

Javier Alejandro Jiménez Toledo



**Propuesta metodológica basada en aprendizaje colaborativo y
modelado gráfico para la enseñanza de los fundamentos de
programación**

Director: PhD(C) María Lili Villegas
Co-Director: PhD. Cesar Alberto Collazos

Universidad Del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Maestría en Computación

Popayán
2016

Javier Alejandro Jiménez Toledo

**Propuesta metodológica basada en aprendizaje colaborativo y
modelado gráfico para la enseñanza de los fundamentos de
programación**

Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Magíster en Computación

Director: PhD (C) María Lili Villegas
Co-Director: PhD. Cesar Alberto Collazos

Popayán
2016

Dedicatoria

A mis amados tesoros, mis hijos Davis y Saray, como una mínima recompensa por el tiempo dejado de compartir durante estos años de estudio.

A mi gran amor, mi hermosa esposa Ximena, quién me apoyó incondicionalmente y cuidó de mis angelitos durante mi ausencia.

A mis padres, Javier y Carmencita, por brindarme la oportunidad de pertenecer a tan lindo hogar y por todo el cariño y apoyo recibido sea cual fuere la situación.

A mis hermanos, Helen y Robinson, y mis sobrinos Santiago y Alejandro, por su apoyo y cariño.

Agradecimientos

Al ser todo poderoso que me ha permitido estar haciendo lo que más me gusta en compañía de los que más adoro.

Con mucho cariño, a mi querida Institución Universitaria CESMAG, quién me apoyó como becario de esta fabulosa experiencia.

Con cariño a la Vicerrectora de Investigaciones de la Institución Universitaria CESMAG, la PhD. (C) María Eugenia Córdoba, por todo su respaldo incondicional brindado en todo este proceso.

Con gratitud a mi estimada directora, la PhD. (C) María Lili Villegas Ramírez, por su tiempo, amabilidad y disposición a lo largo de esta inolvidable experiencia.

Con aprecio a mi estimado co director, el PhD. César Collazos, por su enorme compromiso que sin importar la hora o el día estuvo siempre dispuesto a colaborándome y por sus valiosas enseñanzas en el campo investigativo que solo se viven y se aprenden cerca de su brillante actividad.

Con aprecio a mi estimado profesor, el PhD. Julio Hurtado, en quien encontré un amigo en tierras desconocidas y quién me apoyó incondicionalmente en esta travesía.

Muchas gracias a todos los profesores de la maestría, quienes compartieron además de sus conocimientos, muchas enseñanzas de vida.

Y finalmente a los compañeros de la maestría, con quienes volvimos a sentir la angustia de ser nuevamente estudiantes y con quienes compartimos buenos momentos y experiencias de vida.

Contenido

INTRODUCCIÓN	13
CAPÍTULO 1	
PROYECTO DE INVESTIGACIÓN	15
1.1 Descripción del problema	15
1.2 Pregunta de investigación	17
1.3 Hipótesis de trabajo	17
1.4 Objetivos	17
1.4.1 Objetivo general	17
1.4.2 Objetivos específicos	17
1.5 Metodología	18
CAPÍTULO 2	
MARCO CONCEPTUAL	19
2.1 Aprendizaje colaborativo	19
2.1.1 Objetivos del aprendizaje colaborativo	20
2.1.2 Beneficios del aprendizaje colaborativo	21
2.1.3 Elementos del aprendizaje colaborativo	21
2.1.4 Técnicas de trabajo colaborativo	22
2.1.5 JIGSAW	22
2.2 Pensamiento computacional	23
2.2.1 Dimensiones del pensamiento computacional	23
2.2.2 Etapas del pensamiento computacional	24
2.2.3 Características del pensamiento computacional	24
2.2.4 El pensamiento computacional como habilidad de la era digital	25
2.3 Metodologías colaborativas para la enseñanza aprendizaje de programación	26
CAPÍTULO 3	
METODOLOGÍA BASADA EN APRENDIZAJE COLABORATIVO Y MODELADO GRÁFICO	31
3.1 Propuesta metodológica	31
3.1.1 Configuración inicial	31
3.1.2 Etapa: Fundamentación teórica con modelado gráfico	32
3.1.2.1 Configuración inicial para modelado gráfico	32
3.1.2.2 Planeador de ejemplos	33
3.1.2.3 Recolección de requerimientos	33
3.1.2.4 Fase de análisis	34
3.1.2.5 Fase de diseño	34
3.1.3 Etapa: proceso colaborativo	34

3.1.3.1 Configuración inicial caso de estudio	34
3.1.3.2 Planeador de ejercicio	35
3.2 Descripción del Modelo Colaborativo	35
3.2.1 Rol Docente	36
3.2.2 Configuración inicial	38
3.2.2.1 Configuración inicial para caso de estudio	39
3.2.2.2 Configuración inicial para fundamentación conceptual de modelado gráfico	40
3.2.3 Rol estudiante	40
3.2.4 Caso de estudio	42
3.3 Propuesta de modelado gráfico	46
3.3.1 Elementos de modelado gráfico	47
3.3.2 Fases para propuesta de modelado gráfico	47
3.3.2.1 Fase de recolección de requerimientos	47
3.3.2.2 Fase de análisis	48
3.3.2.3 Fase de diseño	51
3.4 Unidades de competencia con propuesta de modelado gráfico	52
3.4.1 Captura y salida de datos	52
3.4.2 Condicionales	56
3.4.2.1 Condicional compuesto	56
3.4.2.2 Condicional simple	58
3.4.2.3 Condicional anidado	60
3.4.2.4 Condicionales y operadores lógicos	61
3.4.3 Ciclos	63
3.4.3.1 Ciclo Para	63
3.4.3.2 Ciclo Mientras	68
3.4.3.3. Ciclo Hacer Mientras	69

CAPÍTULO 4

VALIDACIÓN PROPUESTA METODOLÓGICA 71

4.1 Sesión de clase	71
4. 2 Resultados de la investigación a expertos	74
4.2 Resultado de investigación con estudiantes	79
4.2.1 Caracterización de población	79
4.2.2 Resultados de postprueba grupo de control	81
4.2.3 Resultado de postprueba grupo experimental	83
4.2.4 Resultados de grupo experimental vs de control	85

CAPÍTULO 5

CONCLUSIONES, TRABAJOS FUTUROS Y DIVULGACIÓN 93

5.1 Conclusiones	93
5.2 Trabajos futuros	94
5.3 Divulgación	95

REFERENCIAS BIBLIOGRÁFICAS

96

ANEXOS

101

Lista de tablas

Tabla 2.1	Metodologías para la enseñanza aprendizaje de los fundamentos de programación de computadores	30
Tabla 3.1	Resumen metodología	31
Tabla 3.2	Formato A	32
Tabla 3.3	Formato B	32
Tabla 3.4	Formato C	33
Tabla 3.5	Formato D	34
Tabla 3.6	Formato E, F, G	34
Tabla 3.7	Formato H	35
Tabla 3.8	Formato I	35
Tabla 3.10	Roles y sub roles por cada unidad de aprendizaje	42
Tabla 3.11	Reglas de actividad por unidad y elemento de competencia	44
Tabla 3.12	Descripción de los sub roles para el rol de estudiante	46
Tabla 3.13	Formato de recolección de requerimiento principal	48
Tabla 3.14	Ejemplos Entradas/salidas para situaciones vivenciales	49
Tabla 3.15	Elementos de GUI	50
Tabla 3.16	Equivalencias entre GUI y DF	52
Tabla 3.17	Recolección de requerimientos para cuadrado de un número.	52
Tabla 3.18	Recolección de requerimientos Concepto cualitativo	57
Tabla 3.19	Recolección de requerimientos descuento de artículo.	58
Tabla 3.20	Recolección de requerimientos positivo, negativo o neutro.	60
Tabla 3.21	Recolección de requerimientos rango de números	62
Tabla 3.22	Recolección de requerimientos números ascendente ciclo Para	64
Tabla 3.23	Análisis ciclo para ascendente	65
Tabla 3.24	Recolección de requerimientos números descendente ciclo Para.	66
Tabla 3.25	Análisis ciclo Para descendente	67
Tabla 3.26	Recolección de requerimientos números ascendente ciclo Mientras.	68
Tabla 3.27	Recolección de requerimientos números ascendente ciclo Hacer Mientras.	69
Tabla 4.1	Acciones para una sesión de clase	72
Tabla 4.2	Expertos por universidad	74
Tabla 4.3	Evaluación por unidad de competencia por expertos	77
Tabla 4.4	Diseño experimental por universidad	79
Tabla 4.5	Caracterización de los grupos participantes	80
Tabla 4.6	Caracterización estudiante de Introducción a la Programación	80
Tabla 4.7	Caracterización estudiantes de Lógica Computacional	81
Tabla 4.8	Caracterización estudiantes de Fundamentos de Programación	81

Tabla 4.9	Caracterización de grupos experimentales y de control	81
Tabla 4.10	Promedio del grupo de control para Introducción a la Programación	82
Tabla 4.11	Promedio del grupo de control para Lógica Computacional	82
Tabla 4.12	Promedio del grupo de control para Fundamentos de Programación	82
Tabla 4.13	Promedio de notas definitivas grupos de control	83
Tabla 4.14	Promedio del grupo experimental Introducción a la Programación	84
Tabla 4.15	Promedio del grupo experimental Lógica Computacional	84
Tabla 4.16	Promedio del grupo experimental Fundamentos de Programación	84
Tabla 4.17	Promedio de notas definitivas por cada curso	85
Tabla 4.18	Grupo de control vs grupo experimental de Fundamentos de Programación	85
Tabla 4.19	Grupo de control vs grupo experimental de Lógica Computacional	86
Tabla 4.20	Grupo de control vs grupo experimental de Fundamentos de Programación	87
Tabla 4.21	T de Student para Entrada y salida de datos de Introducción a la Programación	88
Tabla 4.22	T de Student para Condicionales de Introducción a la Programación	88
Tabla 4.23	T de Student para Ciclos de Introducción a la Programación	89
Tabla 4.24	T de Student para Entrada y salida de datos de Lógica Computacional	89
Tabla 4.25	T de Student para Condicionales de Lógica Computacional	89
Tabla 4.26	T de Student para Ciclos de Lógica Computacional	90
Tabla 4.27	T de Student para Entrada y salida de datos de Fundamentos de Programación	90
Tabla 4.28	T de Student para Condicionales de Fundamentos de Programación	91
Tabla 4.29	T de Student para Ciclos de Fundamentos de Programación	91
Tabla 5.1	Artículos en revistas	95
Tabla 5.2	Ponencias en Congresos Internacionales	95

Lista de figuras

Figura 2.1	Mapa mental del marco conceptual	10
Figura 3.1	Modelo colaborativo	36
Figura 3.2	Rol docente	37
Figura 3.3	Rol estudiante	41
Figura 3.4	Diagrama de actividad general para caso de estudio	42
Figura 3.5	Diagrama de actividades detallado para caso de estudio	42
Figura 3.6	Diagrama de actividad general sub rol analista	45
Figura 3.7	Diagrama de actividad general sub rol diseñador	45
Figura 3.8	Diagrama de actividad sub rol Analista y Diseñador	45
Figura 3.9	Diagrama de actividad retroalimentación parcial	46
Figura 3.10	Diagrama de actividad para retroalimentación caso simultáneo	46
Figura 3.11	Diagrama de Entrada/Salida	49
Figura 3.12	D E/S cajero automático	49
Figura 3.13	D E/S Calculadora básica	50
Figura 3.14	D E/S cuadrado de un número	53
Figura 3.15	GUI inicial	53
Figura 3.16	D E/S Vs GUI	53
Figura 3.17	GUI cálculo de cuadrado	54
Figura 3.18	Orden de presentación de los elementos de una GUI a un usuario	55
Figura 3.19	DF cuadrado de un número	55
Figura 3.20	DF sugerido de cuadrado de un número	56
Figura 3.21	D E/S Concepto cualitativo	57
Figura 3.22	GUI concepto cualitativo de nota	57
Figura 3.23	Relación entre D D/S y DF para condicional compuesto	57
Figura 3.24	DF concepto cálculo cualitativo nota	58
Figura 3.25	D E/S descuento para artículo	59
Figura 3.26	GUI descuento para artículo	59
Figura 3.27	DF descuento para artículo	59
Figura 3.28	D E/S Concepto cualitativo de nota	60
Figura 3.29	GUI positivo, negativo o neutro	60
Figura 3.30	Relación entre D D/S y DF para condicional anidado	61
Figura 3.31	DF positivo, negativo o neutro	61
Figura 3.32	D E/S rango de números	62
Figura 3.33	GUI rango de números	62
Figura 3.34	DF Rango de números	63
Figura 3.35	D E/S Números ascendentes ciclo Para	64
Figura 3.36	GUI Números ascendentes ciclo Para	64
Figura 3.37	DF número ascendentes ciclo Para	65
Figura 3.38	D E/S Números descendentes ciclo Para	66
Figura 3.39	GUI Números descendentes ciclo Para	66
Figura 3.40	DF número descendentes ciclo Para	67

Figura 3.41	DS Números ascendentes ciclo Mientras	68
Figura 3.42	GUI Números ascendentes ciclo Mientras	68
Figura 3.43	DF número ascendentes ciclo Mientras	69
Figura 3.44	D E/S Números ascendentes ciclo Hacer Mientras	70
Figura 3.45	GUI Números ascendentes ciclo Hacer Mientras	70
Figura 3.46	DF número ascendentes ciclo Hacer Mientras	70
Figura 4.1	Diagrama de actividad clase “Captura y salida de datos”	71
Figura 4.2	Diagrama de detalle de actividad rol docente	73
Figura 4.3	Diagrama de detalle de actividad rol estudiante	73
Figura 4.4	Pregunta No. 1 a expertos	74
Figura 4.5	Pregunta No. 2 a expertos	75
Figura 4.6	Pregunta No. 3 a expertos	75
Figura 4.7	Pregunta No. 4 a expertos	75
Figura 4.8	Pregunta No. 5 a expertos	76
Figura 4.9	Pregunta No. 6 a expertos	76
Figura 4.10	Pregunta No. 7 a expertos	76
Figura 4.11	Pregunta No. 8 a expertos	77
Figura 4.12	Pregunta No. 9 a expertos	77
Figura 4.13	Pregunta No. 10 a expertos	78
Figura 4.14	Pregunta No. 11 a expertos	78
Figura 4.15	Pregunta No. 12 a expertos	78
Figura 4.16	Diseño experimental	79
Figura 4.17	Resultado para talleres de Introducción a la Programación	85
Figura 4.18	Resultado para exámenes de Introducción a la Programación	86
Figura 4.19	Resultado para talleres de Lógica Computacional	86
Figura 4.20	Resultado para exámenes de Lógica Computacional	87
Figura 4.21	Resultado para talleres de Fundamentos de Programación	87
Figura 4.22	Resultado para exámenes de Fundamentos de Programación	88

Lista de anexos

Anexo 1	Formato A: Configuración inicial	102
Anexo 2	Formato B: Configuración inicial para fundamentación conceptual de modelado gráfico	103
Anexo 3	Formato C: Planeación de ejemplos de la configuración inicial para fundamentación conceptual de modelado gráfico	104
Anexo 4	Formato D: Recolección de requerimiento principal	105
Anexo 5	Formato E: Diagrama Entrada/Salida (D E/S)	106
Anexo 6	Formato F: Interfaz Gráfica de Usuario (GUI)	107
Anexo 7	Formato G: Diagrama de Flujo (DF)	108
Anexo 8	Formato H: Configuración inicial para el caso de estudio	109
Anexo 9	Formato I: Planeación de ejercicios de configuración inicial para el caso de estudio	110
Anexo 10	Formato H: configuración inicial para el caso de estudio: Fundamentos de Programación	111
Anexo 11	Formato I: planeación de ejercicios de configuración inicial para el caso de estudio: Fundamentos de Programación	114
Anexo 12	Formato B: configuración inicial para fundamentación conceptual de modelado gráfico: Fundamentos de Programación	120
Anexo 13	Formato C: planeación de ejemplos de la configuración inicial para fundamentación conceptual de modelado gráfico: Fundamentos de Programación	121
Anexo 14	Ejercicios complementarios	
Anexo 15	Encuesta a expertos	147
Anexo 16	Formato de caracterización de estudiantes	150
Anexo 17	Notas Introducción a la Programación grupo control	151
Anexo 18	Notas Lógica Computacional grupo control	152
Anexo 19	Notas Fundamentos de Programación grupo control	153
Anexo 20	Notas Introducción a la Programación grupo experimental	154
Anexo 21	Notas Lógica Computacional grupo experimental	155
Anexo 22	Notas Fundamentos de Programación grupo experimental	156
Anexo 23	Revista Investigium Ire	157
Anexo 24	Revista I+T+C	161
Anexo 25	Certificado CICOM	165
Anexo 26	Certificado CACIED	166

INTRODUCCIÓN

Este documento presenta los resultados del proceso investigativo realizado para la construcción de una metodología que mediante un entorno colaborativo combinado con un propuesta de modelado gráfico, pretende la enseñanza de los fundamentos de programación de una forma dinámica donde el estudiante es el actor principal del proceso de aprendizaje. Para ello se construyeron dos roles: docente y estudiante, en primera instancia es el docente el encargado de la configuración inicial del proceso que mediante las directrices de un proceso colaborativo realiza su participación guiado por las características de un mediador cognitivo, instructor y diseñador instruccional con el propósito de presentarle al estudiante los elementos de las unidades de competencia a profundizar, mediante una propuesta basada en modelado gráfico donde todo el proceso de pensamiento algorítmico obedecerá al reconocimiento de cuatro definidas fases inherentes en la construcción de proyectos de software que incluyen las fases de: requerimientos, análisis, diseño y codificación. En una segunda instancia, será el estudiante el actor principal del proceso de aprendizaje.

El propósito de enmarcar la propuesta de modelado gráfico en las cuatro fases obedece únicamente a inculcarle al estudiante que el desarrollo de soluciones de software es un proceso planeado que lleva consigo una serie de fases que harán del producto final un entregable robusto que surge de la aplicación de un enfoque sistémico basado en la Ingeniería de Software.

Además, la propuesta de modelado gráfico basa su accionar en el elemento más básico de la fase de análisis que es el Diagrama de Entrada Salida el cual será identificado por las siglas “D E/S” mediante el cual es posible llegar a la solución de muchas situaciones algorítmicas solo con sus componentes básicos.

Es así como la presente metodología en una primera etapa de trabajo se apoya del rol docente para exponer los elementos de competencia a los estudiantes utilizando modelado gráfico a través de una estrategia didáctica basada en la clase magistral que se realizará únicamente a nivel de introducción en la sesión de clase. Una vez el docente realice su labor introductoria, es donde comienza el verdadero proceso de aprendizaje para el estudiante enmarcado en un entorno colaborativo apoyado con la técnica JIGSAW mediante la cual se planea el trabajo del estudiante dividido en sub roles, que dependiendo del elemento de competencia a estudiar, proporcionará a este una dinámica interactiva que terminará en la solución y conclusión colectiva del grupo de estudio frente al desarrollo de una actividad puntual, la cual será finalmente retroalimentada por el docente.

Finalmente la estrategia propuesta fue llevada a cabo en tres cursos de fundamentos de programación de tres diferentes universidades de la ciudad de San Juan de Pasto Colombia, los cuales fueron considerados los grupos experimentales de la investigación. Los resultados obtenidos en estos tres cursos fueron

contrastados y comparada con tres grupos diferentes de estudiantes de las mismas universidades quienes se llamaron grupos de control. Los resultados obtenidos fueron analizados por técnicas estadísticas para analizar su impacto.

La presentación del proceso investigativo llevado a cabo se organiza en el documento de la siguiente forma: El capítulo uno consta de los elementos que dieron origen al proyecto de investigación como lo es el aporte investigativo y los objetivos planteados. En el capítulo dos se incluye el marco conceptual que soporta a la investigación. En el capítulo tres se presenta toda la propuesta metodológica para la enseñanza de los fundamentos de programación desde un ambiente colaborativo y basada en una propuesta de modelado gráfico. El capítulo cuatro exhibe los resultados de la aplicación de la metodología a seis grupos de estudiantes y su correspondiente análisis estadístico. En el capítulo cinco se presentan las conclusiones y trabajos futuros derivados del presente estudio y finalmente se soportan los referentes de la investigación.

CAPÍTULO 1: PROYECTO DE INVESTIGACIÓN

1.1 Descripción del problema

La enseñanza de los fundamentos de programación es un campo estudio que ha tomado gran importancia debido al auge del software en la actualidad en todos los procesos del accionar humano, hasta tal punto de estar presente en dispositivos con *wearable thechnology* que es una sofisticada tecnología que se puede llevar puesta como otra prenda de vestir [1]. Pese al estado tecnológico actual existen aún problemas relacionados con la fundamentación de los futuros constructores de software cuyo origen se inicia desde el primer curso de programación recibido, el cual es clave en el proceso de formación que tendrá un desarrollador de soluciones software en su etapa de aprendizaje y más tarde en su vida profesional [2]; tales inconvenientes se presenta por múltiples situaciones que van desde la carencia de habilidades para programar, el desconocimiento de la materia y hasta la insuficiencia de disciplina en programación [3].

En la actualidad existen algunas prácticas para la enseñanza de los fundamentos de programación, es así como en los primeros cursos algunos docentes inician esta enseñanza, independiente del paradigma seleccionado (procedural, orientado a objetos, funcional o lógico), con contextos relacionados al campo de la matemática y la geometría ya que los ejemplos matemáticos básicos y especialmente los de geometría plana elemental son posibles de modelar debido a que los estudiantes han adquirido suficientes conocimientos básicos previos en esas disciplinas [4]. En cambio, otros docentes llevan al estudiante a plantear soluciones a problemas a través de un proceso cíclico de “prueba y error” hasta obtener una solución que satisfaga las normas de evaluación propuestas [5]. Además, dentro de un paradigma determinado se visualizan varios enfoques para enseñar: algunos enseñan a programar en un lenguaje de programación particular, utilizando su sintaxis y su semántica, y otros emplean un lenguaje algorítmico lo bastante general cómo para permitir su traducción posterior a cualquier lenguaje de programación [6].

Una de las temáticas más complejas en la enseñanza de los fundamentos de programación es la relacionada con las estructuras iterativas debido a su complejidad estructural y más aún por la falta de asociación vivencial desde el punto de vista de un estudiante. Aprender a programar es reconocido por ser difícil, una de las mayores dificultades se basa en el hecho de que los lenguajes de programación son artificiales [7]. Truong, Bancroft y Roe [8] establecen que la educación en computación se enfrenta a tres grandes problemas: en primer lugar, la mayoría de los novatos estudiantes de programación tienen dificultad al enfrentarse a su primer programa o en la construcción de conocimientos referentes a programación. En segundo lugar, la gran cantidad de estudiantes matriculados a los cursos de introducción a la programación provoca dificultad en propiciar una ambiente de aprendizaje efectivo. Por último, hay una creciente necesidad de apoyar los campus desde múltiples sitios que redundan en el manejo de grandes cantidades de información técnica al estudiante.

En el estudio realizado por Affleck y Smith [9] se ha encontrado que el principal problema de los programadores principiantes es el acceso a los conocimientos previos y la adopción de un enfoque para estudiar, que va más allá de la memorización explícita de conocimientos necesarios para aplicar y transferir el dominio de conceptos a situaciones nuevas. Es así como los métodos de representación de lógica formal como Pseudocódigo, Diagramas de Flujo,

diagramas Nassi-Shneiderman, en otros casos el manejo de paradigmas como el Orientado a Objetos, Lógico o Funcional que se imparten en los primeros cursos de programación, no vivencian explícitamente los resultados de operación y ejecución de una manera clara en su fases iniciales [10].

Para Manuel Gonzales y Maximiliano Paredes [11] son muchos los inconvenientes que debe enfrentar un estudiante de primer curso de programación debido a la complejidad de las temáticas tratadas, por ello, el inicio en el aprendizaje de la programación siempre ha sido un proceso complicado ya que es una disciplina totalmente diferente a lo que se ha visto desde entonces por parte del programador novato porque la programación exige cambiar de manera radical el modo de pensar y analizar las cosas y aun habiendo adquirido los conocimientos teóricos necesarios se ha detectado una gran dificultad de aplicar esos conocimientos teóricos en la resolución de problemas prácticos. La enseñanza de la programación no se puede transmitir directamente desde instructores a los alumnos, debe ser adquirida activamente por los estudiantes [12].

Existen diversas teorías sobre los modelos de enseñanza, que van desde los tradicionales, los constructivistas, hasta los investigativos [13] [14] [15] [16] [17] [18] [19] [20] entre otros como se citó en Romero y Rosero [21]; sin embargo, es importante aclarar que en las prácticas de enseñanza universitaria no se determina un modelo único o ideal, que copiase de manera fotográfica sus postulados, por cuanto no se puede desconocer la realidad que acontece en el aula, el contexto, la interacción entre docente y estudiantes [21].

Para Moroni y Señas [22] una estrategia de aprendizaje de programación se fundamenta en comenzar a enseñar programación utilizando los algoritmos para dar solución a un problema. Esta propuesta lleva al estudiante a realizar sus soluciones en modelos que se construyen en papel y a probarlos mediante pruebas de escritorio. A pesar de aparentemente no tener dificultades con el aprendizaje por parte del estudiante, se ha comprobado que se requiere de mucho tiempo y esfuerzo la obtención de algoritmos semánticamente correctos.

Otras metodologías para la enseñanza de los fundamentos de programación basan su esfuerzo en que los estudiantes desde el primer curso se enfrenten a un lenguaje de programación, es así como Xinogalos [23] determina que esta metodología presenta problemas importantes de lógica computacional en el estudiante, debido a su preocupación por no tener errores sintácticos en el código que los soluciona sin tener en cuenta una verdadera concepción lógica.

Por otro lado, otros autores inician la fundamentación de la lógica computacional en la solución de problemas en el campo de la matemática y la geometría donde se construyen ejemplos y ejercicios matemáticos básicos y de geometría plana debido a los conocimientos previos de los estudiantes. Uno de los problemas de esta metodología se presenta en la falta de amplitud conceptual que en muchas ocasiones conlleva a la pérdida de motivación por parte del estudiante y que por sí mismas no alcanzarían para cubrir los requisitos de un curso introductorio de programación [4].

Villalobos hace la siguiente reflexión en el modelo de enseñanza de los fundamentos de programación:

el modelo tradicional de enseñanza de los fundamentos de programación se basa en que el profesor, después de presentar en clase los conceptos (o pedir que los estudiantes lean de un libro), responde las preguntas que puedan surgir de dichas definiciones, y luego comienza a hacer ejercicios en el tablero, dicha manera de enseñar a programar se basa en la esperanza de que el estudiante sea capaz de detectar patrones en los problemas planteados y los logre asociar con las técnicas que usa el profesor cuando desarrolla un ejemplo en el tablero, y que, luego, el estudiante sea capaz de hacer la generalización adecuada en su cabeza, para poder aplicar esa asociación patrón-técnica para resolver otros problemas. La única técnica que se enseña en los cursos de programación es “dividir y conquistar” y se hace a un nivel de informalidad y vaguedad que no pasa de ser una buena intención. Esta metodología presenta problemas como el que no se genera el vocabulario necesario para hablar de la manera de construir un programa, da la sensación de que el “cómo hacer las cosas” no es enseñable, y que es algo que depende de la inspiración y del ingenio del programador y quizá la más preocupante radica en que la calidad del curso es muy dependiente de la capacidad del profesor para inducir a los estudiantes a aprender por imitación [24].

De continuar con esta situación y como lo demuestra la experiencia en el campo de la enseñanza de los fundamentos de programación sólo un porcentaje mínimo de los estudiantes lograrán el desarrollo de las competencias básicas necesarias en el fortalecimiento de procesos lógicos que lleven al estudiante en etapas posteriores al desarrollo de habilidades cada vez más complejas para el desarrollo de proyectos de construcción de software de acuerdo a los nuevos paradigmas de programación que deberán enfrentar.

1.2 Pregunta de investigación

¿Cómo contribuir con la enseñanza de los fundamentos de programación?

1.3 Hipótesis de trabajo

HI: La propuesta metodológica mediante aprendizaje colaborativo y modelado gráfico contribuye positivamente en la enseñanza de los fundamentos de programación.

HO: La propuesta metodológica mediante aprendizaje colaborativo y modelado gráfico No contribuye positivamente en la enseñanza de los fundamentos de programación.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar una propuesta metodológica para la enseñanza de los fundamentos de programación mediante aprendizaje colaborativo y modelado gráfico.

1.4.2 Objetivos específicos

- ✓ Caracterizar la población y los procesos llevados a cabo para la enseñanza de los fundamentos de programación en instituciones de educación superior a nivel de pregrado.

- ✓ Construir una propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico que permita el estudio de conceptos fundamentales necesarios en la enseñanza de los fundamentos de programación.
- ✓ Validar la propuesta metodológica con estudiantes de fundamentos de programación utilizando un grupo de control y un grupo experimental y así comprobar las hipótesis planteadas.

1.5 Metodología

La metodología general llevada a cabo en el presente estudio tuvo en cuenta los siguientes ítems:

Para caracterizar los procesos llevados a cabo en la enseñanza de los fundamentos de programación fue necesario recopilar información a nivel regional, nacional e internacional de los procesos académicos llevados a cabo en diversos cursos con el fin de establecer los elementos de competencia necesarios para abordar el saber conocer. De igual forma para la caracterización de la población se tomó como muestra de la investigación tres grupos experimentales y tres de control a quienes se les aplicó un cuestionario en el que se determinaron los atributos que identificaron a dicha población; dicho cuestionario fue tabulado y sus datos analizados.

Una vez caracterizados procesos y muestra poblacional se establecieron las estrategias didácticas acordes tanto al grupo de estudiantes como al saber conocer que permitieron diseñar las actividades de aprendizaje y evaluación adecuadas en ambientes multiusuarios y colaborativos para el buen desarrollo del curso propuesto. Con el saber conocer y las estrategias didácticas de las actividades de aprendizaje y evaluación determinadas se construyó una propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para el estudio de los fundamentos de programación. Con la propuesta metodológica construida se inició la fase de validación para la cual se tomó el siguiente diseño experimental:

G ₁	X	O ₁
G ₂	-	O ₂
G ₃	X	O ₃
G ₄	-	O ₄
G ₅	X	O ₅
G ₆	-	O ₆

La investigación fue realizada en seis grupos diferentes pertenecientes a tres instituciones de educación superior del municipio de Pasto (Nariño) para el primer curso de fundamentación de programación. Los grupos G₁, G₃ y G₅ corresponden a los grupos experimentales de cada institución y G₂, G₄ y G₆ fueron sus grupos de control respectivamente, además X fue el tratamiento experimental que consistió en la estrategia didáctica bajo un entorno colaborativo a través de modelado gráfico. A su vez, O₁, O₂, O₃, O₄, O₅, O₆, fueron las post pruebas realizadas al final del tratamiento experimental tanto para los grupos experimentales como los de control. Este diseño experimental fue aplicado a cada temática de estudio determinado en las siguientes unidades de aprendizaje: Captura y salidas de datos, condicionales simples y compuestos, condicionales anidados, condicionales y operadores lógicos, ciclo para, ciclo mientras, ciclo hacer mientras.

CAPÍTULO 2: MARCO CONCEPTUAL

En este capítulo se encuentran los fundamentos teóricos que soportan la investigación realizada que abordan las teorías del aprendizaje colaborativo, el pensamiento computacional y la esencia de los modelos pedagógicos y su incidencia en la ingeniería. En la en la figura 1 se presenta el mapa mental del marco conceptual.

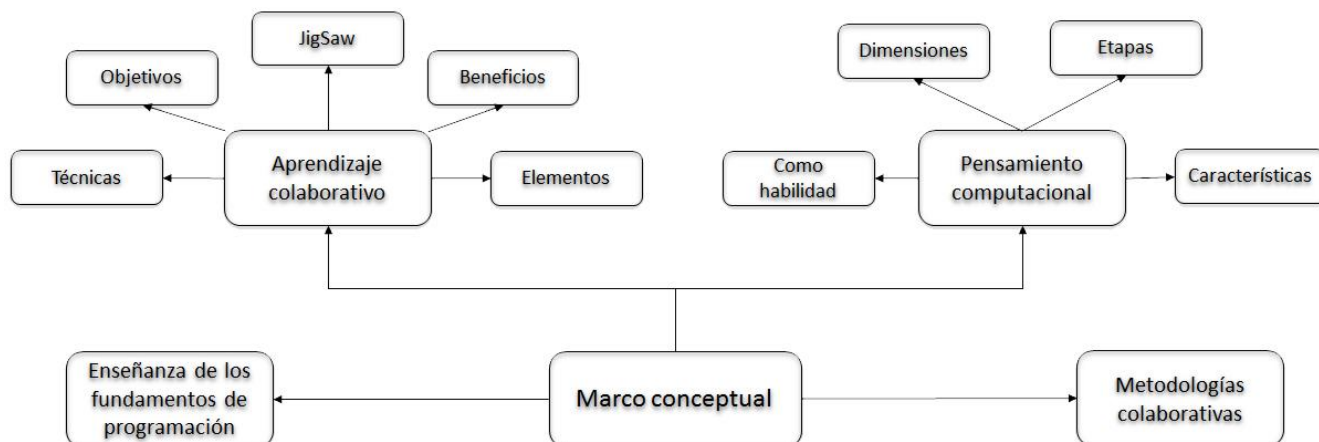


Figura 2.1. Mapa mental del marco conceptual

2.1 Aprendizaje Colaborativo

La colaboración en el aprendizaje no es un tema nuevo y es con la remodelación de la pedagogía en el siglo XX y el desarrollo de las TIC que se regeneraron la reflexión y la práctica acerca el aprendizaje colaborativo [25]. El aprendizaje colaborativo está inmerso en la teoría de constructivismo social [26], y se centra en el proceso de construcción del conocimiento a través del aprendizaje que resulta de la interacción con un grupo y mediante tareas realizadas en cooperación con otros [27].

Las primeras ideas de colaboración en el aprendizaje se vieron reflejadas con las aportaciones de Dewey [28]. En la misma época, el soviético Vygotsky postula la importancia del aprendizaje en un ambiente propicio para el alumno, y considera al individuo como el resultado del proceso histórico y social donde el conocimiento es un proceso de interacción entre el sujeto y el medio, entendido como algo social y cultural, no solamente físico [29].

El aprendizaje colaborativo para Johnson, Johnson y Holubec es el uso instruccional de pequeños grupos de tal forma que los estudiantes trabajen juntos para maximizar su propio aprendizaje y el de los demás [30]. Al mismo tiempo, Brufee considera al aprendizaje colaborativo como la composición de elementos de consenso a través de la colaboración en el aprendizaje, la participación voluntaria en el proceso, el aprendizaje no fundacional, el cambio en la relación profesor-estudiante, la importancia del trabajo y diálogo entre pares donde se discute la autoridad del profesor y la validez de los contenidos gracias al método [31].

Por su parte, Muhlenbrock considera que el aprendizaje colaborativo es el compromiso mutuo establecido entre personas en un grupo, que se unen en un esfuerzo coordinado para dar respuesta a una tarea [32]. A su vez, Dillenbourg determina que la definición más amplia pero

“insatisfactoria” del término aprendizaje colaborativo es la situación en la cual una o más personas aprenden e intentan aprender algo en forma conjunta, además añade que esto incluye todo tipo de grupos ya sean pareja, comunidades, entre otros [33].

Al igual, Salinas considera que el aprendizaje colaborativo es la adquisición de destrezas y actitudes que ocurren como resultado de la interacción en grupo [34], donde en el enfoque colaborativo de organización del trabajo se considera esencial la interacción alumno-profesor pero también la interacción alumno-alumno, los alumnos aprendiendo unos de otros, así como de su profesor y del entorno [35].

Leidner y Jarvenpaa señalan que el aprendizaje colaborativo, además de ayudar a desarrollar el pensamiento crítico en los estudiantes, también contribuye a mejorar las relaciones interpersonales [36], al igual que Barab, Thomas y Merrill se refieren este aprendizaje como la co-construcción de significado que resulta de compartir experiencias personales [37] donde cada miembro del grupo de trabajo es responsable no solo de su aprendizaje, sino de ayudar a sus compañeros a aprender, creando con ello una atmósfera de logro [38].

En este aprendizaje los estudiantes trabajan colaborando, este tipo de aprendizaje no se opone al trabajo individual ya que puede observarse como una estrategia de aprendizaje complementaria que fortalece el desarrollo global del alumno, además, se establece que los métodos de aprendizaje colaborativos traen consigo una renovación en los roles asociados a profesores y alumnos, en el caso de los profesores se establece tres tipos: Profesores como Mediador cognitivo, Instructor y Diseñador Instruccional [39].

El aprendizaje colaborativo, tal y como ahora se conoce, empieza a surgir a principios de los años 1970 [25]. En esa época destacaron los trabajos de Johnson y Johnson [40] en los Estados Unidos, quienes empezaron a desarrollar el aprender juntos, al igual que en España Serrano y Calvo [41] trabajaron sobre la dicotomía cooperación/competición.

2.1.1 Objetivos del aprendizaje colaborativo

Hsu determina que el objetivo del aprendizaje colaborativo es inducir a los participantes a la construcción de conocimiento mediante exploración, discusión, negociación y debate [42].

Por su parte, Brown y Atkins [43] establecen que los objetivos que persigue el aprendizaje colaborativo se centran en el desarrollo de estrategias de comprensión y explicación, de preguntas y respuestas. La discusión y el debate sirven, en primer lugar, para desarrollar las habilidades de comunicación con otros y la utilización precisa del lenguaje. En segundo lugar, genera el desarrollo de competencias intelectuales y profesionales, como por ejemplo: analizar, razonar, pensar críticamente, sintetizar, diseñar, etc. Y, por último, el aprendizaje colaborativo también promueve el crecimiento personal de los estudiantes, que incluye el desarrollo de estrategias de comunicación y pensamiento, el desarrollo de la autoestima, dirige el propio aprendizaje, aprende a trabajar con otros y a conocerse a sí mismo y a los demás [25].

En este modelo de colaboración, los profesores “invitan” a sus estudiantes a definir los objetivos específicos dentro de la temática que se está enseñando, brindando opciones para

actividades y tareas que logren atraer la atención de los alumnos, animando a los estudiantes a evaluar lo que han aprendido [39].

2.1.2 Beneficios del aprendizaje colaborativo

Los beneficios del aprendizaje colaborativo están ampliamente documentados en la literatura del ámbito. Roberts [44] agrupa los principales efectos que el aprendizaje colaborativo puede aportar a los estudiantes de la siguiente manera:

Académicos, puesto que enfatiza en la participación y construcción activa de conocimiento, promoviendo habilidades de alto orden de pensamiento e incrementando los resultados de la clase.

Sociales, teniendo en cuenta que fomenta la creación de una atmósfera positiva para el aprendizaje a la vez que permite desarrollar un sistema de apoyo social para los estudiantes, fomentando la comprensión y enseñanza entre ellos.

Psicológicos, ya que desarrolla actitudes positivas hacia los profesores y puede incrementar la autoestima de los estudiantes.

2.1.3 Elementos del aprendizaje colaborativo

Términos tales como: pasivo, memorización, individual y competitivo, son elementos que no están asociados con AC [45]. Por el contrario, los elementos que siempre están presentes en este tipo de aprendizaje son:

Cooperación. Los estudiantes se apoyan mutuamente para cumplir con un doble objetivo: lograr ser expertos en el conocimiento del contenido, además de desarrollar habilidades de trabajo en equipo. Los estudiantes comparten metas, recursos, logros y entendimiento del rol de cada uno. Un estudiante no puede tener éxito a menos que todos en el equipo tengan éxito.

Responsabilidad. Los estudiantes son responsables de manera individual de la parte de tarea que les corresponde. Al mismo tiempo, todos en el equipo deben comprender todas las tareas que les corresponden a los compañeros.

Comunicación. Los miembros del equipo intercambian información importante y materiales, se ayudan mutuamente de forma eficiente y efectiva, ofrecen retroalimentación para mejorar su desempeño en el futuro y analizan las conclusiones y reflexiones de cada uno para lograr pensamientos y resultados de mayor calidad.

Trabajo en equipo. Los estudiantes aprenden a resolver juntos los problemas, desarrollando las habilidades de liderazgo, comunicación, confianza, toma de decisiones y solución de conflictos.

Autoevaluación. Los equipos deben evaluar cuáles acciones han sido útiles y cuáles no. Los miembros de los equipos establecen las metas, evalúan periódicamente sus actividades e identifican los cambios que deben realizarse para mejorar su trabajo en el futuro.

2.1.4 Técnicas de aprendizaje colaborativo

Cabero y Pérez [46] establecen que en las técnicas centradas en el trabajo en grupo a partir de estructuras comunicativas de colaboración, las metas son compartidas por el grupo y requieren de la participación activa de todos los miembros en una fase colaborativa para continuar en el intercambio de construcción de conocimiento en donde el profesor interviene aportando normas, estructura de la actividad y realizando el seguimiento y valoración de las actividades. Algunas de las principales técnicas que favorecen el trabajo colaborativo son: Trabajo en parejas, lluvia de ideas, rueda de ideas, votaciones, valoración de decisiones, debate y foro, pequeños grupos de discusión, controversia estructurada, grupos de investigación, simulaciones y juegos de rol, estudio de casos y trabajo por proyectos.

De igual manera, Collazos y Mendoza [47] mencionan algunas de las técnicas que se pueden utilizar en el proceso colaborativo, entre ellas están [47]:

Cuestionamiento por pares, que consiste en darles a los estudiantes partes de preguntas para que ellos las completen y se las hagan después a sus compañeros de clase [48] [47].

Compartir pensamiento en pares, los estudiantes primero trabajan en su problema individualmente, luego comparan sus respuestas con un compañero de clases y, por último, sintetizan una solución en conjunto [49] [47].

TAPPS (resolución de problemas en pares pensando en voz alta), los estudiantes trabajan los problemas en pares, con uno de ellos ejerciendo el papel de quien resuelve el problema y el otro el del que escucha la solución. El primero expresa verbalmente todo lo que piensa respecto a la solución del problema y luego los roles se intercambian [50] [47].

JIGSAW, se forman los equipos, en los cuales cada persona se responsabiliza de un aspecto de la tarea en cuestión. Se obtiene así una interdependencia positiva, debido a que cada estudiante tiene diferente información necesaria para completar la tarea [51] [47].

Cabezas numeradas juntas (*Numbered Heads Together*), mediante esta técnica todos los participantes necesitan saber y estar listos para explicar las respuestas del grupo, donde cada estudiante de cada grupo se enumera (1 a 4), luego el profesor o un estudiante hace una pregunta acerca del tema en estudio. Los grupos se juntan para dar una respuesta y finalmente el profesor selecciona un número entre 1 y 4 y la persona que lo tenga explica la respuesta [49] [47].

2.1.5 JIGSAW

Es una técnica específica de aprendizaje cooperativo [51], funciona como un rompecabezas, en el que cada una de las actividades de los estudiantes son piezas de un puzle necesarias para la comprensión y completitud del resultado final [52]. El JIGSAW se realiza en varias fases: en la primera, los alumnos se dividen en grupos reducidos, denominados grupo de expertos, a los cuales se le asigna el estudio de una parte del tema a tratar, de modo que cada uno lee, discute y profundiza en su parte; a continuación, los alumnos son redistribuidos formando nuevos grupos compuestos por, al menos, un experto de cada tema, de forma que cada uno explica y discute su parte con sus compañeros; finalmente, se realiza una actividad de evaluación, como por ejemplo responder a un cuestionario [52].

Aronson manifiesta que a pesar de que la técnica tiene una serie de ventajas [51] no está exenta de algunas limitaciones entre las que se destacan: existencia de un estudiante dominante que domina el grupo, existencia de alumnos más rezagados y dificultad de adaptación al aprendizaje cooperativo. Entre las posibles soluciones que plantea el profesor Aronson destaca: rotación en la elección del líder, la existencia de grupos de expertos puede solucionar el problema de alumnos más rezagados, sirviendo como fase previa a la actualización y nivelación de contenidos entre los grupos, el aburrimiento de alumnos brillantes suele ser un problema común, pero que se manifiesta en menor medida, debido a la interacción que propone la técnica y a pesar de que, a mayor edad, peor asimilación de la técnica, todos los estudiantes afectados pueden beneficiarse de la misma [53].

2.2 Pensamiento computacional

El término Pensamiento Computacional ha sido popularizado por Jeannette M. Wing en el 2006 [54]; desde entonces, ha venido cobrando gran importancia por ser considerado como una habilidad del Siglo 21, la cual deben desarrollar todas las personas para lograr ser competitivos en una economía global [55].

Es así como el pensamiento computacional consiste en la resolución de problemas, el diseño de los sistemas, y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de la informática [56], además, el pensamiento computacional no es sinónimo de capacidad para programar un ordenador, puesto que requiere pensar en diferentes niveles de abstracción y es independiente de los dispositivos, convirtiéndose en una competencia básica que todo ciudadano debería conocer para desenvolverse en la sociedad digital, pero no es una habilidad «rutinaria» o «mecánica», ya que es una forma de resolver problemas de manera inteligente e imaginativa [57].

Existen distintas iniciativas y herramientas educativas para enseñar el pensamiento computacional [58] como ChildProgramming [59], Scratch [60], Alice [61], entre otros, lo que ha conllevado a que los sistemas educativos estén incorporando en sus currículos oficiales nuevos conocimientos relacionados con el pensamiento computacional [57].

Son muchas los beneficios que conlleva la apropiación del pensamiento computacional, Barceló establece que la ventaja de desarrollar el pensamiento computacional es entonces evidente: permite la transformación de una sociedad formada por meros consumidores de tecnología, en una de potenciales desarrolladores de ésta [62]. A su vez, Zapotecatl determina que el objetivo del pensamiento computacional es desarrollar sistemáticamente las habilidades de pensamiento crítico y resolución de problemas con base en los conceptos de la computación, donde los estudiantes y profesionales tendrán la necesidad de aprender y practicar las habilidades del PC para poder utilizar las nuevas tecnologías y confrontar los desafíos del Siglo XXI [63].

