

**INFORME FINAL DE TESIS**  
**Maestría en Educación**  
**Universidad de los Andes – Centro de Investigación y Formación en Educación (CIFE)**

**Aprendizajes de Programación de Computadores y Resolución de Problemas en un  
Ambiente de Trabajo en Colaboración**

**Presentado por Claudia Patricia Castañeda Bermúdez**

**Dirigido por Claudia Lucía Ordóñez Ed. D.**

**Junio de 2007**

## CONTENIDO

1	RESUMEN EJECUTIVO .....	3
2	MARCO CONCEPTUAL .....	4
3	DESCRIPCIÓN DE LA INTERVENCIÓN .....	10
4	LA INVESTIGACIÓN.....	11
5	RESULTADOS.....	17
6	DISCUSIÓN.....	41

## 1 RESUMEN EJECUTIVO

En esta investigación exploro los efectos, en el aprendizaje de programación de computadores de alumnos universitarios, de una estrategia de aprendizaje que integra trabajo en colaboración y resolución de problemas. El aprendizaje de la programación de computadores es tema de estudio en muchas investigaciones que en su mayoría proponen el uso de algún tipo de herramienta, generalmente software. En la innovación pedagógica que introduje en mi curso de programación me enfoqué, antes que en la herramienta didáctica o el contenido del curso, en diseñar la mejor forma de acercar a mis estudiantes a la comprensión del uso real de la programación, para lograr el aprendizaje de esta disciplina. Mi clase se basó en permitir que los estudiantes aprendieran los conceptos básicos de la programación por medio de su aplicación en la solución de problemas, de modo que el continuo análisis y discusión grupal de los problemas los llevara a desarrollar habilidades y conocimientos importantes para la resolución de problemas usando la programación. La investigación que presento sobre el aprendizaje logrado contesta preguntas sobre las habilidades de solución de problemas y los conocimientos de programación adquiridos y sobre la posible influencia del trabajo en colaboración sobre estos logros. Trabajé con un grupo de diecinueve estudiantes de ingeniería electrónica en un primer curso de programación de computadores. Por medio de grabaciones, encuestas, entrevistas y pruebas académicas recogí datos que analicé cuantitativa y cualitativamente. Encontré que los estudiantes adquirieron conocimientos de programación, estrategias de resolución de problemas, habilidades de trabajo en grupo y flexibilidad en la aplicación de sus conocimientos. Los hallazgos de esta investigación contribuyen al estudio sobre la enseñanza y aprendizaje de la programación de computadores y en general a las prácticas pedagógicas en diferentes áreas de conocimiento donde se observan métodos educativos que no reconocen las capacidades de aprendizaje de los estudiantes ni brindan la oportunidad para que ellos descubran la utilidad de lo que aprenden.

## 2 MARCO CONCEPTUAL

El interés de profesores e investigadores en diseñar estrategias pedagógicas que promuevan el aprendizaje no es ajeno al dominio de la programación de computadores. Este interés ha motivado la realización de numerosos estudios y proyectos que buscan encontrar nuevas y mejores maneras de acercar a quienes aprenden a esta interesante disciplina, la mayoría basados en la participación activa de los alumnos en actividades de aprendizaje que se realizan individualmente y en colaboración con compañeros. En este marco conceptual presento, a manera de sustentación de una estrategia pedagógica desarrollada por mí, estudios sobre la enseñanza y el aprendizaje de la programación de computadores, especialmente estudios que se enfocan en diferentes aplicaciones del aprendizaje en colaboración, información y estudios sobre la relación entre el aprendizaje de programación y la resolución de problemas, información sobre una única intervención que cuestiona ya no la forma de aprender sino los aspectos de la programación que se trabajan y, finalmente, un resumen de la teoría de aprendizaje que llevo a la práctica.

Diferentes estrategias de aprendizaje se han propuesto y usado para la enseñanza de la programación. Algunas propuestas se centran en las herramientas empleadas. Según Taylor, Cunniff y Uchiyama (1986), Giannotti (1987) y Guibert, Girard y Guittet (2004), por ejemplo, se puede iniciar a los estudiantes en la programación cambiando el uso de lenguajes e interfaces poco atractivas por herramientas que capturen fácilmente su interés, que les permitan elaborar productos rápidamente y que además les permita aprender a utilizar las estructuras básicas de la programación. Sin embargo, Giannotti (1987), en su estudio, llama la atención sobre el hecho de que los estudiantes lograron aprendizajes sobre programación, pero los procesos de análisis, tan importantes para la programación y la solución de problemas, parecen no haberse promovido con el uso de este tipo de herramientas. Jacovljevic (2003), por su parte, propuso y usó, en un estudio realizado en la universidad de Witwatersrand en Sudáfrica, mapas conceptuales trabajados en

grupos de pares. Sus hallazgos mostraron que los mapas conceptuales pueden ser una herramienta efectiva para construir, organizar y representar los conocimientos de los estudiantes. También hizo hallazgos interesantes relacionados con el trabajo en grupo, pues los alumnos encontraron más fácil discutir y aprender, entre pares podían ayudarse unos a otros y conocer diferentes formas de entender los conceptos.

Muchas investigaciones muestran que los salones de clase con estructuras rígidas y controladas acaban con la motivación y la creatividad de los estudiantes. El aprendizaje en colaboración, una estrategia de aprendizaje basada en principios constructivistas, es reconocido como estrategia que se opone a esta rigidez y mejora la motivación y el aprendizaje. LeJeune (2003) indica que un curso de programación es un ambiente propicio para usar ejercicios de aprendizaje en colaboración. Hace énfasis en que el trabajo en colaboración no es sólo hacer trabajos en grupo, sino que se deben tener en cuenta otros aspectos como la pertinencia de las tareas que se realizan, la creación de colaboración real y el funcionamiento de la responsabilidad individual con el grupo, para asegurar la efectividad de la estrategia. LeJeune (2003) dice que el aprendizaje en colaboración es especialmente benéfico en situaciones que requieren la aplicación de conocimientos complejos y abstractos como los de la programación.

Yerion (1995) ya había destacado la importancia del uso del aprendizaje en colaboración en asignaturas relacionadas con las ciencias de la computación, sobre todo para que los estudiantes aprendan a trabajar con otros, preparándose así para los ambientes de trabajo que se viven en las industrias relacionadas con la tecnología. Sin embargo, llamó la atención sobre la resistencia que se produce en algunos estudiantes que usan esta estrategia pedagógica, debido a que la presión por ser competitivos puede hacer que exijan que sus esfuerzos sean reconocidos a nivel individual. Para contrarrestar esto destacó la importancia del papel del profesor con su soporte y acompañamiento permanente y la necesidad de desarrollar formas efectivas de evaluación de logros tanto individuales como de equipo. Finalmente sugiere la necesidad de realizar muchas

investigaciones que estudien esta resistencia al cambio y el proceso que siguen los estudiantes en estos ambientes de aprendizaje.

El aprendizaje en colaboración se ha utilizado en muchas intervenciones pedagógicas para la enseñanza y el aprendizaje de la programación de computadores, y sus efectos también han sido estudiados. Una versión de esta estrategia es la programación en pares, en la que dos programadores trabajan en colaboración frente a un computador en el análisis y en el diseño de programas. En dos estudios realizados en universidades se usó esta práctica como estrategia para enseñar cursos de programación. En los dos se manejó un grupo de control en el que los estudiantes trabajaron individualmente y un grupo experimental que usó la programación en pares. El primero, realizado en la Universidad de Washington (VanDeGrift, 2004), duró 10 semanas y participaron 550 estudiantes. En el grupo experimental se realizaron tres proyectos de programación en pares junto con la escritura individual de informes de estos proyectos. Se hicieron análisis cualitativos y cuantitativos tanto de los informes como de encuestas aplicadas a los estudiantes. El segundo estudio, realizado en la Universidad de Carolina del Norte (Nagappan, Williams, Ferzli, Wiebe, Yang Miller, & Balik, 2003), duró un año y participaron aproximadamente 700 estudiantes. El grupo experimental tuvo una sesión semanal de tres horas de trabajo en pares, además de una sesión de clase magistral. Los datos se recogieron por medio de observaciones, encuestas, entrevistas y evaluaciones académicas, que fueron analizados cualitativa y cuantitativamente.

Los dos estudios encontraron que los estudiantes percibieron como beneficios del trabajo en compañía el encontrar rápidas respuestas en sus compañeros, disminuir sentimientos de frustración, ver distintas formas de resolver los programas y desarrollar habilidades para el trabajo en colaboración. Como desventaja, los estudiantes manifestaron dificultades por las diferencias de personalidad con sus compañeros. Con respecto a su aprendizaje manifestaron mayor confianza en sus soluciones, mejor entendimiento de los proyectos y mayor capacidad de

detección y corrección de errores en los programas. En el segundo estudio las observaciones mostraron que las sesiones de laboratorio en solitario fueron pasivas y frustrantes, en oposición a las de pares. Igualmente los profesores de las sesiones en solitario respondieron más preguntas básicas que los de las sesiones de pares, donde las preguntas de los estudiantes fueron más elaboradas y demostraron mayores comprensiones. Los puntajes de los proyectos en los dos grupos fueron similares, aunque en algunos casos los estudiantes que trabajaron en pares tuvieron un puntaje mayor que los estudiantes solos.

Otro estudio realizado por Sabin y Sabin (1994) en Loyola College, Maryland, tuvo como propósito hacer un curso más efectivo e interesante por medio del trabajo en colaboración. Participaron 30 estudiantes divididos en un grupo de control, que siguió el método tradicional de enseñanza por medio de conferencias, y un grupo experimental que usó el aprendizaje en colaboración. Los datos se recogieron al inicio y al final por medio de evaluaciones y encuestas. Los resultados académicos de los dos grupos se compararon estadísticamente con pruebas t-Student. No se encontró diferencia significativa en conocimientos entre los dos grupos; sin embargo, al analizar la información cualitativamente los datos mostraron efectos positivos del aprendizaje en colaboración tales como ambiente de clase amigable, uso correcto de la terminología de la programación, mayor confianza para expresar los conocimientos, logro de objetivos en menor tiempo y facilidad para evaluar en clase el trabajo de los estudiantes.

La resolución de problemas es considerada por muchos educadores un objetivo de aprendizaje importante para los estudiantes en diversas áreas del conocimiento. Los resultados de varios estudios muestran que existen habilidades cognitivas desarrolladas en el aprendizaje de la programación de computadores que se requieren en la resolución de problemas (Pea & Kurland, 1984 en Tu & Johnson, 1990; Dalbey & Linn, 1985 en Tu & Johnson, 1990). Polya (1957) propuso que un proceso de resolución de problemas está dividido en cuatro fases: comprensión del problema, diseño de un plan, implementación, y, revisión y verificación. Su propuesta ha

sido ampliamente aceptada como forma de analizar y desarrollar las habilidades de resolución de problemas. Así, Perkins (1981, en Tu & Johnson, 1990) hizo un paralelo entre el proceso de elaboración de programas de computador en un ambiente de LOGO (Papert, 1972; 1980 en Nickerson, Perkins and Smith, 1990) y las fases propuestas por Polya. Según él, determinar las entradas y salidas de un programa es similar al proceso de comprensión inicial del problema; elaborar diagramas y algoritmos es equivalente a la fase de diseño de un plan; codificación es el equivalente a la fase de implementación y por último las pruebas de verificación son similares en los dos procesos.

Un estudio realizado en Pensilvania incorporó al aprendizaje en colaboración la solución de problemas y comparó los efectos académicos y de motivación provocados (Ginsburg-Block y Fantuzzo, 1998). Participaron 104 estudiantes de 4º y 5º grado en un curso de matemáticas. Los estudiantes se asignaron a cuatro grupos distintos: control, solución de problemas, trabajo en colaboración y solución de problemas más trabajo en colaboración. Se recogieron datos durante siete semanas por medio de pruebas académicas, encuestas y observaciones de clase. Para su análisis estadístico se usaron pruebas chi cuadrado y análisis de varianza (ANOVA) en una y dos vías. Uno de los hallazgos fue que el aprendizaje en colaboración estimuló la participación y mejoró la retención de conocimientos. Los resultados mostraron, además, que los estudiantes que participaron en solución de problemas mostraron mejores resultados en análisis y cálculos matemáticos. Los estudiantes que participaron en trabajo en colaboración tuvieron los mejores resultados en niveles de motivación académica y competencias sociales. Pero los mejores resultados en todas las áreas estudiadas fueron los que combinaron aprendizaje en colaboración y solución de problemas.

Sólo hallé una experiencia en la que se cuestiona el contenido mismo de lo que se enseña en programación de computadores y se incluye la solución de problemas en él. Esta experiencia, denominada Proyecto Cupi2 (Villalobos, Casallas y Marcos, 2005), identificó como posible



causa de las dificultades tanto en el aprendizaje como en la enseñanza de la programación el desequilibrio que existe en las propuestas pedagógicas de los cursos entre siete ejes conceptuales alrededor de los cuales el estudio considera que gira la labor de la programación: modelaje y solución de problemas, algorítmica, tecnología y programación, herramientas de programación, procesos de software, técnicas y metodologías de programación y elementos estructuradores y arquitecturas. El proyecto formula una propuesta pedagógica para promover el aprendizaje de la programación que gira alrededor de los siete ejes conceptuales y reconoce al estudiante como el principal protagonista del proceso de aprendizaje (Villalobos, 2007).

Tanto en la educación en programación como en otras disciplinas, hay una tendencia hacia el uso de prácticas pedagógicas que promueven la actividad del estudiante para mejorar su aprendizaje. El propósito es ayudar a los estudiantes a apropiarse de su proceso de aprendizaje, a interactuar con pares para construir conocimientos y a pensar reflexiva y críticamente produciendo argumentaciones lógicas y entendiendo a su vez la lógica de los argumentos de otros. Todas estas prácticas se sustentan en principios constructivistas como la comprensión del aprendizaje como proceso de construcción de conocimientos (Piaget, 1970; Vygotsky, 1978; Ordóñez, 2003), el aprendizaje en interacción social (Vygotsky, 1978), el desempeño como forma de aprender y demostrar comprensión (Perkins, 1998; Ordóñez, 2003), y específicamente el desempeño auténtico disciplinar (Perrone, 1998; Boix-Mansilla & Gardner, 1998; Ordóñez, 2003), y la zona vygotskiana de desarrollo próximo (Vygotsky, 1978; Berk & Winsler, 1995). Entre las diferentes prácticas consistentes con el constructivismo, el aprendizaje en colaboración constituye una estrategia que parece estimular la participación activa de los estudiantes en su aprendizaje y permitir la elaboración individual de conocimientos mediante la interacción social entre alumnos y entre profesor y alumnos (Bruffee, 1999). En ambientes de trabajo en colaboración los estudiantes van elaborando y modificando sus conocimientos por medio de un proceso de intercambio y confrontación de sus comprensiones (Bruffee, 1999; Savery y Duffy, 1996).

Falta, sin embargo, investigación sobre formas de enseñar que mezclen el trabajo en colaboración con desempeños auténticos de la programación como la solución de problemas e informe sobre sus resultados en el aprendizaje de los estudiantes. Por esto diseñé y adelanté una práctica pedagógica con estas características para mis clases de programación de computadores para estudiantes de tercer semestre de ingeniería electrónica y presento aquí los resultados obtenidos.

### **3 DESCRIPCIÓN DE LA INTERVENCIÓN**

Como profesora de programación, me encontraba con muchos estudiantes insatisfechos con el aprendizaje de la programación por encontrarlo difícil y poco útil para sus estudios y su vida profesional. Igualmente en mi papel de profesora me encontraba dedicada a enseñar la programación mostrándoles cómo hacía yo los programas. Luego, como estudiante de maestría en educación, comencé a ver que somos los estudiantes quienes en nuestra actividad encontramos dificultades y entonces, profesor y estudiantes, nos enfrentamos a ellas buscando soluciones y encontrando nuevos aprendizajes. Decidí que los estudiantes debían participar de manera activa en el aprendizaje de la programación, siendo ellos mismos quienes se enfrentarían al diseño de algoritmos y programas, de modo que en esta actividad logaran los aprendizajes junto a sus compañeros de clase, además de encontrar la utilidad de la programación y su aplicación en diferentes áreas.

