

EVALUACIÓN DE COMPETENCIAS DE PROGRAMACIÓN EN LA FORMACIÓN DEL INGENIERO EN SISTEMAS COMPUTACIONALES

A. A. Domínguez Martínez¹
Y. Mexicano Reyes²

RESUMEN

La formación de Ingenieros en Sistemas Computacionales basada en competencias implica que el egresado demuestre las habilidades de su perfil; es importante evaluar en su ciclo los avances intermedios que deben obtenerse para dicho perfil. Una competencia intermedia es la de programación, vital para cursar otras asignaturas del plan de estudios que reportan altos índices de reprobación. El objetivo es proporcionar un programa institucional que permita conocer el desarrollo de la competencia referida. El proyecto está dividido en etapas para atender cada una de las asignaturas. La primera etapa se enfoca en los alumnos que avanzan a Programación Orientada a Objetos. En nuestra primera aplicación se consideran alumnos a quienes se les aplica un examen diagnóstico. Con los resultados evaluados se establece el nivel de las competencias previas ubicando conjuntos de alumnos con carencias que deberán participar en acciones de academia para nivelar el alcance de la competencia y así continuar en forma exitosa en su avance escolar. Se obtiene además un conjunto de acciones de nivelación, el diseño de un sistema de monitoreo de competencias independiente y un sistema de mejora continua para la atención del grupo de materias. Entre las conclusiones obtenidas podemos mencionar el hecho de crear una cultura de demostración de competencias adquiridas por parte de cada alumno y una implementación de un programa de mejora continua para la formación de ingenieros con competencias sólidas.

ANTECEDENTES

La competencia de programación, en el paradigma de orientado a objetos en la carrera de Ingeniería en Sistemas Computacionales, constituye el pilar fundamental de la formación profesional en conjunto con las otras diversas áreas del conocimiento al egreso del nivel licenciatura. En las últimas décadas, la programación de computadoras se ha convertido en un área muy interesante y atractiva para los jóvenes que puede representar un peso importante al momento elegir su carrera. Sin embargo, el aprender a programar no es tarea fácil, muchos programadores principiantes se enfrentan a una amplia gama de dificultades y deficiencias, lo que ha causado que se considere a los cursos de programación como algo difícil. Generalmente se requiere de unos 10 años de experiencia para que un programador novato pueda convertirse en un programador experto (Winslow, 1996). Ante esta afirmación, el cursar una carrera profesional no incluye tiempo suficiente para alcanzar el nivel de experto. Aunque estos datos se obtuvieron cuando prevalecía la programación estructurada, con el cambio de paradigma los índices se mantienen similares.

El paradigma de programación orientado a objetos (POO) en nuestra institución se ha adoptado como el adecuado e indispensable para que los egresados puedan ser competitivos en el mercado laboral. Como menciona Izquierdo (2007), “la programación orientada a objetos debe considerarse por sí misma como una filosofía, una forma de pensar particular ya que representa un modelo de programación, con su teoría y su metodología, que conviene conocer y estudiar antes que nada para apropiarse de este paradigma”.

¹ Profesor de Tiempo Completo del Instituto Tecnológico de Saltillo. adominguez@itsaltillo.edu.mx.

² Profesora de Tiempo Completo del Instituto Tecnológico de Saltillo. ymexrey@gmail.com.

En el paradigma de POO es en el que deben desarrollarse las competencias del segundo curso de la línea de programación, pero debe implementarse y aplicarse considerando un contexto real; para ello se ha seleccionado usar la tecnología Java de Oracle como el lenguaje orientado a objetos a emplear. Esta decisión implica alinear la parte práctica de las asignaturas involucradas para aprovechar dicha tecnología.

Considerando el ambiente profesional de la tecnología Java un estudiante al egresar deberá cumplir con ser competente como Programador Java para la edición estándar, esto involucra tener un buen desempeño en:

- Analizar un problema de negocio para reconocer los objetos y las operaciones que forman los bloques integrantes del diseño de programas Java
- Declarar un método con argumentos y valores de retorno
- Declarar e inicializar variables
- Declarar e instanciar matrices y ArrayLists y poder iterar con ellas
- Definir el término
- Mostrar la sintaxis de programación Java
- Describir ejemplos de cómo se utiliza Java en aplicaciones, así como en productos de consumo
- Describir cómo se manejan los errores en un programa Java
- Describir las ventajas de utilizar un entorno de desarrollo integrado (IDE)
- Desarrollar clases y describir cómo declarar una clase
- Instanciar un objeto y utilizar de forma eficaz variables de referencia de objetos
- Enumerar y describir varias características clave de la tecnología Java
- Enumerar varios tipos de datos primitivos
- Utilizar la herencia para declarar y definir una subclase de una superclase existente
- Utilizar operadores, bucles y construcciones de decisión
- Escribir un programa Java simple que se compile y ejecute correctamente