2.2.1 Dimensiones del pensamiento computacional

Karen Brennan y Mitchel Resnick del MIT Media Lab, en su artículo "Nuevas propuestas para estudiar y evaluar el desarrollo del pensamiento computacional" [64], desarrollaron una definición de Pensamiento Computacional que incluye tres dimensiones [55]:

Conceptos computacionales: están relacionados con los conceptos computacionales que emplean los diseñadores cuando están programando y que son comunes a varios lenguajes de programación; entre estos se encuentran secuencias, ciclos, eventos, paralelismo, condicionales, operadores y datos.

Prácticas computacionales: tienen relación con las prácticas que van realizando los diseñadores a medida que programan, es decir, los procesos de construcción que se emplean cuando se crean los proyectos, los cuales se enfocan a los procesos de pensar y de aprender, al cómo está aprendiendo. Las prácticas identificadas son:

- ✓ Ser incremental e iterativo.
- ✓ Ensayar y depurar.
- ✓ Reusar y remezclar.
- ✓ Abstraer y modularizar.

Perspectivas computacionales: se toma la perspectiva como la forma de ver o considerar las cosas. Esta dimensión tiene relación con las formas de ver de los diseñadores cuando construyen sobre el mundo que está a su alrededor y en ellos mismos; dentro de esta se encuentra:

- ✓ Expresar sus ideas a través de la computación.
- ✓ Conectarse con otras personas, proyectos y perspectivas a través de las redes, creando con otros o para otros.
- ✓ Preguntar sobre y con las tecnologías de la información y la comunicación TIC o responder a cuestionamientos a través del mismo diseño.

2.2.2 Etapas del pensamiento computacional

De acuerdo al estudio realizado por Espino y Gonzales [58], la Sociedad Internacional de la Tecnología en la Educación (ISTE) y la Asociación de Profesores de Informática (CSTA) han colaborado con líderes del mundo de la investigación y la educación superior, la industria y la educación primaria y secundaria para desarrollar estudios en el campo del pensamiento computacional [65], en ellos se determina las siguientes etapas en el proceso de solución de problemas mediante este pensamiento:

- ✓ **Formular problemas.** Para lo cual se permite el uso del computador y otras herramientas para ayudar a resolverlos.
- ✓ **Organizar y analizar lógicamente la información.** Es necesario representar la información con elementos de diseño basados en modelos y simulaciones.
- ✓ **Automatizar las soluciones.** Aquí es necesario utilizar el pensamiento algorítmico como elemento clave del proceso.
- ✓ **Identificar, analizar e implementar posibles soluciones.** Su objetivo es lograr la combinación más efectiva y eficiente de pasos y recursos para obtener una solución.
- ✓ **Generalizar y transferir.** Busca establecer los procesos de resolución de problemas para solucionar una gran variedad de familias de problemas.

2.2.3 Características del pensamiento computacional

Wing [56] determina que el pensamiento computacional tiene una serie de rasgos que lo identifican:

En el pensamiento computacional se conceptualiza, no se programa. Es preciso pensar como un científico de la computación. Se requiere un pensamiento en múltiples niveles de abstracción.

En el pensamiento computacional son fundamentales las habilidades no memorísticas o no mecánicas. Memoria significa mecánico, aburrido, rutinario. Para programar los computadores hace falta una mente imaginativa e inteligente. Hace falta la emoción de la creatividad. Esto es muy parecido al pensamiento divergente, tal como lo concibieron Polya (1989) y Bono (1986).

En el pensamiento computacional se complementa y se combina el pensamiento matemático con la ingeniería. Ya que, al igual que todas las ciencias, la computación tiene sus fundamentos formales en las matemáticas. La ingeniería nos proporciona la filosofía base de que construimos sistemas que interactúan con el mundo real.

En el pensamiento computacional lo importante son las ideas, no los artefactos. Quedan descartados por tanto la fascinación y los espejismos por las novedades tecnológicas. Y mucho menos estos factores como elementos determinantes de la resolución de problemas o de la elección de caminos para resolverlos.

Lo anterior hace que el pensamiento computacional sea importante plantearlo en el contexto de un análisis y de una elaboración interdisciplinaria, ver las implicaciones que tienen estas ideas para una redefinición de un dominio teórico específico dentro de las teorías del aprendizaje y desde luego definir descriptivamente en un primer acercamiento de un currículo adecuado a esos dominios conceptuales para las distintas etapas educativas y para la capacitación de maestros y profesores [66].

2.2.4 El pensamiento computacional como habilidad de la era digital

En el documento presentado por Espino y Gonzales denominado “Estudio sobre diferencias de género en las competencias y las estrategias educativas para el desarrollo del pensamiento computacional” [58], se afirma que:

...las competencias relacionadas con la programación se están considerando destrezas básicas e instrumentales en la Sociedad del Conocimiento junto con las clásicas de lectoescritura y matemáticas, algo que rompe con la cultura tradicional del proceso de enseñanza aprendizaje y exige rápidas adaptaciones por parte del colectivo educativo ante el surgimiento de aceleradas modificaciones. De esta forma, podemos ver que diferentes países se sitúan como pioneros en la enseñanza de programación porque ya han comenzado a desarrollar acciones para promover la adquisición del pensamiento computacional, tales como Nueva Zelanda, Corea del Sur, los EE.UU, Israel y el Reino Unido. Sin embargo, todavía para el resto de países no son prioritarias ni se ha extendido su presencia en sus políticas educativas. A pesar de esto, existen algunos hitos recientes que muestran la preocupación de incorporar enseñanzas de esta índole en la población, debido a que se considera que son imprescindibles para el desarrollo de la nueva economía digital. Por ejemplo, 2014 ha sido declarado Año del Código

(Year of Code) y también en 2014, en Reino Unido se lanzó una nueva iniciativa para capacitar a los docentes en programación y así aumentar la calidad de su enseñanza para el alumnado. Esto se debe a que existe una preocupación de que a pesar de la crisis económica actual, hay y habrán en el futuro millones de puestos de trabajo sin cubrir en el sector tecnológico porque la población no está preparada para desempeñar estas funciones [58].

2.3 Metodologías colaborativas para la enseñanza aprendizaje de programación

Es evidente la preocupación existente en los docentes de los primeros cursos de programación con relación a los resultados obtenidos en el proceso de aprendizaje, por ello se han llevado a cabo en los últimos años varios proyectos encaminados a mejorar dichos procesos donde la mayor atención se enfoca en los primeros niveles de formación. A continuación se presenta un listado de las diversas metodologías vigentes que sirven como herramientas de apoyo en la consolidación de las competencias necesarias que un desarrollador de software debe tener en su formación:

El Trabajo Colaborativo como Estrategia para Mejorar el Proceso de Enseñanza-Aprendizaje – Aplicado a la Enseñanza Inicial de Programación en el Ambiente Universitario. Presenta la incorporación de herramientas que contribuyan al aprendizaje colaborativo en la enseñanza de programación, este tema es de gran interés en el aprendizaje de los primeros niveles de programación en la Universidad y ha sido abordado también por otros autores de hecho las herramientas colaborativas pueden fortalecer aspectos como el razonamiento, el auto-aprendizaje y el aprendizaje colaborativo [67].

De igual manera, VPL (Virtual Programming Lab) es un gestor de prácticas de programación sobre Moodle, de esta forma se permite la incorporación del ambiente de desarrollo de software al aula virtual de las materias donde se utiliza, posibilitando la entrega, edición y ejecución de las prácticas de programación, además de una evaluación continua y automática de estas [67].

Por otro lado, EDCIA (Entorno Colaborativo para el diseño e implementación de algoritmos) es un entorno basado en una herramienta que integra el trabajo con metodologías de diseño de algoritmo y lenguajes de programación. Su principal aporte es adicionar la posibilidad de trabajo colaborativo en la edición de algoritmos, la misma permitirá que alumnos y profesores, vean y editen el mismo diseño simultáneamente, donde mediante un cambio de rol se podrá rotar entre visualización, edición y supervisión [68].

La metodología para el aprendizaje de la Programación Orientada a Objetos basada en Programación Extrema es un proyecto que consiste en la enseñanza de Java como lenguaje de Programación Orientado a Objetos, a través del diseño e implementación de una metodología de aprendizaje basada en la integración de herramientas de Tecnologías de la Información y Comunicaciones como lo son Cmap Tools para el modelado del conocimiento y BlueJ como un ambiente de desarrollo integrado e interactivo para aprender a programar en Java [69].

Así también, Millenium es el nombre del prototipo que valida el modelo de integración de los dos ambientes, Individualizado y el Colaborativo el cual está dirigido a la población universitaria que comienza el estudio del área de Algoritmos y Programación y también a todas aquellas personas que quieran re-afianzar sus conocimientos [67]. El prototipo funciona bajo dos tipos de ambientes de aprendizaje: el Individualizado y el Colaborativo. Los usuarios cuando lo deseen pueden pasar de un ambiente de aprendizaje a otro [70].

Otro proyecto importante es el entorno de integración de PBL y CSCL para la enseñanza de algoritmos y programación en ingeniería que presenta un ambiente instruccional que integra los paradigmas del aprendizaje basado en problemas (PBL) y el Aprendizaje Colaborativo apoyado en computador (CSCL). SÁBATO es el nombre del ambiente, el cual es utilizado en la enseñanza de la asignatura algoritmos y programación en ingeniería. [71].

Un proyecto de gran interés es SCRATCH que se presenta como un lenguaje de programación que facilita crear sus propias historias interactivas, animaciones, juegos, música y arte; además, le permite compartir con otros sus creaciones en la web. A medida que los jóvenes crean y comparten proyectos realizados en Scratch, aprenden importantes ideas matemáticas y computacionales; al mismo tiempo que aprenden a pensar creativamente, a razonar sistemáticamente y a trabajar colaborativamente [72]. Scratch se caracteriza por: es programación orientada a objetos, con las estructuras básicas como: asignaciones, condicionales, bucles y, funciones; Interfaz intuitiva y muy gráfica; Existencia de un banco de recursos en el propio programa: objetos, personajes ,escenarios, sonidos; La acciones de los personajes se forman apilando piezas a modo de puzle; Visualización instantánea de lo que hacemos en el programa; Interacción con los objetos y personajes que introducimos con el ratón y teclado, es decir, una vez que creamos nuestro proyecto el usuario puede interactuar moviendo los objetos en la pantalla; Posibilidad de difundir nuestro proyecto en la web [73].

Otro estudio analiza la Enseñanza de Programación Orientada a Objetos mediante el desarrollo de aplicaciones graficas interactivas, el cual introduce al estudiante mediante aplicaciones prácticas interactivas basadas en videojuegos a la práctica de la programación Orientada a Objetos mediante una metodología activa que conlleva al estudiante a un nivel de mayor profundización en el desarrollo de soluciones software que otro homólogo en una disciplina afín [74].

Un referente importante es Cupi2 de la Universidad de los Andes que tiene como objetivo la búsqueda de nuevas maneras para enfrentar el problema de enseñar a [24]. Cupi2 como proyecto de investigación se encuentra soportado por cuatro componentes que permiten la construcción de una solución balanceada frente problema de aprender a programar. Cada uno de estos componentes ha sido validado en experiencias mundiales de enseñanza de diferentes campos de la ciencia, teniendo importante impacto en la educación en ingeniería [75]. En esta metodología se concibe: Aprendizaje activo, aprendizaje basado en problemas y proyectos, aprendizaje incremental y el aprendizaje basado en ejemplos.

El proyecto denominado “Herramientas y Técnicas para la Enseñanza de la Programación” determina que la enseñanza de la programación se debe sustentar en estrategias didácticas usando la mayoría de los medios que existen en la tecnología de la información, por lo que establece que la creación de estrategias o medios didácticos permiten usar una gran variedad de elementos tanto manuales cómo automáticos. El software en la enseñanza es un medio que facilita el logro de los objetivos, ayuda al profesor y al alumno en su proceso de enseñanza-

aprendizaje, uno cómo el facilitador y el otro cómo generador de su propio conocimiento y combinados adecuadamente permitirán la reducción del tiempo en aprender a programar [76].

Por otro lado, GREENFOOT es un Mundo Visual Interactivo en el que se enseña la orientación a objetos con Java y crea "actores" que viven en "mundos" para construir juegos, simulaciones y otros programas gráficos. Greenfoot es visual e interactivo, los actores se programan en código Java textual estándar, proporcionando una combinación de experiencia en programación en un lenguaje basado en texto tradicional con la ejecución visual [77].

El proyecto JeCo brinda a los estudiantes la posibilidad de colaborar entre sí con las herramientas que se hacen para su propósito. Los estudiantes pueden adquirir vocabulario relacionado con los conceptos de programación de Jeliot y así se crea un contexto para las discusiones. En el caso de JeCo, este vocabulario se convertirá en un conjunto intersubjetivo de conceptos compartidos [78].

Igualmente, ProLearn es un proyecto que se basa en Simulación soportada en animación que se ha propuesto para reducir las dificultades de los estudiantes en el aprendizaje de la programación. ProLearn puede promover la programación dinámica concreta de un programa y visualmente apoyar el trabajo práctico en el propio ritmo de aprendizaje del estudiante. Las vistas animadas pueden ayudar a muchos estudiantes en tres actividades de aprendizaje centrales: Comprender los programas; Evaluar los programas existentes; Desarrollar nuevos programas [79]. Es una herramienta que ayuda a los programadores noveles a resolver problemas usando comunicación basada en texto. El principal objetivo del sistema es la atención del estudiante, de la manera más cercana a los problemas que vaya teniendo en el aprendizaje de programación [11].

PL-DETECTIVE es un proyecto que permite aprender nociones de programación de manera colaborativa [11], fue diseñado para la construcción asignaciones y demostraciones en el contexto de un curso titulado "Principios de lenguajes de programación". PL-Detective asocia elementos de trabajo colaborativo y directamente exponen a los estudiantes a problemas que no tienen una respuesta clara. El PL-Detective es una implementación flexible y extensible del lenguaje llamado "Mystery" [80].

En el proyecto denominado COLLEGE se modela una propuesta de sistema colaborativo síncrono para el aprendizaje de la programación que puede ser utilizado tanto en los laboratorios de prácticas como desde otros espacios (en casa, bibliotecas...). De acuerdo con este modelo se construyó un sistema que soporta la edición, compilación y ejecución de programas en grupo por parte de los alumnos, y la revisión del trabajo efectuado por los estudiantes por parte del profesor para extraer conclusiones sobre el proceso de programación en grupo [81].

El estudio OOP ANIM fue desarrollado para ayudar a estudiantes a aprender los conceptos básicos de la programación orientada a objetos y al desarrollar sus capacidades de programación que utiliza este paradigma. Para lograr los objetivos propuestos los estudiantes deben practicar intensamente el desarrollo y la depuración de los programas. Se piensa que este entorno puede ayudar, ya que utiliza animación para facilitar la comprensión y el error del programa detección / corrección. Este proceso de depuración tiene una gran cantidad de potencial educativo, ya que los estudiantes pueden aprender cuando corregir sus propios

errores. Cuando llegan a un trabajo solución, la experiencia y la confianza normalmente mejora, lo que facilita el aprendizaje posterior [82].

Otro referente es DPE que es el Entorno de Programación Distribuida (DPE) donde los programadores y gerentes de proyectos (o profesores y estudiantes) pueden ser ubicado en cualquier lugar y efectivamente realizar procesos colaborativos una vez que se conectan entre sí a través de Internet, apoyados en herramientas de comunicación utilizando componentes multimedia en tiempo real entre los participantes y de esta manera desarrollar un proyecto conjunto de software [83].

Así mismo, ELP es un entorno que permite a los estudiantes programar en sus primeros cursos sin la necesidad de familiarizarse con sintaxis de lenguajes de programación. Lo más importante es que elimina todas las dificultades asociadas con la instalación y ejecución de un Compilador de Java. Con ELP los estudiantes aprenden y desarrollan habilidades trabajando con las plantilla de ejercicios en la web. ELP ofrece un entorno de aprendizaje que satisface las diversas necesidades de los estudiantes [8].

Por otro lado, BlueJ es un Entorno de desarrollo Integrado (IDE) específicamente desarrollado para enseñar y aprender Programación Orientada a Objetos y presente un enfoque pedagógico desarrollado para ser utilizado con un sistema como el de BlueJ en el cual se desarrollarán ejemplos concretos mediante la presentación de una secuencia de asignaciones destinadas a apoyar y explotar las ideas pedagógicas y posibilidades técnicas del entorno [84].

Igualmente ALICE es un programa para construir mundos virtuales con objetos 3D, donde los objetos pueden moverse, girar, cambiar color, reaccionar al ratón y mucho más. Su interfaz interactiva genera instrucciones al arrastrar y soltar elementos gráficos (drag and drop). Estas instrucciones se corresponden a lenguajes de programación como Java o C++. Al ver en forma inmediata cómo corren los programas de animación, los estudiantes pueden entender con mayor facilidad la relación entre el código y el comportamiento de un objeto. Su entorno visual mejora la retención y el aprendizaje, evitando la frustración de una sintaxis mal utilizada. Los objetos se vuelven obvios y el estudiante puede relacionarse con ellos y la forma en que se programan [85].

El proyecto HABIPRO establece cómo el aprendizaje cooperativo apoyado en computador (CSLC) se puede implementar ara desarrollar nuevas habilidades en estudiantes universitarios que condujeron al desarrollo de un ambiente adaptativo para el manejo de buenos hábitos de programación [86].

Programin en el laberinto es un proyecto que propone la utilización de un programa de ordenador, no para programar, sino para aprender a programar. Se trata de aprovechar las ventajas que nos ofrece el ordenador como recurso didáctico. La utilización de dicho programa en las clases prácticas de la asignatura nos permitirá reforzar los conceptos teóricos, al mismo tiempo que analizar interactivamente los algoritmos diseñados [87].

Así mismo, ALGOARENA es un software educativo para el aprendizaje de programación considerado como un sistema educativo de colaboración a través del cual los estudiantes pueden aprender a programar, participando conjuntamente en la formación de una comunidad de práctica y su identidad en ella [88].

Por otro lado, SIGACLE es un Simulador gráfico asistido por computador para la enseñanza de programación de lenguajes estructurados y su objetivo pedagógico es presentar al estudiante vistas diferentes de lo que sucede en la máquina al ejecutar un programa. Para esto utiliza técnicas de compilación operando en fases bien definidas, cada una de las cuales transforma el código fuente de una representación en otra. Simular gráficamente la depuración de un programa es un concepto que intenta resolver el problema que tiene el estudiante cuando escribe un programa de computador. Al principio, el estudiante no es consciente de la importancia de utilizar metodologías para el desarrollo de programas; ya sean estrategias de planeación y modelaje, o el uso de técnicas formales como el *Personal Software Process (PSP)* [89].

En la tabla 2.1 se categorizan cada una de las metodologías mencionadas de acuerdo a sus características afines.

Tabla 2.1. Metodologías para la enseñanza aprendizaje de los fundamentos de programación de computadores

Metodología	Genera código	Editor gráfico	Software de modelado	Chat	Edición compartida	Síncrono	Asíncrono	conferencia	Utiliza lenguajes de programación	Roles docente y estudiante	Aprendizaje colaborativo
Algoarena						X				X	X
Alice		X				X			X		
BlueJ	X	X	X				X		X		X
COLLEGE			X	X	X	X					X
Cupi 2				X	X	X	X	X	X	X	
DPE			X	X	X	X	X	X		X	X
EDCIA	X	X	X		X	X				X	X
ELP		X				X					X
Grenfoot		X				X			X		
HABIPRO			X	X	X	X				X	X
JeCo			X	X	X	X				X	X
Millenium						X				X	X
OOP ANIM			X	X	X	X			X	X	X
PBL y CSCL			X	X	X	X	X	X	X		X
PLDETECTIVE	X	X	X						X		X
Programin		X					X				
ProLearn		X	X							X	
Scratch		X	X			X					
Sigacle	X	X				X					
VPL				X	X	X				X	X

CAPÍTULO 3: METODOLOGÍA BASADA EN APRENDIZAJE COLABORATIVO Y MODELADO GRÁFICO

3.1 Propuesta metodológica

La presente propuesta metodológica pretende aportar en la enseñanza de los fundamentos de programación combinando elementos de modelado gráfico en un entorno colaborativo, permitiendo al estudiante contextualizar de manera visual y sencilla los conceptos abstractos básicos para la fundamentación de los procesos lógicos necesarios para afrontar los diversos paradigmas de programación en asignaturas posteriores.

La metodología contempla dos etapas: la primera corresponde a la fundamentación teórica con modelado gráfico, la cual se centra en el estudio conceptual de las unidades y elementos de competencia; y la segunda etapa, contempla el proceso colaborativo donde el estudiante desarrolla su pensamiento computacional mediante elementos de análisis y diseño.

La metodología se resume en la tabla 3.1 en donde se destacan las etapas mencionadas y documentos a diligenciar.

Tabla 3.1 Resumen metodología

Etapas: Fundamentación teórica con modelado gráfico						
Formato A. Configuración inicial	Configuración inicial	Planeador de ejemplos	Recolección de requerimientos	Fase de análisis		Fase de diseño
				D E/S	GUI	Algoritmo
	Formato B. Configuración inicial para fundamentación conceptual de modelado gráfico	Formato C. Planeación de ejemplos de la configuración inicial para fundamentación conceptual de modelado gráfico	Formato D. Formato de recolección de requerimientos	Formato E. Diagrama de Entrada / Salida	Formato F. Interfaz gráfica de Usuario	Formato G. Diagrama de flujo
Etapas: Proceso colaborativo						
Configuración inicial				Planeador de ejercicios		
Formato H. Configuración inicial caso de estudio				Formato I. Planeación de ejercicios de configuración inicial para el caso de estudio		

3.1.1 Configuración inicial

Este formato es informativo y pretende instruir al docente en las actividades a realizar durante el desarrollo de todo el curso mediante el planteamiento de preguntas orientadoras y una serie de recomendaciones que le permitirán la implementar la metodología propuesta. Todo esto enmarcado dentro de un entorno colaborativo en el cual interviene los roles de estudiante y docente.

La configuración inicial se encuentra en el formato A (Anexo 1) en el cual se registran actividades referentes al diseño de tareas grupales, planeación de tiempo, recursos o

materiales, conformación de grupos y la distribución del espacio físico. Dichas actividades son analizadas mediante preguntas orientadoras y actividades recomendadas según los aportes de Collazos y Mendoza [47]. El formato A se presenta en la tabla 3.2.

Tabla 3.2 Formato A

Formato A. Configuración inicial		
Actividad	Pregunta orientadora	Actividades recomendadas
Diseñar tareas grupales		
Planear el tiempo estimado para la realización de la tarea		
Determinar Recursos/Materiales		
Formar los grupos		
Distribuir adecuadamente el espacio físico		

3.1.2 Etapa: Fundamentación teórica con modelado gráfico

En esta etapa se fundamenta al estudiante en los conceptos básicos de cada unidad y elemento de competencia propuestos para cursos de fundamentos de programación. Para ello, la metodología se apoya en un entorno colaborativo donde el docente incorpora características de diseñador instruccional, instructor y mediador cognitivo; y el estudiante lo realiza mediante los sub roles de analista y diseñador.

3.1.2.1 Configuración inicial para modelado gráfico

En este formato se realiza una caracterización del curso de fundamentos de programación, incluyendo elementos de identificación del curso y del docente cargo, además, contiene elementos propios del espacio académico como: descripción, competencia central, unidades y elementos de competencia, recursos, distribución en el aula de clase y los referentes bibliográficos.

El formato B (anexo 2) contiene los elementos de configuración inicial para fundamentación conceptual de modelado gráfico (tabla 3.3).

Tabla 3.3 Formato B

Formato B. Configuración inicial para fundamentación conceptual de modelado gráfico				
Identificación del curso				
Nombre del curso				
Área		Componente		Créditos
Código del curso		Semestre	Jornada	
Identificación del docente				
Nombre del docente				
E-mail				
Descripción del curso				

Competencia central		
Unidades y elementos de competencia		
Unidad de competencia	Elementos de competencia	Horas
Recursos y/o materiales		
Distribución en el aula de clase		
Bibliografía		

3.1.2.2 Planeador de ejemplos

Es necesario que el docente realice una planeación adecuada de los ejemplos que se incluirán en la fundamentación teórica tanto de las unidades como de los elementos de competencia que serán presentados, analizados y construidos con la propuesta de modelado gráfico. Dichos ejemplos deben ser categorizados de acuerdo a la unidad y elemento de competencia.

En el formato C (Anexo 3) contiene los campos necesarios para recolectar la información del planeador de ejemplos necesarios para un curso de fundamentos de programación (tabla 3.4).

Tabla 3.4. Formato C

Formato C: planeación de ejemplos de la configuración inicial para fundamentación conceptual de modelado gráfico	
Unidad de competencia	Enunciados del ejemplo

3.1.2.3 Recolección de requerimientos

Uno de los aportes esenciales de esta propuesta metodológica consiste en informar e involucrar al estudiante, desde su primer curso de fundamentos de programación, con las etapas claves de construcción de todo producto de software. Por ello, es necesario que él identifique y se familiarice con algunas de las más importantes como lo son requerimientos, análisis y diseño.

Es importante que el estudiante conceptualice correctamente la situación planteada por su docente para dar una solución acertada a los requerimientos que exige el planteamiento de cada ejemplo y ejercicio de clase. Es así como en el formato D (anexo 4) se recolectan los elementos claves del ejercicio planteado (tabla 3.5).

Tabla 3.5. Formato D

Formato D: Recolección de requerimiento principal	
Nombre del sistema	Requerimiento principal

3.1.2.4 Fase de análisis

Esta propuesta pretende que el estudiante identifique esta importante fase en el proceso de construcción de software, para ello, se ha planteado realizarla a través de dos diagramas claves para el proceso de modelado gráfico aquí planteado, estos dos diagramas son: Diagrama de entrada/salida (D E/S) e Interfaz gráfica de usuario (GUI) cuyos formatos E y F (tabla 3.6) son informativos y contienen los elementos para su respectiva construcción y el formato de la tabla 3.5 permite su descripción (Anexo 5, 6 y 7).

Tabla 3.6. Formato E, F y G

Formato E, F, G	
Símbolo	Descripción

3.1.2.5 Fase de diseño

Permite realizar un primer modelo de acuerdo al análisis presentado mediante D E/S e GUI que son insumos primordiales de la fase de diseño. Esta fase modela la solución utilizando diagramas de flujo que son elementos claves en toda metodología de construcción de software del paradigma imperativo (Orientado a Objetos, Procedural y Orientado a Eventos). El formato G (anexo 7) es informativo y en este se encuentra la descripción y los símbolos de los principales artefactos que utiliza el diagrama de flujo para realizar el modelado (tabla 3.6) requerido en este estudio.

3.1.3 Etapa: proceso colaborativo

Esta etapa se desarrolla bajo un entorno colaborativo en el que el docente, a través del planteamiento de casos de estudio y utilizando el método JIGSAW, genera una dinámica en el grupo de estudiante quienes en sus sub roles de analista y diseñador desarrollan su pensamiento computacional mediante elementos de análisis y diseño.

3.1.3.1 Configuración inicial caso de estudio

En este formato se realiza una caracterización del proceso colaborativo que se llevará a cabo en cada una de las unidades de competencias planteadas para el curso de fundamentos de programación, en el que se establecen objetivos individuales y grupales por cada unidad, se planea la conformación de los grupos y su distribución en el aula de clase y la planeación de recursos tanto para la fase de análisis como para la de diseño propuestas.

El formato H (anexo 8) contiene los elementos de configuración inicial para caso de estudio (tabla 3.7).

Tabla 3.7. Formato H

Formato H: Configuración inicial para el caso de estudio				
Identificación del curso				
Nombre del curso				
Código del curso		Semestre		Jornada
Identificación del docente				
Nombre del docente				
E-mail				

Unidad de competencia				
Fase	Análisis			
Tiempo de trabajo	Unidad de competencia	Elemento de competencia	Horas	
Objetivo general				
Objetivo individual				
Objetivo grupal				
Conformación de grupos				
Distribución en el aula				
Planeación de recursos				
Unidad de competencia				
Fase	Diseño			
Tiempo de trabajo	Unidad de competencia	Elemento de competencia	Horas	
Objetivo general				
Objetivo individual				
Objetivo grupal				
Conformación de grupos				
Distribución en el aula				
Planeación de recursos				

3.1.3.2 Planeador de ejercicio

Es necesario que el docente disponga de una serie de ejercicios pertinentes a cada unidad de competencia, para ello es recomendable que se aborden teniendo en cuenta su nivel de complejidad, es decir, comenzando por aquellos que revisten un nivel bajo de complejidad y finalizando en cada unidad con los de mayor complejidad.

El formato I (Anexo 9) contiene la estructura requerida para el registro de los ejercicios a desarrollar en el caso de estudio para cada unidad y elemento de competencia (tabla 3.8).

Tabla 3.8. Formato I

Formato I: Planeación de ejercicios de configuración inicial para el caso de estudio	
Unidad de competencia	Enunciados del ejemplo
Unidad 1:	
Unidad 2:	
Unidad 3:	

3.2 Descripción del Modelo Colaborativo

La metodología propuesta cuenta con un modelado colaborativo (ver figura 3.1), cuya notación de representación es propia de esta investigación, en el que intervienen dos roles: docente y

estudiante. El rol docente asumirá características de mediador cognitivo, instructor y diseñador instruccional para la realización de una configuración inicial por cada temática orientada la cual tendrá dos momentos: la primera corresponde a nivel de la orientación al grupo de estudiantes respecto de la fundamentación conceptual de las temáticas bajo un modelado gráfico y en la segunda presentará casos de estudio utilizando el método JIGSAW. Por otra parte, el estudiante actuará en dos roles: Analista y diseñador orientado por principios colaborativos que le permitan interactuar con los demás compañeros en los casos de estudio planteados.

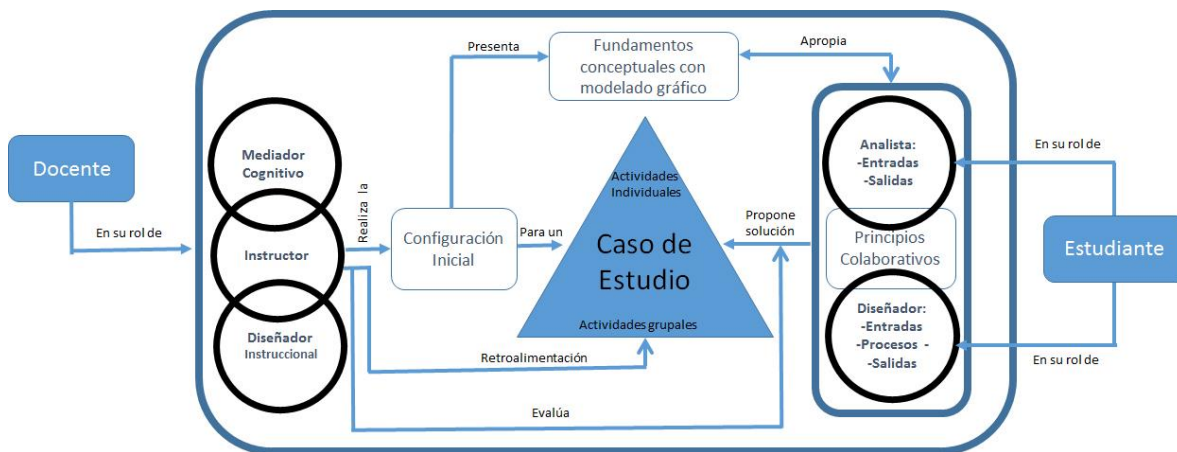


Figura 3.1. Modelo colaborativo

JIGSAW fue utilizado en esta investigación como método colaborativo debido a que ha sido empleado de en diversos procesos de aprendizaje con resultados positivos debido al hecho de convertir a cada estudiante en elemento esencial para la resolución de una situación, lo cual provoca una mayor implicación en el alumnado y, por tanto, mejores resultados globales, constituyendo así una estrategia eficaz de aprendizaje[53]. Además, con esta técnica el estudiante es responsable de conocer a fondo la información que le corresponde, para luego enseñarla y aprender la información presentada por los otros miembros del equipo [90]. Por otro lado, la utilización de JIGSAW permite evidenciar en los estudiantes niveles de motivación y de atención mayor con respecto a la clase tradicional [91] y quienes han participado de ella afirmaron no sentirse aburridos ni estresados a lo largo de varias experiencias [92]; finalmente, JIGSAW ha demostrado su utilidad para reducir la ansiedad y el rechazo a la participación de los estudiantes [93]. Por ello, en este estudio se decidió incorporar a JIGSAW como técnica colaborativa que combinada con una propuesta basada en modelado gráfico permitan mejorar los procesos de enseñanza de los fundamentos de programación.

3.2.1 Rol docente

En el modelo colaborativo propuesto, el docente asume una serie de funciones propias de los roles como diseñador instruccional, mediador cognitivo e instructor, cuyo esquema colaborativo se presentan en la figura 3.2.

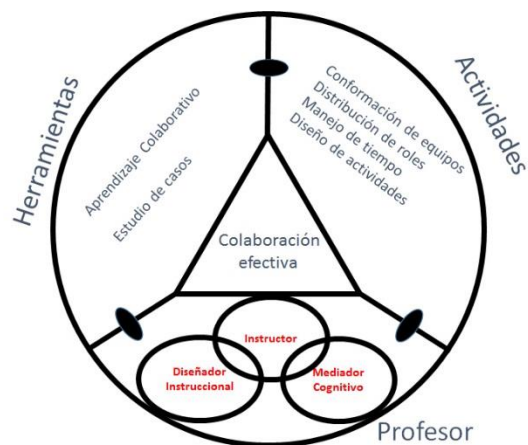


Figura 3.2. Rol docente

Como diseñador instruccional: el docente es quien realiza la planeación tanto de las unidades temáticas como de las actividades de aprendizaje y evaluación que se llevarán a cabo durante el desarrollo del curso de fundamentos de programación, para ello es necesario que el docente incorpore un syllabus para la signatura en donde se debe [110] [111]:

- ✓ Planear los objetivos académicos
- ✓ Determinar claramente las unidades temáticas y los conocimientos mínimos que deben ser adquiridos durante el proceso de enseñanza en cada una de ellas
- ✓ Dividir el tema en subtemas
- ✓ Realizar una sesión inicial de motivación donde se exploren preconcepciones, intereses particulares y grupales de los estudiantes y finalmente se exhiba un panorama general de los logros a alcanzar.

Además, el docente en este rol debe realizar algunas actividades en las sesiones de clase como: determinar la disposición física de los estudiantes en el salón de clase, establecer las normas de conformación de grupos de trabajo, distribución de materiales de trabajo, determinar las normas de comportamiento y participación de los estudiantes tanto al interior de cada subgrupo como en el grupo total.

Como mediador cognitivo: el docente será el encargado de validar el conocimiento adquirido por el estudiante mediante la utilización de diversas estrategias de seguimiento que deben ir desde la observación directa con la cual se puede evidenciar el interés y los alcances conceptuales logrados, la utilización de preguntas de verificación, la realización de actividades de aprendizaje y evaluación que permitan directamente comprobar el aprendizaje de las temáticas.

El proceso de aprendizaje en este rol será soportado por la estrategia didáctica denominada JIGSAW que es un método en el que los alumnos trabajan en equipos heterogéneos, donde el profesor se encarga de dividir el tema que quiere enseñar en partes, cada una de ellas única e imprescindible, cada alumno recibe una parte del tema como una pieza de un puzle JIGSAW y ha de juntarla con las partes que tienen sus compañeros para aprender el cuadro completo, y para ello, el alumno debe estudiar individualmente su parte, discutirla dentro de su grupo de expertos y después enseñarla al resto de su equipo [51]. Con la aplicación de esta estrategia

se logra en el estudiante el manejo de una interdependencia positiva, debido a que cada estudiante tiene diferente información necesaria para completar la tarea [47].

Como profesor instructor: el docente se apoya la clase magistral como estrategia didáctica en la que se encarga tanto de la enseñanza de las unidades temáticas en este caso de la asignatura de fundamentos de programación, como de desarrollar en los estudiantes habilidades sociales, de trabajo en grupo y fundamentalmente actividades colaborativas, modelando habilidades interpersonales positivas y conllevándolos a su práctica [47], para lo cual y apoyado en Johnson & Johnson [94] se requiere tres pasos :

- ✓ chequear si los estudiantes están trabajando juntos.
- ✓ chequear si los estudiantes están haciendo el trabajo bien.
- ✓ observar y dar retroalimentación.

Además, con respecto a la evaluación, los docentes necesitan conducir tres tipos de evaluaciones [47]:

- ✓ Diagnóstica: evalúa el nivel actual de conocimientos y habilidades de los estudiantes.
- ✓ Formativa: monitorea el progreso en el logro de los objetivos.
- ✓ Sumativa: provee datos para juzgar el nivel final del aprendizaje de los estudiantes

En el modelo presentado en la figura 3.2 que se apoya en el esquema colaborativo de Collazos y Mendoza [47], la colaboración efectiva se establece en la necesidad que existe entre los estudiantes de compartir información que permita abordar conceptos y obtener conclusiones de los temas tratados y que apoyados en la estrategia JIGSAW en la que se contempla la utilización de los dos roles de trabajo: Docente y estudiante permitirá finalmente mediante un proceso planeado compartir los hallazgos de conocimiento en todo el grupo.

3.2.2 Configuración inicial

En la implementación de actividades colaborativas, la configuración inicial es una etapa importante en la cual se organiza, planea y establece los materiales, las didácticas, los contenidos, los tiempos, los escenarios y demás elementos necesarios para el desarrollo de una sesión de clase en un adecuado entorno colaborativo.

Collazos y Mendoza [47] consideran que la configuración inicial para mejorar el proceso de enseñanza aprendizaje debe incluir las siguientes tareas (ver tabla 3.9) en la etapa inicial [47]:

Tabla 3.9. Formato de Configuración Inicial

Actividad	Pregunta orientadora	Actividades recomendadas
Diseñar tareas grupales	¿Cuáles son los objetivos generales y particulares de las tareas que realizarán los estudiantes?	<ul style="list-style-type: none"> ✓ Diseñe tareas que consigan que los participantes se conecten unos a otros, y que lo hagan como personas más que como aprendices. ✓ Plantee tareas que les permitan tomar decisiones individuales y grupales.
Planear el tiempo estimado para la	¿Cuánto tiempo se necesita para el cumplimiento total de la tarea, incluyendo el tiempo de formar	<ul style="list-style-type: none"> ✓ Incluya períodos de reflexión de tal forma que los participantes se enfoquen en hacer raciocinios

Actividad	Pregunta orientadora	Actividades recomendadas
realización de la tarea [95].	los grupos, dar instrucciones, explicar la tarea, llevarla a cabo, evaluar y analizar los resultados?	<p>respecto a las actividades realizadas y puedan integrar lo aprendido con lo conocido.</p> <ul style="list-style-type: none"> ✓ Puede organizar los grupos para que duren una hora, un día, una semana o más dependiendo de la actividad. ✓ Nadie sugiere que la clase sea organizada en grupos colaborativos todo el tiempo. Muchas actividades requieren una combinación entre una actividad magistral de iniciativa del profesor y una colaborativa, haga un balance.
Determinar Recursos/Materiales	¿Qué recursos/materiales se necesitan para llevar a cabo la actividad?	<ul style="list-style-type: none"> ✓ Distribuya los materiales de tal forma que los participantes entiendan que deben trabajar juntos o no lograrán la meta. Dé a cada uno solamente una parte del material que será aprendido.
Formar los grupos [49]	¿Cuántos estudiantes conformarán el grupo? ¿Qué características se considerarán para la conformación de los grupos? ¿Se tendrán grupos homogéneos o heterogéneos? ¿Cuánto tiempo estarán los estudiantes en cada uno?	<ul style="list-style-type: none"> ✓ Trate de formar grupos que sean heterogéneos en cuanto a habilidad, personalidad, nivel de escolaridad, profesión, género. ✓ En la medida que sea posible, seleccione usted mismo los grupos, especialmente cuando no son expertos en técnicas colaborativas. ✓ Mientras más pequeños sean mucho mejor. Se recomiendan grupos de no más de cuatro personas. ✓ Utilice diferentes estrategias para formarlos.
Distribuir adecuadamente el espacio físico	¿Cómo distribuir los estudiantes dentro del salón de clases? ¿Cómo organizar los estudiantes dentro de cada grupo?	<ul style="list-style-type: none"> ✓ Ubique los grupos con suficiente distancia entre unos y otros para que no interfieran entre sí. ✓ Organice los participantes del mismo grupo buscando que queden juntos. ✓ El salón debe ser lo suficientemente flexible para que los participantes se puedan mover y cambiar de lugar libremente. ✓ Varíe la posición del escritorio del profesor.

De acuerdo al modelo presentado en la figura 3.1 la configuración inicial se realiza tanto para el caso de estudio como para la fundamentación conceptual con modelado gráfico que se describen a continuación:

3.2.2.1 Configuración inicial para caso de estudio

En esta etapa se deben llevar a efecto las siguientes actividades:

- ✓ Descripción del curso.
- ✓ Definición de objetivos individuales y grupales.
- ✓ Diseño de ejemplos y tareas.

- ✓ Estimación del tiempo para realización de ejercicios y tareas.
- ✓ Planeación de recursos y materiales necesarios para las actividades propuestas.
- ✓ Conformación de los equipos de trabajo.
- ✓ Planeación de la distribución física de los estudiantes en el aula de clase.

El formato de configuración inicial para el caso de estudio propuesto para el presente estudio se presenta en el anexo 10.

En esta fase se deben planear una serie de ejercicios que el estudiante realizará utilizando el método JIGSAW con el propósito de lograr un aprendizaje más significativo. En el anexo 11 se presenta en orden de complejidad la planeación de ejercicios los cuales pueden ser modificados por el docente según sus necesidades.

3.2.2.2 Configuración inicial para fundamentación conceptual de modelado gráfico

En esta etapa se llevaron a efecto las siguientes actividades para cada unidad de aprendizaje:

- ✓ Identificación del curso: área, componente, créditos, semestre, docente.
- ✓ Descripción del curso.
- ✓ Competencia central.
- ✓ Definición de unidades de aprendizaje con sus correspondientes subtemas.
- ✓ Diseño de ejemplos.
- ✓ Estimación del tiempo para desarrollo de temática explicativa.
- ✓ Planeación de recursos y materiales necesarios para las actividades propuestas.
- ✓ Planeación de la distribución física de los estudiantes en el aula de clase.
- ✓ Diseño de tareas y evaluaciones

En el anexo 12 se presenta el formato de configuración inicial para fundamentación conceptual de modelado gráfico para el curso propuesto.

Los ejemplos iniciales planteados en esta configuración deben ser fáciles de asimilar por el estudiante, es decir, deben encontrarse dentro de sus posibilidades de conocimiento, se recomienda presentar máximo dos ejemplos para que en el caso de estudio el estudiante realice varios ejercicios con la metodología planteada con el propósito de obtener un aprendizaje más significativo.

El en el anexo 13 se presenta el formato que contiene la planeación de algunos enunciados base los cuales se encuentran ordenados de acuerdo a su complejidad, el cual puede ser modificado totalmente acorde con las prioridades de cada docente:

3.2.3 Rol Estudiante

De acuerdo a las recomendaciones de Collazos y Mendoza [47]:

Los estudiantes que estén comprometidos con el proceso de aprendizaje deben tener las siguientes características:

- ✓ Ser responsables con el aprendizaje. Se hacen cargo de su propio aprendizaje y se autorregulan. Definen los objetivos del aprendizaje y los problemas que es son

significativos, entienden qué actividades específicas se relacionan con sus objetivos y usan estándares de excelencia para evaluar qué tan bien han logrado dichos objetivos.

- ✓ Estar motivados para aprender. Encuentran placer y excitación en el aprendizaje. Poseen pasión para resolver problemas y entender ideas y conceptos. Para estos estudiantes, el aprendizaje es intrínsecamente motivante.
- ✓ Ser colaborativos. Entienden que el aprendizaje es social. Están “abiertos” a escuchar las ideas de los demás y a articularlas efectivamente; tienen empatía con los otros y una mente abierta para conciliar con ideas contradictorias u opuestas. Poseen la habilidad para identificar las fortalezas de los demás.
- ✓ Ser estratégicos. Continuamente desarrollan y refinan el aprendizaje y las estrategias para resolver problemas. Esta capacidad para aprender a aprender (metacognición) incluye construir modelos mentales efectivos de conocimiento y de recursos, aun cuando los modelos puedan estar basados en información compleja y cambiante. Estos estudiantes son capaces de aplicar y transformar el conocimiento con el fin de resolver los problemas en forma creativa, y de hacer conexiones en diferentes niveles.

En la figura 3.3 y apoyados en el esquema colaborativo de Collazos y Mendoza [47] se muestra tanto las características del estudiante como las herramientas y actividades necesarias para realizar el proceso colaborativo de aprendizaje.

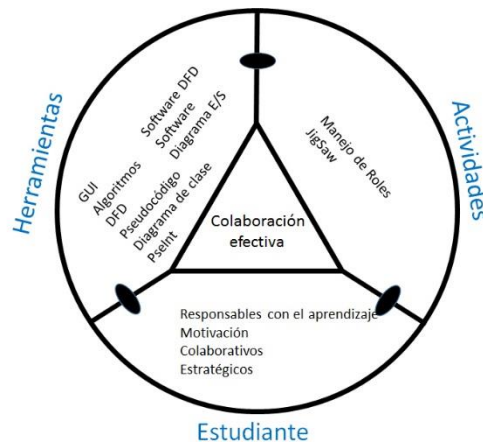


Figura 3.3. Rol estudiante

Una vez motivado el grupo de estudiantes por iniciar el proceso de aprendizaje para la asignatura fundamentos de programación bajo un entorno colaborativo, se realiza la fundamentación de cada unidad de estudio donde por cada temática se inicia su fase de análisis con los diagramas de entrada/salida e interfaz gráfica de usuario (GUI) para luego realizar la fase de diseño con la construcción de algoritmos con diagramas de flujo de datos (DFD) apoyados con la herramienta de software DFD y luego la exploración de pseudocódigo apoyado con PselInt, todo esto combinados con el método JIGSAW y el manejo de roles.

