

# Aplicación del método inductivo modificado (MIM) para incrementar los niveles de aprendizaje en los cursos de programación de los alumnos de primer ciclo de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego de Trujillo

Application of modified inductive method (MIM) to increase the learning levels in programming courses of the first cycle engineering faculty students from the private University Antenor Orrego de Trujillo

Walter Aurelio Lazo-Aguirre<sup>1</sup>

Recibido: 24 de julio de 2018  
Aceptado: 21 de agosto de 2018

## RESUMEN

El presente trabajo tiene como propósito investigar en qué medida la aplicación del método inductivo modificado (MIM) incrementa los niveles de aprendizaje en los cursos de programación de los alumnos de primer ciclo de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego de Trujillo. La investigación es de tipo cuasi-experimental, para lo cual se utilizó una muestra de 100 alumnos. Como instrumentos se empleó la prueba objetiva para evaluar los contenidos conceptuales, se aplicó pretest y postest a ambos grupos de control y experimental para evaluar el rendimiento académico respectivo, lo mismo que otros instrumentos especialmente diseñados para el recojo de la información. Los resultados confirman que la aplicación del método MIM incrementa significativamente los niveles de aprendizaje en los cursos de programación de los alumnos de primer ciclo de la Facultad de Ingeniería, lo cual se refleja en la mejora de su rendimiento académico. Al realizar las pruebas estadísticas se observó en el postest que la nota promedio del grupo experimental fue de 11.86 que supera en forma significativa la nota promedio del grupo de control que fue de 9.33, obteniéndose un  $T_c = 1.798 > T_{tabular} = 1.684$ , para un  $\alpha = 0.05$

**Palabras claves:** Método, aprendizaje, programación, inductivo

<sup>1</sup> Doctor en educación - Universidad Privada Antenor Orrego

## ABSTRACT

The purpose of this paper is to investigate to what extent the application of the Modified Inductive Method (MIM) increases the learning levels in the programming courses of the engineering faculty students from the private University Antenor Orrego de Trujillo. Trujillo. This is a quasi-experimental research, for which a sample of 100 students was used. As instruments the objective test was used to evaluate the conceptual contents, pretest and posttest were applied to both control and experimental groups, to evaluate the respective academic performance, the same than other instruments specially designed to collect information. The results confirm that the application of the MIM method significantly increases the learning levels programming courses of the students of the initial cycles form the Engineering Faculty, which is reflected in the improvement of their academic performance. Due to the statistical tests, it was observed in the posttest that the average of the experimental group was 11.86 points that significantly exceeds the average of the control group which was 9.33 points, obtaining a  $T_c = 1.798 > T_{tabular} = 1.684$ , for  $\alpha = 0.05$

**Keywords:** Method, learning, programming, inductive

## INTRODUCCIÓN

Actualmente, la programación es una materia esencial en cualquier carrera de ingeniería, especialmente en ingeniería de computación, sistemas y ramas afines y en las que se estudia programación desde los primeros semestres de la carrera. Su impacto es muy importante en la mayoría de las asignaturas de la carrera y en el campo profesional del egresado.

Sin embargo, se observa que existe un bajo rendimiento académico en los alumnos en los cursos iniciales de programación de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego de Trujillo y esto representa un problema que se mantiene en forma persistente.

El análisis del rendimiento académico de los alumnos en los cursos de esta área, observado a través de las cantidades de alumnos aprobados, desaprobados e inhabilitados, sugiere que posiblemente es necesario buscar alguna metodología que permita mejorar los niveles de aprendizaje de los alumnos.

El propósito del presente trabajo es aplicar una metodología específica a los alumnos de los cursos de programación de primer ciclo, de la Facultad de Ingeniería de la Universidad Privada Antenor

Orrego de Trujillo, para tratar de incrementar significativamente sus niveles de aprendizaje de programación, lo que se debe reflejar en la mejora de su rendimiento académico, la disminución del alto número de desaprobados que se tiene en estos cursos y elevar el nivel de conocimientos de los alumnos que les permita avanzar en su carrera de manera exitosa.

El planteamiento de una metodología para mejorar el rendimiento académico en los cursos de programación de los alumnos de primer ciclo, de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego de Trujillo es importante por cuanto se encuentra enmarcado dentro de la problemática educacional de la conducción del proceso de enseñanza-aprendizaje en el contexto de la educación universitaria.

