

# Análisis de la realidad del aprendizaje de programación en el Instituto Tecnológico de Orizaba

P. Quitl González <sup>1\*</sup>, C. Nava Arteaga <sup>1</sup>, M.A. Herrera Hernández<sup>2</sup>, D.N. Zilli Vázquez<sup>3</sup>, C. Valdez Jacinto<sup>3</sup>.

<sup>1</sup>Dep. de Sistemas y Computación, Tecnológico Nacional de México/I.T. Orizaba Av. Ote.9 No.852, Orizaba, Ver.

<sup>2</sup>Dep. de Ciencias Básicas, Tecnológico Nacional de México/I.T. Orizaba Av. Ote.9 No.852, Orizaba, Ver.

<sup>3</sup>Estudiantes de Ing. en Sistemas Computacionales, Tecnológico Nacional de México/I.T. Orizaba Av. Ote. 9 No.852, Orizaba, Ver.

\*[pquitl@yahoo.com.mx](mailto:pquitl@yahoo.com.mx)

**Área de participación:** Investigación Educativa

## Resumen

El trabajo presenta los resultados de un estudio que se desarrolló en el Instituto Tecnológico de Orizaba analizando el desempeño de los estudiantes en los Programas Académicos de Ingeniería Industrial, Ingeniería Electrónica e Ingeniería en Sistemas sobre la asignatura de programación. Se identificaron elementos intrínsecos de la materia, como son: Diseño de Algoritmos, Algoritmos Secuenciales y Algoritmos de Control; además se identificó la complejidad de los problemas que se plantean. Por otro lado, se analizó la dificultad del lenguaje de programación que se aplica en cada carrera. Los resultados demuestran que existe una relación directa entre el desempeño de los estudiantes y el nivel de dificultad de los problemas, aunado a la complejidad del lenguaje de programación.

**Palabras clave:** Programación, Algoritmos, Secuenciales, Control.

## Abstract

This paper presents the results of a study that took place at the Instituto Tecnológico de Orizaba analyzing the performance of the students of Industrial Engineering, Electronic Engineering and Systems Engineering on the subject programming. Intrinsic subject elements were identified such as: design of algorithms, sequential algorithms and Control algorithms. In addition, the complexity of the problems posed was identified. On the other hand, discussed the difficulty of the programming language that applies in each race. The results show that there is a direct relationship between the performance of the students and the level of difficulty of the problems, coupled with the of complexity of the programming language.

**Key words:** Programming, Algorithms, Sequential, Control.

## Introducción

En la actualidad, la enseñanza de la Asignatura de Programación, es un tema obligado en los cursos universitarios relativos al área de ingeniería (Barners & Kölling, 2013). El estudio de esta disciplina es importante, que reconocidos autores, han identificado el papel que desempeña este tema a nivel universitario; tal es el caso de Steve Jobs (Jobs, 1995), que en una entrevista televisiva afirmó lo siguiente: “Creo que todo el mundo en este país (E.U.) debería aprender a programar. Aprender a programar te enseña a pensar... es útil porque la programación te obliga a pensar de otra forma. Todo el mundo debería tomar uno o dos años de su vida aprendiendo a programar”, aseguraba el fundador de Apple. Otro caso es el de Luis Joyanes Aguilar (1993), autor español de libros especializados en el área de programación, menciona: “Para el estudiante universitario, las asignaturas de programación se convierten en tópicos clave en su formación, cuyo éxito o fracaso influirá en sus estudios.” (Joyanes Aguilar, 1993, p. XV). Efectivamente, la disciplina de programación es trascendental en las carreras de ingeniería pues contribuye al pensamiento lógico, siendo una habilidad en el ingenio humano que ha permitido el desarrollo tecnológico. En la mayoría de los estudiantes su iniciación en esta asignatura suele ser regularmente incierta (Kernighan & Pike, 2010).

La Programación es una asignatura común y fundamental para las diversas carreras de ingeniería que se ofrecen en el Instituto Tecnológico de Orizaba. Los contenidos, paradigmas y métodos docentes convergen en el desarrollo del pensamiento lógico.

En los programas educativos de Ingeniería Industrial, Ingeniería Electrónica e Ingeniería en Sistemas computacionales que se ofrecen en el Instituto, se imparte la asignatura de Programación que aborda desde el diseño de algoritmos por computadora, hasta la traducción de éstos hacia un lenguaje de programación.

