

Aprendizaje de la Lógica de Programación utilizando Computación Gráfica y soportado por Herramientas y Ejercicios en la Nube

Gonzalo José Hernández

Universidad de Nariño

Pasto – Colombia

gonzalohernandez@udenar.edu.co

Manuel Ernesto Bolaños

Universidad de Nariño

Pasto - Colombia

mbolanos@udenar.edu.co

1. INTRODUCCIÓN

En la formación de un Ingeniero de Sistemas es indispensable desarrollar las competencias necesarias para crear software, razón por la cual en los primeros semestres de la formación académica se cursan asignaturas que buscan desarrollar el pensamiento lógico para diseñar algoritmos que conlleven a dar solución a problemas específicos.

En el pènsum del programa de Ingeniería de Sistemas de la Universidad de Nariño, los cursos relacionados con el àrea de programación presentan un nivel alto de estudiantes que no aprueban la asignatura, esto se debe a que ellos no logran ser permeados por el fundamento lógico de la programación.

El proyecto "Enseñanza de la Lógica de Programación utilizando Computación Gráfica", nace como una necesidad de mejorar el proceso de aprendizaje en una etapa temprana, proporcionando a los estudiantes una nueva metodología que busca apoyarse en cuatro simples principios :

- Motivación: Logrando que los estudiantes participen del proceso en forma más dinámica.
- Simplicidad: Reduciendo el número de instrucciones requeridas en los ejercicios antes de ser ejecutados.
- Verificación precisa de resultados: Proporcionando prontitud en el descubrimiento del origen de un error.
- Facilidad en corrección de errores: Ofreciendo herramientas de corrección asequibles al estudiante para encontrar la fuente de un error y corregirlo rápidamente mientras se tiene claro el problema.

2. PROBLEMA

En las asignaturas relacionadas con lenguajes de programación en el programa de Ingeniería de Sistemas de la Universidad de Nariño, se aprecia que en promedio el 34.5% de los estudiantes reprobaban estas materias con una nota inferior a 3.0. Si se toma como base de aprobación de las materias una nota de 3.5, el porcentaje de estudiantes que no superarían esta nota aprobatoria sería del 64.4% demostrando que la mayoría de los estudiantes no están alcanzando un nivel de madurez deseable en el área de programación. Estos resultados estadísticos pueden ser vistos en <http://192.68.185.32/glearning/annex01.pdf>.

Los estudiantes que logran captar los fundamentos de la "algoritmia" son los que fácilmente pueden diseñar nuevas soluciones utilizando programación de computadores. El problema radica en la persistencia de un porcentaje de estudiantes (64.4%) que no logra obtener las competencias necesarias para desempeñarse en desarrollo de software.

Los elementos claves de la lógica de programación son pocos y bastante simples (secuenciación, datos y estructuras de control) pero su aplicabilidad no es tan clara cuando se afronta a nuevos problemas.

Para el proceso de adquisición de estos elementos, las metodologías tradicionales se apoyan en un "proceso de ejercitación" brindándole al estudiante una colección de problemas que deben ser resueltos bajo la guía de un docente. A manera de buenas prácticas metodológicas, estos problemas están clasificados de acuerdo a su complejidad y aplicación de conceptos, llevando en forma evolutiva al estudiante hacia una completa capacitación. Sin embargo, en muy pocos casos se contemplan otros aspectos metodológicos que permitan que el proceso sea más efectivo y pertinente.

3. PROPUESTA

El conjunto de problemas que tradicionalmente se trabajan en el proceso de aprendizaje, utilizan para la verificación de resultados una interface en donde el estudiante debería ingresar y recibir datos en forma textual. Si bien es cierto que esta forma de interacción con los datos y los resultados es buena, no proporciona de forma evidente cuatro principios básicos:

- Motivación:

La motivación es el principal elemento en el proceso de aprendizaje, despertando el interés del educando hacia la adquisición de nuevo conocimiento.

- Simplicidad:

La simplicidad en el proceso de aprendizaje permite no desviarse del tema central. Es importante no recargar al estudiante de elementos irrelevantes que lo puedan distraer.

- Verificación inmediata de Resultados:

La revisión oportuna y efectiva de resultados es importante en el proceso académico ya que entre más prolongado sea el tiempo en la detección de errores, se disipa más el concepto estudiado.