Con el dominio de estos aspectos, un alumno puede certificarse como Programador Java y si obtiene un resultado acreditado en el examen “Java SE 7 Programmer I 1Z0-803”, será certificado con validez mundial. Debido a que la línea de programación está formada por cuatro asignaturas, es necesario delimitar que desempeños son propiciados por la materia de Fundamentos de Programación y en la primera etapa considerarlos para el diagnóstico del desarrollo de la competencia.

Un componente de este proyecto corresponde a la definición de los instrumentos para cuantificar el logro de competencia alcanzada por los alumnos en asignatura anterior. Por ello es importante definir qué instrumentos de evaluación serán seleccionados para tal fin. Villar y Alegre (2004) mencionan que la evaluación es el elemento de verificación que debe ser utilizado en el proceso de aprendizaje, que nos permite realizar mediciones en forma global del proceso; la evaluación nos permitirá conocer la eficacia del proceso llevado a cabo conforme lo planeado y en caso de ser necesario, la evaluación será una fuente de información imprescindible para reestructurar o modificar, según sea el caso, el conjunto de actividades que forman el proceso de aprendizaje con el objetivo de mejorar los resultados obtenidos en previas implementaciones.

Acorde a esta noción de la evaluación debe considerarse que técnicas de evaluación pudieran ser utilizadas para lograr el propósito establecido y se toman en cuenta las mencionadas por García-Beltrán et al. (2006):

- Exámenes de teoría. Se orientan a valorar la cantidad de información retenida por los alumnos.
- Exámenes de problemas. Permiten valorar la capacidad de relación y de razonamiento de los alumnos.
- Exámenes orales. Permiten el mayor grado de libertad posible al alumno en la respuesta, ya que normalmente se le pide que desarrolle por completo durante un tiempo determinado uno o varios temas teóricos o bien, que resuelva uno o varios problemas concretos.
- Realización de trabajos o proyectos. Puede convertirse en otra forma de evaluación consistente en el diseño y desarrollo de un trabajo o proyecto que puede entregarse durante o al final de la asignatura
- Pruebas de respuesta objetiva. Prueba cerrada o de tipo test tienen como una de sus principales ventajas el que se puedan diseñar para comprobar muchos tipos de objetivo.

Dada la versatilidad de las preguntas de respuesta objetiva que demuestran una ventaja con respecto a las otras técnicas, de acuerdo con las características del objeto de estudio, se decide construir un instrumento de evaluación del tipo de prueba cerrada.

La prueba cerrada puede ser empleada en general a título informativo y el alumno puede realizarla con carácter voluntario, con ello se eliminan inconvenientes como estrés o presión que pudieran afectar el desempeño en la realización de la prueba diagnóstica que se aplica.

Referente a la competencia considerada de programación, ¿es necesario esperar a que se cursen y aprueben todas las asignaturas involucradas para considerar la competencia como lograda? Como respuesta a esta interrogante se establece que en congruencia con los aspectos de una competencia, que si el logro de una competencia implica un proceso, cada asignatura representa un subproceso que puede caracterizarse con sus competencias específicas con todos los aspectos establecidos, considerando que puede lograrse su reforzamiento en los cursos restantes.

Así nuestro escenario de aplicación del proyecto en la primera etapa está concebido que debe desarrollarse la competencia de programación, empleando la tecnología Java posterior a la acreditación de la asignatura de Fundamentos de Programación y previo al arranque formal de la asignatura de Programación Orientada a Objetos. Con esta delimitación de la primera etapa se procede a considerar el listado completo de los desempeños de la competencia de programación y se establece seleccionar los que son propios de la asignatura y que deben ser atendidos al cursarla. Para la asignatura de Fundamentos de Programación se debe atender:

- 1) Escribir un programa Java simple que se compile y ejecute correctamente
- 2) Declarar un método con argumentos y valores de retorno
- 3) Declarar e inicializar variables
- 4) Mostrar la sintaxis de programación Java
- 5) Enumerar varios tipos de datos primitivos

- 6) Utilizar operadores, bucles y construcciones de decisión
- 7) Describir las ventajas de utilizar un entorno de desarrollo integrado (IDE).