Cabe mencionar que de cinco años a la fecha se ha observado una problemática común que se presenta en los estudiantes, que es la dificultad en el desarrollo adecuado y correcto de algoritmos de diversa índole, independientemente de la metodología que se use, esta problemática es recurrente, generando un alto índice de estudiantes que no comprenden la definición de los problemas que se pretenden solucionar, aunado a lo anterior, también exhiben la dificultad para identificar de manera eficiente los elementos implicados en la posible solución de problemas; todo esto se ve reflejado en los índices de aprobación de la Asignatura.

La problemática que se presenta no solo es exclusiva del Instituto Tecnológico de Orizaba, tal es su trascendencia que se han llevado a cabo diversos trabajos que estudian este suceso, dentro de los que se puede mencionar un **análisis comparativo de las herramientas de software actuales para el diseño de algoritmos en la programación** (Herrera, Nava, Quitl, & Herrera, 2018) que actualmente los autores realizan dentro del Instituto Tecnológico de Orizaba y que es el resultado de esta investigación, así como a Pérez del Instituto de Ciencias Básicas e Ingeniería de la Universidad Autónoma del Estado de Hidalgo (Pérez, 2014), presenta la preocupación de identificar las habilidades necesarias para el diseño de algoritmos, así como determinar los errores de diseño que frecuentemente incurren los estudiantes, al igual Ferreira de la Universidad Nacional de Río Cuarto, Argentina (Ferreira & Rojo, 2006), menciona que *“Estamos insertos en una realidad social y educativa que acusa la crisis de su propia identidad y que en cierta medida la deserción e índices de reprobación de estudiantes que se producen está estrechamente relacionada a esa realidad y escapa a las estrategias didácticas para lograr mejores rendimientos Académicos”*.

Ciertamente, la impartición de la asignatura de programación tiene sus bases en diversos paradigmas de programación (Vaca, 2012), independientemente del paradigma, se requiere de una metodología de algoritmos (*“Método de Resolución de Problemas por Computadora”*: **MRPC**) que contempla las siguientes etapas: Análisis del problema, Diseño de la solución, Desarrollo de la solución y Pruebas (Joyanes, 2008).

En la etapa de Análisis del Problema se identifican los datos que se requieren como entrada, se determinan los resultados esperados, así como establecer la forma en que los datos de entrada serán procesados. Así, en el Diseño de la Solución se crean las relaciones que existen entre los datos y la forma de procesarlos mediante patrones gráficos denominados diagramas de flujo o mediante patrones textuales denominados pseudocódigo.

En cuanto al Desarrollo de la Solución se establecen las operaciones sobre los datos a través de un lenguaje de programación; y, finalmente en la etapa de Pruebas se verifica que la solución planteada arroje los resultados planeados.

La metodología anterior facilita la tarea de solucionar problemas mediante la determinación de tres tipos de algoritmos: Secuenciales, Condicionales e iterativos.

Por consiguiente, un algoritmo es una combinación de operaciones lógicas dispuestas de forma organizada y finita para resolver un problema (Joyanes, 2008). Un algoritmo secuencial consiste en un conjunto de operaciones consecutivas que se suceden de forma serial, mientras que un algoritmo condicional evalúa el cumplimiento de una proposición lógica para tomar una decisión de qué alternativa seguir para conseguir los resultados esperados, por último, un algoritmo iterativo también evalúa una proposición lógica, sin embargo, se caracteriza por realizar tareas repetitivas.

Ahora bien, en cuanto a los lenguajes de programación los planes de estudio de las carreras observadas se plantea el uso de lenguaje “C” en los programas educativos de Ingeniería Industrial e Ingeniería Electrónica mientras que para Ingeniería en Sistemas se plantea el uso del lenguaje “Java”.

El lenguaje “C” fue creado en la década de los 70’s, es un lenguaje de propósito general, comúnmente usado como plataforma de desarrollo de software de base, por generar códigos eficientes debido a su nivel de abstracción cercano al código máquina, se basa en un compilador, que cuenta con un repertorio de palabras reservadas relativamente pequeño (Kernighan & Ritchie, 1991) ; en cuanto al lenguaje “Java” éste es orientado a objetos, permite de forma natural expresar la funcionalidad de las operaciones de manera encapsulada, es un

lenguaje interpretado, su nivel de abstracción es complejo (Deitel & Deitel, 2012), es rápido, seguro y fiable; permite generar aplicaciones de software desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet; ambos lenguajes brindan un conjunto de bibliotecas con funcionalidades para reutilizar.