El constructivismo indica que el conocimiento es construido activamente por el estudiante y no es incorporado pasivamente de libros de textos y lecturas. Como la construcción del conocimiento se realiza en base a hechos, ideas y creencias que ya posee el estudiante, cada uno va a construir una versión del conocimiento. Se considera que enseñar con técnicas derivadas de esta teoría es más exitoso que enseñar con técnicas tradicionales

(conductistas) porque requieren explícitamente un proceso de construcción del conocimiento (Azpilicueta, J, 2003)

La "construcción de los aprendizajes" se refiere a que para aprender el alumno elabora nuevos aprendizajes en base a sus conocimientos previos. En este proceso el profesor tiene un rol de mediador y debe hacer que el alumno investigue, compare y exponga sus ideas.

El rendimiento académico "es una forma de expresión del proceso de aprendizaje entendido como un cambio de conducta, es decir, como aparición de habilidades intelectuales (lógicas) o estrategias cognitivas" (Gagné, 1975).

Rendimiento académico es una medida de las capacidades respondientes o indicativas que manifiestan, en forma estimativa, lo que una persona ha aprendido como consecuencia de un proceso de formación o instrucción (Pizarro, 1985)

"Rendimiento académico es la manifestación del aprendizaje logrado, lo cual es factible de medir y valorar" (Capella, 1990)

Fowler (2002) plantea que la idea principal de la taxonomía de Bloom es que lo que los educadores quieren que sepan los alumnos puede ser organizado en una jerarquía desde lo menos complejo hacia lo más complejo.

La taxonomía de Bloom divide en tres dominios la forma en que las personas aprenden. Uno de esos dominios es el cognitivo, que hace énfasis en los desempeños intelectuales de las personas. Este dominio a su vez está dividido en seis categorías o niveles: Conocimiento, comprensión, aplicación, análisis, síntesis y evaluación.

Se puede decir que uno de los dominios en los que tiene aplicación el paradigma de colaboración es en la construcción de programas de computadora o programación. En las asignaturas de Introducción a la Programación de las diferentes enseñanzas universitarias españolas se plantean problemas que los alumnos tienen que resolver en grupos de dos. El uso de esta aproximación en las clases de Introducción a la Programación mejora el aprendizaje del alumno, aumenta su satisfacción, reduce la frustración compartida de los estudiantes y disminuye el número de defectos de los programas. (Bravo y otros 2009)

En "Constructivismo en la educación de las

Ciencias de la Computación. Una propuesta de enseñanza- aprendizaje en aula virtual basada en resolución de problemas", Azpilicueta y Ledesma (2009) establecen que el constructivismo es una teoría de aprendizaje que indica cómo los estudiantes "construyen" el conocimiento en lugar de recibirlo y acumularlo tal como en general lo enseñan los profesores. Este trabajo analiza el constructivismo y muestra la forma en la cual esta teoría puede proveer las bases teóricas para un aprendizaje activo de los alumnos, de los lenguajes de programación, en un ambiente de resolución de problemas en aula virtual.

En "Estrategias para la enseñanza de la programación", Moroni y Peñas (2009) indican que la complejidad de los programas que se desarrollan actualmente produce la necesidad de iniciar a los alumnos en un camino que los conduzca a utilizar efectivas técnicas de programación. Es importante para ello poner énfasis en el diseño previo. Una estrategia valedera es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema.

Una teoría del aprendizaje explica y predice cómo aprende el ser humano, sintetizando el conocimiento elaborado por diferentes autores. Es así como todas las teorías, desde una perspectiva general, contribuyen al conocimiento y proporcionan fundamentos explicativos desde diferentes enfoques y en distintos aspectos. Sin embargo, es necesario hacer la distinción entre teorías del aprendizaje y teorías de la didáctica de la educación. (Stockholm Challenge Award, 2009).

Se podría considerar que no existe una teoría que contenga todo el conocimiento acumulado para explicar el aprendizaje. Todas consisten en aproximaciones incompletas, limitadas, de representaciones de los fenómenos. Con ello es posible entender que en la realidad se puede actuar aplicando conceptos de una y de otra teoría dependiendo de las situaciones y los propósitos perseguidos. (Stockholm Challenge Award, 2009).

La didáctica como disciplina se caracteriza por su carácter formativo y su aporte a través de modelos, enfoques y valores intelectuales más adecuados que permitan organizar decisiones educativas de modo que el pensamiento se decida por la instrucción y el desarrollo reflexivo del saber cultural y artístico (Medina y Salvador, 2009)

Se quiere investigar la mejora del rendimiento en el proceso de aprendizaje de los alumnos de

los cursos de programación de los ciclos iniciales de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego, con la aplicación de una metodología que se adecue a las características de estos cursos.