- Facilidad de corrección:

Reparar un error es parte del aprendizaje, puesto que amerita una revisión del proceso anterior; sin embargo, si esta actividad se tarda más de lo adecuado puede ocasionar en el educando desmotivación y abandono.

Por este motivo se planteó un curso de aprendizaje de programación que (sin perder sus estrategias tradicionales) incluyera ejercicios relacionados con computación gráfica y la utilización de una herramienta que permitiese lograr los cuatros principios básicos expuestos anteriormente y que conserve la sintaxis del lenguaje C++ que siempre ha sido utilizada para este curso.

4. OBJETIVO

Mejorar las competencias en desarrollo de software de los estudiantes de Ingeniería de Sistemas de la Universidad de Nariño, incluyendo estrategias metodológicas de enseñanza por medio de computación gráfica.

5. METODOLOGÍA

El proceso de esta investigación fue enfocado para ser trabajado con los estudiantes que tomaron su primer curso de programación en el programa de Ingeniería de Sistemas de la Universidad de Nariño; de esta forma, se logró aplicar las nuevas estrategias pedagógicas en una etapa temprana del proceso de formación de estos estudiantes.

La metodología llevada a cabo se basó en el método experimental, por medio del cual se tuvieron a disposición a dos grupos de estudiantes: uno experimental y otro de control de aproximadamente 16 estudiantes cada uno organizados aleatoriamente. De esta forma, al finalizar el proceso investigativo, se pudo confrontar los resultados obtenidos que serán discutidos en el Capítulo 7.

Para llevar a cabo este proyecto se contemplaron 3 fases detalladas así:

5.1 Fase 1: Preparación:

Durante la primera etapa se organizaron las herramientas conceptuales y de software que se utilizaron para aplicar las nuevas estrategias metodológicas a los estudiantes del grupo experimental.

La preparación contempló los siguientes elementos:

- Programa académico a seguir.
- Definición de estrategias metodológicas a seguir.
- Adecuación de una herramienta de software para aplicar la metodología.
- Colección de problemas basados en computación gráfica.
- Diseño de instrumentos de evaluación de información de la investigación.

5.2 Fase 2: Aplicación:

Durante la segunda etapa se llevó a cabo el curso en la asignatura "Taller de Programación I" del programa de Ingeniería de Sistemas de tal forma que un grupo de estudiantes desarrolló su proceso en forma tradicional (Grupo Control) y el otro grupo de estudiantes (Grupo Experimental) aplicó las nuevas estrategias metodológicas planteadas en esta investigación.

5.3 Fase 3: Análisis de resultados:

Durante la tercera etapa se realizó la revisión de los instrumentos de recolección de información de la investigación, lo cual permitió determinar la pertinencia de la aplicación de las nuevas estrategias metodológicas.

6. DESARROLLO DE LA INVESTIGACIÓN

6.1 FASE 1

6.1.1 ORGANIZACIÓN DEL CURSO

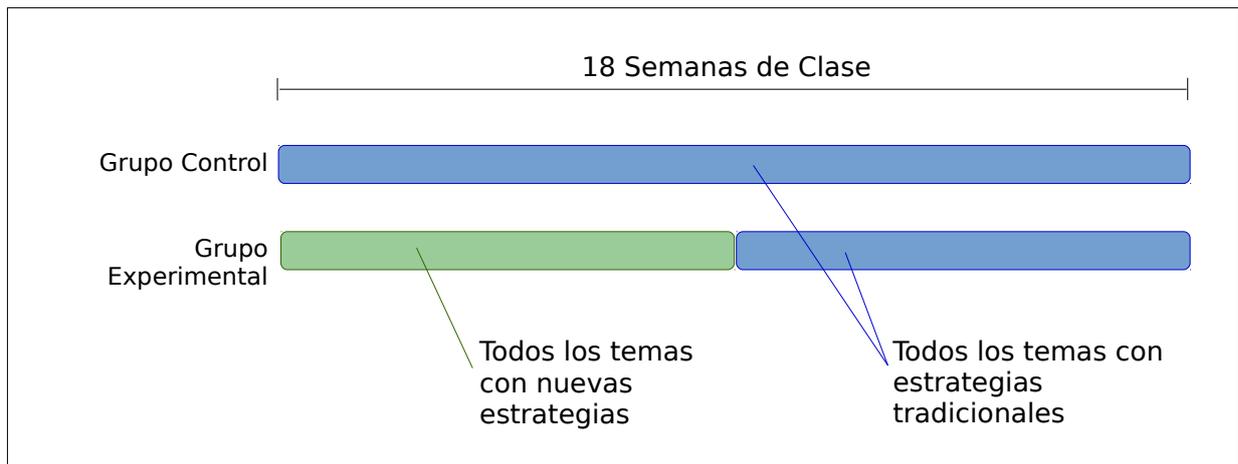
El curso de "Taller de Programación I" contempla 5 temas relevantes dentro de su contenido programático, los cuales se trabajan con un fundamentación teórica inicial y ejercicios de refuerzo.