Diseñé entonces una intervención pedagógica basada en esa búsqueda activa y autónoma de soluciones a problemas que involucraran los diferentes conocimientos de la programación, en interacción social con compañeros de clase. Durante un semestre realicé una prueba piloto de la intervención, luego de la cual realicé ajustes en mi rol de profesora en la clase, en el diseño de algunos problemas y en las formas de observar el aprendizaje de los alumnos. Finalmente, en mi intervención modificada, las actividades que realizaron los estudiantes estuvieron orientadas por las características fundamentales de los problemas y por la colaboración, así:

**Aprendizaje en colaboración.** Los estudiantes trabajaron en las clases del semestre organizados en grupos permanentes de dos o tres para elaborar soluciones a problemas propuestos. Los estudiantes debían todos al interior del grupo, discutir, proponer, colaborar y escribir cada uno su solución, de manera que al final yo siempre elegí cualquiera de ellas para recoger.

**Solución de problemas.** Fueron los estudiantes quienes desarrollaron los contenidos del curso aplicándolos en la construcción de las soluciones a los problemas, en un proceso en el que debieron comprenderlos y planear su solución aplicando la programación. En total los estudiantes trabajaron sobre siete problemas propuestos en diferentes momentos del curso. Los problemas consistieron todos en diferentes situaciones presentadas por medio de una narración, algunas de ellas relacionadas con el ambiente universitario. Con el fin de guiar a los estudiantes en el proceso de resolución, junto con el enunciado de cada problema les formulé una serie de preguntas para responder previo a la elaboración de la solución. A medida que avanzaba el curso, cada problema para su solución requería el uso de los temas que íbamos estudiando.

En su mayoría las sesiones se desarrollaron en un salón de clase sin computadores. Únicamente se realizaron ocho sesiones de práctica en un salón de computadores donde los estudiantes trabajaron las soluciones de algunos de los problemas propuestos. En las dos primeras sesiones en el salón de computadores los estudiantes se dedicaron a conocer la herramienta que usaron en el curso para hacer sus programas. Para esto elaboré tres programas sobre los que hicieron modificaciones y pruebas de ejecución.

Realicé la intervención durante las 15 semanas del curso durante las cuales recogí diferentes productos realizados por los alumnos para producir su nota del curso. Además para la investigación recogí datos que me permitieron analizar en mayor detalle aún los aprendizajes logrados.

## **4 LA INVESTIGACIÓN**

### **4.1. Preguntas**

Las preguntas que orientaron mi investigación fueron las siguientes:

- ¿Qué aprenden de programación de computadores los estudiantes de tercer semestre de ingeniería electrónica en un curso que combina solución de problemas y trabajo en colaboración?
- ¿Qué aprenden de resolución de problemas los mismos estudiantes?
- ¿Cómo influye un ambiente de aprendizaje en colaboración en el logro de estos aprendizajes?

#### **4.2. Metodología**

La metodología de investigación que utilicé en este estudio combina análisis cuantitativo y cualitativo de los datos. Para responder la pregunta sobre lo que se aprende de programación, hice análisis estadístico de las notas de pruebas académicas que di al grupo y análisis cualitativo de productos y grabaciones de clase. Para responder las preguntas sobre lo que aprenden los estudiantes de resolución de problemas y sobre cómo influye un ambiente de aprendizaje en colaboración en los aprendizajes logrados, realicé categorización cualitativa de grabaciones de clase, entrevistas y encuestas, usando un conjunto de categorías que surgieron a partir de alguna teoría y de los mismos datos.

#### **4.3. Participantes**

En este estudio participaron 19 estudiantes universitarios de tercer semestre del programa de pregrado de ingeniería electrónica de la Escuela Colombiana de Ingeniería. El grupo estaba compuesto por 14 hombres y 5 mujeres con una edad promedio de 18 años. El curso en el que realicé la innovación pedagógica es el primero de tres que toman los estudiantes en la línea de programación de computadores. En este curso aprenden conceptos básicos de programación estructurada, iniciando con lenguaje algorítmico y luego usando el lenguaje de programación C. La selección de los estudiantes no fue controlada por la investigación, pues participaron quienes se inscribieron por libre elección en este curso.

Analicé datos de todo el grupo y de una muestra de nueve estudiantes. Para seleccionar esta muestra tuve en cuenta aquellos estudiantes que asistieron continuamente a las clases y que no estaban repitiendo el curso. Conté con la aprobación de los estudiantes para analizar los diferentes datos recogidos de sus pruebas e intervenciones, asegurando su anonimato.

#### **4.4. Recolección y análisis de datos**

**4.4.1. Para responder la primera pregunta** sobre lo que aprendieron los estudiantes de programación de computadores recogí datos de todo el grupo en tres pruebas académicas individuales que evaluaron sus conocimientos en tres momentos diferentes del curso y de un mismo problema que desarrollaron en la quinta y en la última (decimaquinta) semanas de clase y que trabajaron organizados en los mismos grupos de todo el semestre. Esperé hasta la quinta semana para la primera toma de estos datos porque en esta semana ya el grupo había conocido una estructura general para elaborar algoritmos, algunos conceptos sobre datos, tipos de instrucciones y estructuras condicionales. Además tomé las primeras semanas para que conocieran y practicasen la dinámica esperada del trabajo en colaboración y para observar la participación de los estudiantes en los grupos que ellos mismos habían formado, haciendo ajustes en algunos grupos que no interactuaban adecuadamente.

- **Pruebas académicas individuales:** Éstas consistieron en la elaboración de un algoritmo o programa para dar solución a un problema planteado (ver anexos 1, 2 y 3). Después de obtenidas las notas comparé estadísticamente los promedios de la primera y la segunda, la segunda y la tercera, y la primera y la tercera con la prueba t-Student, buscando diferencias significativas que indicaran un incremento de los aprendizajes de los alumnos. Califiqué los programas desarrollados por los alumnos para solucionar estos problemas usando una matriz de evaluación que elaboré para este estudio y que asocia criterios y niveles cualitativos con notas numéricas (ver anexo 4). Para la definición de los criterios cualitativos tuve en cuenta las etapas del proceso de elaboración de un programa de computador (Pressman, 1998), las etapas del proceso de

resolución de problemas (Polya, 1957) y la propuesta de Perkins (1981), quien realizó un paralelo entre ambos procedimientos. Es así como definí once (11) criterios y tres niveles de bondad, que corresponden a un nivel bajo, entre 1.0 y 2.9, un nivel medio, entre 3.0 y 3.9, y un nivel alto, entre 4.0 y 5.0. Los criterios se relacionan con acciones de análisis del problema, diseño e implementación (escritura) y verificación de la solución, a saber:

- **Identifican el problema (análisis):** Los estudiantes demuestran que identifican cuál es el problema, los datos y condiciones del mismo.
  - **Elaboran un plan de solución (diseño):** Los estudiantes muestran mediante un dibujo, esquema o macro algoritmo un plan que guía la escritura del código de su solución.
  - **Manejan datos, entradas y salidas (implementación):** Los estudiantes representan, ingresan (leen), procesan y entregan (escriben) los datos del problema.
  - **Usan estructuras condicionales repetitivas y no repetitivas (implementación):** Los estudiantes usan adecuadamente estas estructuras, planteando la condición y relacionando los procesos correspondientes.
  - **Usan datos estructurados, arreglos y estructuras (implementación):** Los estudiantes usan estos tipos de datos en su solución.
  - **Usan funciones y procedimientos (implementación):** Los estudiantes usan funciones y procedimientos en sus soluciones.
  - **Hacen pruebas de escritorio (verificación):** Los estudiantes demuestran que elaboraron una prueba para comprobar que su solución entrega los resultados esperados.
- **Prueba inicial y final desarrollada en grupos:** La prueba consistió en la presentación de una situación muy sencilla narrada en un párrafo, que para su solución los obligaba a hacer la lectura de una serie de datos y diferentes cálculos matemáticos para sacar los resultados. Con la intención de que los estudiantes analizaran autónomamente el problema, les formulé una serie de preguntas que debían resolver antes del proceso de construcción de la solución (ver anexo 5).

Analicé cualitativamente tanto los trabajos escritos resultantes del trabajo en grupo como grabaciones de la interacción de los grupos mientras resolvían el problema. Los trabajos los analicé usando la misma matriz que usé en las evaluaciones individuales y categoricé las grabaciones buscando los mismos aprendizajes relacionados con la programación que indica la matriz, con el objeto de triangular hallazgos.

**4.4.2. Para responder la segunda pregunta** sobre lo que aprenden los estudiantes de resolución de problemas, recogí datos por medio de grabaciones de clase, entrevistas y encuestas y triangulé hallazgos entre ellos.

- **Grabaciones de clase:** en tres momentos durante el curso, inicio, mitad y final, hice grabaciones de las conversaciones de los tres grupos de estudiantes que constituyeron la muestra, nueve participantes en total. Cada una de las grabaciones correspondió a la resolución de un problema propuesto y fue transcrita literalmente. Analicé la información recogida categorizándola según lo que las intervenciones individuales de los participantes indicaban acerca de lo que aprendieron sobre resolución de problemas. Logré definir estas categorías clasificando primero las intervenciones de los estudiantes en los siguientes tipos de aportes a la discusión (ver anexo 6 para conocer ejemplos de cada uno de estos aportes):

- Expresar que se identifica un dato o enunciado del problema, nombrándolo.
- Pedir información a compañeros sobre los datos y el problema.
- Pedir información a compañeros sobre programación.
- Pedir la validación de lo que se expresa o propone a compañeros.
- Dar una explicación sobre un tema del curso o corregir lo que dicen compañeros.
- Expresar que se aplican conocimientos de programación en la solución del problema.
- Expresar que se está aclarando una comprensión de programación o del problema.
- Proponer escribir la solución de alguna manera específica.
- Expresar inconformidad con la propuesta de un compañero y proponer otra.

- Expresar inconformidad con la propuesta de un compañero, explicándola.
- Expresar que se realiza un proceso de verificación de la solución.

Luego, a partir de la bibliografía revisada sobre resolución de problemas, establecí relaciones entre algunos de esos tipos de aportes y los procesos que se siguen en una estrategia de resolución de problemas: análisis, diseño, implementación y verificación (Polya, 1957). Finalmente definí las siguientes categorías y subcategorías, con las que doy respuesta a mi segunda pregunta:

- **Analizan problemas**

- C1.1. Identifican un dato o enunciado del problema, nombrándolo.
- C1.2. Preguntan sobre los datos e información del problema, expresando comprensión.

- **Resuelven problemas usando la programación**

- C2.1. Preguntan sobre temas de programación, expresando comprensión.
- C2.2. Explican un tema del curso o corrigen lo que dicen los compañeros.
- C2.3. Aplican conocimientos de programación en la solución del problema.
- C2.4. Aclaran una comprensión de programación o del problema.
- C2.5. Proponen escribir la solución de alguna manera específica.
- C2.6. Realizan procesos de verificación de la solución.

- **Reconocen que un problema puede tener diferentes soluciones**

- C3.1. Expresan inconformidad con la propuesta de algún compañero, explicándola.
- C3.2. Expresan inconformidad con una propuesta y hacer una nueva.

En esta categorización no tuve en cuenta las intervenciones donde los estudiantes interactuaban con la profesora, conversaban temas ajenos a la clase, no expresaban ideas completas o no expresaban comprensiones sobre el problema y el programa.

- **Encuestas:** apliqué dos encuestas a todos los estudiantes, una al inicio y otra al final del curso. Contrasté la información inicial con la recogida al final del curso para identificar cambios en sus



concepciones sobre sus aprendizajes (ver instrumentos en anexos 7 y 8) y analicé sus respuestas usando las categorías que utilicé en las grabaciones de clase.

- **Entrevistas semiestructuradas a grupos focales:** apliqué dos entrevistas en momentos diferentes durante el curso a siete de los nueve estudiantes seleccionados como informantes en las grabaciones (ver instrumentos en anexos 9 y 10). Conformé dos grupos focales mezclando los estudiantes de los tres grupos grabados. De nuevo analicé la información suministrada usando las mismas categorías empleadas en el análisis de las grabaciones, con el fin de buscar evidencia que corroborara la información recogida de los estudiantes en las grabaciones.

**4.4.3. Para responder la tercera pregunta** sobre el efecto de un ambiente de aprendizaje en colaboración en el aprendizaje de los estudiantes utilicé los datos recogidos por medio de las encuestas finales, de las entrevistas y de las grabaciones de clase. Analicé los datos y encontré tres formas básicas en las que el trabajo en colaboración contribuyó al aprendizaje de los estudiantes, las cuales usé como categorías de análisis tomadas de los datos:

- Juntos aportan diferentes ideas.
- Juntos resuelven dudas y corrigen errores.
- Juntos logran otros aprendizajes.

## **5 RESULTADOS**

El análisis que hice de los datos mostró que los estudiantes desarrollaron aprendizajes de programación y habilidades importantes para la resolución de problemas. La innovación pedagógica parece haber creado un ambiente propicio para la discusión sobre los problemas y la construcción de comprensión acerca de los conceptos de la programación en la elaboración de soluciones. Igualmente los estudiantes reconocieron aprendizajes de tipo social como la responsabilidad en la participación en un grupo, el reconocimiento de comprensiones diversas y la tolerancia en medio de esa diversidad.

### **5.1. Aprendizajes sobre programación de computadores.**

Al emprender la respuesta a la primera pregunta de investigación, los resultados de las evaluaciones muestran, en general, que los estudiantes mejoraron sus aprendizajes de programación a medida que avanzó el curso. La tabla No.1 presenta los promedios del grupo participante a partir de las notas individuales obtenidas en las tres pruebas académicas aplicadas en momentos diferentes durante el semestre.

**Tabla No.1. Promedios de notas en pruebas académicas (n=19)**

Pruebas	Media	SD	Rango	Frecuencias en los rangos		
				5.0 - 4.0	3.9 - 3.0	2.9 - 1.0
Prueba 1	3,337	0,813	1.8 - 5.0	4	8	7
Prueba 2	3,716	1,433	1.0 - 5.0	11	2	6
Prueba 3	4,063	0,866	2.0 - 5.0	13	3	3

La tabla muestra que la media de las pruebas se incrementó cada vez, lo que indica que, en promedio, los estudiantes incrementaron sus comprensiones a medida que avanzó el curso, aunque también aumentaron las diferencias individuales indicadas por la desviación estándar. Esta medida es mayor en la prueba 2, indicando que en esta prueba las diferencias individuales de las comprensiones mostradas por los estudiantes son mayores que en las otras dos pruebas; el rango de notas más amplio en esta prueba así lo corrobora y el grupo parece dividirse en dos, un grupo en nivel alto y un grupo en nivel bajo. En la prueba 1 la mayoría de estudiantes se encuentra en los niveles medio y bajo, mientras en la prueba 3 la mayoría de estudiantes se encuentra en el nivel alto.

La tabla No.2 presenta los resultados de la comparación de los promedios de las tres pruebas, al emparejarlas para buscar diferencias significativas mediante la prueba t-Student.