Una vez obtenidas las competencias a desarrollar por la primera asignatura de la línea, se establece una clasificación para manejar el desarrollo del diagnóstico y contar así con un primer elemento de caracterización del programa. Se formularon las siguientes categorías:

- Saberes. Se incluirán las competencias relacionadas con el conocimiento léxico, sintáctico y semántico del Lenguaje de Programación.
- Capacidades. Aquí se toman en cuenta las aptitudes necesarias para la programación dentro del contexto de Java.

METODOLOGÍA

Construcción de instrumento diagnóstico.

Para la elaboración del instrumento de evaluación diagnóstica se establece desarrollarlo en dos partes: una primera que concentre el diagnóstico en la clase de elementos de competencia relacionados con saberes y una segunda centrada en los elementos de capacidades.

Como información adicional se solicita en cada instrumento de evaluación el nombre y semestre que cursa el alumno. Así el primer instrumento de evaluación diagnóstica se prepara con los siguientes reactivos:

- 1) Simple para operadores aritméticos.
- 2) Simple para palabras reservadas.
- 3) Simple para operadores relacionales.
- 4) Simple para clases predefinidas.
- 5) Simple para operadores lógicos.
- 6) Simple para tipos de datos primitivos.
- 7) Múltiple para expresiones.
- 8) Libre para instrucciones condicional.
- 9) Libre para instrucciones de iteración con n indicada.
- 10) Libre para instrucciones de iteración con n no indicada.

El segundo instrumento de evaluación diagnóstica se prepara con los siguientes reactivos:

- 1) Simple para representación de algoritmos.
- 2) Simple para interpretación de pseudocódigo.
- 3) Simple para interpretación de diagrama de flujo.
- 4) Libre para implementación de métodos basados en diagrama de flujo.
- 5) Libre para implementación de soluciones con uso de arreglos.

En el diseño del instrumento cada uno de los reactivos denominados simples, se configura como una pregunta de opción múltiple con solo tres opciones de respuesta, una correcta y dos incorrectas.

El reactivo denominado múltiple se construye con una batería de seis ejercicios de la misma naturaleza, cada uno incluye una fórmula clásica de ingeniería, un conjunto de valores para cada dato mencionado en la fórmula y se pide que el alumno codifique la fórmula en Java e indique el valor resultante de acuerdo con los valores proporcionados.

Finalmente, los reactivos denominados libres corresponden a reactivos en los cuáles, el alumno deberá escribir código Java en su modalidad de sentencia simple o programa Complet, para atender la solicitud del enunciado incluido en cada reactivo.

Con los instrumentos diseñados se procede a reproducirlos en cantidad suficiente para aplicarlo con los alumnos que están iniciando el curso de Programación Orientada a Objetos. Para ello, se informa a los alumnos con dos días de anticipación que deberán participar en la actividad de diagnóstico en el horario asignado de clase, un día se aplica el instrumento 1 relacionado con saberes y en un día posterior el instrumento 2 relacionado con capacidades. En ambos casos se otorga un máximo de 60 minutos para realizar la aplicación de cada instrumento por parte del docente a cargo de cada grupo.

Culminado el proceso de aplicación de los instrumentos, el docente procede a realizar la evaluación de ejemplar contestado, recopilando la información por alumno y por reactivo. En este caso los reactivos simples se califican como correcto o incorrecto. Los múltiples se califican como correcto o incorrecto por cada elemento de la batería dispuesta para tal fin. En el caso de los reactivos libre se establece una escala de tres niveles: respuesta nula o deficiente, respuesta con algunas imprecisiones y respuesta correcta.

DISCUSIÓN DE RESULTADOS

Con respecto al semestre de los alumnos diagnosticados se muestran los resultados en la Figura 1. Ahí se observa que un 73% de los alumnos sí pertenece al 2 semestre, ubicación que coincide con la retícula de la carrera. Se observa que en este caso, un 27% de alumnos en base a su semestre se muestran como irregulares por cursar la asignatura fuera del semestre habitual.



Figura 1. Distribución por semestre de los alumnos participantes

En el instrumento de saberes se agrupan los resultados de los primeros cinco reactivos como se muestra en la Figura 2. En ella se hace evidente que el tema de tipos de datos primitivos sí es dominado por la mayoría de los alumnos (97%). Los operadores aritméticos son los siguientes elementos con mayor dominio, 81% para aritméticos y 68% para operadores

relacionales y operadores lógicos. Para el caso de las palabras reservadas sólo la mitad de los alumnos proporciona respuestas correctas, un 54%. El tema crítico en este grupo es el de clases predefinidas, donde las respuestas correctas solo alcanzan un 41%.