## MATERIAL Y MÉTODO

El estudio corresponde a una investigación aplicada de tipo cuasi-experimental con grupo de control no equivalente. La población de estudio estuvo constituida por los alumnos de primer ciclo de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego, matriculados en los cursos de programación. En la selección de la muestra se consideró a los alumnos del curso de Programación de I ciclo de la Escuela Profesional de Ingeniería de Computación y Sistemas, donde se eligió grupos de 21 alumnos para que pertenezcan al grupo experimental y al grupo de control.

Los criterios de selección fueron considerar alumnos matriculados en forma regular en todos los cursos del ciclo correspondiente al curso de Programación, registrados en primera matrícula. Los participantes que respondieron la totalidad de los reactivos propuestos. Alumnos hombres y mujeres sin límite de edad.

Se excluyeron alumnos que no cumplieron lo indicado en el párrafo anterior.

## MÉTODO

Se dispuso de dos grupos de alumnos, se evaluó a ambos en la variable dependiente, luego a uno de ellos se le aplicó el tratamiento experimental y el otro siguió con las tareas o actividades rutinarias.

### Variable independiente (VI)

Método inductivo modificado (MIM)

### Variable dependiente (VD)

Niveles de aprendizaje en los cursos de programación de los alumnos de los ciclos iniciales de programación en la Facultad de Ingeniería de la Universidad Privada Antenor Orrego.

### Instrumentos de recolección de datos

Para recolectar datos se utilizaron pruebas

objetivas y pruebas de comprobación validadas por un experto y también por los alumnos. Estas pruebas objetivas y pruebas de comprobación se aplicaron como pretest y postest, para ir comparando el desempeño de los alumnos en el tiempo y hacer las correcciones y retroalimentación que sean necesarias.

1. Prueba objetiva que evaluó los contenidos conceptuales globales del curso de Programación, antes (pretest) y después (postest) de la aplicación de la propuesta metodológica.
2. Prueba de madurez mental de California, serie intermedia 1950-S, para comprobar que los grupos eran homogéneos en cuanto a su cociente intelectual.
3. Pruebas de comprobación elaboradas en base a los contenidos establecidos para las unidades correspondientes en el sílabo del curso de Programación. Las pruebas fueron empleadas como pretest y postest.
4. Lista de cotejos que evaluó el logro de los objetivos por parte del estudiante a los que se aplicó el método inductivo modificado (MIM).

### Procedimiento para recolectar la información

Para la recolección de la información durante el desarrollo de la investigación se siguió el siguiente procedimiento:

1. Se seleccionó la población y muestra de estudio.
2. Al inicio del semestre, se aplicó a la muestra el pretest para evaluar a los estudiantes en los contenidos conceptuales del curso de Programación.
3. Con el grupo experimental se desarrolló el curso de programación, aplicando el método propuesto MIM por unidades temáticas, de acuerdo al diseño de investigación planteado.
4. Al grupo de control se le desarrolló el contenido de la asignatura con el método tradicional, presencial y expositivo.
5. En cada unidad temática se aplicaron pruebas objetivas de rendimiento en pretest y postest para ambos grupos.
6. Se aplicó la prueba de madurez mental California, serie intermedia 1950-S
7. Al finalizar el semestre, se procedió a la

aplicación del postest en ambos grupos a fin de evaluar comparativamente el progreso en el proceso de aprendizaje.

8. Se realizó el análisis estadístico y la interpretación de los resultados.

### Descripción del método MIM

#### Observación

La observación permite concentrar la atención del alumno en objetos, hechos o fenómenos, tal como se presentan en la realidad.

- **Exposición docente**

El profesor explica algunos ejemplos a partir de los cuales el alumno deberá ir captando la idea de lo que se desea que aprenda. El alumno observa con atención la explicación y trata de captar las ideas principales que se le desea transmitir.

#### Experimentación

La experimentación consiste en provocar el fenómeno que se estudia para que pueda ser observado. Permite comprobar o examinar las características de un hecho o fenómeno.

- **Trabajo en grupos**

El trabajo en grupos de dos o tres alumnos permite que el alumno que sabe más pueda compartir su conocimiento, explicando los métodos de solución a sus compañeros y esto a la vez le permite reforzar sus conocimientos.