**Tabla No.2. Prueba t de Student entre tres pruebas académicas (n=19)**

Pruebas comparadas	Primera prueba		Segunda prueba		t-Student	
	Media	Varianza	Media	Varianza		
Pruebas 1 y 2	3,337	0,661	3,716	2,054	1,541	*
Pruebas 2 y 3	3,716	2,054	4,063	0,749	1,457	*

Pruebas 1 y 3	3,337	0,661	4,063	0,749	3,674	***
---------------	-------	-------	-------	-------	-------	-----

\*p<0,1; \*\*p<0,01;\*\*\*p<0,001

En la tabla anterior se aprecian diferencias estadísticamente significativas entre las pruebas 1 y 2 a un nivel de significación  $p < 0,1$ ; entre las pruebas 2 y 3 a un nivel de significación  $p < 0,1$ ; y entre las pruebas 1 y 3 a un nivel de significación  $p < 0,001$ . Esto corrobora que en las pruebas académicas los estudiantes mostraron cada vez mejores desempeños, con la mayor diferencia entre los resultados de la primera y la última prueba.

La tabla No. 3 presenta un panorama de los avances demostrados por los seis grupos en que trabajaron los estudiantes en la solución del problema único desarrollado tanto al inicio como al final del curso. Indica el número de grupos que se localizaron en los rangos alto, medio y bajo de calificación en los diferentes criterios de la matriz de evaluación aplicados a las soluciones del problema.

**Tabla No.3 Análisis del problema único solucionado en dos momentos del semestre. (N = 19 agrupados en 6 subgrupos).**

	Criterios de evaluación	Solución inicial			Solución final		
		Rangos			Rangos		
		5.0 - 4.0	3.9 - 3.0	2.9 - 1.0	5.0 - 4.0	3.9 - 3.0	2.9 - 1.0
1	Identifican el problema *	4	2	0	6	0	0
2	Elaboran un plan de solución *	0	0	0	4	0	0
3	Representan y procesan datos	0	5	1	5	1	0
4	Hacen la entrada de datos	2	4	0	6	0	0
5	Hacen la salida de datos	3	2	1	4	2	0
6	Usan estructuras condicionales no repetitivas	2	4	0	4	2	0
7	Usan estructuras condicionales repetitivas	4	2	0	6	0	0
8	Usan datos estructurados arreglos y estructuras	0	0	0	4	2	0
9	Usan funciones y parámetros	0	0	0	3	2	1
10	Hacen pruebas de verificación *	0	0	0	1	0	0

\*Estos criterios de evaluación están relacionados también con el proceso de resolución de problemas.

La tabla anterior muestra que en ocho de los criterios evaluados (1, 2, 3, 4, 5, 6, 7 y 10), más grupos de estudiantes mostraron mejores resultados al final del curso, lo que indica que lograron

aprendizajes relacionados con estos criterios. En relación con los criterios 8 y 9, al momento de desarrollar la solución inicial los estudiantes no conocían aún estos temas, de modo que la tabla sólo indica que la mayoría de los grupos logró aprendizajes de calidades buena y media en estos temas nuevos del curso. En cuanto a los criterios 2 y 10, son aspectos que los estudiantes no trabajaron conscientemente en la prueba inicial, lo que sí hicieron en la final, aunque las pruebas de verificación aparecen como el aprendizaje menos logrado.

A continuación triangulo lo resumido en la tabla con datos tomados directamente de las conversaciones grabadas del trabajo en colaboración de los grupos en clase y con datos tomados de sus trabajos, en la solución del problema único.

**5.1.1. Identifican el problema (análisis).** Los estudiantes aprenden a identificar el problema y sus datos. En la solución final al problema único todos los programas muestran este aprendizaje. La siguiente ilustración muestra el análisis elaborado por los estudiantes del grupo 3 en la prueba final, donde identifican todos los datos del problema, incluidos los resultados que deben entregar:

Datos:  
 nombre de los estudiantes, de tipo carácter.  
 notas del primer tercio, de tipo Real.  
 nota promedio del curso, de tipo Real.  
 nota del quiz, tipo Real.  
 nota de la prueba corta, de tipo Real.  
 nota del parcial, de tipo Real.

Fig. 1 Grupo 3

En las grabaciones de clase final, se encuentra la conversación relacionada con este proceso de análisis elaborado por el grupo 3 donde los estudiantes analizan el problema identificando los datos y resultados que deben entregar:

**Grabación 3, grupo 3:**

- E3: Bueno los datos, el nombre y el tipo, estudiantes, entonces toca carácter...
- E1: No, ¿por qué? Ah no, bueno sí, puede ser. Entonces un carácter que se llame nombre... Ah no, porque eso lo podemos hacer.... Es que eso se le hace en la estructura.....
- E3: Exacto, si los estudiantes son de tipo carácter, es un vector. Los otros son las notas.
- E1: ¿De tercio o de semestre?

- E3: *No, sólo es de tercio. Mira hay tres.*
- E1: *Nota definitiva del tercio, ¿pero es que ahí no nos dicen de qué tercio?*
- E3: *Pues del primero.*
- E2: *Es que no son las notas finales, solamente las notas del primer tercio.*
- E1: *¿Cómo vamos a hacer? ¿Cuáles son los que no presentaron prueba? Pidámosles el carné o el nombre, algo así.*
- E3: *No, porque...*
- E1: *Tiene que ser así para saber quien no presentó la prueba...*
- E3: *En la prueba van a tener un cero, entonces comparamos la nota.*
- E2: *Que cuando sea cero no multiplique por 0.3 sino por 0.8. Pues por ahora yo digo que esos son los datos.*
- E3: *Hay que sacar el promedio, ¿cierto?*
- E2: *La función principal... entonces va a haber una función que calcule el promedio... déjame yo leo.*
- E3: *Mira, mira, mira, dice 'la nota definitiva de cada uno de los estudiantes'.*
- E1: *Qué más hay que hacer, mira: crear archivo, pedir la información, calcular la nota y escribir los resultados y ya.*

**5.1.2. Elaboran un plan de solución (diseño).** Los resultados muestran que al comienzo del curso ninguno de los grupos elaboró un esquema o plan, antes de iniciar la escritura del algoritmo solución. Al final del curso, en cuatro (4) de los seis (6) productos, los estudiantes hicieron un esquema o planteamiento inicial del problema que evidenció un proceso de diseño del programa que debían elaborar. La elaboración de este diseño se reflejó en que lograron un mejor manejo de los datos y de las estructuras condicionales en sus programas. A continuación dos ilustraciones de estos diseños en las pruebas finales:

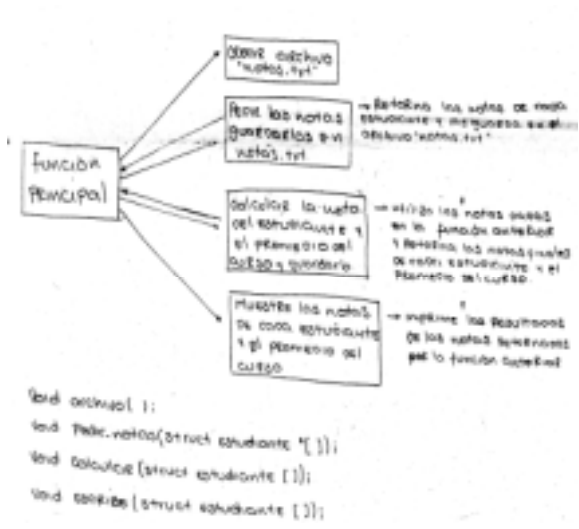


Fig. 2. Grupo 3

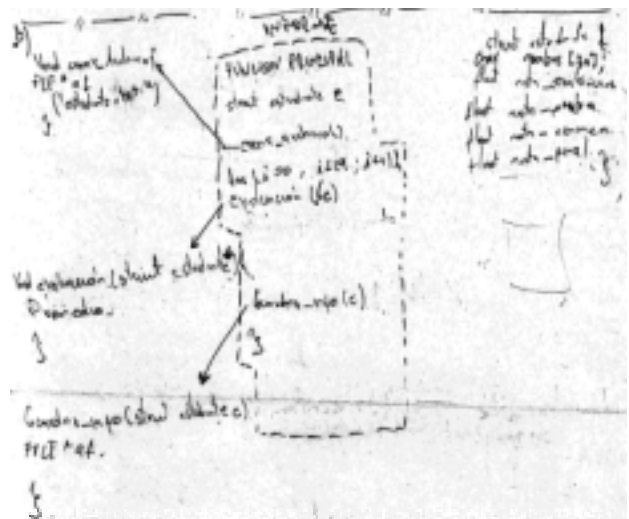


Fig. 3. Grupo 1

Y en las grabaciones de la clase final, se encuentran conversaciones relacionadas con este proceso de diseño. A continuación, los estudiantes del grupo 1 están haciendo el diseño de su solución, ilustrada en la figura 3. El estudiante E2 propone hacer un esquema de la estructura y de las funciones que tendrá su solución.

### **Grabación 3, grupo 1**

- E2: *Entonces...primero necesitamos como una estructura que sea de tamaño 19.*
- E1: *Una estructura sí.*
- E2: *Entonces en la estructura vamos a guardar... Hagamos un esquema de la estructura... entonces sería como struct estudiante...y aquí entonces...*
- E1: *No, una estructura de uno, ¿no?*
- E2: *Sí, sí, pero se hace que se repita en un ciclo de 19. Porque son 19 estudiantes. Entonces 'float' nota examen...digo nota evaluación...otro flotante nota prueba y nota examen.*
- E2: *Bueno así va a ser la estructura. Entonces una función que pida...*
- E1: *Toca crear una función que cree el archivo.*
- E2: *Bueno sí, entonces aquí por ejemplo, 'void crear archivo', entonces aquí en la función principal llamamos a crear archivo, ¿sí o no? Llamamos a crear archivo. Aquí es de tipo 'file', listo. Entonces hagamos otra función, esa que crea el archivo.*
- E1: *Después de eso tiene que pedir notas, ¿no?*
- E2: *Pero entonces sería que pida las notas y calcule de una vez.*
- E1: *Toca, uno, que pida notas, y otra función que calcule lo que necesita.*
- E2: *Ah, pero venga, es que la estructura sólo necesita la nota final, o sea que aquí faltaría algo como nota final, porque es lo que interesa. Entonces hagamos una función que, pongámosle 'evaluación', en esa función va a haber un ciclo que vaya de cero a 19,... va a hacer este cálculo y lo va a guardar de una vez ahí en nota final, ¿sí me entiende?*
- E1: *Listo.*
- E2: *Y calcula la nota final.*

**5.1.3. Representan y procesan datos (implementación).** En este aspecto los resultados finales muestran que los estudiantes lograron aprendizajes importantes. En todos los productos iniciales encuentro que aunque los estudiantes identifican y declaran algunos datos, varios de ellos se declaran del tipo incorrecto. Encuentro también inconsistencias de tipo de datos en las instrucciones de asignación; no identifican constantes y en dos (2) de ellos no distinguen entre nombre y contenido de las variables. En contraste, en cinco (5) de los programas finales, se encuentran todos los datos del problema declarados con nombres significativos y de tipo correcto. Además en los programas demuestran que comprenden los resultados de las

instrucciones de asignación, inicializan variables e identifican cuando un dato puede tener diferentes valores y los validan.

En la siguiente grabación se aprecia el momento cuando los estudiantes del grupo 1 hacen la representación de los datos del problema final. El estudiante E1 hace la declaración de los datos que requiere en su solución, el tipo que usa para cada dato es correcto y los nombres que asigna indican claramente qué información contiene cada dato.

### **Grabación 3, grupo 1.**

- E1: *Bueno la estándar... ¿Ya metió la librería?, Sí... ahora declaración de estructuras. Entonces 'struct estudiante', necesitamos el nombre también, 'char nombres', 30, 'float nota evaluación', 'nota pruebas', 'nota examen' y 'nota final'. Prototipos de funciones. Entonces, tenemos...'void crear el archivo', ¿qué sigue?...*
- E2: *Esto... ¿pedir datos?...*

**5.1.4. Hacen la entrada de datos (implementación).** Sólo en dos (2) de los productos iniciales los estudiantes identifican los datos que deben leer y los validan, en cambio en los seis (6) productos finales los estudiantes identificaron, leyeron y validaron todos los datos esperados para solucionar el problema.

A continuación un ejemplo de validación de datos encontrado en una de las grabaciones finales. En esta conversación, el estudiante E1 quiere hacer la validación de un dato con una estructura “if – else if”, pero el estudiante E2 le sugiere a su compañero que la manera correcta de validar el dato que están leyendo es con una estructura “while”.

### **Grabación 3, grupo 1.**

- E1: *Listo. Entonces ya se terminó la función, no necesitamos hacer nada más. Ahí si podíamos hacer la validación como yo lo estaba haciendo 'else if'.*
- E2: *¿Para qué?*
- E1: *Para hacer de una vez la validación. 'Else if' este entra este, si es diferente de alguno de estos dos entonces decir 'respuesta incorrecta'.*
- E2: *Ah sí, de una. Pero sabe qué, esa validación no se puede hacer ahí porque lo va sacar del programa, lo va a sacar de acá. Toca hacer un 'while' antes de aquí que se repita hasta que él ingrese el dato correcto.*

Y en efecto, en la solución final del grupo 1 encontré la validación que el estudiante E2 le sugirió a su compañero. Los estudiantes leyeron el dato “prueba” y lo validaron con un ciclo “while”.

**5.1.5. Hacen la salida de datos (implementación).** En tres (3) de los programas iniciales los estudiantes entregan todos los datos que solucionan el problema y en los tres (3) restantes sólo entregan algunos resultados. Al final, en cuatro (4) programas finales los estudiantes entregan todos los datos que solucionan el problema. En los otros dos (2) programas no encuentro todos los resultados porque el código del programa está incompleto. Por ejemplo, en la prueba final el grupo 5 elaboró un subprograma<sup>1</sup>, para escribir todos los datos resultados usando la función para escritura: “printf”.

En el siguiente extracto de una grabación, el grupo 3 está haciendo la parte correspondiente a la escritura de los resultados del programa final. El estudiante E1 menciona algunos de los datos que debe escribir la solución y junto con E3 dicen cómo los van a escribir usando para ello un condicional repetitivo “for”.

#### **Grabación 3, grupo 3.**

- E1: *¿Tú ya hiciste cuál?*
- E3: *El de los resultados, lo estoy haciendo.*
- E1: *¿Es el que muestra información?*
- E3: *¿Está bien?*
- E1: *Si, pero es que cuando esté escribiendo tiene que decir nota del estudiante 1 es tal, nota del estudiante 2 es tal...*
- E3: *Si, con un ‘for’, la nota del estudiante, el porcentaje y nada más.*
- E1: *Si, desde el contador empezando desde 1 hasta 20 que va ya imprimiendo.*

**5.1.6. Usan estructuras condicionales no repetitivas (implementación).** Los resultados muestran que al inicio sólo dos (2) programas plantean las condiciones de manera correcta así como también identifican los procesos que dependen de ellas. Al final, cuatro (4) programas usan en forma correcta estas estructuras de control, mientras que en dos (2) de ellos encuentro que repiten instrucciones innecesariamente o la condición planteada no está bien. Un ejemplo de uso correcto de estas estructuras lo encontré en la solución final del grupo 4 que por medio de

---

<sup>1</sup> Porción de código de programa que se especializa en una tarea específica.



una estructura condicional no repetitiva, “if – else if” organizan los diferentes subprogramas<sup>1</sup> que conforman el programa solución.

En la siguiente grabación el estudiante E2 propone a su compañero cómo hacer una función usando un condicional no repetitivo “si- entonces-sino”.