Los temas a tratar son:

- Constantes y Variables
- Instrucciones de Entrada y Salida
- Estructura de decisión
- Ciclos
- Vectores

Para el Grupo Experimental, se re-acomodaron los contenidos programáticos de la asignatura, de tal forma que todos los temas del curso fueran dictados en la primera mitad utilizando las estrategias metodológicas planteadas por esta investigación y luego en la segunda mitad del curso se dictaron todos los temas nuevamente pero con las estrategias tradicionales, la Figura No. 1 muestra dicha distribución.

Figura No. 1



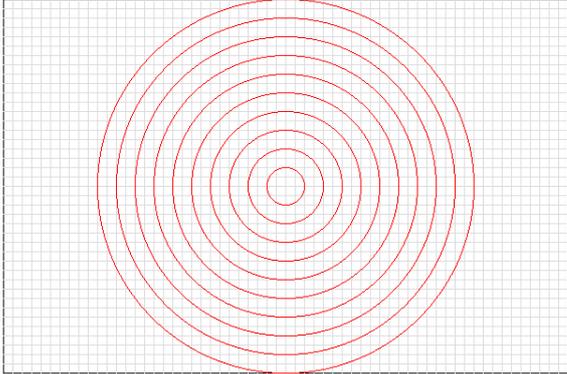
El detalle de la planeación del curso de Taller de Programación I se lo puede ver en <http://192.68.185.32/glearning/annex02.pdf>.

6.1.2 ESTRATEGIAS METODOLÓGICAS IMPLEMENTADAS

Tal como se plantea en el objetivo general de esta investigación, se busca aprovechar algunas bondades que se dispone al trabajar con ejercicios relacionados con computación gráfica que permitan optimizar los cuatro principios básicos: Motivación, Simplicidad, Verificación inmediata de resultados y Facilidad de corrección

Estrategia 1. Diseño de ejercicios cuya ejecución estimule el sentido de la vista. Esto facilitará aumentar el interés del estudiante por cuanto es divertido dibujar figuras geométricas. La Figura No. 2 muestra un ejemplo de un ejercicio planteado en el curso.

Figura No. 2



HomeWork

–

Draw some circles (using loop structure) so that, all circles are drawing in the center of the window with a 20 pixels of separation.

Estrategia 2. Diseño de ejercicios cuya solución solo requiera pocas instrucciones y sin adornos sintácticos. En todos los ejercicios planteados durante la ejecución del curso se obtuvo un promedio de 8 instrucciones por ejercicio. La Figura No 3 muestra el código necesario para ejecutar el ejercicio de la Figura No. 2.

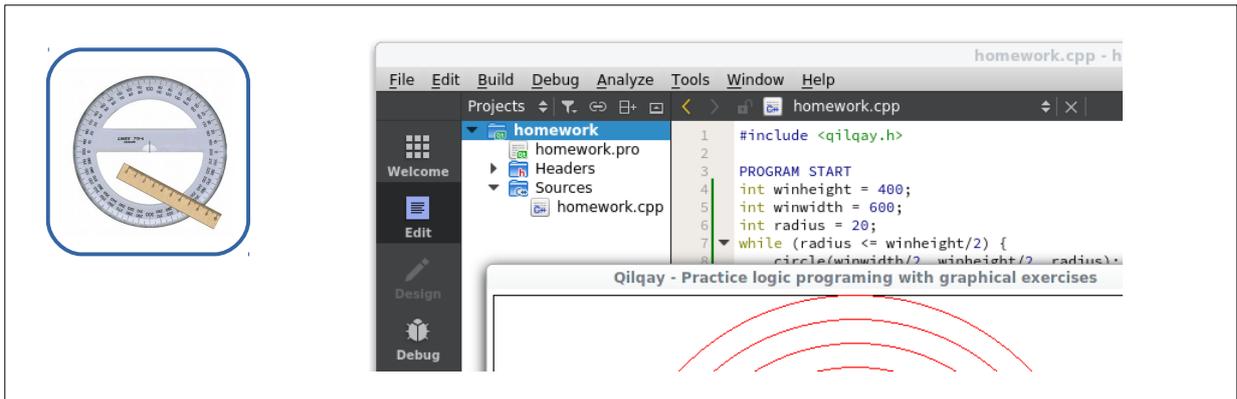
Figura No. 3

```
#include <qilqay.h>

PROGRAM START
int winwidth = 600;
int winheight = 400;
int radius = 20;
while (radius <= winheight/2)
{
    circle(winwidth/2, winheight/2, radius);
    radius = radius+20;
}
END
```