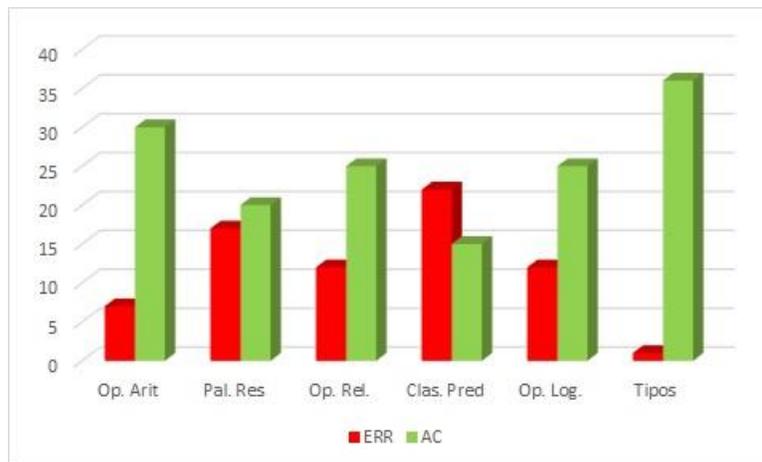


Figura 2. Dominio de elementos léxicos de Java

En el mismo instrumento en la batería de expresiones se muestra en la Figura 3 el desempeño obtenido. Se observa que en el caso de representar fórmulas básicas (73%) e implementar la precedencia correcta (78%) son buenos resultados. En el resto de las expresiones el desempeño correcto es muy pobre, en las operaciones con enteros únicamente alcanzo el 8%, por un 41% en operaciones con números reales y tan sólo un 19% en expresiones con exponentes y expresiones booleanas.

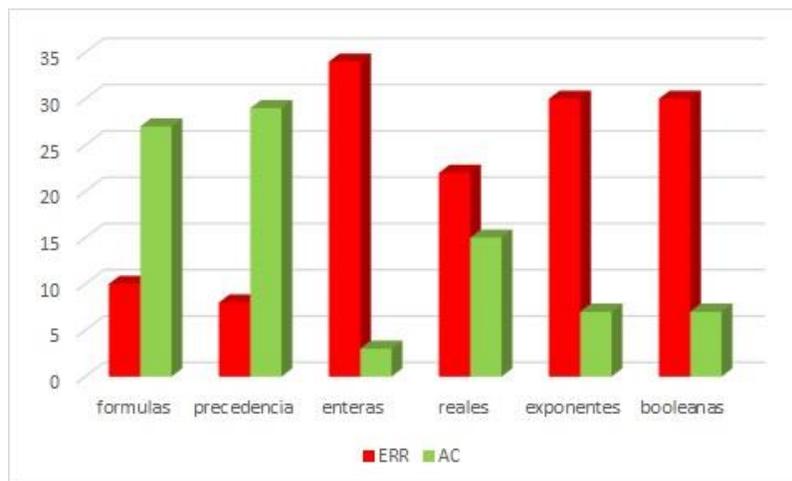


Figura 3. Dominio de expresiones en Java.

En la Figura 4 se muestran los resultados de los reactivos sobre instrucciones, en este caso hay tres niveles posibles, por orden de aparición, deficiente, impreciso y correcto. Solo el condicional muestra resultados mínimos aceptables, porque en el caso de los ciclos los resultados son muy pobres.