### **Grabación 3, grupo 1.**

- E2: *Entonces pregunta primero si presentó prueba corta o no, si presentó prueba corta [entonces] le va a preguntar nota de evaluación, y la va a guardar acá, nota de prueba corta y la va a guardar también, nota de examen tal y entonces con estos va a hacer este cálculo de nota final, ¿si me entiende?*
- E1: *Si.*
- E2: *...y se sale. Pregunta otra vez si presentó prueba corta, si no presentó prueba corta, que pregunte sólo evaluación y examen y calcula la nota final.*
- E1: *Listo.*

**5.1.7. Usan estructuras condicionales repetitivas (implementación).** En cuatro (4) de los programas recogidos al inicio encuentro un correcto uso de estas estructuras, sólo en dos (2) soluciones encuentro que los estudiantes plantean mal la condición y repiten instrucciones que no deben repetirse. Y al final, en todos los programas el uso de estas estructuras es correcto, plantean bien las condiciones, manejan bien los contadores e identifican las instrucciones que deben repetirse. El grupo 2, por ejemplo, usa en su solución final, una estructura condicional repetitiva, “while”, que les permite mantener su programa en ejecución mientras que el usuario no elija la opción 3 de salida o terminación del programa.

En la siguiente grabación encontramos dos estudiantes discutiendo acerca de dónde debe ir un ciclo, estructura condicional repetitiva, en su solución final, cada uno expone su propuesta y lo que pasaría si se hace como cada uno propone. Las dos propuestas y sus consideraciones son correctas.

### **Grabación 3, grupo 1.**

- E2: *Pero entonces ¿sabe qué? Lo que pasa es que tendríamos que mandar guardar en el archivo aquí mismo, porque es que cuando vuelva a pasar el ciclo va a llenar sobre la misma estructura porque es una sola estructura la que tenemos. O sea, si queremos hacer*

---

<sup>1</sup> Porciones de código de programa que se especializan en una tarea específica.

*otra función para guardar, tenemos que cuadrar una estructura de 19, llenarla toda y después ahí si enviársela al archivo para que la guarde.*

- E1: *Lo que yo digo es que envía la desde la función principal, o sea el ciclo hacerlo desde la función principal de forma que llega a esa posición, le manda esa posición, sale esta función y uno le manda a guardar a archivo en esa posición.*
- E2: *No, porque lo que pasa es que si usted dice que va a hacer el ciclo en la función principal, entonces ese 'i' va es en la función principal y aquí sólo va la estructura. Entró al ciclo, llamó esto, listo lo creó, llamó esta, llenó la estructura, siguiente vez guardar en archivo, guardó esa estructura en el archivo y volvió otra vez al ciclo, volvió y llamó esto, llenó otra vez la estructura, borró los datos de antes y llenó la estructura y los envió, pero igual este i no está haciendo nada ahí. Porque ahí simplemente usted va a preguntar, como tiene una sola estructura... ¿si me entiende?*
- E1: *Ah, si sí.*
- E2: *El ciclo se va a repetir es en la función. Esto todo va a ir metido en el ciclo...  $I = 0$ ,  $I < 19$  y  $I++$ . Que llama a 'evaluación', que llama a 'guardar info' y le enviamos la estructura. ¿Si o no?*

### **5.1.8. Usan datos estructurados: arreglos y estructuras (implementación).**

En relación con estos aprendizajes encuentro que en todos los productos finales los estudiantes incorporaron estos tipos de datos a su solución, tanto arreglos como estructuras. Su uso correcto en cuatro (4) de los productos finales demuestra comprensión y aprendizaje sobre tipos de datos estructurados.

Una ilustración de lo anterior la encontré en las soluciones que elaboró el grupo 1, mientras en la solución inicial usaron variables simples para representar los datos del problema, en la solución final usan estructuras de datos y arreglos.

En la siguiente grabación los estudiantes del grupo 2 están diseñando la estructura de datos que usarán en su solución final.

### **Grabación 3, grupo 2**

- E2: *Hay que declarar la estructura.*
- E1: *Una estructura nombre char, float nota 1, nota 2,.....hay que validar las notas definitiva...este es el alumno 1, este el alumno 2, pues para actualizar los datos. Hay que declarar una estructura. Pero eso va al principio. Sí, para que le diga el nombre y todo eso. Declaración de estructuras...Una puede ser el nombre, ¿cierto? Nombre de ¿cuántas? Cincuenta. Entonces nota1, nota2...cuántas notas son, el quiz, otra...*

### **5.1.9. Usan funciones y parámetros (implementación).**

Todos los grupos de estudiantes incorporaron el uso de funciones y parámetros en sus productos finales. Encontré que tres (3) de estos productos tienen errores en el uso de los parámetros y que el diseño de las funciones se

puede mejorar. Sin embargo destaco comprensiones logradas por los estudiantes en relación con estos temas de estudio y que se evidencian en sus productos: el concepto mismo de lo que es una función, el concepto de llamado de una función y el concepto de envío de datos a una función.

Las figuras 4 y 5, abajo, muestran los prototipos de funciones que elaboraron los grupos 2 y 3 en sus soluciones finales. En estos productos finales se observan los prototipos de las funciones que los estudiantes diseñaron para elaborar su solución, y se puede observar que cada función corresponde a una tarea específica que el programa realizará para encontrar la solución. Esto indica también que los estudiantes comprendieron el problema de tal manera que lograron plantear estas funciones.

```

} ; float Definitiva;
} ; int codigo;
/* Prototipos de funciones */
int menu OPCIONES (void);
void pedir_datos (void);
void menu_notas (void);
void mostrar_lista (void);
void mostrar_estudiante (void);
void promedio_grupo (void);
/* Funcion principal */

```

Fig. 4 Grupo 2

```

/* Prototipos de funciones */
void crear_archivo ();
void pedir_notas_guardar (struct notas [5]);
void calcular (struct notas [5], float promedio);
void mostrar_inf (struct notas [5], float promedio);

/* Funcion principal */
void main () {
/* Declaracion de variables */
struct notas not[5];
float promedio;
/* procesos */
crear_archivo ();
pedir_notas_guardar (not);
mostrar_inf (not);
}
/* fin funcion principal */

```

Fig. 5 Grupo 3

Y en la siguiente grabación el estudiante E1, dice cuáles serán las funciones que tendrá su programa. Cuando dice que vienen los prototipos, se refiere a las funciones que está proponiendo para su solución final.

**Grabación 3, grupo 2.**

- E1: ¿Eso es 'int'? Ah no, esto es 'float'. Pero hay que validar el dato porque sólo puede ser entre cero y cinco. Entonces nota1, nota2, definitiva... Listo, ahora si vienen los prototipos. Una para pedir los datos, otra para tener la nota definitiva y el promedio del grupo, entonces hay que hacer un menú.
- E2: Listo.

**5.1.10. Hacen pruebas de verificación.** En este aspecto sólo un (1) grupo de estudiantes en la prueba final hizo una verificación de su solución. En el resto de los productos analizados no se encuentran evidencias que muestren que los estudiantes realizaron pruebas de verificación a sus algoritmos o programas. La ilustración de esta prueba encontrada en una grabación es la siguiente, en ella, el estudiante E2 sugiere a E1 hacer la verificación o “prueba de escritorio” y E1 expresa este proceso de verificación, haciendo el seguimiento a los datos de su solución:

### **Grabación 3, grupo 1**

- E2: *Pero ahí tocaría hacer la prueba de escritorio.*
- E1: *Espera que yo... Si no, entonces sumatoria...le asigno... nota1 por 0.15 más, nota2 por 0.15 más, nota3 por 0.20 más, nota4 por 0.50...*
- ...
- E1: *Y ahora escribir...ah listo. Sale de aquí...entonces escribir...promedio, falta una para promedio...y a promedio asígnele sumatoria F sobre 19 y...hagamos la prueba de escritorio entonces: nota1, nota2, nota3, nota4, basura, basura, basura. Asignar cuatro notas: 3.0, 4.5, 5.0 y 2.0. Si nota3 es igual a cero, nada, sino entonces hace esto... Oiga se me hace que hace falta algo para que...Entonces dice si nota3 no es igual a cero, se sale de todas estas condiciones y pasa acá, con esa sumatoria le asigna todo esto, que es igual a... le asigna 3.0 por 0.15 más 4.5 por 0.15 más 5.0 por 0.20 más 2.0 por 0.50, 3.1 da. Listo, tiene esta nota, entonces a la sumatoria final le asigna cero, que la inicializamos con cero más esto y termina el ciclo. Entonces sigue 1, sigue 2, entonces vuelve con las notas y asigna...*

En el proceso de verificación anterior se puede apreciar también cómo el estudiante expresa de manera correcta sus comprensiones sobre las instrucciones, los datos y las estructuras condicionales que tiene su solución final.

## **5.2. Aprendizajes sobre resolución de problemas**

Los estudiantes reconocen aprendizajes relacionados con el análisis de los problemas, el reconocimiento de la programación como herramienta para resolverlos y el de que un problema tiene varias soluciones. Corroboré estos hallazgos en las grabaciones de clase, encuestas y entrevistas. En la siguiente tabla presento los resultados del análisis de frecuencias hecho a las intervenciones individuales de los estudiantes en las conversaciones grabadas, según las categorías y subcategorías propuestas para este análisis.

**Tabla No. 4 Frecuencias de las intervenciones de los estudiantes por categorías (n=9 individuos en 3 grupos)**

	Analizar problemas				Resolver problemas usando la programación												Un problema y diferentes soluciones				Total
	C1.1		C1.2		C2.1		C2.2		C2.3		C2.4		C2.5		C2.6		C3.1		C3.2		
	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	
E1	31	19	9	6	16	10	7	4	27	17	2	1	36	22	4	3	21	13	8	5	166
E2	28	15	16	8	10	5	6	3	48	25	1	1	52	26	3	2	14	7	16	8	199
E3	6	43	1	7	1	7	1	7	0	0	0	0	3	22	0	0	2	14	0	0	14
E4	34	20	5	3	3	2	21	13	30	18	0	0	56	33	0	0	14	8	5	3	168
E5	17	20	15	18	8	9	1	1	8	9	1	1	19	22	2	2	9	11	6	7	91
E6	28	27	19	18	22	21	1	1	5	5	0	0	19	20	0	0	4	4	4	4	106
E7	39	27	17	12	7	5	3	2	31	22	1	1	30	22	2	1	10	7	1	1	144
E8	25	24	7	7	3	3	8	8	17	16	0	0	28	28	0	0	13	13	1	1	104
E9	31	31	11	11	4	4	2	2	11	11	2	2	26	26	0	0	11	11	1	1	99

**Convenciones:**

- C1.1. Identifican un dato o enunciado del problema, no mbrándolo.
- C1.2. Preguntan sobre los datos e información del problema, expresando comprensión.
- C2.1. Preguntan sobre temas de programación, expresando comprensión.
- C2.2. Explican un tema del curso o corrigen lo que dicen los compañeros.
- C2.3. Aplican conocimientos de programación en la solución del problema.
- C2.4. Aclaran una comprensión de programación o del problema.
- C2.5. Proponen escribir la solución de alguna manera específica.
- C2.6. Realizan procesos de verificación de la solución.
- C3.1. Expresan inconformidad con la propuesta de algún compañero, explicándola.
- C3.2. Expresan inconformidad con un propuesta y hacen una nueva.

En la tabla anterior se aprecia que los tipos de intervenciones que más frecuentemente ocurren en el proceso de resolución de los problemas propuestos se relacionan con la categoría de análisis: pedir información sobre los datos y el problema e identificar un dato o idea importante, nombrándolos. Para cinco (5) de los nueve (9) estudiantes participantes en los trabajos en colaboración grabados, el porcentaje de intervenciones relacionadas con analizar los problemas identificando sus datos y hechos es mayor a un tercio del número total de sus intervenciones. Por otro lado, para seis (6) de los nueve (9) estudiantes el número de intervenciones relacionadas con la categoría de aplicar conocimientos de programación en la solución de problemas es de los más altos en relación con el número total de intervenciones. Dentro de esta categoría para los nueve (9) estudiantes el número de intervenciones relacionadas con proponer escribir la solución de alguna manera específica es también de las más frecuentes en relación con el número total de intervenciones.

Los otros tipos de intervenciones que también se presentan son los relacionados con las subcategorías de expresar inconformidad con la propuesta de algún compañero, explicándola o proponiendo otra. Todos los estudiantes hacen intervenciones de estos tipos y aunque en la tabla anterior se observa que este número de intervenciones no es tan alto en comparación con las intervenciones relacionadas con el análisis y el uso de la programación, las consideré importantes porque sustentan las manifestaciones de los estudiantes en las encuestas y entrevistas en relación con el hecho de comprender que un problema puede tener diferentes soluciones. Cuando un estudiante discute con sus compañeros la manera de solucionar el problema, observa otras soluciones o expresa su comprensión comparándola contra las de sus compañeros, lo que también está entendiendo es que existen diferentes maneras de comprender y de solucionar un problema.

A continuación triangulo estos hallazgos con los datos recogidos por medio de las encuestas y entrevistas y presento ejemplos ilustrativos.

### **5.2.1. Analizan problemas**

En las encuestas encontré información relacionada con los aprendizajes logrados, en las respuestas a las preguntas sobre lo que aprendieron a hacer y sobre otros aprendizajes logrados en el curso. Con respecto a esta primera categoría, en las encuestas iniciales ningún estudiante menciona procesos de análisis o procesos relacionados como la organización e identificación de datos. En cambio al final, siete (7) de los dieciséis (16) estudiantes encuestados mencionan organizar ideas, analizar problemas y manejar información, como aprendizajes logrados. Dicen cosas como las siguientes: [Aprendí] *“manejo de datos”* (E12); *“... a organizar ideas y buscar maneras prácticas para solucionar problemas”* (E10); *“...a analizar más los problemas planteados”* (E2). Además, *“entendí mucho mejor el manejo de la información y eso me permitió hacer mejor uso de ésta”* (E6).

También en las entrevistas, las respuestas de cinco (5) de los siete (7) estudiantes interrogados sobre lo que aprendieron y sobre la utilidad de los problemas en este proceso muestran que reconocen como aprendizajes logrados en el curso algunos que relaciono con el proceso de análisis necesario para la resolución de problemas. Identifican, por ejemplo, el problema, los datos y aspectos asociados, como la organización de las ideas: “[Aprendí] *a organizar las ideas y...uno resuelve muchas cosas pero en desorden, no con un método ordenado como lo que hacíamos en clase*” (E3) y “[aprendí] *a discutir sobre un problema... porque la vida es un solo problema, son muchas cosas que hay que discutir. Uno no puede resolver las cosas sin saber por dónde se empieza*” (E4). Una estudiante indicó que “... *al principio criticaba, porque yo venía acostumbrada con otro método... en el colegio... a uno le decían 'sume dos números, sume tales matrices'... Con E6 decíamos por qué en los otros grupos [clases de programación] les dicen 'llene una matriz de tal forma y multiplíquela',...por qué no, si en los parciales seguramente nos irá a salir eso... Yo ahorita prefiero estos problemas que uno se complica más porque a uno no le dan las cosas tan masticadas como decir 'sume dos más dos'... es una situación polémica que a uno le toca analizar...qué es lo que realmente uno tiene que hacer, porque uno no se va a enfrentar todos los días con 'multiplíqueme eso', sino uno se va a enfrentar con 'multiplíqueme un número real con un número imaginario', ya se vuelven situaciones problema que uno tiene que resolver no solamente en Programación*” (E7). Otro alumno dijo que “*un problema que uno le dé a una persona, ella lo puede solucionar en su cabeza, pero lo difícil es saber cómo organizar las ideas para que el programa le funcione bien*” (E2).