La propuesta metodológica contempla básicamente en el rol de estudiante dos sub roles: analista y diseñador. En la tabla 3.10 se presenta el manejo de los sub roles por cada unidad de aprendizaje:

Tabla 3.10. Roles y sub roles por cada unidad de aprendizaje

Unidad de competencia	Elemento de competencia	Rol 1	Rol 2
Entrada/salida	Entrada/salida	Analista de entradas	Analista de salidas
		Analista de entradas/salidas	Analista de GUI
Estructura condicional	Condicional compuesto	Analista de entradas/salidas/GUI	Diseñador
	Condicional simple		
	Condicional anidado		
	Condicional y operadores lógicos		
Estructuras iterativas	Ciclo Para	Analista de entradas/salidas/GUI	Diseñador
	Ciclo Mientras		
	Ciclo Hacer Mientras		

3.2.4 Caso de estudio

Es en el caso de estudio donde se realiza el verdadero proceso colaborativo, donde se integra el rol del estudiante y rol docente quien apoyado en una guía de modelado gráfico expone la temática inicial para luego bajo el entorno colaborativo fundamentado con el método JIGSAW y el manejo de roles y sub roles busca la apropiación de cada temática de manera individual y colectica.

En la figura 3.4 se presenta el diagrama general de actividades para el caso de estudio, donde es necesario primero realizar la configuración inicial de cada unidad temática (teniendo en cuenta el paso previo bajo la fundamentación de modelado gráfico) para luego aplicar la realización de una serie de ejercicios (propios del caso de estudio) en un entorno colaborativo mediado por JIGSAW y el manejo de roles para finalmente realizar una retroalimentación con todo el equipo se trabajo



Figura 3.4. Diagrama de actividad general para caso de estudio

En la figura 3.5 se presenta las actividades necesarias para la ejecución del caso de estudio que inicia con el reconocimiento general de la población de estudiantes y finaliza con la presentación de los roles de trabajo por parte de los estudiantes.



Figura 3.5. Diagrama de actividades detallado para caso de estudio

A continuación se realiza una descripción de cada una de las actividades presentadas en la figura 3.5:

Definir población. El docente debe tener en cuenta elementos como: número de estudiantes inscritos al curso, lugar donde se realizarán las sesiones de clase, estadística de los estudiantes que han matriculado el curso por primera vez y aquellos que la están repitiendo.

Revisar unidades de competencia. El docente debe llevar un estricto control de la unidad de competencia a tratar durante el tiempo planeado, para ello es necesario tener en cuenta la información establecida en la tabla 2 relacionada con el formato de configuración inicial para el caso de estudio.

Revisar elemento de competencia. El docente debe tener muy claro los elementos de competencia a profundizar con el grupo de estudiantes. Favor tener como base los elementos de competencia presentes en la tabla.

Revisar saberes esenciales. El docente debe tener la suficiente experiencia y claridad tanto para las unidades como elementos de competencia a profundizar.

Revisar contenidos abordados. El docente debe llevar un estricto control de las actividades ya realizadas con el propósito de avanzar significativamente con el grupo de estudiantes en la adquisición y profundización de saberes. Es recomendable tener un control tanto de las unidades como de los elementos de competencia sugeridos en esta investigación.

Diseño de actividades de aprendizaje colaborativo. El docente apoyado de la metodología propuesta en este estudio debe diseñar sus sesiones de clase teniendo en cuenta el modelo y herramientas de aprendizaje sugeridas. Para ello revisar el final de la presente unidad donde se encuentra la planeación completa de una unidad de competencia.

Diseño de actividades de evaluación colaborativo. El diseño de las actividades de evaluación es una actividad eminentemente libre por parte del docente, ya que la intención de la metodología planteada es presentar una propuesta que permita apoyar la enseñanza de los fundamentos de programación en la que el docente asume el importante rol de acompañamiento en el proceso de aprendizaje, pero es el estudiante el dinamizador del conocimiento obtenido personal y grupalmente.

Especificar reglas de actividad. Es necesario que el docente establezca de forma clara una serie de reglas de actividad en la realización del proceso metodológico, de lo contrario se puede llegar a tener niveles preocupantes de ruido que llevaría a la promulgación de desorden general, perdiendo así el objetivo planteado.

En la tabla 3.11 se presentan las reglas de actividad por unidad y elemento de competencia para cada sub rol del rol estudiante.

Tabla 3.11. Reglas de actividad por unidad y elemento de competencia

Unidad de competencia	Elemento de competencia	Sub Rol 1	Sub Rol 2	Secuencia
Captura y salida de datos	Captura y salida de datos	Analista de entradas	Analista de salidas	Primero sub Rol1 y luego sub Rol 2 Simultaneo
		Analista de entradas/salidas	Analista de GUI	Primero sub Rol1 y luego sub Rol 2 Simultaneo
Estructura condicional	Condicional compuesto	Analista de entradas/salidas/ GUI	Diseñador	Primero sub Rol1 y luego sub Rol 2 Simultaneo
	Condicional simple			Primero sub Rol1 y luego sub Rol 2 Simultaneo
	Condicional anidado			Primero sub Rol1 y luego sub Rol 2 Simultaneo
	Condicional y operadores lógicos			Primero sub Rol1 y luego sub Rol 2 Simultaneo
Estructuras iterativas	Ciclo Para	Analista de entradas/salidas/ GUI	Diseñador	Primero sub Rol1 y luego sub Rol 2 Simultaneo
	Ciclo Mientras			Primero sub Rol1 y luego sub Rol 2 Simultaneo
	Ciclo Hacer Mientras			Primero sub Rol1 y luego sub Rol 2 Simultaneo

Inicialmente se conformaran grupos de dos estudiantes quienes inicialmente realizarán una propuesta personal de cada ejercicio planteado y luego abandonarán su grupo base para reunirse paulatinamente con “pares” de otros grupos hasta finalmente obtener una solución general. Luego cada integrante regresará a su grupo base con el ejercicio terminado para su rol y se llevará a cabo nuevamente el mismo ejercicio para el otro rol. Lo anterior concluye en dar solución total al ejercicio planteado con la participación de todo el grupo de estudiantes. En la figura 3.6 se presenta el diagrama de actividad general para el sub rol de analista del rol de estudiante.

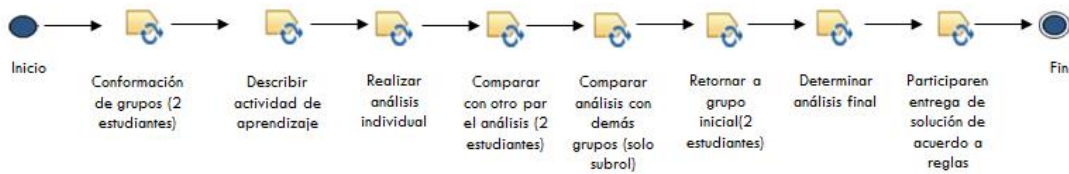


Figura 3.6. Diagrama de actividad general sub rol analista

En la figura 3.7 se presenta el diagrama de actividad general para el sub rol de diseñador del rol de estudiante.

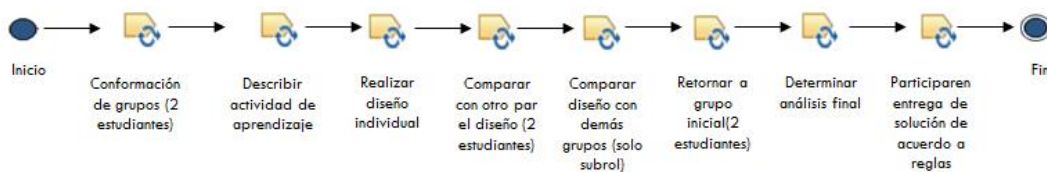


Figura 3.7. Diagrama de actividad general sub rol diseñador

Cuando el grupo de estudiantes ya tiene una experticia adquirida de la metodología, al crear los primeros grupos de dos estudiantes se puede hacer la realización del ejemplo en simultaneo con la participación de los dos roles en el que cada uno realiza su primer borrador y luego abandonan el grupo para unirse a otros “pares” hasta regresar nuevamente al grupo base y compartir la experiencia (ver figura 3.8).



Figura 3.8. Diagrama de actividad sub rol Analista y Diseñador

Es recomendable que para la conformación de grupos de trabajo se adopten las siguientes normas:

- ✓ Los estudiantes conformen de libre voluntad los grupos
- ✓ Se adopten diversos mecanismos para conformación de grupos como: azar, intencionalidad, nivel de conocimiento, edad, y por género.
- ✓ Exista rotación de roles entre los integrantes de un grupo

La retroalimentación se debe realizarse de acuerdo con la secuencia de trabajo con los sub roles, es decir puede ser parcial, inmediatamente se termine la actividad por cada sub rol, en la cual el docente debe guiar al grupo en la consolidación de la respuesta al ejercicio planteado antes de realizar el siguiente paso con el otro sub rol y así finalmente generalizar junto con los resultados obtenido por el grupo en general los hallazgos encontrados frente a una situación planteada (ver figura 3.9).



Figura 3.9. Diagrama de actividad retroalimentación parcial

De igual manera la retroalimentación puede ser después de que los dos sub roles terminen la secuencia total de las actividades, es decir con ejecución en simultáneo (ver figura 3.10).

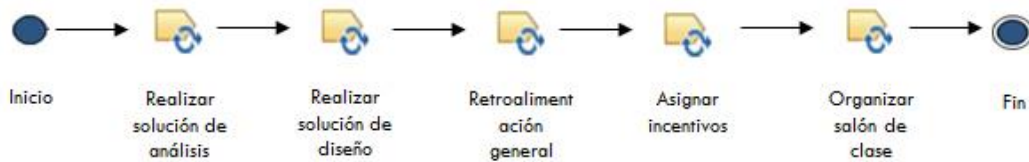


Figura 3.10. Diagrama de actividad para retroalimentación caso simultáneo

Presentar roles

Es necesario que el docente informe al grupo de estudiantes de la adopción de roles por cada uno de ellos (sub roles en el contexto de la presente investigación). En la tabla 3.12 se presenta las actividades que debe desarrollar cada sub rol bajo el rol de estudiante.

Tabla 3.12. Descripción de los sub roles para el rol de estudiante

Nombre de sub rol	Descripción
Analista de entradas	Encargado del análisis y construcción de las entradas y que deben representarse en un Diagrama de Entrada/salida
Analista de salidas	Encargado del análisis y construcción de las salidas y que deben representarse en un Diagrama de Entrada/salida
Analista de entradas/salidas	Encargado del análisis y construcción de las entradas y salidas que deben representarse mediante un Diagrama de Entrada/salida
Analista de GUI	Encargado del análisis y construcción de la Interfaz Gráfica de Usuario (GUI) el cual debe diagramarse con las normas sugeridas en el ítem 3.2
Analista de entradas/salidas/GUI	Encargado del análisis y construcción de las entradas, salidas y GUI. Dichas actividades forman parte de la fase de análisis que presenta esta investigación.
Diseñador	Encargado de leer y comprender la fase de análisis y convertirla en diseño utilizando herramientas de diseño algorítmico como DFD y PselInt, o hasta llegar a plantear un diagrama de clases

3.3 Propuesta de modelado gráfico

El modelado gráfico propuesto parte del objeto de análisis más elemental a la hora de concebir un proyecto de software que es el “diagrama de entrada salida”, con el cual es posible

identificar elementos claves como estructuras condicionales, estructuras repetitivas, clases, variables generales, variables de clase, métodos y otros elementos necesarios en los fundamentos de programación. A través del modelado gráfico se pretende que el estudiante mediante una abstracción visual modele su pensamiento computacional de una manera sencilla pero eficiente, el cual se verá reforzado por la metodología colaborativa mencionada que permitirá al estudiante estar en un ambiente con altos niveles de interacción entre “pares” que le permitirá conocer y afianzar las temáticas abordadas.

3.3.1 Elementos de modelado gráfico

La metodología planteada para llevar a cabo la enseñanza de los fundamentos de programación, se basa en dar a conocer al estudiante de primer curso de fundamentos de programación, la realización sistemática de una serie de pasos necesarios para la obtención de un producto de software que tenga algunos estándares metodológicos de construcción, por ello, dicha metodología inculca al estudiante la claridad que debe tener, independiente de la metodología de software que en un futuro seleccionarán para llevar a efecto proyectos de software, en la realización de unas fases definidas que inician desde la recolección de información y terminan en el planteamiento de un modelo de construcción que más tarde se convertirá en código fuente.

3.3.2 Fases para propuesta de modelado gráfico

La presente metodología concibe algunas fases para la enseñanza de los fundamentos de programación, haciendo énfasis en que el estudiante debe identificarlas claramente ya que son transversales a muchas metodologías de desarrollo de software existentes y cuyo propósito es el de familiarizarlo con los procesos desarrollados en los futuros cursos de Ingeniería de Software que deberá cursar. Entre las fases iniciales propuestas para un primer curso de fundamentación para programación se recomienda: Fase de recolección de requerimientos, fase de análisis, fase de diseño y fase de codificación (que inicialmente no será parte del presente estudio pero es necesario que el estudiante la conozca).

3.3.2.1 Fase de recolección de requerimientos

El docente debe informar al estudiante la importancia de recolectar información como una etapa importante en la construcción de proyectos de software. Es necesario que el docente concientice al estudiante que si se lleva a cabo un buen proceso de recolección de información, la probabilidad de éxito del proyecto será alta.

Debido a que la metodología presente está pensada en un primer curso de fundamentos de programación, el docente debe informar al estudiante que la fase de recolección de requerimientos (formato 9) será conducida directamente por parte del docente, quien a través del enunciado de los diversos ejemplos y ejercicios presentará una descripción detallada de todos los elementos necesarios para su estudio.

Es necesario que el docente explique que a través del enunciado del problema es posible tener un listado de todos los requerimientos que va a necesitar el estudiante para dar solución a una situación planteada y teniendo como base algunos elementos claves de la Ingeniería de Software que el estudiante profundizará más adelante, mediante la tabla 3.13 se hará un

proceso inicial cuyo fin es informarle al estudiante que antes de llevar a cabo un proyecto de software es necesario recolectar adecuadamente su información mediante una notación formal.

Tabla 3.13. Formato de recolección de requerimiento principal

Nombre del sistema	Requerimiento principal
Se asigna un nombre al enunciado dado por el docente	Se describe brevemente el objetivo del ejemplo

La intención del formato de requerimiento principal de la tabla 3.17 es que el estudiante vuelva a leer el enunciado del ejercicio para que extraiga la idea principal de este como un primer paso valioso.

La definición de esta fase para el estudiante puede ser: La fase de requerimientos consiste en la recolección de información necesaria para el desarrollo de un software.

3.3.2.2 Fase de análisis

Es necesario que el docente le informe al estudiante que después de apropiarse y realizar la fase de recolección de requerimientos (que para efectos de la presente propuesta metodológica se tomará a través del enunciado del problema) es necesario un proceso de análisis de la información, cuyo objetivo es determinar mediante la simulación de un entorno Cliente – Constructor de software donde el docente asumirá el rol de cliente y el estudiante de constructor de software, la claridad que debe tener el constructor de software frente a las necesidades de un cliente para que el proyecto de software colme las expectativas que tiene su cliente, es decir, mediante esta fase el estudiante debe tener la certeza de que lo que piensa frente a la solución de la situación planteada (como constructor de software) es lo que realmente el usuario necesita.

La definición de esta fase para el estudiante puede ser: La fase de análisis utiliza algunos métodos para analizar la información recolectada en la fase de requerimientos.

Esta fase estará compuesta por dos actividades: construcción del diagrama Entrada/Salida y la elaboración de la Interfaz gráfica de usuario.

Nota. Se sugiere dar a conocer a los estudiantes los conceptos de D E/S y GUI como se describen a continuación.

Diagrama entrada/salida (D E/S)

Permite el análisis de un sistema como si fuese una “caja negra” donde solo se tienen en cuenta las entradas y salidas de un sistema. El estudiante en este diagrama solo debe preocuparse por identificar entradas y salidas sin tener en cuenta el funcionamiento interno del sistema.

En la figura 3.11 se muestra la representación gráfica de un D E/S es:

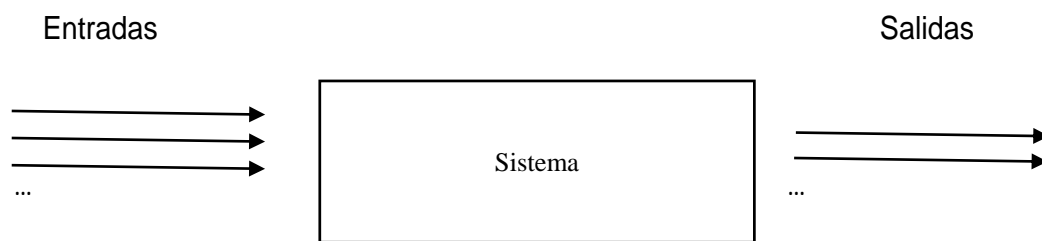


Figura 3.11. Diagrama de Entrada/Salida

El D E/S es el fundamento de la presente investigación, en la cual se concibe que si el estudiante identifica claramente las entradas y salidas de un sistema que será convertido a una solución computacional, entonces la probabilidad de éxito en su construcción será alta.

Para ilustrar al estudiante sobre el análisis de D E/S se recomienda que el primer ejemplo se realice con situaciones vivenciales y cada uno de ellos se asocie al concepto de “sistema” como lo son el uso de: cajero automático, calculadora, lavadora, horno microondas, entre otros, y verbalmente analizar sus posibles entradas y salidas cuyo análisis se presenta en la tabla 3.14.

Tabla 3.14. Ejemplos Entradas/salidas para situaciones vivenciales

Sistema	Entradas	Salidas
Cajero automático	Tarjeta, clave, valor, opciones, etc.	Dinero, recibo, audios, mensajes, etc.
Calculadora	Dígitos, operaciones	Resultados
Lavadora	Ropa sucia, agua, detergente, etc.	Ropa limpia, audios,
Horno Microondas	Comida, opciones	Comida caliente, audios, luz

Luego se debe realizar al menos uno de ellos mediante el D E/S informándole al estudiante de la importancia de hacer el proceso con las etapas sugeridas en esta investigación. A continuación se presentan algunos ejemplos del D E/S.

Ejemplo. Realice un D E/S para el sistema Cajero Automático (ver solución en figura 3.12):

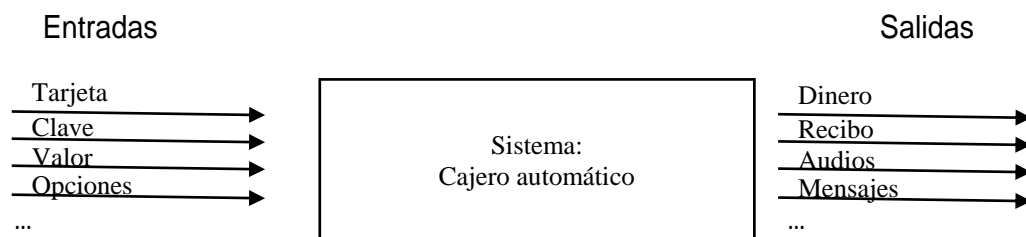


Figura 3.12. D E/S cajero automático

Ejemplo. Realice un D E/S para el sistema calculadora básica (ver solución en figura 3.13):



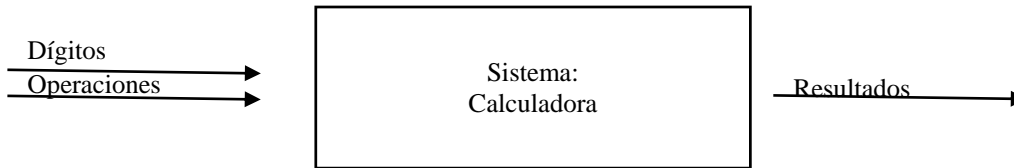


Figura 3.13. D E/S Calculadora básica

Una vez realizado el ejemplo 1 y 2, el docente debe invitar a que cada estudiante plantee un D E/S y al cabo de un lapso de tiempo indagaría algunos de ellos.

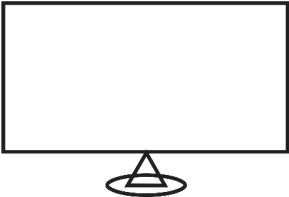


Interfaz Gráfica de Usuario (GUI)

En esta fase se concibe un primer boceto general de cómo se puede visualizar el diseño del software mediante una representación simbólica de ubicación de una serie de elementos utilizando el dispositivo de salida de un computador, es decir la pantalla.

La GUI permite representar los elementos analizados en un D E/S. Además, la GUI utiliza una serie de símbolos que gráficamente sirve de guía tanto al programador para tener certeza del software que va a construir como al usuario de que su necesidad de software fue comprendida por el programador.

Nota al docente. Teniendo en cuenta las herramientas de desarrollo de software existentes se plantea una GUI acorde a los estándares de construcción de software gráfico. En la tabla 3.15 se presentan los elementos para una GUI:

Tabla 3.15. Elementos de GUI

Símbolo	Descripción
	Pantalla: corresponde al dispositivo donde se realizará la impresión de los resultados tanto de entradas y salidas en una solución de software.
	Entrada: Permite el ingreso de datos al sistema
	Salida: Permite la impresión de datos a través de la pantalla

Nota para docente. El objetivo de utilizar como símbolo de entrada al rectángulo de línea continua tiene que ver con elementos reales de construcción de GUI incorporados en los lenguajes de programación actuales haciendo una similitud con los campos de texto que permiten la captura de datos por teclado (como por ejemplo los JTextField de Java o los TextBox de .Net). Así mismo, el símbolo de salida utilizado corresponde a un rectángulo de

línea punteada que cumple las veces de un Label (por ejemplo un JLabel en Java o un Label en .Net).

Una vez revisado los conceptos de D E/S y GUI el docente debe recalcar al estudiante que la fase de requerimientos se realizará en el enunciado del problema y la fase de análisis comprenderá los dos conceptos anteriormente estudiados.

3.3.2.3 Fase de Diseño

En esta fase el docente le informará al estudiante quien ya tiene claro la fase de recolección de requerimientos y la fase de análisis, que es importante construir un modelo que cumpla con los requerimientos exigidos y con los elementos realizados en la fase de análisis.

De esta manera en la fase de diseño se construirá un modelo basado en algoritmos que tome como base la información existente en la fase de análisis y que presente una solución que puede ser tratada como un proceso algorítmico y que puede ser llevada a una representación más abstracta basada en el modelado Orientado a Objetos.

La definición de esta fase para el estudiante puede ser: Toda solución de software requiere de una etapa de modelamiento que se conoce con el nombre de fase de diseño, para ello es indispensable la fase de análisis y el modelado algorítmico cuenta con varias metodologías entre ellas: Pseudocódigo, diagramas Nassi-Shneiderman (NSD), Diagrama de Flujo (DF), entre otros.

Bailey y Lundgaard proponen diversas estructuras formales que permiten separar la lógica de programación de un lenguaje de programación en particular, entre estas estructuras están: diagramas de flujo, pseudocódigo, diagramas de entrada y salida, gráficos de jerárquicos, entre otros, permiten enriquecer la explicación de diferentes técnicas para la resolución de problemas computacionales [96].

Inicialmente la metodología planteada y en su carácter gráfico abordará el diseño de algoritmos utilizando DF, para ello el docente debe brindar al estudiante la concepción teórica. A continuación se propone las definiciones para algoritmo y diagrama de flujo.

Algoritmo. Es un conjunto de instrucciones ordenadas de forma lógica para resolver una situación. Un algoritmo se caracteriza por ser claro, preciso y finito.





Diagrama de flujo (DF). Es la representación gráfica de un algoritmo. El anexo 7 contiene los símbolos utilizados por un DF.

Diseño de algoritmos

Esta metodología concibe la relación existente entre: Diagrama Entrada/Salida (D E/S) y la Interfaz Gráfica de Usuario (GUI) y por otra parte la GUI con Diagrama de Flujo (DF), por ello el docente debe tomar la GUI construida en la fase de análisis para convertirla en algoritmo a

través del DF en la fase de diseño. Para ello es necesario tener en cuenta la simbología de la GUI y su relación con los elementos del DF como lo muestra la tabla 3.16.

Tabla 3.16. Equivalencias entre GUI y DF

Elemento	Equivalente en GUI	Equivalente en DF
Salida		
Entrada		

3.4 Unidades de competencia con propuesta de modelado gráfico

De acuerdo a la planeación realizada en el formato de configuración inicial para fundamentación conceptual de modelado gráfico presentado en la tabla 3.4 se abordan las siguientes unidades y sus correspondientes elementos de competencia:

3.4.1 Capturas y salida de datos

En esta unidad de competencia el docente dará a conocer los elementos básicos que involucren capturas (entradas) e impresiones (salidas). A continuación el docente presentará algunos ejemplos para desarrollarlos por las tres fases: Recolección de requerimientos, análisis y diseño.

Ejemplo. Proponer una solución computacional que capture un número e imprima su cuadrado.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.17)

Tabla 3.17. Recolección de requerimientos para cuadrado de un número.

Nombre del sistema	Requerimiento principal
Cálculo cuadrado	Capturar un número y elevarlo al cuadrado

Solución

2. Fase de análisis

a. Diagrama Entrada/Salida (D E/S)

En la figura 3.14 se presenta el D E/S aclarando que existe una entrada y una salida para el sistema:

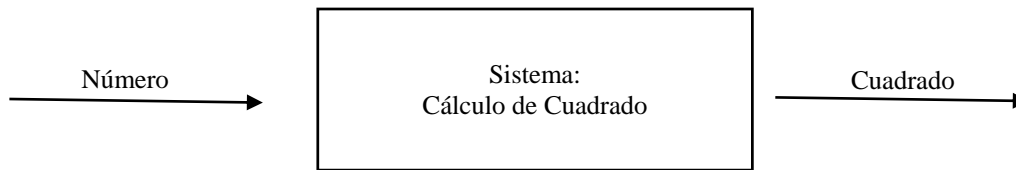


Figura 3.14. D E/S cuadrado de un número

b. Interfaz Gráfica de Usuario (GUI)

Para efectos didácticos se dividirá en sub pasos su construcción:

Primer paso: como existe una entrada y una salida en el D E/S, la primera versión de la GUI se muestra en la figura 3.15 donde la entrada se simboliza con un rectángulo de línea continua y la salida con un rectángulo de línea punteada.

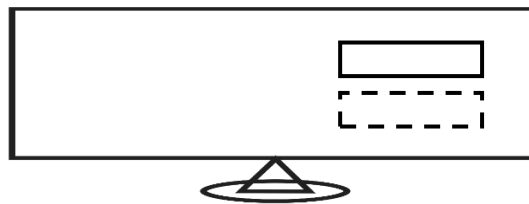


Figura 3.15. GUI inicial

La figura 3.16 es el resultado de unir el D E/S y la GUI, las líneas azules solo indican cómo la entrada y la salida de D E/S se representa en la GUI.

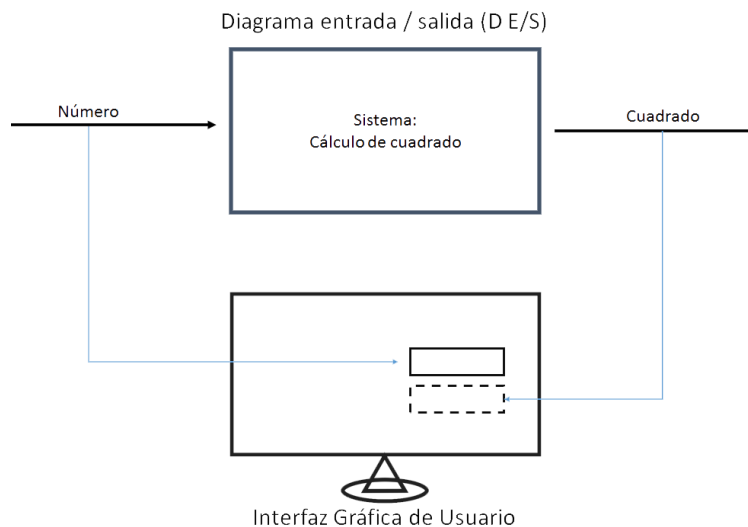


Figura 3.16. D E/S Vs GUI

Segundo paso: La GUI presentada en la figura 3.16 a pesar que cumple estrictamente con el D E/S planteados, no representa una información completa e intuitiva para que el usuario utilice la herramienta de software, por ello es necesario completarla con algunos rótulos de orientación. Pare ello, se propone colocar básicamente dos elementos.

- a. Un título a la aplicación. El cual será el nombre del sistema que aparece en el D E/S que para este ejercicio será "Cálculo de cuadrado".

- b. Un rótulo explicativo para cada entrada y salida existentes en el D E/S. en el caso de la entrada de este ejemplo su rótulo explicativo debería decir “Digite un número” y para la salida sería “Su cuadrado es”.

Por lo tanto la GUI final se presenta en la figura 3.17 y por recomendación luego de colocar las entradas y salidas correspondientes se debe colocar el título del sistema y luego colocando los rótulos explicativos a cada entrada y salida de arriba abajo.

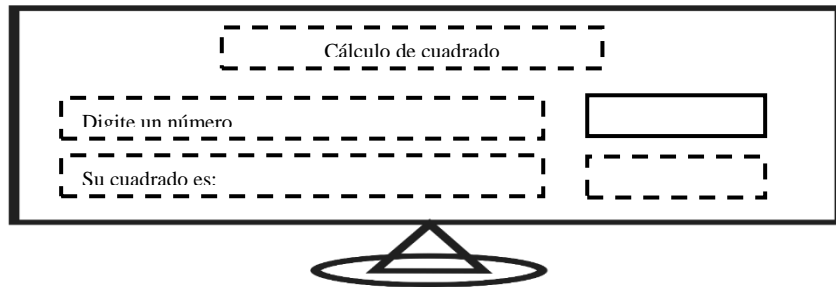


Figura 3.17. GUI cálculo de cuadrado

3. Fase de diseño

A Continuación se explicará las instrucciones que se deben tener en cuenta para convertir la GUI del ejemplo 3 y convertirla en algoritmo:

1. Es necesario informarle al estudiante que ya hemos pasado por las fases de requerimientos y análisis (mediante D E/S y DF) y ahora se realizará la fase de diseño.
2. Una recomendación importante es generar buenas prácticas de fundamentación de programación para el estudiante, por ello es necesario inculcarle que todo proyecto de software debe comenzar por un título alusivo al contenido del software a desarrollar. Por ello, la primera instrucción del algoritmo debe ser el título que de acuerdo a la GUI de la figura 3.17 es “Cálculo de cuadrado”.
3. Es necesario informarle al estudiante que el algoritmo se debe construir con base a la GUI y recalcarle la importancia de realizar ordenadamente los pasos y una forma sencilla de comentárselo es teniendo en cuenta cómo debería ir apareciendo la información en la pantalla al usuario como si el proceso se realizara en cámara lenta, presentándole uno a uno los elementos que ya hemos diseñado en la figura 3.17, para ello se sugiere el siguiente orden:
 - a. Al usuario en primer lugar debería presentar el título del software.
 - b. Luego el rótulo explicativo de la entrada.
 - c. A continuación se debería capturar la entrada.
 - d. Luego el rótulo explicativo de la salida.
 - e. Finalmente imprimir la salida.

En la figura 3.18 se presenta el orden que deben tener las instrucciones realizadas en la GUI y su correspondiente algoritmo se muestra en la figura 3.19.

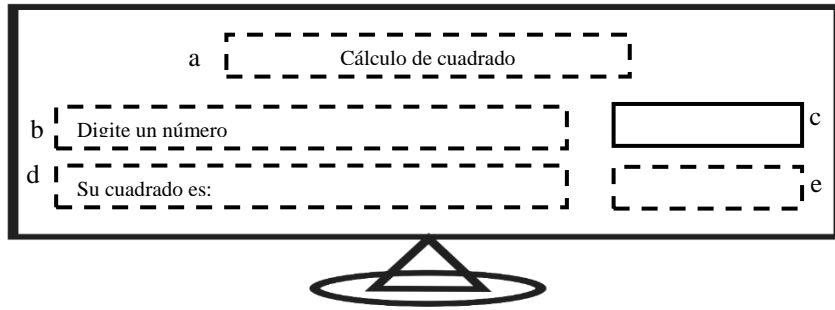


Figura 3.18. Orden de presentación de los elementos de una GUI a un usuario

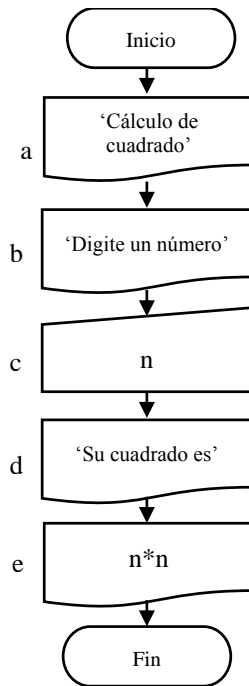


Figura 3.19. DF cuadrado de un número

Es recomendable informarle al estudiante que cuando existen dos salidas juntas donde una contiene un rótulo y la otra un valor procedente o no de variables, es una buena práctica de notación unir las dos salidas mediante una coma (ver figura 3.20).

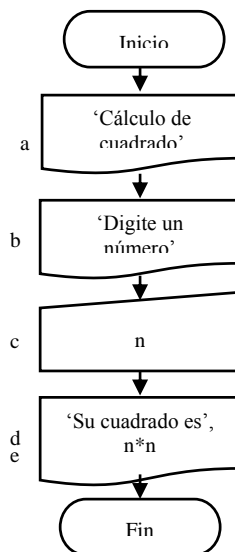


Figura 3.20. DF sugerido de cuadrado de un número

En el anexo 14 se presentan ejemplos complementarios para el curso planteado.

1.4.2 Condicionales

Es importante seguir recalcando la necesidad de continuar realizando las tres fases para abordar el tema de condicionales.

La definición de esta fase para el estudiante puede ser: Los condicionales son estructuras que permiten evaluar una o más proposiciones de forma excluyente y tienen como máximo dos alternativas de solución: verdadero o falso. Los condicionales pueden ser: simples, compuestos, anidados y condicionales con operadores lógicos.

1.4.2.1 Condicional Compuesto

Es aquel que ante la evaluación de sus proposiciones realiza instrucciones tanto por verdadero como por falso.

Nota al docente. En la concepción de esta temática el docente debe motivar al estudiante con ejemplos vivenciales de lo que es un condicional.

A continuación se contemplan algunos ejemplos de condicionales compuestos con sus tres fases:

Ejemplo. Diseñar una solución computacional que capture la nota definitiva para un estudiante de Introducción a la Ingeniería y determine si aprueba o reprueba el curso.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.18)

Tabla 3.18. Recolección de requerimientos Concepto cualitativo

Nombre del sistema	Requerimiento principal
Concepto cualitativo de nota	Capturar la nota definitiva de un estudiante y determinar si aprueba o reprueba el curso

2. Fase de Análisis

a. D E/S

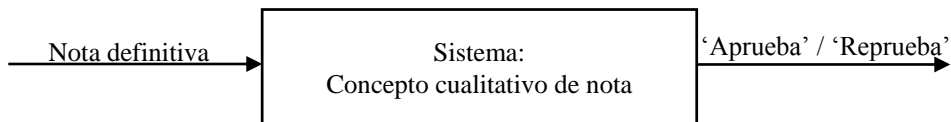


Figura 3.21. D E/S Concepto cualitativo

Nota al docente. Es importante indicarle al estudiante que el D E/S es clave ya que ante una salida con dos alternativas de solución se debe pensar en la utilización de un condicional compuesto.

b. GUI



Figura 3.22. GUI concepto cualitativo de nota

En la figura 3.23 se muestra la relación existente entre la salida del D E/S y su relación con un condicional.

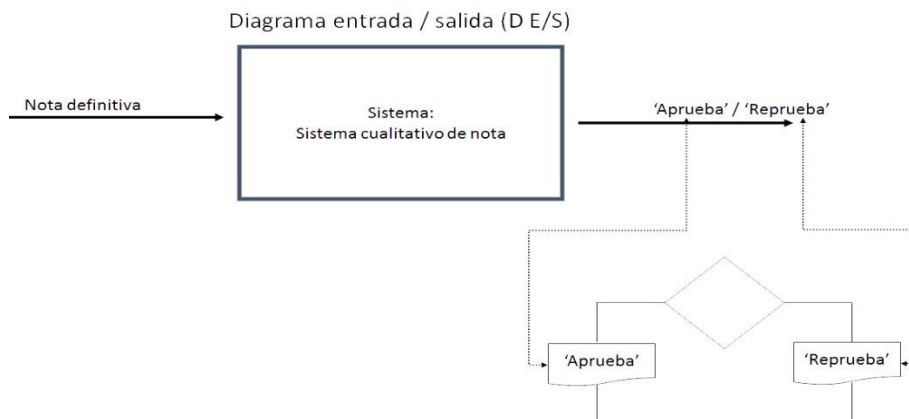


Figura 3.23. Relación entre D D/S y DF para condicional compuesto

3. Fase de diseño

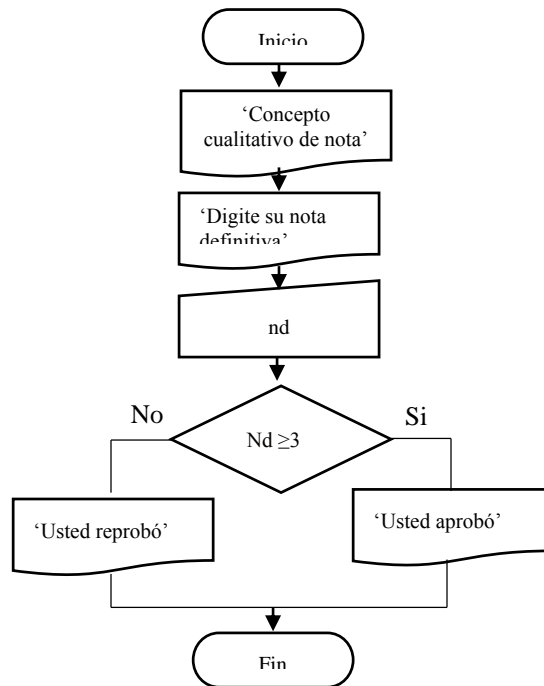


Figura 3.24. DF concepto cálculo cualitativo nota

En el anexo 14 se presentan ejemplos complementarios para el curso planteado.

3.4.2.2 Condicional Simple

La definición para el estudiante puede ser: Es aquel que ante la evaluación de sus proposiciones solo realiza instrucciones por verdadero.

Ejemplo. Diseñar una solución computacional que capture para un artículo de papelería su valor de venta, si dicho valor es mayor a \$20.000 obtendrá un descuento del 10%. Calcular el descuento realizado.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.19)

Tabla 3.19. Recolección de requerimientos descuento de artículo.

Nombre del sistema	Requerimiento principal
Descuento de artículo	Capturar valor de venta de un artículo y calcular el descuento realizado.

2. Fase de Análisis

a. D E/S

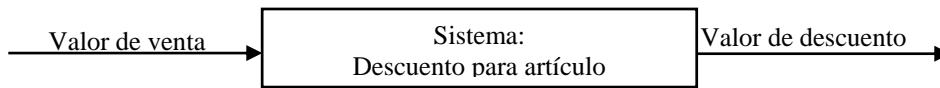


Figura 3.25. D E/S descuento para artículo

b. GUI

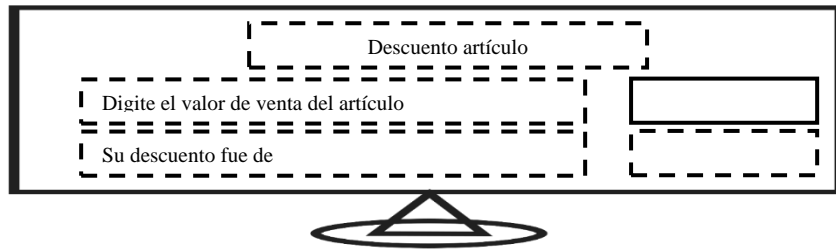


Figura 3.26. GUI descuento para artículo

3. Fase de diseño

Se recomienda realizar una breve explicación de lo que es un proceso y una asignación para realizar este ejemplo:

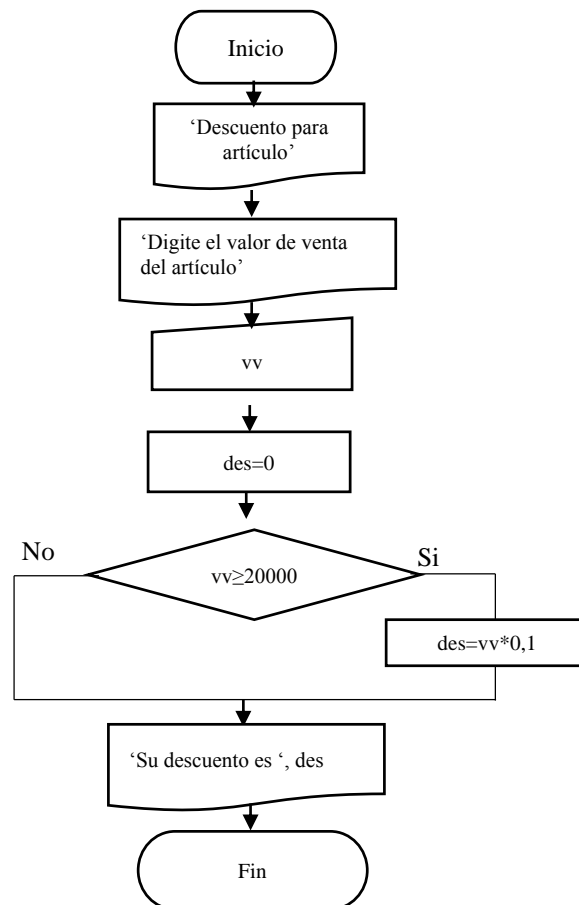


Figura 3.27. DF descuento para artículo

En el anexo 14 se presentan ejemplos complementarios para el curso planteado.

3.4.2.3 Condicional anidado

La definición para el estudiante puede ser: son estructuras condicionales que se encuentran dentro de otra estructura del mismo tipo

Ejemplo. Diseñar una solución computacional que capture un número y determine si es positivo, negativo o neutro.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.20)

Tabla 3.20. Recolección de requerimientos positivo, negativo o neutro.

Nombre del sistema	Requerimiento principal
Positivo, negativo o neutro	Capturar un número y determinar si es positivo, negativo o neutro

2. Fase de Análisis

a. D E/S

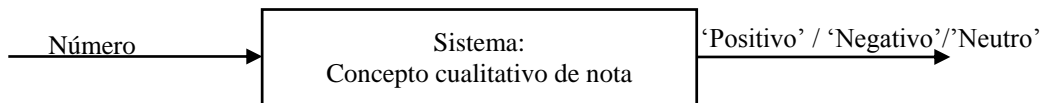


Figura 3.28. D E/S Concepto cualitativo de nota

Nota al docente. Es importante indicarle al estudiante que el D E/S es clave ya que ante una salida con más de dos alternativas de solución se debe pensar en la utilización de un condicional anidado.

b. GUI

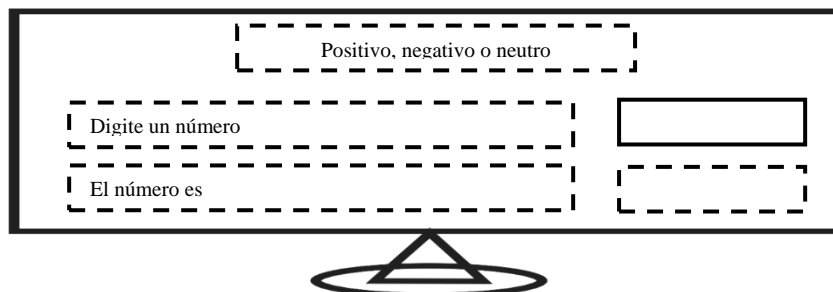


Figura 3.29.GUI positivo, negativo o neutro

En la figura 3.30 se muestra la relación existente en entre la salida del D E/S y su relación con un condicional anidado.