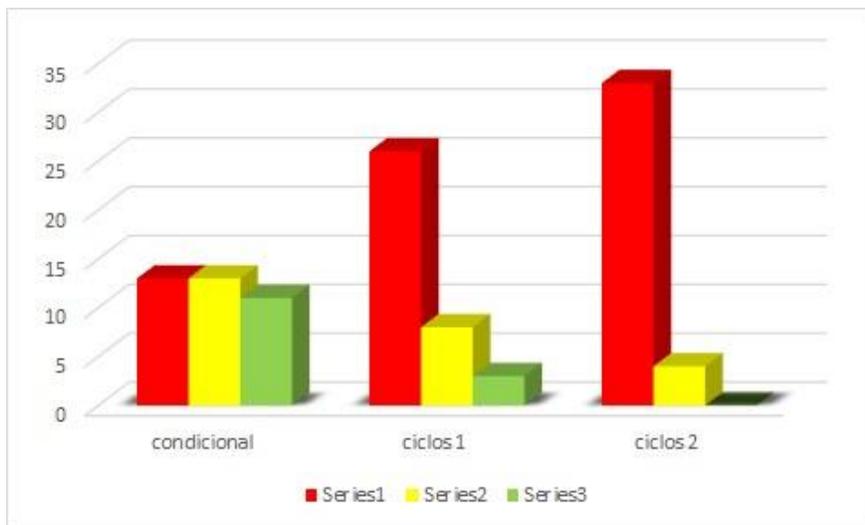


Figura 4. Dominio de instrucciones condicionales y de iteración.

Para el instrumento de capacidades se agrupan los reactivos en dos conjuntos. La Figura 5 nos muestra el desempeño con respecto a los algoritmos. El primero hace referencia a la forma de representación con porcentaje de respuestas correctas del 57%, el segundo que refiere a interpretar un algoritmo es un resultado excelente del 100% y el tercero que refiere a probar un algoritmo también es un buen resultado al alcanzar un 84% de respuestas correctas.

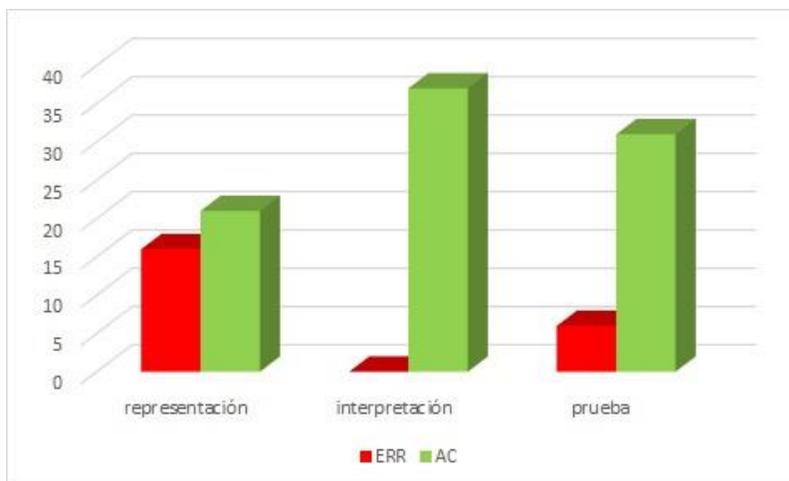


Figura 5. Dominio de algoritmos y sus representaciones.

En la Figura 6 se muestra el desempeño con los reactivos de instrucciones condicionales y de arreglos. En estos reactivos nuevamente se consideran tres posibles resultados. Así el reactivo de condicionales solo presenta un 3% correcto y un 68% acumulado si consideramos como aceptables las respuestas con imprecisiones. En el reactivo de arreglos los resultados son pobres, un 5% correcto y tan solo un 27% acumulado si consideramos también las respuestas con imprecisiones.

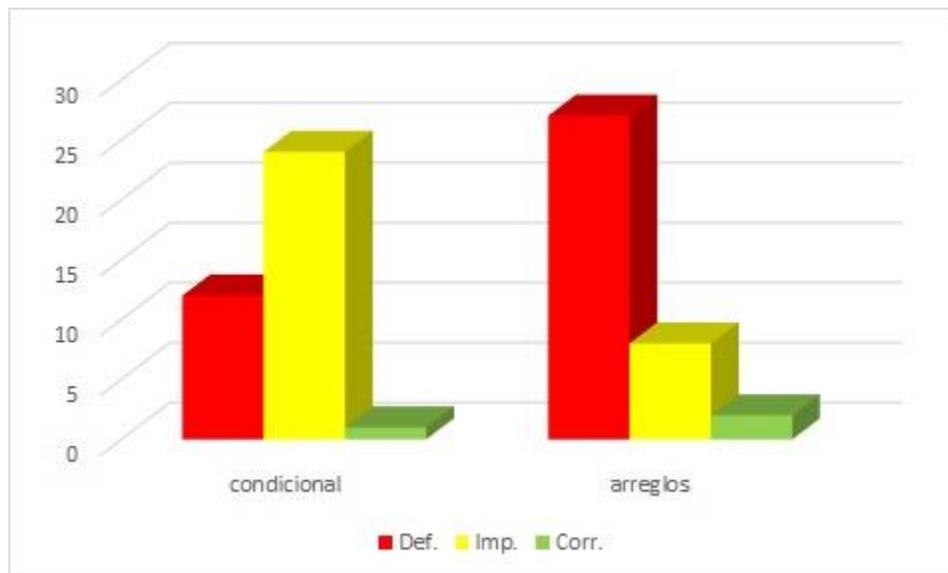


Figura 6. Dominio de condicionales y arreglos.

