de primer curso, anual, de 12 créditos ECTS y común a los nuevos títulos de Grado que se pusieron en marcha en el curso académico 2010-11 [7]. La asignatura sigue la misma distribución de teoría y prácticas que sus antecesoras, pero con una hora menos de teoría a la semana. El temario presenta algunas novedades con respecto a IP1 e IP2. Señalamos esquemáticamente las más importantes:

- Se introduce una metodología clara y precisa para el diseño de tipos que engloba la definición de la interfaz y el desarrollo de su clase asociada.
- Se hace un uso exhaustivo de toda las facilidades del entorno de desarrollo Eclipse.
- Se realiza un uso mayoritario de los tipos List y Set de java.util para implementar colecciones de objetos. El tipo array se mantiene por razones históricas pero se le da un uso mínimo.
- Uso generalizado del *for* extendido de Java para recorrer arrays e iterables en detrimento de otras estructuras repetitivas como *for* clásico y *while*. Los alumnos deben conocer estas estructuras, pero utilizar aquellas que provoquen menos errores en tiempo de ejecución.
- Se siguen explicando los esquemas para las operaciones más comunes pero se formaliza su uso y se hace obligatorio para los problemas de la asignatura.

Didáctica en los estudios de ingeniería informática

- Se explican los tipos *Ordering*, *Predicate* y *Function* de la librería Guava de Google [6]. Estos tipos proporcionan los métodos necesarios para generalizar las operaciones de filtro, ordenación y transformación de objetos.
- En la segunda parte de la asignatura, los esquemas se sustituyen por la reutilización de la clase de utilidad *Iterables*, también de Guava. Esta clase proporciona métodos que implementan los esquemas básicos para filtrar, buscar, transformar o extraer información sobre objetos iterables.
- A petición de otros departamentos se ha ampliado el temario de C a cinco semanas al final del curso. Se hace especial hincapié en la diferente filosofía de programación, la estructura de un programa, el paso de parámetros, el uso de punteros y las operaciones de entrada/salida.

Los cambios en la asignatura siguen la misma línea de los últimos saltos anteriormente presentados. En este caso, el cambio más destacado es la introducción de Guava en el temario de la segunda parte de la asignatura. Tras superar el primer parcial, los alumnos deben dominar el uso de los esquemas básicos. Dichos esquemas son implementados en Guava por la empresa Google con la intención de que sus programadores reutilicen el código en lugar de hacer sus propias rutinas. Esta librería contiene los esquemas de recorrido sobre colecciones de uso más frecuente en desarrollos reales de Google y viene a confirmar que el camino que se inició en la Escuela de Informática hace diez años es el seguido en otras instituciones y empresas de reconocido prestigio.

3. Resultados

Históricamente, las asignaturas de primer año de programación en la Escuela Técnica Superior de Ingeniería Informática (ETSII) de la Universidad de Sevilla han tenido un alto grado de abandono y por lo tanto, un bajo porcentaje de aprobados respecto a matriculados. Para analizar las causas de este hecho, haría falta un estudio mucho más extenso que el presentado en este artículo, aunque una de las razones más plausibles puede ser la falta, en la mayor parte del alumnado, de formación específica en programación. Así pues, un aspecto importante a tener

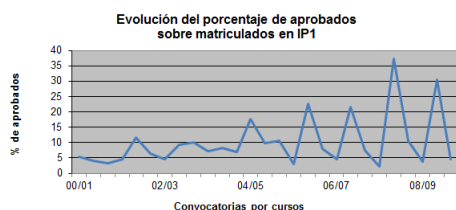


Figura 2: Resultados de IP1 durante los años 2000-09.

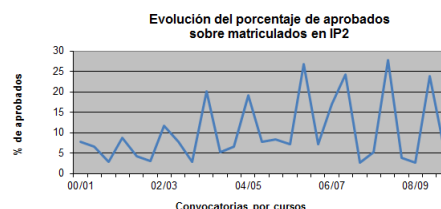


Figura 3: Resultados de IP2 durante los años 2000-09.

en cuenta es la reacción del alumnado ante los distintos cambios. En esta sección, mostramos distintas estadísticas recogidas a lo largo del tiempo que evalúan el rendimiento obtenido haciendo especial énfasis en el índice de aprobados sobre matriculados y presentados.