- **Resolución de problemas**

El alumno resuelve problemas propuestos por el docente, para ello debe realizar trabajos prácticos que garanticen el logro de los objetivos educativos. En esta etapa se debe respetar un orden de dificultad gradualmente creciente.

#### Comparación

Se establecen las semejanzas o diferencias entre hechos o fenómenos observados.

- **Sustentación oral**

El alumno, explica y defiende en forma oral y delante de todos sus compañeros las soluciones planteadas para los problemas resueltos en la fase de experimentación. Justifica las acciones

propuestas. Se establecen similitudes y diferencias, fortalezas y debilidades entre los diferentes planteamientos de sus compañeros de clase.

#### Abstracción

Se selecciona los aspectos comunes a varios fenómenos o hechos estudiados y observados en conjunto, luego se generaliza para fenómenos o hechos similares.

- **Análisis crítico alumno**

Los alumnos analizarán las diferentes soluciones y planteamientos propuestos por sus compañeros en la etapa previa y en base a estos establecerán una o más posibles soluciones correctas para los problemas específicos. Luego se trata de establecer una forma de solución para problemas similares.

#### Generalización

Se aplica las características de los hechos estudiados a todos los que tienen su misma naturaleza. El hacer generalizaciones permite comprobar el resultado del procedimiento inductivo.

- **Resumen docente-alumno**

En esta etapa se establece una forma de solución estándar para problemas análogos. Esto se hace con la participación del docente y los alumnos, en forma conjunta.

#### Evaluación

La evaluación permite comprobar si se ha captado lo que se ha mostrado durante las etapas previas. Para esto se utilizan problemas similares o parecidos a los desarrollados.

- **Individual-grupal**

La evaluación se hará en forma individual o grupal, dependiendo de los temas tratados, de los tipos de problemas y del tipo de evaluación requerida por la institución.

#### Retroalimentación

Se requiere de una lista de problemas resueltos y propuestos por el profesor y de guías de trabajos prácticos. Se debe seguir un orden de dificultad gradualmente creciente.

- **Trabajos –apuntes WEB**

Se brinda al alumno más información referente al tema tratado a través de apuntes, problemas propuestos, problemas resueltos, sea en forma escrita o a través de la página WEB de la Universidad. El objetivo es reforzar lo aprendido por el alumno con trabajos a desarrollar fuera de horas de clase.

### Análisis de datos

Los datos obtenidos se procesaron mediante las herramientas estadísticas recomendadas por Samanamud (2001):