Tomando en consideración lo anteriormente expuesto, se llevó a cabo una investigación que ayude a determinar qué elementos intrínsecos de la asignatura representan dificultad de aprendizaje en los estudiantes, para contar con una base de resultados reales que permitan en un futuro cercano establecer estrategias didácticas de mejora en el proceso de comprensión y desarrollo de habilidades en la asignatura de programación.

## Metodología

Para abordar la problemática señalada, se realizó un estudio descriptivo, no experimental y longitudinal, sobre la situación que existe en el aprovechamiento de la signatura de programación en programas académicos de Ingeniería Industrial, Ingeniería Electrónica e Ingeniería en Sistema para los semestres correspondientes al periodo de 2015-2017. Para tal fin se consultó el desempeño de los estudiantes mediante información proporcionada por los docentes que impartieron las asignaturas, reportando un total de 1270 estudiantes atendidos.

Los siguientes incisos describen detalladamente como se aplicó la investigación:

### a) Elementos intrínsecos de la asignatura para el estudio

En cada una de los programas educativos la Asignatura lleva la siguiente identificación: “Algoritmos y Lenguajes de Programación” en Ingeniería Industrial, “Programación Estructurada” en Ingeniería Electrónica y “Fundamentos de Programación” para Ingeniería en Sistemas. No obstante, en la realidad los contenidos persiguen el mismo objetivo, que es desarrollar las habilidades lógicas para la solución de problemas por computadora.

Por tal motivo se distinguieron tres elementos intrínsecos que distinguen a las asignaturas y se catalogaron en (Tabla 1): Diseño de Algoritmos, Algoritmos Secuenciales y Algoritmos de Control (abarcan algoritmos condicionales y cíclicos), mediante lo cual se persigue determinar la complejidad a la que se enfrentan los estudiantes para desarrollar las habilidades básicas de la programación de computadoras.

El objetivo del Diseño de algoritmos es desarrollar la habilidad de distinguir los elementos que forman parte de problemas planteados y proponer soluciones mediante el uso de pseudocódigo, se realiza una traducción de la solución diseñada hacia un lenguaje representativo independiente a cualquier lenguaje de programación. El pseudocódigo omite detalles que no son esenciales para la comprensión del algoritmo, tales como declaraciones de variables, código específico del sistema y algunas subrutinas.

Esta es una de las primeras actividades que contempla la metodología **MRPC** (Método de Resolución de Problemas por Computadora). Posterior a esta actividad, se procede al manejo de un lenguaje de programación, a través de un proceso de abstracción que concurre en el planteamiento de la solución, dónde los problemas que se exponen evolucionan de acuerdo a su grado de complejidad, en los cuales se aplican estructuras para su resolución como algoritmos secuenciales y estructuras de control.

**Tabla 1, Categorías de los elementos intrínsecos de la asignatura de programación en los programas académicos de Ingeniería Industrial, Ingeniería Electrónica e Ingeniería en Sistemas del Instituto Tecnológico de Orizaba.**

Categoría	Objetivo didáctico
Diseño de algoritmos	Distinguir los elementos que forman parte de problemas planteados y proponer soluciones mediante el uso de pseudocódigo
Algoritmos secuenciales	Desarrollar soluciones de problemas con un nivel de complejidad que requiera procesos mentales consecutivos, así como el uso de datos correspondientes, aplicando un lenguaje de programación.
Algoritmos de control	Construir soluciones de problemas con un nivel de complejidad que requiera procesos mentales concurrentes mediante la toma de decisiones, así como la aplicación de procesos repetitivos, aplicando un lenguaje de programación.

En cuanto a los lenguajes de programación que se aplican en la Asignatura, de acuerdo a los programas académicos analizados, se comparan sus relaciones con las categorías identificadas, mediante la Tabla 2.

**Tabla 2, Categorías de los elementos intrínsecos de la asignatura de programación en los programas académicos de Ingeniería Industrial, Ingeniería Electrónica e Ingeniería en Sistemas del Instituto Tecnológico de Orizaba.**

Categoría	Ingeniería Industrial	Ingeniería Electrónica	Ingeniería en Sistemas
Diseño de algoritmos	Pseudocódigo	Pseudocódigo	Pseudocódigo
Algoritmos secuenciales	Lenguaje C	Lenguaje C	Lenguaje Java
Algoritmos de control	Lenguaje C.	Lenguaje C.	Lenguaje Java

**b) Lenguajes de programación que se usan en las asignaturas del estudio**

Referente a las características de los lenguajes de programación que representan su complejidad de uso, se determinaron criterios tales como: paradigma de programación, tipo de traductor en que se implementa, necesidad de manejo de tipos de datos para la definición de variables y número de tipo de datos básicos, rigurosidad de su sintaxis, dificultad en la estructura del cuerpo del programa, y disponibilidad de documentación. Ver tabla 3.