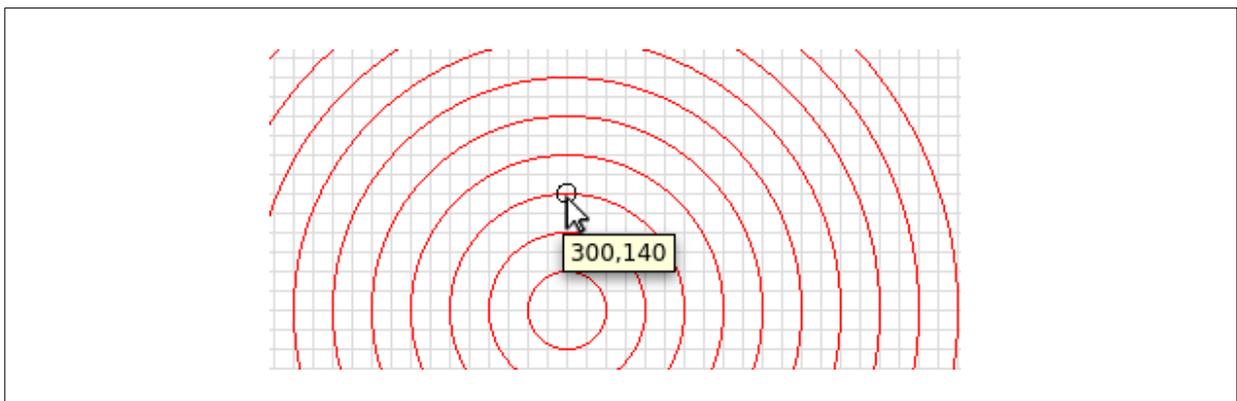
Estrategia 3. Emplear una herramienta computacional que permita probar los ejercicios sin tener que requerir un proceso complejo de ejecución y que mantenga la sintaxis de lenguaje C++. Una de las actividades desarrolladas en esta misma etapa fue la adecuación en una herramienta de apoyo docente (Detallada en la sección 6.1.3). La Figura No. 4 muestra un ejemplo de su ejecución.

Figura No. 4



Estrategia 4. Emplear una herramienta computacional que disponga de elementos que permitan al estudiante verificar oportunamente los resultados obtenidos. Esto con el fin de que la retroalimentación del proceso sea pertinente. La Figura No. 5 presenta una de las utilidades de la herramienta de apoyo docente, que permite al estudiante verificar las coordenadas por donde se dibujó un gráfico, así como una grilla personalizable que permite apreciar distancias y tamaños.

Figura No. 5



6.1.3 ADECUACIÓN DE UNA HERRAMIENTA

Debido a que las estrategias que se planearon desarrollar no recibían respaldo por parte de los IDE's (Integrated Development Environment) disponibles en el mercado, se decidió adecuar una herramienta que ya se había implementado en un proyecto anterior, de tal forma que sirviera de apoyo al trabajo docente de esta investigación.

La herramienta utilizada es un conjunto de dos componentes: Una librería y un asistente. El primer componente: *library*, es el espacio que tiene la definición de todas las clases con sus

respectivos métodos escritos en lenguaje C++ y usando un diseño Orientado a Objetos. Y el segundo componente: *wizard*, es un script que permite a QtCreator iniciar un proyecto con todos los archivos necesarios listos para su ejecución, éste se encuentra escrito con una sintaxis especial para que sea interpretado por el IDE.