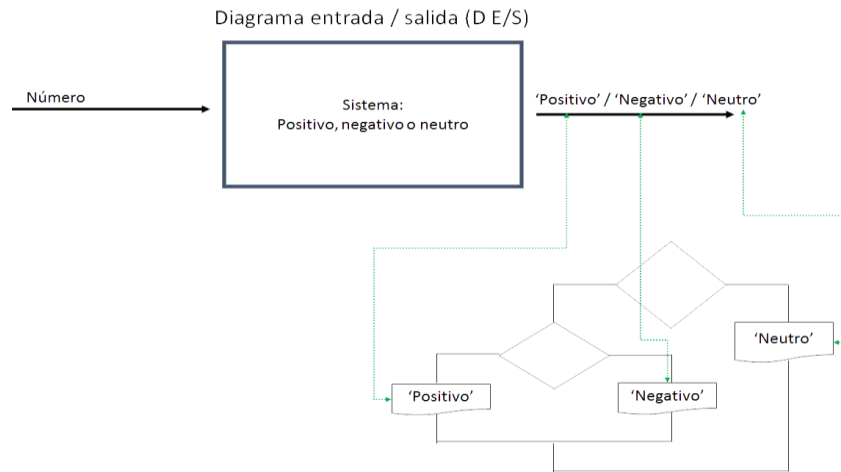


Figura 3.30. Relación entre D D/S y DF para condicional anidado

3. Fase de diseño

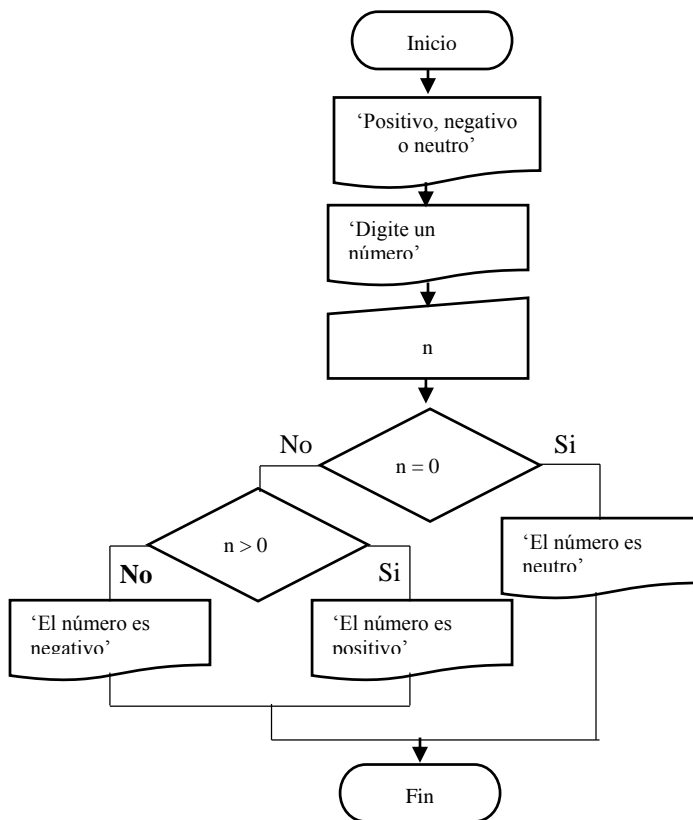


Figura 3.31. DF positivo, negativo o neutro

En el anexo 14 se presentan ejemplos complementarios para el curso planteado.

3.4.2.4 Condicionales y operadores lógicos

La definición para el estudiante puede ser: simplifican en algunos casos a los condicionales anidados y la evaluación de sus proposiciones depende del valor de las tablas de verdad.

Ejemplo. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(20 \leq n \leq 300)$.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.21)

Tabla 3.25. Recolección de requerimientos rango de números

Nombre del sistema	Requerimiento principal
Rango de números	Capturar un número y determinar si se encuentra o no en un rango

2. Fase de Análisis

a. D E/S

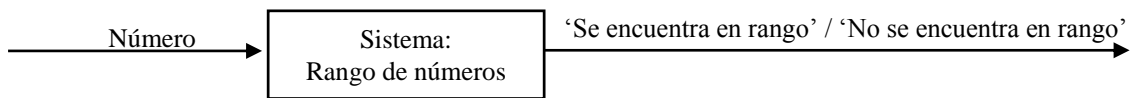


Figura 3.32. D E/S rango de números

Nota al docente. En este caso la salida por tener dos alternativas el condicional a utilizar debe ser compuesto como se aprecia en la figura 3.34.

b. GUI

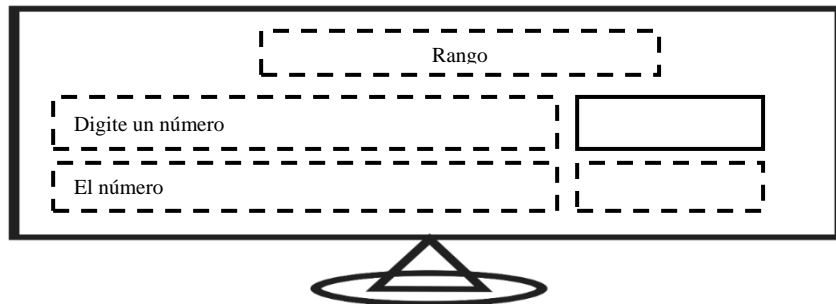


Figura 3.33 GUI rango de números

3. Fase de diseño

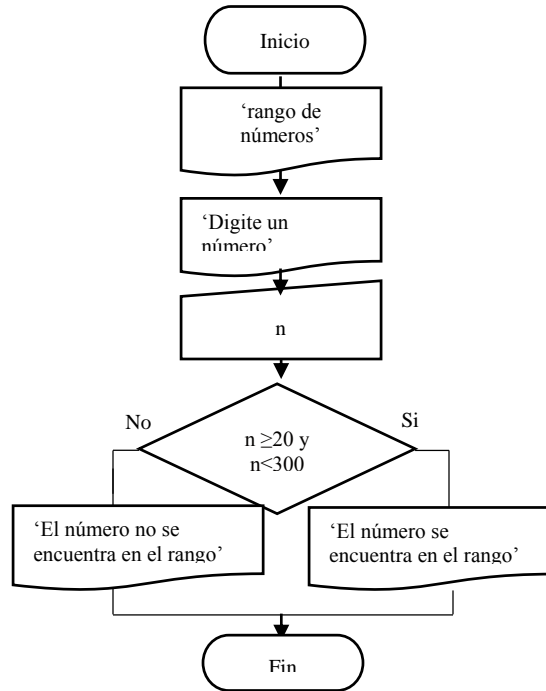


Figura 3.34. DF Rango de números

En el anexo 14 se presentan ejemplos complementarios para el curso planteado.

3.4.3 Ciclos

La definición para el estudiante puede ser: son estructuras que permiten repetir instrucciones una cierta cantidad de veces. Las estructuras cíclicas más utilizadas son: Ciclo Para, Ciclo Mientras y Ciclo Hacer Mientras.

3.4.3.1 Ciclo Para

La definición para el estudiante puede ser: Es una estructura iterativa que se caracteriza por contener tres parámetros: inicio, condición o fin y pasos.

A diferencia de muchos referentes que tratan estos ciclos, en esta investigación se propone dividir el estudio de los tres ciclos en ascendentes y descendente por la naturaleza de comportamiento de sus parámetros.

Ciclo Para ascendente

La definición para el estudiante puede ser: el ciclo Para ascendente se caracteriza por:

- El valor de inicio es mayor que el valor de fin.
- La condición o fin únicamente permite la utilización de los operadores relacionales menor o menor igual.
- Los pasos son positivos.

Ejemplo. Diseñar una solución computacional que imprima los siete primeros números en forma ascendente.

Solución

1. **Fase de Recolección de requerimientos** (ver solución tabla 3.22)

Tabla 3.22. Recolección de requerimientos números ascendente ciclo Para

Nombre del sistema	Requerimiento principal
Números ascendentes ciclo para	Imprimir los siete primeros números en forma ascendente

2. **Fase de Análisis**

a. **D E/S**

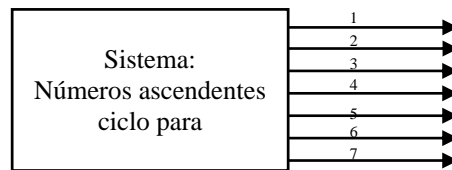


Figura 3.35. D E/S Números ascendentes ciclo Para

b. **GUI**

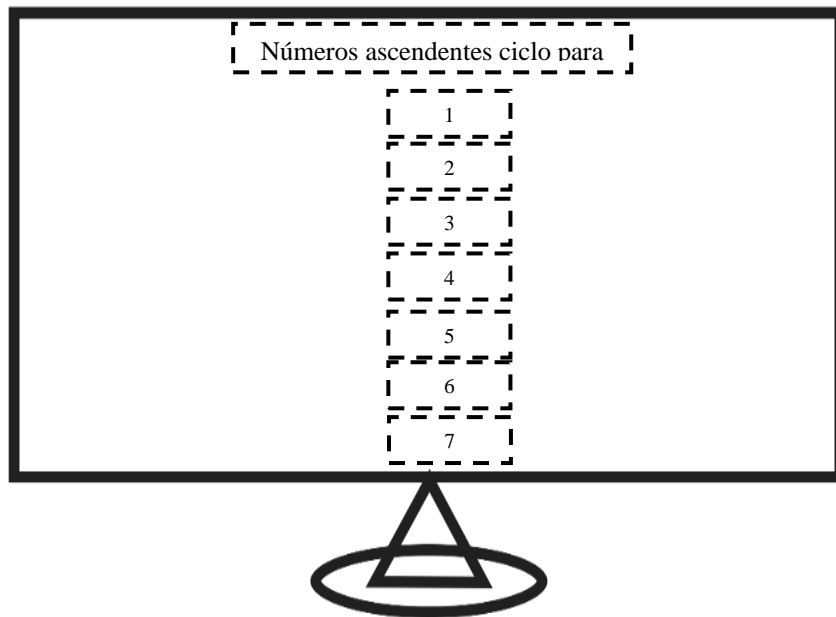
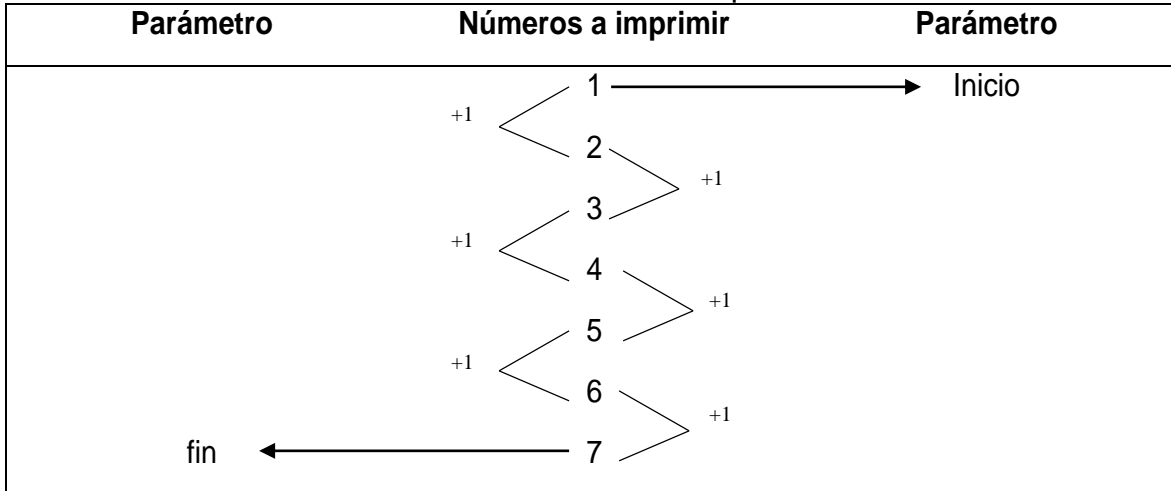


Figura 3.36.GUI Números ascendentes ciclo Para

Se recomienda que el docente presente el análisis sugerido en la tabla 3.27 para la configuración de los tres parámetros del Ciclo Para descendente:

Tabla 3.23. Análisis ciclo para ascendente



El parámetro de **inicio** debe comenzar en 1, el valor de **condición o fin** en 7 y los **pasos** son positivos, en este caso es necesario sumar 1 a cada número para obtener el siguiente número (es decir si al inicio 1 se le suma +1 el resultado es el segundo valor 2 y así sucesivamente) por lo cual y asociándolo a una variable de control, los parámetros deben ser: $i=1$, $i \leq 7$, 1. Estos parámetros se ajustan a las tres características mencionadas anteriormente del ciclo para ascendente.

3. Fase de diseño

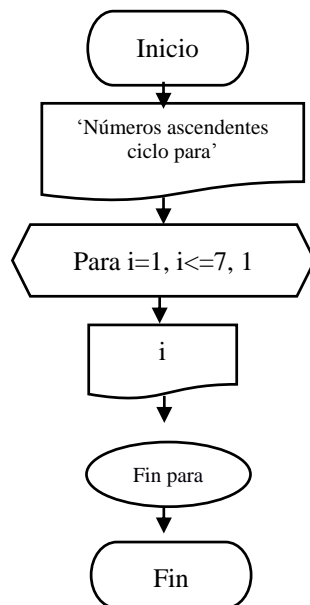


Figura 3.37. DF número ascendentes ciclo Para

En el anexo 14 se presentan ejemplos complementarios para el curso planteado.

Ciclo Para descendente

La definición para el estudiante puede ser: el ciclo Para descendente se caracteriza por:

- El valor de inicio es menor que el valor de fin.
- La condición o fin únicamente permite la utilización de los operadores relacionales mayor o mayor igual.
- Los pasos son negativos.

Ejemplo. Diseñar una solución computacional que imprima los siete primeros números en forma descendente.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.24)

Tabla 3.24. Recolección de requerimientos números descendente ciclo Para

Nombre del sistema	Requerimiento principal
Números descendentes ciclo para	Imprimir los siete primeros números en forma descendente

2. Fase de Análisis

a. D E/S

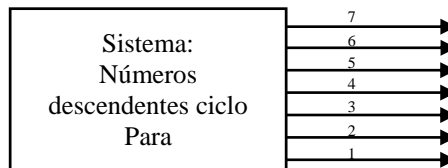


Figura 3.38. D E/S Números descendentes ciclo Para

b. GUI

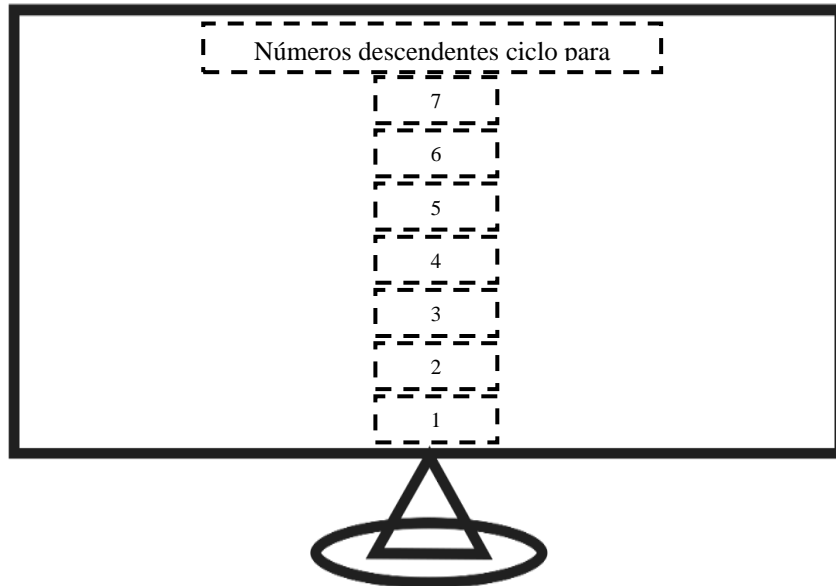


Figura 3.39. GUI Números descendentes ciclo Para

Se recomienda que el docente presente el análisis sugerido en la tabla 3.29 para la configuración de los tres parámetros del Ciclo Para descendente:

Tabla 3.25. Análisis ciclo Para descendente

Parámetro	Números a imprimir	Parámetro
	7	Inicio
-1	6	-1
-1	5	-1
-1	4	-1
-1	3	-1
-1	2	-1
fin	1	

El parámetro de **inicio** debe comenzar en 7, el valor de **condición o fin** en 1 y los **pasos** son negativos, en este caso es necesario sumar -1 a cada número para obtener el siguiente número (es decir si al inicio 7 se le suma -1 el resultado es el segundo valor 6 y así sucesivamente) por lo cual y asociándolo a una variable de control, los parámetros deben ser: $i=7$, $i \geq 1$, -1. Estos parámetros se ajustan a las tres características mencionadas anteriormente del ciclo para ascendente.

3. Fase de diseño

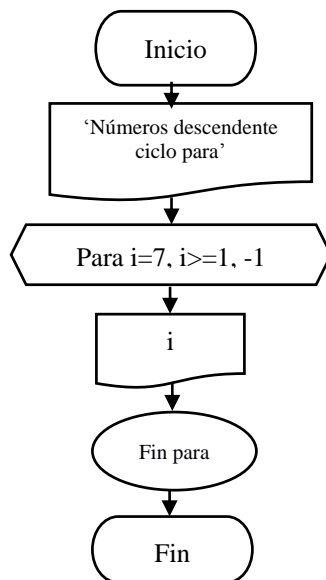


Figura 3.40. DF número descendentes ciclo Para

3.4.3.2 Ciclo Mientras

La definición para el estudiante puede ser: Es una estructura iterativa que evalúa una o más proposiciones repitiendo las instrucciones encontradas hasta que dicha evaluación deje de cumplirse. Se caracteriza por primero evaluar las proposiciones para ingresar a la estructura del ciclo.

Ejemplo. Diseñar una solución computacional que imprima los siete primeros números en forma ascendente.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.26)

Tabla 3.26. Recolección de requerimientos números ascendente ciclo Mientras

Nombre del sistema	Requerimiento principal
Números ascendentes ciclo Mientras	Imprimir los siete primeros números en forma ascendente

2. Fase de Análisis

a. D E/S

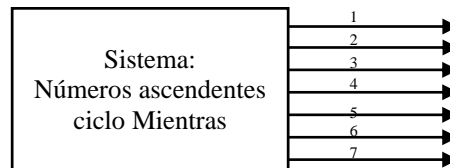


Figura 3.41. D E/S Números ascendentes ciclo Mientras

b. GUI

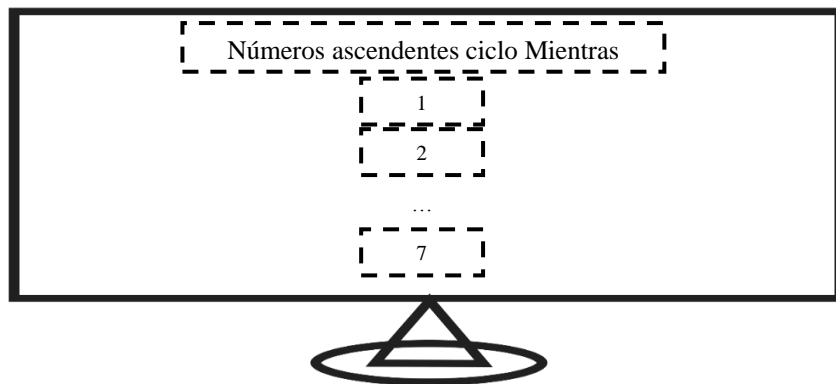


Figura 3.42. GUI Números ascendentes ciclo Mientras

Se recomienda que el docente presente el análisis sugerido en la tabla 3.23 para la configuración de los parámetros del Ciclo Mientras.

3. Fase de diseño

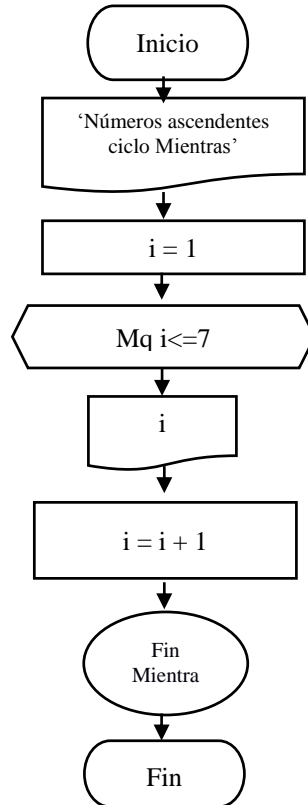


Figura 3.43. DF número ascendentes ciclo Mientras

3.4.3.3 Ciclo Hacer Mientras

La definición para el estudiante puede ser: Es una estructura iterativa que evalúa una o más proposiciones repitiendo las instrucciones encontradas hasta que dicha evaluación deje de cumplirse. Se caracteriza porque al menos realiza una vez las instrucciones contenidas en su estructura.

Ejemplo. Diseñar una solución computacional que imprima los siete primeros números en forma ascendente.

Solución

1. Fase de Recolección de requerimientos (ver solución tabla 3.27)

Tabla 3.27. Recolección de requerimientos números ascendente ciclo Hacer Mientras

Nombre del sistema	Requerimiento principal
Números ascendentes ciclo Hacer Mientras	Imprimir los siete primeros números en forma ascendente

2. Fase de Análisis

a. D E/S

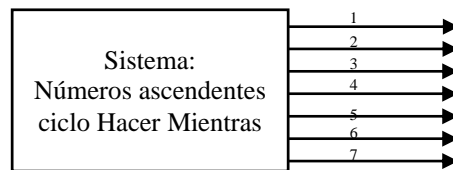


Figura 3.44. D E/S Números ascendentes ciclo Hacer Mientras

b. GUI

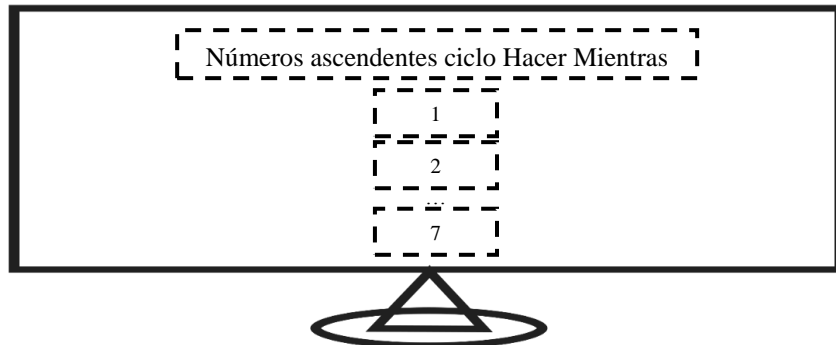


Figura 3.45. GUI Números ascendentes ciclo Hacer Mientras

3. Fase de diseño

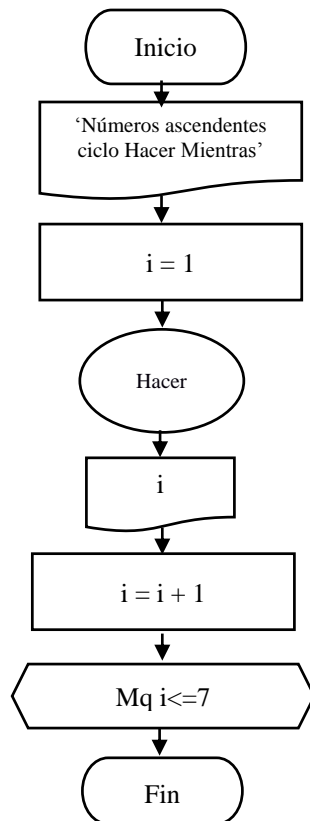


Figura 3.46. DF número ascendentes ciclo Hacer Mientras

CAPÍTULO 4: VALIDACIÓN PROPUESTA METODOLÓGICA

Antes de presentar los resultados obtenidos en el proceso investigativo, es conveniente mostrar a nivel de ejemplo cómo se llevaría a cabo una sesión de clase con la metodología propuesta. A continuación se muestra el desarrollo de una sesión de clase.

4.1 Sesión de clase

Se tomará como ejemplo la sesión de clase para estudiar los procesos básicos de “Captura y salida de datos”, para ello, el docente teniendo en cuenta el modelo colaborativo presentado en la figura 2.1 realizará los pasos indicados en el diagrama de actividad que se muestra en la figura 4.1:

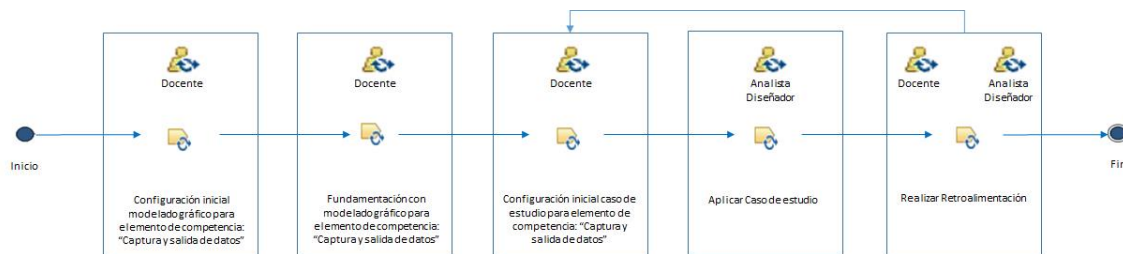


Figura 4.1. Diagrama de actividad clase “Captura y salida de datos”

En la figura 4.1 se muestra la realización de las siguientes actividades:

1. Configuración inicial de los fundamentos conceptuales con modelado gráfico para el tema “Captura y salida de datos”. En la cual el docente prepara previo a la sesión de clase los materiales y saberes necesarios para abordar este elemento de competencia. Para ello el docente debe guiarse en los elementos de configuración inicial para fundamentación conceptual de modelado gráfico presentado en el anexo 12 y del formato de planeación de ejemplos referenciado en el anexo 13, no sin antes aclarar que en el ejercicio completo del curso el docente ya ha revisado los elementos presentes en la tabla 3.9 de configuración inicial.
2. Fundamentación inicial al grupo de estudiantes sobre “capturas y salidas de datos” utilizando modelado gráfico. En esta actividad el docente debe exponer al grupo de estudiantes los saberes adecuados y presentados en el numeral 3.4.1 de capturas y salida de datos.
3. Configuración inicial para caso de estudio. El docente debe antes de la sesión de clase preparar los casos de estudio que les presentará a los estudiantes para que a través de un entorno colaborativo mediado por JIGSAW haya una apropiación significativa del elemento de competencia tratado. Para ello el docente se apoyará del formato de configuración inicial para el caso de estudio presentado en el anexo 10.
4. Presentación y realización de casos de estudio con el grupo utilizando JIGSAW. Para ello el docente ejecutará la planeación del formato de configuración inicial para el caso de estudio, organizando el salón de clase de tal manera que los estudiantes queden reunidos

en grupos de dos y distribuidos en todo el recinto, demás el docente será el encargado únicamente de presentar los enunciados que se encuentran en el formato de planeación de ejercicios de configuración inicial para el caso de estudio del anexo 10 de acuerdo al diagrama de actividad para el caso de estudio de la figura 3.12 y expuesto paso a paso en el diagrama de actividades detallado para el caso de estudio presentado en la figura 3.5. Cada ejemplo será analizado por los dos estudiantes cada uno en el rol de analista de estradas/salidas y analista de GUI como lo indican las reglas de actividad por unidad y elemento de competencia presentado en la tabla 3.15. Para ello una vez presentado el enunciado del ejercicio, cada estudiante propondrá una solución desde su rol y la compartirá con su compañero de grupo base, luego cada uno se reunirá con otro “par” de otro grupo para analizar cada propuesta y concluir una sola y luego esos dos se reunirán con otros “pares” hasta que la mitad del grupo quede reunido bajo un rol y la otra mitad con el otro rol los cuales obtendrán una solución al ejercicio, finalmente cada estudiante volverá a su grupo base para compartir la solución con su compañero y entre los dos proponer la solución final. Los diagramas de actividad para el sub rol analista y diseñador se presentan en la figura 3.6, 3.7 y 3.8.

5. Retroalimentación por parte del docente. Al terminar cada ejercicio el docente realizará una retroalimentación que consistirá en concluir junto con los estudiantes la solución final y las consideraciones importantes y pertinentes. el diagrama de actividad para que el docente refuerce y realice la retroalimentación se presenta en la figura 3.9 y 3.10. Luego de realizar la retroalimentación, el docente presentará el siguiente ejercicio y si es necesario intercambiará los roles entre los dos integrantes de cada grupo o los renovará totalmente.

Nota. Para abordar esta temática y conforme al formato presentado en el anexo 10 el estudiante ya conoce los siguientes elementos de competencia: Diagramas Entrada/Salida e interfaz gráfica de usuario y en este caso el ejercicio se realizará solo para la fase de análisis.

En la tabla 4.1 se describe roles, realización y duración de cada una de las actividades para realizar un sesión de clase.

Tabla 4.1. Acciones para una sesión de clase

Acción	Responsable (rol)	Realización	Duración
Realizar la configuración inicial de los fundamentos conceptuales con modelado gráfico para el tema “Captura y salida de datos”	Docente	Antes de la sesión de clase	A consideración del docente
Ejecutar la fundamentación inicial al grupo de estudiantes sobre “capturas y salidas de datos” utilizando modelado gráfico	Docente	Salón de clase	20 minutos
Realizar la configuración inicial para caso de estudio	Docente	Antes de la sesión de clase	A consideración del docente
Ejecutar la presentación y realización de casos de estudio con el grupo utilizando JIGSAW	Estudiante	Sesión de clase	80 minutos para 10 ejercicio
Realizar la retroalimentación	Docente	Al finalizar cada caso de estudio	20 minutos en total

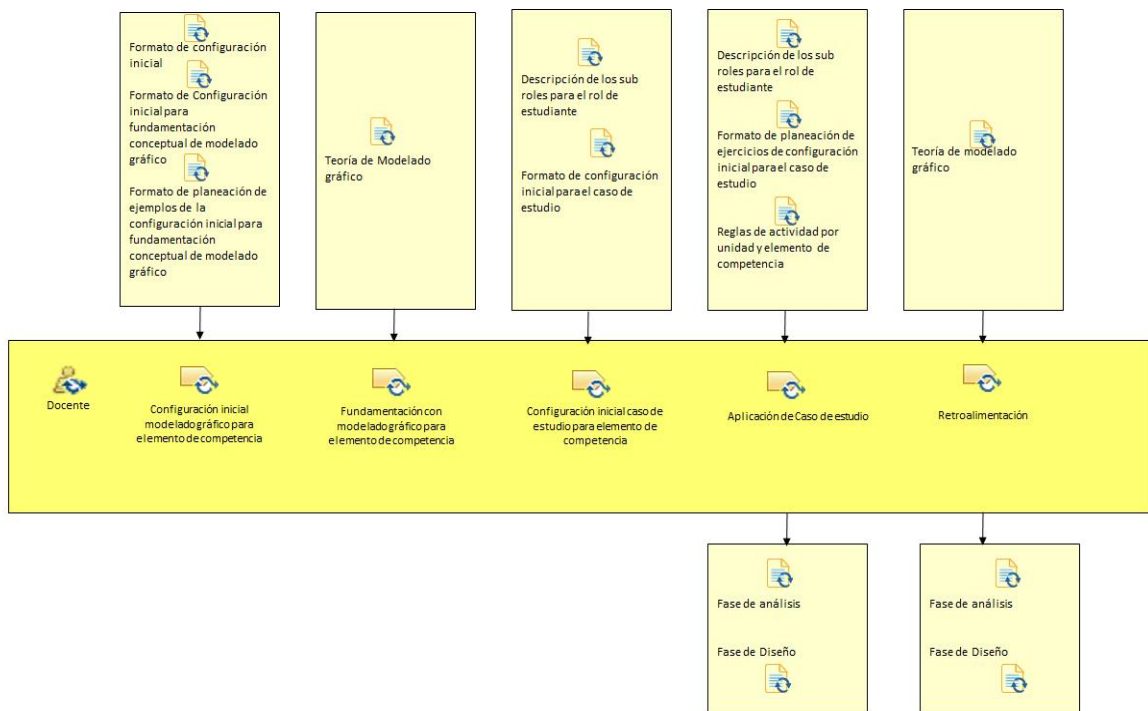


Figura 4.2. Diagrama de detalle de actividad rol docente

Todas las sesiones de clase deben ser realizadas teniendo como base el diagrama de actividades de la figura 90 teniendo únicamente en cuenta el tiempo de realización de cada elemento de competencia. Es así como en la figura 4.2 y 4.3 se presenta el diagrama de detalle de actividad para el rol docente y el rol estudiante respectivamente cuyos diseños fueron realizados con notación SPEM (*Software Process Engineering Metamodel*).

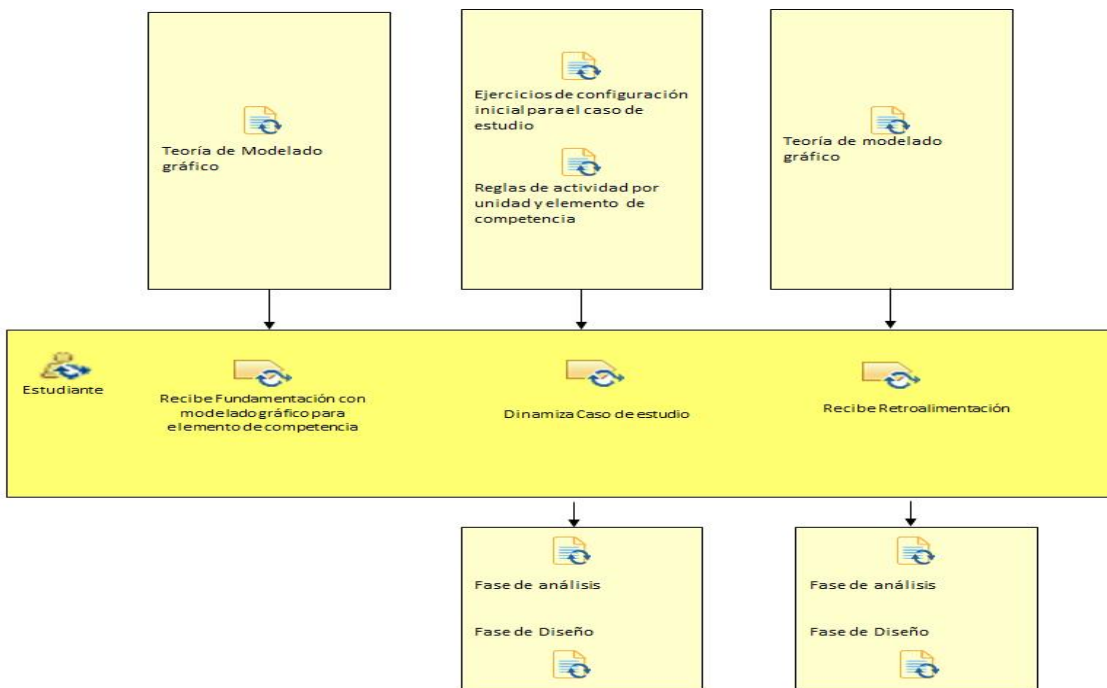


Figura 4.3. Diagrama de detalle de actividad rol estudiante

4.2 Resultados de la investigación a expertos

La metodología planteada fue dada a conocer a 4 expertos de las universidades más importantes del departamento de Nariño (ver tabla 4.2) a quienes de manera personal se les dio a conocer las particularidades de la propuesta e implementaron cada uno al menos una sesión de sus clases con ella.

Tabla 4.2 Expertos por universidad

No	Nombres y apellidos	Universidad	Último Nivel de formación	Años de experiencia en cursos de programación
1	Jesús Insuasty	Universidad de Nariño	Candidato a doctor en educación	16
2	Gonzalo Hernández	Universidad de Nariño	Magister en Ingeniería	20
3	Robinson Jiménez	Universidad Mariana	Magister en educación	16
4	Mauricio Casanova	Institución Universitaria CESMAG	Candidato a Magister en Software Libre	14

A continuación se presenta los resultados del cuestionario (anexo 15) aplicado a cada experto de acuerdo a su percepción con la metodología propuesta:

A la pregunta 1: ¿Cuál es su experiencia docente en cursos de fundamentos de programación?, los resultados se aprecian en la figura 4.4.

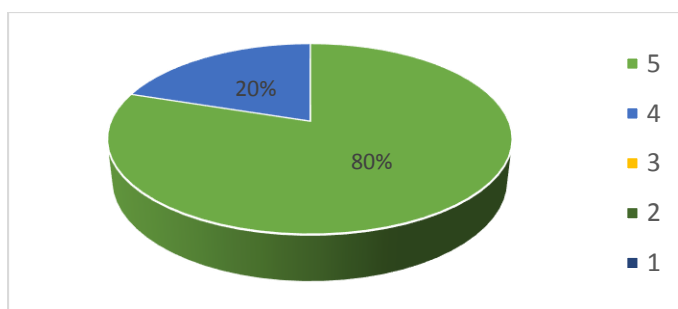


Figura 4.4 Pregunta No. 1 a expertos

En la figura 4.4 se aprecia que cuatro de los expertos consultados calificaron con 5 esta pregunta y uno de ellos asignó el valor de 4, lo que significa que existe una amplia experiencia en el manejo de cursos de fundamentos de programación por parte de los cinco expertos.

A la pregunta 2: ¿Cuál es su experiencia con métodos colaborativos como estrategias didácticas en sus clases?, la figura 4.5 indica sus resultados:

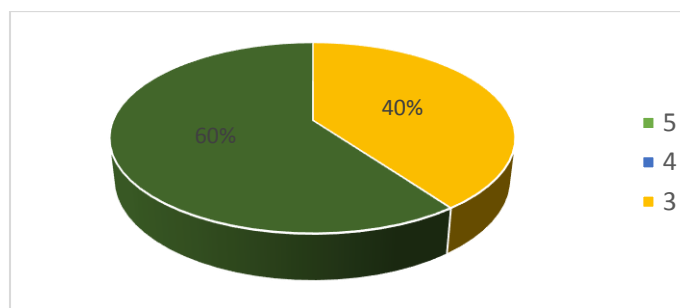


Figura 4.5 Pregunta No. 2 a expertos

En la figura 4.5 se muestra que tres de los expertos calificaron como 2 su experiencia con métodos colaborativos, mientras que dos de ellos la calificaron como 3, lo que concluye que los expertos no han tenido mucha experiencia en la utilización de métodos colaborativos como estrategias didácticas en sus clases.

Frente a la pregunta 3: ¿El modelo colaborativo presentado es claro? Los resultados obtenidos se aprecian en la figura 4.6.

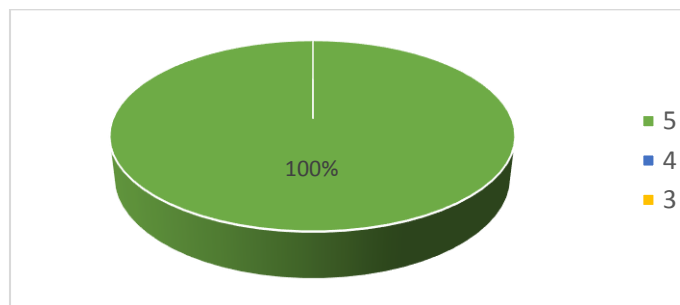


Figura 4.6 Pregunta No.3 a expertos

En la figura 4.6 se puede apreciar que todos los expertos concuerdan que el modelo colaborativo propuesto en la metodología es claro para cada uno de ellos.

A la pregunta 4: ¿El modelado gráfico propuesto es claro?, las apreciaciones de los cinco expertos se presentan en la figura 4.7.

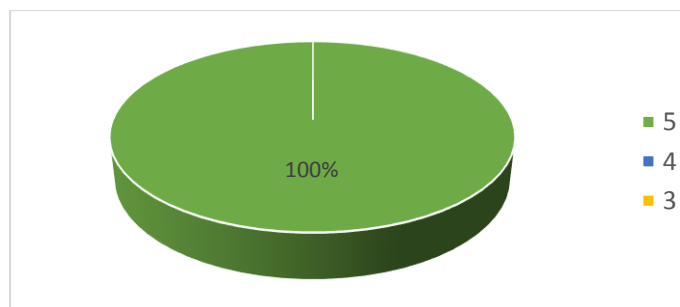


Figura 4.7 pregunta No.4 a expertos

Se puede afirmar que de acuerdo a los resultados presentados en la figura 4.7, los expertos coinciden que el modelo gráfico presentado en la metodología es bastante claro.

A la pregunta 5: Considera usted que el empleo de JIGSAW ¿es una estrategia didáctica apropiada para esta propuesta metodológica?, la figura 4.8 presenta los resultados.

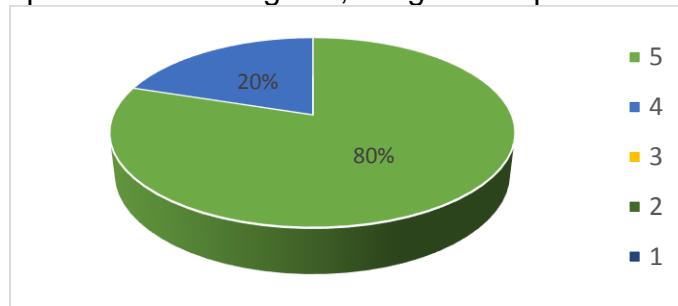


Figura 4.8 Pregunta No.5 a expertos

En la figura 4.8 se puede observar que 4 de los expertos califican con máximo puntaje (5) el empleo de JIGSAW como la estrategia didáctica apropiada a la metodología planteada.

A la pregunta 6: Evalúe únicamente la experiencia del proceso metodológico que realizó usted con su grupo de trabajo en clase, los expertos respondieron como lo indica la figura 4.9.

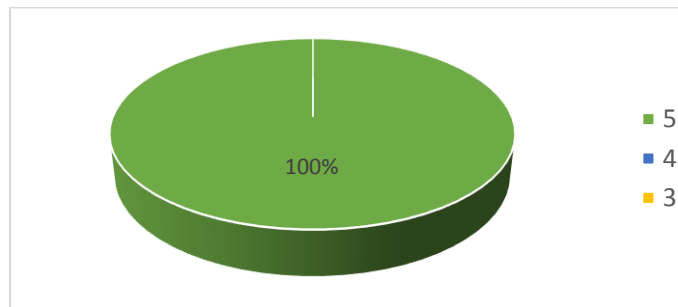


Figura 4.9 Pregunta No.6 a expertos

La figura 4.9 muestra que todos los expertos coinciden con máxima puntuación (5) la experiencia del proceso metodológico que realizó cada uno con su grupo de trabajo en clase.

A la pregunta 7: ¿Cómo considera usted el tomar el D E/S de base fundamental del modelado gráfico propuesto en esta metodología?, los resultados se muestran en la figura 4.10.

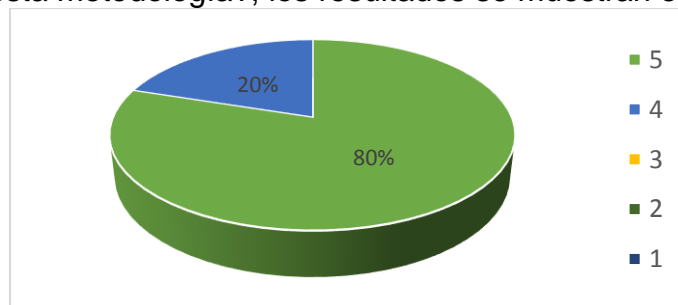


Figura 4.10 Pregunta N0.7 a expertos

Los resultados presentados en la figura 4.10 muestran que cuatro de los expertos califican con puntaje máximo el hecho de tomar el D E/S como base fundamental del modelado gráfico propuesto en esta metodología.

A la pregunta 8: Evalúe el nivel de dificultad o facilidad encontrado por cada unidad de competencia, sus resultados se presentan en la tabla 4.3 y figura 4.11.

Tabla 4.3 Evaluación por unidad de competencia por expertos

Unidad de competencia	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Promedio
Captura y salida de datos	5	5	5	5	5	5,0
Condicionales simples	5	5	5	5	5	5,0
Condicionales compuestos	5	5	5	5	5	5,0
Condicionales anidados	5	5	5	5	5	5,0
Cond. y operadores lógicos	4	4	5	5	4	4,4
Ciclo Para ascendente	4	4	4	5	4	4,2
Ciclo Para descendente	5	4	4	5	5	4,6
Ciclo Mientras	4	4	4	4	4	4,0
Ciclo Hacer Mientras	4	4	4	4	4	4,0
Promedio	4,6	4,4	4,6	4,8	4,6	4,6

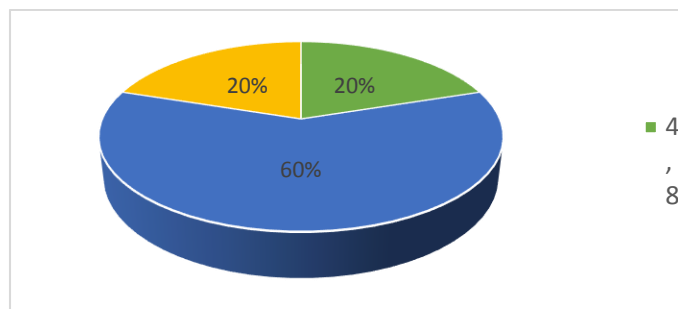


Figura 4.11 Pregunta No.8 a expertos

En la tabla 4.3 y en la figura 4.11 se aprecia que los expertos evalúan con los valores máximos de la escala propuesta a todas las unidades de competencia consideradas en la metodología, donde las unidades de captura y salida de datos junto con condicionales tienen la máxima puntuación.

A la pregunta 9: Los roles de docente y estudiante ¿son pertinentes a la metodología planteada?, las respuestas de los expertos se aprecian en la figura 4.12.

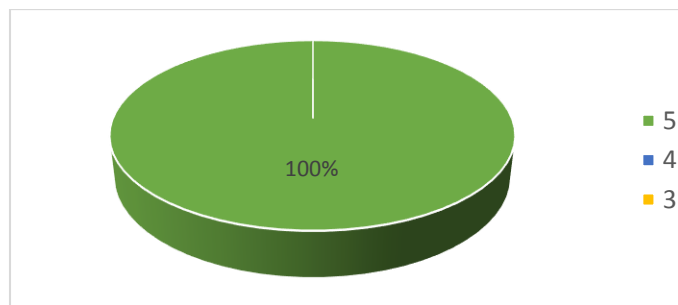


Figura 4.12 Pregunta No.9 a expertos

El grupo total de expertos que los roles generales propuestos en el estudio son apropiados como lo muestra la figura 4.12.

A la pregunta 10. Los sub roles de los estudiantes: Analista y diseñador ¿son pertinentes a la metodología planteada?, los resultados se aprecian en la figura 4.13.

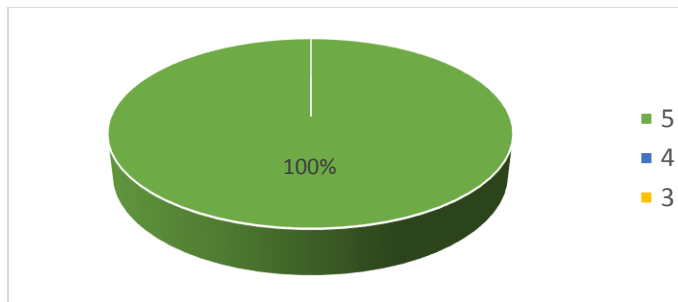


Figura 4.13 Pregunta No.10 a expertos

La gráfica 4.13 demuestra que los todos los expertos estuvieron de acuerdo con los dos sub roles de los estudiantes propuestos en esta investigación.