**Tabla 3, Criterios que representan la complejidad de uso de los lenguajes de programación.**

Lenguaje	Características	Complejidad						
		PP	TT	NTD/C	NPRv	RSTXS	DECP	DD
Pseudocódigo	Es lenguaje de especificación de algoritmos. Permite generar una versión de la solución concentrándose en las operaciones sin describir detalles de los datos del problema, dado que el pseudocódigo tiene que traducirse posteriormente a un lenguaje de programación.	no	No	no	11	baja	baja	No
Lenguaje C	Uno de los objetivos de diseño del lenguaje C es que sólo sean necesarias unas pocas instrucciones para crear un programa. El lenguaje C está disponible en un amplio abanico de plataformas, se usa en diversos contextos de la programación.	Estructurado	Compilador	Si/5	32	media	media	si
Lenguaje Java	Es un lenguaje orientado a objetos, de una plataforma independiente. Se pueden realizar distintos tipo de aplicaciones de software para distintas plataformas.	Objetos	Compilador Interprete	Si/8	50	alta	alta	si

**Nomenclatura**

PP	TT	NTD/C	NPRv	RS	DECP	DD
Paradigma de Programación	Tipo de Traductor	Necesidad de Tipos d datos/ Cantidad	Número de Palabras Reservadas	Rigurosidad en la sintaxis	Dificultad en la estructura del cuerpo del programa,	Disponibilidad de Documentación

Una vez distinguidas las categorías de las asignaturas que se incorporaron al estudio, se procedió a realizar un conteo de los estudiantes que se atendieron, reportando un total de 1270.

En cuanto al tipo de problemas que se plantean de acuerdo a los programas académicos, los maestros ofrecieron la siguiente información que se concentra en la Tabla 4.

**Tabla 4. Porcentaje de tipo de problemas planteados a cada programa académico.**

Programa Académico	%tipo de problemas planteados	
	Administrativos	Físico-Matemáticas
Ingeniería Industrial	60	40
Ingeniería en Sistemas	40	60
Ingeniería Electrónica	40	60

**c) Población**

En este estudio se analizan los programas académicos de Ingeniería industrial, Ingeniería Electrónica e Ingeniería en Sistemas, tomando en cuenta los semestres que comprenden los años de 2015 al 2017, obteniendo una población de estudio de 1270 estudiantes, y tomando la fórmula que se presenta en la Figura 1, se calculó una muestra de 451 estudiantes, con un nivel de confianza del 96.5% y margen de error del 3.5%, considerando la misma como suficiente y representativa para observar el desempeño de los mismos.

$n = \frac{N * Z^2 * \sigma^2}{e^2(N - 1) + Z^2 \sigma^2}$	$n = \frac{1270 * (1.82)^2 * (0.5)^2}{(0.035)^2(1269) + (1.82)^2 * (0.5)^2} = 451 \text{ alumnos}$
--	--

**Figura 1. Fórmula para el cálculo de la muestra.**

Donde:

n=Tamaño de la muestra

N=Tamaño de la población

Z=Estándar para el nivel de confianza distribución normal

$\sigma^2$ =Desviación estándar (0.5)

e= Límite aceptable de error

**Resultados y discusión**

Después de la recolección de datos, se concentraron los porcentajes de aprobación de los estudiantes considerados en este estudio, así como de los programas educativos y las categorías sujetas al análisis, ver Tabla 5.