En todas las grabaciones de clase, los estudiantes, algunas veces por medio de las preguntas de sus compañeros, identifican los datos de los problemas. He aquí algunos ejemplos:

**Grabación 1, grupo 1.** El estudiante E1 identifica los datos del problema:

- E1: *Ese es el primer problema...Nota 1 vale el 15%. La nota 2 vale... también el 15 %, la nota 3 vale el 20% y la nota 4 el 50%.*

**Grabación 1, grupo 2.** El estudiante E1 responde a su compañero e identifica los datos del problema:

- E2: *Bueno, debe mostrar la nota definitiva del tercio de cada uno, pero ¿y el resto de notas? O ¿esa es la nota definitiva ya?*
- E1: *No, la de cada tercio. Tiene que primero darle la de cada tercio y después que le dé la definitiva.*

**Grabación 3, grupo 3.** El estudiante E3 identifica datos del problema:

- E3: *Exacto, si los estudiantes son de tipo caracter, es un vector. Los otros son las notas.*
- E1: *¿De tercio o de semestre?*
- E3: *No, sólo es de tercio. Mira hay tres.*
- E1: *Nota definitiva del tercio, pero es que ahí no nos dicen de qué tercio.*
- E3: *Pues del primero.*

### 5.2.2. Resuelven problemas usando la programación

En las encuestas también encontré información sobre aprendizajes relacionados con el uso de la programación como una nueva herramienta para resolver problemas en las respuestas a las preguntas sobre qué se aprende y para qué piensa que le servirá saber programar. Sobre lo que aprendieron, al final del curso siete (7) estudiantes dijeron específicamente que aprendieron a solucionar problemas por medio de programas. Algunas ilustraciones de estas respuestas son: “Aprendí a aplicar un razonamiento a través de un lenguaje específico para dar un cuerpo a la idea desarrollada representada en un programa” (E3); “... a solucionar problemas por medio de programas” (E14); “... a hacer programas, diseñar algoritmos para solucionar problemas” (E16) y “... a solucionar problemas modularmente, solucionar problemas algorítmicamente” (E7). Y sobre la utilidad de la programación seis (6) de los estudiantes encuestados dicen que encontraron en la programación una forma distinta para razonar y solucionar problemas, expresando literalmente la idea con palabras como: ‘otra forma...’, ‘una forma distinta...’ y ‘una manera alterna...’.

Los resultados de las entrevistas me ayudan a sustentar la información encontrada en las encuestas para esta categoría de solución de problemas con la programación. A las preguntas sobre lo que aprendieron en el curso cinco (5) de los estudiantes indican que aprendieron a resolver problemas aplicando la programación. Lo dicen expresando que ahora saben que la programación les sirve para resolver problemas de una manera ordenada y que ahora conocen



otra forma de resolver problemas. Para tres (3) estudiantes, aprender a desarrollar un problema significa, además, poder aplicar la habilidad en otros contextos: “... orden[ar] qué es lo que voy a hacer y ahí lo hago... poder utilizar diferentes medios para resolver problema; como que uno empieza a aplicar todas esas cosas indirectamente en la vida de uno” (E7); “...cuando yo estoy en cálculo, que me ponen un problema supercomplicado y yo no encuentro la forma de hacerlo, como que ya mecánicamente uno empieza a aplicar lo que dice siempre debajo de los problemas [de programación]: cuál es el problema, cuáles son los datos del problema, todo eso. Yo aplico eso sobre todo para materias como cálculo y física, donde hay que resolver problemas... antes uno también tenía la idea, pero uno no lo conocía como programación” (E8); “... no tenía ni idea de qué era la programación. Yo veía a la gente con sus programas pero no me imaginaba qué era y llegar a este estado de saber que tengo unas herramientas para diseñar una respuesta a problemas, porque nosotros ya lo podemos aplicar no solo a la electrónica, sino también a problemas pequeños, empresariales, cosas que ya sabemos hacer, que antes no sabíamos nada de un computador” (E2).

Por otro lado las frecuencias de intervenciones en las grabaciones de clase muestran que dos de los tipos más frecuentes durante la resolución de los problemas son las que están relacionadas con la aplicación de la programación en la elaboración de la solución: aplicar conocimientos de programación en la solución de problemas y proponer escribir la solución de alguna manera específica. Los estudiantes hacen propuestas para la elaboración de la soluciones incorporando los conceptos de programación, como se ve en los siguientes ejemplos.

**Grabación 2, grupo 1.** El estudiante E1 propone cómo escribir la solución usando los ciclos, como conceptos de programación:

- E2: ¿Cómo sería otra forma de medir áreas?
- E1: Porque sería decir ‘ahora mídame las longitudes de la oficina... ¿tiene rectángulos? Sí... ¿Cuántos tiene? Tres... Déme las longitudes de esos tres rectángulos... Se acaba ese ciclo y entonces ¿tiene círculos?...no...Pum, se acaba ese ciclo... ¿Tiene trapecios? Sí ¿Cuántos? Dos...dícteme las longitudes, le dicta las longitudes, le saca el área y eso lo va sumando en otro lado y así con las cuatro figuras.

**Grabación 2, grupo 2.** El estudiante E1 expresa cómo escribe la solución, identificando los tipos de las variables que necesita (flotante):

- E3: *¿Así?*
- E1: *Sí, porque eso ya lo preguntamos, ese es el dato que solicitamos porque aquí en la función principal lo que necesita es acumular para sumar todas las áreas; entonces necesitamos una variable que equivalga a todas las oficinas, otra variable que sea la cocina, otra variable que sea el baño de hombres, otra que sea qué...ah sí, seis, ya están, estas son de tipo flotante.*

**Grabación 3, grupo 2.** El estudiante E1 le cuenta a su compañero cómo usar una estructura del lenguaje de programación:

- E2: *¿Usted sabe usar eso del switch?*
- E1: *Pues usémoslo acá.*
- E2: *¿Y es fácil o es igual?*
- E1: *Pues es fácil cuando el número es 1, 2, porque cuando va condicional que mayor, igual...entonces coloca switch, coloca entre paréntesis lo que quiere comparar, lo que queremos comparar es 'opc', y ya abrimos un corchete y eso quiere decir que cuando 'opc' sea igual a tal, entonces colocamos case en caso de que sea igual a 1, esto quiere decir en caso de que opción sea igual a 1, dos puntos y listo ya, ahí sigue...Pero acá se nos olvidó mostrar el menú...Entonces acá sería 'opc' es igual a menú, ahora si switch case 1 ...cierro. Lo que uno hace es colocar 'break', entonces ahí se sale, hasta ahí coge el caso 1. Ahora sigue case 2 igual, dos puntos...*

### **5.2.3. Reconocen que los problemas pueden tener diferentes soluciones**

En las encuestas encontré información relacionada con esta tercera categoría de aprendizaje, en las respuestas a las preguntas sobre lo que aprendieron y sobre otros aprendizajes logrados. Al inicio del curso ningún estudiante menciona este aspecto sobre los problemas, en cambio al final cinco (5) estudiantes expresan que un problema puede tener '*formas diferentes*' y '*nuevas formas*' de solucionarse y que puede haber '*varias y diferentes*' soluciones para un problema.

Las entrevistas sustentan estos resultados. Encontré información que muestra que cinco (5) de los siete (7) estudiantes entrevistados expresan de diferentes maneras que un problema puede tener diferentes soluciones. A continuación la ilustración de algunas de estas expresiones:

[aprendí] "*a buscarle a los problemas no sólo una solución sino varias soluciones. Saber que hay muchas maneras de resolver*" (E3); "*yo aprendí a discutir sobre algunos problemas porque todo el mundo piensa diferente en cuanto a la solución de algo, entonces a veces uno no se*

*entiende, entonces como a tratar de llegar a un acuerdo para hacer las cosas bien, o sea, tomando lo mejor que cada uno tiene” (E6).*

Las grabaciones de clase muestran que los estudiantes cuando expresan inconformidad con la propuesta de algún compañero, explicándola o proponiendo otra, observan diferentes propuestas de solución, las siguientes grabaciones ilustran estas situaciones.

**Grabación 1, grupo 1.** Los estudiantes están juntos construyendo la solución, pero cuando E2 le hace una pregunta mostrando su inconformidad, E1 se da cuenta que su solución no es la misma de E2:

- E1: *Ahí si ya se saca la nota que es... sumatoria.*
- E2: *¿O sea que toca volver a escribir todo eso?*
- E1: *Ah, pero ahí... si es que yo lo estoy haciendo diferente. Ahora miramos a ver... Esa sumatoria le asigno no tal por 0.15 más nota2 por 0.15 más nota4 por 0.70, listo.*
- E2: *Si no tiene, entonces la nota3 toca asignarle...*
- E1: *Espera, porque yo me estoy confundiendo.*

**Grabación 2, grupo 3.** E1 propone una manera de escribir la parte de la solución donde piden los datos, pero E2 le dice que tiene pensado hacerlo de otra manera y dice cómo:

- E2: *Acá, pidamos la base.*
- E1: *¿Cuánto mide la base? Entonces toca empezar cuánto mide la base del área de la cocina, cuánto mide la base del área de las oficinas.*
- E2: *Es que yo tenía pensado... dentro de donde uno las va llamando decir: ‘la cantidad del área de la cocina es’, entonces llama uno la función que calcula el valor y lo guarda, para no hacer todo el proceso sino llamar la función de cada uno.*

**Grabación 3, grupo 2.** E2 piensa enviar un dato a una función, pero E1 piensa que no es necesario porque en cambio de eso usará un archivo:

- E2: *Hay que declarar la estructura y enviarla.[a una función]*
- E1: *No eso no es necesario porque ahí la podemos declarar y ahí le podemos preguntar también. Ahí no sería necesario eso.*
- E2: *Pero queda mejor declararlas y enviarlas a la función.*
- E1: *Pero para qué se la mandamos si hay mismo la podemos pedir y grabarla de una vez y no hay necesidad de declarar eso.*
- E2: *Pero igual cuando van a sacar los promedios y eso hay que mandárselas.*
- E1: *No. Porque se puede cargar el archivo y ahí carga todo, abre el archivo y ahí si graba todo. Bueno pero como quiera.*

**5.3. Cómo influye el aprendizaje en colaboración en el logro de los aprendizajes de programación y de resolución de problemas.**

Los resultados muestran que los estudiantes durante la intervención encontraron un ambiente que les permitió aportar ideas, expresar y corregir sus comprensiones, así como también, reconocen que lograron otros aprendizajes como el aprender a trabajar en equipo, aprender a escuchar y a coordinar con varias personas el logro de un objetivo común.

### **5.3.1. Juntos aportan diferentes ideas.**

Según los datos recogidos, los estudiantes reportan como el principal aporte del trabajo en colaboración la oportunidad de compartir y enriquecerse con diferentes ideas y algunos relacionan esta oportunidad con el hecho de ver diferentes soluciones a los problemas y hacer mejor el trabajo.

En los datos recogidos por medio de las encuestas finales, catorce (14) de los dieciséis (16) estudiantes manifestaron que el trabajar con sus compañeros les permitió aprender más porque ven diferentes puntos de vista y diferentes formas de resolver un problema. Dicen cosas como las siguientes: *“Se complementan los conocimientos y sobre todo sabe uno que existen otras posibilidades [de solución]”* (E5); *“se comparten ideas y otros puntos de vista”* (E8); *“compartir ideas y ver otras soluciones”* (E11) y *“aportamos ideas diferentes”* (E13). Sólo un estudiante, de los dieciséis encuestados, manifestó que no le gustaba trabajar en grupo y su razón fue que *“en grupo no se trabaja”*. También manifestó que el trabajo con sus compañeros no le ayudó a aprender más porque *“en muchas ocasiones diferimos de opiniones, entonces no se avanza”* (E14). Sin embargo, esta razón contrasta con las que expresaron los otros quince (15) estudiantes para explicar su gusto y el beneficio que les trae el trabajo en grupo: *“[en clase me gusta trabajar en grupo porque] las ideas de los demás aportan a la calidad del trabajo”* (E1); *“es mejor recibir varias ideas para que los trabajos sean mejores”* (E2) y *“hay mas aportes e ideas para realizar el trabajo en cuestión”* (E7). De acuerdo con los datos de las encuestas, cinco (5) de los estudiantes mencionan que gracias al trabajo con sus compañeros lograron

comprender mejor temas como diseño de los programas, funciones, parámetros, arreglos y el lenguaje de programación.

En las entrevistas también encontré información que está relacionada con este aspecto, en cinco (5) de los estudiantes entrevistados. Dos (2) estudiantes responden, a la pregunta sobre otros aprendizajes logrados: “*Compartiendo ideas con mis compañeros, trabajando en grupo, escuchándolos a todos [aprendí] a buscarle a los problemas no sólo una solución sino varias soluciones. Saber que hay muchas maneras de resolver*” (E3); y “*yo aprendí a discutir sobre algunos problemas porque todo el mundo piensa diferente en cuanto a la solución de algo; entonces a veces uno no se entiende; entonces como a tratar de llegar a un acuerdo para hacer las cosas bien, o sea, tomando lo mejor que cada uno tiene*” (E6). Un estudiante a la pregunta sobre si le gustó trabajar con sus compañeros respondió: “*Yo diría que seguir trabajando en grupo es lo mejor porque comparte uno muchas ideas, se da cuenta de que hay muchas soluciones y va a entender mejor las cosas, porque va a tener a alguien más cercano a quien preguntarle*” (E2).

En todas las grabaciones recogidas, los estudiantes permanentemente están trabajando sobre la solución de los problemas, identificando datos y proponiendo o aportando ideas diferentes para construir la solución. Las siguientes intervenciones ilustran interacciones en las que los estudiantes ven las soluciones de sus compañeros.

**Grabación 1, grupo 1.** Aquí un estudiante mira la solución propuesta por otro estudiante compañero de grupo y logra entenderla y aceptarla:

E1: *No. Toca asignárselo a otra casilla... y una vez hechos todos estos procesos de multiplicar, ahí si se suman todas esas...*

E2: *Ah no, pues a la definitiva aumentarle el contador para que me lo meta en otro estudiante, ¿si me entiende? Como son 19. Claro así más, más [expresión en lenguaje de programación]*

E1: *Pero es que ahí le falta a usted sacar las notas de cada estudiante.*

E2: *Es que ahí saqué la primera.*

E1: *¡Ah, ya ya! Sí claro... Pero le toca en otro numeral porque o sino se le borra ese contador, después le asigna a esto todo se lo borra, tiene que asignarle otro.*

E2: *¿Aquí otro?*

E1: *Otro, sí. Humm, ya, ya entendí que fue lo que usted hizo.*

**Grabación 3, grupo 2.** Un estudiante piensa manejar el tipo de dato de una manera y su compañero se decide por otra:

E2: *El código en la estructura...*

E1: *Falta ponerle tipo 'char'.*

E2: *¿De cuánto el código?*

E1: *Solamente dos letras, entonces hay que dejarla de tres porque siempre hay un carácter a lo último...*

E2: *Pongám osle de cinco.*

E1: *No porque solamente es hasta 19, son dos números que se van a digitar.*

E2: *No, es que yo lo estaba pensando como nuestro código de estudiantes...*

E1: *No, para esa gracia un entero.*

E2: *Pues si es sólo de dos es más fácil con el entero. Pero es que yo estaba pensando en el código largo, el número del carné...*

E1: *No pues yo lo hago con el entero y usted con el char.*

E2: *Bueno.*

**Grabación 2, grupo 3.** Aquí los estudiantes discuten la manera de hacer la solución; un estudiante propone hacerlo de una manera diferente a la del resto del grupo pero no logra convencer a sus compañeros y decide seguir haciéndolo como piensa:

- E1: *¿Cuántas funciones son? Una, dos, tres, el área.*

- E3: *¿Sabes cómo podríamos hacer? Llamar a estos sectores para los valores de estas áreas y restarlas*

- E2: *¿De qué estás hablando?*

- E3: *Mira para hallar esto. Llamar esa función por los valores de esas áreas.*

- E2: *O sea, primero calculamos estas tres y luego...*

- E3: *Y después sacar todas estas y las restamos. Es más fácil.*

- E2: *Pero entonces ahí sí necesitaríamos todas las medidas, ¿me entiendes?*

- E3: *No, porque digamos la medida de esto es fácil, es la medida del triángulo...*