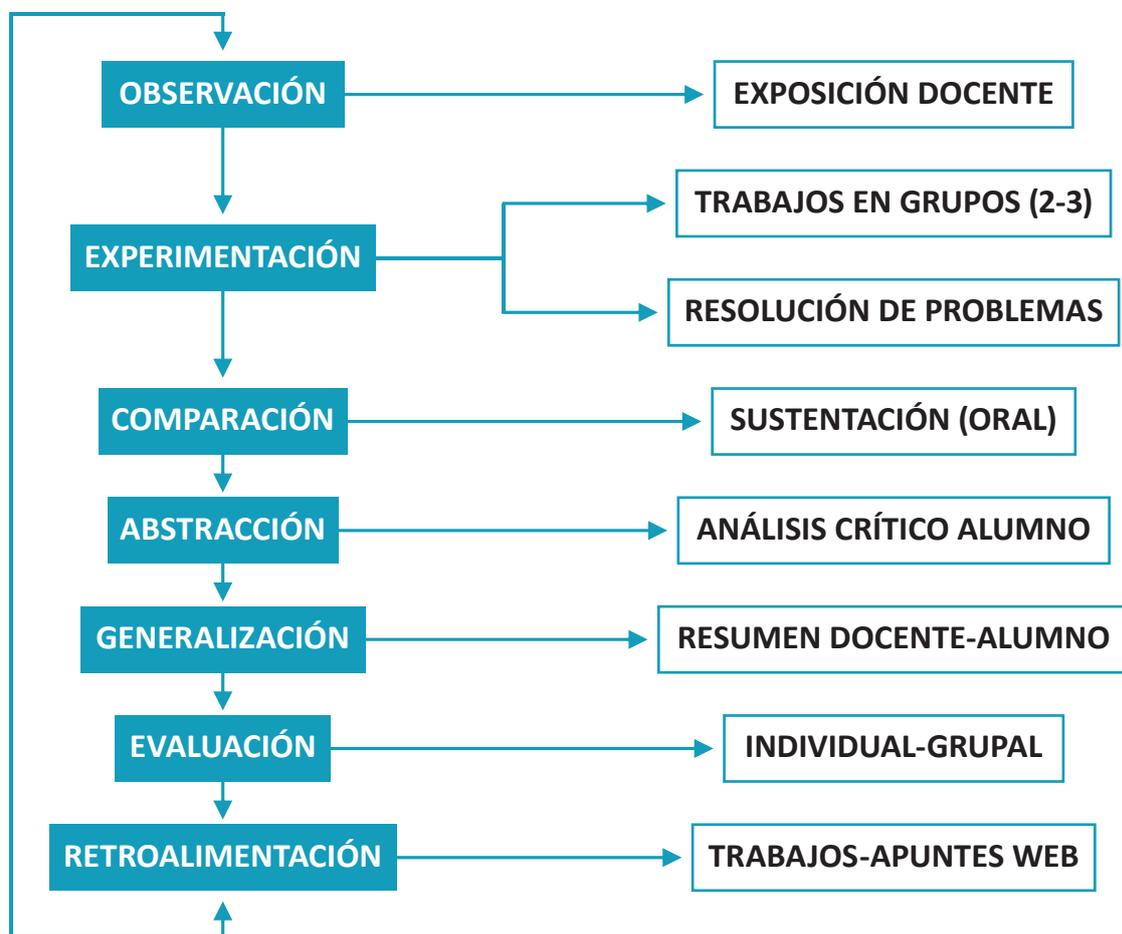


Figura N° 1: Estructura del método inductivo modificado (MIM).

## RESULTADOS Y DISCUSIÓN

A continuación, se muestran los cuadros con los datos de las diferentes pruebas realizadas y los resultados obtenidos

Alumnos	Grupos	Grupo de control		Grupo experimental	
		Pretest	Postest	Pretest	Postest
1		10	11	1	11
2		3	11	10	19
3		3	13	5	15
4		1	14	1	14
5		1	11	1	11
6		4	0	3	12
7		4	12	4	15
8		4	11	1	12
9		6	14	11	20
10		1	6	1	11
11		1	6	4	12
12		11	11	1	5
13		2	11	6	20
14		3	5	1	4
15		3	14	1	5
16		5	13	4	11
17		4	0	3	11
18		1	0	5	11
19		3	11	1	6
20		5	11	4	13
21		1	11	3	11
Promedio		3.62	9.33	3.38	11.86
Des. Estándar		2.75	4.62	2.89	4.50
CV %		75.91	49.49	85.46	37.93

**Tabla N° 1:** Rendimiento académico global del pretest y postest en los grupos experimentales de los alumnos del curso de Programación de los alumnos de primer ciclo de la Escuela de Ingeniería de Computación y Sistemas.

Estadísticas de rendimiento	Grupo de control		Grupo experimental	
	Pretest	Postest	Pretest	Postest
Mínimo	1	0	1	4
Máximo	11	14	11	20
Promedio	3.62	9.33	3.89	11.86
Desviación estándar	2.75	4.62	2.89	4.50
Coficiente de Variación	75.91	49.49	45.46	37.93
Número de alumnos	21	21	21	21

**Tabla N° 2:** Estadísticas de centralización y dispersión del rendimiento académico de los grupos de control y experimental correspondientes al pretest y postest.

GRUPO CONTROL						GRUPO EXPERIMENTAL					
Alumnos	Edad cronológica	Puntaje total	Edad mental	Cociente inteligencia	Categoría mental	Alumnos	Edad cronológica	Puntaje total	Edad mental	Cociente inteligencia	Categoría mental
1	192	75	184	95	Normal	2	192	86	197	104	Normal
2	192	47	146	76	Inferior	5	192	61	163	84	Inferior
4	192	70	174	91	Normal inferior	6	192	84	195	113	Normal Superior
11	192	96	210	105	Normal	13	192	68	172	91	Inferior
12	192	79	190	99	Normal	14	192	77	188	99	Normal
15	192	106	221	116	Normal superior	16	192	83	194	120	Superior
17	192	77	188	98	Normal	18	192	83	194	101	Normal
21	192	78	189	98	Normal	22	192	80	191	102	Normal
26	192	77	188	98	Normal	27	192	93	204	105	Normal
32	192	75	184	90	Normal	30	192	91	202	105	Normal
33	192	96	210	109	Normal	34	192	89	200	106	Normal
36	192	103	218	114	Normal superior	35	192	83	194	100	Normal
38	192	82	193	102	Normal	40	192	86	197	104	Normal
39	192	70	174	91	Normal inferior	41	192	77	188	99	Normal
42	192	64	167	87	Normal inferior	46	192	110	226	101	Superior
43	192	83	199	105	Normal	47	192	86	197	102	Normal
44	192	77	188	97	Normal	41	192	77	188	97	Normal
45	192	87	198	103	Normal	47	192	82	193	102	Normal
50	192	72	178	93	Normal inferior	51	192	99	213	93	Normal
56	192	97	211	110	Normal superior	58	192	82	193	100	Normal
57	192	77	188	98	Normal	60	192	86	197	104	Normal
Promedio				99.00		Promedio				101.50	
Des. Estándar				9.54		Des. Estándar				7.3	