**Tabla 5. Concentrado del número de estudiantes aprobados por categoría de aprendizaje, recabada en los años 2015-2017.**

Programa Académico	Categorías			Semestre
	Diseño de algoritmos	Algoritmos secuenciales	Algoritmos de control	
A	86.66	83.33	56.66	E-J 2015
B	53.333	33.33	46.66	
C	72.00	44.00	48.00	
A	48.27	51.72	65.51	A-D 2015
B	36.84	36.84	5.26	
C	83.33	55.55	38.88	
A	89.65	89.65	75.86	E-J 2016
B	77.27	77.27	36.36	
C	50.00	50.00	52.77	

**Tabla 5 continuación. Concentrado del número de estudiantes aprobados por categoría de aprendizaje, recabada en los años 2015-2017.**

Programa Académico	Categorías			Semestre
	Diseño de algoritmos	Algoritmos secuenciales	Algoritmos de control	
A	50.00	54.54	50.00	A-D 2016
B	51.42	25.71	25.71	
C	23.80	57.144	66.66	
A	85.18	70.37	62.96	E-J 2017
B	48.14	18.51	18.51	
C	48.38	41.93	38.70	
A	66.66	55.55	62.96	A-D 2017
B	59.25	63.15	26.31	
C	78.94	78.94	52.63	

**Nomenclatura**

E-J	Enero-Junio
A-D	Agosto-Diciembre
A	Ingeniería. Industrial
B	Ingeniería Sistemas Computacionales
C	Ingeniería Electrónica

En la Tabla 6 se agruparon los datos del periodo 2015-2017. Obteniendo el promedio por programa académico y categorías en porcentajes de aprobación.

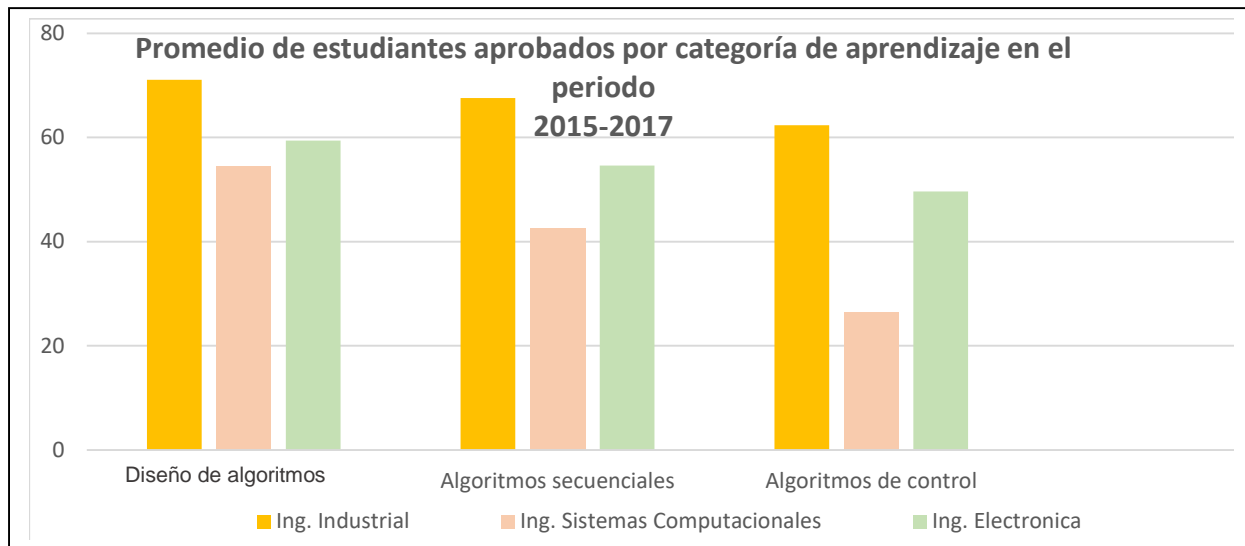
**Tabla 6. Resumen del promedio de los estudiantes aprobados por programa académico.**

Programa Académico	Categorías		
	Diseño de algoritmos	Algoritmos secuenciales	Algoritmos de control
Ing. Industrial	71.07	67.53	62.32
Ing. Sistemas Computacionales	54.38	42.47	26.47
Ing. Electrónica	59.41	54.59	49.61

De los datos anteriores se observa su comportamiento en la Gráfica 1:

Considerando los datos presentados, se puede distinguir que los estudiantes que cursan el programa académico de Ingeniería industrial presentan un mejor desempeño en comparación de los otros dos programas académicos, a su vez Ingeniería Electrónica presenta mejor rendimiento que Ingeniería en Sistemas.

Esto refleja una situación que llama la atención, pues los estudiantes de Ingeniería en Sistemas ejercerán una profesión dónde un aspecto de su perfil profesional es el desarrollo de soluciones mediante la elaboración de sistemas de software basados en uso adecuado de lenguajes de programación poniendo en práctica su razonamiento lógico.