La información completa acerca de esta herramienta puede ser encontrada en: <http://sonar.udenar.edu.co/glearning-tool>

6.1.4 COLECCIÓN DE PROBLEMAS BASADOS EN COMPUTACIÓN GRÁFICA

Antes de iniciar el curso en la Fase 2 se definieron algunos ejercicios que servirían como base, estos fueron clasificados en 6 niveles así:

- Ejercicios de Razonamiento Espacial
- Ejercicios usando Variables e instrucciones de entrada de datos
- Ejercicios de Decisión
- Ejercicios de Ciclos
- Ejercicios con Funciones
- Ejercicios de Vectores

Los ejercicios planteados para cada uno de los categorías pueden ser encontrados en <http://sonar.udenar.edu.co/glearning>.

6.1.5 DISEÑO DE INSTRUMENTOS DE INVESTIGACIÓN

Con el objetivo de recolectar la información necesaria y cumplir con el objetivo de la investigación, se elaboraron 3 instrumentos :

- **Instrumento1:** Caracterización de Estudiantes:

Tipo: Informativo

Modo de aplicación: Digital

Este primer instrumento se diseñó con el objetivo de tener una base informativa acerca de las características de cada uno de los estudiantes que participarían en el proceso. Al

iniciar el proyecto se identificó que la información obtenida de este instrumento no aportaría información concluyente para investigación, pero se recolectó para futuras conclusiones y proyectos de investigación.

- **Instrumento 2:** Saberes previos

Tipo: Importante

Modo de aplicación: Digital

Con este instrumento se propuso identificar el grado de homogeneidad del grupo de estudiantes respecto a sus conocimientos relacionados con programación.

- **Instrumento 3:** Test de evaluación final

Tipo: Indispensable

Este instrumento fue diseñado con el objetivo de analizar el rendimiento académico de los dos grupo (Control y Experimental), permitiendo finalizar con las conclusiones y recomendaciones de esta investigación.

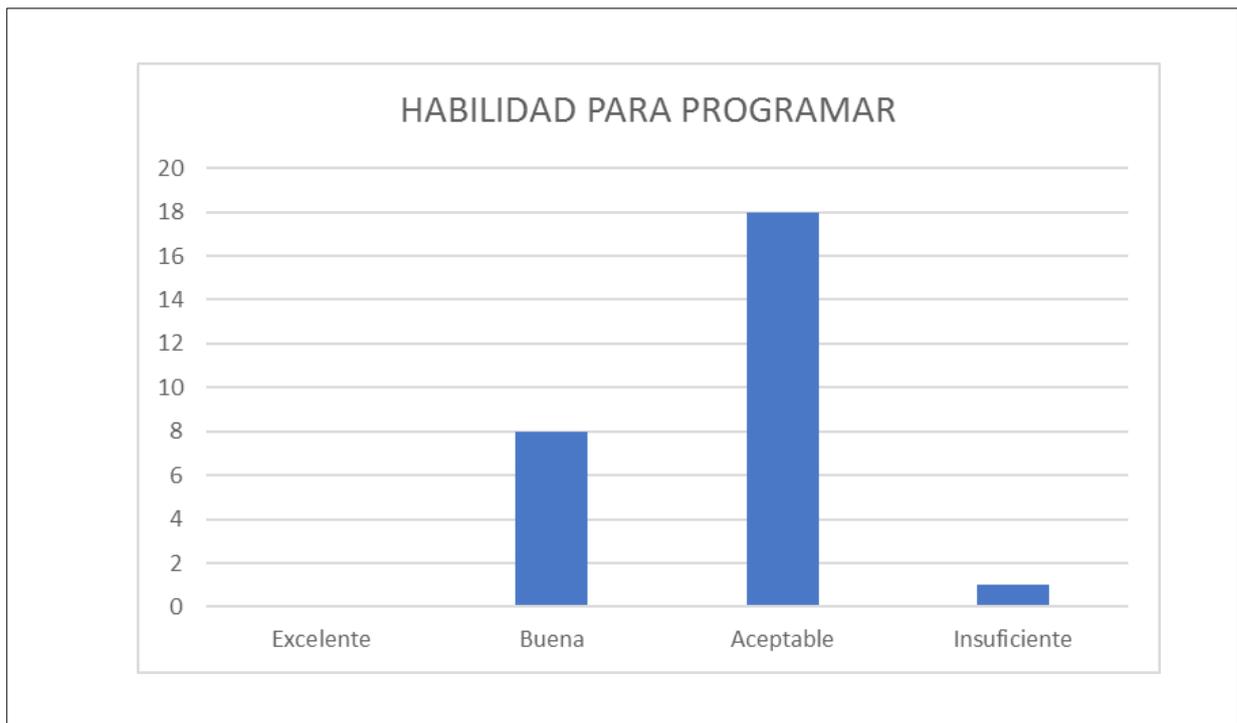
6.2 FASE 2

Para la puesta en marcha de esta etapa se organizó a los estudiantes del curso de Taller de Programación I de Ingeniería de Sistemas de la Universidad de Nariño, en dos grupos: Uno fue denominado Grupo Experimental, el cual tuvo una asistencia regular de 19 estudiantes y otro denominado Grupo Control, en el cual asistieron 13 estudiantes en forma regular.