- E2: *No, pero...*

- E3: *Mira es que es este grande te m enos todos estos...*

- E2: *Por eso, entonces necesitamos saber cuánto mide esto, cuánto mide esto y seña pedir otros datos.*

- E2: *Pero igual si los vamos a pedir.*

- E1: *Y después restarle todo eso. Ven... ¿cuántas variables son? Una por cada área, me imagino.*

- E3: *Sí, son muchísimas.*

- [discuten por el número de variables]

- E2: *...pero primero tienes que sacar esta y esta.*

- E3: *Ah no, yo voy a hacer lo de las oficinas [se refiere al cálculo de una áreas] suponiendo que ya tenemos estas. [otras áreas] [el estudiante insiste en su propuesta inicial]*

- E2: *No muchachos, mejor así no. El problema es que sino lo hacemos, acá nos toca hacer...no sé cómo hacer... otro rectángulo aquí...*

- E3: *Un trapecio. Si es para sacar eso, yo pienso que es mejor así. O no, tranquilas, ya!...Sigan frescas que yo sigo por aquí. [el estudiante decide seguir con su propuesta]*

**Grabación 3, grupo 3.** Dos estudiantes comparan sus soluciones:

- E3: *¿Cómo llameste la función de pedir información?*

- E1: *Pedir notas o algo así.*

- E3: *¿Notas?*

- E1: *Tienes que abrir el archivo. Tienes que antes de que empieces a hacer todo eso, escribir AP igual a...*
- E3: *Es que yo no lo hice para que guardara todavía.*
- E1: *Pues de una vez. ¿No?*
- E3: *No mira le envío la información a esta que si la graba.*
- E1: *Ah bueno sí, también se puede...pero yo no hago otra función. Es mejor de una vez porque no tengo que enviar datos.*

### **5.3.2. Juntos resuelven dudas y corrigen errores.**

Según los datos recogidos, otro de los aspectos que los estudiantes más reportan en relación con el aporte del trabajo en colaboración es la oportunidad de resolver dudas y corregir errores. En las encuestas siete estudiantes (7) responden que trabajar con sus compañeros les ayudó a aprender más porque así se resuelven dudas o se corrigen errores: *“Somos personas que se nos presentan la mismas preguntas y llegar a una solución nos hace crecer más en el aprendizaje”* (E1); *“Ellos [los compañeros] le aclaran a uno dudas”* (E2); *“A veces uno no se da cuenta de los errores que tiene y otra persona sí”* (E4); *“Siempre hay alguien que sabe lo que yo no sé”* (E10).

En las entrevistas, a la pregunta de si les parece que trabajando con sus compañeros aprenden, cuatro (4) estudiantes responden que los compañeros ayudan a despejar dudas y corregir errores: *“Sí [aprendo]. Es un refuerzo. Es como cuando uno prepara un taller o un quiz; a pesar de que uno lo va haciendo independientemente, cuando uno se queda o hay una duda, hay alguien que le despeja esa duda, o si no le refuerza o le da otros conceptos”* (E4); *“Así yo aprendo más; es que tres me parece que es un número bueno para estudiar. Uno sólo de pronto no puede ayudarse y uno se complica mucho... yo estudiando sola y no entiendo y lo intento hacer... y me desespero... en cambio con ellos me ayudan, yo me relajo y sigo...”* (E7); *“...ahí uno se da cuenta de los errores que comete. Muchas veces uno está escribiendo, entonces uno le dice al otro no mira eso ahí no es entero puede ser un real... Nos pasa que comenzamos con la intención de que cada uno lo haga en su cuaderno hasta que comenzamos y comparamos no mira ahí está mal; entonces nos ayudamos.”* (E8).

También en las nueve (9) grabaciones de clase realizadas, encuentro intervenciones donde algún estudiante corrige su comprensión con la ayuda de sus compañeros.

**Grabación 1, grupo 1.** Un estudiante corrige con la ayuda de sus compañeros una operación matemática que haría en el procedimiento:

- E1: *Ese es el primer problema...Nota 1 vale el 15%. La nota 2 vale... también el 15 %, la nota 3 vale el 20% y la nota 4 el 50%.*
- E2: *Entonces to ca hacer cuatro notas y una se multiplica por esta, por esta y por esta...Pero entonces ¿cómo sacar la definitiva? que las multiplique, las suma y las divide?*
- E1: *No, porque o sino da una nota muy bajita...mire .. una calculadora...claro si usted divide cuatro ...*
- E2: *Si usted tiene cuatro notas, ¿cómo saca su nota definitiva?*
- E1: *Las multiplica por cada porcentaje y las suma.*
- E2: *Las suma, ¿cierto?*
- E1: *Las suma pero nada más, pero no se dividen.*
- E2: *Si, si, las suma pero no se dividen tiene razón, si.*

**Grabación 1, grupo 2.** Un estudiante corrige a su compañero y le dice cómo funciona un ciclo:

- E3: *Bueno si empieza desde 19 pero en escribir ¿qué colocamos?*
- E2: *Digite su nota...*
- E3: *¿19 veces así?*
- E2: *No porque después de eso se devuelve. Claro, usted cuando se devuelve acá, este le vuelve a aparecer “escriba su nota”, 19 veces, “escriba su nota”, es diferente la nota. El número va cambiando.*

**Grabación 3, grupo 3.** Una estudiante ante la duda de su compañera, afirma su comprensión:

- E1: *Entonces la [función] que muestra información.*
- E2: *Pero crea el archivo.*
- E1: *Ya, mira. Es que es un tricitico [sic] ahí y ya quedó listo.*
- E2: *Pero ¿por qué lo creaste como lectura? Ah no, mentiras.*
- E1: *¿Escritura? No, por eso lectura. Lectura es con R.*

En la misma grabación una estudiante expresa una nueva comprensión.

E1: *Si, pero es que cuando esté escribiendo tiene que decir nota del estudiante 1 es tal, nota del estudiante 2 es tal...*

E3: *Si, con un ‘for’, la nota del estudiante, el porcentaje ese y nada más.*

E1: *¿El contador empieza desde 1 o desde cero? ¿El estudiante cero tiene apuntador? ¿No es mejor empezar desde 1 y acá poner otro contador?*

E2: *Los vectores inician desde cero.*

E1: *Oye, eso si no me lo sabía, que siempre empezaban en cero.*

### 5.3.3. Juntos logran otros aprendizajes.

En relación con el trabajo en colaboración, los estudiantes mencionan otro tipo de beneficios que les trae esta forma de trabajo en clase, como seguridad y confianza, aprender a escuchar, aprender a trabajar en grupo y cumplir objetivos. Sin embargo, algunos también expresan que



aprenden a seleccionar sus compañeros de trabajo, porque reconocen que al comienzo es incómodo trabajar con estudiantes que no conocen.

En las encuestas nueve (9) estudiantes mencionan que lograron aprendizajes de sus compañeros como la responsabilidad, trabajo en equipo, tolerancia y orden: “[aprendí] *a trabajar en equipo y poder coordinar con más personas un mismo fin*” (E10); “*el trabajo en grupo, el ser responsable y cumplir con los objetivos*” (E2) y “*a trabajar en equipo y ser más humano*” (E15).

En las entrevistas un estudiante dijo: “[aprendí] *a tener mucha paciencia con mis compañeros. Yo al principio me desesperaba por cualquier cosa*” (E5).

Y también en las entrevistas, cuatro (4) estudiantes mencionan que aprenden a trabajar en grupo. Al definir lo que consideran como trabajo en grupo en su curso de programación, ellos mencionan la necesidad de que todos comprendan el trabajo de los demás para lograr hacer bien los programas: “*Hacer un programa tiene que ser secuencial. Yo puedo hacer una parte, pero si no concuerdan los datos [con los de los demás], no te va a salir. Entonces uno tiene que conocer qué han hecho tus compañeros para saber si yo lo puedo utilizar... cada uno tiene que saber qué hizo y cómo lo hizo*”. (E1); “*uno sí practica lo que es cuando es en grupo, no como en Cálculo, que uno dice trabajamos en grupo pero resultamos trabajando solos, porque cada uno se hace aparte; acá cada cosa va a estar articulada con otra. Hay que saber hacer de todo, porque si no, no vamos a poder hacer lo que nos toca*”. (E3); y “*A pesar de que la persona tenga otra idea o concepto del tema que estén trabajando, siempre es lo mismo, porque siempre es el mismo problema. Se cuestionan y depende del grupo si se quiere llegar a estar bien. Si uno forma un grupo, no todos saben, sino que se complementan y otros refuerzan y otros aceptan sus errores, pero pueden caminar hacia el mismo objetivo todos*”. (E6).

## **6 DISCUSIÓN**

Los hallazgos de la investigación sobre mi intervención pedagógica muestran que los estudiantes lograron aprendizajes de programación relacionados con la representación y proceso de los

datos, el uso de estructuras de control (condicionales repetitivos y no repetitivos), y el diseño y uso de subprogramas (funciones y procedimientos). Durante el curso, los estudiantes encontraron la necesidad y la oportunidad de aplicar los aprendizajes que iban logrando en la elaboración de los programas que solucionaron los problemas propuestos. Y en el trabajo con sus compañeros pudieron expresar tanto sus comprensiones como sus dudas para afianzar o corregir aprendizajes. De acuerdo con los resultados, aplicar los conocimientos en la elaboración de soluciones a problemas sumada a la interacción con otros permitió a los estudiantes lograr los aprendizajes de programación actuando como protagonistas del proceso.

En particular sobre el diseño y uso de subprogramas (funciones y procedimientos), todos los estudiantes lograron dos aprendizajes importantes: primero, plantear sus soluciones en términos de subprogramas y segundo, hacer los programas completos haciendo uso de ellos. Las funciones y procedimientos son porciones de código de programa que se especializan en alguna tarea específica, también conocidos como subprogramas. Aunque el diseño de todas las funciones elaboradas por los estudiantes no fue el óptimo, si es importante resaltar que el sólo hecho de hacer sus programas usando funciones es ya un aprendizaje importante en el proceso de aprender a elaborar programas con diseño modular<sup>1</sup>.

Uno de los procesos importantes tanto en la elaboración de programas como en la resolución de problemas y acerca del cual no encontré evidencia suficiente que me permita afirmar que los estudiantes lograron aprendizajes, es el proceso de verificación de programas. En este proceso los estudiantes tienen la oportunidad de revisar no sólo su solución sino también sus comprensiones. Pero los estudiantes, desafortunadamente, daban por terminada su labor cuando escribían la última línea del programa solución. Este resultado me permite reflexionar sobre los objetivos de aprendizaje de un primer curso de programación de computadores llegando a comprender que no es conveniente orientar el curso únicamente hacia el aprendizaje de un

---

<sup>1</sup> Modularidad es una estrategia de diseño y construcción de programas de computador

lenguaje de programación o la escritura de programas (Villalobos, et al., 2005) sino que es necesario promover también el aprendizaje de estrategias de planeación, diseño y verificación de los programas e insistir en su práctica desde el inicio del aprendizaje de la programación.

Por otro lado, los hallazgos de este estudio muestran que en relación con la resolución de problemas, los estudiantes aprendieron a analizar los problemas, reconocer la programación como una herramienta que les permite solucionar problemas y observar diferentes soluciones. Para el análisis de los problemas, los estudiantes se vieron enfrentados a diferentes situaciones que los obligaron a realizar procesos cognitivos de análisis y comprensión de información. Identificar cuál era el problema, identificar los datos y condiciones del problema fue un ejercicio repetitivo que consideré necesario promover e insistir permanentemente por medio de las preguntas que incluí en cada nuevo problema propuesto. Aunque al inicio del curso los estudiantes no prestaron la atención requerida a estos procesos de análisis previos al diseño y la elaboración, a medida que avanzó el curso aprendieron por medio de la práctica que cuando estos procesos no se realizaban con el rigor necesario demoraban más en la elaboración de la solución o entregaban resultados que no eran los esperados. Finalmente esta práctica constante les permitió desarrollar estrategias de análisis que les brindaron seguridad para enfrentarse a situaciones problemáticas influyendo favorablemente en la elaboración de sus soluciones (Gannotti, 1987).

Resultado de la intervención los estudiantes reconocieron la utilidad del aprendizaje de la programación y su aplicación como herramienta para la resolución de problemas. En esta experiencia observé, que no sólo los conocimientos y habilidades adquiridas con la programación son importantes para el desarrollo de habilidades de resolución de problemas (Pea & Kurland, 1984 en Tu & Johnson, 1990; Dalbey & Linn, 1985 en Tu & Johnson, 1990) sino también en sentido contrario, siendo áreas de conocimiento que se complementan. Y de acuerdo con Perrone (1998), Boix-Mansilla y Gardner (1998) y Ordóñez (2003), cuando los estudiantes

aplicaban la programación a la solución de problemas, encontraron sentido al estudio de esta disciplina, logrando construir aprendizajes de manera significativa.

Un aspecto que considero necesario cuidar en experiencias orientadas a la resolución de problemas es el grado de dificultad de los mismos. En dos ocasiones, en mi intervención, los problemas resultaron tan complejos que el proceso de construcción del programa no se logró. Me parece importante graduar la complejidad de los problemas para lograr un equilibrio entre los procesos de resolución del problema incluida la elaboración del programa en el caso de la programación.

Otros aprendizajes interesantes que los estudiantes lograron con esta intervención son el aprender a trabajar en equipo, aprender a escuchar y a coordinar con varias personas el logro de un objetivo común, coincidiendo con los resultados de estudios presentados en el marco teórico (Jacovljevic, 2003; LeJeune, 2003; Sabin y Sabin, 1994). Sin embargo el trabajo en grupo no necesariamente garantiza aprendizajes, en esta intervención fue necesario guiarlos permanentemente para que se comunicaran de manera adecuada, insistirles en la importancia del aporte de todos los integrantes del grupo, así como en la necesidad de que todos escribieran de manera individual su solución, así fuera el resultado de la discusión y el trabajo de todos. No puedo asegurar que fruto de esta experiencia los estudiantes ya saben trabajar en grupo de manera efectiva, pero fue un primer logro hacia esta forma de aprender. Sugiero muchas más experiencias de aprendizaje en ambientes de colaboración porque los resultados no se ven sólo en el aprendizaje de las áreas de estudio sino también en aprendizajes sociales.

Aunque la motivación no fue el propósito de esta intervención, a lo largo del curso los estudiantes se mostraron motivados con la elaboración de sus programas y por esta misma razón con la adquisición de los nuevos aprendizajes que les requería cada problema. Teniendo en cuenta la dificultad que manifiestan los estudiantes en el aprendizaje de la disciplina, durante la intervención se acercaron a los conocimientos básicos de la programación por medio de su

aplicación en programas sencillos que les permitieron apreciar los resultados de su esfuerzo y aprendizaje. De acuerdo con la propuesta de Taylor, et al. (1986), Giannotti (1987) y Guibert, et al. (2004), es importante lograr interés y motivación en los estudiantes como aspecto básico para promover aprendizajes, pero al mismo tiempo presentarles retos para construir conocimientos más elaborados.

Igualmente, para todos los estudiantes las prácticas que realizamos juntos en el salón de computadores resultaron muy activas y valiosas para su aprendizaje. La dificultad que representa el aprendizaje de un lenguaje para la elaboración de los programas va disminuyendo a medida que la práctica en el computador aumenta. Sin embargo la presencia del computador en el aula de clase representó para algunos estudiantes un elemento que desvió por momentos los objetivos de aprendizaje propuestos. Por eso me parece importante seguir estudiando los efectos de las prácticas en las salas de computadores enmarcadas en un ambiente constructivista de aprendizaje y considerando consistentemente al computador y lenguajes de programación como herramientas para apoyar el aprendizaje y no como únicos objetos de estudio, aún en cursos de programación de computadores (Villalobos, et al., 2005).