**Tabla N° 3:** Resultados muestrales del test de California en los grupos de control y experimental de los alumnos del curso de Programación del I ciclo de la Escuela de Ingeniería de Computación y Sistemas-UPAO.

CONSOLIDADO DEL GRUPO CONTROL						
Categoría mental	Superior	Normal superior	Normal	Normal inferior	Inferior	Total
Alumnos	0	3	13	4	1	21
Porcentaje	0	14.3	61.9	19.0	4.8	100
CONSOLIDADO DEL GRUPO EXPERIMENTAL						
Categoría mental	Superior	Normal superior	Normal	Normal inferior	Inferior	Total
Alumnos	1	1	17	0	2	21
Porcentaje	4.8	4.8	80.9	0	9.5	100

**Tabla N° 4:** Resultados consolidados del test de California en los grupos de control y experimental de los alumnos del curso de Programación del I ciclo de la Escuela de Ingeniería de Computación y Sistemas-UPAO.

GRUPO CONTROL								
Alumno	Unidad I		Unidad II		Unidad III		Unidad IV	
	Pretest	Postest	Pretest	Postest	Pretest	Postest	Pretest	Postest
1	14	7	2	8	0	10	5	6
2	5	3	1	2	0	9	3	6
3	1	7	1	7	0	10	5	6
4	5	7	1	8	2	10	10	13
5	2	6	2	10	1	11	4	14
6	10	16	2	16	2	15	7	16
7	4	5	1	7	1	10	11	6
8	5	9	1	5	0	1	4	9
9	5	7	2	9	0	7	5	9
10	3	7	1	5	1	11	9	7
11	4	9	2	16	2	16	3	13
12	16	18	1	19	2	18	11	15
13	4	8	0	5	2	2	4	5
14	5	13	2	9	3	12	4	13
15	2	2	1	7	1	11	4	12
16	4	8	1	17	2	17	7	17
17	4	11	1	14	0	11	4	16
18	8	6	1	9	2	7	5	6
19	7	8	1	14	1	10	6	10
20	3	5	4	17	0	16	6	16
21	6	6	1	6	1	11	5	7
22	10	15	3	16	0	10	6	14
Promedio	5.77	8.31	1.455	10.27	1.045	10.68	5.82	10.545
Des. Estándar	3.78	4.04	0.858	4.93	0.95	4.21	2.42	4.07
CV %	65.459	48.57	58.98	47.989	90.87	39.44	41.64	38.57

GRUPO EXPERIMENTAL								
Alumno	Unidad I		Unidad II		Unidad III		Unidad IV	
	Pretest	Postest	Pretest	Postest	Pretest	Postest	Pretest	Postest
1	4	8	1	9	3	10	4	9
2	5	7	1	11	2	9	3	6
3	4	7	2	12	3	7	4	8
4	1	7	1	11	0	9	2	8
5	8	11	1	11	1	9	3	10
6	5	8	2	12	0	8	1	12
7	6	8	1	9	1	7	4	11
8	2	11	2	17	4	13	4	14
9	3	9	1	8	2	7	3	8
10	4	10	2	13	6	11	2	9
11	7	15	1	11	4	8	8	8
12	3	9	2	12	2	10	3	8
13	8	12	1	13	5	10	6	12
14	2	4	1	12	3	10	9	5
15	8	17	5	20	10	20	10	19
16	7	11	1	13	0	10	3	11
17	2	11	1	13	3	11	4	12
18	5	16	2	16	2	19	6	17
19	6	14	1	16	4	11	6	13
20	7	11	1	12	3	13	5	1
Promedio	4.85	10.3	1.5	12.55	2.9	10.6	4.45	10.7
Des. Estándar	2.23	3.33	0.95	2.87	2.34	3.50	2.42	3.54
CV %	45.995	32.29	63.06	22.90	80.60	33.02	54.30	33.09

**Tabla N° 5:** Rendimiento académico en las unidades temáticas I, II, III y IV en pretest y postest en los grupos de control y experimental de los alumnos del curso de Programación del I ciclo de la Escuela de Ingeniería de Computación y Sistemas-UPAO.