El curso se inició con la aplicación de los dos primeros instrumentos (Caracterización de estudiantes y Saberes previos).

Con los resultados obtenidos se pudo analizar que los estudiantes sí tenían el grado de homogeneidad esperado.

Figura No. 6



La Figura 6 muestra que la mayoría de los estudiantes se encuentra en un rango intermedio, solo un estudiante fue clasificado en el nivel de insuficiente. Esta caracterización también identificó que más del 75% de los estudiantes se encuentran en edades entre 17 y 19 años, que solo tres estudiantes expresaron no tener buenas habilidades para el dibujo y que todos tienen acceso a un computador. De esta forma se justificó que el grupo de estudiantes era homogéneo.

El Grupo Control, tomo sus clases según la metodología convencional, y el Grupo Experimental procedió con las siguientes actividades:

Actividad	Tiempo
Inducción y conceptos básicos	Semana 1
Constantes y Variables	Semana 2
Instrucciones de Entrada y Salida	Semana 3 y 4
Estructuras de decisión	Semana 5 y 6
Ciclos	Semana 7 y 8
Vectores	Semana 9
Repeticón del curso utilizando metodología convencional	Semana 10 a 18

Los conceptos básicos que se trabajaron en la primera semana, consistieron en razonamiento espacial fundamentado en el ambiente gráfico de una pantalla de computador sobre el cual se dispone de dos dimensiones que permiten dibujar por medio de coordenadas.

En la segunda semana, se trabajaron los tipos de datos más comunes: Enteros, Reales, Texto y Booleanos. También se utilizó valores aleatorios en la ejecución de los ejercicios de esta sección.

En la tercera y cuarta semana, se trabajaron ejercicios en los cuales para cumplir el objetivo era necesario involucrar al usuario, para este fin la herramienta contaba con la posibilidad de solicitar que el usuario introdujera coordenadas por medio del ratón.

En la quinta y sexta semana, los ejercicios podían tener comportamientos diferentes de acuerdo a condiciones previamente programadas, incluso la gratificación también podía depender de información proporcionada por el usuario.

En la séptima y octava semana, se logró realizar gráficos más complejos gracias a la posibilidad de la reutilización de instrucciones, así como la posibilidad de mantener un programa corriendo mientras alguna condición preprogramada se cumpla.

En la novena semana, se realizaron ejercicios en los cuales se podía tener una pequeña base de datos de valores que podía ser utilizada como coordenadas o distancias para la realización de gráficos. También se realizaron ejercicios en los cuales un arreglo de valores se podía ir cargando en tiempo de ejecución.

Las semanas siguientes fueron utilizadas para repetir el curso de programación, pero utilizando ejercicios convencionales (no relacionados con gratificación).

Al finalizar el curso se procedió con la aplicación del tercer instrumento de investigación (Test de evaluación final) a los dos grupos en conjunto (Grupo Experimental y Grupo de Control)

6.3 FASE 3

Durante la tercera etapa de la investigación se organizaron los datos recolectados, los cuales fueron organizados y analizados para lograr determinar si la utilización de la nueva estrategia fue adecuada o no.