A la pregunta 11: Evalúe únicamente la experiencia del proceso académico obtenido con su grupo de trabajo, la figura 4.14 muestra las respuestas del grupo de expertos consultados.

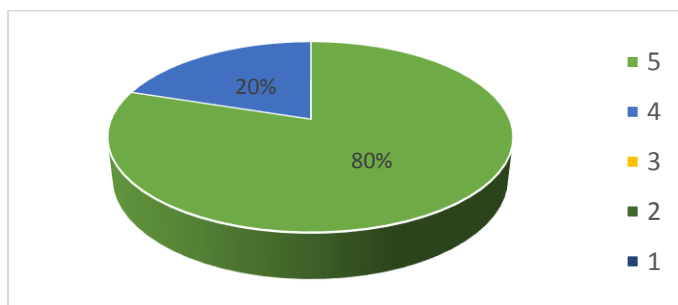


Figura 4.14 Pregunta No.11 a expertos

En la figura 4.14 se aprecia que el grupo de expertos evalúa con la puntuación máxima la experiencia del proceso académico que se sugirió realizar con sus grupos de trabajo.

Finalmente, a la pregunta 12 ¿Recomendaría la metodología desarrollada con otros colegas de trabajo?, en la figura 4.15 se presentan los resultados.

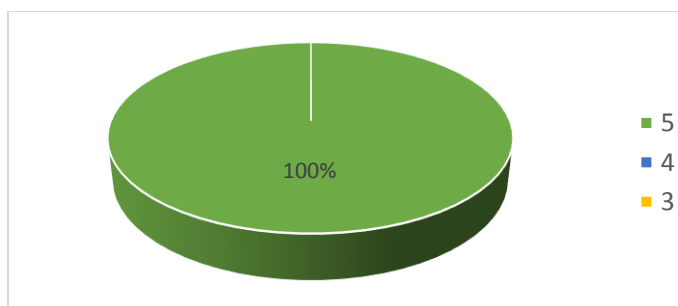


Figura 4.15 Pregunta No. 12 a expertos

En la figura 4.15 se puede apreciar los resultados satisfactorios de la metodología propuesta, a tal punto que los expertos están es su totalidad dispuestos a recomendarla a otros pares.

4.3 Resultados de la investigación con estudiantes

4.2.1 Caracterización de población

La metodología de investigación planteada se desarrolló bajo el paradigma positivista por lo que se fundamentó en el conocimiento científico, con enfoque cuantitativo que permitió examinar datos de manera numérica, utilizando el método empírico analítico porque los datos fueron tratados con técnicas estadísticas y bajo un tipo de investigación correlacional que permitió medir el grado de relación que tuvieron las variables en estudio y con un diseño experimental general basado en un cuasiexperimento como lo muestra la figura 4.16.

G ₁	X	O ₁
G ₂	-	O ₂
G ₃	X	O ₃
G ₄	-	O ₄
G ₅	X	O ₅
G ₆	-	O ₆

Figura 4.16. Diseño experimental

La investigación fue realizada en seis grupos diferentes pertenecientes a tres instituciones de educación superior del municipio de Pasto (Nariño) para el primer curso de fundamentación de programación. En la tabla 4.4 se muestra el diseño experimental aplicado en cada universidad:

Tabla 4.4. Diseño experimental por universidad

Institución	Diseño Experimental
Institución Universitaria CESMAG (I.U.CESMAG) (Privada)	G ₁ X O ₁ G ₂ - O ₂
Universidad Antonio Nariño (UAN) (Privada)	G ₃ X O ₃ G ₄ - O ₄
Universidad de Nariño (UDENAR) (Pública)	G ₅ X O ₅ G ₆ - O ₆

Los grupos G₁, G₃ y G₅ corresponden a los grupos experimentales de cada institución y G₂, G₄ y G₆ fueron sus grupos de control respectivamente, además X fue el tratamiento experimental que consistió en la estrategia didáctica bajo un entorno colaborativo a través de modelado gráfico. A su vez, O₁, O₂, O₃, O₄, O₅, O₆, fueron las post pruebas realizadas al final del tratamiento experimental tanto para los grupos experimentales como los de control. Además, en la tabla 4.5 se encuentra en detalle la información de los grupos participantes (Anexo 16).

Tabla 4.5 Caracterización de los grupos participantes

Institución	Facultad	Programa	Asignatura	Semestre	Jornada
Institución Universitaria CESMAG (I.U.CESMAG)	Ingeniería	Ingeniería de Sistemas	Introducción a la programación	1	Diurna
Universidad Antonio Nariño (UAN)	Electromecánica	Ingeniería electromecánica	Lógica computacional	1	Nocturna
Universidad de Nariño (UDENAR)	Ingeniería	Ingeniería electrónica	Fundamentos de programación	1	Diurna

El primer grupo experimental G_1 fue conformado por 33 estudiantes de la asignatura de Introducción a la programación de primer semestre de Ingeniería de Sistemas correspondientes al periodo académico II-2015 a quienes se les aplicó el tratamiento experimental X y finalmente se le aplicó una prueba posterior O_1 . El grupo de control G_2 estuvo conformado por 27 estudiantes del periodo académico I -2015 del mismo semestre y asignatura del grupo experimental a quienes no se les aplicó tratamiento experimental y las notas obtenidas del tema de estudio fueron consideradas como O_2 . En la tabla 4.6 se aprecia la caracterización de los estudiantes participantes para esta institución.

Tabla 4.6. Caracterización estudiantes de Introducción a la Programación

grupo	Cantidad de estudiantes	Cantidad de hombres	Cantidad de Mujeres	Rango de edad	Cantidad que repiten asignatura
G_1	33	27	6	17 a 21 años	0
G_2	27	23	4	16 a 22 años	2

El segundo grupo experimental G_3 estuvo constituido por 11 estudiantes de la asignatura de Lógica Computacional de primer semestre de Ingeniería Electromecánica correspondientes al periodo académico II-2015 a quienes también se les aplicó el mismo tratamiento X con su correspondiente evaluación O_3 y su grupo de control G_4 fueron los 14 estudiantes del periodo académico I -2015 del mismo semestre y asignatura del grupo experimental a quienes tampoco se les aplicó tratamiento experimental, considerando también las notas obtenidas del tema en estudio O_4 . En la tabla 4.7 se aprecia la caracterización de los estudiantes participantes para esta Universidad.

Tabla 4.7. Caracterización estudiantes de Lógica Computacional

grupo	Cantidad de estudiantes	Cantidad de hombres	Cantidad de Mujeres	Rango de edad	Cantidad que repiten asignatura
G ₃	11	11	0	18 a 35 años	0
G ₄	14	14	0	22 a 40 años	0

El tercer grupo experimental G₅ estuvo constituido por 29 estudiantes de la asignatura de fundamentos de programación de primer semestre de Ingeniería Electrónica correspondientes al periodo académico II-2015 a quienes también se les aplicó el mismo tratamiento X con su correspondiente evaluación O₅ y su grupo de control G₆ fueron los 33 estudiantes del periodo académico I -2015 del mismo semestre y asignatura del grupo experimental a quienes tampoco se les aplicó tratamiento experimental, considerando también las notas obtenidas del tema en estudio O₆. En la tabla 4.8 se aprecia la caracterización de los estudiantes participantes para esta Universidad.

Tabla 4.8. Caracterización estudiantes de Fundamentos de Programación

grupo	Cantidad de estudiantes	Cantidad de hombres	Cantidad de Mujeres	Rango de edad	Cantidad que repiten asignatura
G ₄	29	24	5	17 a 20 años	0
G ₅	33	30	3	17 a 19 años	0

En la tabla 4.9 se muestra el consolidado de la información correspondiente a grupos experimentales y de control por universidad.

Tabla 4.9 Caracterización de grupos experimentales y de control

Institución	Grupo	Tipo de grupo	Periodo académico	Cantidad de estudiantes
I.U. CESMAG	G ₁	Experimental	II-2015	33
	G ₂	Control	I-2015	27
UAN	G ₃	Experimental	II-2015	11
	G ₄	Control	I-2015	14
UDENAR	G ₅	Experimental	II-2015	29
	G ₆	Control	I-2015	33

4.2.2 Resultados de postprueba grupo de control

Los grupos de control (G₂, G₄ y G₆) para cada grupo experimental estuvieron conformados por los cursos correspondientes al I periodo académico de 2015 (anexo 17, 18 y 19) para cada uno de los cursos con los cuales se llevó a cabo la investigación, con los cuales se trabajó con la metodología tradicional donde el docente expone la temática de estudio, se llevan a cabo una serie de ejemplos y ejercicios para finalmente realizar evaluaciones.

El proceso investigativo para estos tres grupos de control consistió en la realización de cuatro actividades evaluativas por cada unidad de competencia para cada semestre, que consistieron en la aplicación de dos talleres grupales con la participación de dos estudiantes (40%) y dos seguimientos individuales (60%), con el propósito de establecer cuantitativamente el proceso de apropiación de cada una de las unidades mencionadas con la metodología tradicional de clase.

Los resultados de postprueba aplicados para el curso Introducción a la Programación del grupo de control en el periodo académico I-2015 se muestran en la tabla 4.10.

Tabla 4.10. Promedio del grupo de control para Introducción a la Programación

Unidad competencia de	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
Captura y salida de datos	4,30	4,00	4,15	3,80	4,00	3,90	4,00
Condicionales	3,30	3,20	3,25	3,10	3,00	3,05	3,13
Ciclos	4,00	3,80	3,90	3,70	3,60	3,65	3,75

De igual manera, los resultados de postprueba aplicados para el curso Lógica Computacional para el grupo de control para el periodo académico I-2015 se muestran en la tabla 4.11.

Tabla 4.11. Promedio del grupo de control para Lógica Computacional

Unidad competencia de	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
Captura y salida de datos	4,10	4,00	4,05	3,50	3,50	3,50	3,72
Condicionales	3,90	3,80	3,85	3,70	3,20	3,45	3,61
Ciclos	4,00	3,50	3,75	3,60	3,30	3,45	3,57

Finalmente, los resultados de postprueba aplicados para el curso Fundamentos de Programación para el grupo de control en el periodo académico I-2015 se muestran en la tabla 4.12.

Tabla 4.12. Promedio del grupo de control para Fundamentos de Programación

Unidad competencia de	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
Captura y salida de datos	4,70	4,40	4,55	4,30	4,20	4,25	4,37

Unidad de competencia	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
Condicionales	4,30	4,00	4,15	4,00	4,00	4,00	4,04
Ciclos	4,2	4,3	4,30	4,10	4,20	4,15	4,21

En la tabla 4.13 se aprecia los resultados definitivos de grupo de control para cada curso.

Tabla 4.13. Promedio de notas definitivas grupos de control

Curso	Captura y salida de datos	Condicionales	Ciclos	Definitiva
Introducción a la programación	4,00	3,13	3,75	3,63
Lógica Computacional	3,72	3,61	3,57	3,63
Fundamentos de programación	4,37	4,04	4,21	4,21
Promedio	4,03	3,59	3,84	3,82

Los resultados de las notas mostradas en la tabla 4.13 y que fueron obtenidos en los tres cursos para grupos de control, evidencian dificultad en el estudio de condicionales, un mejor resultado en el manejo de ciclos y una buena apropiación en el manejo de captura y salida de datos.

Los resultados definitivos evidencian de igual manera la influencia que tiene el proceso de selección de los estudiantes en la universidad pública, frente a los de las dos universidades privadas cuya política de ingreso es abierta.

4.2.3 Resultados de postprueba grupo experimental

Una vez aplicado el tratamiento investigativo en cada grupo experimental (G_1 , G_3 y G_5) se procedió a la realización de cuatro actividades evaluativas por cada unidad de competencia para cada semestre (anexos 20, 21 y 22), que consistieron en la aplicación de dos talleres grupales con la participación de dos estudiantes (40%) y dos seguimientos individuales (60%), con el propósito de establecer cuantitativamente el proceso de apropiación de cada una de las unidades mencionadas.

Los resultados de postprueba aplicados para el curso Introducción a la Programación para el grupo experimental para el periodo académico II-2015 se muestran en la tabla 4.14.

Tabla 4.14. Promedio del grupo experimental Introducción a la programación

Unidad de competencia	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
Captura y salida de datos	4,50	4,70	4,60	4,60	4,40	4,50	4,54
Condicionales	4,10	4,00	4,05	3,90	3,80	3,85	3,93
Ciclos	4,40	4,30	4,35	4,00	4,00	4,00	4,14

De igual manera, los resultados de postprueba aplicados para el curso Lógica Computacional para el grupo experimental para el periodo académico II-2015 se muestran en la tabla 4.15.

Tabla 4.15. Promedio del grupo experimental Lógica Computacional

Unidad de competencia	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
Captura y salida de datos	4,40	4,80	4,60	4,20	4,40	4,30	4,42
Condicionales	4,60	4,60	4,60	4,20	4,40	4,30	4,42
Ciclos	4,50	4,40	4,45	4,30	4,10	4,20	4,30

Finalmente, los resultados de postprueba aplicados para el curso Fundamentos de Programación para el grupo experimental para el periodo académico II-2015 se muestran en la tabla 4.16.

Tabla 4.16. Promedio del grupo experimental Fundamentos de Programación

Unidad de competencia	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
Captura y salida de datos	5,00	4,80	4,80	4,80	4,88	5,00	4,80
Condicionales	4,90	4,30	4,60	4,45	4,63	4,90	4,30
Ciclos	4,65	4,60	4,80	4,70	4,68	4,65	4,60

Como se puede apreciar en las tablas 4.14, 4.15 y 4.16, los resultados obtenidos en los talleres (que son grupales y estuvieron conformados por 2 estudiantes) son superiores a los exámenes individuales, lo cual demuestra que el estudiante adquiere mejores resultados trabajando en grupo que haciéndolo individual, lo cual concluye que el proceso colaborativo llevado con el conjunto de estudiantes incide de manera positiva en sus resultados grupales.

De igual manera, se puede observar que la unidad de competencia relacionada con “Condicionales” es la que menor valor cuantitativo tiene frente a las otras dos unidades de competencia en los tres grupos experimentales, lo indica que los procesos lógicos que requieren evaluación de proposiciones tienen un cierto grado de dificultad para los estudiantes.

Por otro lado, se puede apreciar en los tres grupos que en la unidad de competencia relacionada con ciclos el estudiante se siente más a gusto ya que sus notas mejoran con relación a la unidad de condicionales.

Finalmente se puede concluir que en la unidad de competencia “Captura y salida de datos” hay una buena apropiación por parte del estudiante ya que es la que más valor cuantitativo tiene en cada grupo con relación a las otras unidades de competencia. En la tabla 4.17 se aprecia los resultados definitivos de cada curso.

Tabla 4.17. Promedio de notas definitivas por cada curso

Curso	Captura y salida de datos	Condicionales	Ciclos	Definitiva
Introducción a la programación	4,54	3,93	4,14	4,20
Lógica Computacional	4,42	4,42	4,30	4,38
Fundamentos de programación	4,80	4,30	4,60	4,57

Los resultados de las notas mostradas en la tabla 4.17 y que fueron obtenidos en los tres cursos, están por encima de la escala cuantitativa de 4.0, lo cual demuestra que hay una apropiación buena por parte de los estudiantes en forma general a finalizar dichos cursos.

4.2.4 Resultados de grupo experimental vs grupo de control

Una vez terminado el proceso investigativo, a continuación se hace el paralelo entre el grupo de control y el experimental para cada uno de los cursos de este estudio. En la tabla 4.18 se exhibe los resultados finales obtenidos por el grupo de control y experimental para el curso de fundamentos de programación.

Tabla 4.18. Grupo de control vs grupo experimental de Fundamentos de Programación

Unidad de competencia	Grupo de control			Grupo experimental		
	Total Talleres	Total Examen	Definitiva	Total Talleres	Total Examen	Definitiva
Captura y salida de datos	4,15	3,90	4,00	4,60	4,50	4,54
Condicionales	3,25	3,05	3,13	4,05	3,85	3,93
Ciclos	3,90	3,65	3,75	4,35	4,00	4,14

La tabla 4.18 muestra la incidencia que tuvo el tratamiento experimental propuesto en esta investigación para el curso de Introducción a la Programación del Programa de Ingeniería de Sistemas de la Institución Universitaria CESMAG. Los datos presentados evidencian que las notas obtenidas por el grupo experimental son mayores a las registradas por el grupo de control tanto en las actividades grupales (Talleres) como en las individuales (Exámenes) como lo indican las figuras 4.17 y 4.18.

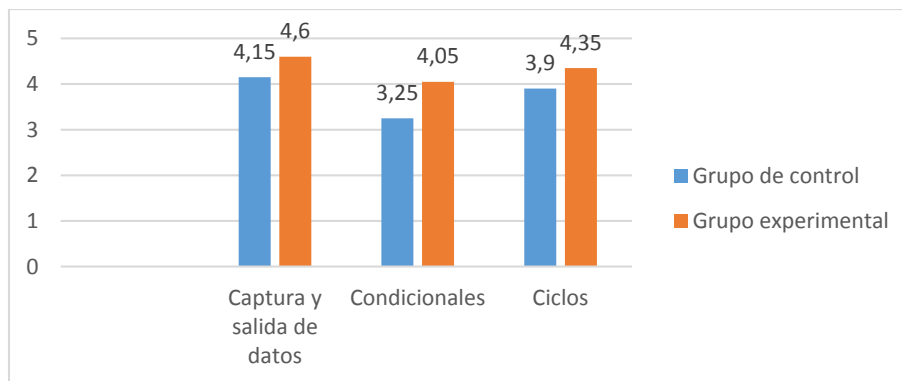


Figura 4.17. Resultado para talleres de Introducción a la Programación

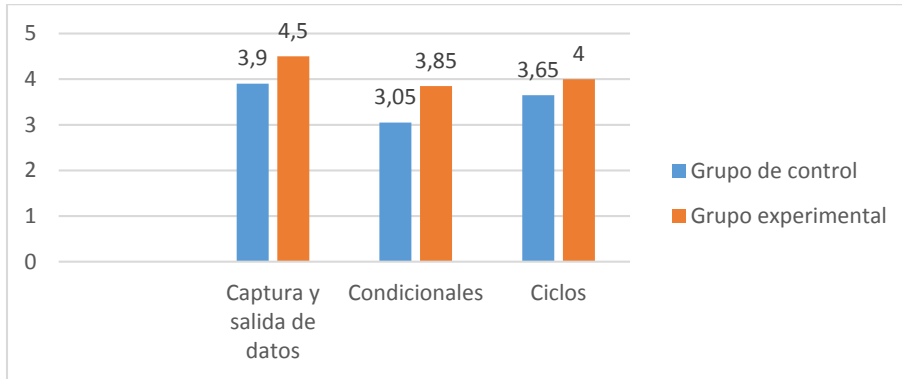


Figura 4.18. Resultado para exámenes de Introducción a la Programación

En la tabla 4.19 se exhibe los resultados finales obtenidos por el grupo de control y experimental para el curso de Lógica Computacional.

Tabla 4.19 Grupo de control vs grupo experimental de Lógica Computacional

Unidad de competencia	Grupo de control			Grupo experimental		
	Total Talleres	Total Examen	Definitiva	Total Talleres	Total Examen	Definitiva
Captura y salida de datos	4,05	3,50	3,72	4,60	4,30	4,42
Condicionales	3,85	3,45	3,61	4,60	4,30	4,42
Ciclos	3,75	3,45	3,57	4,45	4,20	4,30

En la tabla 4.19 se muestra la incidencia que tuvo también el tratamiento experimental para el curso de Lógica Computacional del programa de Ingeniería Electromecánica de la Universidad Antonio Nariño. Al igual que en la I.U.CESMAG, los datos obtenidos evidencian que los resultados del grupo experimental son mejores a los del grupo de control tanto en las actividades grupales (Talleres) como en las individuales (Exámenes) como lo indican las figuras 4.19 y 4.20.

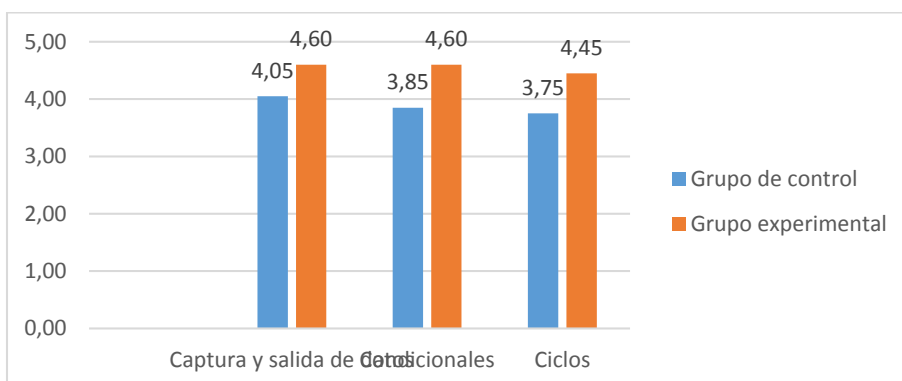


Figura 4.19. Resultado para talleres de Lógica Computacional

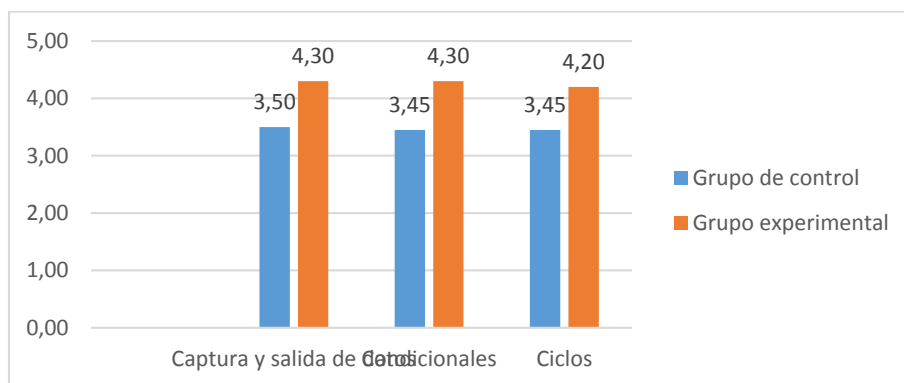


Figura 4.20. Resultado para exámenes de Lógica Computacional
En la tabla 4.20 se exhibe los resultados finales obtenidos por el grupo de control y experimental para el curso de Fundamentos de Programación.

Tabla 4.20. Grupo de control vs grupo experimental de Fundamentos de Programación

Unidad de competencia	Grupo de control			Grupo experimental		
	Total Talleres	Total Examen	Definitiva	Total Talleres	Total Examen	Definitiva
Captura y salida de datos	4,55	4,25	4,37	4,80	5,00	4,80
Condicionales	4,15	4,00	4,04	4,60	4,90	4,30
Ciclos	4,30	4,15	4,21	4,80	4,65	4,60

En la tabla 4.20 se muestra la incidencia que tuvo también el tratamiento experimental para el curso de Fundamentos de Programación del programa de Ingeniería Electrónica de la Universidad de Nariño. Al igual que en la I.U.CESMAG y la UAN, los datos obtenidos evidencian que los resultados del grupo experimental son mejores a los del grupo de control tanto en las actividades grupales (Talleres) como en las individuales (Exámenes) como lo indican las figuras 4.21 y 4.22.

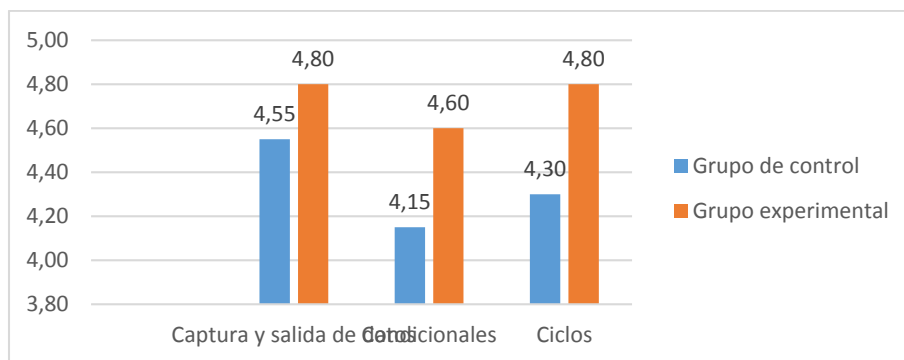


Figura 4.21. Resultado para talleres de Fundamentos de Programación

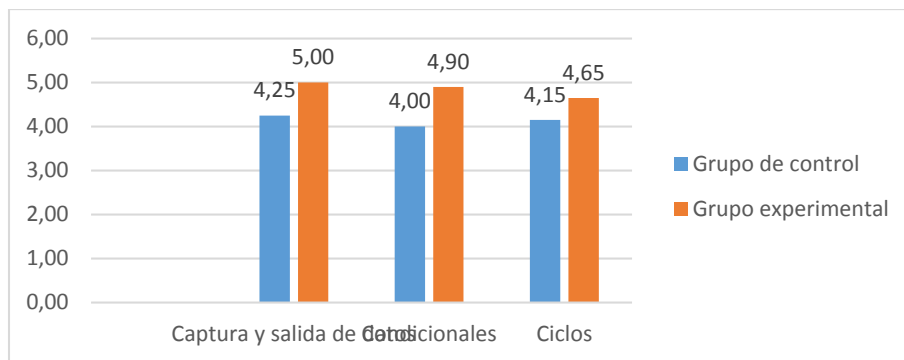


Figura 4.22. Resultado para exámenes de Fundamentos de Programación

Finalmente, se realiza un análisis estadístico para determinar mediante la distribución de probabilidad T de Student, la cual se utiliza para examinar la diferencia entre dos muestras independientes y pequeñas [97], la diferencia que existe entre las notas obtenidas por el grupo experimental y las del grupo de control. En las tablas 4.21, 4.22 y 4.23 se muestra el análisis comparativo entre el grupo experimental y el de control respecto a las tres unidades de competencia planteadas en la investigación para el curso Introducción a la Programación.

Tabla 4.21. T de Student para Entrada y salida de datos de Introducción a la Programación

Concepto	Grupo experimental	Grupo de control
Media	4,49575758	4,02962963
Varianza	0,73820227	0,67062678
Observaciones	33	27
Varianza agrupada	0,70790981	
Diferencia hipotética de las medias	0	
Grados de libertad	58	
Estadístico t	2,13490917	
P(T<=t) una cola	0,01850238	
Valor crítico de t (una cola)	1,67155276	
P(T<=t) dos colas	0,03700477	
Valor crítico de t (dos colas)	2,00171748	

Tabla 4.22. T de Student para Condicionales de Introducción a la Programación

Concepto	Grupo experimental	Grupo de control
Media	3,87545455	3,1437037
Varianza	0,57988182	1,09901652
Observaciones	33	27
Varianza agrupada	0,81259738	
Diferencia hipotética de las medias	0	
Grados de libertad	58	
Estadístico t	3,1281585	
P(T<=t) una cola	0,0013754	
Valor crítico de t (una cola)	1,67155276	
P(T<=t) dos colas	0,0027508	
Valor crítico de t (dos colas)	2,00171748	

Tabla 4.23. T de Student para Ciclos de Introducción a la Programación

Concepto	Grupo experimental	Grupo de control
Media	4,12484848	3,66148148
Varianza	0,55394451	1,0152208
Observaciones	33	27
Varianza agrupada	0,76072353	
Diferencia hipotética de las medias	0	
Grados de libertad	58	
Estadístico t	2,04726897	
P(T<=t) una cola	0,02258528	
Valor crítico de t (una cola)	1,67155276	
P(T<=t) dos colas	0,04517056	
Valor crítico de t (dos colas)	2,00171748	

Las tablas 4.21, 4.22 y 4.23 evidencian el resultado de la aplicación del tratamiento experimental en cada una de las unidades de competencia establecidas, los resultados estadísticos obtenidos por el grupo experimental G1 tienen un valor estadístico t (2,13490917, 3,1281585 y 2,04726897) mayor tanto al valor crítico de t de una cola (1,67155276, 1,67155276 y 1,67155276) como al valor crítico para dos colas (2,00171748, 2,00171748 y 2,00171748) y el valor de P (para una y dos colas) es menor al 5% en los tres casos, lo cual concluye que la diferencia de notas entre grupo experimental y grupo de control por cada unidad de competencia es estadísticamente significativa para $t=5\%$.

En las tablas 4.24, 4.25 y 4.26 se muestra el análisis comparativo entre el grupo experimental y el de control respecto a las tres unidades de competencia planteadas en la investigación para el curso Lógica Computacional.

Tabla 4.24. T de Student para Entrada y salida de datos de Lógica Computacional

Concepto	Grupo experimental	Grupo de control
Media	4,44454545	3,68071429
Varianza	0,75478727	0,60919176
Observaciones	11	14
Varianza agrupada	0,67249416	
Diferencia hipotética de las medias	0	
Grados de libertad	23	
Estadístico t	2,31176389	
P(T<=t) una cola	0,01504712	
Valor crítico de t (una cola)	1,71387153	
P(T<=t) dos colas	0,03009423	
Valor crítico de t (dos colas)	2,06865761	

Tabla 4.25. T de Student para Condicionales de Lógica Computacional

Concepto	Grupo experimental	Grupo de control
Media	4,36181818	3,58571429
Varianza	0,46483636	1,0465033
Observaciones	11	14
Varianza agrupada	0,79360463	

Concepto	Grupo experimental	Grupo de control
Diferencia hipotética de las medias	0	
Grados de libertad	23	
Estadístico t	2,16226104	
P(T<=t) una cola	0,02062095	
Valor crítico de t (una cola)	1,71387153	
P(T<=t) dos colas	0,04124189	
Valor crítico de t (dos colas)	2,06865761	

Tabla 4.26. T de Student para Ciclos de Lógica Computacional

Concepto	Grupo experimental	Grupo de control
Media	4,28909091	3,613571429
Varianza	0,40290909	0,55304011
Observaciones	11	14
Varianza agrupada	0,48776575	
Diferencia hipotética de las medias	0	
Grados de libertad	23	
Estadístico t	2,40061536	
P(T<=t) una cola	0,01242588	
Valor crítico de t (una cola)	1,71387153	
P(T<=t) dos colas	0,02485177	
Valor crítico de t (dos colas)	2,06865761	

De igual manera, Las tablas 4.24, 4.25 y 4.26 evidencian el resultado de la aplicación del tratamiento experimental en cada una de las unidades de competencia ya mencionadas, los resultados estadísticos obtenidos por el grupo experimental G2 también tienen un valor estadístico t (2,31176389, 2,16226104 y 2,40061536) mayor tanto al valor crítico de t de una cola (1,71387153, 1,71387153 y 1,71387153) como al valor crítico para dos colas (2,06865761, 2,06865761 y 2,06865761) y el valor de P (para una y dos colas) es menor al 5% en los tres casos, lo cual concluye que la diferencia de notas entre grupo experimental y grupo de control por cada unidad de competencia es estadísticamente significativa para $t=5\%$.

Por último, en las tablas 4.27, 4.28 y 4.29 se muestra el análisis comparativo entre el grupo experimental y el de control respecto a las tres unidades de competencia planteadas en la investigación para el curso Fundamentos de programación.

Tabla 4.27. T de Student para Entrada y salida de datos de Fundamentos de Programación

Concepto	Grupo experimental	Grupo de control
Media	4,85517241	4,39393939
Varianza	0,01470443	1,37121212
Observaciones	29	33
Varianza agrupada	0,7381752	
Diferencia hipotética de las medias	0	
Grados de libertad	60	
Estadístico t	2,10911757	
P(T<=t) una cola	0,01955823	
Valor crítico de t (una cola)	1,67064886	

Concepto	Grupo experimental	Grupo de control
P(T<=t) dos colas	0,03911647	
Valor crítico de t (dos colas)	2,00029782	

Tabla 4.28. T de Student para Condicionales de Fundamentos de Programación

Concepto	Grupo experimental	Grupo de control
Media	4,55862069	4
Varianza	0,54161946	1,28125
Observaciones	29	33
Varianza agrupada	0,93608908	
Diferencia hipotética de las medias	0	
Grados de libertad	60	
Estadístico t	2,26839432	
P(T<=t) una cola	0,01345841	
Valor crítico de t (una cola)	1,67064886	
P(T<=t) dos colas	0,02691682	
Valor crítico de t (dos colas)	2,00029782	

Tabla 4.29. T de Student para Ciclos de Fundamentos de Programación

Concepto	Grupo experimental	Grupo de control
Media	4,6862069	4,15151515
Varianza	0,39533867	1,10132576
Observaciones	29	33
Varianza agrupada	0,77186512	
Diferencia hipotética de las medias	0	
Grados de libertad	60	
Estadístico t	2,39107334	
P(T<=t) una cola	0,00997637	
Valor crítico de t (una cola)	1,67064886	
P(T<=t) dos colas	0,01995274	
Valor crítico de t (dos colas)	2,00029782	

Finalmente, las tablas 4.27, 4.28 y 4.29 evidencian el resultado de la aplicación del tratamiento experimental en cada una de las unidades de competencia, los resultados estadísticos obtenidos por el grupo experimental G3 al igual que G1 y G2 tienen un valor estadístico t (2,10911757, 2,26839432 y 2,39107334) mayor tanto al valor crítico de t de una cola (1,67064886, 1,67064886 y 1,67064886) como al valor crítico para dos colas (2,00029782, 2,00029782 y 2,00029782) y el valor de P (para una y dos colas) es menor al 5% en los tres casos, lo cual concluye que la diferencia de notas entre grupo experimental y grupo de control por cada unidad de competencia es estadísticamente significativa para $t=5\%$.

Por lo tanto, el análisis estadístico anterior demuestra la incidencia que tiene el tratamiento presentado en esta investigación en los grupos experimentales frente a los grupos de control, estableciendo que al incorporar estrategias didácticas adecuadas en la complejidad de una temática y combinándola en un entorno colaborativo de una manera apropiada se obtienen resultados académicos que benefician de una manera directa a los estudiantes.

El proceso colaborativo llevado a cabo en la investigación permitió que los estudiantes se interrelacionara de forma activa, logrando así una dinámica interactiva que mejoró las relaciones interpersonales del grupo, que por cierto fue sencilla de realizar dadas las características de los grupos experimentales por tratarse de estudiantes de primer semestre de pregrado en los cuales aún no existe la diferenciación de grupos marcados que se presentan generalmente con el paso de tiempo. Esto permitió que los estudiantes se acoplaran fácilmente al rol establecido el cual fue dinámico durante los diversos casos de estudio planteados.

En el estudio se pudo evidenciar que al mezclar la rigurosidad de unas teorías (en este caso las tres unidades de competencia) con procesos que se encuentren dentro de los esquemas de acción de los estudiantes (dinamización de actividades) y mediado con una didáctica de participación activa (estrategia colaborativa utilizando el método JIGSAW), es más sencillo obtener por tiempos más prolongados la atención del estudiante, que involucrado en un proceso totalmente interactivo comienza a construir las bases de su lógica computacional cuya importancia es vital para los procesos que se requieren en posteriores semestres.

Finalmente y especialmente en el campo de la enseñanza para formación de futuros constructores de software, se hace necesario involucrar aspectos didácticos que coloquen al estudiante como elemento central y activo del proceso de aprendizaje y que lo incluyan de una manera totalmente activa en la construcción de su lógica computacional, inculcándole la necesidad del autoaprendizaje en un ambiente colaborativo que mejore las actitudes y aptitudes del estudio y trabajo en equipo.

CAPÍTULO 5: CONCLUSIONES, TRABAJOS FUTUROS Y DIVULGACIÓN

5.1 Conclusiones

Después de una ardua labor investigativa se obtienen las siguientes conclusiones a todo este proceso que deja las puertas abiertas para seguir mejorando esta etapa inicial de la metodología propuesta.

- ✓ De acuerdo a los resultados obtenidos tanto por el grupo de expertos como con los grupos de estudiantes se puede afirmar que la hipótesis de investigación planteada se cumple, para HI: La propuesta metodológica mediante aprendizaje colaborativo y modelado gráfico contribuye positivamente en la enseñanza de los fundamentos de programación.
- ✓ La evaluación realizada por los expertos determina la validez de la propuesta metodológica planteada al cuantificar sus aspectos con los mayores valores de la escala propuesta.
- ✓ La importancia de la propuesta planteada se evidencia en la recomendación que expresan de forma unánime y en su totalidad el grupo de expertos al otorgar el máximo valor cuando se les preguntó si recomendarían la metodología a otros colegas de trabajo.
- ✓ Los resultados de las notas obtenidos en los tres cursos para los grupos de control, evidencian dificultad en el estudio de condicionales, un mejor resultado en el manejo de ciclos y una buena apropiación en el manejo de captura y salida de datos.
- ✓ Los resultados definitivos llevados a cabo por los grupos de control y experimentales evidencian la influencia que tiene el proceso de selección de los estudiantes en la universidad pública, frente a los de las dos universidades privadas cuya política de ingreso es abierta.
- ✓ Los resultados obtenidos por los grupos experimentales en los talleres (que son grupales y estuvieron conformados por 2 estudiantes) son superiores a los exámenes individuales, lo cual demuestra que el estudiante adquiere mejores resultados trabajando en equipo que haciéndolo individual, lo cual concluye que el proceso colaborativo llevado con el conjunto de estudiantes incide de manera positiva en sus resultados colectivos obtenidos.
- ✓ Se puede observar tanto en los grupos experimentales como en los de control que la unidad de competencia relacionada con “Condicionales” es la que menor valor cuantitativo tiene frente a las otras dos unidades de competencia, lo indica que los procesos lógicos que requieren evaluación de proposiciones tienen un cierto grado de dificultad para los estudiantes.
- ✓ La investigación mostró en los seis grupos (experimentales y de control) que en la unidad de competencia relacionada con ciclos el estudiante se siente más a gusto ya que sus notas mejoran con relación a la unidad de condicionales.
- ✓ Tanto en grupos experimentales como de control, la unidad de competencia “Captura y salida de datos” tiene una buena apropiación por parte del estudiante ya que es la que más valor cuantitativo tiene en cada grupo con relación a las otras unidades de competencia.
- ✓ Los resultados de las notas obtenidos en los tres cursos, están por encima de la escala cuantitativa de 4.0, lo cual demuestra que hay una apropiación buena por parte de los estudiantes en forma general a finalizar dichos cursos.
- ✓ En los tres cursos de Fundamentos de Programación, Lógica Computacional y Fundamentos de Programación, los datos obtenidos evidencian que los resultados de los

grupos experimentales son mejores a los de los grupos de control tanto en las actividades grupales (Talleres) como en las individuales (Exámenes).

- ✓ Los resultados estadísticos obtenidos por los tres grupos experimentales G_1 , G_2 y G_3 tienen un valor estadístico t mayor tanto al valor crítico de t de una cola como al valor crítico para dos colas y el valor de P (para una y dos colas) es menor al 5% en cada caso, lo cual concluye que la diferencia de notas entre los grupos experimentales y los grupos de control por cada unidad de competencia son estadísticamente significativas para $\alpha=5\%$.
- ✓ El análisis estadístico realizado demuestra la incidencia que tiene el tratamiento presentado en esta investigación en los grupos experimentales frente a los grupos de control, estableciendo que al incorporar estrategias didácticas adecuadas en la complejidad de una temática y combinándola en un entorno colaborativo de una manera apropiada se obtienen resultados académicos que benefician de una manera directa a los estudiantes.
- ✓ El proceso colaborativo llevado a cabo en la investigación permitió que los estudiantes se interrelacionara de forma activa, logrando así una dinámica interactiva que mejoró las relaciones interpersonales del grupo, que por cierto fue sencilla de realizar dadas las características de los grupos experimentales por tratarse de estudiantes de primer semestre de pregrado en los cuales aún no existe la diferenciación de grupos marcados que se presentan generalmente con el paso de tiempo. Esto permitió que los estudiantes se acoplaran fácilmente al rol establecido el cual fue dinámico durante los diversos casos de estudio planteados.
- ✓ En el estudio se pudo evidenciar que al mezclar la rigurosidad de unas teorías (en este caso las tres unidades de competencia) con procesos que se encuentren dentro de los esquemas de acción de los estudiantes (dinamización de actividades) y mediado con una didáctica de participación activa (estrategia colaborativa utilizando el método JIGSAW), es más sencillo obtener por tiempos más prolongados la atención del estudiante, que involucrado en un proceso totalmente interactivo comienza a construir las bases de su lógica computacional cuya importancia es vital para los procesos que se requieren en posteriores semestres.
- ✓ En el campo de la enseñanza para formación de futuros constructores de software, se hace necesario involucrar aspectos didácticos que coloquen al estudiante como elemento central y activo del proceso de aprendizaje y que lo incluyan de una manera totalmente activa en la construcción de su lógica computacional, inculcándole la necesidad del autoaprendizaje en un ambiente colaborativo que mejore las actitudes y aptitudes del estudio y trabajo en equipo.

5.2 Trabajos futuros

Una vez terminada la investigación es necesario seguirla mejorando y para ello es necesario tener en cuenta las siguientes acciones:

- ✓ Es pertinente realizar la automatización del proceso que aborda la fase de análisis, es decir que mediante una solución de software se pueda diseñar el D E/S que automatizaría la creación de la GUI y la fase de Diseño correspondiente.
- ✓ Es posible adaptar un nuevo módulo que permita extraer de la fase de análisis y diseño un proceso pensando en un paradigma de programación como el Orientado a Objetos, que a través de la fase de análisis se puede fácilmente obtener la fase de diseño mediante un diagrama inicial de clases

- ✓ Es posible adaptar esta metodología por la sencillez presentada en el entorno colaborativo y gráfico a niveles más tempranos que el realizado por esta investigación en el pregrado, es decir plantearla a niveles de niños y jóvenes de educación primaria y secundaria.
- ✓ Es pertinente adecuar esta metodología que inicialmente está pensada para la enseñanza de los fundamentos de programación y re arreglar su contenido para que sea base como proceso de aprendizaje de estudiantes.