UNIDAD I					UNIDAD II				
Alumno	Pretest	Posttest	di	di <sup>2</sup>	Alumno	Pretest	Posttest	di	di <sup>2</sup>
1	4	8	4	16	1	1	9	8	64
2	5	7	2	4	2	1	10	9	81
3	4	7	3	9	3	2	9	7	49
4	1	7	6	36	4	1	11	10	100
5	8	11	3	9	5	1	11	10	100
6	5	8	3	9	6	2	12	10	100
7	6	8	2	4	7	1	7	6	36
8	2	11	9	81	8	2	15	13	169
9	3	9	6	36	9	1	8	7	49
10	4	10	6	36	10	2	10	8	64
11	7	15	8	64	11	1	8	7	49
12	3	9	6	36	12	2	12	10	100
13	8	12	4	16	13	1	13	12	144
14	2	4	2	4	14	1	12	11	121
15	8	17	9	81	15	5	20	15	225
16	7	11	4	16	16	1	11	10	100
17	2	11	9	81	17	1	13	12	144
18	5	16	11	121	18	2	14	12	144
19	6	14	8	64	19	1	14	13	169
20	7	11	4	16	20	1	12	11	121
Suma total			109	739	Suma total			201	2129
Promedio			5.45		Promedio			10.1	
Desv. Estándar			2.76		Desv. Estándar			2.39	
UNIDAD III					UNIDAD IV				
Alumno	Pretest	Posttest	di	di <sup>2</sup>	Alumno	Pretest	Posttest	di	di <sup>2</sup>
1	3	10	7	49	1	4	9	5	25
2	2	9	7	49	2	2	6	4	16
3	3	7	4	16	3	4	8	4	16
4	0	9	9	81	4	2	8	6	36
5	1	9	8	64	5	3	10	7	49
6	0	8	8	64	6	1	12	11	121
7	1	7	6	36	7	4	11	7	49
8	4	13	9	81	8	4	10	6	36
9	2	7	5	25	9	3	8	5	25
10	6	11	5	25	10	2	8	6	36
11	4	8	4	16	11	8	6	-2	4
12	2	10	8	64	12	3	8	5	25
13	5	10	5	25	13	6	12	6	36
14	3	10	7	49	14	9	5	-4	16
15	10	20	10	100	15	10	19	9	81
16	0	10	10	100	16	3	10	7	49
17	3	11	8	64	17	4	12	8	64
18	2	19	17	289	18	6	16	10	100
19	4	11	7	49	19	6	12	6	36
20	3	13	10	100	20	5	14	9	81
Suma total			154	1346	Suma total			115	901
Promedio			7.7		Promedio			5.75	
Desv. Estándar			2.9		Desv. Estándar			3.552	

**Tabla N° 6:** Diferencias de los promedios en las unidades temáticas I, II, III y IV en pretest y posttest en los grupos de control y experimental de los alumnos del curso de Programación del I ciclo de la Escuela de Ingeniería de Computación y Sistemas-UPAO.

Característica evaluativo	T tabular	T <sub>c</sub>
a) Rendimiento académico global del pretest en los grupos control y experimental.	$T_{60} (0.05)$ $T = 1.684$	1.1460
b) Rendimiento académico global del postest en los grupos control y experimental.	$T_{60} (0.05)$ $T = 1.684$	1.885
c) Cociente intelectual promedio de grupos control y experimental.	$T_{60} (0.05)$ $T = 1.684$	1.09
d) Rendimiento en unidad I de pretest y postest del grupo experimental.	$T_{60} (0.05)$ $T = 1.684$	1.7229
e) Rendimiento en unidad II de pretest y postest del grupo experimental.	$T_{60} (0.05)$ $T = 1.684$	1.8073
f) Rendimiento en unidad III de pretest y postest del grupo experimental.	$T_{60} (0.05)$ $T = 1.684$	-0.0666
g) Rendimiento en unidad IV de pretest y postest del grupo experimental.	$T_{60} (0.05)$ $T = 1.684$	0.1268

Tabla N° 7: Resumen del análisis estadístico de las hipótesis y la decisión respectiva.