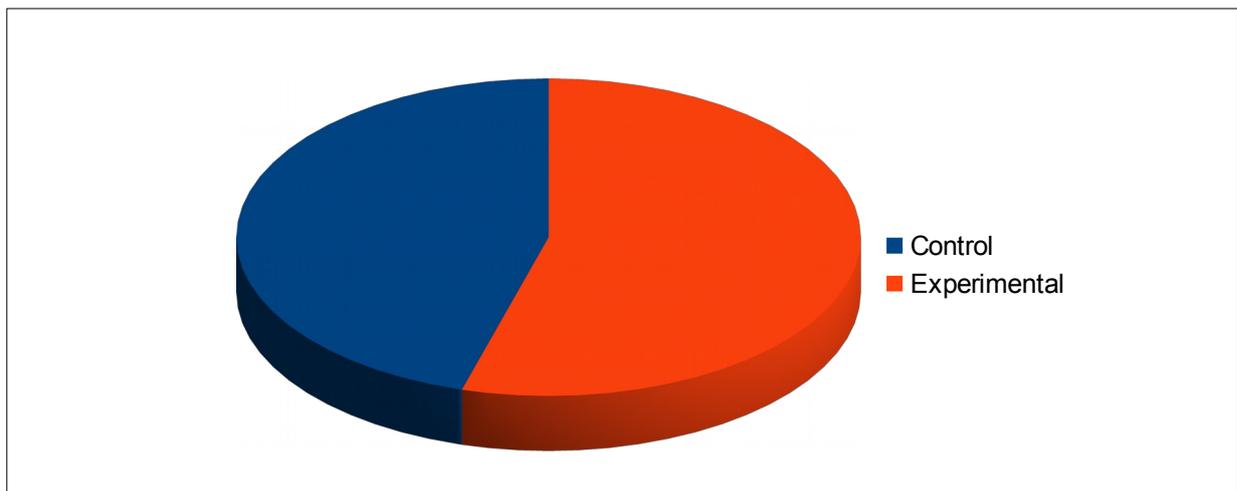
En esta fase también se documentó el proceso investigativo, finalizando con el capítulo donde se discuten los resultados obtenidos.

7. DISCUSIÓN DE RESULTADOS

Gracias a la información recolectada con los instrumentos diseñados en la FASE 1, fue posible obtener los siguientes resultados.

Con la aplicación del Test de Evaluación Final se pudo determinar que en general, los estudiantes del Grupo Experimental superaron a los estudiantes del Grupo Control. En total, el Grupo Experimental obtuvo un puntaje de 4.2 respecto a 3.5 obtenido por el Grupo Control.