5.2 Divulgación

Durante este estudio se han realizado los siguientes productos investigativos que se han presentado como artículos científicos (ver tabla 5.1) y ponencias en congresos internacionales (ver tabla 5.2)

Tabla 5.1 Artículos en revistas

No	Título	Tipo	Nombre	Autores	ISSN
1	Estrategia colaborativa en entornos tridimensionales como estrategia didáctica de aprendizaje de estructuras iterativas en programación computacional (Anexo 23)	Artículo en revista Indexada C por Colciencias	Revista Investigium Ire	Javier Jiménez César Collazos Julio Hurtado Libardo Pantoja	Vol.VI No. 2 pp.1-158 Julio - Diciembre 2015 ISSN Impreso:2216-1473
2	Estrategia colaborativa en entornos tridimensionales como estrategia didáctica de aprendizaje de estructuras iterativas en programación computacional (Anexo 23)	Artículo en revista Electrónica	Revista Investigium Ire	Javier Jiménez César Collazos Julio Hurtado Libardo Pantoja	Vol.VI No. 2 pp.1-158 Julio - Diciembre 2015 ISSN Electrónico: 2357-5239 DOI: http://dx.doi.org/10.15658/CESMAG15.05060103
3	Aprendizaje colaborativo con técnicas de realidad aumentada como estrategia didáctica para la enseñanza de estructuras iterativas en fundamentos de programación de computadores (Anexo 24)	Artículo en revista de divulgación nacional	Revista I+T+C Investigación, Tecnología y Ciencia	Javier Jiménez César Collazos Julio Hurtado Libardo Pantoja	ISSN: 1909-5775

Tabla 5.1 Ponencias en Congresos Internacionales

No	Título	Nombre	Autores	Documento
1	Recomendaciones para la construcción de entornos inmersivos 3d utilizando conceptos de Human Computer Interaction (Anexo 25)	Congreso Internacional de computación Colombia – México CICOM 2015	Javier Jiménez César Collazos Julio Hurtado Libardo Pantoja	ISSN: 2462-9588 Página Inicial: 497 Página Final: 502 Certificado de Ponencia
2	HCI y Diseño Basado en el Usuario para la construcción de entornos inmersivos 3D minería de datos (Anexo 26)	Congreso Andino de Computación, Informática y Educación CACIED 2015	Javier Jiménez César Collazos Julio Hurtado Libardo Pantoja	Certificado de Ponencia en Poster

REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Vidal, *Zerintia. Innovando con Madrid tour. Revista de turismo y patrimonio cultural*, 1 (2) ed. .
- [2] G. Hernández, R. Jiménez, and Á. Martínez, “Creencias docentes sobre la importancia de la didáctica en la orientación de la enseñanza del primer curso de programación de computadoras,” *Rev. Univ. docencia, Investig. e innovación*, vol. 2, pp. 87–103.
- [3] M. Oviedo, “La enseñanza de la programación,” *UPIICSA, México*, 2002. [Online]. Available: <http://bibliotecadigital.conevyt.org.mx/colecciones/documentos/somece2002/Grupo4/Oviedo.pdf>. [Accessed: 21-May-2014].
- [4] G. Bett, J. Fernández, V. Frittelli, D. Serrano, F. Steffolani, A. Strub, and R. Teicher, “Desarrollo de Juegos como Estrategia Didáctica en la Enseñanza de la Programación,” *Universidad Tecnológica Nacional, Facultad Regional Córdoba.*, 2013. [Online]. Available: <http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/4/120-429-1-DR.pdf>.
- [5] M. Murillo, “Exploring the teaching-learning process,” *Actual. Investig. en Educ. Explor.*, vol. 6, pp. 1–28.
- [6] A. F. Szpiniak and G. A. Rojo, “Enseñanza de la programación,” 2007.
- [7] R. Mosser, “In anual In Annual Joint Conference Integrating Technology into Computer Science Education ACM Press New York,” *NY, USA*, pp. 114–116, 1997.
- [8] N. Truong, P. Bancroft, and P. Roe, “A Web Based Environment for Learning to Program,” vol. 16, 2003.
- [9] T. Affleck, G. and Smith, “Identifying a need for web - based course support,” 1999.
- [10] J. Jiménez, C. Collazos, J. Hurtado, and W. Pantoja, “Estrategia colaborativa en entornos tridimensionales como estrategia didáctica de aprendizaje de estructuras iterativas en programación computacional,” *INVESTIGIUM IRE Ciencias Soc. y Humanas*, vol. VI, no. 2, pp. 80–92, 2015.
- [11] M. González De Rivera Fuente and M. Paredes Velasco, “Aprendizaje con programación Colaborativa,” pp. 1–33, 2008.
- [12] M. Ben-Ari, “Journal of Computers in Mathematics & Science Teaching,” vol. 20, pp. 24–73, 2001.
- [13] B. . Joyce and M. Weil, *Modelos de enseñanza*. Madrid: Anya, 1985.
- [14] L. S. Shulman, “Those who understand: knowledge growth in teaching,” *Educ. Res.*, vol. 15, no. 2, pp. 4–14, 1986.
- [15] Y. Chevallard, *Del sabio al saber enseñado*. Buenos Aires: Aiqué, 1991.
- [16] R. Porlan, *Constructivismo y escuela. Hacia un modelo de enseñanza aprendizaje basado en la investigación*. Sevilla: Diada Editora, 1993.
- [17] R. Porlan, *Qué y cómo enseñar desde una perspectiva constructivista. En Mariño, G. (Comp.)*. Bogotá: Dimensión Educativa, 1995.
- [18] G. Sacristán and G. Pérez, *Comprender y transformar la enseñanza*, 3rd ed. Madrid: Morata, 1994.
- [19] P. García and F. Angulo, “Un modelo didáctico para la formación inicial del profesorado de ciencias,” *Rev. Interuniv. Form. Profr.*, vol. 17, no. 001, pp. 37–49, 2003.
- [20] T. Rodríguez, *Teorías y modelos de enseñanza: posibilidades y límites*. Milenio Lleida, 1999.
- [21] C. Chaves and M. M. Rosero, “Procesos Metacognitivos En Programación De Teaching

- Approach and Its Relationship With Meta-Cognitive,” *Rev. Educ. en Ing.*, vol. 9, pp. 1–12, 2014.
- [22] N. Moroni and S. Perla, “Estrategias para la enseñanza de la programación,” *Prim. Jornadas en Educ. en Informática y TICS. JEITICS.*, 2005.
- [23] S. Xinogalos, M. Sartatzemi, and Dagdilelis, Vassilios Evangelidis, “Proceedings of the Sixth,” *Int. Conf. Adv. Learn. Technol. (ICALT’06). IEEE*, 2006.
- [24] J. A. Villalobos Salcedo, “Proyecto Cupi2 – una solución integral al problema de enseñar y aprender a programar 10°,” pp. 1–37, 2009.
- [25] G. Lavigne, M. P. Vasconcelos Ovando, J. O. Sandoval, and L. M. Salas, “Exploración preliminar del aprendizaje colaborativo dentro un entorno virtual,” vol. 12, no. 3, pp. 1–20, 2012.
- [26] C. Gosden, *Social Being and Time*. Oxford: Blackwell, 1994.
- [27] N. I. Scagnoli, “Estrategias para motivar el aprendizaje colaborativo en cursos a distancia,” *Investig. Cienc.*, vol. 14, no. 36, pp. 39–47, 2005.
- [28] J. Dewey, *Cómo pensamos: Nueva exposición de la relación entre pensamiento y proceso educativo*. Madrid: Paidós, 1989.
- [29] P. Ríos, “La mediación del aprendizaje,” in *Cuadernos UCAB (1) Lev Vygotsky, aportes para el siglo XXI*, Caracas: Publicaciones UCAB., 2005, pp. 34–40.
- [30] D. W. Jonhson, R. Jonhson, and E. Holubec, “Circles of learning (4th ed.),” 1993.
- [31] K. A. Bruffee, *Collaborative Learning Higher Education, Interdependence and the Authority of Knowledge*. Baltimore: The Johns Hopkins University Press, Second Edition, 1999.
- [32] M. Muhlenbrock, “A system for analyzing collaborative problem solving,” 1999. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.25.8372>.
- [33] P. Dillenbourg, M. Baker, A. Blaye, and C. Malley, “The evolution of research on collaborative learning,” in *E. Spada & P Reiman (Eds) Learning in Humans Machine: Towards an interdisciplinary learning science*, Oxford, 1996, pp. 189–211.
- [34] J. Salinas, *El aprendizaje colaborativo con los nuevos canales de comunicación.* Madrid: Síntesis, 2000.
- [35] P. E. Férez Glinz, “Un acercamiento al trabajo colaborativo,” *Rev. Iberoam. Educ.*, vol. s.n., pp. 1–14, 2005.
- [36] D. Leidner and S. Jarvenpaa, “The use of information technology to enhance management school education: a theoretical view,” *MIS Q.*, pp. 265–291.
- [37] S. Barab, M. Thomas, and H. Merrill, “Online Learning: From Information Dissemination to Fostering Collaboration,” *J. Interact. Learn. Res.*, vol. 12 (1), pp. 105–143, 2001.
- [38] I. T. Monterrey, “Aprendizaje Colaborativo, técnicas didácticas,” *Programa de Desarrollo de Habilidades Docentes*, 2008.
- [39] C. Collazos, L. Guerrero, and A. Vergara, “Aprendizaje Colaborativo: un cambio en el rol del profesor,” 2012.
- [40] D. W. Johnson and R. T. Johnson, *Cooperation and competition: Theory and Research*. Edina. MN: Interaction Book Co., 1989.
- [41] J. M. Serrano and M. T. Calvo, *Aprendizaje cooperativo. Técnicas y análisis dimensional*. Murcia: Obra Cultural de Cajamurcia, 1994.
- [42] W. Y. Hsu, “Online education on campus: A technological frames perspective on the process of technology appropriation,” London, University of London, 2002.
- [43] A. Escribano González, “Aprendizaje cooperativo y autónomo en la enseñanza universitaria,” *Enseñanza Teach. Rev. Interuniv. didáctica*, vol. 13, pp. 89–104, 1995.
- [44] T. S. Roberts, “Computer-supported collaborative learning in higher education: An introduction,” *Comput. Collab. Learn. High. Educ.*, no. Hershey: Idean Group Publishing,

pp. 1–18, 2005.

- [45] D. W. Johnson and F. P. Johnson, “Joining Together: Group Theory and Group Skills,” in *Needham Heights, MA: Allyn & Bacon*, 1997.
- [46] J. Cabero Almenara and A. Pérez, “Técnicas de trabajo colaborativo,” 2012. [Online]. Available: <http://tecnologiaedu.us.es/mec2005/html/cursos/fina/3-3.htm>. [Accessed: 03-May-2016].
- [47] C. Collazos and J. Mendoza, “Cómo aprovechar el ‘aprendizaje colaborativo’ en el aula,” *Educ. y Educ.*, vol. 9, no. 2, pp. 61–76, 2006.
- [48] A. King, “From sage on the satge to guide on the side,” *Coll. Teach.*, vol. 41, no. 1, pp. 30–35, 1993.
- [49] S. Kagan, *Cooperative learning*. San Juan Capistrano, CA: Kagan Cooperative Learning, 1992.
- [50] J. Lochhead and A. Whimbley, *Teaching analytical reasoning through thinking aloud pair problem solving*, vol. 30. San Francisco: Josswey-Bass, 1987.
- [51] E. Aronso, N. Blaney, C. Stephan, J. Sikes, and M. Snapp, *The Jigsaw Classroom*. Beverly Hills: Sage, 1978.
- [52] R. R. Amorim, E. Sánchez, M. Lama, S. Barro, and X. A. Vila, “Representación y ejecución de unidades educativas a través de una ontología de diseños de aprendizaje,” *Intel. Artif.*, vol. 11, no. 33, pp. 61–77, 2007.
- [53] M. V. Vargas, “Aprendizaje cooperativo en entornos virtuales: el método Jigsaw en asignaturas de estadística,” 2012.
- [54] C. . Selby and J. Woollard, “Computational Thinking: The Developing Definition.,” 2010.
- [55] L. Y. Gómez Giraldo, “Competencias M{í}nimas En Pensamiento Computacional Que Debe Tener Un Estudiante Aspirante a La Media T{é}cnica Para Mejorar Su Desempe{ñ}o En La Media T{é}cnica De Las Instituciones Educativas De La Alianza Futuro Digital Medell{í}n,” UNIVERSIDAD EAFIT, 2014.
- [56] J. M. Wing, “Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use,” *Commun. ACM*, vol. 49, No. 3, 2006.
- [57] J. Valverde Berrocoso, M. R. Fernández Sánchez, and M. del C. Garrido Arroyo, “El pensamiento computacional y las nuevas ecolog{í}as del aprendizaje,” *RED - Rev. Educ. a Distancia*, no. 46, pp. 1–18, 2015.
- [58] E. Espino and C. González, “Estudio sobre diferencias de género en las competencias y las estrategias educativas para el desarrollo del pensamiento computacional,” *Rev. Educ. a Distancia.*, no. 46, pp. 1–20, 2015.
- [59] J. A. Hurtado, C. A. Collazos, S. T. Cruz, and O. E. Rojas, “Child Programming: una estrategia de aprendizaje y construccion de Software basada en la l{ú}dica, la colaboracion y la agilidad,” *Rev. Univ. RUDIC*, vol. 1, no. 1, 2012.
- [60] MIT, “Scratch.” p. <http://wiki.scratch.mit.edu>, 2008.
- [61] U. C. Mellon, “Alice,” <http://alice.org/index.php>. 2003.
- [62] P. Barcelo, “Editorial,” *Bits Cienc.*, vol. 1, no. 1, 2015.
- [63] J. L. Zapotecatl, “Pensamiento Computacional,” in *Instituto Nacional de Astrofísica, Óptica y Electrónica*, 1st ed., Purbla, México: Luis Enrique Erro, 2014.
- [64] K. Brennan and M. Resnick, “Nuevos marcos de referencia para estudiar y evaluar el desarrollo del pensamiento computacional,” 2012. [Online]. Available: <http://www.eduteka.org/pdfdir/EvaluarPensamientoComputacional.pdf>.
- [65] J. Moreno, “El pensamiento computacional,” *Proy. Programamos*, 2014.
- [66] M. Zapata Ros, “Pensamiento computacional: Una nueva alfabetizacion digital,” *RED. Rev. Educ. a Distancia*, no. 46, pp. 1–47, 2015.

- [67] E. J. Moreno, P. M. Vera, R. A. Rodríguez, D. A. Giulianelli, M. G. Dogliotti, and G. Cruzado, "El Trabajo Colaborativo como Estrategia para Mejorar el Proceso de Enseñanza-Aprendizaje – Aplicado a la Enseñanza Inicial de Programación en el Ambiente Universitario," *Univ. Nac. La Matanza Dep. Ing. e Investig. Tecnológicas GIDFIS – Grup. Investig. Desarro. y Form. en Innovación Softw.*, 2013.
- [68] C. Fracchia, N. Baeza, and A. Martins, "ECDIA: Entorno Colaborativo para el Diseño e Implementación de Algoritmos," 2011.
- [69] D. S. Araúz, "Una Metodología para el aprendizaje de la Programación Orientada a Objetos basada en Programación Extrema (XP) ' A Methodology for learning Object Oriented Programming based on Extreme Programming (XP)," *Univ. Tecnológica Panamá*, 2010.
- [70] J. C. Rodríguez del Pino, E. Rubio Royo, and Z. J. Hernández Figueroa, "VPL: laboratorio virtual de programación para Moodle," *Dep. Informática y Sist. Univ. Las Palmas Gran Canar.*, 2009.
- [71] J. A. Jiménez, M. A. Pavony Meneses, A. F. Álvarez, and Serna, "enseñanza de algoritmos y programación en ingeniería Integration environment of PBL and CSCL for teaching algorithms and programming in engineering," *Rev. Av. en Sist. e Informática*, vol. 5, no. 3, pp. 189–194, 2008.
- [72] MIT, "Scratch - Imagine, Program, Share," 2008. [Online]. Available: <http://scratch.mit.edu/>. [Accessed: 22-Apr-2014].
- [73] N. C. Colmenar, "ENTORNOS PARA ENSEÑAR PROGRAMACIÓN EN SECUNDARIA. NUEVOS ENFOQUES. Nieves Carralero Colmenar. IES Pedro Mercedes. Junta de Comunidades de Castilla-La Mancha. España.," pp. 1–13, 2011.
- [74] D. Rodriguez Losada, M. Hernando, R. Gonzales, C. Platero, P. L. Castedo, L. Davila, P. . San Segundo, C. Santos, and S. López, "Enseñanza de programación orientada a objetos mediante el desarrollo de aplicaciones graficas interactivas," *Grup. Sist. Telemáticos Apl. a la Educ.*, 2008.
- [75] E. De Graff and A. Kolmos, "Management of Change. Implementation of Problem-Based and Projec-Based learning in Engineering," *Sense Publ.*, 2007.
- [76] R. Pérez Calderón, "Una Herramienta y Técnica p ara la E nseñanza de la P rogramación," 2008.
- [77] P. E. T. G. University of Kent in Canterbury, "Greenfoot," 2008. [Online]. Available: <http://www.greenfoot.org/door>. [Accessed: 22-Apr-2014].
- [78] N. Myller and J. Nuutinen, "JeCo : Combining Program Visualization and Story Weaving," vol. 5, no. 2, pp. 267–276, 2006.
- [79] A. J. Mendes, "ProLEARN , A PLATFORM TO SUPPORT PROGRAMMING," 2006.
- [80] L. Concepts, A. Diwan, W. M. Waite, and M. H. Jackson, "PL-Detective : A System for Teaching Programming," 2005.
- [81] M. Á. Redondo, "Aprendizaje en grupo de la programación mediante técnicas de colaboración distribuida en tiempo real," pp. 351–357, 2004.
- [82] M. Esteves and a. J. Mendes, "A simulation tool to help learning of object oriented programming basics," *34th Annu. Front. Educ. 2004. FIE 2004.*, pp. 811–816, 2004.
- [83] C. Jo and A. J. Arnold, "A Portable and Collaborative Distributed Programming Environment The architecture of DPE," 2003.
- [84] M. Kölling, B. Quig, A. Patterson, and J. Rosenberg, "The BlueJ system and its pedagogy 1," vol. 13, no. 4, pp. 1–12, 2003.
- [85] U. University Carnegie Mellon, "Alice.org," 2003. [Online]. Available: http://www.alice.org/index.php?page=what_is_alice/what_is_alice. [Accessed: 22-Apr-2014].

- [86] A. Vizcaíno, J. Contreras, J. Favela, and M. Prieto, "An Adaptive , Collaborative Environment to Develop Good Habits in Programming," pp. 262–271, 2002.
- [87] R. S. Cuerda, F. Llorens, J. Antonio, and P. García, "Enseñar Programación en las Ingenierías," 1999.
- [88] H. Suzuki and H. Kato, "Identity formation / transformation as the process of collaborative learning through AlgoArena," *CSCL '97 Proc.*, pp. 280–289.
- [89] W. S. Humphrey, *Introduction to the Personal Software Process*, Addison We. 1997.
- [90] U. Universidad Autónoma de Barcelona, "Métodos de aprendizaje colaborativos," 2014. [Online]. Available: <http://grupsderecerca.uab.cat/grai/es/content/m%C3%A9todos-de-aprendizaje-cooperativo>.
- [91] T. Wang, "The Comparison of the Difficulties between Cooperative Learning and Traditional Teaching Methods in College English Teachers," *J. Hum. Resour. Adult Learn.*, vol. 3, no. 2, pp. 123–456, 2007.
- [92] A. Gocer, "A comparative research on the effectivity of cooperative learning method and jigsaw technique on teaching literary genres," *Educ. Res. Rev.*, vol. 5, pp. 439–445, 2010.
- [93] M. Quiao and X. Jin, "Jigsaw strategy as a cooperative learning technique: focusing on the language learners," *Chinese J. Appl. Linguist.*, vol. 33, no. 4, pp. 123–456, 2010.
- [94] D. Johnson and R. Johnson, *Learning together and alone. alone*, Englewood Cliffs, NJ.: Prentice Hall, 1987.
- [95] D. Dishon and P. W. O'leary, *A guidebook for cooperative learning: A technique for creating more effective schools*. Holmes Beach, Fl: Learning Publications, 984.
- [96] T. Bailey and K. Lundgaard, *Program design with pseudocode*. Estados Unidos: Brooks/Cole Pub Co, 1989.
- [97] R. A. Sanchez Turcios, "T-Student, usos y abusos," *Rev. Mex. Cardiol.*, vol. 26, no. 1, pp. 59–61, 2015.

ANEXOS



Anexo 1

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato A: Configuración inicial

Actividad	Definición	Actividades recomendadas
Tarea grupal		
Tiempo estimado para la realización de la tarea		
Recursos/Materiales		
Formar los grupos		
Distribución física		



Anexo 2

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

<p>Formato B: Configuración inicial para fundamentación conceptual de modelado gráfico</p>				
Identificación del curso				
Nombre del curso				
Área		Componente		Créditos
Código del curso		Semestre		Jornada
Identificación del docente				
Nombre del docente				
E-mail				
Descripción del curso				
Competencia central				
Unidades y elementos de competencia				
Unidad de competencia	Elementos de competencia			Horas
Recursos y/o materiales				
Distribución en el aula de clase				



Anexo 3

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato C: Planeación de ejemplos de la configuración inicial para fundamentación conceptual de modelado gráfico

Unidad de competencia	Enunciados del ejemplo



Anexo 4

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato D: Recolección de requerimiento principal	
Nombre del sistema	Requerimiento principal



Anexo 5

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

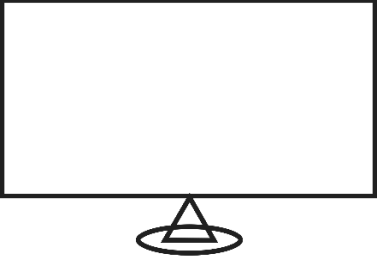


Formato E: Diagrama Entrada/Salida (D E/S)	



Anexo 6

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato F: Interfaz Gráfica de Usuario (GUI)	
Símbolo	Descripción
	Pantalla: corresponde al dispositivo donde se realizará la impresión de los resultados tanto de entradas y salidas en una solución de software.
	Entrada: Permite el ingreso de datos al sistema
	Salida: Permite la impresión de datos a través de la pantalla



Anexo 7

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato G: Diagrama de Flujo (DF)	
Símbolo	Descripción
	Inicio/fin. Indica donde comienza y termina un DF. Todo DF tiene un solo inicio y un único fin
	Flujo. Determina hacia donde se dirigen los datos
	Entrada. Permite capturar datos
	Salida. Permite la visualización de datos al usuario
	Proceso. Utilizado para realizar acciones las cuales generalmente son de tipo aritmético y se realizan mediante la operación de asignación
	Decisión. Permite evaluar una o más proposiciones. Como máximo tiene dos alternativas de solución las cuales son excluyentes entre sí.
	Ciclo. Utilizado para procesar instrucciones de forma repetitiva o iterativa



Anexo 8

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato H: Configuración inicial para el caso de estudio

Identificación del curso				
Nombre del curso				
Código del curso		Semestre		Jornada
Identificación del docente				
Nombre del docente				
E-mail				



Anexo 9

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato I: Planeación de ejercicios de configuración inicial para el caso de estudio

Unidad de competencia	de	Enunciados del ejemplo
Unidad 1:		
Unidad 2:		
Unidad 3:		
Unidad 4:		



Anexo 10

Universidad del Cauca

Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato H: configuración inicial para el caso de estudio: Fundamentos de Programación			
Identificación del curso			
Nombre del curso	Fundamentos de programación		
Código del curso	232	Semestre	1
		Jornada	Diurna
Identificación del docente			
Nombre del docente	Javier Alejandro Jiménez Toledo		
E-mail	jajimenez@iucesmag.edu.co		
Unidad de competencia	Unidad 1: Captura y Salida de datos		
Fase	Análisis		
Tiempo de trabajo	Unidad	Elemento de competencia	Horas
	Entradas y salidas	Entradas y Salidas	2
Objetivo general	Una vez expuesta por parte del docente la temática de Entradas y salidas utilizando los diagramas de entrada/salida (D E/S) y la interfaz gráfica de usuarios (GUI), el estudiante la apropiará a través del desarrollo de ejercicios mediante la estrategia didáctica JIGSAW		
Objetivo individual	Apropiar el proceso de entradas y salidas de la fase de análisis		
Objetivo grupal	Compartir experiencias individuales con todo el grupo de trabajo en el estudio de entradas y salidas		
Conformación de grupos	2 estudiantes tomados al azar, uno para el rol de analista de entradas/salidas y otro para constructor de interfaz gráfica de usuario		
Distribución en el aula de clase	Se debe distribuir a los estudiantes inicialmente en grupos de dos, ordenadamente en todo el salón de clase con sus respectivas sillas universitarias		
Planeación de recursos	Cada estudiante debe tener únicamente su cuaderno de apuntes y sus lapiceros		
Unidad de competencia	Unidad 1: Captura y Salida de datos		
Fase	Diseño		
Tiempo de trabajo	Unidad de competencia	Elemento de competencia	Horas
	Captura y salida de datos	Captura y Salida de datos	4
Objetivo general	Una vez expuesta por parte del docente la temática de Entradas y salidas utilizando algoritmos, el estudiante la apropiará a través del desarrollo de ejercicios mediante la estrategia didáctica JIGSAW		
Objetivo individual	Apropiar el proceso de entradas y salidas de la fase de diseño		
Objetivo grupal	Compartir experiencias individuales con todo el grupo de trabajo en el estudio de entradas y salidas en su fase de diseño		

Conformación de grupos	2 estudiantes tomados al azar, uno para el rol de analista de entradas/salidas e interfaz gráfica de usuario y otro diseñador del algoritmo
Distribución en el aula de clase	Se debe distribuir a los estudiantes inicialmente en grupos de dos, ordenadamente en todo el salón de clase con sus respectivas sillas universitarias
Planeación de recursos	Cada estudiante debe tener únicamente su cuaderno de apuntes y sus lapiceros.

Unidad de competencia	Unidad 2: Estructuras condicionales		
Fase	Análisis		
Tiempo de trabajo	Unidad	Elemento de competencia	Horas
	Estructuras condicionales	Condicionales simples y compuestos	2
		Condicionales anidados	2
		Condicionales y operadores lógicos	2
Objetivo general	Una vez expuesta por parte del docente la temática de estructuras condicionales utilizando los diagramas de entrada/salida (D E/S) y la interfaz gráfica de usuarios (GUI), el estudiante la apropiará a través del desarrollo de ejercicios mediante la estrategia didáctica JIGSAW		
Objetivo individual	Apropiar la estructura condicional en la fase de análisis		
Objetivo grupal	Compartir experiencias individuales con todo el grupo de trabajo en el estudio de condicionales		
Conformación de grupos	2 estudiantes tomados al azar, uno para el rol de analista de entradas/salidas y otro para constructor de interfaz gráfica de usuario		
Distribución en el aula de clase	Se debe distribuir a los estudiantes inicialmente en grupos de dos, ordenadamente en todo el salón de clase con sus respectivas sillas universitarias		
Planeación de recursos	Cada estudiante debe tener únicamente su cuaderno de apuntes y sus lapiceros		

Unidad de competencia	Unidad 2: Estructuras condicionales		
Fase	Diseño		
Tiempo de trabajo	Unidad	Elemento de competencia	Horas
	Estructuras condicionales	Condicionales simples y compuestos	4
		Condicionales anidados	4
		Condicionales y operadores lógicos	4
Objetivo general	Una vez expuesta por parte del docente la temática de estructuras condicionales utilizando algoritmos, el estudiante la apropiará a través del desarrollo de ejercicios mediante la estrategia didáctica JIGSAW		
Objetivo individual	Apropiar la estructura condicional en la fase de diseño		
Objetivo grupal	Compartir experiencias individuales con todo el grupo de trabajo en el estudio de condicionales		
Conformación de grupos	2 estudiantes tomados al azar, uno para el rol de analista de estructuras condicionales y otro diseñador del algoritmo		
Distribución en el aula de clase	Se debe distribuir a los estudiantes inicialmente en grupos de dos, ordenadamente en todo el salón de clase con sus respectivas sillas universitarias		

Planeación de recursos	Cada estudiante debe tener únicamente su cuaderno de apuntes y sus lapiceros
-------------------------------	--

Unidad de competencia	Unidad 3: Estructuras iterativas		
Fase	Análisis		
Tiempo de trabajo	Unidad	Elemento de competencia	Horas
	Estructuras iterativas	Ciclo para	2
		Ciclo mientras	2
		Ciclo hacer mientras	2
Objetivo general	Una vez expuesta por parte del docente la temática de estructuras iterativas utilizando los diagramas de entrada/salida (D E/S) y la interfaz gráfica de usuarios (GUI), el estudiante la apropiará a través del desarrollo de ejercicios mediante la estrategia didáctica JIGSAW		
Objetivo individual	Apropiar las estructuras iterativas en su fase de análisis		
Objetivo grupal	Compartir experiencias individuales con todo el grupo de trabajo en el estudio de estructuras iterativas		
Conformación de grupos	2 estudiantes tomados al azar, uno para el rol de analista de entradas/salidas y otro para constructor de interfaz gráfica de usuario		
Distribución en el aula de clase	Se debe distribuir a los estudiantes inicialmente en grupos de dos, ordenadamente en todo el salón de clase con sus respectivas sillas universitarias		
Planeación de recursos	Cada estudiante debe tener únicamente su cuaderno de apuntes y sus lapiceros		

Unidad de competencia	Unidad 3: Estructuras iterativas		
Fase	Diseño		
Tiempo de trabajo	Unidad	Elemento de competencia	Horas
	Estructuras iterativas	Estructura Para	4
		Estructura Mientras	4
		Estructura Hacer mientras	4
Objetivo general	Una vez expuesta por parte del docente la temática de estructuras iterativas utilizando algoritmos, el estudiante la apropiará a través del desarrollo de ejercicios mediante la estrategia didáctica JIGSAW		
Objetivo individual	Apropiar las estructuras iterativas en la fase de diseño		
Objetivo grupal	Compartir experiencias individuales con todo el grupo de trabajo en el estudio de condicionales		
Conformación de grupos	2 estudiantes tomados al azar, uno para el rol de analista de estructuras iterativas y otro diseñador del algoritmo		
Distribución en el aula de clase	Se debe distribuir a los estudiantes inicialmente en grupos de dos, ordenadamente en todo el salón de clase con sus respectivas sillas universitarias		
Planeación de recursos	Cada estudiante debe tener únicamente su cuaderno de apuntes y sus lapiceros		



Anexo 11

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato I: planeación de ejercicios de configuración inicial para el caso de estudio: Fundamentos de Programación	
Unidad de competencia	Enunciados del ejemplo
Unidad 1: captura y Salida de datos	<ol style="list-style-type: none"> 1. Diseñar una solución computacional que calcule el área de un cuadrado en función de sus lados (área=lado*lado) 2. Diseñar una solución computacional que calcule el área de un círculo en función de su radio (área=3.1416*r*r) 3. Diseñar una solución computacional que capture una medida en kilómetros y la convierta a metros. 4. Diseñar una solución computacional que capture la distancia recorrida de un objeto y el tiempo empleado para recorrerla y calcule su velocidad promedio ($X = V*T$) 5. Diseñar una solución computacional que calcule el área de un rectángulo en función de sus lados (área=lado menor*lado mayor) 6. Diseñar una solución computacional que capture dos números e imprima las cuatro operaciones básicas: Suma, resta, multiplicación y división 7. Diseñar una solución computacional que convierta de Gigabytes a Bytes 8. Diseñar una solución computacional que convierta de pesos dólares solicitando el valor de la tasa de cambio 9. Diseñar una solución computacional que capture para un artículo de papelería su valor unitario y el número de unidades a comprar. Calcular total de la compra 10. Diseñar una solución computacional que capture para un empleado las horas trabajadas mensuales (ht) y el valor de la hora pagada (vh). Con estos datos se calcula el sueldo básico del empleado (sb) así: $sb=ht*vh$. Todo empleado debe aportar al sistema de seguridad social (ss) el 8% del sueldo básico así: $ss=sb*0.08$. Además, todo empleado tendrá una bonificación (bon) del 10% del sueldo básico donde $bon=sb*0,1$. Calcule e imprima el sueldo neto a pagar (sn) así: $sn=sb-ss+bon$
Unidad 2. Estructura condicional	<p>Condicionales compuestos</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture dos números e imprima el mayor de ellos. 2. Diseñar una solución computacional que capture dos números y determine si son iguales o diferentes 3. Diseñar una solución computacional que capture un número e imprima su valor absoluto 4. Diseñar una solución computacional que capture una hora en formato militar y la convierta a formato am/pm 5. Diseñar una solución computacional que capture para un artículo de papelería su valor de venta, si dicho valor es mayor a \$100.000 obtendrá un descuento del 20% de lo

**Formato I: planeación de ejercicios de configuración inicial para el caso de estudio:
Fundamentos de Programación**

Unidad de competencia	Enunciados del ejemplo
	<p>contrario su descuento será del 10%. Calcular el valor del descuento realizado</p> <ol style="list-style-type: none"> 6. Diseñar una solución computacional que capture para un artículo de papelería su valor de venta, si dicho valor es mayor a \$100.000 obtendrá un descuento del 20% de lo contrario su descuento será del 10%. Calcular el valor de la venta realizada 7. Diseñar una solución computacional que capture un número y determine si es par o impar 8. Diseñar una solución computacional que capture dos números y determine si el primer número es múltiplo del segundo 9. Diseñar una solución computacional que capture dos números, si la suma de ellos es positiva deberá imprimir la suma de sus cuadrados, de lo contrario deberá imprimir el producto de sus cubos. 10. Diseñar una solución computacional que capture para un empleado las horas trabajadas mensuales (ht) y el valor de la hora pagada (vh). Con estos datos se calcula el sueldo básico del empleado (sb) así: $sb=ht*vh$. Todo empleado debe aportar al sistema de seguridad social (ss) el 8% del sueldo básico así: $ss=sb*0.08$. Además, si el sueldo básico del empleado es menor o igual a 1 SMLV tendrá una bonificación (bon) del 10% del sueldo básico de lo contrario su bonificación será del 1%. Calcule e imprima el sueldo neto a pagar (sn) así: $sn=sb-ss+bon$ <p>Condicionales simple</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture para un PC su valor de venta, si dicho valor es menor a \$2.000.000 no pagará impuesto de importación de lo contrario gravará un 16% sobre su valor. Calcular el valor del impuesto a pagar. 2. Diseñar una solución computacional que capture para un PC su valor de venta, si dicho valor es menor a \$2.000.000 no pagará impuesto de importación de lo contrario gravará un 16% sobre su valor. Calcular el valor del PC. 3. Diseñar una solución computacional que capture un número e imprima su valor absoluto 4. Diseñar una solución computacional que capture la nota definitiva para un estudiante de Introducción a la Ingeniería, si dicha nota es mayor a 4.5 se registrará 5.0 como su nota definitiva. Calcular la nota final. 5. Diseñar una solución computacional que capture para un artículo de papelería su valor de venta, si dicho valor es mayor a \$20.000 obtendrá un descuento del 10%. Calcular el valor de la venta. <p>Condicionales anidados</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture un número y determine si es par, impar o neutro. 2. Diseñar una solución computacional que capture dos números y determine si son iguales o cuál es el menor de ellos.

**Formato I: planeación de ejercicios de configuración inicial para el caso de estudio:
Fundamentos de Programación**

Unidad de competencia	Enunciados del ejemplo
	<ol style="list-style-type: none"> 3. Diseñar una solución computacional que capture dos números y determine si son iguales o cuál es el menor de ellos y el mayor. 4. Diseñar una solución computacional que capture tres números y determine si son iguales o cuál es el menor de ellos. 5. Diseñar una solución computacional que capture tres números y determine si son iguales o cuál es el menor de ellos y el mayor. 6. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(-50 \leq n \leq 100)$ y $(n > 500)$ 7. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(n < -300)$ y $(-10 < n \leq 100)$ y $(360 \leq n \leq 500)$ 8. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(n < -35)$ y $(80 \leq n < 800)$ y $(2000 < n < 2500)$ y $(n \geq 3570)$ 9. Diseñar una solución computacional que capture un número y determine si se encuentra en el rango A,B o en ninguno de ellos: $(-300 < A \leq -35)$ y $(-20 \leq B \leq 40)$ y $(200 \leq B < 750)$ y $(900 < A < 1350)$ y $(B \geq 1580)$ 10. Diseñar una solución computacional que capture un número y determine si se encuentra en el rango A,B,C o en ninguno de ellos: $(C < -1000)$ y $(-700 < B \leq -200)$ y $(-20 < C \leq 20)$ y $(87 \leq A < 175)$ y $(400 < B \leq 750)$ y $(1200 \leq A < 750)$ y $(C \geq 1000)$ <p>Condicionales y operadores lógicos</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture dos números y determine si son iguales o cuál es el menor de ellos. 2. Diseñar una solución computacional que capture dos números y determine si son iguales o cuál es el menor de ellos y el mayor. 3. Diseñar una solución computacional que capture tres números y determine si son iguales o cuál es el menor de ellos. 4. Diseñar una solución computacional que capture tres números y determine si son iguales o cuál es el menor de ellos y el mayor. 5. Diseñar una solución computacional que capture cuatro números y determine si son iguales o cuál es el menor de ellos y el mayor. 6. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(-50 \leq n \leq 100)$ y $(n > 500)$ 7. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(n < -300)$ y $(-10 < n \leq 100)$ y $(360 \leq n \leq 500)$ 8. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(n < -35)$ y $(80 \leq n < 800)$ y $(2000 < n < 2500)$ y $(n \geq 3570)$

Formato I: planeación de ejercicios de configuración inicial para el caso de estudio: Fundamentos de Programación	
Unidad de competencia	Enunciados del ejemplo
	<p>9. Diseñar una solución computacional que capture un número y determine si se encuentra en el rango A,B o en ninguno de ellos: $(-300 < A \leq -35)$ y $(-20 \leq B \leq 40)$ y $(200 \leq B < 750)$ y $(900 < A < 1350)$ y $(B \geq 1580)$</p> <p>10. Diseñar una solución computacional que capture un número y determine si se encuentra en el rango A,B,C o en ninguno de ellos: $(C < -1000)$ y $(-700 < B \leq -200)$ y $(-20 < C \leq 20)$ y $(87 \leq A < 175)$ y $(400 < B \leq 750)$ y $(1200 \leq A < 750)$ y $(C \geq 1000)$</p>
Unidad 3: Estructuras iterativas	<p>Ciclo Para, Mientras y Hacer Mientras ascendente</p> <p>1. Diseñar una solución computacional que imprima los 7 primeros números pares en forma ascendente</p> <p>2. Diseñar una solución computacional que capture un rango (Límite inferior y límite superior) e imprima los números encontrados en él en forma ascendente.</p> <p>3. Diseñar una solución computacional que capture un número e imprima sus diez primeros múltiplos en forma ascendente.</p> <p>4. Diseñar una solución computacional que capture un rango e imprima los números pares encontrados en forma ascendente.</p> <p>5. Diseñar una solución computacional que capture un rango y un valor e imprima los diez primeros múltiplos de dicho valor en forma ascendente.</p> <p>6. Diseñar una solución computacional que capture número (n) e imprima los n primeros múltiplos de dicho número en forma ascendente.</p> <p>7. Diseñar una solución computacional que capture para n estudiantes su nota definitiva del curso de Introducción a la ingeniería y por cada uno de ellos imprima si prueba o reprueba el curso.</p> <p>8. Diseñar una solución computacional que capture n valores y por cada uno de ellos determine si: es par o impar; positivo, negativo o neutro; imprimir su valor absoluto.</p> <p>9. Diseñar una solución computacional que capture n valores y por cada uno de ellos determine si se encuentra o no en el siguiente rango: $(n < -35)$ y $(80 \leq n < 800)$ y $(2000 < n < 2500)$ y $(n \geq 3570)$</p> <p>10. Diseñar una solución computacional que capture n valores y por cada uno de ellos determine si se encuentra en el rango A,B,C o en ninguno de ellos: $(C < -1000)$ y $(-700 < B \leq -200)$ y $(-20 < C \leq 20)$ y $(87 \leq A < 175)$ y $(400 < B \leq 750)$ y $(1200 \leq A < 750)$ y $(C \geq 1000)$</p> <p>Ciclo Para, Mientras y Hacer Mientras descendente</p> <p>1. Diseñar una solución computacional que imprima los 7 primeros números pares en forma descendente.</p> <p>2. Diseñar una solución computacional que capture un rango (Límite inferior y límite superior) e imprima los números encontrados en él en forma descendente.</p> <p>3. Diseñar una solución computacional que capture un número e imprima sus diez primeros múltiplos en forma descendente.</p>

**Formato I: planeación de ejercicios de configuración inicial para el caso de estudio:
Fundamentos de Programación**

Unidad de competencia	Enunciados del ejemplo
	<ol style="list-style-type: none"> 4. Diseñar una solución computacional que capture un rango e imprima los números pares encontrados en forma descendente. 5. Diseñar una solución computacional que capture un rango y un valor e imprima los diez primeros múltiplos de dicho valor en forma descendente. 6. Diseñar una solución computacional que capture número (n) e imprima los n primeros múltiplos de dicho número en forma descendente. 7. Diseñar una solución computacional que capture para n estudiantes su nota definitiva del curso de Introducción a la ingeniería y por cada uno de ellos imprima si prueba o reprueba el curso. 8. Diseñar una solución computacional que capture n valores y por cada uno de ellos determine si: es par o impar; positivo, negativo o neutro; imprimir su valor absoluto. 9. Diseñar una solución computacional que capture n valores y por cada uno de ellos determine si se encuentra o no en el siguiente rango: ($n < -35$) y ($80 \leq n < 800$) y ($2000 < n < 2500$) y ($n \geq 3570$) 10. Diseñar una solución computacional que capture n valores y por cada uno de ellos determine si se encuentra en el rango A, B, C o en ninguno de ellos: ($C < -1000$) y ($-700 < B \leq -200$) y ($-20 < C \leq 20$) y ($87 \leq A < 175$) y ($400 < B \leq 750$) y ($1200 \leq A < 750$) y ($C \geq 1000$).



Anexo 12

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato B: configuración inicial para fundamentación conceptual de modelado gráfico: Fundamentos de Programación					
Identificación del curso					
Nombre del curso		Fundamentos de programación			
Área	Formación específica	Componente	Programación	Créditos	3
Código del curso	232	Semestre	1	Jornada	Diurna
Identificación del docente					
Nombre del docente		Javier Alejandro Jiménez Toledo			
E-mail		jajimenez@iucsmag.edu.co			
Descripción del curso					
El curso de Introducción a la programación, pretende llevar al estudiante al aprendizaje de los conceptos básicos necesarios para adquirir los conocimientos y la lógica previa en la programación de computadoras, basado en el enfoque orientado a objetos; que le ayudarán a enfrentarse a la solución de casos problemáticos y de problemáticas reales en su entorno.					
Competencia central					
Modelar situaciones y problemas mediante los fundamentos esenciales de las técnicas y destrezas del análisis y diseño bajo el enfoque Orientado a Objetos con el fin de iniciar al estudiante en procesos de desarrollo de software.					
Unidades y elementos de competencia					
Unidad de competencia	Elementos de competencia				Horas
Generalidades	Motivación Paradigmas de programación				2
Captura y salida de datos	Motivación				2
	Diagrama entrada/salidas (D E/S)				1
	Interfaz gráfica de usuario (GUI)				2
	Análisis de problemas				2
Estructura condicional	Diseño de soluciones				2
	Motivación				2
	Condicional compuesto				1
	Condicional simple				2
Estructuras iterativas	Condicionales anidados				2
	Condicionales y Operadores Lógicos				2
	✓ Motivación				2
Estructuras iterativas	✓ Ciclo Para				2
	✓ Ciclo Mientras				2
	✓ Ciclo Hacer Mientras				1
	✓ Acumuladores				1

	✓ Contadores	1
Recursos y/o materiales		
<p>Para el desarrollo de las sesiones presenciales de clase es necesario que el estudiante cuente con su cuaderno de apuntes y una vez indicada la temática de estudio y realizado el proceso de aprendizaje en entorno colaborativo es necesario que el estudiante realice prácticas utilizando el software de modelado DFD</p>		
Distribución en el aula de clase		
<p>Para la configuración inicial de fundamentación conceptual de modelado gráfico es necesario que el docente disponga de un salón de clase en el que se desarrollarán todas las sesiones de trabajo para posteriormente realizar algunas sesiones prácticas sobre lo estudiado en aulas de informática.</p> <p>La distribución de los estudiantes en el salón de clase para esta configuración inicial poder ser en orden estricto de sillas en disposición de filas o a gusto del estudiante que en muchas ocasiones le da seguridad de trabajo.</p>		
Bibliografía		
<p>Booch, Grady.El lenguaje unificado de modelado / Grady Booch; James Rumbaugh, Ivar Jacobson ; traduccion, Jose Saez Martinez. Madrid : Addison Wesley, 1999.</p> <p>Rumbaugh, James.El lenguaje unificado de modelado : Manual de referencia / James Rumbaugh, Ivar Jacobson, Grady Booch. Madrid : Addison Wesley, 2000. 526 p</p> <p>Fowler, Martin.UML gota a gota / Martin Fowler con Kendall Scott ; traduccion Jaime Gonzalez V., David Morales Peake ; revision tecnica Gabriel Guerrero. Mexico : Pearson Educacion, 1999. 203 p</p> <p>Larman, Craig.UML y Patrones: Una introduccion al analisis y Diseño orientado a objetos y al proceso unificado / Craig Larman; traduccion Begona Moros Valle; supervision de la traduccion y revision tecnica Jesus Garica Molina. Madrid : Pearson, 2006. 590 p</p> <p>Joyanes Aguilar Luis. Fundamentos de programación. Algoritmos y estructuras de datos. Segunda Edición. Ed. Mc Graw Hill.</p> <p>Joyanes Aguilar Luis. Fundamentos de programación. Libro de problemas. Segunda Edición. Ed. Mc Graw Hill.</p>		



Anexo 13

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Formato C: planeación de ejemplos de la configuración inicial para fundamentación conceptual de modelado gráfico: Fundamentos de Programación	
Unidad de competencia	Enunciados del ejemplo
Unidad 1: Entradas y Salidas	<ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture un número e imprima su cuadrado. 2. Diseñar una solución computacional que calcule el área de un triángulo en función de su base y altura (área = base * altura/2)
Unidad 2. Estructura condicional	<p>Condicional compuesto</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture la nota definitiva para un estudiante de Introducción a la Ingeniería y determine si aprueba o reprueba el curso. 2. Diseñar una solución computacional que capture para una persona su edad y determine si es mayor o menor de edad. <p>Condicional simple</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture para un artículo de papelería su valor de venta, si dicho valor es mayor a \$20.000 obtendrá un descuento del 10%. Calcular el descuento realizado. 2. Diseñar una solución computacional que capture la nota definitiva para un estudiante de Introducción a la Ingeniería, si dicha nota es mayor a 4.5 el estudiante será acreedor a una bonificación de 2 décimas. Calcular la nota final. <p>Condicionales anidados</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture un número y determine si es positivo, negativo o neutro. 2. Diseñar una solución computacional que capture tres números y determine si son iguales o diferentes. 3. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(20 \leq n \leq 300)$ <p>Condicionales y operadores lógicos</p> <ol style="list-style-type: none"> 1. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(20 \leq n \leq 300)$ 2. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(200 \leq n \leq 1200)$ y $(n \geq 3000)$
Unidad 3: Estructuras iterativas	Ciclo Para, Mientras y Hacer Mientras ascendente

Formato C: planeación de ejemplos de la configuración inicial para fundamentación conceptual de modelado gráfico: Fundamentos de Programación

Unidad de competencia	Enunciados del ejemplo
	<ol style="list-style-type: none">1. Diseñar una solución computacional que imprima los siete primeros números en forma ascendente.2. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde 1 hasta dicho valor en forma ascendente <p>Ciclo Para, Mientras y Hacer Mientras descendente</p> <ol style="list-style-type: none">1. Diseñar una solución computacional que imprima los siete primeros números en forma descendente.2. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde 1 hasta dicho valor en forma descendente.



Anexo 14

Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Ejercicios complementarios

Captura y salida de datos

Ejemplo 1. Proponer una solución computacional que calcule el área de un triángulo en función de su base y altura (área = base * altura/2).