En lo que respecta a las antiguas asignaturas cuatrimestrales de IP, las Figuras 2 y 3 muestran su progresión en el número de aprobados respecto a matriculados a lo largo de las tres convocatorias anuales (febrero o junio, septiembre y diciembre) de los últimos diez años. Como puede observarse ambas figuras arrancan en números muy bajos de aprobados y a partir del curso 2004/05, ambas muestran una tendencia de crecimiento hasta el curso 2008/09 donde parece que los ratios se estabilizan. Un aspecto muy interesante es que los resultados muestran una mayor diferencia de porcentajes entre convocatorias del mismo curso. Teniendo en cuenta que los máximos se encuentran normalmente en la primera convocatoria, este hecho podría indicar que los alumnos de nuevo ingreso se adaptaron a las asignaturas de una manera mucho más rápida después de los cambios realizados. En cualquier caso, se puede observar que el trabajo de adaptación de las asignaturas hacia una metodología *objects-first* ha tenido un efecto positivo, confirmando los resultados de estudios similares como el de Gayo et al. [11].

Puesto que la asignatura FP sólo lleva en funcionamiento dos años, no es posible extraer conclusiones firmes de sus resultados pero sí una aproximación del rendimiento del alumnado en la nueva asignatura. Para ello, hemos seleccionado los datos de las primeras convocatorias de los cursos 2003/04, 2006/07 y 2008/09, por ser los cursos en los que cada etapa en las antiguas IP estuvo "madura". Sus resultados se han comparado con los obtenidos durante

el curso 2010/11 en la asignatura FP. El hecho de escoger la primera convocatoria para la comparativa es debido a que, como hemos comentado, esta convocatoria indica de manera sencilla la respuesta de los alumnos de nuevo ingreso a la asignatura. En los Cuadros 1, 2 y 3, se pueden observar los resultados para cada una de las ingenierías.

El primer dato interesante que se extrae de los cuadros es que no existen diferencias significativas entre las distintas titulaciones en lo que a resultados se refiere.

Por otro lado, es interesante ver que en la asignatura IP1, hay un gran salto entre los datos del 2003/04 y los recogidos en el curso 2006/07, no tanto así en la asignatura IP2. Este hecho se debe a que los alumnos que no superaban la primera asignatura cuatrimestral, solían abandonar la segunda. Como puede observarse, este problema se consiguió corregir con los cambios del curso 2004-05.

En lo que respecta a FP, puede parecer que dicha asignatura ha obtenido unos resultados ligeramente inferiores a los de las antiguas IP en su última etapa. Sin embargo, hay que tener en cuenta que los porcentajes mostrados para FP no incluyen alumnos repetidores, puesto que son los resultados del primer año en el que se impartió. Así pues, se puede concluir que los resultados son mejores a los de etapas anteriores puesto que sin alumnos "veteranos" se obtienen similares resultados.

4. Conclusiones

Este artículo ha presentado los cambios experimentados en los temarios de las asignaturas relacionadas con la introducción a la programación impartidas en las titulaciones de Informática de la Universidad de Sevilla en el período 2001-2011. El artículo

Asignatura	Curso	Matriculados	Presentados	Aprobados	% $\frac{\text{aprobados}}{\text{presentados}}$	% $\frac{\text{aprobados}}{\text{matriculados}}$
IP1	2003/04	576	159	44	27,67	7,64
	2006/07	414	124	85	68,55	20,53
	2008/09	258	124	98	79,03	37,98
IP2	2003/04	642	213	148	69,48	23,05
	2006/07	397	133	101	75,94	25,44
	2008/09	277	111	79	71,17	28,52
FP	2010/11	229	97	56	57,73	24,45

Cuadro 1: Resultados en la primera convocatoria de Ingeniería Informática y del Grado en Tecnologías Informáticas en las distintas etapas.

Asignatura	Curso	Matriculados	Presentados	Aprobados	% $\frac{\text{aprobados}}{\text{presentados}}$	% $\frac{\text{aprobados}}{\text{matriculados}}$
IP1	2003/04	345	92	22	23,91	6,38
	2006/07	347	116	88	75,86	25,36
	2008/09	264	99	74	74,75	28,03
IP2	2003/04	370	114	76	66,67	20,54
	2006/07	339	112	93	83,04	27,43
	2008/09	275	83	59	71,08	21,45
FP	2010/11	234	96	54	56,25	23,08

Cuadro 2: Resultados en la primera convocatoria de Ingeniería Técnica Informática de Gestión y del Grado en Ingeniería del Software en las distintas etapas.