CONCLUSIONES

Con los resultados obtenidos en este diagnóstico se pueden establecer algunos hechos interesantes sobre el nivel de logro de competencia en la asignatura previa a Programación Orientada a Objetos.

En primer lugar, sobre la población que cursa la asignatura, se puede establecer que 3 de cada 4 se pueden considerar alumnos regulares, el resto evidencia una irregularidad de avance que deberá ser atendida por el departamento académico, ya que son alumnos con alto riesgo de abandono escolar, para ello se implementa un programa especial de tutorías para mejorar sus oportunidades.

Con respecto a los conocimientos sobre léxico se puede establecer que el dominio sobre tipos de datos es excelente, el de operadores aritméticos es bueno, el de operadores relacionales y lógicos es suficiente y sobre palabras reservadas y clases predefinidas son insuficientes, por lo que el docente de la asignatura de POO deberá incluir en sus actividades elementos de reforzamiento en dichos temas para mejorar el desempeño de los alumnos.

Par el manejo de expresiones en general se puede indicar buen desempeño en los procesos de abstracción de fórmulas básicas y su conversión a lenguaje Java. Pero también se denota deficiencia en el conocimiento de cómo opera el lenguaje al realizar operaciones con enteros, con reales, de cómo se implementan operaciones no básicas y las expresiones booleanas. Por ello, el docente debe incluir el reforzamiento de estos temas en sus actividades de aprendizaje.

Para las instrucciones condicionales y de ciclos el panorama es muy diferente en cada caso, si bien en las instruccionales, en conjuntar como aceptables las respuestas correctas con las de imprecisiones nos colocan en tener 2 de 3 correctas, éste es un punto de atención para las

actividades diseñadas por el docente, consideramos preferible solo tomar en cuenta las respuestas correctas. Ante este escenario se considera a nivel departamental ofrecer asesorías académicas sobre este tema y los otros identificados. Así también, entonces el tema de ciclos debe tener el mismo tratamiento mencionado para ciclos.

En alusión, con respecto al instrumento de capacidades, es grato identificar que la interpretación de algoritmos presenta muy buenos resultados, esto permite reconocer que las actividades de aprendizaje sobre este tema son efectivas, por lo que se debe retroalimentar a los docentes de la asignatura de Fundamentos de Programación de este hecho. En contraparte, resulta preocupante los resultados sobre el tema de arreglos, cuyos resultados de aprendizaje muestran evidencia muy pobre de éxito, en este caso se emprende acciones en dos directrices, los docentes de POO deberán agregar actividades extras cuyo objetivo sea retomar el tema de arreglos para los alumnos actuales. La otra acción es retroalimentar a los maestros de Fundamentos de Programación para que amplíen esfuerzos en las actividades de aprendizaje de arreglos.

Finalmente la formación de ingenieros debe ser una de las principales ocupaciones de una Institución de Educación Superior y por ello debe implementar acciones que le permitan conocer a lo largo del proceso, el grado de avance en el logro de las competencias de egreso que establece en sus programas de estudio, ya que con ello se contribuye al progreso de la sociedad de la economía y en general de nuestro país.

BIBLIOGRAFÍA

- García Beltrán, A. R. Martínez, R., J. Jaén y S. Tapia. La autoevaluación como actividad docente en entornos virtuales de aprendizaje/enseñanza. *Revista de Educación a Distancia 2006*. Obtenida el 24 de enero del 2017, de http://www.um.es/ead/red/M6/garcia_beltran.pdf.
- Izquierdo, L. R. *Introducción a la programación orientada a objetos*. Obtenida el 20 de enero de 2017, de <http://luis.izqui.org/resources/ProgOrientadaObjetos.pdf>.
- Villar, L.M. y Alegre O.M. (2004), *Manual para la excelencia en la enseñanza superior*, España Mc Graw-Hill.
- Winslow, L. E. "Programming pedagogy – a psychological overview," *SIGCSE Bulletin*, vol. 28, no. 3, Septiembre 1996