**Gráfica 1, Desempeño los estudiantes aprobados por categoría identificada.**

## Trabajo a futuro

Si bien es cierto, que los resultados vislumbran que uno de los factores que afectan a la asimilación de los elementos intrínsecos de la Asignatura de Programación es la complejidad del lenguaje de programación que se utiliza en cada programa académico, también se sabe que el proceso enseñanza-aprendizaje es complejo, y que en este proceso existen otros factores externos que influyen en el desempeño de los estudiantes.

Por consiguiente, una de las metas es continuar analizando el rendimiento de los estudiantes de los programas académicos observados, además de ampliar el estudio incluyendo factores externos que pudieran influir en el desarrollo de las habilidades de los estudiantes.

Adicionalmente, en la actualidad se está trabajando en el desarrollo de una plataforma Web didáctica que concentre y ofrezca diversas herramientas de software que apoyen a la asignatura de programación.

## Conclusiones

Considerando los resultados del estudio, se ha identificado que los estudiantes que cursan Ingeniería industrial presentan mejor desempeño en comparación de los otros dos programas académicos, sin embargo, este criterio no es significativo pues la complejidad de problemas que se desarrollan en esta carrera es menor en comparación de los otros dos.

Un aspecto interesante en el análisis, es que Ingeniería Electrónica presenta un mejor desempeño en comparación con Ingeniería en Sistemas, pues la complejidad de los problemas que se plantean es la misma, sin embargo, la diferencia estriba en la complejidad del lenguaje de programación que se utiliza.

Con respecto al análisis anterior, este estudio ha permitido vislumbrar que uno de los factores que afectan la asimilación de la signatura de programación es la complejidad del lenguaje de programación que se debe utilizar en cada programa académico.

Definitivamente, este hallazgo invita a las autoridades que diseñan o aplican los planes de estudio, a reflexionar, sobre sobre el de Ingeniería en Sistemas, tomando en cuenta que el objetivo de la asignatura de Fundamentos de Programación es el desarrollo del razonamiento lógico del estudiante, se recomienda cambiar el lenguaje sobre el cual se ha trabajado hasta la fecha y considerar otros lenguajes actuales que requieren menos complejidad de uso.

## Agradecimientos

Agradecemos al Tecnológico Nacional de México por el apoyo brindado a éste proyecto de Investigación del cuál surge este trabajo, así como al Instituto Tecnológico de Orizaba por la oportunidad de contribuir al Coloquio de Investigación Multidisciplinaria CIM-Orizaba-2018, de igual forma a los estudiantes que fueron parte del equipo de trabajo y pusieron no sólo tiempo, sino interés y amor por esta labor.

## Referencias

1. Barners, D., & Kölling, M. (2013). *Programación orientada a objetos con Java usando BlueJ* (5 ed.). Madrid España: Pearson, Printece Hall.
2. Cairo, O. (2014). *Metodología de la Programación, 3a Ed.* Mexico, D.F.: AlfaOmega.
3. Deitel, P., & Deitel, H. (2012). *Java, Cómo Programar.* México: Person.
4. Ferreira, A., & Rojo, G. (2006). Enseñanza de la programación. *Revista Iberoamericana de tecnología en Educación y Educación en Tecnología.*
5. Flores, J. (2014 de marzo de 2014). Método de las 6'D, Modelamiento-Algoritmo-Programación - Tomo I. I.
6. Herrera, M., Nava, C., Quitl, P., & Herrera, A. (2018). Enfoques y herramientas de software para el diseño de algoritmos en la programación estructurada. *5o Foro UNIVO. Conocimiento en Red, 1*, 50-58.
7. Jobs, S. (1995). The Triumph of the Nerds: The Rise of Accidental Empires. (R. Crngely, Entrevistador) Obtenido de <https://youtu.be/mCDkxUbaICw>
8. Joyanes, L. (2008). *Fundamentos de Programación: Algoritmos y Estructuras de Datos.* Madrid, Esp.: McGraw-Hill Interamericana.
9. Kernighan, B., & Pike, R. (2010). *La práctica de la programación.* Printice Hall.
10. Kernighan, B., & Ritchie, D. (1991). *El Lenguaje de Programación C.* D.F, México: Person Education.
11. Pérez, I. (2014). Estudio de la problemática presente en el diseño de algoritmos por computadora.
12. Vaca, C. (2012). Paradigmas de Programación. Valladolid, España. Recuperado el 14 de diciembre de 2017, de <https://www.infor.uva.es/~cvaca/asigs/docpar/intro.pdf>