Asignatura	Curso	Matriculados	Presentados	Aprobados	% $\frac{\text{aprobados}}{\text{presentados}}$	% $\frac{\text{aprobados}}{\text{matriculados}}$
IP1	2003/04	493	139	37	26,62	7,51
	2006/07	355	52	16	30,77	4,51
	2008/09	261	96	66	68,75	25,29
IP2	2003/04	538	147	96	65,31	17,84
	2006/07	446	131	93	70,99	20,85
	2008/09	306	102	67	65,69	21,9
FP	2010/11	122	39	27	69,23	22,13

Cuadro 3: Resultados en la primera convocatoria de Ingeniería Técnica Informática de Sistemas y del Grado en Ingeniería de Computadores en las distintas etapas.

lo ha hecho especial hincapié en las principales innovaciones introducidas a lo largo de dicho período: orientación a objetos en lugar de programación estructurada, uso de Java como primer lenguaje en el aprendizaje, introducción de esquemas y librerías (Guava), etc. Posteriormente, ha mostrado los resultados alcanzados por nuestros alumnos a lo largo de estos diez años. Dichos resultados demostraron que los sucesivos cambios han facilitado la introducción del alumnado (especialmente de nuevo ingreso) en la programación.

En lo que respecta al trabajo futuro, hay que tener en cuenta que este artículo se ha centrado exclusivamente en los cambios en el temario pero no ha evaluado el efecto que han tenido otros cambios relacionados con otros aspectos metodológicos (uso de plataformas online como WebCT [8], cambios en las distribuciones de horas de teoría y prácticas, distribución del profesorado, etc.) en la mejora de los resultados de nuestros alumnos aunque sí se puede descartar la influencia del profesorado puesto que el grueso del grupo se ha mantenido constante en los últimos diez años. En cualquier caso, será necesario, como trabajo futuro, discriminar con más detalle qué aspectos de la metodología han sido claves para ayudar a los alumnos a adquirir los conocimientos requeridos en las distintas etapas.

Agradecimientos

Queremos dar las gracias a todos los compañeros del departamento de Lenguajes y Sistemas Informáticos y en especial, a los que en algún momento han participado en la docencia de las asignaturas de introducción a la programación. Sin su inestimable ayuda, la experiencia descrita en este artículo no hubiese podido ser llevada a cabo. Especialmente, debemos acordarnos en estas líneas de Miguel Toro, y agradecerle su buen hacer dirigiendo nuestros pasos para dar un mejor servicio a nuestros alumnos y a nuestra comunidad.

Referencias

- [1] Aguilar, J. S., González, M., Riquelme, J. C., *Una propuesta para asignaturas de introducción a la programación*, Jenui, 2000.
- [2] Cordero, J.M., González, J.M., Romero, R. Martínez, *Introducción a la programación, un enfoque práctico*, Algaida, 1996.
- [3] Java API, <http://docs.oracle.com/javase/6/docs/api/>, Oracle, 2011.
- [4] Ministerio de Educación y Ciencias, *REAL DECRETO 55/2005, de 21 de enero*, BOE, 2005.
- [5] Weitzenfeld, A., *Ingeniería de Software Orientada a Objetos Con Java E Internet*, Thomson International, 2006.
- [6] Guava: Google Core Libraries for Java 1.5+, <http://code.google.com/p/guava-libraries/>, Google, 2012.
- [7] E.T.S. Ingeniería Informática, <http://www.informatica.us.es/index.php/estudios-y-titulaciones/grados>, Universidad de Sevilla, 2012.
- [8] Guía para el alumno de los cursos virtuales en WebCT, <http://www.uned.es/ca-campo-de-gibraltar/Alumnos/GUIA%20WEBCT.pdf>, UNED Campo de Gibraltar.
- [9] Sánchez Laguna, J. A., *iJava: un nuevo lenguaje para facilitar el paso del paradigma imperativo al orientado a objetos*, Jenui 2009, pages 107-114, 2007.
- [10] Gómez Albarrán, M., *Metodología basada en descomposición funcional y orientación a objetos en la introducción a la programación*, Jenui, 2002.
- [11] Gayo Avello, D., Cernuda del Río, A., Cueva Lovelle, J.M., Díaz Fondón, M., García Fuente, M.P.A., Redondo López, J.M., *Reflexiones y experiencias sobre la enseñanza de POO como único paradigma*, JENUI, 2003.
- [12] TIOBE, <http://www.tiobe.com/index.php>