De acuerdo a la información presentada en este cuadro se observa lo siguiente:

- **En a:** En el pretest los alumnos del grupo control y experimental ingresaron con similares competencias al curso de Programación, por lo que son considerados grupos homogéneos.
- **En b:** El promedio del postest del grupo experimental es significativamente mayor que el promedio del postest del grupo control.
- **En c:** El cociente intelectual promedio de los alumnos del grupo experimental no se diferencia del cociente intelectual promedio del grupo control.
- **En d, e, f y g:** El promedio del postest del grupo experimental es significativamente mayor que el promedio del grupo control en las unidades I, II, III y IV.

Con los resultados estadísticos obtenidos se evidenció que la nota promedio en el postest del grupo experimental (11.86) es significativamente mayor al del grupo control (9.33). Esta diferencia permite afirmar que la aplicación del MIM permitió incrementar en forma significativa el rendimiento académico en el curso de Programación.

## CONCLUSIONES

- Se evaluó, como pretest, el rendimiento académico en el curso de Programación de los alumnos de primer ciclo de Ingeniería de Computación y Sistemas de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego, obteniéndose resultados antes de aplicar el método propuesto.
- Se estructuró el método inductivo modificado (MIM) en siete etapas: observación, experimentación, comparación, abstracción, generalización, evaluación y retroalimentación. Cada etapa con un conjunto de estrategias didácticas.
- Se aplicó el método inductivo modificado (MIM) en el grupo experimental y se evaluó como postest el rendimiento académico de los alumnos del grupo de control y del grupo experimental en el curso de Programación de los alumnos de primer ciclo de Ingeniería de Computación y Sistemas de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego.
- Se encontró que existe diferencia significativa entre el promedio obtenido en el postest del grupo experimental (11.86) con el promedio obtenido en el postest del grupo control (9.33), lo que indica que el MIM incrementa significativamente el rendimiento académico y los niveles de aprendizaje de los estudiantes del curso de Programación.

## REFERENCIAS BIBLIOGRÁFICAS

1. Azpilicueta, J. (2003). *Enseñanza de la matemática para no matemáticos: Una propuesta para considerar la resolución de problemas como metodología activa de aprendizaje del análisis matemático*. Universidad Tecnológica Nacional. Facultad Regional Córdoba.
2. Azpilicueta J., Ledesma, A. *Constructivismo en la educación de las Ciencias de la Computación. Una propuesta de enseñanza-aprendizaje en aula virtual basada en resolución de problemas*. Recuperado Enero 2009, de <http://biblo.una.edu.ve/docu.7/bases/anali/texto/Azpilicueta.pdf>
3. Bravo C., Redondo; Miguel, Ortega Manuel (2009). *Aprendizaje en grupo de la programación mediante técnicas de colaboración distribuida en tiempo real*. Universidad de Castilla. La Mancha. España Recuperado Enero 10, 2009, de <https://aipo.es/articulos/3/50.pdf>
4. Capella R. J. (1990). *Aprendizaje y constructivismo*. Lima: ACEI and Vanier.
5. Fowler (2002) *La taxonomía de Bloom y el pensamiento crítico*. Recuperado enero 15, 2008, de <http://www.eduteka.org/profeinvidad.php3?ProfInvID=0014>
6. Gagné, R. (1975). *Principios básicos del aprendizaje para la instrucción*. México: Diana
7. Pizarro R. (1985). *Rasgos y actitudes del profesoro efectivo*. Tesis para optar el grado de Magister en Ciencias de la Educación. Pontificia Universidad Católica de Chile. Chile.
8. Medina Rivilla, A. y Salvador Mata, F. (2009). *Didáctica general*. Madrid: Prentice Hall.
9. Moroni N., Señas P., *Estrategias para la enseñanza de la programación*, Recuperado Enero 2009, de
10. [http://sedici.unlp.edu.ar/bitstream/handle/10915/18901/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/18901/Documento_completo.pdf?sequence=1)
11. Samanamud Ríos, V. (2001). *Estadística aplicada a la educación*. Trujillo: Universidad Privada Antenor Orrego.
12. Stockholm Challenge Award (2009). *Teorías del aprendizaje, nuevo enfoque*. Recuperado en enero 2009 de <https://werina2000.wordpress.com/>
13. Williams, L. & Upchurch, R.L. (2001). In Support of Student Pair-Programming. ACM SIGCSE Conference for Computer Science Educators, February.