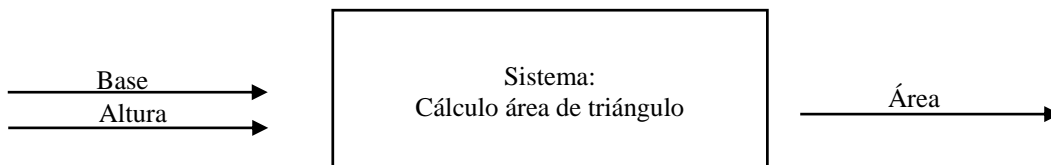
Solución

2. Fase de recolección de requerimientos

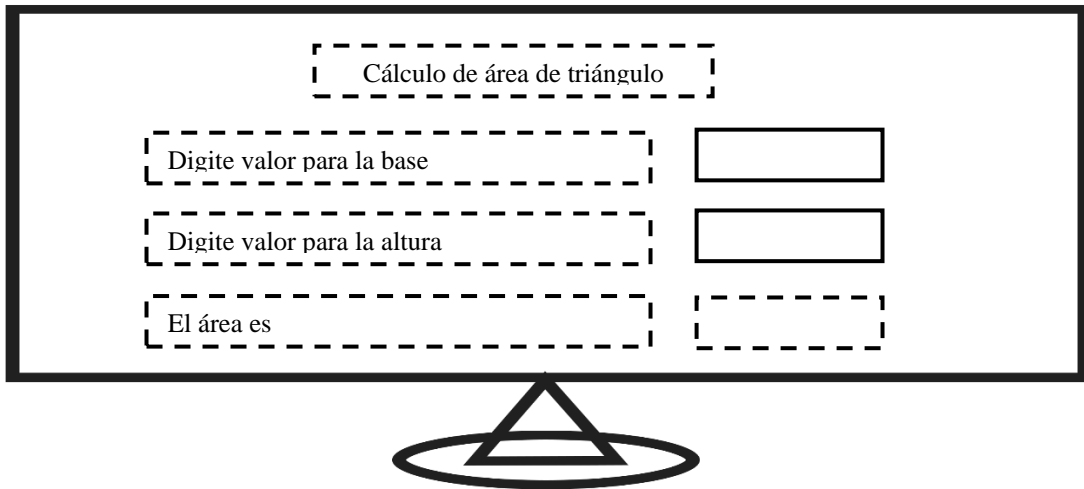
Nombre del sistema	Requerimiento principal
Cálculo área de triángulo	Capturar la base y altura de un triángulo para calcular su área

3. Fase de análisis

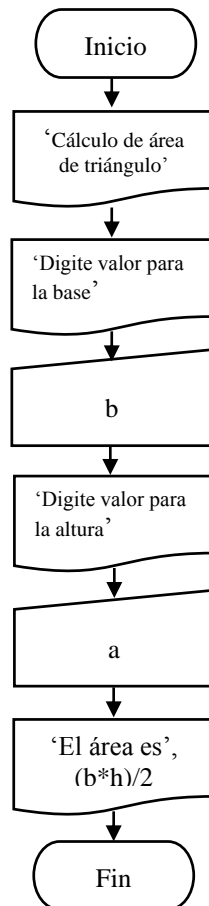
a. Diagrama Entrada/Salida (D E/S)



b. Interfaz Gráfica de Usuario (GUI)



4. Fase de diseño



Nota al docente. En estos primeros ejemplos no se utilizará "procesos" en la estructura del algoritmo, se utilizarán luego.

Ejemplo 2. Proponer una solución computacional que capture para un estudiante su nota total de los parciales realizados durante su semestre (70%) y la nota total de sus talleres (30%) y calcule la nota definitiva (parciales *0.7 + talleres * 0.3).

En este ejemplo se mostrará la forma como el estudiante debe realizar por completo un ejercicio con sus tres fases, sin olvidar recordar que la fase de requerimientos está en el enunciado del ejemplo y el docente deberá indagarlos sobre la comprensión del mismo (con esto se tendría a este nivel simbólicamente la fase de requerimientos cumplida) para poder avanzar a la fase de análisis y posteriormente a la de diseño.

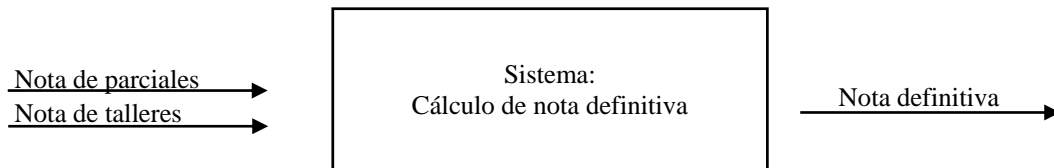
Solución

2. Fase de requerimiento

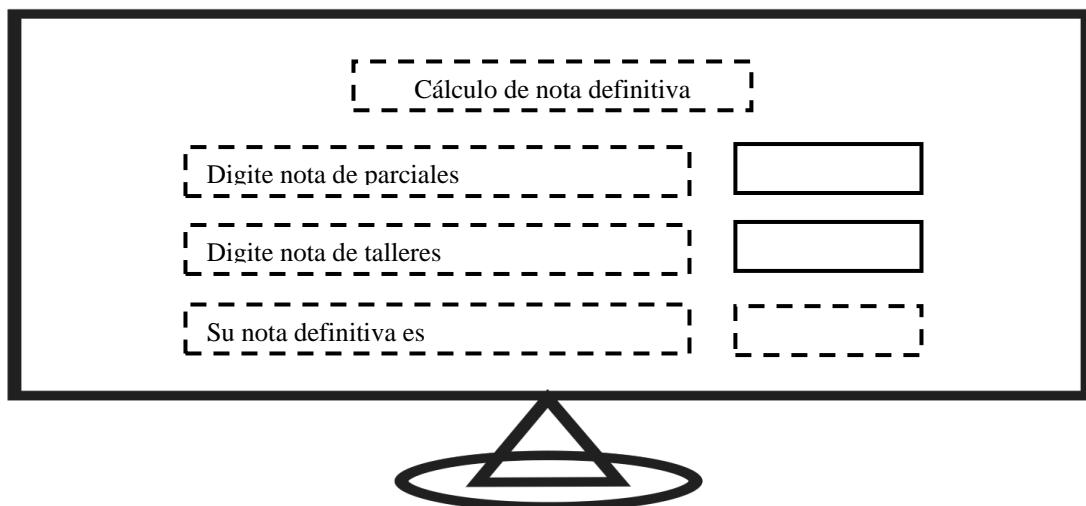
Nombre del sistema	Requerimiento principal
Calculo de Nota definitiva	Capturar para un estudiante su nota de parciales (70%) y talleres (30%) para calcular la nota definitiva (parciales *0.7 + talleres * 0.3)

3. Fase de análisis

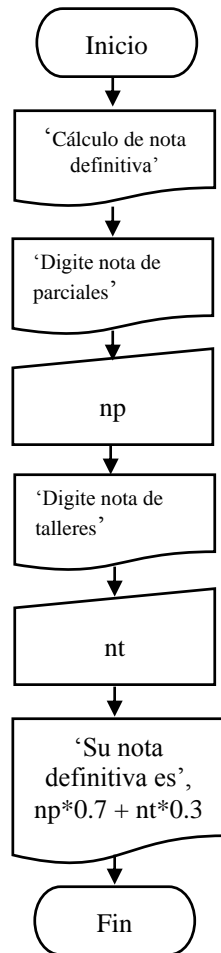
a. D E/S



b. GUI



4. Fase de diseño (Diagrama de Flujo DF)



Condicional compuesto

Ejemplo. Diseñar una solución computacional que capture para una persona su edad y determine si es mayor o menor de edad.

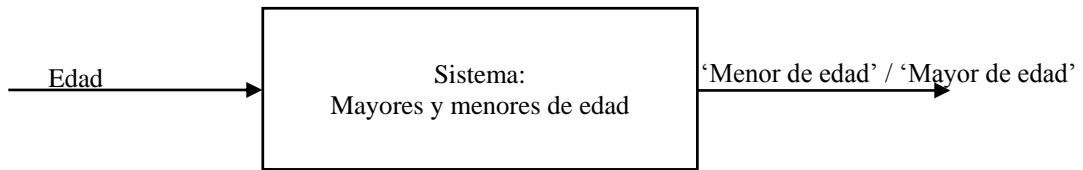
Solución

1. Fase de Recolección de requerimientos

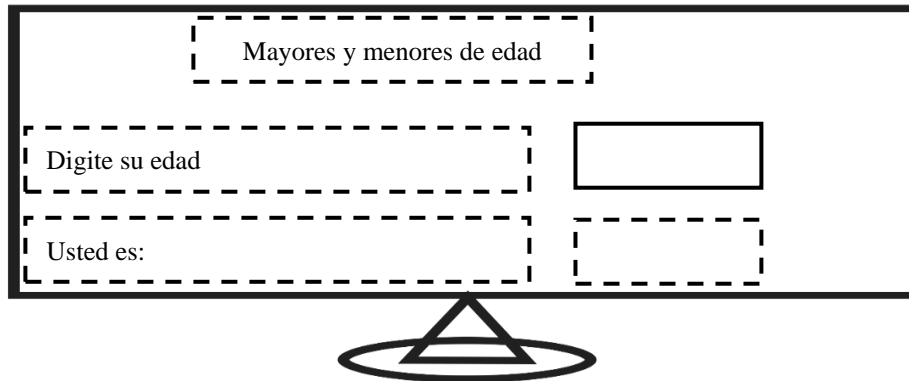
Nombre del sistema	Requerimiento principal
Mayores y menores de edad	Capturar una edad y determinar si esa persona es menor o mayor de edad

2. Fase de Análisis

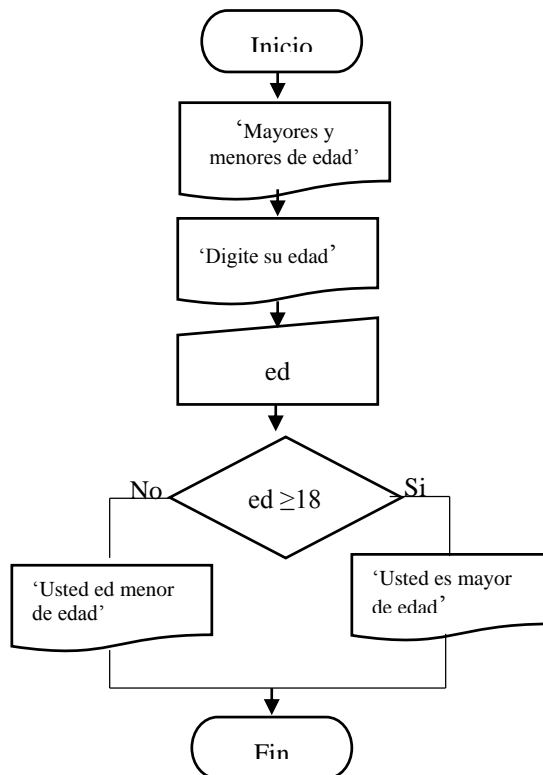
c. D E/S



d. GUI



3. Fase de diseño



Condicionado anidado

Ejemplo. Diseñar una solución computacional que capture tres números y determine si son iguales o diferentes.

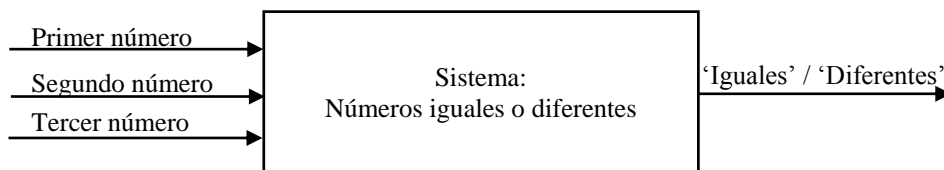
Solución

1. Fase de Recolección de requerimientos

Nombre del sistema	Requerimiento principal
Números iguales o diferentes	Capturar tres número y determinar si son iguales o diferentes

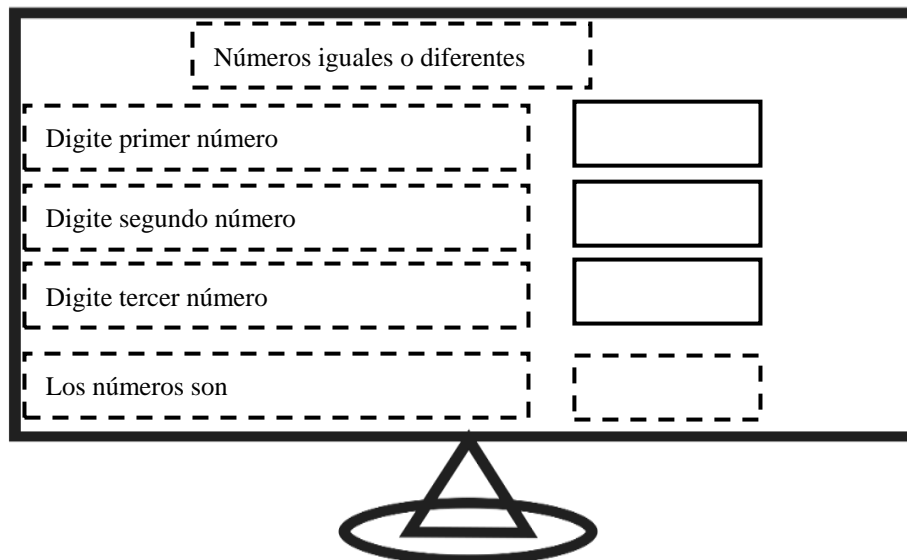
2. Fase de Análisis

a. D E/S

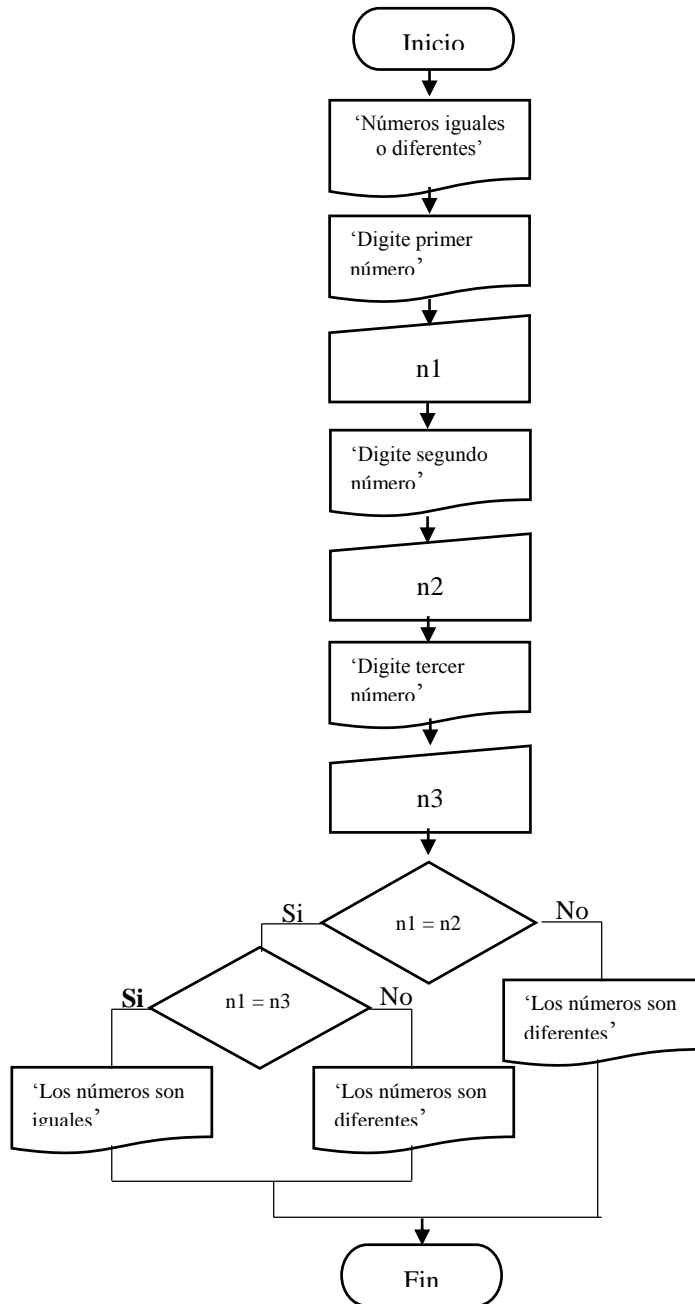


Nota al docente. Es importante indicarle al estudiante que a pesar de que existen dos alternativas para el usuario general (iguales o diferentes), en realidad a nivel de analista son tres las posibilidades: Iguales, diferentes o diferentes como se aprecia en la figura 43.

b. GUI



3. Fase de diseño



Ejemplo. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(20 \leq n \leq 300)$.

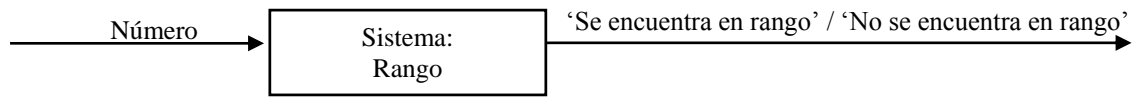
Solución

1. Fase de Recolección de requerimientos

Nombre del sistema	Requerimiento principal
Rango	Capturar un número y determinar si se encuentra o no en un rango

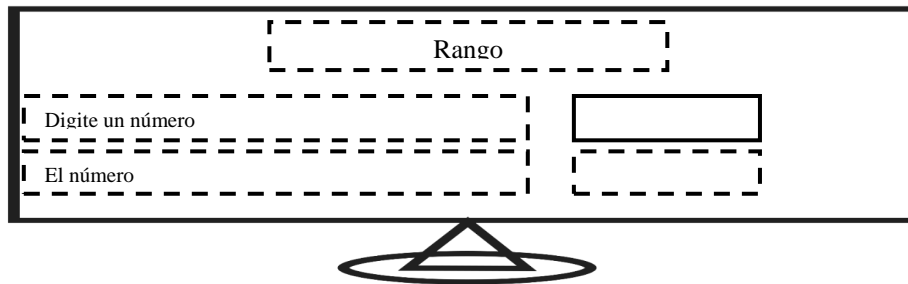
2. Fase de Análisis

a. D E/S

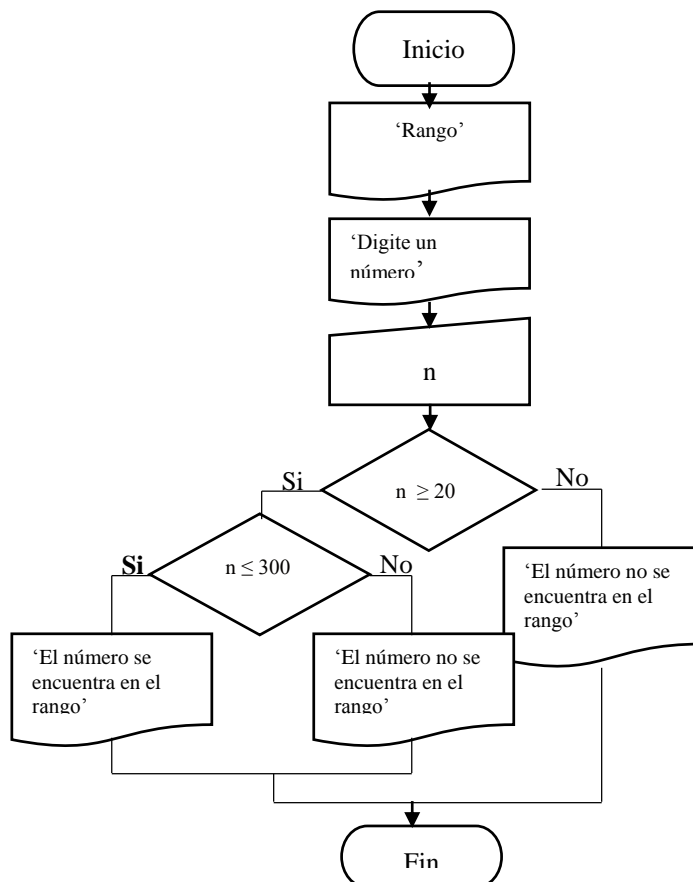


Nota al docente. Es importante indicarle al estudiante que a pesar de que existen dos alternativas para el usuario general ('se encuentra en rango' o 'no se encuentra en rango'), en realidad a nivel de analista son tres las posibilidades: 'Se encuentra en rango', 'No se encuentra en rango' o 'No se encuentra en rango' como se aprecia en la figura 3.46.

b. GUI



3. Fase de diseño



Condicionales y operadores lógicos

Ejemplo. Diseñar una solución computacional que capture un número y determine si se encuentra o no en el siguiente rango: $(200 \leq n \leq 1200)$ y $(n \geq 3000)$.

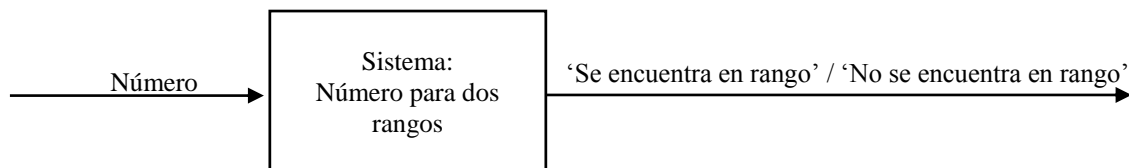
Solución

1. Fase de Recolección de requerimientos

Nombre del sistema	Requerimiento principal
Numero para dos rangos	Capturar un número y determinar si se encuentra o no en un rango

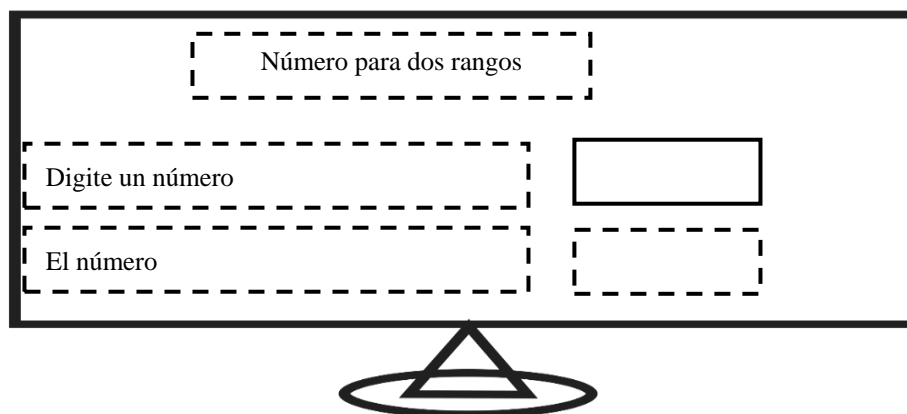
2. Fase de Análisis

a. D E/S

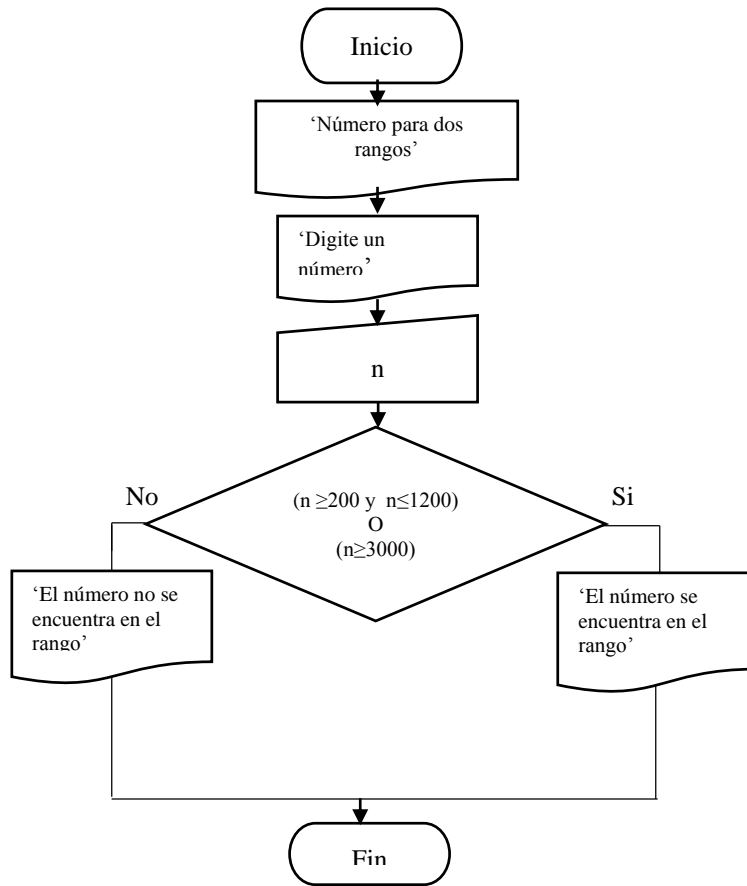


Nota al docente. En este caso la salida por tener dos alternativas el condicional a utilizar debe ser compuesto

b. GUI



3. Fase de diseño



Ciclo Para Ascendente

Ejemplo. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde dicho valor en forma ascendente

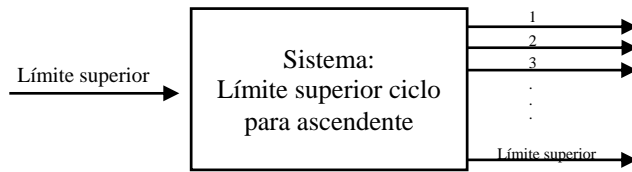
Solución

1. Fase de Recolección de requerimientos

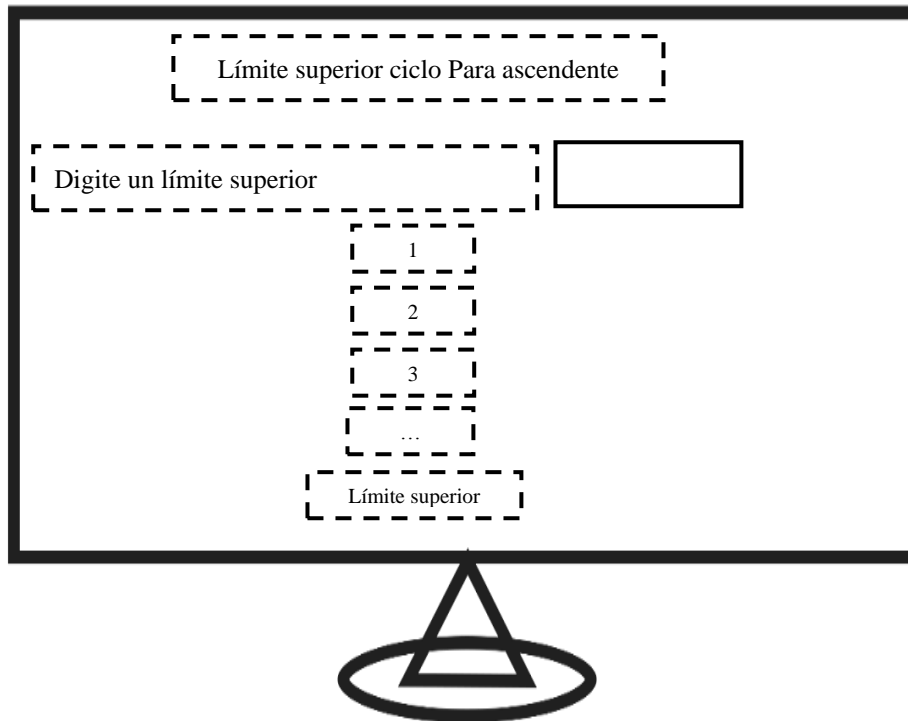
Nombre del sistema	Requerimiento principal
Límite superior ciclo para ascendente	Capturar un número e Imprimir los valores desde 1 hasta dicho límite superior en forma ascendente

2. Fase de Análisis

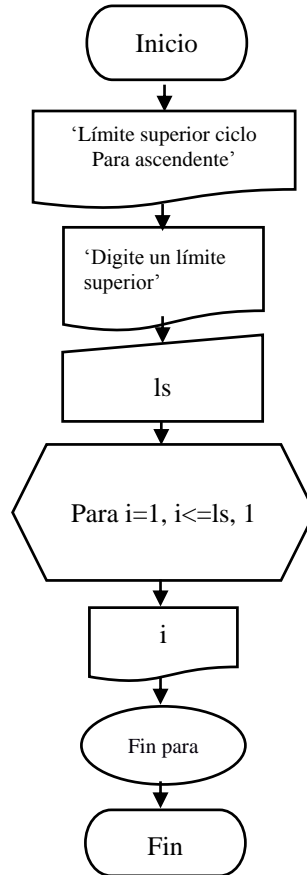
a. D E/S



b. GUI



3. Fase de diseño



Ciclo Para Ascendente

Ejemplo. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde dicho límite hasta 1 en forma descendente.

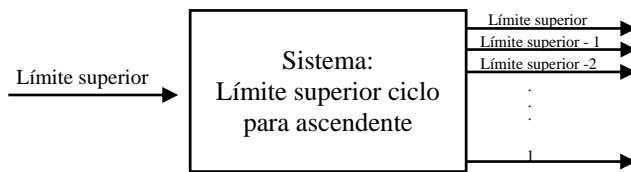
Solución

1. Fase de Recolección de requerimientos

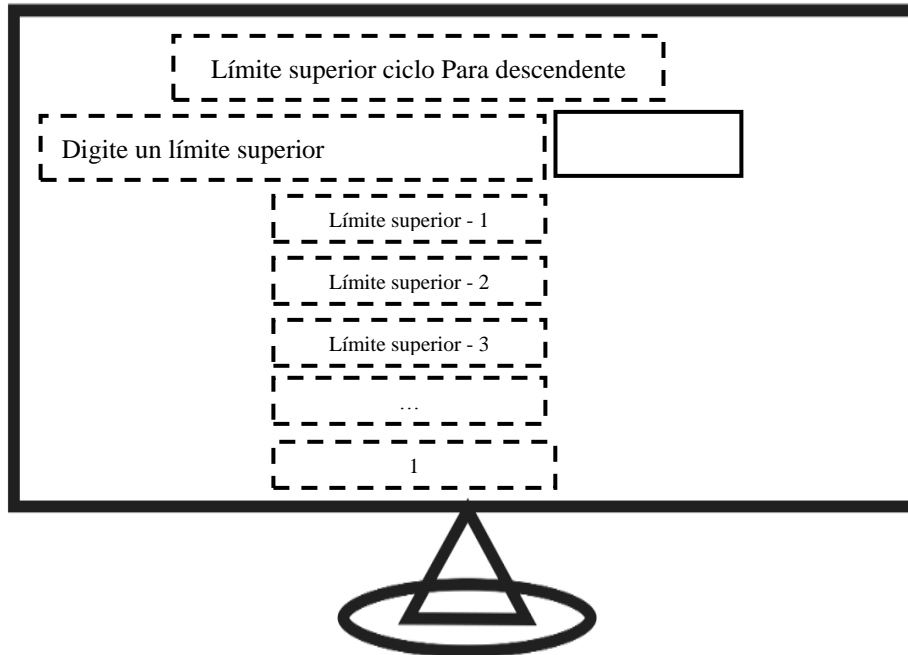
Nombre del sistema	Requerimiento principal
Límite superior ciclo para descendente	Capturar un número e Imprimir los valores desde dicho límite superior hasta 1 en forma descendente

2. Fase de Análisis

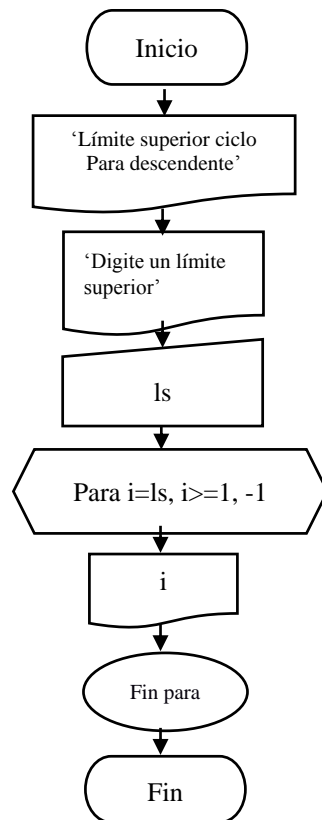
a. D E/S



b. GUI



3. Fase de diseño



Ciclo Mientras

Ejemplo. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde dicho valor en forma ascendente

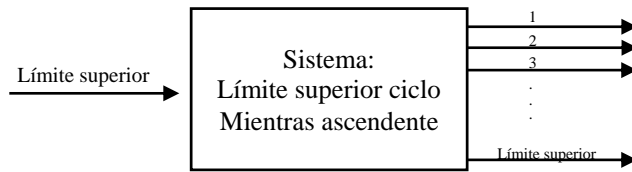
Solución

1. Fase de Recolección de requerimientos

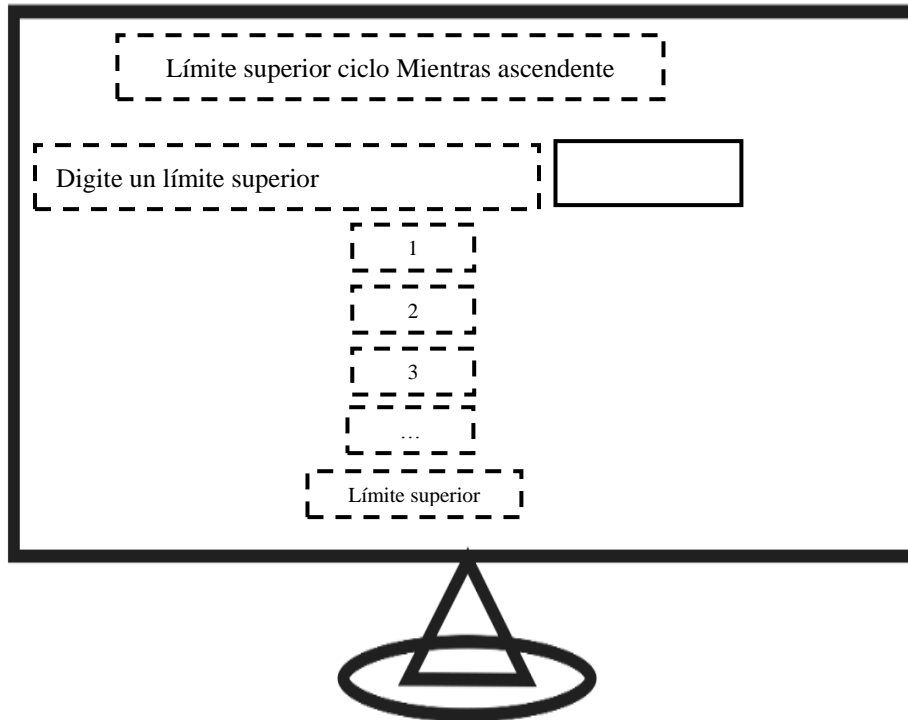
Nombre del sistema	Requerimiento principal
Límite superior ciclo Mientras ascendente	Capturar un número e Imprimir los valores desde 1 hasta dicho límite superior en forma ascendente

2. Fase de Análisis

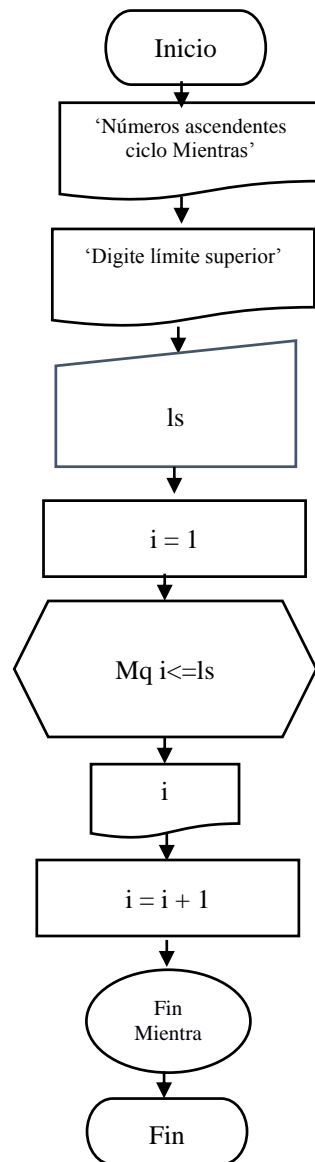
a. D E/S



b. GUI



3. Fase de diseño



Ejemplo. Diseñar una solución computacional que imprima los siete primeros números en forma descendente.

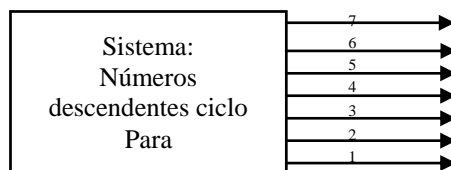
Solución

1. Fase de Recolección de requerimientos

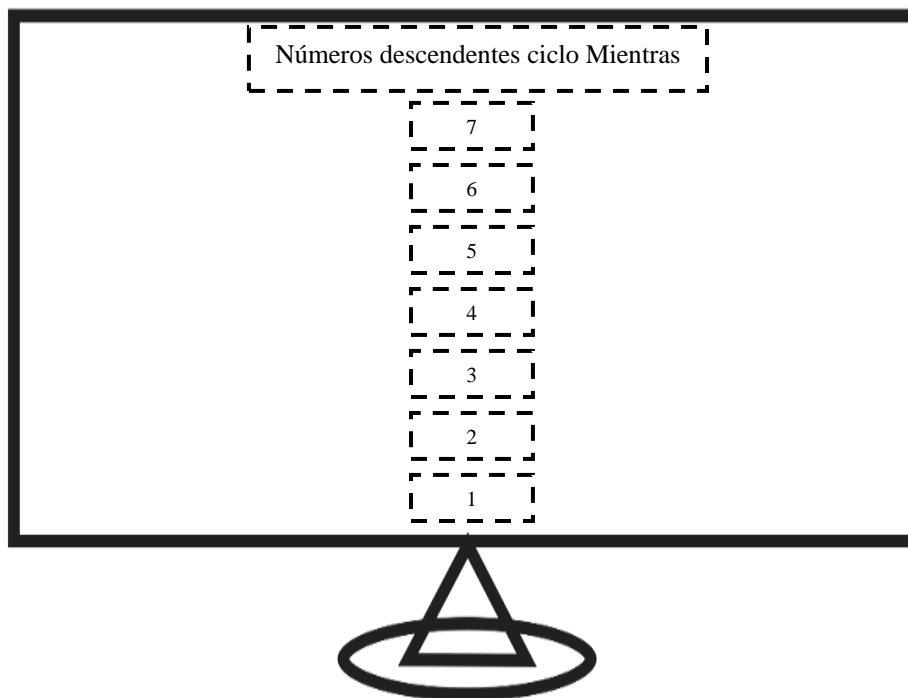
Nombre del sistema			Requerimiento principal
Números descendentes	ciclo	Mientras	Imprimir los siete primeros números en forma descendente

2. Fase de Análisis

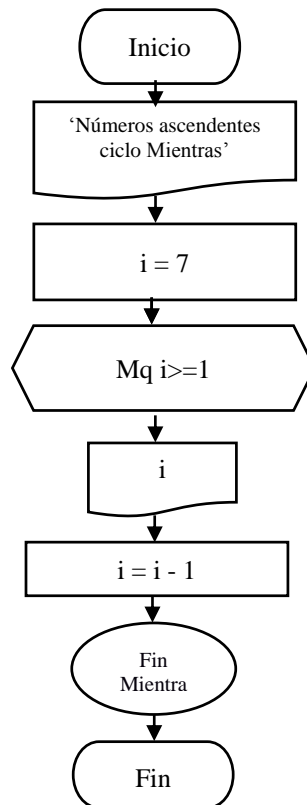
a. D E/S



b. GUI



3. Fase de diseño



Ejemplo. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde dicho límite hasta 1 en forma descendente

Solución

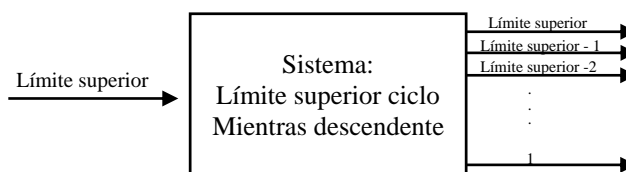
1. Fase de Recolección de requerimientos

Tabla 3.42. Recolección de requerimientos límite superior ciclo Mientras descendente

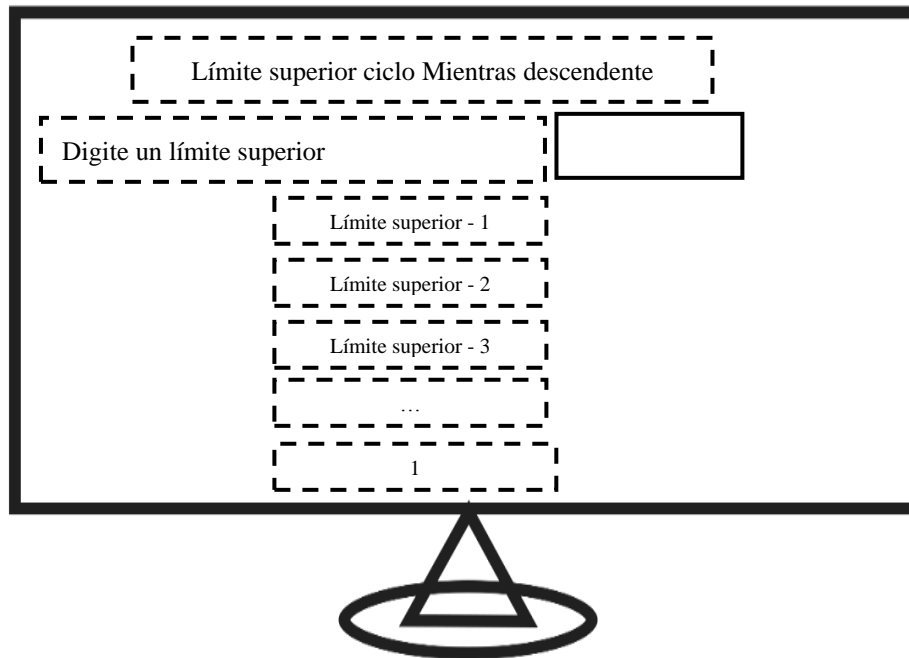
Nombre del sistema	Requerimiento principal
Límite superior ciclo Mientras descendente	Capturar un número e Imprimir los valores desde dicho límite superior hasta 1 en forma descendente

2. Fase de Análisis

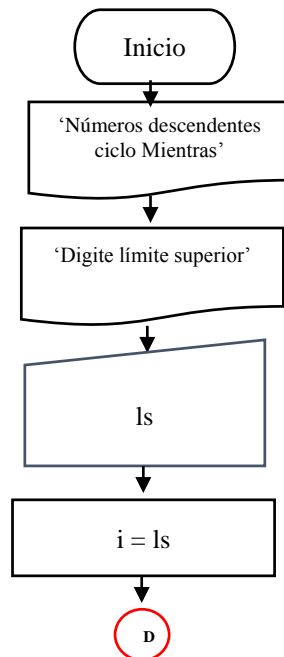
a. D E/S

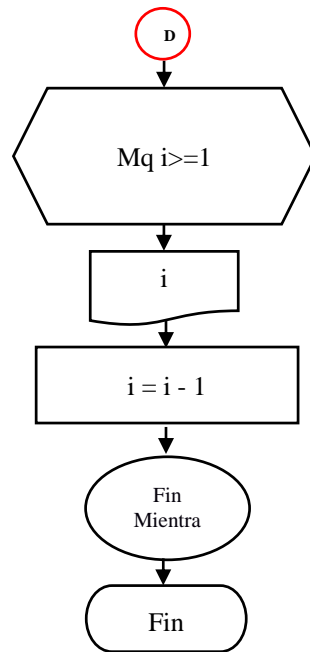


b. GUI



3. Fase de diseño





Ciclo Hacer Mientras

Ejemplo. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde dicho valor en forma ascendente.

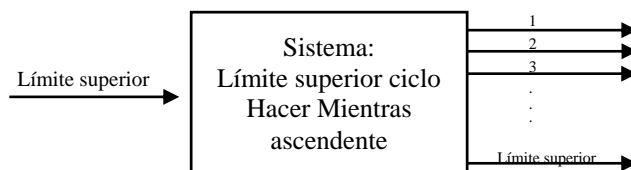
Solución

1. Fase de Recolección de requerimientos

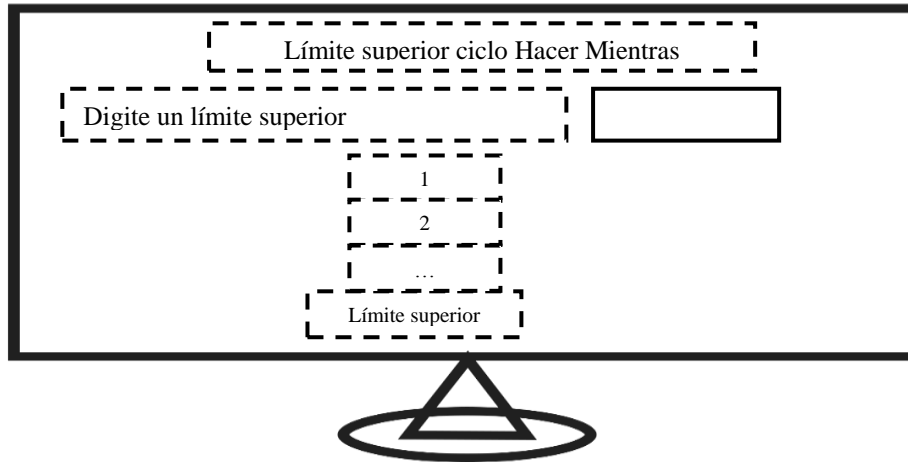
Nombre del sistema	Requerimiento principal
Límite superior ciclo Hacer Mientras ascendente	Capturar un número e Imprimir los valores desde 1 hasta dicho límite superior en forma ascendente

2. Fase de Análisis

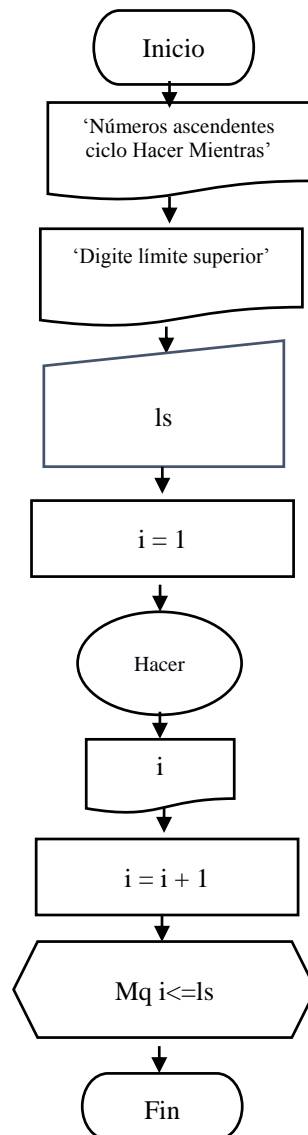
a. D E/S



b. GUI



3. Fase de diseño



Ejemplo. Diseñar una solución computacional que imprima los siete primeros números en forma descendente.