Por último, resultado de esta innovación, como profesora igualmente lo gré aprendizajes. Aprendí a escuchar a mis estudiantes, a observar cómo procede su aprendizaje y a graduar mi intervención de acuerdo con esto. Supe de qué hablan en sus clases y cómo se enfrentan a lo que desconocen. Ahora reconozco sus capacidades de aprendizaje y entiendo que los estudiantes no aprenden más por lo que yo les diga en clase, ellos aprenden más en la medida en que la estrategia pedagógica que elabore para ellos les presente retos y nuevas oportunidades para aprender.

## REFERENCIAS

- Berk, L. & Winsler, A. (1995). *Scaffolding children's learning*. Washington, E.U.: NAEYC.
- Boix-Mansilla, V. & Gardner, H. (1998). What Are the Qualities of Understanding? En M.S. Wiske (Ed.) *Teaching for Understanding: Linking Research with Practice* (pp. 161-183). San Francisco: Jossey-Bass.
- Bruffe, K. (1999). *Collaborative learning, higher education, interdependence, and the authority of knowledge*. Baltimore, MD: Johns Hopkins University Press.
- Gibert, N., Girard, P. & Guittet, L. (2004). Example-based Programming: a pertinent visual approach for learning to program. Artículo recuperado de la base de datos ACM en Abril de 2007.
- Giannotti, E. (1987) Algorithm Animator: A Tool for Programming Learning. Artículo recuperado de la base de datos ACM en Octubre de 2006.
- Ginsburg-Block, M. y Fantuzzo, J. (1998). An Evaluation of the Relative Effectiveness of the NCTM Standard-Based Interventions. *Journal of Education Psychology*, 90 (3), 560 – 569. Artículo recuperado de la base de datos APA el 11 de Mayo de 2004.
- Jakovljevic, M. (2003). Concept Mapping and Appropriate Instructional Strategies in Promoting Programming Skills. Artículo recuperado de la base de datos ACM el 28 de Mayo de 2004.
- LeJeune, N. (2003). Critical Components for Successful Collaborative Learning in CS. *JCSC*, 19 (1), 275 - 285. Artículo recuperado de la base de datos ACM el 8 de Mayo de 2004.
- Nagappan, N. & Williams, L. & Ferzli, M. & Wiebe, B. & Yang, K. & Miller, C. & Balik, S. (2003). Improving the CS1 Experience with Pair Programming. *SIGCSE*. Artículo recuperado de la base de datos ACM el 28 de Mayo de 2004.
- Nickerson, R., Perkins, D. y Smith, E. (1990). Enseñar a pensar. Aspectos de la aptitud intelectual. (Romano, L. y Ginard, C., Trads.) Barcelona, España: Ediciones Paidós Ibérica S. A. (Trabajo original publicado en 1987).
- Ordóñez, C. (2003). Currículo para los Colegios de la Asociación Alianza Educativa. Artículo recibido del autor en Octubre de 2004.
- Perkins, D. (1998). What is Understanding? En M.S. Wiske (Ed.) *Teaching for Understanding: Linking Research with Practice* (pp. 39-57). San Francisco: Jossey-Bass.
- Perrone, V. (1998). Why Do We Need a Pedagogy of Understanding? En M.S. Wiske (Ed.) *Teaching for Understanding: Linking Research with Practice* (pp. 13-38). San Francisco: Jossey-Bass.

- Piaget, J. (1970). Piaget's Theory. En P. Mussen (Ed) Carmichael's Manual of Child Psychology (Vol.1). New York: Wiley.
- Polya, G. (1957). How to Solve It. A New Aspect of Mathematical Method. (2a. Ed) Princeton, New Jersey, EE.UU.: Princeton University Press.
- Pressman, R. (1998). Ingeniería del Software. Un enfoque práctico. (Ojeda, R., Sánchez, J., Yagüe, V., Zurdo, J., García, J. y Morales, F., Trads.) Madrid, España: McGraw Hill. (Trabajo original publicado en 1998).
- Sabin, R. y Sabin, E. (1994). Collaborative Learning in an Introductory Computer Science Course. Artículo recuperado de la base de datos ACM el 8 de Mayo de 2004.
- Savery, J. y Duffy, T. (1996). Problem based learning: an instructional model and its constructivist framework. En B.G. Wilson (Ed) Constructivist learning environments: case studies in instructional design. New Jersey: Educational technology publications, Englewood Cliffs.
- Taylor, R., Cunniff, N. & Uchiyama, M. (1986). Learning, Research, and the Graphical Representation of Programming. Artículo recuperado de la base de datos ACM el 31 de Marzo de 2004.
- Tu, J. & Johnson, J. (1990) Can Computer Programming Improve Problem Solving Ability? SIGCSE, 22 (2), 30-37. Artículo recuperado de la base de datos ACM el 3 de Marzo de 2004.
- Van De Grift, T. (2004). Coupling Pair Programming and Writing: Learning about Student's Perceptions and Processes. SIGCSE. Artículo recuperado de la base de datos ACM el 11 de Mayo de 2004.
- Villalobos, J. (2007). Tutorial: Cómo enseñar a programar: un enfoque efectivo. Recuperado en Mayo de 2007 del sitio Web de la Universidad de los Andes: <http://cupi2.uniandes.edu.co/proyecto/presentaciones/Cupi2-2007-04-18-CCC07.pdf>.
- Villalobos, J., Casallas, R. y Marcos, K. (2005) El Reto de Diseñar un Primer Curso de Programación de Computadores. Recuperado en Febrero de 2007 del sitio Web de la Universidad de los Andes: <http://cupi2.uniandes.edu.co/docs/Villalobos-Casallas-Marcos-CIESC05.pdf>.
- Vygotsky, L. (1978). Interaction between Learning and Development. En: Mind in Society: The development of higher psychological processes. Cambridge, MA: Harvard University Press.
- Yerion, K. y Rinehart, J. (1995). Guidelines for Collaborative Learning in Computer Science. SIGCSE Bulletin, 27 (4), 29 -35. Artículo recuperado de la base de datos ACM el 11 de Mayo de 2004.

## Anexo 1

### Prueba académica 1

En la inscripción de participantes a Eciciencia 2006 se preguntará a cada participante el número de su documento de identidad y otros datos personales. Elabore un algoritmo que capture el número del documento de identidad, tenga en cuenta que a los menores de edad se les pregunta número de la Tarjeta de Identidad y a los mayores de edad número de Cédula. El algoritmo debe hacer esta diferenciación en el mensaje que captura el dato. También a los hombres mayores de edad se les preguntará el número de la Libreta Militar. A todos sin importar sexo ni edad se les preguntará el número del carné estudiantil si es estudiante y si es profesor se le preguntará número de Cédula, se supone que todos los profesores son mayores de edad, pero igual si son hombres se preguntará el número de su Libreta Militar.

Elabore el algoritmo primero para preguntar los datos a un sólo participante y luego modifique el algoritmo de manera que sirva para preguntar a todos los participantes. ¿Cómo sabrá cuántos participantes son?

#### **Guía para solucionar el problema**

¿Cuál es el problema?

¿Qué es lo que necesitamos hacer?

¿Qué conoce del problema?

¿Qué no conoce del problema y usted piensa que es importante conocer para poder solucionarlo?

Identifique todos los datos del problema, asígneles un nombre y decida de qué tipo son.

Identifique qué datos debe preguntar al usuario.

Intente hacer un dibujo o macro algoritmo con no más de cinco pasos muy generales tratando de encontrar la solución.

¿Qué conoce sobre Programación de Computadores que le sirva para resolver el problema?

Revise el programa del curso y lo que usted ha aprendido hasta ahora.

Elabore el algoritmo.



## Anexo 2

### Prueba académica 2

La directora de la biblioteca de la universidad quiere saber qué tanto consultan los libros los estudiantes de la universidad y le parece que puede empezar por conocer el número de préstamos realizados en un día normal de funcionamiento. Los libros están clasificados en cuatro grandes áreas y ella cree que los libros que más se consultan son los de ingeniería electrónica. ¿Su impresión será cierta? Las áreas definidas por la biblioteca son:

- E - Ing Electrónica
- B – Ciencias básicas
- C – Ing Civil
- A – Administración y Economía

La directora acude a usted para que con sus conocimientos sobre programación le ayude a generar la información que necesita.

#### **Guía para solucionar el problema**

¿Cuál es el problema? Defina el problema. ¿Qué es lo que no se están pidiendo?

¿Qué conoce del problema?

Identifique todos los datos del problema, ¿en qué tipo de variables los va a almacenar?, ¿tiene constantes?

¿Necesita condicionales no repetitivos?

¿Necesita manejar ciclos?

¿Qué datos va a preguntar? ¿Qué datos va a calcular?

## Anexo 3

### Prueba académica 3

**Actividad:** trabaje individualmente para dar una solución al problema planteado.

#### Guía para solucionar el problema

Defina el problema ¿Cuál es el problema?

¿Qué conoce del problema?

¿Cuál es el dato o los datos que solucionan el problema?

Identifique los datos del problema, ¿dónde los va a almacenar?

¿Qué conoce sobre Programación de Computadores que le sirva para resolver el problema: condicionales, ciclos, arreglos, funciones?

¿Qué datos va a preguntar?

#### Problema

El instituto de Medio Ambiente Distrital está adelantando un estudio sobre el clima de nuestra ciudad. Para ello recogió cada una de las temperaturas registradas (en grados centígrados) en el termómetro del instituto cada hora durante los siete días de la semana anterior. Ahora quieren saber con base en esa información cuáles fueron las temperaturas promedio de cada hora del día durante esa semana.

El siguiente es un ejemplo del registro de estas temperaturas:

Día \ Hora	0:00	1:00	2:00	3:00	4:00	5:00	6:00	...	23:00
Lunes	-1,0	-1,0	2,0	2,3	1,9	3,0	5,0	...	2,8
Martes	1,4	2,4	3,2	4,8	2,2	3,7	4,4	..	3,2
Miércoles	1,6	2,5	3,3	4,6	3,8	3,5	6,4	...	3,5
Jueves	1,8	2,9	3,6	4,2	3,4	7,3	7,2	...	2,6
Viernes	2,5	3,6	3,7	4,9	4,0	6,4	8,1	...	4,1
Sábado	2,7	3,6	4,1	3,1	6,2	6,3	5,9	...	4,3
Domingo	3,4	4,0	4,7	3,2	4,5	8,2	6,3	...	3,8

Elabore un programa para ayudarle al instituto a obtener la información requerida.

## Anexo 4

### Matriz de evaluación

Criterios de evaluación		Rangos		
1	<b>Identifican el problema</b>	5.0 - 4.0 El algoritmo refleja que identificaron el problema.	3.9 - 3.0 En el algoritmo se observan algunos aspectos relacionados con el problema planteado.	2.9 - 1.0 El algoritmo contiene procesos que no se plantean en el problema.
2	<b>Elaboran un plan de solución</b>	Usan alguna herramienta que les ayude a representar su plan de solución (dibujo, macro algoritmo, etc.)		No realizan un plan de solución previo a la elaboración del algoritmo.
3	<b>Datos</b>	Identifican datos. Identifican constantes y variables. Reconocen la diferencia entre nombre y contenido. Declaran datos variables con los tipos correctos. Asignan nombres claros. Identifican posibles valores. Comprenden que una instrucción de asignación modifica el valor de una variable. Hacen correcto uso de variables auxiliares como contadores, acumuladores y banderas.	Identifican y declaran algunos datos. Algunos datos no son declarados del tipo correcto. No identifican constantes. Reconocen la diferencia entre nombre y contenido. Asignan nombres claros. Comprenden los resultados de las instrucciones de asignación. No inicializan variables.	No identifican o no declaran los datos. Confunden nombre de la variable con su contenido. Los nombres de variables no tienen relación con su contenido. No tienen claro cómo modificar el contenido de una variable.
4	<b>Entradas</b>	Identifican datos de entrada. Envían mensajes claros al usuario. Validan datos. El algoritmo lee los datos necesarios para solucionar el problema o situación planteada.	Identifican algunos datos de entrada. Envían algunos mensajes claros al usuario. Validan algunos datos. El algoritmo lee algunos de los datos necesarios para solucionar el problema o situación planteada.	No identifican datos de entrada. Los mensajes al usuario no son claros o no envían. No validan datos. El algoritmo no lee los datos necesarios para solucionar el problema.
5	<b>Salidas</b>	Identifican datos de salida. Los mensajes son claros para el usuario. El programa entrega los resultados esperados.	Identifican algunos datos de salida. Los mensajes no son claros para el usuario. El programa entrega algunos de los resultados esperados.	No hay mensajes para el usuario. El programa no entrega los resultados esperados.
6	<b>Estructuras condicionales no repetitivas</b>	Plantean la condición de manera correcta. Identifican los procesos que dependen de la condición. Estructuras completas. Identifican qué forma de estructura condicional usar en la situación correspondiente. Identifican cuándo usar estructura repetitiva y cuándo no.	Plantean la condición pero tiene errores. Identifican los procesos que dependen de la condición. Estructuras completas. Algunas estructuras condicionales usadas no son las correctas. Repiten instrucciones.	Las estructuras están incompletas o condicionan procesos que no deberían estarlo.
7	<b>Estructuras condicionales repetitivas</b>	Identifican los procesos que se repiten. Plantean la condición de manera correcta. Identifican cuándo para el ciclo. Estructuras completas. Identifican cuándo usar estructura repetitiva y cuándo no.	Estructuras incompletas. Se repiten instrucciones que no deben repetirse.	Estructuras incompletas. Ciclos infinitos.

8	<b>Datos estructurados - Arreglos y Estructuras</b>	Identifican y declaran arreglos. Hacen correcto uso de los arreglos. Identifican y definen estructuras de info. Declaran datos tipo estructuras. Hacen correcto uso de las estructuras.	Identifican y declaran arreglos pero hacen usos incorrectos de ellos. Identifican y declaran estructuras pero hacen usos incorrectos de ellas.	No usan arreglos donde los podrían usar o los declaran pero su uso es siempre incorrecto. No usan estructuras donde las podrían usar o las incorporan pero su uso es siempre incorrecto.
9	<b>Uso de funciones y parámetros</b>	Definen funciones y procedimientos El diseño de las funciones y los procedimientos es el adecuado El uso de las funciones y procedimientos es correcto El uso que hacen de argumentos y parámetros es correcto Usan apuntadores (parámetros por referencia) Demuestran que comprenden el concepto de apuntadores: los usan bien.	Tienen algunos errores en la definición o llamado de las funciones. Algunos parámetros no coinciden con los definidos.	No hacen uso de funciones o las usan pero no están definidas o no coinciden con el prototipo. Mal uso de los parámetros. Invocan mal los subprogramas.
10	<b>Pruebas de escritorio</b>	Hacen pruebas para verificar su solución		No las hacen

## Anexo 5

### Prueba Inicial y Final

**Actividad:** trabaje en grupo para dar una solución a la situación planteada. Todos los integrantes del grupo deben escribir mostrando el proceso seguido y la solución final propuesta. Recuerden trabajar en hojas sueltas para poder entregar el trabajo realizado.

#### Situación planteada

La profesora P necesita entregar un informe a Registro Académico al final de este primer tercio. Este informe debe mostrar la nota definitiva del tercio de cada uno de los 19 estudiantes del curso de Programación de Computadores. También deberá mostrar la nota promedio del curso en este primer tercio. Dos estudiantes que no asistieron a la prueba corta, presentaron excusa justificada. Por tal razón para estos estudiantes el examen vale por el porcentaje del examen más el porcentaje de la prueba corta.

#### Una guía para solucionar problemas

¿Cuál es el problema o la necesidad del usuario?

¿Qué es lo que necesitamos hacer?

¿Qué conoce del problema?

¿Qué no conoce del problema y usted piensa que es importante conocer para poder solucionarlo?