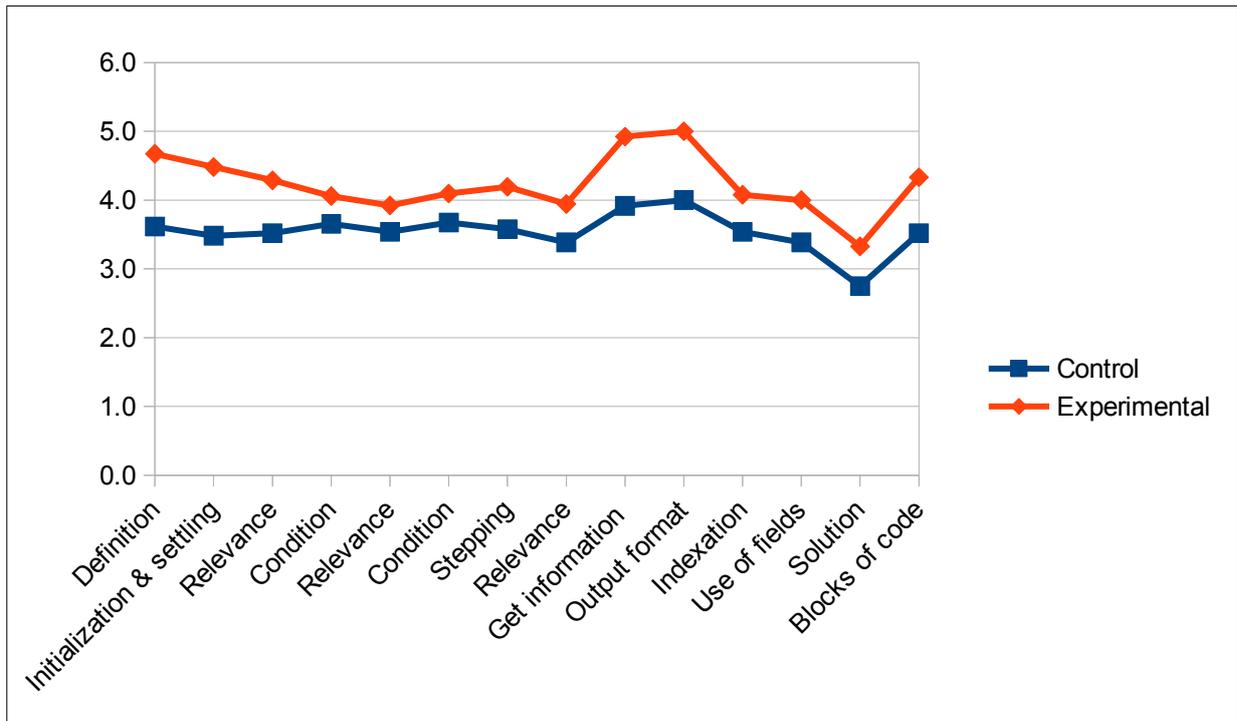
Figura No. 7



Esta situación reflejada en la Figura 7, apoya la hipótesis de que la estrategia de utilizar computación gráfica como mecanismo de enseñanza de la programación, ayudó a que los estudiantes del Grupo Experimental cumplieran en mayor grado los objetivos planteados en el curso de Taller de Programación I.

El Grupo Experimental superó al Grupo de Control en todos los aspectos tal como lo muestra la Figura 8, en donde en el Grupo Experimental, el aspecto mejor valorado está relacionado con las instrucciones de Entrada y Salida con una puntuación de 4.9 para entrada de datos y de 5.0 para salida de información.

Figura No. 8



En la Figura 8 también se puede observar que el aspecto en el cual aún se debe mejorar, es el relacionado con la efectividad de la solución, es decir, que la solución cumpla con todos los requerimientos solicitados. Sin embargo el Grupo Experimental obtuvo una calificación mayor a la del Grupo Control con una diferencia de 5 décimas (2.8 vs 3.3).

Además de los resultados obtenidos por medio del Test de Evaluación Final, también se pudo determinar las siguientes apreciaciones anotadas como diario de campo por parte del profesor que impartió el curso con el Grupo Experimental:

Para el docente fue más fácil explicar el objetivo de un ejercicio (relacionado con computación gráfica) debido a que en ocasiones una imagen era suficiente para que el estudiante comprendiera lo que debe hacer.

Para el estudiante también existió mayor facilidad en la comprensión de los requerimientos de un ejercicio, esto se pudo notar en cada una de las sesiones en las cuales los estudiantes no requerían de mayores explicaciones.

Durante las prácticas en el laboratorio de informática, se notó a los estudiantes más motivados a resolver los problemas propuestos, puesto que la disposición era más evidente que en cursos de semestres pasados.

También fue notoria la satisfacción cuando los estudiantes lograban resolver los ejercicios propuestos, esto, debido a la prontitud de la presentación de los resultados por parte del programa.

La ventaja de tener ejercicios con objetivos muy específicos, aligeró el proceso de evaluación y calificación de los mismos.

8. CONCLUSIONES

Con los resultados obtenidos se estima que la estrategia implementada consistente en trabajar ejercicios relacionados con computación gráfica, es adecuada para ser utilizada en la enseñanza de lógica de programación.

Tal como se presentó en los resultados obtenidos, se considera que la motivación se considera como un elemento esencial dentro del proceso de aprendizaje.

La herramienta utilizada para llevar a cabo el curso de programación con esta nueva estrategia fue esencial, puesto que gracias a ella fue posible compilar y ejecutar los ejercicios sin necesidad de pasos anteriores ni intermedios.

La experiencia obtenida en esta investigación muestra que es importante seguir innovando en estrategias para mejorar el proceso de aprendizaje en las áreas de programación.

BIBLIOGRAFIA

[1] Departamento de Ingeniería de Sistemas y Computación Universidad de los Andes. Proyecto cupi2. <http://cupi2.uniandes.edu.co/sitio>, 2004.

[2] José Escaño. Motivar a los alumnos y enseñarles a motivarse. AULA. De Innovación Educativa, 2001.

[3] James D. Foley. Computer Graphics: Principles and Practice in C (2nd Edition). Addison-Wesley, 1995.

[4] Donald E. Knuth. The Art of Computer Programming, volume 1. Addison-Wesley, Canada, 2011.

[5] Adrian Nye. Xlib Programming Manual for Version 11. O'Reilly, 1994.

[6] Ronald L. Rivest Clifford Stein Thomas H. Cormen, Charles E. Leiserson. Introduction to Algorithms. The MIT Press, Massachusetts London, 2001.

[7] Silvio R. Timarán. Un cambio de paradigma en la enseñanza de fundamentos de programación en ingeniería de sistemas. Educacion en Ingenieria, 7:120–128, 2009.