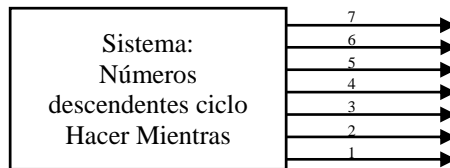
Solución

1. Fase de Recolección de requerimientos

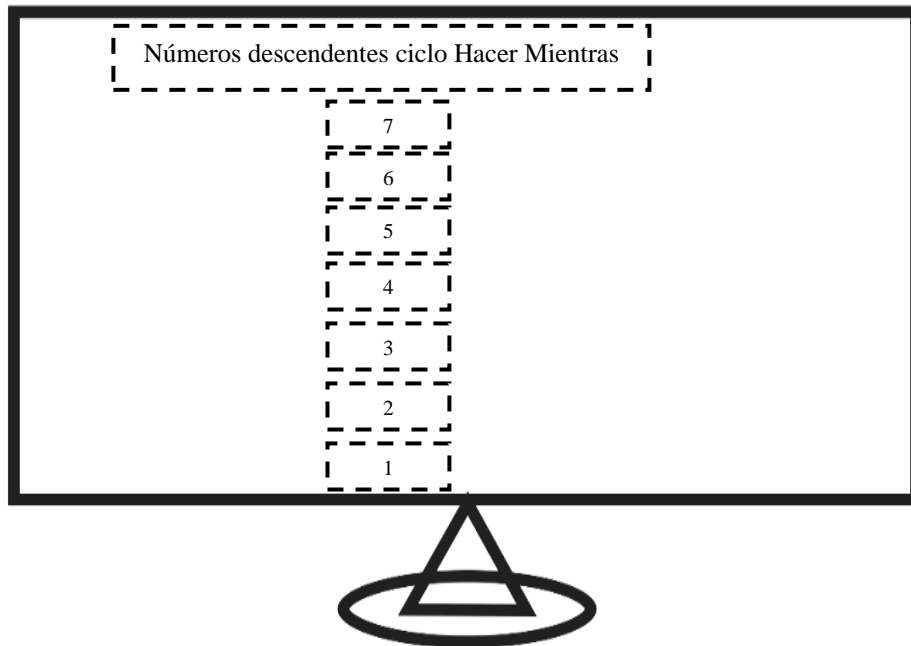
Nombre del sistema	Requerimiento principal
Números descendentes ciclo Mientras	Imprimir los siete primeros números en forma descendente

2. Fase de Análisis

a. D E/S



b. GUI



3. Fase de diseño

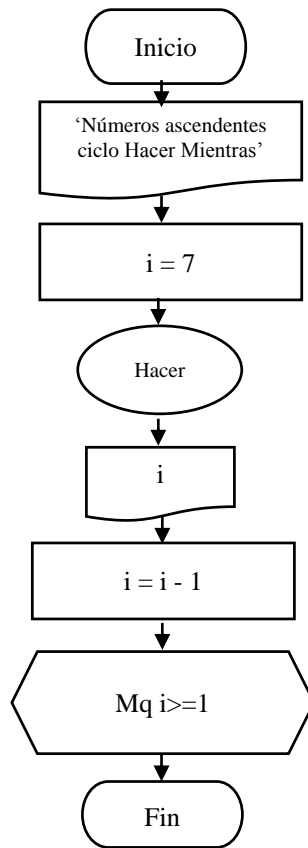


Figura 3.85. DF número descendente ciclo Hacer Mientras

Ejemplo. Diseñar una solución computacional que capture un límite superior positivo e imprima los números comprendidos desde dicho límite hasta 1 en forma descendente

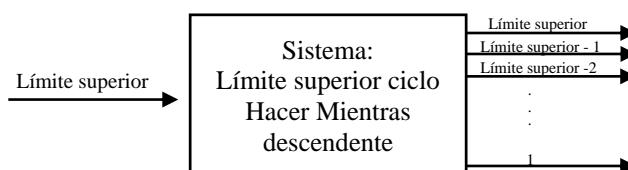
Solución

1. Fase de Recolección de requerimientos

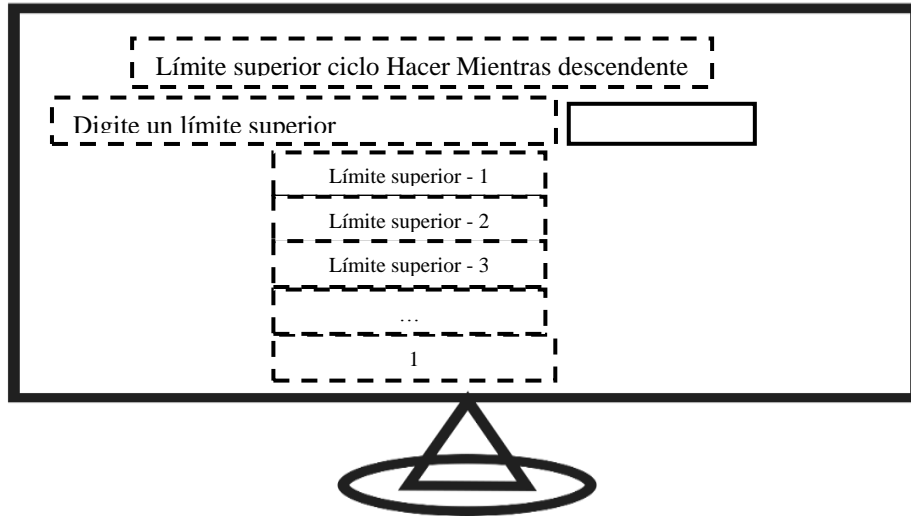
Nombre del sistema	Requerimiento principal
Límite superior ciclo Hacer Mientras descendente	Capturar un número e Imprimir los valores desde dicho límite superior hasta 1 en forma descendente

2. Fase de Análisis

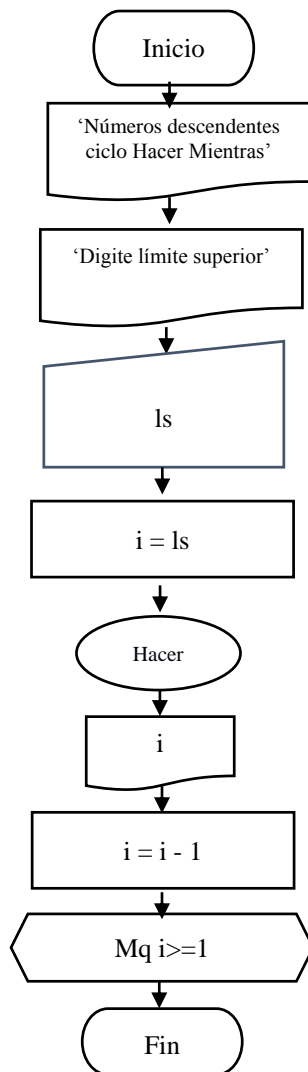
a. D E/S



b. GUI



3. Fase de diseño



Anexo 15



Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Encuesta a expertos

Nombre: _____

Universidad: _____

Último nivel de formación: _____

Años de experiencia en cursos de programación: _____

Califique de 1 a 5 (donde 1 es el menor valor y 5 es el mayor) los siguientes ítems:

1. ¿Cuál es su experiencia docente en cursos de fundamentos de programación?

1	2	3	4	5

2. ¿Cuál es su experiencia con métodos colaborativos como estrategias didácticas en sus clases?

1	2	3	4	5

3. ¿El modelo colaborativo presentado es claro?

1	2	3	4	5

4. ¿El modelado gráfico propuesto es claro?

1	2	3	4	5

5. Considera usted que el empleo de JIGSAW ¿es una estrategia didáctica apropiada para esta propuesta metodológica?

1	2	3	4	5

6. Evalúe únicamente la experiencia del proceso metodológico que realizó usted con su grupo de trabajo en clase

1	2	3	4	5

7. ¿Cómo considera usted el tomar el D E/S de base fundamental del modelado gráfico propuesto en esta metodología?

1	2	3	4	5

8. Evalúe el nivel de dificultad o facilidad encontrado por cada unidad de competencia:

Unidad de competencia	1	2	3	4	5
Captura y salida de datos					
Condicionales simples					
Condicionales compuestos					
Condicionales anidados					
Condicionales y operadores lógicos					
Ciclo Para ascendente					
Ciclo Para descendente					
Ciclo Mientras					
Ciclo Hacer Mientras					

9. Los roles de docente y estudiante ¿son pertinentes a la metodología planteada?

1	2	3	4	5

10. Los sub roles de los estudiantes: Analista y diseñador ¿son pertinentes a la metodología planteada?

1	2	3	4	5

11. Evalúe únicamente la experiencia del proceso académico obtenido con su grupo de trabajo.

1	2	3	4	5
----------	----------	----------	----------	----------

--	--	--	--	--

12. ¿Recomendaría la metodología desarrollada con otros colegas de trabajo?

1	2	3	4	5

Anexo 16



Universidad del Cauca Maestría en Computación

Propuesta metodológica basada en aprendizaje colaborativo y modelado gráfico para la enseñanza de los fundamentos de programación

Fecha: _____

Formato de caracterización de estudiantes	
Nombres	
Apellidos	
Edad	
Universidad	
Facultad	
Programa	
Semestre	
Jornada	
Asignatura por primera vez _____ repite una vez _____ repite más de una vez _____	
Observaciones _____ _____ _____ _____	

Anexo 17

INSTITUCIÓN UNIVERSITARIA CENTRO DE ESTUDIOS SUPERIORES MARIA GORETTI - IUCESMAG

Listado Asistencia de Matriculados por Asignatura

Ingeniería de Sistemas

Periodo académico: 2015-1

Semestre/Grupo: 1A-M

Asignatura: INTRODUCCION A LA PROGRAMACION

Docente: JIMENEZ TOLEDO ROBINSON ANDRES

No.	CÓDIGO	NOMBRE ESTUDIANTE	Captura y salida	Condicionales	Ciclos	Definitiva
1	I047115	ACOSTA MOSQUERA BRYAN YESSID	4,0	4,5	4,1	4,2
2	I054115	ALVAREZ VILLOTA JAIME DAVID	5,0	1,0	5,0	3,7
3	I051115	BENAVIDES MARCILLO LUIS MIGUEL	5,0	1,0	2,0	2,7
4	I025115	BUCCELLI BENAVIDES CAMILO ANDRES	3,5	3,9	3,5	3,6
5	I004115	CARLOSAMA DAZA LUIS GABRIEL	3,4	2,7	2,5	2,9
6	I017115	CERON CEPEDA WILLIAN ALEXANDER	4,1	3,6	3,6	3,8
7	I041115	DELGADO MOREANO ESTEBAN DAVID	5,0	2,5	2,6	3,4
8	I046115	ESPAÑA MONTAÑO JEISON HAMMER	3,4	3,8	5,0	4,1
9	I012115	GETIAL BOTINA NELSON JOHAN	3,3	2,3	2,0	2,5
10	I065115	GETIAL . IVAN ANDRES	5,0	3,7	3,7	4,1
11	I043115	GOMEZ HERNANDEZ JOHN ALEJANDRO	5,0	3,9	3,9	4,2
12	I015115	GRAJALES PATIÑO CARLOS ANDRES	3,7	3,3	5,0	4,0
13	I067115	GUANCHA NAVARRO DAYERLINE	4,0	4,0	4,1	4,0
14	I005115	GUERRERO RUIZ CAMILO SEBASTIAN	3,8	3,4	5,0	4,1
15	I061115	GUSTIN DIAZ JHON JAIRO	3,0	3,1	3,1	3,1
16	I044115	JUAJINYO DIAZ FRANCISCO JAVIER	4,0	4,7	4,7	4,4
17	I021115	MADROÑERO MELO EDWIN MAURICIO	3,4	2,8	2,5	2,9
18	I068115	MAIGUAL MORENO LUIS ANDRES	5,0	2,9	2,9	3,6
19	I059115	MARTINEZ PASCUAZA EDISON ORLANDO	5,0	2,4	2,4	3,2
20	I062115	OBANDO ARAUJO DIEGO ALEXANDER	4,0	4,2	4,2	4,1
21	I057115	OLIVA MANCHABAJAY LUIS ENRIQUE	5,0	1,0	5,0	3,7
22	I022115	RAMIREZ BARCO DIEGO HERNANDO	3,9	4,0	3,0	3,6
23	I058115	ROSETO AREVALO JOSE ENRIQUE	5,0	2,5	2,5	3,4
24	I018115	ROSETO PAZOS MARIA CAMILA	4,0	4,3	4,0	4,1
25	I064115	SOLARTE FIGUEROA LUIS MANUEL	3,3	3,6	3,6	3,5
26	I028214	TULCAN CRIOLLO EDISSON DAVID	2,0	2,0	5,0	3,0
27	I055115	ZAMBRANO ROSETO MAURICIO FERNANDO	3,0	4,0	4,0	3,7

Anexo 18

UNIVERSIDAD ANTONIO NARIÑO

Ingeniería Electromecánica Lógica Computacional

Semestre 1

Periodo I - 2015

No.	Código	Estudiante	Captura y salida	Condicionales	Ciclos	Definitiva
1	21131515446	ANDRADE GOMEZ DAVID ALEJANDRO	2,8	4,1	4,1	3,7
2	21511515470	BASTIDAS PAREDES JUAN DAVID	4,0	3,0	4,0	3,7
3	21511512559	CHAVEZ CUELLAR DANOVIZ REMBERTO	5,0	3,0	3,0	3,7
4	21511518989	CRIOLLO INSUASTY PEDRO DANIEL	2,9	5,0	4,0	4,0
5	21131512410	ERASO ESTRELLA RUBEN DARIO	4,3	2,0	2,0	2,8
6	21511517932	FIGUEROA ADARME EDISON HARLEY	3,8	4,1	3,5	3,8
7	21511511365	GALEANO QUETAMA ALVARO ANDRES	3,9	5,0	5,0	4,6
8	21131515155	LOZANO VARGAS JOHN ALEXANDER	4,1	4,0	4,0	4,0
9	21511514589	MESIAS ENRIQUEZ GERMAN MAURICIO	4,9	5,0	4,0	4,6
10	21511516062	NUPAN ROSERO JHOAN SEBASTIAN	3,0	2,0	4,0	3,0
11	21511515237	PRADO PEJENDINO JHON SEBASTIAN	2,8	4,0	4,0	3,6
12	21511511590	RIAÑO INSUASTY ANDRES LIBARDO	3,7	3,0	3,0	3,2
13	21511519693	ROSAS BERMEO JORGE	2,5	3,0	3,0	2,8
14	21511516342	SARASTY CHATES JAVIER ORLANDO	3,8	3,0	3,0	3,3

Javier Alejandro Jiménez Toledo

Docente

Anexo 19



UNIVERSIDAD DE NARIÑO
Facultad de Ingeniería

Periodo académico: 2014-II

Semestre/Grupo:

1-1

Asignatura: FUNDAMENTOS DE PROGRAMACION

No.	CÓDIGO	Apellidos	Nombres	Captura y salida	Condi- cionales	Ciclos	Def.
1	2140341002	ACOSTA PEREZ	NIXON DANILO	5,0	4,0	4,0	4,3
2	215034012	ARANDO RIVEROS	GRINY LIZETH	2,0	3,0	4,0	3,0
3	215034024	BASTIDAS BENAVIDES	CRISTHIAN CAMILO	4,0	5,0	5,0	4,7
4	215034026	BASTIDAS LUNA	RICARDO SEBASTIAN	5,0	1,0	1,0	2,3
5	215034049	BOTINA CARLOSAMA	JUAN PABLO	5,0	5,0	5,0	5,0
6	215034088	CHAPAL VALLEJO	JHONNY EDISON	5,0	5,0	5,0	5,0
7	215034094	CHAZATAR CORDOBA	DANIEL CAMILO	5,0	2,5	2,5	3,3
8	215034097	CONTRERAS CABRERA	ALVARO DAVID	2,0	3,0	3,0	2,7
9	215034102	CORDOBA TONGUINO	ROMEL ESTEBAN	5,0	5,0	5,0	5,0
10	215034107	CUAICAL MONTENEGRO	MAURICIO CAMILO E.	5,0	5,0	5,0	5,0
11	215034112	CUASTUMAL PAREDES	JUAN CAMILO	5,0	2,5	2,5	3,3
12	215034125	DELGADO ORDONEZ	SANTIAGO DAVID	5,0	5,0	5,0	5,0
13	215034134	ERASO MORA	EDGAR MAURICIO	5,0	3,5	3,5	4,0
14	215034145	ESCOBAR BOLAÑOS	JHON STEVEN	2,0	5,0	5,0	4,0
15	215034152	FLOREZ SANTANDER	JUAN JOSE	5,0	5,0	5,0	5,0
16	215034154	FUENMAYOR TOVAR	WILMER DANILO	5,0	5,0	5,0	5,0
17	215034155	GALEANO VARGAS	JEISON MARTIN	5,0	5,0	5,0	5,0
18	215034157	GARCES VILLARREAL	JUAN ALEJANDRO	5,0	3,0	3,0	3,7
19	215034159	GARZON MORA	JEFERSON ESTEVEN	5,0	5,0	5,0	5,0
20	215034167	GOMEZ SEPULVEDA	CAROLINA	5,0	3,0	3,0	3,7
21	215034181	GUDINO PLAYONERA	CARLOS ANDRES	5,0	3,5	3,5	4,0
22	215034184	GUERRERO CALDERON	JUAN CAMILO	2,0	3,0	3,0	2,7
23	215034191	GUERRERO JOSA	CRISTIAN ALEXANDER	5,0	3,0	3,0	3,7
24	215034194	GUERRERO MUÑOZ	MARIA JOSE	2,0	5,0	5,0	4,0
25	215034196	GUERRERO RIASCOS	JOHN SNYDER	5,0	5,0	5,0	5,0
26	215034197	GUERRERO TORRES	ANGIE PAOLA	4,0	5,0	5,0	4,7
27	215034201	GUZMAN NARVAEZ	CARLOS ANDRES	5,0	5,0	5,0	5,0
28	215034206	HIDALGO RODRIGUEZ	JUAN FELIPE	5,0	5,0	5,0	5,0
29	215034229	JURADO ORTEGA	LUIS CARLOS	2,0	3,0	4,0	3,0
30	215034232	LASSO GUZMAN	CAMILO DAVID	5,0	5,0	5,0	5,0
31	215034243	LUNA OSORIO	MATEO	5,0	3,0	4,0	4,0
32	215034269	MEJIA GUERRERO	NATALIA	5,0	3,0	4,0	4,0
33	2130341237	MONTEZUMA OJEDA	FRANCO MAURICIO	5,0	3,0	4,0	4,0

Jesús Insuasty
Docente

Anexo 20

INSTITUCIÓN UNIVERSITARIA CENTRO DE ESTUDIOS SUPERIORES MARIA GORETTI - IUCESMAG

Listado Asistencia de Matriculados por Asignatura

Ingeniería de Sistemas

Periodo académico: 2015-2

Semestre/Grupo: 1A-M

Asignatura: INTRODUCCION A LA PROGRAMACION

Docente: JIMENEZ TOLEDO JAVIER ALEJANDRO

No.	CÓDIGO	NOMBRE ESTUDIANTE	Captura y salida	Condicionales	Ciclos	Definitiva
1	I003115	AGREDA MUÑOZ JOHNNY ALEXANDER	5,0	2,6	2,6	3,4
2	I010115	ALVAREZ SANCHEZ JEFFERSON	3,4	4,0	3,8	3,7
3	I056115	ALVAREZ VILLOTA WILLIAM HERNANDO	5,0	5,0	5,0	5,0
4	I009115	ASTORQUIZA BURBANO CRISTHIAN	4,5	4,0	4,1	4,2
5	I037115	BOLANOS EGAS LUISA FERNANDA	5,0	5,0	5,0	5,0
6	I042115	BURBANO BENAVIDES CRISTOFER	4,6	4,6	4,5	4,6
7	I069115	CAICEDO RIOS ROGER DARIO	3,8	4,5	4,5	4,3
8	I050115	CHARFUELAN CUARAN JOSE ORLANDO	4,1	4,4	4,4	4,3
9	I034115	CUASTUMAL MORA YILVER ANDRES	5,0	4,0	4,0	4,3
10	I011115	DELGADO MONCAYO TANYA MARCELA	4,9	3,8	4,0	4,2
11	I032115	DIAZ MUÑOZ LUIS FERNANDO	5,0	3,7	3,7	4,1
12	I006115	DUEÑAS ROSERO JHONATAN DAVID	4,7	2,9	5,0	4,2
13	I070115	ERAZO CHAMORRO JUAN SEBASTIAN	5,0	5,0	5,0	5,0
14	I035115	ESTRADA CHALIAL JULIAN DAVID	5,0	4,2	4,2	4,4
15	I020115	GOMEZ NARVAEZ ROSY ANYELI	4,5	3,0	3,0	3,5
16	I016115	GRANJA QUIÑONES BRANDON STEVEN	5,0	3,4	3,4	3,9
17	I007115	GUERRRERO CAICEDO MARIA FERNANDA	0,8	3,0	5,0	2,9
18	I053115	JATIVA VILLOTA CHRISTIAN EDUARDO	5,0	3,0	3,0	3,7
19	I052115	LAGOS CABEZAS DILSA YANETH	5,0	3,3	3,3	3,9
20	I027115	LATORRE ZAPATA DUVAN RENE	5,0	4,4	4,4	4,6
21	I001115	LLORENTE CUASPUD JOSE DARIO	4,6	4,0	4,0	4,2
22	I039115	MAYA RIASCOS DARIO ALEJANDRO	4,9	2,8	5,0	4,2
23	I045115	PALACIOS CALPA CARLOS MARIO	5,0	4,4	4,4	4,6
24	I049115	PAZ PANTOJA DIEGO ANDRES	5,0	4,0	4,0	4,3
25	I008115	RIASCOS BONILLA JUAN FELIPE	5,0	5,0	5,0	5,0
26	I048115	RODRIGUEZ BASTIDAS LUIZ ANTONIO	3,5	2,7	5,0	3,7
27	I014115	ROSETO LOPEZ ANITA ALEXANDRA	3,1	3,3	3,3	3,2
28	I026115	ROSETO NARVAEZ SANTIAGO NICOLAS	4,0	5,0	5,0	4,7
29	I030115	TIMANA LOPEZ CARLOS JAIR	5,0	3,5	3,0	3,8
30	I002115	TORRES GARCIA JULIAN ESTEBAN	5,0	5,0	5,0	5,0
31	I024115	URREA URRESTY CLAUDIA VANESSA	4,4	3,5	3,5	3,8
32	I036115	VALLEJO ROSETO CHRISTIAN ORLANDO	4,1	3,1	3,1	3,4
33	I040115	VILLARREAL RIVAS DANIEL FERNANDO	4,5	4,0	4,0	4,2

Anexo 21

UNIVERSIDAD ANTONIO NARIÑO

Ingeniería Electromecánica

Lógica Computacional

Semestre 1

Periodo II - 2015

No.	Código	Estudiante	Captura y salida	Condi- cionales	Ciclos	Defi- nitiva
1	21511521042	BASTIDAS BARCENAS NEYER FERNANDO	5,0	4,8	4,5	4,8
2	21511523630	BENAVIDES CEBALLOS ELDER ALBERTO	3,3	2,6	2,6	2,8
3	21511528101	BUESAQUILLO RIRAMAG WILMAN DIEGO	4,8	4,3	4,3	4,5
4	21511528890	CAICEDO ARELLANO LUIS CARLOS	4,9	4,0	4,0	4,3
5	21511528214	CAICEDO RIVERA NELSON RICARDO	5,0	5,0	5,0	5,0
6	21511522012	CASTRO - HERNAN ARIEL	3,0	4,0	4,0	3,7
7	21511526656	CHAVEZ MARTINEZ PEDRO BOLIVAR	5,0	4,8	4,8	4,9
8	21511528791	PORTILLA VILLARREAL GABRIEL ALEJANDRO	5,0	4,4	4,4	4,6
9	21511524496	ROJAS TIMARAN JEYSON WILLIAM	3,1	4,5	4,5	4,0
10	21511527724	ROSERO ESCOBAR JONNATHAN ALEJANDRO	4,9	5,0	4,5	4,8
11	21511528806	TERAN CUASPA EDWUAR ANDRES	5,0	4,6	4,6	4,7

Javier Alejandro Jiménez Toledo

Docente

Anexo 22



UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
 Ingeniería Electrónica

Periodo académico: 2015-II

Semestre/Grupo:
 1-2

Asignatura: FUNDAMENTOS DE PROGRAMACION

No.	CÓDIGO	Apellidos	Nombres	Captura y salida	Condi-cionales	Ciclos	Def.
1	215160015	ANGANROY MORENO	FAUSTO ANDRES	4,9	3,0	5,0	4,3
2	215160026	ARROYO BELALCAZAR	KARLA ESTEFANI	4,9	5,0	5,0	5,0
3	215160041	BELTRAN DE LA CRUZ	MARIO ALEJANDRO	4,9	3,5	3,5	4,0
4	215160042	BENAVIDES CHINGAL	IVAN DARIO	4,6	5,0	5,0	4,9
5	215160056	BOLAÑOS PANTOJA	NELCY MILENA	4,9	3,4	3,4	3,9
6	215160063	BURBANO BENAVIDES	DONOVAN STEVEN	4,9	3,3	3,3	3,8
7	215160078	CABRERA GUERRERO	JAVIER ELIECER	4,9	5,0	5,0	5,0
8	215160079	CABRERA NARVAEZ	BRAYAN JAVIER	4,9	3,5	3,5	4,0
9	215160081	CABRERA ZAMBRANO	MARIO FERNANDO	4,9	5,0	5,0	5,0
10	215160086	CAICEDO ORTEGA	JHONATAN JESUS	4,9	3,5	3,5	4,0
11	27160252	CANO RODRIGUEZ	SERGIO ARMANDO	4,9	5,0	5,0	5,0
12	215160093	CARLOSAMA GOMEZ	DANIEL FELIPE	4,9	5,0	5,0	5,0
13	215160109	CHAMORRO FUERTES	LIZET PAOLA	4,9	5,0	5,0	5,0
14	215160111	CHAMORRO MEJIA	EDGAR DANIEL	4,9	5,0	5,0	5,0
15	215160125	CHITAN BASTIDAS	DIEGO ANDRES	4,9	5,0	5,0	5,0
16	215160131	CÓRDOBA TORRES	HUMBERTO SEBASTIÁN	5,0	5,0	5,0	5,0
17	215160155	DEL VALLE INSUASTY	CRISTHIAN CAMILO	5,0	5,0	5,0	5,0
18	215160153	DELGADO ROSERO	DIEGO ALDAIR	4,9	5,0	5,0	5,0
19	215160157	DIAZ PANTOJA	YELI TATIANA	4,9	5,0	5,0	5,0
20	215160158	DIAZ ROSALES	OSCAR ANDRES	4,9	5,0	5,0	5,0
21	215160167	ERASO GUERRA	ANYI PAOLA	4,6	5,0	5,0	4,9
22	215160168	ERASO MISNAZA	SEBASTIAN	4,9	3,3	5,0	4,4
23	215160174	ERAZO TREJO	MAURICIO FERNANDO	4,6	5,0	5,0	4,9
24	215160193	GARCES DELGADO	LAURA MARCELA	4,5	5,0	5,0	4,8
25	215160210	GUAQUEZ YAMA	JONH JAIRO	4,8	5,0	5,0	4,9
26	215160212	GUERRERO CHAVES	WILSON JACOB	4,9	3,8	3,8	4,1
27	215160233	INSANDARA SINSAJOA	DIEGO FERNANDO	4,9	5,0	5,0	5,0
28	215160237	INSUASTY ROMO	SANTIAGO ANDRES	4,9	5,0	5,0	5,0
29	27166231	MARCILLO PATIÑO	HEIDY	4,8	5,0	5,0	4,9

Javier Jiménez Toledo
 Docente

Anexo 23

Revista
INVESTIGIUM IRE
Ciencias Sociales y Humanas



Publicación científica que contribuye a la divulgación y difusión de conocimiento de carácter científico e investigativo en Ciencias Sociales y Humanas desde la Institución Universitaria Centro de Estudios Superiores María Goretti.

Rev.investigium:cienc.soc.hum./Pasto-Colombia/Vol. VI/No. 2/pp.1-158 /Julio - Diciembre 2015/ISSN:2216-1473 e ISSN: 2357-5239

Revista Investigium IRE: Ciencias Sociales y Humanas
Vol. VI. No. 2. julio - diciembre de 2015
ISSN impreso: 2216 -1473. ISSN electrónico: 2357-5239

Publicación científica que contribuye a la divulgación y difusión de conocimiento de carácter científico e investigativo en Ciencias Sociales y Humanas desde la Institución Universitaria Centro de Estudios Superiores María Goretti (I.U. CESMAG).

Directora

María Eugenia Córdoba

Editora

Cristina Romero Chaves, revistainvestigaciones@iucsmag.edu.co
ocromero@iucsmag.edu.co

Coeditoras

Diana Cristina Córdoba Cely, dccordoba@iucsmag.edu.co
Sandra Lucía Bolaños Delgado, sbolanos@iucsmag.edu.co
Adriana Lagos Mora, alagos@iucsmag.edu.co
Carmen Cecilia Cabeza Cortés, cccabezas@iucsmag.edu.co

Asesores editoriales

Armando José Quijano Vodniza

Gestor de soporte

Diego Martínez Hernández, dmartinez@iucsmag.edu.co

Comité Editorial

Mauro Benente, Universidad de Buenos Aires (Argentina).
Jesús Jornet Meliá, Universidad de Valencia (España).
Joan Bau Martínez, Universidad Politécnica de Valencia (España).
Jesús Redondo Pacheco, Universidad Pontificia Bolivariana (Colombia).
Melquicedec Lozano Posso, Universidad Icesi (Colombia).
Tania Bolaños Enríquez, Universidad Cooperativa de Colombia (Colombia).
Gerardo León Guerrero Vinuesa, Universidad de Nariño (Colombia).
Esperanza Agreda Montenegro, Universidad Antonio Nariño (Colombia).
José Edmundo Calvache López, RUDECOLOMBIA (Colombia).

Comité Científico

Diana Mata Codesal, University of Sussex (Reino Unido).
Luz María Lepe, Universidad Autónoma de Querétaro (México).
Jorge Larrosa Bondía, Universidad de Barcelona (España).
José Luís Leal Espinoza, Universidad de Alicante (España).
Guillermo Campoy Menéndez, Universidad de Murcia (España).
Cándido Inglés Saura, Universidad Miguel Hernández de Elche (España).
Roberto Ramírez, Universidad de Nariño, (Colombia).
Javier Taborda Chaurra, Universidad de Caldas (Colombia).
Susana Borrás Pentinat, Universidad Rovira y Virgili (España).
Celina De Jesús Trimiño Velásquez, Universidad Pedagógica y Tecnológica de Colombia (Colombia).

Corrección de Estilo

Carlos Arcos Guerrero

Título

Revista Investigium IRE: Ciencias Sociales y Humanas.

Título Abreviado

Rev.investigium:cienc.soc.hum.

Periodicidad: Semestral

Formato: 20 cm x 26 cm

Traductores

Rosa Mila Ibarra Cadena (Inglés)
Francisco David Delgado Montero (Portugués)

Diagramación

James H. Montilla

Maquetación e Impresión

Editorial Kimpres SAS
Calle 19 sur No. 69C-17 Bogotá - Colombia
PBX: 4136884 • info@kimpres.com

Diseño de Portada

James H. Montilla

Editorial

Editorial Institución Universitaria CESMAG
editorial@iucsmag.edu.co
www.iucsmag.edu.co/editorial

Suscripción

Precio Unitario Nacional: \$12.000
Precio Internacional: US\$6

Correspondencia y suscripciones

Institución Universitaria CESMAG
Vicerrectoría de Investigaciones
Carrera 20ª No. 14-54 Pasto- Nariño-Colombia. 520003
PBX: +572 7216535, Fax: +572 7212314

Canje

Biblioteca Remigio Fiore Fortezza
biblioteca@iucsmag.edu.co

Sitio Web

<http://investigiumire.iucsmag.edu.co>

Indexación

Actualidad Iberoamericana, COPAC, Copernicus, DICE, E-revistas, GALE, ERIH, Google Académico, ISBNdB, Latindex, Publindex -Colciencias (Colombia) Categoría C, Sholar China, WorldCat.

Nota legal

© Institución Universitaria CESMAG
Todos los derechos reservados bajo la licencia de uso y distribución Creative Commons Reconocimiento no comercial 4.0 Internacional



Cada autor es responsable del contenido de su respectivo artículo incluyendo cualquier cambio sugerido por el Comité Editorial y no compromete a la Institución Universitaria CESMAG. Se permite la citación y reproducción nombrando la fuente. Cualquier tipo de reproducción, transformación o distribución, se puede hacer con autorización de la Institución Universitaria o de su editorial.



Contenido

• PRESENTACIÓN	3
• EDITORIAL La comunicación y la evaluación científica en la emergencia de subjetividades productivas del profesor universitario. <i>Cristina Romero Chaves</i>	7
RESULTADOS DE INVESTIGACIÓN CIENTÍFICA	
• “Malentretados y prostitutas”. Prácticas judiciales contra la vagancia en el centro occidente de Colombia 1859-1876. <i>Edwin Monsalvo Mendoza</i>	12
• Propensión a asumir riesgos, tomar decisiones y tolerar la incertidumbre en los emprendedores de Valledupar. <i>William Hernando Rocha Jácome y Gilberto de Jesús Giraldo Gómez</i>	24
• Fundamentos del pensamiento de diseño. <i>Carlos Córdoba Cely, Javier Arteaga Romero y Harold Bonilla Mora</i>	38
• Implementación de un ambiente virtual de aprendizaje para la enseñanza complementaria del derecho en tiempo independiente. <i>Carlos Eduardo López Dávila</i> ..	51
• Intelectualidad orgánica y ejercicio docente: posibilidades de acción en la reducción de las desigualdades escolares. <i>Nilson Javier Ibagón Martín</i>	67
• Estrategia colaborativa en entornos tridimensionales como estrategia didáctica de aprendizaje de estructuras iterativas en programación computacional. <i>Javier Alejandro Jiménez Toledo, Cesar Alberto Collazos Ordoñez, Julio Ariel Hurtado Alegría y Wilson Libardo Pantoja Yépes</i>	80
• Aproximaciones a una perspectiva integral para el estudio de los movimientos sociales. <i>Alexis Vladimir Pinilla Díaz</i>	93
REVISIÓN Y REFLEXIÓN INVESTIGATIVA	
• La ciudad del siglo XXI: ¿ocupada o habitada? <i>Jorge Armando Arturo Calvache</i>	108
• GUÍA PARA AUTORES.....	128
• POLÍTICAS ÉTICAS	154
• COLABORADORES	156



ESTRATEGIA COLABORATIVA EN ENTORNOS TRIDIMENSIONALES COMO ESTRATEGIA DIDÁCTICA DE APRENDIZAJE DE ESTRUCTURAS ITERATIVAS EN PROGRAMACIÓN COMPUTACIONAL¹

Recibido: abril 09 de 2015/ **Revisado:** noviembre 18 de 2015/ **Aceptado:** noviembre 5 de 2015
Por: Javier Alejandro Jiménez Toledo², Cesar Alberto Collazos Ordoñez³,
Julio Ariel Hurtado Alegría⁴ y Wilson Libardo Pantoja Yépez⁵

Para citar este artículo/To reference this article/Para citar este artigo

Jiménez, J., Collazos, C., Hurtado, J. & Pantoja, W. (Julio- diciembre, 2015). Estrategia colaborativa en entornos tridimensionales como estrategia didáctica de aprendizaje de estructuras iterativas en programación computacional. *Investigium IRE: Ciencias Sociales y Humanas*, VI (2), 80-92. Doi: <http://dx.doi.org/10.15658/CESMAG15.05060207>

RESUMEN

Este artículo presenta una nueva propuesta didáctica que surge a partir de un proceso investigativo cuyo propósito planteado pretende facilitar el estudio de las estructuras iterativas en el área de los fundamentos de programación, ofreciendo al estudiante conceptos complejos en un contexto apropiado a su pre saber. La investigación se desarrolló bajo un enfoque cuantitativo con el método empírico analítico y cuyo diseño se realizó de manera cuasi experimental en dos universidades con dos grupos de control y dos experimentales de estudiantes de pregrado de primer semestre de Ingeniería quienes fueron evaluados mediante aplicación de post pruebas después de recibir o no tratamiento experimental el cual consistió en la aplicación de escenarios apoyados con procesos de realidad aumentada con objetos 3D en un contexto colaborativo. Los datos obtenidos fueron analizados con la distribución de probabilidad t de Student con la que se comprobó que la diferencia de notas entre grupo experimental y grupo de control para cada universidad participante es estadísticamente significativa, lo que concluyó el éxito del tratamiento experimental.

Palabras Clave: Aprendizaje colaborativo, estructuras iterativas, fundamentos de programación, realidad aumentada.

¹ Artículo derivado del proyecto de investigación "Propuesta metodológica basada en modelado gráfico para la enseñanza de los fundamentos de programación mediante un entorno colaborativo inmersivo 3D", inscrito en el grupo de investigación TECNOFILIA de la Institución Universitaria CESMAG y en el grupo de investigación IDIS de la Universidad del Cauca como tesis del programa de Maestría en Computación.

² Candidato a Magister en Computación Universidad del Cauca, Especialista en Docencia Universitaria, Ingeniero de Sistemas y Computación, Profesor tiempo completo, Programa de Ingeniería de Sistemas, Institución Universitaria CESMAG, Pasto (Colombia). Correo electrónico: jajimenez@iucsmag.edu.co.

³ PostDoctorado Ciencias de la Computación, Doctor en Ciencias mención Computación, Ingeniero de Sistemas, profesor tiempo completo, Departamento de Sistemas, Universidad del Cauca, Popayán (Colombia). Correo electrónico: ccollazos@unicauca.edu.co.

⁴ Doctor en Ciencias mención Computación, Especialista en redes y servicios telemáticos, Especialista en procesos para desarrollo de software, Ingeniero en electrónica y telecomunicaciones, profesor tiempo completo, Departamento de Sistemas, Universidad del Cauca, Popayán (Colombia). Correo electrónico: ahurtado@unicauca.edu.co.

⁵ Magister en Computación, Especialista en redes y servicios telemáticos, Ingeniero de Sistemas, profesor tiempo completo, Departamento de Sistemas, Universidad del Cauca, Popayán (Colombia). Correo electrónico: wpantoja@unicauca.edu.co.

Anexo 24





Corporación
Universitaria
Comfacauca

Unicomfacauca



Directora

Isabel Ramírez Mejía

Editor

Centro Editorial Unicomfacauca

Comité Científico

Pedro Gonzáles del Foyo
Fabricio Junqueira
Carlos Humberto Mora
Madeleine Lita Barriga

Comité Editorial

Isabel Ramírez Mejía
Ricardo Adrián Gonzáles
Alejandra María Rodríguez
Victoria Eugenia Patiño
Helmer Paz Orozco
Diego Andrés Campo
Blanca Lilia Bonilla

Rectora

Isabel Ramírez Mejía

Vicerrectora General

Victoria Eugenia Patiño

Vicerrectora Académica

Maryem Rocío Ibarra

Directora Administrativa y Financiera

Gaby Edith Tintinago

Directora de Planeación

Diana Ximena Sanchez

Director de Proyección Social

Francisco Eduardo Bonilla

Diseño y diagramación

Diego Fernando Chávez

Redacción, suscripción y canjes

Corporación Universitaria Comfacauca
Revista I+T+C: Investigación, Tecnología y Ciencia
Calle 4º N° 8-30 Tel. (5-72) 822 05 17 ext. 122
Popayán, Cauca, Colombia
E-mail. revistaitc@unicomfacauca.edu.co
ISSN 1909-5775

Nota de responsabilidad frente a los artículos
Los conceptos y opiniones expresadas en los artículos
son responsabilidad exclusiva de los autores y
no representan necesariamente la opinión de la
Institución Universitaria Tecnológica de Comfacauca

Contenido

4 EDITORIAL

SECCIÓN INGENIERÍAS

- | | | | |
|----|---|----|--|
| 7 | Propuesta metodológica de implementación del gobierno de las tecnologías de la información, mediante el modelo GTI4U, para universidades en Colombia: caso Corporación Universitaria Comfacaucá | 15 | Aprendizaje colaborativo con técnicas de realidad aumentada como estrategia didáctica para la enseñanza de estructuras iterativas en fundamentos de programación de computadores |
| 23 | Influencia de la medición en la aplicación del control estadístico de procesos en el contexto del software | 33 | Uso de herramientas computacionales en entornos colaborativos como base para el desarrollo de un modelo ubicuo de aprendizaje asistido por computador |
| 43 | Propuesta para caracterizar el diseño de juegos corporativos soportada en aspectos de la experiencia de usuario relacionada con la satisfacción en uso | 49 | Arquitectura Orientada a Servicios – SOA, aplicada a la industria |
| 57 | Comprendiendo los procesos de abstracción computacional en los niños: un estudio de caso exploratorio | 67 | Aproximación de diseño emocional centrado en el usuario para entornos virtuales de aprendizaje |
| 75 | WhatsApp: huyendo de la voz hacia el “supertexto” | 83 | La familia y la escuela en la colonización del imaginario alimentario en la edad escolar (6 a 12 años) |
| | | 95 | Experimentación del e-portafolio en el ámbito universitario: usos de Mahara-Moodle |

AUTOR INVITADO

107

Modelo para a implantação de ferramentas tecnológicas nas escolas de medicina

123

GUÍA DE AUTORES

Aprendizaje colaborativo con técnicas de realidad aumentada como estrategia didáctica para la enseñanza de estructuras iterativas en fundamentos de programación de computadores

Javier A. Jiménez Toledo
Grupo Tecnofilia
I.U. CESMAG
jajimenez@iucsmag.edu.co

Cesar A. Collazos
Grupo IDIS
Universidad del Cauca
ccollazo@unicauca.edu.co

Julio A. Hurtado
Grupo IDIS
Universidad del Cauca
ahurtado@unicauca.edu.co

Libardo Pantoja Yépez
Grupo IDIS
Universidad del Cauca
wpantoja@unicauca.edu.co

Fecha Recepción: 24/10/14 - Fecha Aprobación: 4/12/14

Resumen: Este artículo, resultado de investigación, presenta la experiencia de validación de un modelo de aprendizaje colaborativo apoyado con técnicas de realidad aumentada como estrategia didáctica para la enseñanza de las estructuras iterativas, con estudiantes de primeros cursos de fundamentos de programación en instituciones de educación superior de modalidad presencial. El diseño de investigación planteado evaluó el tratamiento experimental, basado en aprendizaje colaborativo, cuyos resultados generaron interesantes indicadores de rendimiento académico con altos niveles de expectativas de aprendizaje en los estudiantes.

Palabras clave: Aprendizaje colaborativo, fundamentos de programación de computadores, realidad aumentada.

Abstract: This paper presents the experience of validating a model supported collaborative learning with augmented reality techniques as a teaching strategy for teaching students the cyclic structures of first programming fundamentals courses in institutions of higher education modality. The proposed research design evaluated the experimental treatment based on collaborative learning, the results generated interesting indicators of academic achievement with high expectations for student learning.

Keywords: Collaborative learning, fundamentals of computer programming, augmented reality, cycles.

Introducción

La programación de computadores es un área de conocimiento donde confluyen conceptos y habilidades esenciales para la práctica de la programación independiente del paradigma (Association for Computing Machinery, 2009). Pese al estado tecnológico actual y la utilización del software en todos los campos de actuación humana, existen algunos problemas relacionados con la fundamentación de los futuros constructores de software, debido a una cantidad de situaciones que van desde la carencia de habilidades para programar, el desconocimiento de la materia y hasta la carencia de disciplina en programación (Oviedo, 2002). Dichos inconvenientes se gestan desde el primer curso de programación recibido, el cual es clave en el proceso de

formación que tendrá un desarrollador de soluciones software en su etapa de aprendizaje y más tarde en su vida profesional (Hernández, Jiménez, & Martínez, 2012).

La enseñanza de los fundamentos de programación ha llevado a muchas reflexiones por las etapas que de acuerdo a la evolución de sus propios paradigmas se han generado. Establecer relaciones entre la evolución del pensamiento y los paradigmas de programación permite encontrar algunas relaciones causales entre cada uno de los paradigmas y la forma de modelar y enseñar soluciones computacionales (Buriticá & Ivan, 2012). Es así como en los primeros cursos algunos docentes inician la enseñanza de los fundamentos de programación, independiente del paradigma seleccionado (procedural, orientado a objetos,



CICOM 2015

**5° Congreso Internacional de Computación
Colombia - México
XV Jornada Académica en Inteligencia Artificial**

La Universidad Distrital Francisco José de Caldas y la Universidad Autónoma de Guerrero

CERTIFICA QUE:

JAVIER ALEJANDRO JIMÉNEZ TOLEDO

Presentó la Ponencia: "RECOMENDACIONES PARA LA CONSTRUCCIÓN DE ENTORNOS INMERSIVOS 3D UTILIZANDO CONCEPTOS DE HUMAN COMPUTER INTERACTION", en el 5° CONGRESO INTERNACIONAL DE COMPUTACIÓN COLOMBIA - MÉXICO, realizado los días 24, 25 y 26 de septiembre del año 2015 en Cartagena de Indias - Colombia

Nelson Becerra Correa
Presidente CICOM 2015
Universidad Distrital Francisco José de Caldas - Colombia

Edgar Altamirano Carmona
Copresidente CICOM 2015
Universidad Autónoma de Guerrero - México



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS



Anexo 26



Otorgan el presente

Certificado

**Javier Jiménez Toledo, César Collazos, Julio Ariel Hurtado y
Wilson Libardo Pantoja**

**HCI y Diseño Basado en el Usuario para la construcción de
entornos inmersivos 3D**

Por la exposición de un *Poster* en el segundo *Congreso Andino de Computación, Informática y Educación* CACIED 2015, que tuvo lugar en la ciudad de Guayaquil - Ecuador los días 4, 5 y 6 de Noviembre del 2015 con una duración de 24 horas.

Guayaquil, 6 de Noviembre de 2015

M.Sc. Miguel Yapur
Decano de la Facultad de Ingeniería en Electricidad y Computación

Ph.D. Carlos Monsalve
Comité Organizador CACIED 2015