Discuta este punto con sus compañeros y lleguen a un acuerdo.

Identifique todos los datos del problema, asígneles un nombre y decida de qué tipo son.

Identifique qué datos debe preguntar al usuario.

Intente hacer un dibujo o macro algoritmo con no más de cinco pasos muy generales tratando de encontrar la solución. Revíselo con sus compañeros y lleguen a un acuerdo.

¿Qué conoce sobre Programación de Computadores que le sirva para resolver el problema?

Revise el programa del curso y lo que usted ha aprendido hasta ahora.

Elabore el algoritmo con sus compañeros.

## Anexo 6

### Categorización inicial de grabaciones de clase para buscar aprendizajes relacionados con la solución de problemas

- Expresa que identifica un dato o enunciado del problema porque lo nombra.

Ejemplos: (Grupo 1)

E3: *Ah! Noo... Pero toca sacarle el promedio.* [**Identifica un dato de la solución, el promedio**]

-----

E1: *Y aquí ¿cuál es la prueba corta?*

E2: *La que tiene porcentaje del 20%.* [**Identifica uno de los datos del problema**]

- Pide información a sus compañeros sobre los datos y el problema.

Ejemplo: (Grupo 1)

E1: *Yo ahí tengo una duda. Aquí no dijeron dos estudiantes, ¿se supone que solo no asistieron esas dos personas o pueden [sic] haber más?* [**Pide información sobre el problema**]

E2: *Pueden [sic] haber más....*

E1: *O sea que... sólo dos presentaron excusa.*

- Pide información a sus compañeros sobre programación.

Ejemplo: (Grupo 1)

E1: *Entonces “para”, ¿ustedes lo van a hacer con “mientras” o con “para”?* [**Pide información sobre el tipo de estructura a utilizar en el programa**]

E2: *Con “para”.*

E1: *“Para i”....*

- Pide la validación de lo que expresa o propone a sus compañeros.

Ejemplo: (Grupo 2)

E3: *Pero ¿qué medidas? Puede ser ésta o ésta, pero nos sirve es hasta acá.*

E2: *Y para este lado y este lado y ya, nosotros sacamos el resto.*

E3: *Pues sí, ¿pero este pedazo como lo sacamos? Toca que diga las medidas de la oficina, ¿cierto? Con dos datos se puede sacar el resto, con uno de acá y uno de arriba.* [**Propone y pide validación de lo que dice**]

E2: *No, porque decimos, diga las medidas de la cocina, diga las medidas del baño de hombres, diga las medidas del baño de mujeres y hacerlo por figuras.*

- Da una explicación sobre un tema del curso o corrige lo que dicen sus compañeros.

Ejemplo: (Grupo 3)

E3: *Pero eso queda muy largo, ¿no? O sea, son tantos, 1, 2, 3, 4, 5 vectores.*

E2: *Pero es que eso no quiere decir que tú vas a hacer 19 operaciones. Para eso son los ciclos. Ahorita (sic) yo les explico. Lo que pasa es que cuando ustedes van en un ciclo, así como pueden aumentar una variable, pueden aumentar la posición del vector, entonces tu no*

*vas a tener que andar escribiendo esto cada dos minutos, sino en el “para” o en el “mientras que” vas a poner los datos por el contador, o sea acá adentro no va a haber un número sino digamos un contador. Luego en el “para” tu lo que vas a hacer es contador + 1, entonces ya no va a ser 1 sino 2, la misma operación con valores. [Explica el uso de los vectores a sus compañeros]*

- Expresa que aplica conocimientos de programación en la solución del problema.

Ejemplo: (Grupo 3)

E1: *¿No deberíamos escribir, digite la nota del segundo tercio del código del estudiante respectivo?*

E2: *Por eso, mira, como el contador empieza en 1, dé la nota del segundo tercio del estudiante 1. [Expresa que comprende el uso del contador y lo aplica]*

- Expresa que está aclarando una comprensión de programación o del problema.

Ejemplo: (Grupo 1)

E2: *Si usted tiene cuatro notas, ¿cómo saca su nota definitiva?*

E1: *Las multiplica por cada porcentaje y las suma.*

E2: *Las suma, ¿cierto?*

E1: *Las suma pero nada más, pero no se dividen.*

E2: *Sí, sí, las suma pero no se dividen tiene razón, sí. [Aclara su comprensión]*

- Propone escribir la solución de alguna manera específica.

Ejemplo: (Grupo 2)

E2: *¿Cómo sería otra forma de medir áreas?*

E1: *Porque sería decir: ‘Ahora mídame las longitudes de la oficina... ¿Tiene rectángulos? Sí... ¿Cuántos tiene? Tres... Deme las longitudes de esos tres rectángulos’... Se acaba ese ciclo y entonces, ‘¿Tiene círculos?...No’... Pum (sic), se acaba ese ciclo... ‘¿Tiene trapecios? Sí. ¿Cuántos? Dos... Dícteme las longitudes’, le dicta las longitudes, le saca el área y eso lo va sumando en otro lado y así con las cuatro figuras. [Propone una manera de escribir la solución]*

- Expresa inconformidad con la propuesta de algún compañero y propone otra.

Ejemplo: (Grupo 3)

E2: *Acá ya sabemos que son dos estudiantes...*

E3: *Pero toca preguntar si usted la presentó o no la presentó. Yo digo que para no estar digitando todos los datos pongamos estudiante1, estudiante2, estudiante3 y así... los estudiantes son una lista.*

E2: *¿Pero y los nombres? Preguntemos los nombres.*

E3: *No, porque eso es lo que yo te digo, vamos a tener un número, para no escribir Carlos, Mateo, Juan, sino sólo estudiante1. O sea yo voy a decir los estudiantes por la lista que tiene el profesor, para no tener que estar diciendo tu nombre. Cada quien tiene un número en la lista, entonces cada quien debe saberse ese número como el código. [Expresa inconformidad con la idea de E2 y propone la suya]*

- Expresa inconformidad con la propuesta de algún compañero, explicándola.

Ejemplo: (Grupo 2)

E2: *O sea, lo que yo hago es hacer procesos por figuras, o sea, hallar áreas por figuras. Nos queda más fácil que meternos con sectores. ¿Si me entiende? Clasificarlos por figuras... que si la figura es un cuadrado, empiece a hallar áreas, si la figura es un trapecio, empiece a hallar áreas...* [**Explica su propia propuesta**]

E1: *O sea, que el dijera:... “hálleme el área del cuadrado”, o “necesita un rectángulo”, entonces “hálleme todas las áreas de los rectángulos”... ¿me dice usted? O sea, que empiece a hallar esta área y después ésta... y después, ¿cómo las clasificamos? Ahí es el problema.* [**Explica la propuesta de E2 y expresa su inconformidad**]

E2: *Por sectores... ¿me dice por sectores?*

E1: *Usted me dice que diga cuántos rectángulos hay, entonces él dice 6, empiece a darme las longitudes, le da la del 1°, le da la del 2°, 3°, 4°... así. Después cómo hace usted para saber si el 1° pertenecía a la cocina o a la oficina... ese es el problema.* [**Expresa inconformidad con la propuesta de E2 explicándola**]

- Expresa que realiza un proceso de verificación de la solución.

Ejemplo: (Grupo 1)

E1: *Y ahora escribir... ah listo. Sale de aquí... entonces escribir... promedio, falta una para promedio... y a promedio asígnale sumatoria F sobre 19 y... hagamos la prueba de escritorio entonces: nota1, nota2, nota3, nota4, basura, basura, basura. Asignar cuatro notas: 3.0, 4.5, 5.0 y 2.0. Si nota3 es igual a cero, nada, sino entonces hace esto... Oiga se me hace que hace falta algo para que... Entonces dice si nota3 no es igual a cero, se sale de todas estas condiciones y pasa acá, con esa sumatoria le asigna todo esto, que es igual a... le asigna 3.0 por 0.15 más 4.5 por 0.15 más 5.0 por 0.20 más 2.0 por 0.50, 3.1 da. Listo, tiene esta nota, entonces a la sumatoria final le asigna cero, que la inicializamos con cero más esto y termina en “fin para”. Entonces sigue 1, sigue 2, entonces vuelve con las notas y asigna...* [**La prueba de escritorio es un proceso de verificación que se hace a los algoritmos y programas**]



## Anexo 7

### Protocolo de encuesta inicial

Apreciado estudiante:

Esta encuesta tiene como propósito conocer su pensamiento sobre el desarrollo de esta asignatura, con el propósito de mejorarla, reforzando lo positivo y corrigiendo las fallas que se evidencien. Le solicito el favor de responder con sinceridad, objetividad y compromiso.

Gracias

1. ¿Qué piensa usted que se aprende en este curso?

---

---

2. ¿Sabe que es la programación de computadores?    Sí    No

3. Si su respuesta anterior fue afirmativa, explique brevemente qué es la programación de computadores.

---

---

4. ¿Ha trabajado previamente con computadores?    Sí    No

5. ¿Para qué ha utilizado el computador?

Tareas y trabajos académicos \_\_\_\_

Para el aprendizaje de otras materias o de un programa en particular \_\_\_\_

Para diversión y entretenimiento \_\_\_\_

Tareas y trabajos personales \_\_\_\_

Otros \_\_\_\_\_

6. ¿Sabe programar o ha recibido clases de programación?    Sí    No

7. ¿Si sabe programar, dónde aprendió a programar?

---

8. ¿Si sabe programar, qué lenguaje maneja?

---

9. ¿Recuerda algún programa que haya hecho? ¿Cuál?

---

10. ¿Si sabe programar, considera que programar es una tarea fácil o difícil? Explique brevemente.

---

11. ¿Para qué piensa usted que sirve aprender a programar?

---

12. ¿Si sabe programar, le gusta programar?    Sí    No



## Anexo 8

### Protocolo de encuesta final

Apreciado estudiante:

Esta encuesta tiene como propósito conocer su pensamiento sobre el desarrollo de esta asignatura, con el propósito de mejorarla, reforzando lo positivo y corrigiendo las fallas que se evidencien. Le solicito el favor de responder con sinceridad, objetividad y compromiso.

Gracias

1. ¿Qué piensa usted que aprendió a HACER en este curso?

\_\_\_\_\_

\_\_\_\_\_

2. ¿Usted considera que aprendió a programar?

Si \_\_\_\_\_ No \_\_\_\_\_ Mucho \_\_\_\_\_ Un poco \_\_\_\_\_

Explique brevemente su anterior respuesta.

\_\_\_\_\_

\_\_\_\_\_

3. Mencione tres temas del curso que usted considera que aprendió a HACER.

3.1. \_\_\_\_\_

3.2. \_\_\_\_\_

3.3. \_\_\_\_\_

4. Además de programación, ¿usted piensa que en este curso aprendió alguna otra cosa?

Sí \_\_\_\_\_ No \_\_\_\_\_

5. Si su respuesta anterior fue afirmativa, ¿qué es lo que usted piensa que aprendió, además de programación?

\_\_\_\_\_

\_\_\_\_\_

6. ¿Recuerda algún programa que haya hecho? Si \_\_\_\_\_ No \_\_\_\_\_

¿Cuál? \_\_\_\_\_

7. ¿Piensa que programar es una tarea fácil o difícil? Explique brevemente.

\_\_\_\_\_

\_\_\_\_\_

8. ¿Para qué piensa usted que le servirá saber programar?

\_\_\_\_\_

\_\_\_\_\_

9. ¿Le gusta programar? Sí \_\_\_\_\_ No \_\_\_\_\_

¿Por qué? \_\_\_\_\_

10. ¿Le gusta estudiar sólo o en compañía de otros? Explique brevemente

- \_\_\_\_\_
- \_\_\_\_\_
11. ¿Conoce estrategias o metodologías para solucionar problemas? Sí \_\_\_ No \_\_\_
  12. ¿Aplica alguna estrategia o metodología para solucionar los problemas que le plantean en sus materias de estudio o en las demás clases? Sí \_\_\_ No \_\_\_
  13. ¿Si su anterior respuesta fue afirmativa, qué estrategias o metodologías aplica?  
\_\_\_\_\_  
\_\_\_\_\_
  14. ¿En sus clases le gusta trabajar en grupo? Sí \_\_\_ No \_\_\_  
¿Por qué? \_\_\_\_\_  
\_\_\_\_\_
  15. ¿Usted piensa que el trabajar con sus compañeros en clase le ayudó a aprender más? Sí \_\_\_ No \_\_\_  
Explique brevemente su respuesta \_\_\_\_\_  
\_\_\_\_\_
  16. ¿Cuál o cuáles de los siguientes recursos le ayudó a aprender más? Señale sólo los dos que más le ayudaron.  
  

Libros _____	Problemas _____
Tablero _____	Amigo _____
Apuntes _____	Profesor _____
Compañero de curso _____	Otro _____ ¿Cuál? _____
Clase salón de computadores _____	
  17. ¿Su estrategia o metodología de resolución de problemas cambió después de este curso?  
Si \_\_\_ No \_\_\_
  18. ¿Los problemas sobre los que se trabajó durante el curso le ayudaron a comprender mejor los diferentes temas? Si \_\_\_ No \_\_\_
  19. Mencione algo que haya aprendido de sus compañeros de trabajo \_\_\_\_\_  
\_\_\_\_\_
  20. ¿Qué sugerencias puede aportar para lograr mayores aprendizajes en este curso?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Anexo 9

### Protocolo de entrevista inicial a grupos focales

Fecha: \_\_\_\_\_ Hora inicio: \_\_\_\_\_ Hora fin: \_\_\_\_\_

Número de participantes: \_\_\_\_\_ Número hombres: \_\_\_\_\_ Número mujeres: \_\_\_\_\_

Preguntas:

1. ¿Qué es programación de computadores?
2. ¿Cómo se sienten hasta ahora en nuestra clase, han aprendido algo? ¿Qué han aprendido?
3. ¿Qué hacen para solucionar un problema? ¿Tienen alguna estrategia o metodología de solución de problemas?
4. ¿Creen que pueden mejorar sus estrategias de solución de problemas?, ¿cómo?
5. ¿Qué piensan que podemos mejorar en nuestra clase para aprender más?
6. ¿Cómo se sienten estudiando y trabajando con sus compañeros de clase?
7. ¿Sienten que aprenden de esta manera? ¿Por qué?
8. ¿Encuentran dificultades con esta manera de aprender?
9. ¿Qué opinan sobre los problemas que resolvemos en clase, les podemos mejorar algo, el tema, las preguntas, etc.?

## Anexo 10

### Protocolo de entrevista final a grupos focales

Fecha: \_\_\_\_\_ Hora inicio: \_\_\_\_\_ Hora fin: \_\_\_\_\_

Número de participantes: \_\_\_\_\_ Número hombres: \_\_\_\_\_ Número mujeres: \_\_\_\_\_

Preguntas:

1. ¿Qué aprendieron en este curso?
2. ¿Lo que aprendieron en este curso, coincide con lo que esperaban aprender?
3. ¿Creen que lo aprendido en este curso les sirve para algo útil? ¿Para qué?
4. ¿Cómo les pareció la metodología empleada en este curso? (facilitó o dificultó el aprendizaje, retroalimentación oportuna, participación activa de los estudiantes, agradable, aburrida, evaluaciones)
5. ¿Creen que la programación les ayuda a solucionar problemas?
6. ¿Qué hacen para solucionar un problema?
7. Después del curso, ¿cambió su percepción o idea de lo que es la programación?
8. ¿Les gusta trabajar en grupo? ¿Por qué?
9. ¿Cómo fue su participación en la clase? (activa, motivada, comprometida, dedicada, puntual)
10. ¿Ahora piensan que ya saben programar? ¿Por qué?
11. ¿Programar es fácil o difícil?
12. ¿El trabajo en grupo que hicimos durante el semestre cambió en algo su idea de lo que consideraban ustedes como trabajo en grupo?